Using Interpretable Machine Learning for Indoor CO_2 Level Prediction and Occupancy Estimation

Chika E. Ugwuanyi

Dr Marilyn Lennon Supervisor [1]

Dr Richard Bellingham Supervisor [2]

Department of Computer and Information Sciences
Institute for Future Cities

University of Strathclyde

PhD Thesis

Doctor of Philosophy

14 June 2021

Abstract

Management and monitoring of rooms' environmental conditions is a good step towards achieving energy efficiency and a healthy indoor environment. However, studies indicate that some of the current methods used in environmental room monitoring are faced with some challenges such as high cost and lack of privacy. As a result, there is need to use a method that is simpler, reliable, affordable and without any privacy issues. Therefore, the aims of this thesis were: (i) to predict future CO_2 levels using environmental sensor data, (ii) to determine room occupancy using environmental sensor data and (iii) to create a prototype dashboard for possible future room management based on the models developed for room occupancy and CO_2 prediction. Machine learning methods were used and these included: Gradient Boosting ensemble model (GB), Long Short-Term Memory recurrent neural network model (LSTM) and Facebook Prophet model for time series (Prophet). The sensor data were recorded from three different office locations (two test sites at a university and a real-world commercial office in Glasgow, Scotland, UK). The results of the analysis show that with LSTM method, a Root Mean Square Error (RMSE) (absolute fit of the model results to the observed data) of 0.0682 could be achieved for two-hour time interval CO_2 prediction and with GB, of 82% accuracy could be achieved for proposed room occupancy estimation. Furthermore, as the model understanding was raised as a key issue, interpretable machine learning methods (SHapley Additive exPlanation. (SHAP) and Local Model-agnostic explanations. (LIME)) were used to interpret room occupancy results obtained by GB model. In addition a dashboard was designed and prototyped to show room environmental data, predicted CO_2 levels and estimated room occupancy based on what the sensor data and models might provide for people managing rooms in different settings. The proposed dashboard that was designed in this research was evaluated by interested participants and their responses show that the proposed dashboard could potentially offer inputs to building management towards the control of heating, ventilation and air-conditioning systems. This in turn could lead to improved energy efficiency, better planning of shared spaces in buildings, potentially reducing energy and operational costs, improved environmental conditions for room occupants; potentially leading to improved health, reduced risks, enhanced comfort and improved productivity. It is advised that further studies should be conducted at multiple locations to demonstrate generalisation of the results of the proposed model. In addition, the end benefits of the model could be assessed through applying its outputs to enhance the control of HVAC systems, room management systems and safety systems. The health and productivity of the occupants could be monitored in detail to identify whether resulting environmental improvements deliver improvements in health and productivity. The findings of this research contribute new knowledge that could be used to achieve reliable results in room occupancy estimation using machine learning approach.

Contents

List of Figures

List of Tables

List of Equations

1

Introduction

Constantly monitoring and managing our indoor environments with regards to its air quality should be important to our daily lives because people spend about 90% of their time indoors. When our indoor environments are not properly monitored, it could affect the health of the room occupants as well as make the rooms unfit for its purpose. Using some established and interpretable machine learning (ML) methods in a unique way could be a good approach to help to predict and ultimately better manage indoor air quality for the well-being of the occupants. Furthermore, the method of using interpretable ML could help non-experts to understand how the ML model works. This thesis will explore the advance prediction of indoor CO_2. concentration levels and estimation of room occupancy by using time series ML model and ensemble regression model respectively. A room monitoring dashboard was then designed, prototyped and evaluated to examine the possible utility of room monitoring systems that are ML driven.

1.1 Problem Statement

Our indoor environment should be planned in such a way that it could guarantee occupants' health and overall well-being. Some of the factors that should be considered in the room planning are 1) maintenance of good air quality 2) rooms' management capacity and 3) energy efficiency. When these factors are not properly monitored, the well-being of occupants as well as the rooms' maintenance cost with regards to its energy supply could be adversely affected. These main factors and its associated problems will be discussed in the following sub-sections:

1.1.1 Adequate Air Quality

Some indoor environmental variables' levels such as CO_2 concentrations levels and temperature should be constantly monitored because they have human tolerance levels. For instance, a room that constantly have some of these variables' levels below or above the recommended standard could affect building material Künzel et al. [2005]. When there are constant wetting and drying of the material of a building, some harmful conditions such as moulds, dampness or premature failure of the building may begin to develop Künzel et al. [2005].

Furthermore, Allen et al. [2016], Gall et al. [2016] have shown that high indoor CO_2 concentrations in enclosed space without enough air exchange could cause various health related problems like sick building syndrome(SBS) symptoms, dampness, growth of fungi (moulds) and negative impact to human cognition (reasoning, judgement) and above all instant death when its level is as high as 40,000 parts per million (ppm) because of oxygen deprivation.

1.1.2 Rooms' Capacity Management and Energy Efficiency

Oftentimes when the thermal comfort of an indoor environment is inadequate, occupants of such places could be affected with some cold related illnesses such as cold stress and coughing Ferng and Li-Wen [2002], Rudnick and Milton [2003]. In the same vein, excessive supply of lighting, heating, ventilation and air conditioning into a room without fine-grained occupancy and indoor environmental variable information of that room could cause energy inefficiency Fasiuddin and Budaiwi [2011], Pérez-Lombard et al. [2008], Zhao and Magoulès [2012] . This also undermines fight for climate change and hinder both management and financial strategies, which sometimes results in costs that can never be recovered Law [2004]. Therefore, taking into consideration some of these factors in order to make our indoor environment fit for purpose and healthy for occupants is an important challenge.

1.1.3 Summary

Various studies have proposed the use of real-time indoor environmental monitoring systems and being able to detect the number of people in the room at all times in order to curb some of these challenges. Unfortunately, most of the commercially available environmental monitoring systems and room occupancy detection methods are faced with lots of drawbacks such as sensitivity to noise Ding et al. [2004], Kumar et al. [2012], Zhang et al. [2007]. , expensive and intrusive Chen et al. [2018b], Jiang et al. [2016].

A control technique for energy management known as Model Predictive Control (MPC) was implemented by Oldewurtel et al. [2012], Privara et al. [2011]. However, their methods causes request for more computational resources, large dependence on air temperatures and sunshine during heating, complexity of the system's building model and lack of consideration for occupants' presence. Moreover, most traditional HVAC control systems retain indoor thermal comfort notwithstanding if the occupants are present or not. Others use a setting logic known as "nighttime setback" Brooks et al. [2015]. This means that the control system is relaxed in the night when it is assumed that there is no one present in the room. This approach is unreliable because it could cause serious discomfort to occupants who might be present.

Other gaps in the literature are 1) lack of consideration for different room settings such as available mechanical or natural means of ventilation, variable number of occupants in a room and the outdoor environmental weather conditions. 2) most ML methods used for the occupancy estimations were not interpreted nor explained for non-experts' understanding and contributions. 3) some other means of detecting occupancy are expensive and intrusive for example the use of video cameras and mobile devices and 4) the actual time in advance that requires room occupant to be aware of the real-time CO_2

concentration levels were not made clear.

There are lots of methods available for estimating room occupancy and predicting CO_2 which has been used in the past in the literature, but this research will concentrate only on using the ML method because of its obvious advantage such as identifying trends and patterns in dataset, handling multi-dimensional and multi-variety dataset and continuous improvement in it wide applications. Predicting room occupancy offers a solution to meet this aim. ML could be used for this but the most suitable and best performing methods are not known yet. If this could be researched, then this could offer potential usable solutions for efficient and smarter room management.

1.2 Research Aim and Objectives

This research will focus on two main aims. 1) To explore how a commonly available environmental sensor data could be used to predict indoor CO_2 please see (Chapter 5) and estimate room occupancy via interpretable ML please see (Chapter 6). 2) to investigate the usability and utility of a visual dashboard using this data to help users manage rooms more effectively please see (Chapter 7).

The overall objectives of this research are divided into four main steps as follows:

1. To conduct semi structured interview with potential users bout their indoor environmental monitoring needs.

2. To analyse recorded environmental data from Netatmo sensor Online Repository [a] using ensemble regression and time series ML models in order to estimate indoor room occupancy in real time and forecast indoor CO_2 levels respectively.

3. To use explainable machine learning method to interpret the results of the estimated occupancy level so that non experts can be able to understand how the ML model predicted its results.

4. To design a prototype dashboard for indoor environmental monitoring and room occupancy management.

1.3 Research Questions

To achieve the above listed objectives, the following research questions and its corresponding questions will be answered respectively:

1. By using the sensed indoor environmental data, can ML be used to accurately and reliably predict indoor CO_2?

   - What advance time would the indoor CO_2 levels be predicted so that room managers could have enough time to act and prevent any issues that may arise.

2. By using the sensed indoor environmental data, can ML be used to accurately and reliably predict and interpret room occupancy?

   - Which ML method can be used to estimate room occupancy.

- How best can the ML model interpretability be evaluated in order to assess the interpretability of the ML model used for the estimation of room occupancy.
- Which of the interpretable ML method would be easy for non-experts to understand.

3. Can users see potential benefit for a dashboard for managing rooms that is based on interpretable ML methods from environmentally sensed data?

    - How will the dashboard be designed so that it can help in visualizing the levels of indoor environmental variables, the predicted indoor CO_2 levels and the estimated room occupancy.
    - How will the stakeholders' contributions be identified and incorporated in the design phase of the system?
    - What are the functional requirements that would help the system users to perform their task effectively?
    - What software tool will be used to test the dashboard system for easy evaluation?
    - How will the designed system be evaluated in order to determine if the system can perform its required tasks when developed.

1.4 Research Contributions

To the best of the researchers' knowledge, some of the application of these research methods are novel in this particular area of research (predicting indoor CO_2 concentration levels, estimating room occupancy and some functional features of the dashboard to be designed). Therefore, the application novelty of this research methods are listed as follows:

1. Though ensemble and other ML methods have been used in the past for room occupancy estimation however, none of the ML models were interpreted with interpretable ML method during their studies.

2. Though indoor CO_2 concentration levels have been predicted in the past however, previous indoor CO_2 studies concentrated on using the CO_2 levels to forecast indoor air quality as such did not try to predict future CO_2 concentration levels which could help to forestall any health hazard because of its potential to do so.

3. The use of prediction interval approach for the room occupancy estimation has not been done in the past studies.

4. The identified space where the proposed room occupancy method is applicable are 1) indoor space with between 10-100 sitting capacity, 2) a space where HVAC is constantly on between 50 - 100% of the time and 3) a space where its windows and doors are constantly closed most of the time.

5. Three different studies were conducted so that rooms can be better monitored and managed easily in one place and timely by room occupants and mangers. This combined approach has not been done before in the literature.

6. Interpretable ML was applied only to the occupancy study because it is easier to understand the pattern of people count with the help of a model than CO_2 level that doe not have definite pattern.

7. Though different software systems for environmental room monitoring have been designed in the past. However, to the best of my knowledge none of the systems has potential to estimate room occupancy and predict CO_2 as its functionality.

Therefore, it is believed that once this research methods are followed and implemented, some of the current gaps in the literature could be bridged. This will in turn contribute towards the management of our indoor environment for everyone's overall well-being.

2

Background

2.1 Introduction

The previous chapter has noted that there is need for our indoor environment to be constantly monitored and that the room occupants and managers should be aware of their indoor environmental conditions at all times so that they can stay healthy indoors. This chapter provides the background of previous research done in indoor environmental monitoring with respect to 1) indoor air quality (Indoor Air Quality (IAQ)), 2) thermal comfort and 3) energy efficiency. This chapter also discusses various patent apparatus and methods that were previously invented in order to prevent and/or control adverse effect that could emanate from inadequate indoor environmental conditions. This chapter did not discuss various ML methods used in this research because they were discussed in its corresponding study chapters. This chapter ends with the discussion of room capacity management.

2.2 Air Quality and Comfort in Indoor Work Environment

People spend up to 90% of their time indoors Marques and Pitarma [2016]. Therefore, being aware of their health as it concerns indoor air quality should be important in their daily lives. World Health Organization (WHO) has defined the concept of achieving health and well-being in our various indoor environments as one's state or perception of physical, mental and social comfort rather than non-existence of disease in those indoor places Rocca [2017]. According to Rocca, these human perception depends on four environmental factors such as thermal environment, lighting, noise and indoor air quality. Hence, these factors have contributed in making occupants' indoor health an important topic as long as their health and well-being is concerned. Rocca also observed that when these environmental factors are not adequately supplied, it could portend danger to occupants' health. A pictorial representation of environmental factors and how they relate to human health and comfort in an indoor environment are shown in Figure 2.1. Some health problems brought about by the inadequate supply of these mentioned environmental factors are cold stress, noise and air pollution. As a result it is believed that indoor air poses much health risk than the outdoor air Sundell [2017].

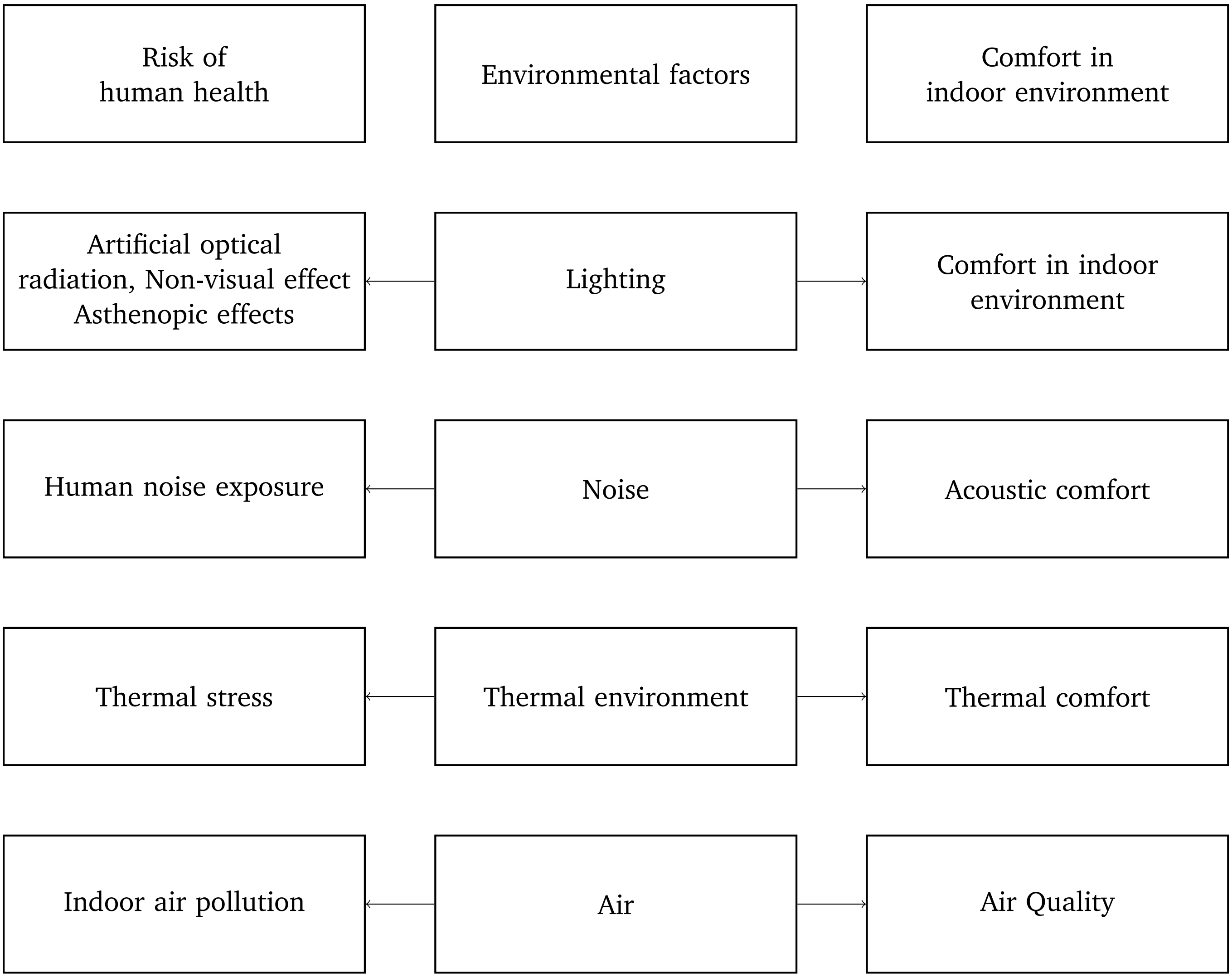


Fig. 2.1. Factors affecting human health risk and comfort in indoor environments
Source: Rocca [2017]

Seppänen and Fisk observed that there is a relationship between ventilation systems and the health of occupants in Seppänen and Fisk [2004]. Seppänen and Fisk also noted that there is complexity in using ventilation system to dilute the air pollutants generated in rooms. The reason is because the limits of all the pollutants and the essential ventilation rates based on the pollutants' concentrations are unknown. Poorly designed indoor ventilation system could introduce harmful substances into the indoor environment. One of such substance is moisture which decays the building's envelope. When occupants are exposed to indoor pollutants, there are associated negative effects such as undesirable odour, moulds, respiratory diseases among children and mucous membrane irritation mostly caused by formaldehyde Seppänen and Fisk [2004].

Additionally, poorly designed mechanical ventilation system like HVAC in a non-residential environment could increase symptoms of sick building syndrome (SBS) by

30% and 200% in the rooms where they are used Seppanen and Fisk [2002]. In many cases when the ventilation rate is below $10Ls^{-1}$ per person, the prevalence of SBS symptoms increases even more Wargocki et al. [1999]. Whereas increase in ventilation rates between $10Ls^{-1}$ to $20Ls^{-1}$ per person significantly reduces the symptoms of SBS in a non residential environment that are built with the current building standards. An obvious reduction in the symptoms of headache and lack of concentration when the ventilation rate was raised from 9.8 to 22.7 Ls^{-1} per person was observed in a study Wyon and Wargocki [2003]. Therefore, in order to improve the air quality of various indoor environment, providing a good means of ventilation that are adequate for all occupants is important to occupants' productivity, health and well-being.

2.2.1 Monitoring Indoor Air Pollutants

Some indoor air pollutants that contributes to poor IAQ are chemicals used in producing building materials such as formaldehyde, bioeffulents, tobacco or charcoal smoke and high CO_2 concentrations. According to WHO, these air pollutants causes various health problems that leads to 4.6 million deaths yearly [as cited in Rajasegarar et al., 2014]. Some of the health problems associated with these indoor pollutants are lung cancer, asthma, bronchitis, airborne infections, fungi like moulds and sick building syndrome symptoms. Moreover, evidence has shown that the sources of excessive morbidity in various indoor environments are as a result of poor indoor air quality (IAQ). For example, Sundell [2004] observed that there are about 2,000,000 deaths yearly (mostly children and women) in developing countries due to indoor burning of biomass as a means of cooking. Tracking and diluting these indoor pollutants is essential to occupants' overall well-being and comfort.

Fortunately, the emergence of Internet of things (IoT) has brought about a sequence of events where ubiquitous smart sensors can now monitor, communicate and transfer remotely some of these indoor environmental pollutants' data to one another without the help of humans Moatamed et al. [2016]. Most of the smart sensors also track volatile organic compounds (VOCs) and inorganic compounds via a technique known as remote data collection which could be analysed for insights Runge et al. [2002]. These techniques have no doubt helped in monitoring and controlling air pollution in both open and enclosed spaces, but how reliable, effective, affordable and non-complex some of these systems are remains a key challenge to both occupants and estate managers.

Before the advent of wireless air quality monitoring sensors, fixed monitoring devices were used. These fixed devices were negatively affected by some factors like weather spikes, movement of people, and noise from moving objects. Currently, a wireless sensor network (WSN) with a transmitter node is being used to monitor, record and transmit data to a base station over a web so that it can be assessed through the computer Saad et al. [2013]. For the sensor to work in different rooms, auto-calibration technique is used for better data accuracy Römer et al. [2005]. Ensuring auto-calibration of the device in a new place have reduced biased in the results of the recorded data. It is in the light of this improved technique, that this research study adopted the use of WSN for improved measurement of indoor environmental variables in the place where the research study was conducted.

2.2.2 Thermal Comfort and Its Impact to Climate Change

The current British thermal comfort standard (BS EN ISO 7730) is determined by two factors namely, the metabolic rate (ISO 8996) and clothing insulation (ISO 9920) Olesen and Parsons [2002]. The thermal comfort indices was developed by Fanger et al. [1970], which is based on predicted mean vote (PMV) and predicted percentage of dissatisfied (PPD). The approach of trying to seal a building in order to control the indoor air temperature has resulted in sharp criticism been directed at the thermal comfort policy Chappells and Shove [2005]. Chappells and Shove also observed that these practises have lead to the consideration of broader international perspective on the thermal comfort policies which its acceptable standard have caused heavy energy consumption in heating and cooling of various indoor environments. As a result, there is greater danger to global warming and climate change Chappells and Shove [2005]. Therefore, increased dependence on air-conditioning will further increase the energy demand and consequently increase the CO_2 emissions.

Research studies in Pérez-Lombard et al. [2008], Zhao and Magoulès [2012] indicates that the building sector in European countries accounts for about 40% of the total energy consumed by all the sectors. This figure makes the building sector higher in terms of energy consumption than compared with the other sectors such as transportation and industrial sector. The steady increase in energy usage that are mostly found in the commercial buildings are caused mainly by Heating and cooling systems Fasiuddin and Budaiwi [2011].

Due to the danger posed by increased energy consumption in the industrialized countries such as UK, an energy white paper on the target of UK's housing stock in terms of CO_2 emission was published by the UK government Johnston et al. [2005]. According to Chappells and Shove [2005], for Trade and Industry [2003], the energy white paper states that:

> *"Energy is often wasted because of poorly insulated buildings or where heating, ventilation and air-conditioning and lighting are poorly controlled."*
>
> – Department of Trade and Industry, 2003, p.32

Meanwhile, the UK's energy policy target is to achieve 60% CO_2 emission reduction by the year 2050. In order to achieve this set target, a bottom-up energy and CO_2 emissions model was developed by Shorrock et al. [2001] using BREHOMES[1]. According to Johnston et al. [2005], it is technically feasible to achieve a reduction of 60% CO_2 emissions from the UK housing stock by the year 2050 using the current model. However achieving such feat will be technically demanding due to the range of technical measures such as increase in population, number of UK households, thermal comfort standard and service standards that occupants expect. Therefore, it is important that thermal comfort of occupants and energy usage are managed together in order to avoid energy waste which helps to mitigate the effects of climate change.

2.3 Indoor CO_2 Concentrations and Standards

Over the years, indoor ventilation has been recognized and acknowledged as a major health factor affecting occupants of different buildings. As a result some initial recommendations on buildings' ventilation were published in the 19th century. Ever since then, these recommendations have been transformed into standards in the 20th century by the advent of American Society of Heating, Refrigerating and Air-conditioning Engineers(ASHRAE) 62, with the first version been published in 1973 Persily [2015]. Henceforth, ASHRAE 62 has continuously been reviewed and modified so that the standards can reflect on the global environmental conditions and impact as they change.

Typical outdoor CO_2 concentration level is between 250 - 350 parts per million(ppm) with ambient air, whereas a typical considerable indoor CO_2 concentration level could range from 350ppm - 1000ppm with good air exchange see Figure 2.2. According to American Society of Heating and conditioning Engineers [2016], occupants usually complain of some health symptoms like drowsiness and poor air when indoor CO_2 concentration is between 1,000ppm and 2,500ppm. Low indoor CO_2 concentrations levels mainly found in typical office buildings might not pose a direct health risks to its occupants, but it could be used as an indicator of occupants' odours and physical activity levels Kajtar et al. [2006].

However, previous researches Kajtar et al. [2006], Milton et al. [2000], Shendell et al. [2004], Vehviläinen et al. [2016], still believe that CO_2 concentration levels between 350ppm and 1000ppm, higher than the ones found in a typical indoor settings could also cause drowsiness, headache, poor work performance and mucosal irritation. Nonetheless, the accepted levels of CO_2 concentrations in an office setting, has the maximum CO_2 exposure limits for a 8hr day work as 5,000ppm as a time weighted average (TWA). Furthermore, CO_2 concentrations exposure levels greater than 40,000ppm could cause serious health risks which might lead to permanent brain damage, state of coma and instant death Lipsett et al. [1994]. Nevertheless, CO_2 concentrations up to 5,000ppm in a typical indoor environment could be assumed not to have any direct negative impact on the occupants.

The recommendations of indoor CO_2 concentration levels by ASHRAE 62 are shown in Figure 2.2. The figure shows from normal to risky exposure levels. In order for occupants to avoid or control any possible rise in their indoor CO_2 concentration levels, they should be constantly shown the levels. The constant reminder of the occupants indoor CO_2 levels could prevent any health concerns that often comes with its excessive exposure.

Indoor Sources of CO_2

CO_2 emissions from human activities causes increase in CO_2 concentrations in the atmosphere. One of such activities is burning of fossil fuel and is estimated at 87% compared to other human activities Le Quéré et al. [2012]. Some of the poorly ventilated indoor areas where CO_2 concentrations could mostly build up, either as a result of indoor human activities or as a result of environmental conditions are 1) Long-distant urban transport buses, 2) Industrial areas, 3) Residential areas, 4) Non-industrial areas (open spaces like offices) and 5) Stop and Go urban buses Energy repository, Le Quéré et al. [2012]. However, this research will not be investigation some of these areas.

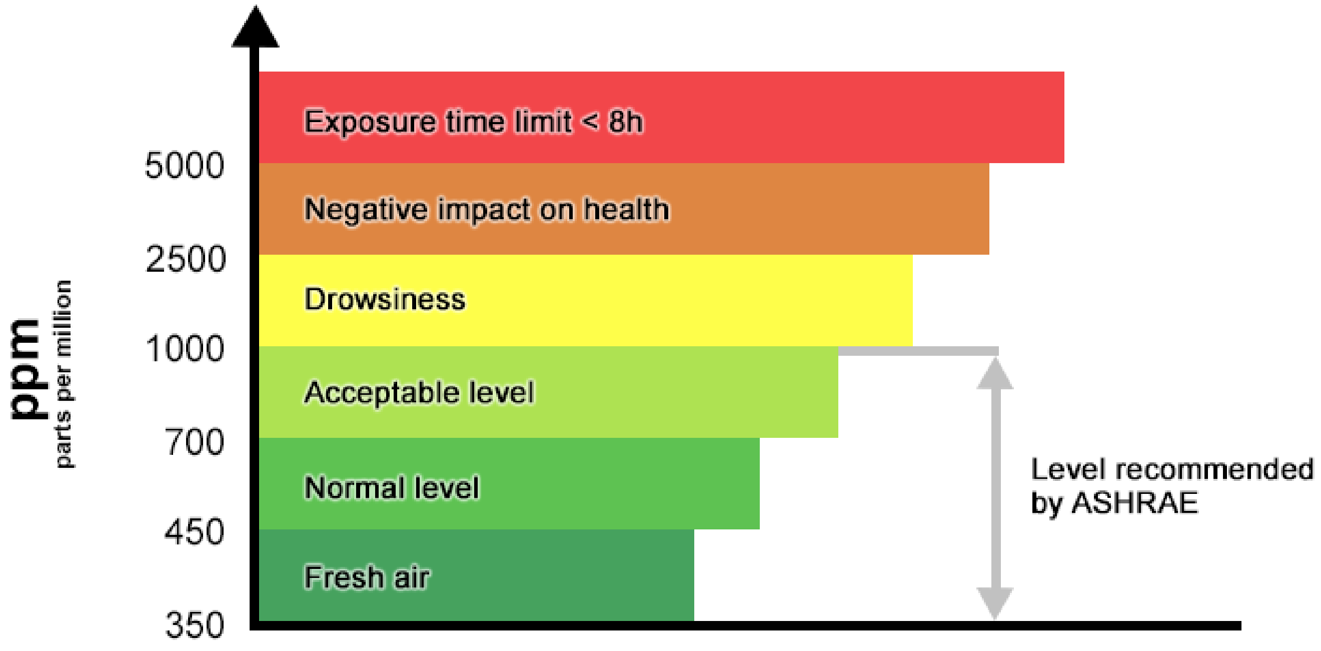


Fig. 2.2. Air Quality by ASHRAE
Source: American Society of Heating and conditioning Engineers [2016]

2.4 Monitoring Systems and Control Methods

To ensure the continuation of economic growth and well-being of the environment, the efficiency of the indoor energy consumption and indoor environmental conditions should be handled with serious concern Sbci [2009]. Part of the reason is because indoor environmental condition plays important role in the productivity level of humans' activities Nicol and Humphreys [2002]. These indoor environmental conditions could be humidity, temperature, sunlight, etc., and the accepted levels for environmental conditions all differ based on the prevailing circumstances and need.

In addition to the monitoring of some of the environmental conditions, an environmental greenhouse gas known as CO_2 concentration is also monitored because it is believed to be air pollutant which are mostly found in a poorly ventilated indoor environment. Undue exposure to high level of CO_2 concentration causes some adverse effect to humans. Some of these negative effects are discussed in Section 2.3 above. United nations environment programme (UNEP) reports that buildings use about 40% of the global energy which is responsible for about one-third of the green house emission Sbci [2009].

Based on the successes recorded in the past on environmental monitoring systems and the need for preventive and control mechanism to be used to regularly monitor our various indoor environmental conditions, similar approaches and methods are to be adopted in this research for the design of the environmental monitoring system proposed in this study.

2.5 Room Management and Occupancy Detection Methods

2.5.1 Indoor Room Capacity Management and Pitfalls

Indoor places such as hotel rooms, offices and hospitals without adequate occupancy information could cause capacity management issues, hinder economic activities and tourism in terms of administrative policies Tsai and Gu [2012]. While undercapacity could bring about economic losses, overcapacity could bring about unhealthy competition with declining profit. Thus, neither overcapacity nor undercapacity is desirable in room management Law [2004], Zhang et al. [2009].

A study Romeo [1997] (as cited in Tsai and Gu 2012) noted oversupply and distress of U.S. hotel rooms in the 1980s in addition with the 1990-1991 economic recession where most hotels and motels were made to operate below the break-even point. This resulted in the bankruptcy of two-thirds of the U.S hotels. Other problems that are caused by lack of adequate room capacity information are 1) in the room/offices; poor indoor air quality, and 2) in hospitals; poor patient care that often leads to high mortality rate in the case of hospitals' Emergency departments (ED) McCarthy et al. [2008].

Furthermore, in most U.S EDs, it has been reported that due to overcrowding in the EDs, most patients either overstays, leaves the ED without been seen or leaves without authorization Wang et al. [2017]. When ED overcrowding happens, some developed ED crowding tools Khalifa [2015] is often utilized in order to reduce the ED crowding. However, some of these tools either lack the ability for early crowding detection or the ideal measurement frequency of the ED crowding due to inconsistencies in the crowding measurement tools Wang et al. [2017].

Contrary to the believe that ED volume is extremely unpredictable, daily prediction of number of admissions is remarkably achievable Salway et al. [2017]. Therefore, according to Salway et al. early information on ED occupancy rate could help decrease overcrowding in the EDs, increase quality of care, reduce medical errors and consequently reduce the patients' mortality rate.

Other areas where reliable room occupancy information is important are lecture halls and meeting rooms that require occupancy slot allocation and use of HVAC systems and lighting appliances based on availability. When there are no fine tuned occupancy information for the relevant spaces, there could be waste of maintenance costs which might lead to excess energy usage because of unused space. Another cause of excess energy usage is leaving electrical appliances on when rooms are unoccupied. Therefore, having occupancy information for adequate room management is a good step towards achieving energy efficiency by cutting waste.

To maintain these economic, tourism growth, sustainable growth, decent profitability, and efficient energy usage, a healthy and reasonable room occupancy information is important to stakeholder so that they can plan its capacity carefully based on demand Tsai and Gu

[2012].

2.5.2 Non Data Analysis Method for Room Occupancy Detection

There are various ways of detecting occupancy other than machine learning methods. Though these methods are beyond the scope of this study, but some of these methods will be reviewed in this section.

An image based occupancy detection method was discussed in Petersen et al. [2016] and Erickson and Cerpa [2010]. Petersen et al. method detected ground truth entrance of room occupant by using image processing technique such as simple line crossing and infrared depth frame images were used. Three lines CFV, PVL and SVL receptively were monitored for potential occupant crossing. For the count to occur, the occupant must cross atleast CFV and SVL. Though this method was successful in detecting the number of people in the room. However, one major drawback with this method was counting object as occupant rather than only humans. Another method is the use of smart meter to count room occupants via power consumption Chen et al. [2018b]. In this method occupancy was detected based on the interaction between the occupants and the appliance. In other word, the building occupancy can be inferred through power consumption. However Chen et al observed that the relationship between the occupancy and energy consumption is not simple. In the same vein, the use of energy data has been used in Chen et al. [2013]. In this method, it was observed that electrical pattern usage often changes when there was a change in the number of room occupant as a result of their interaction with electrical loads. However, this method is unreliable because not all occupant would be using their electrical appliance when indoor.

Other method of occupancy detection is the use of surveillance video Zou et al. [2017] and the use of location sensing prototype system known as LANDMARC Ni et al. [2003]. This method used radio frequency technology known as Radio Frequency Identification (RFID) to locate any object inside the building. This method is based on electromagnetic signal detection. Both active or passive mode could be sensed. Despite its potential, however, these methods cannot guarantee sensing only human by having extra fixed location reference tags to help location calibration. Also, this method caused inconsistency result of delay in signal sending caused by different levels of power. Another study Bahl and Padmanabhan [2000] used a technology based on radio frequency (RF) signal. These methods could be certainly affected by electromagnetic conditions Yang et al. [2016]. Additional drawback is the intrusive nature of the methods. Therefore, it is important that non intrusive methods are considered in order to bridge some of the gaps in the literature.

2.5.3 Data Analysis Method for Room Occupancy Detection

In order to solve the privacy issue that often arise when room occupancy i detected with cameras, infrared etc., data analysis were used. More detailed review has been discussed (please see Section 3.3.2).

A plug and play occupancy detection method bed on trajectory sensor data was discussed in Pedersen et al. [2017]. In this method climate sensor data such CO_2, relative humidity, temperature, noise and volatile organic compound (VOC) were used. Before the plug and ply method w used, machine learning method known as exponential moving verge (EMA) was used to reduce the pike in the CO_2 data. This method only calculated the binary occupancy value of the room. Though this method was effective in detecting when the room was occupied or not, however, it recorded false negative and false positive in it findings. Another occupancy detection and behaviour study based on data analysis was discussed in Zhao et al. [2018]. Here time series methods known as recurrent neural network and support vector regression(RNN and SVR) were used on various inputs to a building. Such inputs are temperance, HVAC, lighting and occupancy data to achieve accuracy. One major drawback of this method was that its accuracy depend on the feature selection of the variable inputs used.

Other machine learning method for room occupancy detection has been discussed in Section 3.3.2. Occupancy study Kröse et al. [1993], Zhao et al. [2018,?] done previously in the literature showed that detecting room occupancy by the use of data analysis showed that more reliable occupancy result were achieved by the use of 1) environmental sensor and 2) indoor environmental data such temperature, humidity and CO_2. Other environmental data are HVAC information and lighting. These findings aided us to consider data analysis option via machine learning in order to help us answer our research question. In addition to machine learning, environmental data were used in this research.

2.6 Conclusion

In this chapter various issues relating to poor monitoring of indoor environment and methods of occupancy detection were discussed in this chapter. These issues are thermal comfort of room occupants and how it can be maintained. Other issues are health risks associated with some certain levels of indoor CO_2 concentration levels and the causes of its rise, including places where they are prone to occur. This chapter also discussed the advantages of knowing the exact number of people in the room with respect to energy savings and room capacity management. The intrusive nature of some of the room occupancy detection techniques and its often unreliable results. Furthermore, different non-machine learning methods of detecting room occupants, how the different methods affects the privacy of occupants, how unreliable some of the methods could be and the justification for choosing ML method and indoor environmental variables were also discussed in this chapter. Next chapter will now discuss the literature review and the different sources where the search were conducted including how it helped to formulate the research questions stated in Section 1.3.

The following chapter will discuss the scoping review of application of the ML methods in room occupancy detection and the prediction of indoor CO_2 concentration levels and their challenges. The review only focussed on the actual ML methods used in this research.

3

A Scoping Review of Machine Learning Methods for Room Occupancy Estimation, Indoor CO_2 Prediction and Environmental Monitoring Systems

3.1 Introduction

The use of machine learning (ML) methods for room occupancy research and the routine monitoring of indoor environmental conditions are currently gaining momentum Chen et al. [2018b], Pedersen et al. [2017], Uziel et al. [2013]. For instance, in the area of room occupancy research, there are progress in three different directions namely: (1) occupancy detection (OD), (2) occupancy estimation (OE) and (3) occupancy prediction (OP). The first direction (OD) consists of ML methods being used to know whether an occupant is present in the room or not. The second direction (OE) is for ML methods that are able to roughly determine the exact number of occupants in the room and the third direction (OP) uses ML to predict when the room will be occupied and how many people will likely occupy the room in the future.

The contribution herein is to review these methods and systems mostly from the ML perspectives and to examine some open limitations using any of the ML methods. This chapter additionally attempts to provide a representative of CO_2 prediction study that was done in the past with regards to IAQ. This chapter ends by highlighting progresses made in this research area, current challenges being faced and the gap that this research study will address.

3.2 Methods

3.2.1 Goal

As previously stated, this review focused on identifying and summarizing the various gaps in ML methods of estimating room occupancy, predicting indoor CO_2 concentration and methods mainly used for designing indoor environmental monitoring software systems. Carefully identifying these gaps will help in designing the right study that will answer some of the problems identified.

3.2.2 Searched Sources and Strategy

The review of the literature conducted in this research adopted a scoping literature review approach, in which a holistic in-depth search of the previous and current methods relating to the proposed research questions were conducted. The databases searched comprises of articles and journals published in three different databases namely, Association for Computing Machinery (ACM) digital library, Institute of electrical and electronics engineers (Institute of Electrical Electronics Engineers digital library (IEEE)) digital library and Google scholar). The main reason for the choice of the three main databases was because they are known and reliable academic library for data science.

Two phases of search were conducted. The first phase was mostly to obtain the overall background and review of the subject of discussion. This was done in October 2017. The search terms used to query the three databases are shown in Table 1 for ACM and Table 2 for IEEE and Table 3 for Google scholar. Please see Figure 3.1. The search queries for google scholar database is slightly different from IEEE and ACM because it does not have an advanced search functions like others.

This review includes evidences of its applications, strengths, and limitations especially as it concerns indoor environments only. The search was augmented by a general internet search on United Kingdom (UK) and United States (US) government environmental regulation agency website focusing on key authors, titles and year of publication which returned a total of n=10 articles. Please see Figure 3.1.

Most of the papers originated in the United States, Europe or China. The references found from the search results were used in this thesis in order to acquire more information. A total of n=360 duplicates were removed from the phase one records, after which a total of n=1360 articles were left. The screened the remaining articles' abstracts and titles based on the inclusion and exclusion criteria shown in Table 3.1.

The second phase of this article search was done in December 2019 and February 2021. The reason for it was to obtain any additional new information that could assist in the study already conducted. The search was started by querying the same three databases with the same search terms shown in Table 1 for ACM and Table 2 for IEEE and Table 3 for Google scholar. However, a check was placed on the search in order to ensure that only the articles recently published between October 2017 to February 2020 were returned. This check also ensured that all the articles returned during the first phase were not added to the current search.

3.2.3 Inclusion, Exclusion Criteria and Analysis

As was previously stated, the review focussed on the articles that used ML to predict indoor CO_2 and room occupancy. In addition to that, the review also looked at various ways of monitoring indoor environmental variables through the use of software application and those studies that used explainable ML approach for their room occupancy study. Table 3.1 explains the inclusion criteria that were applied in this review. However, some of the articles were ineligible if they meet the following criteria:

- If the study was not in English.

- If the study was not about ML methods of estimating room occupancy.

- If the study used simulated dataset rather than real-world dataset.

- If the ML methods used for the study were not regression or time series problems as it relates to scores and interpretation.

- If the articles are of low quality without result output.

- If the study limitations were not explained or considered.

In Chapter 2, some original development studies were included so as to identify major areas such as causes of poor indoor air quality as it relates to indoor pollutants, diseases associated with high indoor CO_2 concentration levels in a poorly ventilated indoor environment, room monitoring systems, methods of occupancy detection and advantages of room occupancy estimation especially as it relates to energy efficiency and climate change. This inclusion was done in the background section so that the review chapter will be more of summary of the current methods that addresses the research questions.

The inclusion and exclusion criteria used in this research were done based on the procedure for performing systematic literature review for software engineering researchers proposed in Kitchenham [2004]. Kitchenham et al. stated that during the study selection procedure that forms part of the inclusion and exclusion criteria, it is important to do the selection criteria on the subset of primary studies conducted in the background section. This same methods was later adapted by Bacca et al Bacca Acosta et al. [2014] and other thousand researchers. Therefore, for the clarity and better understanding of this review, the prediction and estimation scores from the eligible studies were stated and analysed based on validations from various ML methods of estimating indoor room occupancy and indoor CO_2 concentrations. The results were compared based on their performances in their respective areas. A flow chart shown in Figure 3.1 gives an overview of how the final articles were selected for the review.

3.2.4 Study Selection

As shown in Figure 3.1, the study selection for this review was done in four different stages namely; identification, screening, eligibility and included. In the identification stage; two phases (Phase 1 and Phase 2) were involved. Phase 1 had a total of n=1720 articles that were returned from the three databases and internet search (please see Section 3.2.2). Phase 2 had a total of n=534 articles from the three databases to be screened. In the screening stage, among the n=1720 returned articles from Phase 1, n=360 duplicates were removed leaving a total of n=1360. Similarly, duplicates of n=504 were removed from phase 2, leaving a total of n=30 articles. After the removal of the duplicates from the two phases, a total of n=1390 articles were left to be screened. Each abstract of the articles n=1390 were screened, a total of n=1333 were excluded after the screening as a result of criteria listed in Table 3.1 which were derived based on problem intervention comparator (PICOS) search tool Methley et al. [2014] . In the eligibility stage, the author was left with a total of n=57 articles. These remaining articles were read

Criteria	**Description**
Interest	1**) Indoor environment:** Information included in this research mainly focussed on indoor environmental studies for CO_2 prediction. Few information about energy savings from room occupancy estimation were included. 2) **Sensed environmental data:** This review centred on the use of sensed environmental data such as CO_2, temperature etc.. Other variables that were considered in order to discuss the history and transition to the current non-intrusive methods were included sparingly. 3) **Estimation/prediction with ML method:** In this review, regression methods were mainly considered especially in the area of room occupancy estimation. 4) **Software application for room monitoring:** This review only considered real time web application that can show real-time values of current variables were considered. **5) Explainable machine learning method:** In this review only the explainable ML for room occupancy models were considered
interventions	Previous methods of room occupancy estimation were intrusive, expensive and unreliable. Therefore, this review excludes those intrusive and expensive methods. Which means that methods that did not use ML were not considered in the final review. Additionally, previous ML methods for room occupancy estimation did not explain how the model results were obtained, which means that ML methods that both the studies that used or did not use interpretable models such as SHAP and LIME were also considered at the end of the review.
Comparator	The reviews considered both comparator and non-comparator studies.
Outcomes	Both qualitative and quantitative studies showing the importance of room occupancy estimation for energy efficiency and environmental room monitoring for healthy, functional indoor environment were reported.
Study Types	Only primary studies were considered important in this review.

in their entirety. A total of n=18 articles were removed because they are ML classification problems rather than regression problems, n=16 articles were removed because they did not give an in-depth analysis of how their results were obtained. All these resulted in the total of n=23 articles being chosen for the final review in the included stage.

Furthermore, the abstract of the articles was reviewed so that only studies that deal mostly with ML methods were chosen and compared with intrusive (non ML) methods; such as use of cameras, PIR etc. Additionally, abstract of articles were reviewed based on indoor CO_2 levels, indoor room occupancy estimation and the ML methods used for the analysis. Thirty extra papers were found from the reference list of the selected articles and from hand-search. Three of the papers from the reference list were excluded because they were not in English, fifteen of the articles were excluded because they studied outdoor CO_2, thirty-five articles were excluded because they did not relate to indoor environmental studies and eighteen were excluded because they studied classification ML problem. Three hundred and sixty papers were removed because the were duplicates. Sixteen articles were removed because they did not give in depth analysis of the study. The flow diagram of the searched, screened and selected articles is shown in Figure 3.1.

3.2.5 Analysis

For the analysis of the selected articles, a similar framework model Smith and Firth [2011] used for the analysis of qualitative study was used to analyse the selected studies. During this process a thematic framework analysis w developed. This framework was able to interpret and identify pattern within the selected articles which has the potential to answer the research questions earlier stated in Section 1.3. This was later improved as new themes and sub themes emerged. There are two main emerged themes, each of the themes have three and four sub themes respectively. These resulted to a total of n=9 themes and sub themes. The emerged themes with their sub themes are 1) Types of ML methods; support vector machine, neural network, extreme learning machine and 2) Types of monitoring systems with analytic backend; real-time systems, web application and mobile systems. The themes showed to be advantageous in addressing the problems initially stated in Section 1.1. The selected articles were reviewed, including the framework used in this analysis. Advice were offered on the final review notes and conclusions.

3.2.6 Results and Citation Management

The management of this references was done with the use of BibTex as used in the Latex files. Both sorting and formatting was done with the BibTex tools for latex files Patashnik [1988]. Additionally, excel table was created to state the attributes of each study. These characteristics are 1) the title, 2) database, 3) abstract summary, 4) year of publication, 5) outcomes stated in the study, 6) the publisher and 7) limitation of the study. This can be seen in Section M.

3.3 Findings

3.3.1 Articles Selected

All the 23 articles that were finally selected for this literature review were published between the year 2004 to 2019. The breakdown of the articles are shown in Figure 3.2

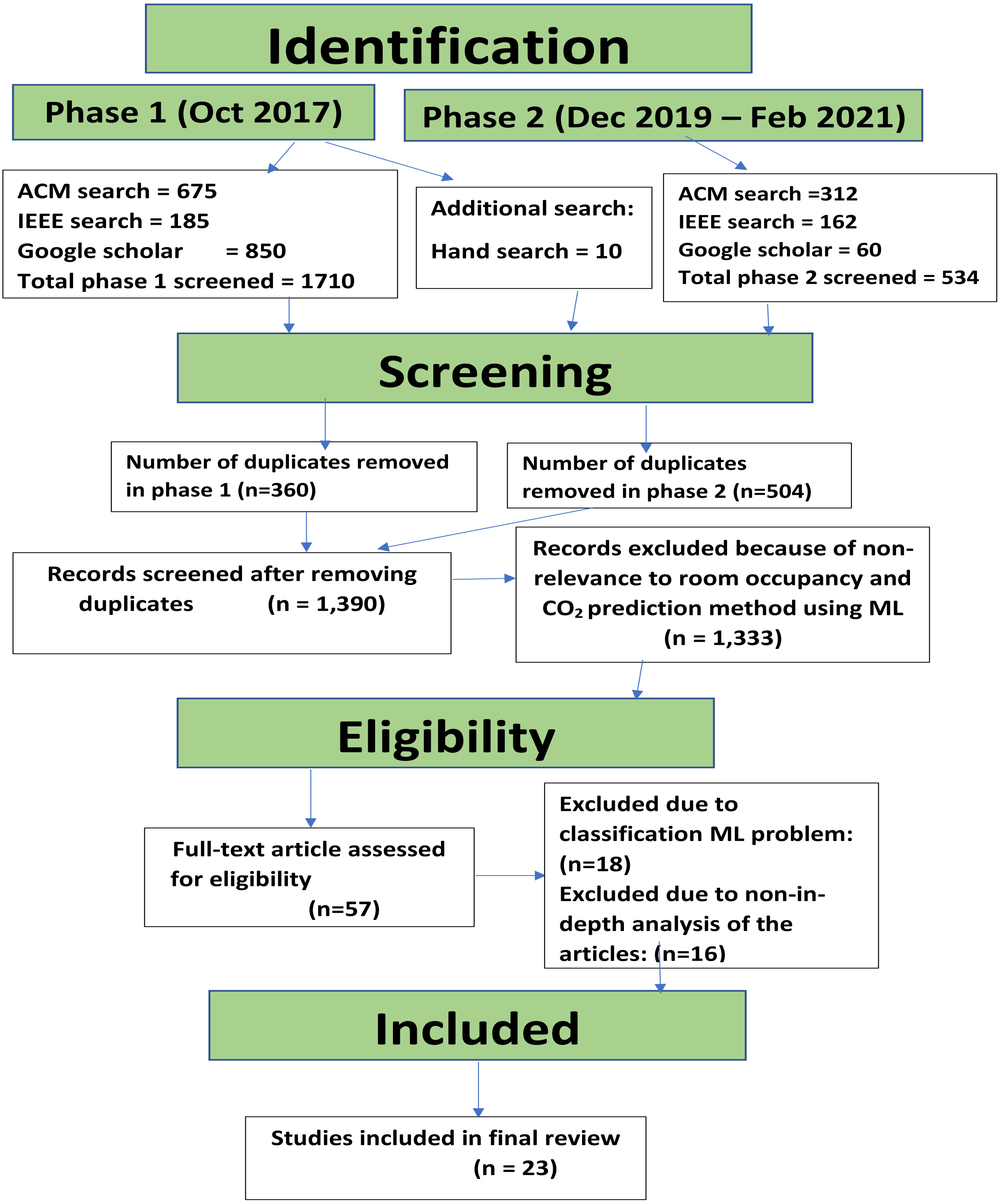

Identification
Phase 1 (Oct 2017)
Phase 2 (Dec 2019 – Feb 2021)
ACM search = 675
IEEE search = 185
Google scholar = 850
Total phase 1 screened = 1710
Additional search:
Hand search = 10
ACM search =312
IEEE search = 162
Google scholar = 60
Total phase 2 screened = 534
Screening
Number of duplicates removed in phase 1 (n=360)
Number of duplicates removed in phase 2 (n=504)
Records screened after removing duplicates (n = 1,390)
Records excluded because of non-relevance to room occupancy and CO₂ prediction method using ML (n = 1,333)
Eligibility
Full-text article assessed for eligibility (n=57)
Excluded due to classification ML problem: (n=18)
Excluded due to non-in-depth analysis of the articles: (n=16)
Included
Studies included in final review (n = 23)

Fig. 3.1. Literature review flow diagram showing the two phases of the search, the databases searched, the exclusion and inclusion criteria of the search conducted

and interpreted as follows:

1. **Room occupancy studies represents:** n=11 articles are for room occupancy studies Alam et al. [2017], Amayri et al. [2016], Arief-Ang et al. [2017, 2018a], Chen et al. [2018b], Fabi et al. [2012], Hong et al. [2016], Jiang et al. [2016], Peng et al. [2018], Tyhurst [2019], Yang et al. [2012] which represents 47.82% of the selected articles.

2. CO_2 **prediction study:** n=3 articles are for indoor CO_2 prediction studies Chen et al. [2018a], Li et al. [2010], Wang and Wang [2012] which represents 13.04% of the total articles selected.

3. **Environmental monitoring systems:** n=9 articles are for the environmental monitoring system application studies Chen et al. [2016], Cheng et al. [2014], Hsu et al. [2019], Jiang et al. [2011], Kim and Paulos [2010], Kumar et al. [2012], Mayer et al. [2004], Peng et al. [2014], Singh et al. [2017] which represents 39.13% of the total articles selected for the review.

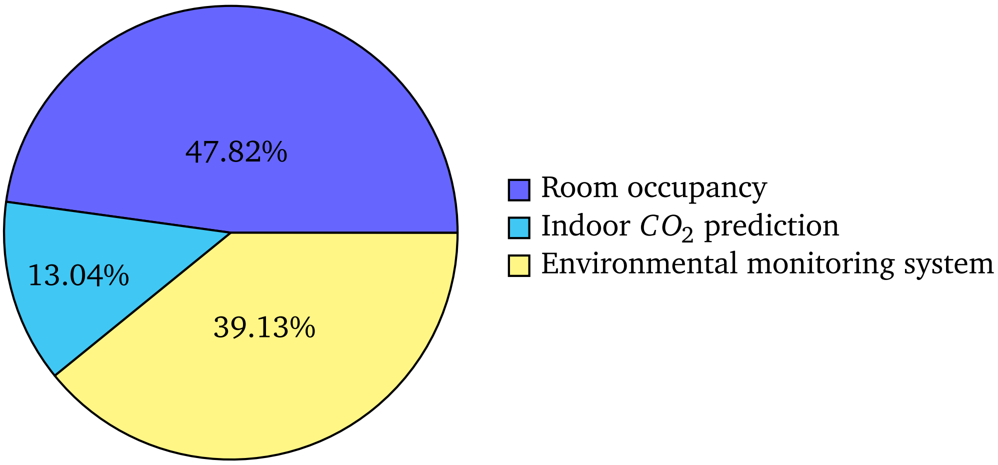


Fig. 3.2. Pie chart representation of the reviewed articles

The increased number of articles selected for each subject above shows the increased awareness of the problems highlighted in this research with regards to healthy indoor environment, energy savings and functional indoor places. Notwithstanding, all the studies were carried out in the developed countries with different or similar climate conditions as it was in the United kingdom (UK). Hence, the generalization of our findings are limited especially as it concerns 1) the quantity of HVAC (either for heating or cooling) been used during the data collection, 2) the type of environmental sensors used for data collection, 3) the position of the sensors and 4) the best frequency for data collection. This clearly justifies our decision to conduct more studies on this topic.

3.3.2 Previous Machine Learning Methods used For Indoor CO_2 Prediction and Room Occupancy Estimation

Eleven articles on room occupancy studies and three articles on CO_2 prediction studies were all done using different machine learning methods. In order to throw more light on the outcome and the results of those fourteen articles, these ML methods applied in each study will be summarized in the following subsections.

3.3.2.1 Support Vector Machine and KNN

An indoor pollutant (CO_2) prediction study was conducted in Chen et al. [2018a] in order to improve indoor air quality and at the same time assess the importance of data pattern on the ML with regards to prediction accuracy. It was observed from the 1765 instances of the dataset used, that 70% of the dataset constitutes the training set and 30% test set. That Support vector machine (SVM) outperformed the Gaussian process (GP) and back propagation neural network (BPNN) used in the analysis with coefficient of determination (R^2) of 98.83. BPNN did not perform its best even though it has the potential to detect data pattern as a network model more than other ML models. One main reason for this low performance is because neural network improves with more dataset Kröse et al. [1993]. Therefore this research will try to use more dataset in its CO_2 prediction study. By using more dataset, it would be possible to access and compare the performance of indoor CO_2 prediction with neural network model.

Another limitation of the Chen et al. is that no other input feature was considered during the analysis as such the reliability of the predictive model could be questioned. Furthermore, this study did not consider ways of showing the indoor air pollutants to the occupants of those rooms. Displaying the results to occupants are necessary in curbing the indoor pollutants.

In the same vein, a room occupancy study by Arief-Ang et al. Netatmo sensor was used for the monitoring and recording of CO_2 data Arief-Ang et al. [2018b]. In this study Arief-Ang et al. proposed a method known as domain adaptive for carbon dioxide-human occupancy counter plus plus (DA-HOC++). The DA-HOC++ method was used in addition to support vector regression (SVR) to predict room occupancy with CO_2 concentration data. The SVR method used in this study achieved a binary prediction of about 91% and counting prediction (estimation) of about 15% in either classroom or laboratory. Arief-Ang et al. was able to show that it was possible to train one day data with support vector regression (SVR) model and able to predict real-time occupancy counter of five rooms. Though this method considered additional input (occupancy count) during its prediction however, no further effort was made to show the results to the people who will benefit from them.

Furthermore, another room occupancy study Peng et al. [2018] that was conducted in order to improve the energy efficiency of the HVAC system. In this study a ML regression model known as k-nearest neighbor (KNN) was used to predict when a room was occupied or not. In other words occupants' behaviour in an office was studied to enable demand driven HVAC supply and also reduce the need for human interference in the control of HVAC systems. Peng et al. showed that it was possible to save about 53% energy used in offices against scheduled cooling. However, some limitations of this study by Peng et al. are 1) there was no explanation of how the results of the model were achieved, and 2) there was no model performance result that was shown in this study. Hence, the effectiveness of the KNN used for this study could not be assessed.

3.3.2.2 Neural Network

A neural network is a spacial interpolation method of multiple dimension Specht [1991]. In room occupancy estimation study Yang et al. [2012], a special kind of neural network model called radial basis function (RBF) was used to estimate the number of occupants in a room. In this study, eight environmental variables; temperature, relative humidity, lighting, CO_2 concentration, sound, motion and infrared were used to estimate the room occupancy.

Yang et al. captured 20-day data at an interval of one minute starting from 00AM to 00PM daily. The accuracy score (R^2) achieved by the RBF model was between 64.83% and 87.62% with tolerance rate (number of occupants) of n=1. In this study, error tolerance rate of 1 was introduced by Yang et al. in order to allow for differences in the actual result and the predicted value by the RBF model. Though this study considered the use of environmental variables as additional input which helped to boost the predictability of the model. However, the limitation of the study lies in the fact that the results obtained by the models were not explained as such only experts who conducted the study have knowledge of the model.

In another study Alam et al. [2017], a neural network model was used to estimate room occupancy with indoor CO_2 concentration. The main aim of the study conducted by Alam et al. was to know if environmental variables such as airflow rate are more important than the variability of the occupants in predicting the exact number of people in the room. It was thus proved that the variation in the occupancy frequency was a better parameter than the airflow rate. This study also proves the fact that room occupancy could be accurately predicted if the room occupancy information is known.

3.3.2.3 Extreme Learning Ensemble

A room occupancy study conducted in Jiang et al. [2016] was done with a ML method called feature scaled-extreme machine learning (FS-EML). During this process the features (variables) of the dataset were scaled as a means of data preprocessing because the units of the dataset were not of the same range values. In this study, Jiang et al. used only the indoor CO_2 concentrations for the estimation of room occupancy. The method used in this study achieved an accuracy rate of 94% with tolerance rate of 4 occupants. However, this study did not use another input variables such as environmental variables for the study. This limitation was also pointed out by the authors where they mentioned that future studies should consider adding environmental variables for the the room occupancy estimation.

On the other hand, other input features such as temperature, humidity, pressure were used in conjunction with indoor CO_2 in another study by Masood et al. [2015]. The same extreme machine learning (EML) model was used in order to estimate room occupancy. However, none of the features were scaled despite differences in the range of values of the input features. Masood et al. was able to show that a pressure variable can improve the proposed ELM model more than other variables with an accuracy score of about 76.05%. This same performance will be tested in this research for comparison.

3.3.3 Previous Study on Environmental Room Monitoring

A real time system was developed by Chen et al. [2016] so as to detect CO_2 concentration levels in airport terminals and presents the data through image and chart to the passengers via user interface. This system was believed to have helped in the improvement of the efficiency of environmental air quality in the airport terminal. There were three structures in this monitoring system and they consist of 1) detector (consists of solar charging unit and CO_2 sensor), 2) wireless equipment (receives and sends the recorded CO_2 data to the user interface) and 3) user interface (displays the data recorded from the detector).

Though this environmental system successfully helped in the monitoring of CO_2 levels that is believed to be sensitive to passengers over all health, however, the system did not make provisions for the forecast of future CO_2 concentration levels or incorporate additional input variable so as to improve the performance of the model. Future CO_2 prediction could help to forestall any future health issues that might arise from the high CO_2 concentration levels. Which could affect passengers health at the airport terminal or it could lead to outbreak of communicable diseases at the airport terminal.

Similarly, an odour detection system was developed in Hsu et al. [2019]. Its main aim was to enable the urban community members to record bad smell as well track the areas where these odours are concentrated. The odour and air quality data reports are designed to be visualized on a map. Hsu et al. system also predicts (based on text analytics) future smell events and sends warning notifications in form of push buttons to the local community. The smell system has helped local communities to advocate for better air quality once the health quality of their environment is of concern. The features of the smell system are of two forms; 1) mobile interface for reporting and visualization of the shared recorded data and 2) push notification for future smell notification.

Though Hsu et al. system predicted future smell events and informs its users of these results, however there are two major limitations to this system 1) the system depends on human reported smell to be able to conduct its analytics as such this system in prone to inherent human error which could affect the performance of the google analytics used for this study 2) the google analytics model used in its prediction was considered a black box model incapable of its pattern been interpreted by its users. Therefore, further work needs to be done in this study.

Air monitoring systems have been developed in Mayer et al. [2004], Peng et al. [2014].Both systems use graphical user interface to display the measured data. The approach of showing the data on a graphical or user interface has helped in the control of indoor air quality Kumar et al. [2012]. The reason is because, visualization of the data on a screen creates conscious awareness to users or stakeholders. Usually, these air monitoring systems are either in form of a web application or a mobile application.

For instance, Jiang et al. developed a personalized mobile system known as MAQS in Jiang et al. [2011]. The MAQS measures IAQ using CO_2 sensors. After the deployment and evaluation of MAQS by its users. It was observed that MAQS accurately reports IAQ of individuals, however there is cost associated with deployment of MAQS because of its platform specific design (only works on AVR-based Arduino). On the other hand, Kim and Paulos [2010] developed a system known as inAir that measures, visualises and shares

IAQ data across different locations.

In conclusion, designing monitoring systems helps to increase the awareness and behavioural change of users and interested stakeholders. However, further studies are still needed in order to bridge the limitations of these existing indoor air monitoring systems.

3.3.4 Discussion of Studies Limitations

The outcomes of the review showed that estimating room occupancy, forecasting future indoor CO_2 and constant monitoring of some indoor environmental variable levels via a graphical user interface has the potential to change occupants behaviour towards making informed decision about their various indoor environment. This in turn helps in making their indoor environment healthy for occupants' overall well-being and potentially reduce the amount of energy consumed by the buildings' HVAC. Therefore, the ML methods reviewed in this chapter was to prove the advantages of economic, non-intrusive and reliable approach of using ML models in conducting the studies that helped to answer the proposed research questions.

Furthermore, some of the current studies showed that CO_2 based demand-controlled ventilation was possible. Another point to note is that Arief-Ang et al used previously recorded dataset in its prediction rather than real-time dataset. This has been found out to have serious faults in some practical applications Chen et al. [2007], because the ML model used usually tends to over fit. Though Arief-Ang et al method showed significant success when tested in different domain, yet the historical data used in the study could have serious concerns because the state of the data might have been outdated because of other environmental condition events that might have occurred during the data recording. Therefore is advised that current or real-time data should be used.

Similarly, Yang et al. [2012] showed that the amount of energy used by HVAC systems in a multi-occupancy space can be reduced. But, the proposed algorithm by Yang et al. needs frequent recalibration of the sensors before it can be applied to a different office for success to be achieved. Nonetheless, Fabi et al. [2012] has shown that some occupants' behaviour such as state of the window (opening and closing) has impact on the energy consumed by the building which consequently affects the predictive environmental variables used for room occupancy estimation. Another concern is that Li et al. [2010] could not explain in details the machine learning method used for predicting the indoor CO_2 concentration levels, which makes it difficult to judge the accuracy of the result obtained.

Jiang et al used a locally smoothed CO_2 concentration data to achieve high accuracy in the room occupancy estimation. However smoothing non-equidistant observational data can make the dataset loose its original information when the chosen smoothing method is not in accordance with the estimated scale of the observational data Terrell [1990]. Other issues such as lack of result testing in a live environment was noticed in some studies Masood et al. [2015], which could reduce the reliability of the ML model used in the study. Furthermore, Hong et al. [2016] discussed progresses and limitations have been made in the occupancy detection research with regards to energy savings. Some of these advances as discussed by Hong et al. are 1) the ability of wireless sensors to detect the presence of occupants in a room. This has lead to more standardized occupancy schedule for effective use of energy in various buildings, 2) due to the ability to detect occupants' presence, thermal comfort of occupants has also improved as a result of better HVAC-occupancy control strategy and 3) the duration of the data collected for the occupancy study. Hong et al observed that the more data that is collected, the better the result.

Conversely, Hong et al observed that different challenges have also hindered the progresses made so far in occupancy research as a result, there are uncertainties in predicting energy usage by occupants in a building. These limitation are 1) difficulties in data collection due to frequent behavioural change of occupants as a result of situational awareness, 2) lack of verification or testing of the different models in more than one domain, which has heightened the scepticism often associated with occupancy research. As such another ML methods needs to be trained in such a way that it can handle insufficient data, historical and real-time data. And the result should be tested in more than one domain for verification.

3.3.5 Conclusion

In conclusion, from the results of the above scoping review conducted in this research, the outcome of the review captured four vital areas of concern which constitutes the present limitations in the available literature. These are 1) the various ML methods used for the studies and how applicable the model could be when used in a different domain, 2) types of sensors used for the data recording (real-time and historical data), 3) the ability of the researchers to interpret or explain their ML models for non-experts' understanding and 4) What time in advance of CO_2 concentration levels could be predicted.

From this review, it was observed that in the previous indoor CO_2 prediction study, only short term predictions were made. Some of the sensors used for the studies were not real-time sensors but rather historical data that might have been outdated, which could affect the performance of the model's results. On the issue of occupancy detection without ML method, 1) smart meters used in measuring the amount energy consumed by room occupants can also be used to estimate the number of occupants in a room but, it is expensive and non-preventive, 2 digital cameras used to detect and estimate room occupancy are intrusive, causes high computational complexity and expensive due to its high resolution properties and 3) the use of occupants' smart phones with Wifi signals of the occupants to detect occupancy by matching the received signal strength (RSS) with the measured anchors in different room locations has shown to be unreliable and complicated in real life because the assumed situation is not always the case.

Furthermore, on the issue of the use of ML methods for room occupancy studies, most of the ML models are room dependent. This means that when the same model is being used for another room, either the model's parameter or the complete model is changed so that the model can achieve optimal result. This makes the proposed model unreliable. Another problem with the generalization of the ML model used for room occupancy study is the weather conditions (especially when the study involves the use of environmental variables) of the originating countries where the studies were conducted and the frequency of the data recorded. Different weather conditions and large dataset could affect the final result of the ML models.

Some of ML models used in the previous studies are SVM, K nearest neighbor (KNN), auto regressive moving average (ARMA), ensemble learning, google analytics and artificial neural network (ANN). None of these ML models used by the authors was considered to be interpreted (explaining how the model obtained its results) because it wasn't considered by the researcher in the first place. This can be assumed to be a limitation to the study because interested users whom those results are meant for might want to make some decisions on their own based on the results.

On the issue of indoor environmental monitoring systems, the issues of concern are 1) expensive and complex systems; making the system difficult to acquire despite its great potentials in the management of rooms environmental conditions and 2) effortless systems without any predictive or analytic backend. Though this might make the system easy to understand intuitively, but it lacks the preventive potentials that could be of more concern to room occupants or managers. Environmental monitoring systems should be able to send warning signals in advance to its users in order to prevent any risk associated with poor indoor environmental conditions.

In conclusion, despite these challenges, greater chances exists when real-time environmental data recorded from sensors are analysed in such a way that occupants or room managers can be guided in making their decisions. Therefore, the limitations observed from the review conducted, the following can be deduced:

- For accurate and consistent monitoring of various rooms:
  1. Fusion of multiple sensors is usually adopted for data collection purposes.
  2. A standalone, web application should be designed for the display of the results to occupants or room managers.
- For room occupancy estimation and indoor CO_2 forecast, the best methodology to be combined and adopted would be:
  1. Data-driven approach where the relationship between the input and the output are modelled using ML methods.
  2. ML model used for the room occupancy estimation should be interpreted for non experts to understand.
  3. The ML models should be tested in different domain in order to check if the model is room dependent or not.
  4. Multiple time-steps CO_2 prediction should be considered so that occupants can have enough time to act.

In order for us to find suitable ML methods that could help to answer the research questions stated in Section 1.3, LSTM and Prophet will be used to predict advanced indoor CO_2 concentration levels because 1) LSTM and Prophet can deal with data that has multivariate variables that are time dependent, 2)LSTM can solve the problem of gradient disappearance and excessive gradient Hochreiter and Schmidhuber [1997] which CO_2 concentration levels often exhibit. Please see Section 5.2.4, 3) Prophet can handle seasonality, trend and holidays which could help to make accurate predictions. Please see Section 5.2.5 for more information on why Prophet was chosen for this research. In the same vein, as a family of ensemble ML, GB will be used to estimate room occupancy because 1) some researchers Jiang et al. [2016], Masood et al. [2015, 2017] have used it in the past and recorded relatively good result, Jiang et al. advised that further research should be carried out for more reliable conclusions. 2) as an ensemble tree method that uses forward stage-wise manner to optimize differential loss function, which is recommended for interpretable ML method because of its suitability as decision trees. Please see Chapter 4 and Section 6.4.3 for more explanation.

Achieving some of these will ensure that the room occupants are on the know of their indoor environmental condition. Hence, this research will try to bridge some of the gaps observed in this scoping review by answering the research questions stated in Section 1.3. The following Chapter 4 will discuss the methods used in this research to answer the research questions. These methods could aid in achieving the aim of improving occupants' comfort, energy savings and functional indoor environment.

4

Methods

The previous chapter explored the scoping literature review of the past studies done in the use of ML methods for room occupancy estimation and indoor CO_2 prediction. Among the challenges identified in the previous chapter are 1) lack of privacy, 2) no interpretable ML approach for room occupancy study and 3) complex, expensive and non predictive indoor environmental monitoring systems. This chapter will explain with the use of a diagram, shown in Figure 4.1 how some of these identified drawbacks will be bridged in order to answer the research questions and what makes the research methodology new in the area of this research.

4.1 Overview of Methods and Approach

The scoping review conducted in this research has identified some strengths and challenges in the literature that deals with the use of ML methods for indoor CO_2 prediction and room occupancy estimation. The review also identified some gaps existing in the literature about various types of software systems used for monitoring indoor environment. Based on the identified strengths and limitations from the literature especially with regards to room occupancy estimation, this research has sought to adopt one of its strength which is, the use of ML methods for room occupancy estimation and future CO_2 prediction. The two main reasons for choosing ML methods over the use of digital camera, WiFi and passive infrared is because it is inexpensive and non-intrusive.

Therefore, this research will use ensemble ML method known as Gradient boosting (GB) ensemble to estimate the number of people in the room and also use the time series ML methods to predict future CO_2 levels. The GB ensemble ML model was chosen because of its suitability for the application of prediction interval approach for regression problems and for better model interpretability. Other ensemble models such as decision trees and random forest (RF) would have been used, but they both have overlap in terms of how it learns from its problems. In order words, RF uses majority rule that happens at the end of every computation (process) to make its prediction whereas, GB starts its combination at the beginning of the process.

Other strengths for adopting all the approaches used in this research are mentioned in Section 1.4. Hence this research has sought to bridge some of the identified gaps in the literature by answering the three main research questions shown in Section 4.1. In this table, there are four columns. The first column called “RQ-Numbers” represents each of the research questions number, the second column called “Study Chapters” represents the individual study chapters where each of the research questions were answered, the third column called “Study Name” represents the name of each study chapters and the fourth column called “Research Questions” represents each of the research questions adopted and answered in this research. Therefore, the three of the research questions shall be referred to as RQ1 (Chapter 5), RQ2 (Chapter 6) and RQ3 (Chapter 7) and they are shown in Section 4.1.

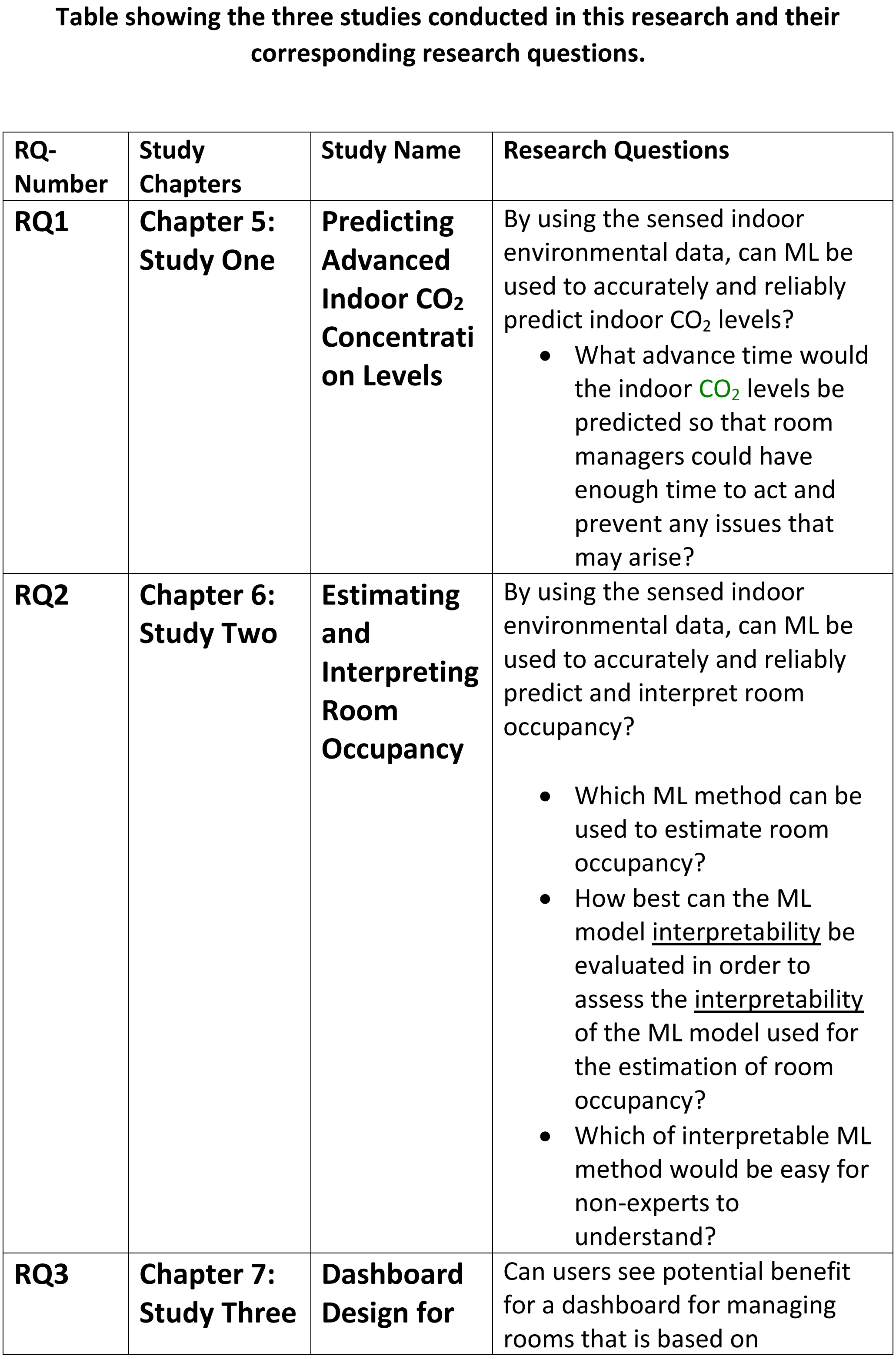


Table showing the three studies conducted in this research and their corresponding research questions.

RQ-Number	Study Chapters	Study Name	Research Questions
RQ1	**Chapter 5: Study One**	**Predicting Advanced Indoor CO_2 Concentrati on Levels**	By using the sensed indoor environmental data, can ML be used to accurately and reliably predict indoor CO_2 levels? • What advance time would the indoor CO_2 levels be predicted so that room managers could have enough time to act and prevent any issues that may arise?
RQ2	**Chapter 6: Study Two**	**Estimating and Interpreting Room Occupancy**	By using the sensed indoor environmental data, can ML be used to accurately and reliably predict and interpret room occupancy? • Which ML method can be used to estimate room occupancy? • How best can the ML model interpretability be evaluated in order to assess the interpretability of the ML model used for the estimation of room occupancy? • Which of interpretable ML method would be easy for non-experts to understand?
RQ3	**Chapter 7: Study Three**	**Dashboard Design for**	Can users see potential benefit for a dashboard for managing rooms that is based on

Table 4.1: Research Questions Study Mapping

		Room Monitoring	interpretable ML methods from environmentally sensed data? • How will the dashboard be designed so that it can help in visualizing the levels of indoor environmental variables, the predicted indoor CO_2 levels and the estimated room occupancy? • How will the stakeholders' contributions be identified and incorporated in the design phase of the system? • What are the functional requirements that will help the system users to perform their task effectively? • What software tool will be used to test the dashboard system for easy evaluation? • How will the designed system be evaluated in order to determine if the system can perform its required tasks when developed?

Table 4.2: Research Questions Study Mapping

The research methodology that was chosen to answer these research questions were as follows:

- To conduct first quantitative study that seeks to explore how advance indoor CO_2 concentration levels can be predicted using time series ML models.
- To conduct a second quantitative study that seeks to explore how room occupancy can be estimated using boosting ensemble ML method through prediction interval approach.
- To interpret the ensemble ML model used for room occupancy estimation with a suitable ML interpretable method.
- To conduct a third study that seeks to explore ways to design a prototype dashboard system that helps its users to visualise the results of the quantitative studies and other environmental variables.
- To conduct qualitative studies that will help in the analysis of the quantitative study and the design of the dashboard system.
- To test each of the proposed models used for the quantitative studies with different dataset recorded from another room.
- To evaluate the interpretability of the GB ML model with non-experts.
- To test and evaluate the designed dashboard with people who may use it.

The reasons for choosing these main research methods and approach are as follows:

- Future forecast of indoor CO_2 concentration levels in real-time will give room occupants enough time to act against any perceived health risk.
- Using time series ML methods for only CO_2 prediction study was because the model can deal with dataset with multivariate values and the ones prone to spikes such as CO_2 values.
- Using prediction interval approach for the estimation of room occupancy and testing the models with another dataset will ensure that the models can perform optimally in a room with the same environmental setting where the studies were conducted.
- Interpreting and evaluating the ensemble ML model will help non-experts to understand the rationale behind the model's method of prediction and also help in assessment of its level of explainability.
- Interpreting ML model for only room occupancy estimation was because the pattern was better to understand and predict by non expert if modelled.
- Designing a dashboard system for the real-time visualisation of the predicted results and environmental variables will ensure easy and faster development with the hope of contributing towards the improvement of the existing methods of ensuring healthy and safe indoor environment for room occupants.

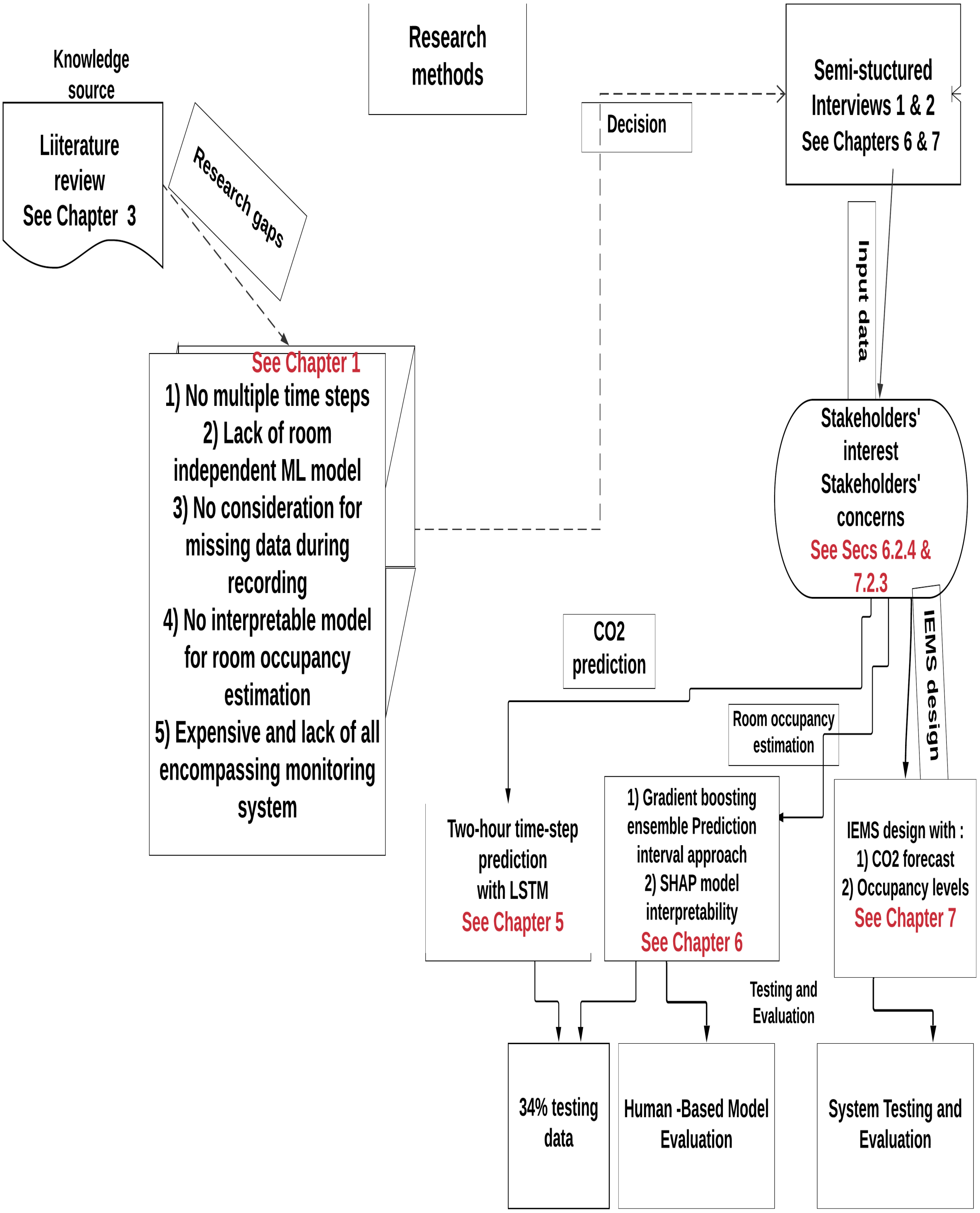

Knowledge source
Liiterature review
See Chapter 3
Research methods
Research gaps
Decision
Semi-stuctured Interviews 1 & 2
See Chapters 6 & 7
Input data
See Chapter 1
1) No multiple time steps
2) Lack of room independent ML model
3) No consideration for missing data during recording
4) No interpretable model for room occupancy estimation
5) Expensive and lack of all encompassing monitoring system
Stakeholders' interest
Stakeholders' concerns
See Secs 6.2.4 & 7.2.3
IEMS design
CO2 prediction
Room occupancy estimation
Two-hour time-step prediction with LSTM
See Chapter 5
1) Gradient boosting ensemble Prediction interval approach
2) SHAP model interpretability
See Chapter 6
IEMS design with :
1) CO2 forecast
2) Occupancy levels
See Chapter 7
Testing and Evaluation
34% testing data
Human -Based Model Evaluation
System Testing and Evaluation

Fig. 4.1. Research Methods Diagram

- Conducting the qualitative studies will ensure more user centred research study.
- Testing and evaluating the designed dashboard with the potential users will help to ascertain if the purpose of the dashboard was actually met.

4.2 Research Methodology Diagram and Explanations

The diagram shown in Figure 4.1 represents how this research study started and the road-maps that were followed in order to bridge the identified gaps in the literature. Firstly, the symbol named **Knowledge source** shows how this research was started by conducting a literature review in order to understand the topic and the reason for the research. This was done by searching three different databases such as ACM, IEEE and Google Scholar in order to identify 1) what has already been done in the literature, 2) the limitations in the area of research and 3) to formulate research questions based on the identified study limitations.

The research questions shown in **research gaps** symbol was obtained after the literature review search and can be seen in Section 4.1. In order to add to the research questions, this research proceeded to conduct two qualitative studies as can be seen in the **semi-structured interviews 1 & 2** symbol. These two qualitative studies were in form of semi-structured interviews shown and they are shown in Section 6.2 for room occupancy study and Section 7.3 for indoor environmental monitoring system. The number of participants for the first qualitative study were five (n=5) and the number of participants for the second qualitative study were two (n=2). These qualitative studies were done in order to help this research to focus more in identifying the need or interest of the potential stakeholders. The answers obtained from the semi-structured interviews further solidified the initially stated research questions. The obtained interview findings can be seen in Section 6.2.4 and Section 7.4.1.

Some of the additional information obtained from the interview data that were added to the research questions are 1) the display format for the CO_2 prediction and room occupancy estimation results. 2) the number of advanced time steps that can give the room occupants more time to react in their various indoor environment and 3) the functional and non-functional requirements of the dashboard system to be designed. All these information were discussed in chapters shown in each flowchart symbol for further information.

Furthermore, after formulating the research questions based on the information available from the literature and the additional information obtained from the interview data as **input data**, possible solutions that will help in bridging the gap in the literature were categorized into three studies namely; 1) indoor CO_2 prediction study, 2) room occupancy estimation study and 3) room monitoring system design study. Section 4.1 shows the representation of each study and how they correspond to each of the research questions. Moreso, the reason for choosing the three studies were 1) to ensure healthy indoor environment, our rooms should be regularly monitored and tracked in order to void any health challenge that could affect room occupants, 2) this room monitoring can only be done scientifically with the use of approved device such sensors, 3) the results from the recorded data should be analysed for any hidden insight so that occupants can take decisions and change the way they live indoors and 4) showing the room occupants these results is a great way to constantly remind them of how healthy their indoor environment are and the need to take action if need be.

Finally, in the **"Testing and Evaluation"** section, the interpretability of the GB ML model used for the estimation of room in this research were evaluated using human-based approach and its results are discussed in Section 6.7. Likewise the testing and evaluation of the dashboard designed in this research. Four participants took part in the model evaluation, while six participants took place in the dashboard evaluation. The results of the dashboard evaluation are discussed in Section 7.5. The reason for these choices are in order to ensure that our indoor environment is not only healthy all the time but also, energy efficient and fit for any kind of purpose for everyone. Additionally, various ML models used for the quantitative data analysis were also shown in each **study** symbol in Figure 4.1. These ML models were chosen based on the proposed problem that was intended to solve by this research and interpretability of the chosen ML models especially for the room occupancy study.

4.3 Dashboard Design Methods

Before deciding on what features and functions to be included in the three studies conducted in this research, interviews were conducted and analysed, the resultant user requirements were prioritized. The prioritization of the user requirement was done in such a way that more efforts were put in the most important system requirements which were obtained from the summary of the interview analysis. For example, the technique used to select the most important user requirements was ranking technique. Which means that requirement "1" was considered the most important because of the number of participants that suggested it during the interview and requirement "10" was the least because only one participant thought that it was necessary.

4.4 Conclusion

This chapter has shown the complete strategy and research methods used in this research to answer the research questions. This chapter was important because it helped to highlight to the reader what has been done in the past, what this research has identified to do in order to bridge some of the gap in the literature, how this research intends to do that and the reason for choosing its approach. This chapter has also highlighted why the research approach is novel in this area of study. Most importantly, this chapter serves as a means of simplifying the complexity involved during the course of this research by a means of a diagram so that the importance and benefits of the research can be observed by the reader in an unambiguous manner.

This chapter went further to identify the interview content analysis method used to analyse all the interviews conducted in this research. This chapter also stated various ML models used for regression problem that were considered necessary for conducting the quantitative part of this research. Such models are time series models (Vector Auto Regression, Facebook Prophet and Long Short-Term Memory) and ensemble tree model (Boosting). The algorithm for ensemble method were also discussed for more understanding. Two methods (root mean square error and mean absolute error) of calculating the performance of the ML models as it was used in the evaluation of the results obtained in this research were discussed.

Data preprocessing method such as feature scaling and encoding of categorical variables as a means of feature engineering were discussed. These data preprocessing techniques were prerequisite for some of the ML models used. Finally, in order to answer part of the research questions stated earlier in this research, two ML methods (local interpretable model-agnostic explanations (LIME) and SHapley Additive exPlanation of model interpretability (SHAP)) were discussed. Their individual approaches as it relates and differs from each other were discussed in this chapter. The following chapters will now discuss all the three studies involved in this research in a more elaborate and descriptive style.

5

Study One: Advanced Indoor CO$_2$ Prediction Study

In Chapter 4, different steps to be taken in order to bridge the identified gaps in the literature were stated. The research questions to be answered with the steps discussed are stated in Section 1.3. The previous chapter also discussed the number of studies to be conducted in this research and how it relates to answering each of the research questions, please see Section 4.1 for reference. This chapter will now discuss the first study conducted in this research. The different ML methods used and reasons for each choice including comparing the results of all the time series ML models chosen for this first study.

5.1 Introduction

In this first quantitative study, an indoor CO_2 prediction study was conducted. The overall study design and different phases that were followed in order to answer part of the research question (RQ1) in Section 4.1 is shown in Figure 5.1. In this diagram, two stages of CO_2 prediction study were conducted namely preliminary and confirmatory study. The reasons for conducting preliminary and confirmatory studies are as follows:

1. The preliminary study will serve as an introductory basis where two time series model will be tested in order to ascertain the best model among the two. In this case, the two different time series model to be used for this first study are LSTM Section 5.2.4 and Facebook Prophet Section 5.2.5.

2. The confirmatory study will be used to confirm the performance the best model in (1) above. Thereby, helping in determining and confirming the best performed model for indoor CO_2 prediction.

Indoor CO_2 concentration levels are highly unpredictable, this means that there are no obvious pattern that is known to be consistent in perfectly modelling the indoor CO_2 levels of our rooms. As a result, a good ML time series model suitable for multivariate time series problem is needed for observing the history of sequence of data in order to correctly predict what the future level is going to be. Therefore, the reasons for choosing these multivariate time series models; LSTM and Prophet models are as follows:

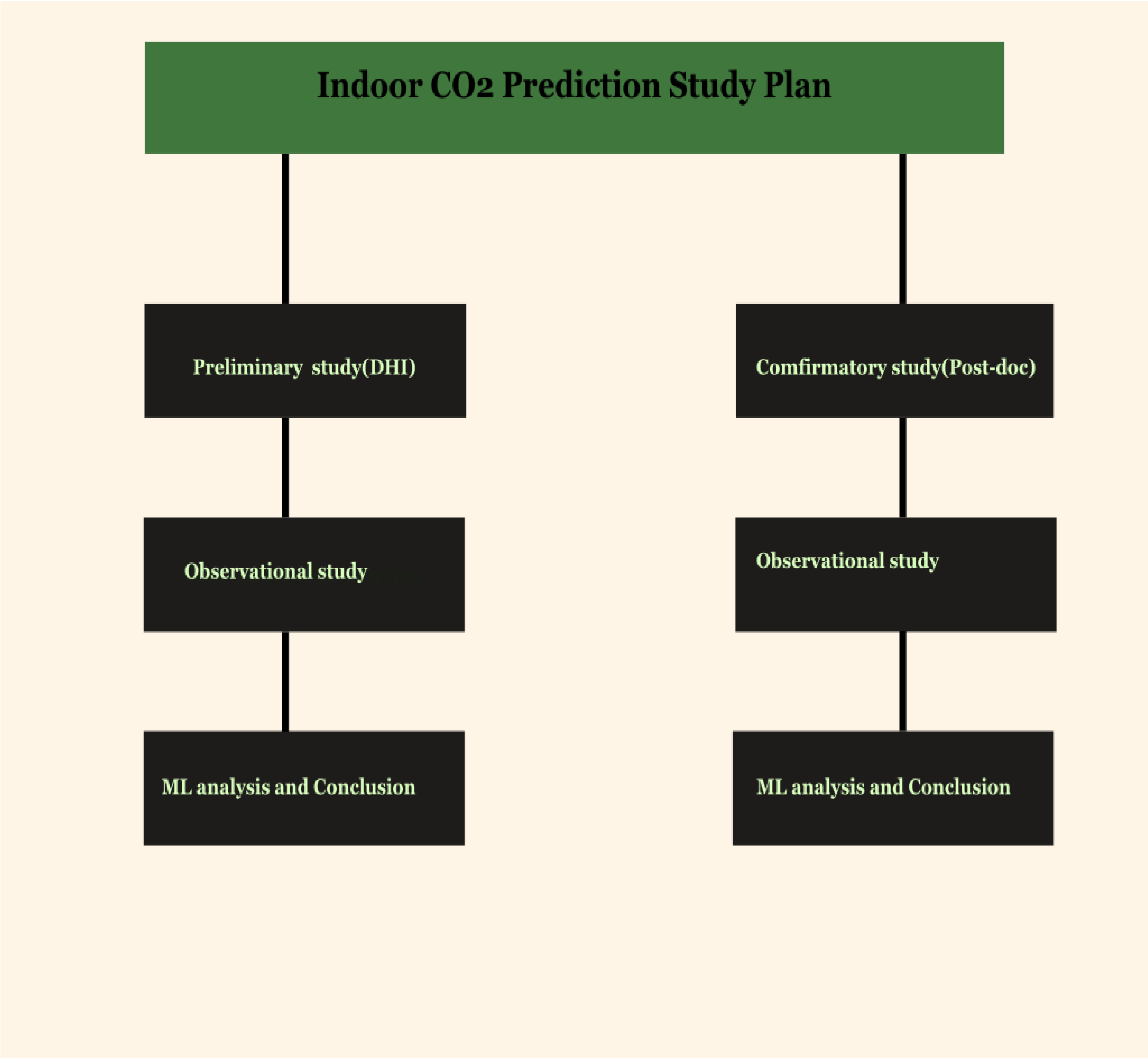

Indoor CO2 Prediction Study Plan
Preliminary study(DHI)
Comfirmatory study(Post-doc)
Observational study
Observational study
ML analysis and Conclusion
ML analysis and Conclusion

Fig. 5.1. The study design diagram for different phases of advanced indoor CO_2 prediction study at two locations namely Swarm-CO2 and Post-Doc

- The dataset recorded from sensor data for the indoor CO_2 prediction study is a multivariate (multiple variable) time series problem. So because of that, only time series models that can be used for multivariate problems were chosen.

- The Prophet model uses an approach known as “forecasting at scale” (where both human and automated task are used) to make its prediction Taylor and Letham [2017]. This approach makes it easy for non-expert to understand its methods. Additionally, Prophet is robust to noise and missing data.

- The LSTM model is suitable for understanding the long sequence of structured data overtime Hochreiter and Schmidhuber [1997].

5.1.1 Aims

The main aim of this indoor CO_2 prediction study is to explore and select the best ML time series model that best predicts 1-hour and 2-hour times of indoor CO_2 concentration levels of a room in advance. The objective of this study is to train and test the first datasets (Swarm-CO2) with the two time series ML models listed above, after which the best performed model for multivariate prediction will again be applied on the confirmatory Post-Doc dataset in order to confirm its performance.

5.1.2 Hypothesis

The hypothesis for this quantitative study one is as follows:

- If the indoor environmental variable levels such as temperature, pressure, humidity and noise of any room are known, then at least 1 hour forecast of the future indoor CO_2 concentration levels could accurately be predicted.

- If the time history of the CO_2 levels are known, then atleast 1 hour forecast of the future indoor CO_2 could be accurately predicted.

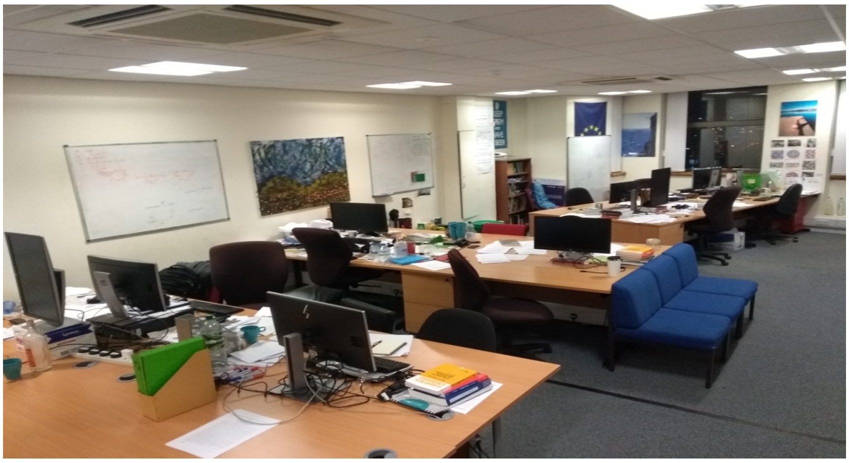


Fig. 5.2. Picture of the Post Doctoral room where the confirmatory CO_2 prediction study was conducted

5.2 Machine Learning Time Series Models Used

Time series model is regarded as a hypothesis about the probability expectation over time Tong [2012]. Before a time series is modelled or analysed, a general approach is followed in order to introduce dependency and stationarity if the series does not have one Brockwell et al. [2002]. The first approach is to plot the series and examine the features to see if there are 1) trend, 2) seasonal component, 3) any obvious sharp changes and 4) outliers in the observations or data. Once any of these listed features is found in the series, it is removed using a standard procedure. Brockwell et al. represents the time series data with outliers below using a decomposition model:

$$X_t = m_t + s_t + Y_t \tag{5.1}$$

In Equation (5.1), m_t is a function representing a trend component, s_t is a seasonal component and Y_t is a stationary noise component. Other approaches such as removal of seasonal trend or component and applying of best fit models can be used also in time series analysis.

5.2.1 Test for Stationarity

Generally speaking, a time series $\{X_t, t = 0, \pm 1, ...\}$ is referred to as stationary if its statistical properties are the same as the time-shifted series $\{X_{t+h}, t = 0, \pm 1, ...\}$ for each integer h Brockwell et al. [2002]. Mathematically a stationary time series $\{X_t\}$ with a mean function and covariance function are represented in Equation (5.2) and Equation (5.3) respectively. Where $E(X_t^2) < \infty$ for all integers r and s.

$$\mu X(t) = E(X_t) \tag{5.2}$$

$$\gamma X(r,s) = Cov(X_r,X_s) = E[(X_r - \mu X(r))(X_s - \mu X(s))] \tag{5.3}$$

Similarly, the autocovariance function (ACVF) of a stationary time series $\{X_t\}$ at lag h is represented in the Equation (5.4). While the autocorrelation (ACF) of a stationary time series at lag h is represented in Equation (5.5).

$$\gamma X(h) = Cov(X_t + h, X_t) \tag{5.4}$$

$$\rho X(h) \equiv \frac{\gamma X(h)}{\gamma X(0)} = Cov(X_t + h, X_t) \tag{5.5}$$

Instead of graphical inspection as a way of checking for stationarity as mentioned above, there are other ways of checking whether a time series is stationary or non-stationary. One method of doing that which was used in this research is the use of augmented Dickey-Fuller (Augmented Dickey-Fuller (ADF)) and Dickey-fuller (Dickey-Fuller (DF)) test Dickey and

Fuller [1979], Said and Dickey [1984]. The ADF method is commonly used to test if a time series has a unit root and the method is dependent on the lag order and the critical values in finite samples Cheung and Lai [1995]. Dickey and Fuller [1979] showed that DF estimator for a autoregressive (AR) series is represented as Equation (5.6).

$$Y_t = \rho Y_{t-1} + \exp_t, \; t = 1, 2, ..., \infty \tag{5.6}$$

in Equation (5.6), $Y_0 = 0$,ρ is real number and {exp} is a sequence of independent normal random variables with mean zero and variance σ^2. The Y_t converges to stationary if $|\rho| < 1$. When $|\rho| = 1$, then the time series is not stationary and the variance of Y_t is $t\sigma^2$. Given n finite observations $(Y_1, Y_2, ..., Y_n)$, ρ can be estimated using the least squares estimator as in Equation (5.7).

$$\hat{\rho} = (\sum_{t=1}^{n} {Y_{t-1}}^2)^{-1} \sum_{t=1}^{n} Y_t Y_{t-1} \tag{5.7}$$

Asymptotically, ADF test statistics does not depend on the lag order. However empirically, when the observations are finite the ADF distribution is sensitive to lag order and finite-sample critical values MacKinnon [1991]. Through substantial response surface regression estimation of quantiles MacKinnon [1994] showed that ADF distribution function can be calculated and the results obtained was based on the three versions of ADF test. Cheung and Lai [1995] study investigated each of the roles of the sample sizes and the lag order k in specifying the critical values of ADF test when the sample size is known and observed that ADF test is dependent on lag-adjusted critical values in a finite set of observations. However, the problem of applying the right lag-adjusted critical values is still unresolved.

Comparatively, DF Dickey and Fuller [1979] uses only the version when $k = 1$. However, Said and Dickey [1984] uses three 1%, 5% and 10% versions of ADF unit root test to obtain critical values of time series problem. Hence in this research three versions 1%, 5% and 10% of ADF was obtained in order to test for stationarity of the time series observations.

5.2.2 Feature Engineering Methods Used

5.2.2.1 Data Preprocessing

Data collected from more than one resource are often affected by different issues such as range inconsistency and redundancy during the integration process Al Shalabi et al. [2006]. These resources are databases, data cubes and flat files. One example of inconsistency in data is different ranges or scales of input and output variables in ML, which is an important factor in the learning process of a ML model. When an input variable is not scaled, it could make a learning process slow and unstable, whereas an unscaled target variables could collapse the gradients and lead to failure of the learning process Motulsky and Christopoulos [2004].

Causes of different scales of input variables in regression analysis are 1) lack of uniform units (kilometres, hours etc..) in the input variables, 2) when there are outliers in the dataset and 3) quasi periodic signal series that may be contaminated due to background flow level noise Wu et al. [2009]. Hence applying robust scalers or transformers are often beneficial for most regression analysis. Data preprocessing techniques used in ML analysis are normalization, transformation, data cleansing, feature extraction and selection. These techniques basically means to rescale the input and the output variables before training the model with the dataset. In most cases, data normalization does not alter the best-fit parameters rather it provides an opportunity for the data analyst to observe what has happened in the experiment so as to be able to compare it with other analysis Motulsky and Christopoulos [2004].

Therefore, in order to prepare our regression data before the regression model was applied, a data preprocessing technique known as normalization was applied in order to improve the accuracies and performances of some of our regression models such as LSTM and GB. The two most important scope of data normalization Kotsiantis et al. [2006] are represented mathematically below:

$$v\prime = \frac{v - min}{max - min}(new\_max - new\_min) + new\_min \tag{5.8}$$

$$v\prime = \frac{v - mean}{stand\_dev} \tag{5.9}$$

In equation Equation (5.8) and eq. (5.9) v is the observed feature and $v\prime$ is the normalized feature.

5.2.2.2 One Hot Encoding

The existence of categorical variables in a table is often considered to be distinct entities. These distinct entities are usually converted to feature vectors in a process known as dummy coding in order to avoid data redundancy during statistical computing Cerda et al. [2018]. A simple and common encoding method often used in regression problems is called One Hot Encoding (OHE); the conversion of categories into dummy variables Alkharusi [2012], Berry et al. [1998], Cohen et al. [2013] .

For example a categorical variable called “colour” with green, black and white as its categories can be encoded with 3-dimensional feature vectors as {[1,0,0],[0,1,0],[0,0,1]}. In other words, each existing membership in the category is coded as one while the non-membership in the category is coded as zero. This makes each category in the resultant vector equidistant to each other. According to Cerda et al., sometimes OHE could lead to high-cardinality because often for a table with more of its variables having higher number of categories the resultant feature matrix leads to high-dimensionality. This in turn increases computational cost. This problem can only be tackled with dimensionality reduction after OHE of the variables.

Other methods of encoding are suggested by such as hash encoding Weinberger et al. [2009] that reduces data dimensions through hashing method. Another encoding method is encoding with target statistics Duch et al. [2000], in this case category are encoded based on the impact it has on the target variable. In regression problems, categorical variables are not numerical but rather symbolic, hence there is need to define a feature matrix from its relation before the application of any statistical method.

There are different OHE methods in linear regression problems however, Cohen et al. [2013] observed that they often have the same R^2 score. Mathematically, OHE can be seen as a method that sets each feature vector as what is shown in Equation (5.10). when A is a categorical variable with categories $K \geq 2$, the domain $A = \{d_l, 1 < l \leqslant K\}$ and $t^i(A) = d^i$. In Equation (5.10), the $1_{d1}(.)$ is the indicator function over a singleton 1_{d1}.

$$X^i = [1_{d1}(d^i), 1_{d2}(d^i), ..., 1_{dk}(d^i)] \in R^k \tag{5.10}$$

5.2.3 Performance Indices Used for Time series Model Evaluation

Two performance indices used in this research are RMSE and Mean absolute Error (MAE). Both of them have been used over the years as a standard statistical metrics for the assessment of the performance of air quality, climate and environmental research studies Chai and Draxler [2014]. However, Willmott and Matsuura [2005] raised some concerns on their model performance abilities. Willmott and Matsuura claims that RMSE is often misinterpreted as a measure of absolute error hence misleading and inappropriate. On the contrary, Chai and Draxler cautioned against the avoidance of the use of RMSE in spite of their valid points. In fact, it was observed that the RMSE was more appropriate when the expected error distribution is Gussian.

Furthermore, RMSE penalizes variance by giving errors with larger absolute values more weight and the errors with smaller absolute values less weight Chai and Draxler [2014]. Interpreting RMSE and MAE mathematically, assuming there exits n samples of model errors ϵ and is being calculated as $(e_i, 1, 2, ..., n)$ without the consideration of the uncertainties in the methods used and with the assumption that ϵ is unbiased. The RMSE Equation (5.11) and the MAE Equation (5.12) are calculated as follows respectively:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2} \tag{5.11}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |e_i| \tag{5.12}$$

Therefore, RMSE is assumed to present unbiased errors with normal distribution. This characteristic makes RMSE easy in terms of providing the exact picture of error distribution. Another distinct characteristics of RMSE is that it does not make use of absolute values unlike MAE as shown in Equation (5.11). When the values of RMSE is low the better the result Aptula et al. [2005]. Therefore, the performance of this model was evaluated based of accuracy and loss for the LSTM. Accuracy in this context is the measure of fit of the model between the observed (test) data and the predicted data. RMSE is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit.

5.2.4 Long Short-Term Memory

Over the years time based modelling and predictions have been challenging to researchers and speculators because most of the data have outliers and are non-stationary Long et al. [2019]. However, with the development of deep learning neural network, feature learning is now been done with better accuracy. One of the ML neural network model often used for the time series prediction problem in known as recurrent neural networks (RNNs). Nevertheless, the RNNs have a major problem of disappearance of gradient Fang et al. [2019]. As a result of that, modern data scientist now use Long Short-Term Memory (LSTM) Hochreiter and Schmidhuber [1997], which is a branch of Recurrent Neural Network model (RNN), to solve the problem of gradient disappearance and excessive gradient. LSTM is local in space and time with computational complexity of 0.1, making it to learn faster than most neural network (NN) models.

5.2.4.1 The Structure of LSTM:

LSTM has a complex structure named LSTM cell in its hidden layer Duan et al. [2016]. It comprises of three distinct gates called input, forget and output gates. These gates authorize the movement of information through cell and the NN. Duan et al. observed that while constructing the structure of LSTM at any given time t, there is a corresponding input and hidden layer output x_t and h_t respectively. The previous output layer is h_{t-1}, the cell input state is $\tilde{C}_t$, the cell output state is C_t and the former cell state is C_{t-1}. The three LSTM gate states comprises of i_t, f_t and o_t. The following equation shows how to calculate the output and the input states and gates as stated in Duan et al. [2016].

$$i_t = \sigma(W_1^i \,.\, x_t + W_h^i \,.\, h_{t-1} + b_i) \tag{5.13}$$

$$f_t = \sigma(W_1^f \,.\, x_t + W_h^f \,.\, h_{t-1} + b_f) \tag{5.14}$$

$$o_t = \sigma(W_1^o \,.\, x_t + W_h^o \,.\, h_{t-1} + b_o) \tag{5.15}$$

$$\tilde{C}_t = tanh(W_1^C \,.\, x_t + W_h^C \,.\, h_{t-1} + b_C) \tag{5.16}$$

The Equation (5.13) is the input gate, Equation (5.14) is the forget gate, Equation (5.15) is the output gate and Equation (5.16) is the input cell. The W_1^i, W_1^f, W_1^o and W_1^C are the matrices of weight that links x_t to the three gates and the cell input; $W_h^i, W_h^f, W_h^o, W_h^C$. The bias terms are b_i, b_f, b_o and b_C. The sigmoid and the hyperbolic functions are σ and $tanh$. The cell output can be calculated as the following in Equation (5.17):

$$C_t = i_t * \tilde{C_t} + f_t * C_{t-1} \tag{5.17}$$

The hidden layer and the network output can be calculated as follows in Equation (5.18) and Equation (5.19) respectively:

$$h_t = o_t * tanh(C_t) \tag{5.18}$$

$$x_{\tilde{t+1}} = W_2 \, . \, h_t + b \tag{5.19}$$

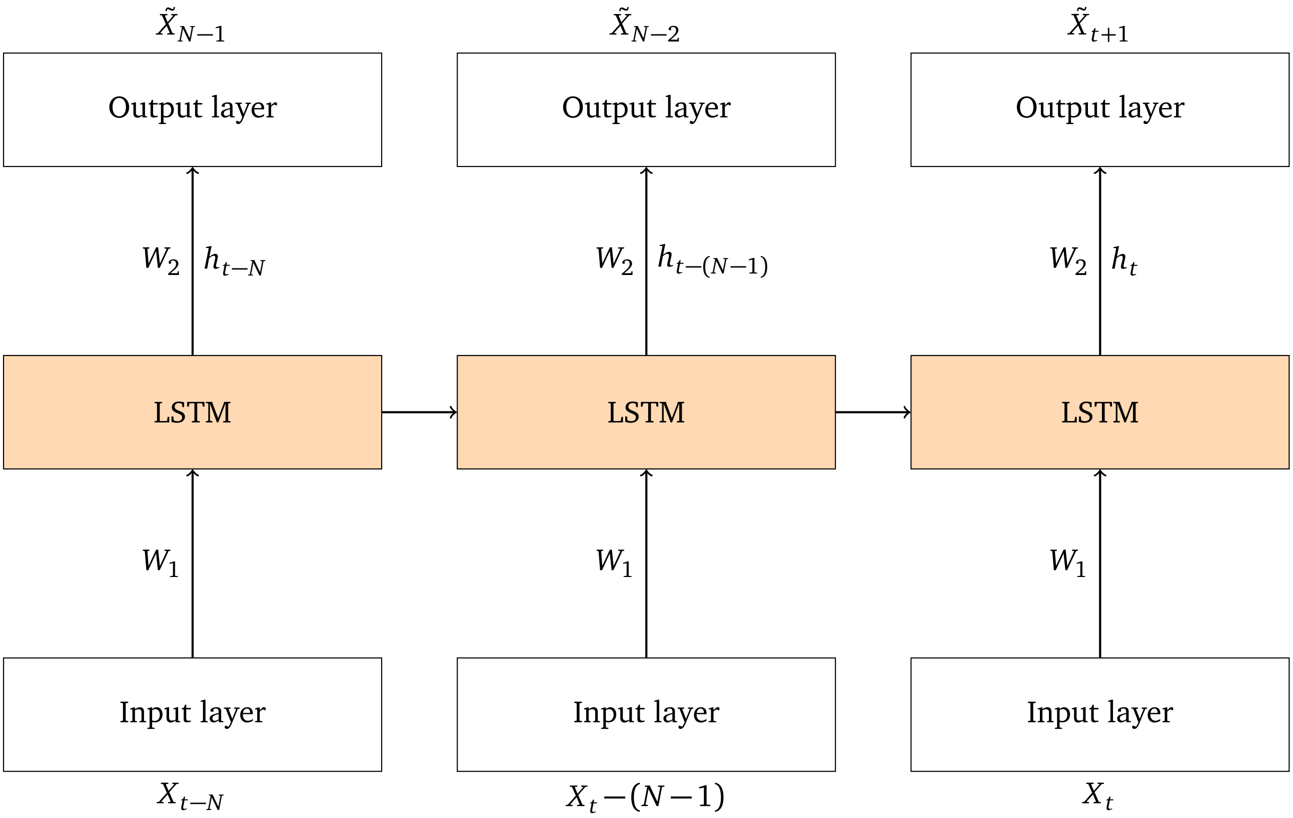


Fig. 5.3. LSTM structure for series prediction

Figure 5.3 represents RNN and LSTM network cells at time t. The observed or the history data is represented as x_t, while the output or the predicted future data is represented as $x_{\tilde{t+1}}$. W_2 is the weight between the output and the hidden layer. N is the number of the historical observation that will be used to feed the series in order to make LSTM predictions.

5.2.5 Facebook Prophet

Facebook Prophet model was developed with the intention of producing a more reliable forecast by using a theme known as “forecasting at scale” Taylor and Letham [2017]. This approach uses interchangeable regression model with explainable and adjustable parameters by time series experts. By forecasts at scale Prophet means the business time series forecasting; 1) where large number of non-experts in the time series methods are allowed to make forecast, 2) there are forecasting problems with distinctive features and 3) where large number of forecast are generated and evaluated even when they might be seen to be performing poorly. The forecasts at scale means that human feedbacks are used to fix the performance problems identified by the Prophet model. In other words, Prophet uses both human and automated task to make its predictions.

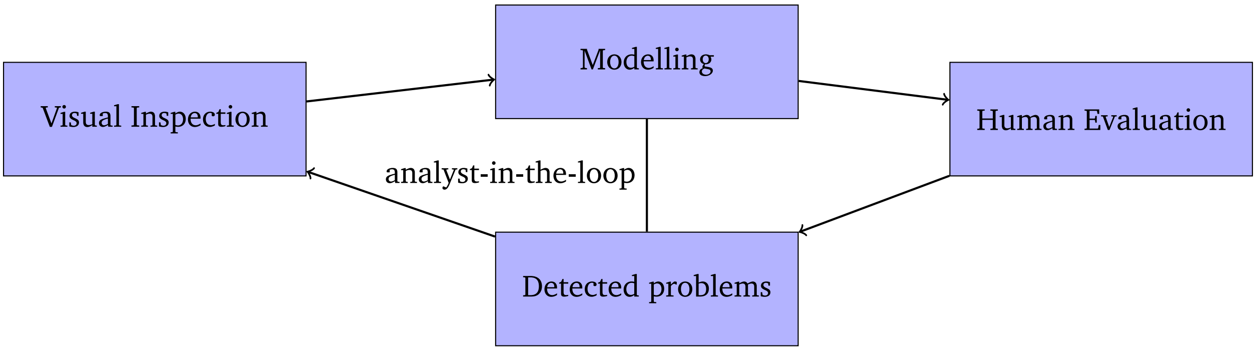


Fig. 5.4. Facebook Prophet analysts in the loop diagram

Figure 5.4 explains the Facebook Prophet “analyst in the loop” approach where a data is first modelled using the Prophet method, the result of the forecast is then evaluated in order to find out the model performance and identify some errors in the forecast. After the problems have been identified by experts, the necessary model adjustment will be made and the loop continues again.

5.2.5.1 Prophet Model Parameters:

There are parameters been offered by Prophet that are adjustable without any expert knowledge of the underlying model. These Prophet parameters were formulated based on the traditional time series structure proposed by Harvey and Peters [1990] which are seasonality, trend and holidays. The following Equation (5.20) represents the components of Prophet models that can be adjusted intuitively.

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \tag{5.20}$$

In Equation (5.20), $g(t)$ is the trend function without the periodic changes in the time series data, $s(t)$ is the periodic changes such as weekly and yearly seasonality, $h(t)$ is the holiday function which could vary country by country. The ϵ_t is the error term that shows any peculiar changes not accounted for by the model.

Some of the advantages of Prophet model are 1) flexibility; in terms of allowing for the adjustment of seasonality with multiple periods, 2) missing data; in terms of accommodating for irregularities in the frequency of data Fang et al. [2019], 3) fast; the model fitting is fast and 4) interpretability; its parameters are interpretable giving room for adjustment based on assumptions by experts. The trend model is calculated as shown in Equation (5.21).

$$g(t) = \frac{C}{1 + exp(-k(t-m))\prime} \tag{5.21}$$

From Equation (5.21), C represents the capacity, k represents the rate of growth and m represents the offset parameter. There is also "changepoints" parameter offered by Prophet in order to account for a forecasting problems with constant growth rate. This changepoints parameter is allowed to be specified by the analyst if they are aware of the dates of the event changes. Automatic changepoint selection can also be done with the Equation (5.22).

$$g(t) = (k + a(t)^T \delta)t + (m + a(t)\gamma) \tag{5.22}$$

In Equation (5.22), δ is the rate adjustment and γ is a contiguous function. The seasonal component $s(t)$ is derived as in Equation (5.23). β is used to smoothen the curve before seasonality trend. Furthermore, holidays and events components of Prophet does not have any periodic pattern, because different countries events and holidays vary. However the effects of the holidays event in each country repeats itself yearly. As a result of these differences, Prophet models allows analyst to choose the dates and the country they wish to incorporate in the forecast.

$$s(t) = X(t)\beta \tag{5.23}$$

$$Z(t) = [1(t \in D_1), \ldots, 1(t \in D_l)] \tag{5.24}$$

$$h(t) = Z(t)\kappa \tag{5.25}$$

The holiday events can be calculated as in Equation (5.24). After including the holiday and seasonal component of the Prophet model fitting, they are used to form a matrix in addition with the change point $a(t)$. Just like a typical ML model approach of "fit and predict", Prophet uses a similar Stan probability code for statistical problems discussed in Carpenter et al. [2017], which basically estimates the maximum and full posteriori Taylor and Letham [2017] in order to account for uncertainty in both its parameters and forecast.

5.3 Methods

To the best of my knowledge, one main gap observed in the literature in the previous research for indoor CO_2 prediction was that previous indoor CO_2 studies concentrated on using the CO_2 levels to forecast indoor air quality, as such did not deal with future forecast which could help to forestall any high CO_2 related health issue Pantazaras et al. [2016], Wang and Wang [2012]. Therefore, this study is proposing an hourly future forecast. This hourly forecast was chosen based on the study conducted in Frank et al. [2000], where it was advised that one should be cautions in choosing an appropriate window size in neural network methods of time series prediction. Frank et al. advised that adequate large time delay is necessary to time series predictor. However, if a window is too small or too large it could lead to network impairment and consequently loss of performance by the model.

Future indoor CO_2 concentration forecast has lots of advantages such as 1) helping in the demand control HVAC systems of a room that relies on its CO_2 levels as well as help fight climate change by ensuring that only the needed carbon is supplied to the system thereby reducing excess carbon emissions into the atmosphere, 2) helping the occupants to take timely decisions by reducing the indoor CO_2 when it is expected to be higher than normal and 3) preventing any future health breakdown of the occupants.

By using the sensed indoor environmental data, which time series and ensemble machine learning model can accurately predict indoor CO_2 levels of rooms in advance and estimate room occupancy. This proposed solution will help to answer the following research questions: **"What time in advance will the indoor CO_2 levels be predicted so that the room managers can have enough time to act and prevent any unforeseen issues that may arise."** This will in turn help room occupants to prevent or minimize any health hazard that is often associated with exposure to high level of indoor CO_2 concentrations in their various indoor environments. Moreover, this proposed solution will contribute towards helping the room managers to save energy used on HVAC during the day as well as help fight climate change by reducing the amount of carbon emissions into the atmosphere. The following section will explain why the real-time sensor called Netatmo was chosen for this study.

5.3.1 Justification of Sensor Used

In the last decade, there have been slight improvements in the manufacturing of friendly weather stations and environmental sensors. Most of these products are controlled by the wish to obtain automated frequent real time observations that are usually stored electronically for easy retrieval, analysis and data sharing. Thereby, making manual observations such as those taken by participating individuals or members at stipulated observing times unpopular these days. This improvement has enabled data scientist to access both historical and real time data that will help everyday people to take-charge of their decision making.

Expectedly, these weather stations provide data scientists with their own data at their specified locations and also helps to supplement insufficient manually recorded data with real time data. The awareness of these weather stations are steadily increasing and the observations are usually measured once daily or at the interval of 5, 10, 15 minutes Bell et al. [2013].

Therefore, in this research, a wireless sensor network known as Netatmo weather station Online Repository [a] was used to conduct all the quantitative studies. This is because the sensor can sense, measure and obtain information from real world and transmit the data to a dedicated user's account. The Netatmo weather station shown in Figure 5.5 has the

Fig. 5.5. Picture of the Netatmo sensor used for the recording of the environmental variables.

following capacity:

- **Ease of installation and data retrieval:** Guide on how to install Netatmo sensor device are found in the manual provided and it takes few unambiguous steps and minutes to install. Retrieval of the measured data is easy and can be obtained from the users' online account created during installation.

- **Frequency of the data recorded:** Just like most other sensors Netatmo sensor records data in the frequency of 5, 15, and 30 minutes simultaneously. Thereby providing the user with the different choices of data. This will assist the user to do their choice of data analysis with the aim of obtaining sufficient result.

- **The data storage capacity:** One drawback of the Netatmo sensor is its storage capacity which saved recorded data for only three months. However, not having enough storage capacity solely depends on the purpose of the study.

- **Number of environmental variables measured:** Another advantage of Netatmo sensor is that it has all-in-one measurements of environmental variables in one place making it easy for retrieval and data preprocessing. These variables are Temperature, Humidity, Noise, Pressure and CO_2.

- **Compatibility:** Netatmo sensor is compatible with various devices like Android, Windows, Ipad and Iphone. These obvious advantage endears it to its users.

- **Privacy:** Asides measuring environmental variables listed above, this device does not record voices or images of people. It is safe for any environment it is installed and does not raise any privacy issues.

- **Calibration:** The sensor calibration is automatically done based on the environment it is installed. So no need for the user to do the calibration.

5.3.2 Justification for Environmental Variables Used for the Indoor CO_2 Prediction and Room Occupancy Study

The variables recorded are CO_2 (measured in parts per million), Temperature (measured in degrees centigrade), Noise (measured in decibel), Pressure (measured in millibar) and Humidity (measured in %). The reason for choosing these variables are 1) these variables have been used and recommended in the literature Arief-Ang et al. [2018b], Chen et al. [2018a], Yang et al. [2012] for indoor air quality studies and room occupancy estimation study with relatively good results, 2) these variables are compatible and easy to model with the ML methods adopted in this research.

5.3.3 Justification for Univariate and Multivariate Studies for the Indoor CO_2 Prediction

In the literature Chen et al. [2018a], Wang and Wang [2012] used only one data point (indoor CO_2 concentration levels) to forecast CO_2 levels. As a result, dataset can only be analysed in one way which could led to low performance of some models such as neural network Kröse et al. [1993]. However with multiple data points, dataset are analysed in multi-dimensional ways. Therefore this research used both single and multiple variable in its CO_2 prediction study because the proposed model used for the analysis in this thesis is a branch of neural network known as recurrent neural network. By using more dataset, it would be possible to access and compare the performance of multivariate and univariate indoor CO_2 prediction with neural network model. For instance, this study will analyse 1) the relationship between the CO_2 data and other environmental variables with regards to its concentration levels in rooms and 2) insights about the peak and fall periods of the environmental variable in rooms will be investigated.

5.3.4 Choice of Locations for the Indoor CO_2 Prediction Study

In this study, the indoor CO_2 prediction was conducted in two different offices. The choice for the two offices was dependent on the following criteria:

- **Availability of the venue:** Because of concerns for occupants' privacy and interest in the project, some of the rooms that the research study would have benefited from were not permitted to be used for the study. As such other locations that were both readily available for the study were used.

- **Constant power and internet reception:** Due to the fact that the CO_2 sensor used for the study needs steady constant power and internet supply. Any location of choice must have these two requirements before been considered for the study.

- **Security of the venue:** To avoid theft of the study materials or interruption of the study which often leads to missing data, the security of the study locations must be guaranteed.

The Post-doc room Figure 5.2 has between 8 to 10 seating capacity. It also has only one door for entrance and exit of the occupants. This room was used for the confirmatory indoor CO_2 prediction study. These two rooms are different from each other with respect to the number of seating capacity and room occupants which will ultimately impact the result of the analysis.

5.3.5 Sensor Set Up, Configuration, Location and Data Collection

The Netatmo sensor used in this research study was the indoor (bigger) module shown in Figure 5.5. During the setup and configuration of the indoor module, few processes were involved. The indoor module were configured with a Windows 10 laptop in the following manner. Firstly, the indoor sensor was connected to the laptop with a provided USB cable for installation on the laptop. After the sensor has been installed on the laptop. Its USB cable was then unplugged from the personal computer (PC) and connected to the wall adapter to ensure steady power supply. The module's setup location was not considered during its configuration because the choice of best location is still an open problem in the literature Patwari et al. [2003].

The recorded environmental data of the indoor module was stored in a Netatmo cloud online account and was accessed using a Wifi access point. The results was then downloaded as a "csv" file for data analysis on to Jupyter (Ipython) notebook Shen [2014]; an ideal platform notebook for data visualization and analysis Hamrick [2016]. The indoor module measures five environmental variables; temperature, CO_2, humidity, pressure and noise. The following were stated in doc [2020] as the indices used by the sensor for the measurement of the environmental variables.

1. **Temperature:** The indoor comfort temperature is between 20°C and 24 °C in winter and between 23°C and 26°C in summer. The instrument used for measuring temperature is thermometer. Comfort temperature depends on the humidity level. the dryer is the air the better is the comfort temperature.

2. **Pressure:** Barometer is used for the measurement of the pressure and its SI unit is inHg. The pressure values of the sensor represents the mean sea level brought about by the ground elevation. The pressure measurement is basically matched by the measurements of different elevations such as sea, mountain and valley. The mean sea level is usually reported by the TV and the radio stations across the world. The barometric pressure of the indoor and the outdoor are always the same.

3. **CO_2:** Indoor CO_2 concentration levels is measured by the CO_2 sensor by a method know as optical process. This means that once there is an emission of light by the light bulb and IR receiver embedded in the sensor, the CO_2 in the air partially absorbs it. Increase in CO_2 causes increase in the light being absorbed by the CO_2 sensor. The CO_2 are produced by activities of human which tends to concentrate in a confined place. The unit of CO_2 is parts per million (ppm). Both the indoor and the outdoor module measures the CO_2 levels. Typical indoor CO_2 comfort levels is between 350ppm to 1000ppm. While the CO_2 levels for outdoor air in usually 400ppm.

4. **Humidity:** Hygrometer is the instrument used for measuring humidity. Humidity level for indoor comfort is between 30% to 70%. Netatmo sensor measures relative humidity which depends on the amount of water vapour in the air at a specific temperature. Both the indoor and the outdoor module measures the relative humidity. The sensor was placed in a dry non-wet environment to prevent the station from being saturated which has the tendency to increase the humidity levels.

5. **Noise:** Noise measurement otherwise known as Acoustic comfort level is measured in decibel (dB). The sensors measures average noise level using a sound meter in the interval of 5 minutes.

6. **Calibration:**The CO_2 sensor is frequently and automatically recalibrated. The process of recalibration is done based on how low the level of CO_2 concentration surrounding the station occurs weekly; typically once a week. Proper ventilation of the room ensures precise recalibration.

5.3.6 Room Information for Indoor CO_2 Prediction Study

The overall description of the data set for the indoor CO_2 prediction study are itemized as follows:

1. There are two different experiments that were carried out in order to test the hypothesis mentioned in Section 5.1.2 above. The two rooms that the experiments were conducted are situated on the thirteenth floor Figure 5.2 of the Livingstone tower in the University of Strathclyde (Post-Doc) and at SwarmOnline office. The Post-Doc room takes in an average of eight to ten staff daily of the university, while the Swarm office accommodates approximately 26 employees.

2. The appropriate location to place the weather station for optimal performance was not advised in the documents and is a known open problem currently.

5.3.7 Description of Dataset Used in this Study

Bellow are the summary statistics for the total dataset recorded from the two rooms used for the indoor CO_2 prediction study:

- There are 9180 observations for the (Post-Doc dataset with 5 columns (variables). The dataset was recorded from 07/05/2017 at 11.55am to 05/06/2017 at 11.52am for twenty four hours a day and four weeks a month. The dataset was recorded for every 5 minutes interval. Though there were +/- 3 minutes difference in the recorded times.

- There are 4011 observations for the Swarm-CO2 dataset with 5 columns (variables). The dataset was recorded from 01/10/2016 at 02:15 to 31/10/2016 at 23.52pm for twenty four hours a day and four weeks a month. This dataset was recorded for every 10 minutes interval of time.

- After the datasets recording, all the weekend data were removed in order to reflect week day (Mondays to Fridays) data only. The reason for this data removal was because occupants are only present in that room on week days.

- Each of the observations tracked and recorded from an independent device. The variables recorded are CO_2 (measured in parts per million), Temperature (measured in degrees centigrade), Noise (measured in decibel), Pressure (measured in millibar) and Humidity (measured in %).

Date-time	temp	hum	co2	noise	pres	day-of-week
2017-05-04 11:55:00	24.0	37	392	48.0	1022.0	Thursday
2017-05-04 12:00:00	24.0	37	411	49.0	1021.9	Thursday
2017-05-04 12:05:00	24.0	37	398	52.0	1021.8	Thursday
2017-05-04 12:10:00	24.0	37	409	51.0	1021.8	Thursday
2017-05-04 12:15:00	24.0	37	401	57.0	1021.8	Thursday

Table 5.1: First five rows for dataframe obtained from the Post-Doc room for the confirmatory CO_2 prediction study

Date-time	temp	hum	co2	noise	pres	day-of-week
2016-10-01 02:15:02	20.3	36.0	837.0	72.0	1004.0	Saturday
2016-10-01 02:25:09	20.3	35.0	802.0	72.0	1004.0	Saturday
2016-10-01 02:55:29	20.0	35.0	739.0	72.0	1003.0	Saturday
2016-10-01 03:05:36	19.9	35.0	718.0	72.0	1003.0	Saturday
2016-10-01 05:06:55	19.2	36.0	581.0	72.0	1003.0	Saturday

Table 5.2: First five rows for dataframe obtained from the Swarm-CO2 room for the preliminary CO_2 prediction study

- These dataset were recorded into the device and extracted as a “csv” file from the online account. The data covers a period of one month (30 days) from each room. There was no case of a missing or duplicate data. All the datasets are of float data type. The dataset is easy to understand.

- It can be said that for any given indoor Temperature combined with Noise, Humidity and Pressure, that there is a corresponding indoor Carbon dioxide concentrations been recorded regardless of time it was recorded. CO_2 is the target problem (expected be to predicted) also known as “dependent” variable whereas Temperature, Humidity, Pressure and Noise will serve as the independent variables also known as “predictors”.

- This problem will be treated as both the multivariate and univariate time series problem because they are more than one input variable and it is dependent on time. The model developed will show the relationship between all of the independent variables and the dependent variable.

5.3.8 Dataset Visualization for Swarm-CO2 and Post-Doc

Sometimes, there are challenge in understanding the pattern or trend of data especially when that data is to be used to make forecast for the future. This problem always affect both structured and unstructured data. However, deep learning has shown that it is easy to learn patterns of different kinds of data in a network form. Therefore in this study, LSTM of recurrent neural networks in Keras was used to analyse long sequence of structured data from Swarm-CO2 dataset. This has aided in further explaining the data’s short and long term dependencies and its temporal differences.

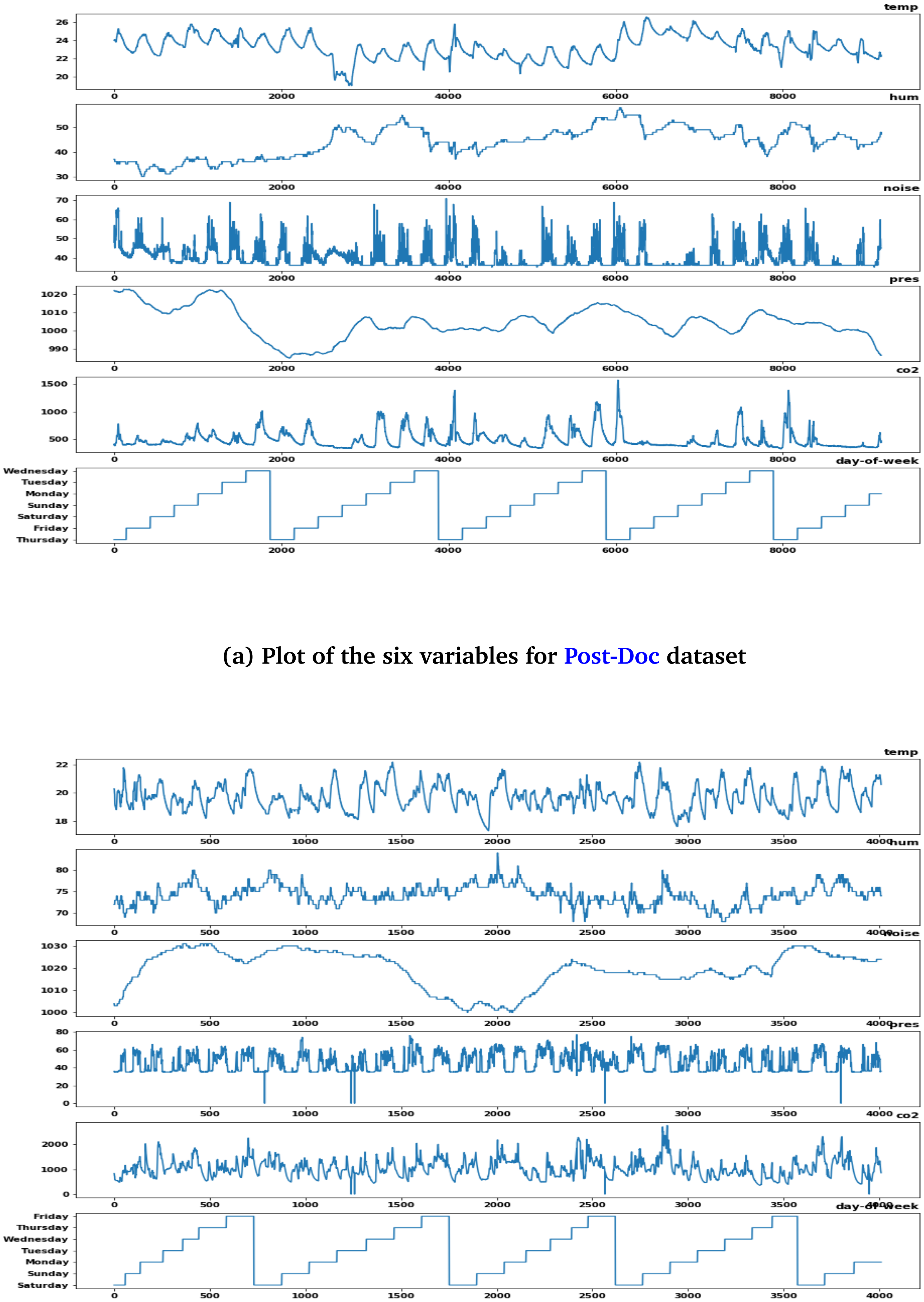
The visual representation of the two plots shown in Figure 5.6 represents the two plots of all the variables; CO_2, temperature, noise, humidity and pressure. Comparing the two plots, CO_2 shows an inconsistent (upward and downward) pattern of the CO_2 levels, thereby giving indication that the CO_2 data is stationary. In Figure 5.6b, the pattern of humidity , temperature and CO_2 are the same, while noise has a complete different cyclic pattern of upward trend and sharp downward pattern direction. The pressure data pattern has some few outliers with different levels of pressure. Similarly, in the Figure 5.6a, the CO_2 and noise have similar pattern of upward and downward movement including its rate of level rise. The humidity have opposite pattern with pressure and temperature at some point in the observation and similar downward and upward pattern at every other places.

Another plot with additional feature; day of the week was created out from the “Date-time” column. The two plots are shown in Figure 5.7 and they represent the weekly plot for only the CO_2 data versus day of the week. The Post-Doc dataset visualisation plot shown in Figure 5.7a shows that the CO_2 levels are usually higher during the week day with Wednesday and Thursday been the peak day, while the lowest CO_2 levels are weekends; Saturdays and Sundays. Furthermore, it has a CO_2 pattern where the CO_2 concentration levels begins to build up from the middle of the week (Wednesday) to Friday and sharply falls on Saturday till Sunday. This is an indication that no or fewer students are present in the room on weekends.

On the other hand, the company office shown in Figure 5.7b shows brakes on Sundays and Saturdays in its CO_2 levels though, some levels are above 2000ppm. This could be a case of CO_2 build up from during the week and poor ventilation that prevents its diffusion. However, Mondays, Thursdays and Fridays shows high CO_2 levels of above 2000ppm. While, Tuesdays records the least CO_2 levels.

In Figure 5.7a and Figure 5.7b, similar high CO_2 concentration levels were recorded during the week days. The average CO_2 levels throughout the week in the Swarm-CO2 room is slightly higher than 2000ppm, whereas the average CO_2 levels throughout the week in the Post-Doc room is close to 1000ppm. The reason for the difference in the average CO_2 levels of the two rooms is because Swarm-CO2 is more busy than the Post-Doc. This helps to confirm the fact that humans are a good source of CO_2 generation in indoor environment. The python code for plotting the visualization of the Swarm-CO2 and Post-Doc dataset can be seen in Listing 5

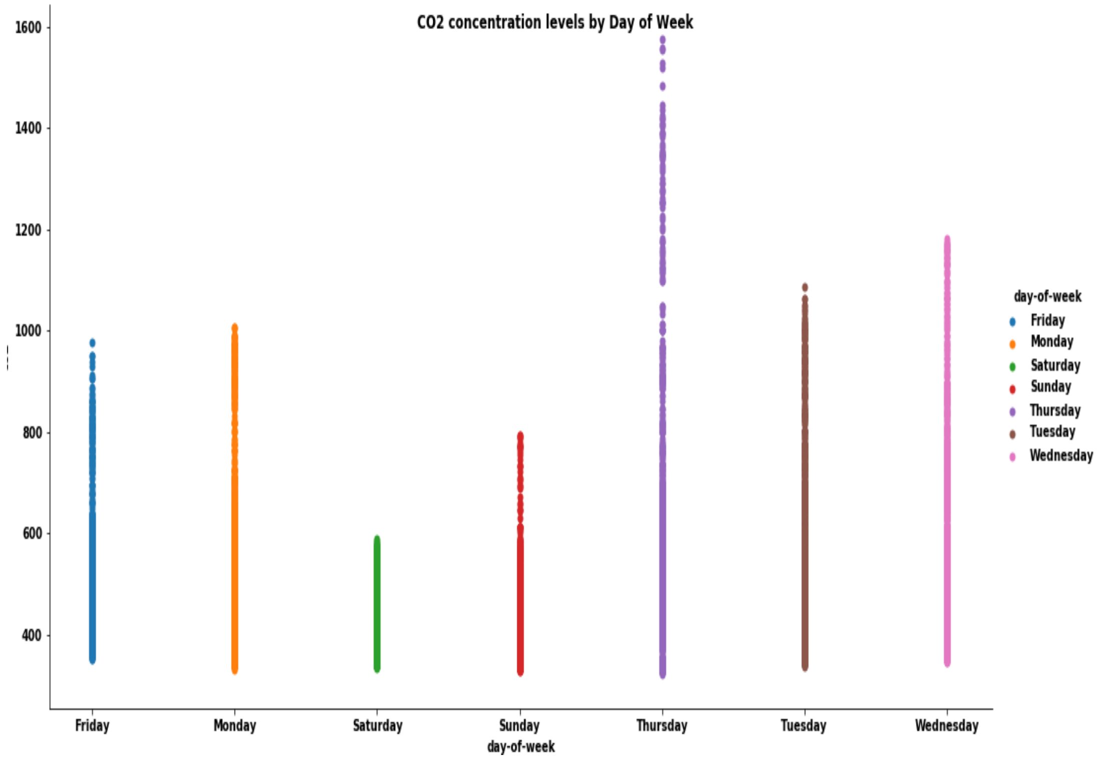
Based on the results of the dataset visualization shown in Figure 5.6 and Figure 5.7, this study will now treat this CO_2 prediction study as a multivariate and univariate time series using the two models (LSTM and Prophet) mentioned above.



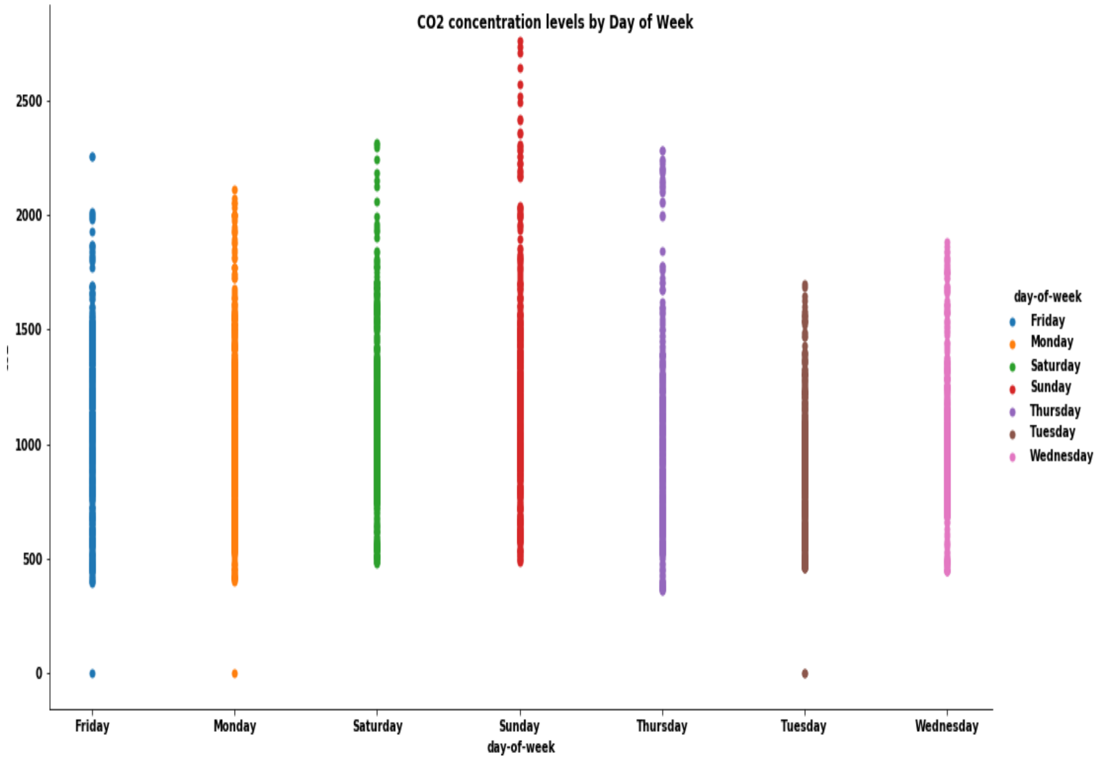
(a) Plot of the six variables for Post-Doc dataset

(b) Plot of the six variables/column for Swarm-CO2 dataset

Fig. 5.6. Dataset Visualization for Swarm-CO2 and Post-Doc Dataset



(a) Plot of only the CO_2 vs day of the week for the Post-Doc dataset



(b) Plot of CO_2 vs day of the week column from the Swarm-CO2 dataset

Fig. 5.7. Day of the week plot for Post-Doc and Swarm-CO2 dataset respectively

5.4 Preliminary Study Analysis

5.4.1 Testing for Stationarity with Post-Doc Dataset

In this research, stationarity will be tested on all the features (temperature, humidity, pressure and noise). To do that, the Johansen test Lütkepohl [2005] for stationarity was used. In Johansen test, the eigenvalues were checked. If the eigenvalues return zero, it means that the series are not cointegrated and that the series are stationary. Whereas if the eigenvalues returns negative values, it means that the series is not stationary and that stationarity can be created.

In order to run Johansen test, $coint\_jojansen$ method by statsmodel was used. From the Johansen function parameters, the -1 value of the second term shows that the series has a time trend in the polynomial, while the third term value of 1 specifies the number of lag differences. In Listing 3, the eigenvalues returned from the Johansen test are all less than zero, which shows that the series (all the features) are stationary.

5.4.2 Univariate Forecasting Steps with Swarm-CO2 Dataset

5.4.2.1 Long Short-Term Memory Model for Univariate Indoor CO_2 Prediction

LSTM (please see Section 5.2.4) recurrent neural network is believed to effortlessly model univariate and multivariate time series problem Gers et al. [1999]. LSTM is a great benefit to time series forecasting because linear methods are difficult to adapt to multivariate forecasting. Before the construction of LSTM model, the following data preprocessing and data engineering was done to the Swarm-CO2 dataset and the python code and all of its output can be seen in Section P:

1. **Ensure that the Swarm-CO2 dataset are all float.**

2. **Normalize the features of the Swarm-CO2 dataset to have be within the range (0 to 1).**

3. **Split the normalized Swarm-CO2 dataset into training and test set.**

4. **Convert the matrix into supervised learning of the inputs.**

5. **Reshape the input features into 3D shape (number of samples, number of time steps, number of features).**

6. **Choose look-back value as 12 for 1 hour forecast and 24 for 2 hour forecast.** Here look-back is a back-propagation through time method of RNN where the ordered series links one time step to another. This means that it is the number of time steps in a window.

During the analysis, the LSTM model architecture was defined with 100 neurons in the first hidden layer and one neuron in the output layer for predicting indoor CO_2 levels. The two input shapes are 1) 12 time step with 11 features and 2) 24 time step with 11 features. All these are shown in Section P. The "Dropout" layer was 20%. The Mean Square Error (MSE) was used as the loss function with efficient Adam version of stochastic gradient descent. The model was then fitted with 20 epochs and 35 batch size.

A univariate prediction of indoor CO_2 concentration levels was done using 12 and 24 time steps approach as mentioned above. However, only the target variable; CO_2, was used against the date-time stamps. The LSTM model loss (MSE) value summary after fitting the training and the test set for 24 and 12 time steps are shown in Section P. From the results summary, the lowest MSE value for the two time steps is 0.0025 and 0.0024 respectively. MSE of 0.0025 is good, but it will be compared with the MSE results of the multivariate time series when fitted in the coming section.

Furthermore, the performance indices; RMSE for the 24 and the 12 time steps for univariate indoor CO_2 prediction 0.0341 and 0.0347 respectively.

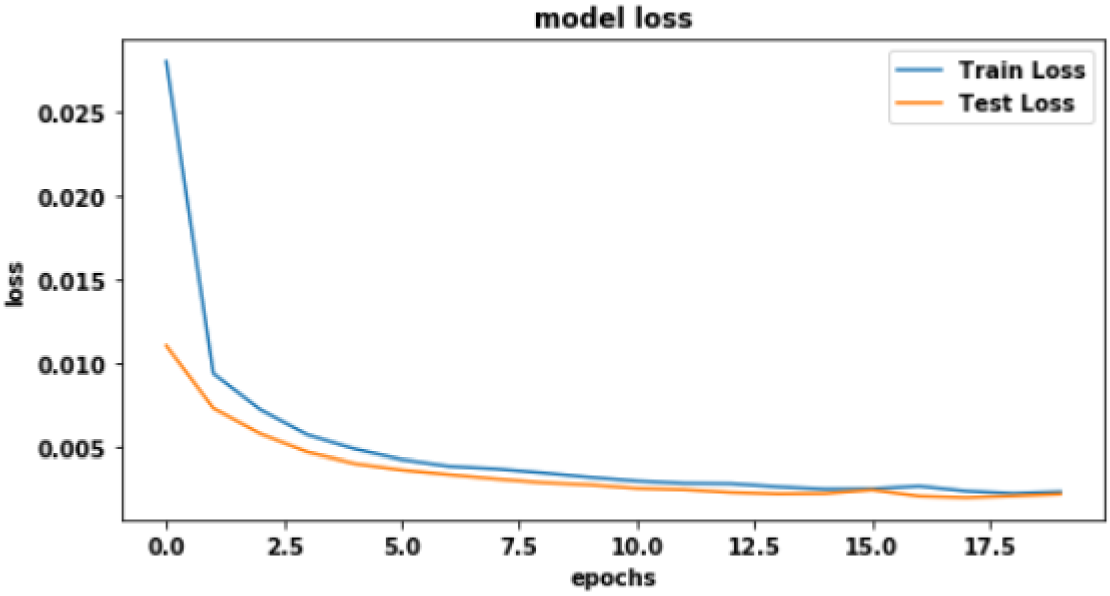


Fig. 5.8. LSTM model 24 time steps loss of train versus test set for univariate CO_2 prediction for Swarm-CO2 dataset

The loss plot of the training versus test set for univariate indoor CO_2 prediction using 24 and 12 time steps shows almost the same pattern of loss and they are shown in Figure 5.8 and Figure 5.10 respectively. The LSTM model loss plot after running 18 epochs shows that the training and the test set loss are closely aligned with each other with a loss value of about 0.010. At some point the model achieved a loss value for training set at above 0.025 after running the initial epochs. Similarly, the univariate actual versus the predicted values for the 24 and 12 time steps using LSTM are shown in Figure 5.9 and Figure 5.11 respectively and the two plots are very similar to each other with a very close pattern between the actual and the predicted CO_2 levels.

The Figure 5.9 and Figure 5.11 are the two performance plot for the univariate times series with LSTM. In the two performance plots, the actual CO_2 levels are shown in blue while the predicted values are shown in red.

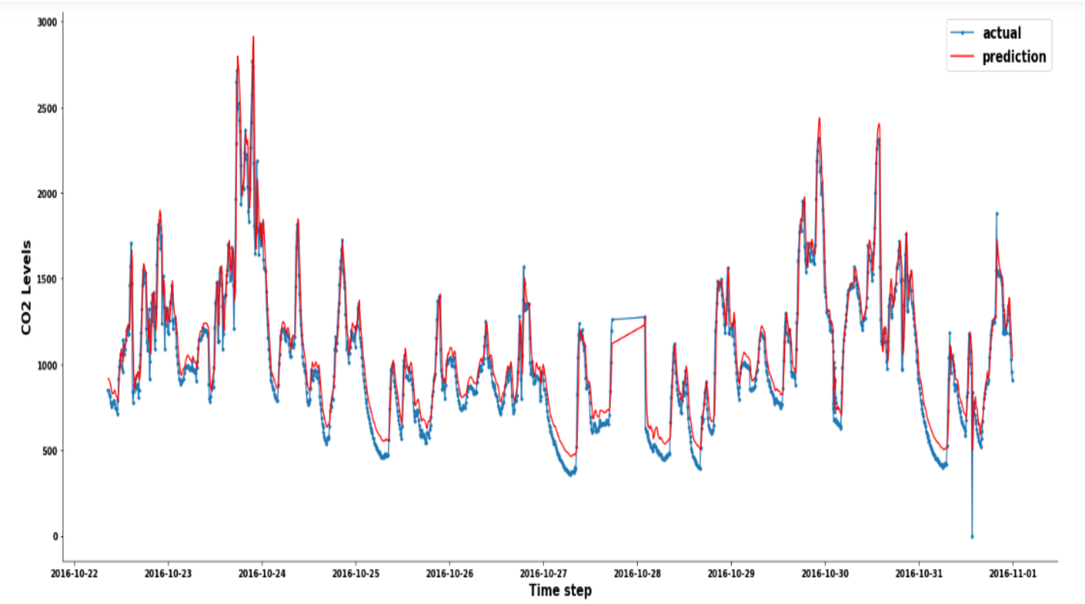


Fig. 5.9. 24 time steps univariate prediction versus actual of last of the test set with LSTM model using Swarm-CO2 dataset

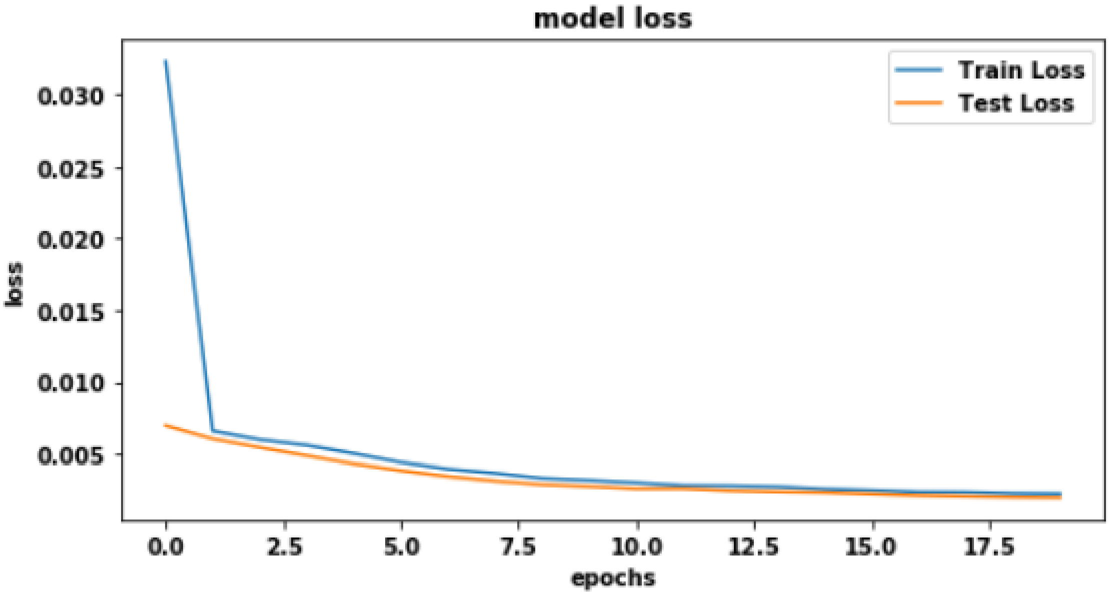


Fig. 5.10. LSTM model loss of train versus test set for univariate time series prediction of 12 time steps using Swarm-CO2 dataset

5.4.2.2 Prophet Model for Univariate Indoor CO_2 Prediction

Prophet model has been discussed in Section 5.2.5 in more detail. Facebook's data scientists believe that other models for predicting time series problem are not strongly built for missing data such as holiday events. This is because it is believed that most time series problems are inherently faced with missing data problems during observation Fang et al. [2019]. As a result of these limitations, facebook developed its own model similar to Sklearn model API by creating its own class instance known as Prophet. Expectedly,

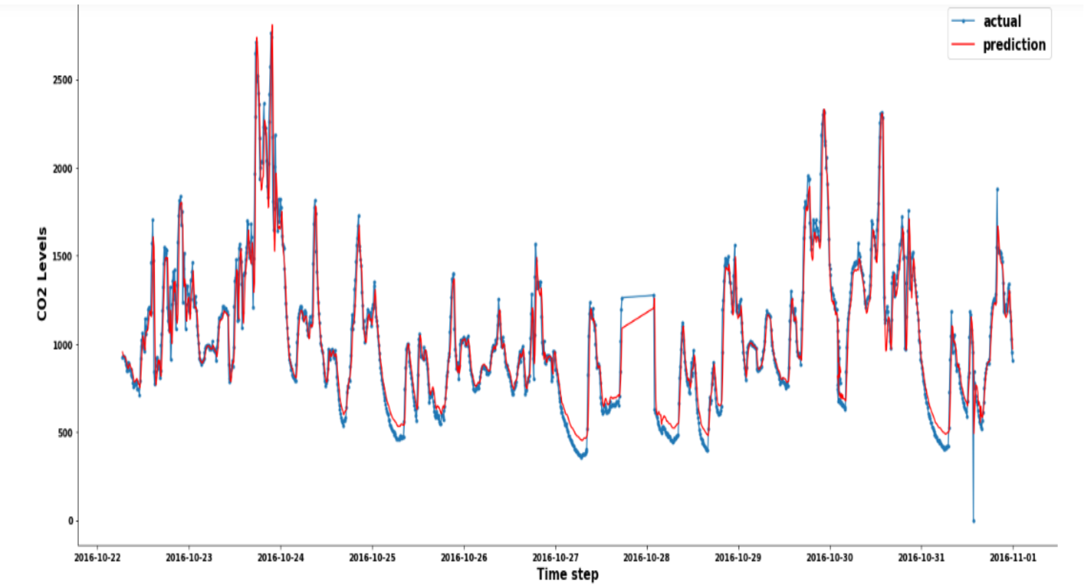


Fig. 5.11. 12 time steps univariate prediction versus actual of the test set with LSTM model using Swarm-CO2 dataset.

Prophet's class instance has fit and predict class just like a classical ML model. Prophet is also believed to have the ability to predict highly seasonal and non seasonal data such as yearly, daily, hourly and holiday effect Asha et al. [2019], Fang et al. [2019]. Prophet has many advantages of Bayesian statistics that favours seasonality and domain knowledge. Based on these advantages, Prophet was applied on the Swarm-CO2 dataset without calling the Prophet additional regressor method which is meant for multivariate time series prediction. Figure 5.12 and Figure 5.13 represents the plots of the predicted CO_2 versus actual CO_2 for the 12 and 24 time steps respectively. All the python codes and the results can be seen in Section P.

In order to build the model for the fitting of 5 minutes frequency of observation with Swarm-CO2 dataset two arguments of the Prophet class instance known as "*changepoint_prior_scale*"; for adjusting trend flexibility, and *changepoint_range*; for automatic detection of change points were used. The purpose of using the two arguments was to detect the actual trajectory of the change points and to allow for the trend and frequency to adapt well with the Prophet model.

Another option used was the addition of holiday option for the country; United Kingdom (UK). UK was chosen because the datasets were recorded in UK. The reason for the addition of holiday event in Prophet model was because of the inherent unnatural event that often causes the trend of the dataset to deviate from its baseline and return to normal once the holiday event is over.

Furthermore, the changepoints range parameter will place the potential changepoints in the first 90% of the series. While the change point scale of 0.05 will make the trend more flexible and vice versa when the values are reduced. The changepoints scale will be 0.05

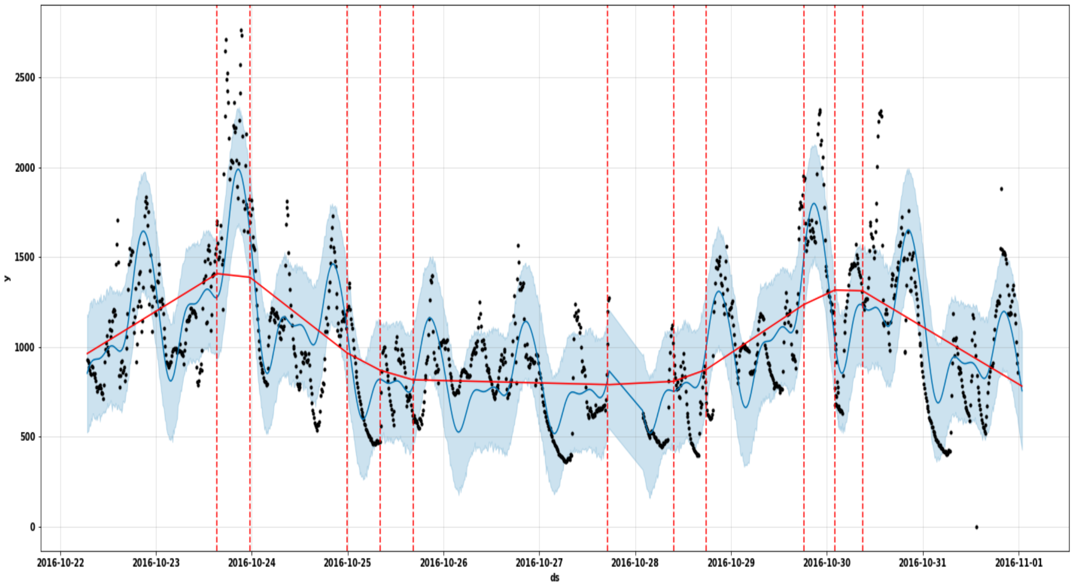


Fig. 5.12. 12 time steps univariate prediction versus actual with Prophet model using Swarm-CO2 dataset

because of the 5 minutes interval and the changepoints range will be 0.9. Furthermore, before fitting the Prophet model with the Swarm-CO2 dataset, the dataset was splitted into 33% test set and 67% training set for training and validation purposes.

In addition to the dataset split, the periods parameter accepts no less than hourly frequency, thereby making us to assign the period parameter of the model either 1 for one hour time steps or 2 for two hour time steps of advance forecast. This means that the last date-time of the observed data available in Swarm-CO2 dataset was 2016-10-31 23:52:22 and the future forecast for 1 hour by Prophet will be 2016-11-01 00:52:22. While the forecast for next 2 hours will start from 2016-11-01 00:52:22 and ends at 2016-11-01 01:52:22 All the python codes are shown in Section P. Interpreting the two plots; Figure 5.12 and Figure 5.13, the actual CO_2 values are shown in black dots and the forecast values in blue line.

The area shaded with the light blue color is the confidence interval. Obviously, it can be assumed that the Prophet model performed very well in univariate time series prediction more than its multivariate, because the area of the confidence interval covers all the two lines (actual and predicted) and the predicted and the actual lines closely have the same pattern. However, there are many vertical dashed red lines that represents the change points that were identified in the CO_2 levels which is an indication of continuous fluctuations of the CO_2 data against time, just like the pattern of the changed point events of the CO_2 against other variables shown in Figure 5.18.

The MSE performance indices of 12 and 24 time steps of univariate indoor CO_2 prediction with Prophet are 197.613 and 199.231 values respectively, while its RMSE are 253.898 and 255.856 respectively. Though these performance indices values seem high, but its

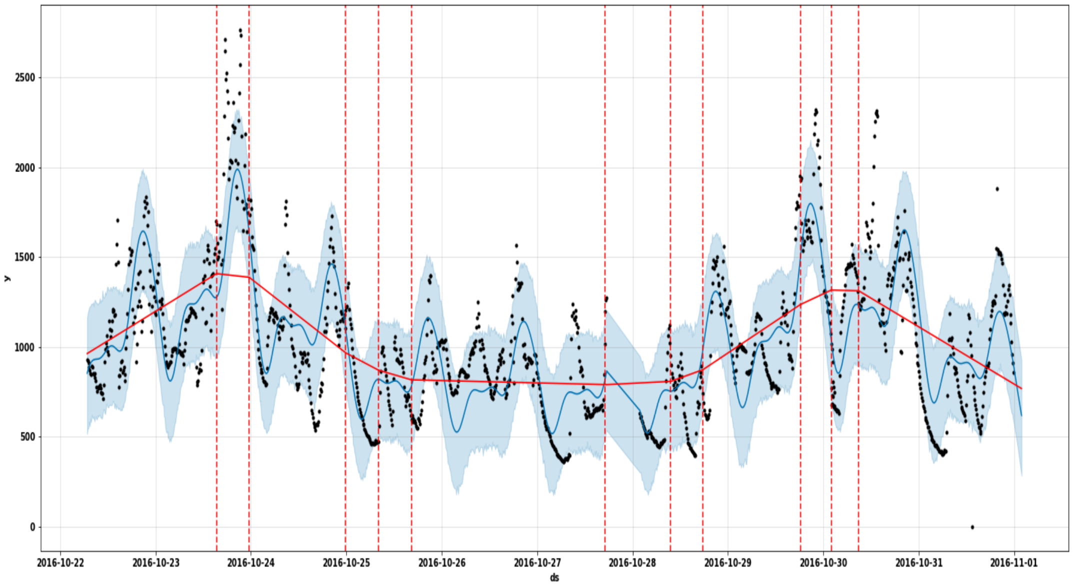


Fig. 5.13. 24 time step plot of actual versus predicted for univariate prediction with Prophet model using Swarm-CO2 dataset

plot are closely aligned. Furthermore, the Prophet seasonality component plot for the univariate CO_2 prediction is shown in Figure 5.14.

In Figure 5.14, the trend component plot of the univariate time series shows CO_2 concentration levels that rises within 2 to 3 days and falls sharply after four days and the cycle continues. The holiday component event did not show any unusual holiday event throughout the period of observation. The daily CO_2 concentration levels shows peak periods from 8pm in the night to the early hours of the morning with a stabilizing and lower levels from 8am to 4pm daily. The rise of the CO_2 concentration levels in the midnight to the early morning might be as a result of poor ventilation that does not allow enough air to circulate in to the room.

5.4.3 Multivariate Forecasting Steps with Swarm-CO2 Dataset

5.4.3.1 Long Short-Term Memory for Multivariate Indoor CO_2 Prediction

In the multivariate time series for the indoor CO_2 prediction, the same method used in Section 5.4.2.1 was used. Also in the Section P, the first part of the data preprocessing was the conversion of categorical variable; “day-of-week” into a dummy variable through a method known as OHE, which has been discussed in Section 5.2.2.2. After the conversion all the resultant features were scaled into a uniform range of 0 and 1 using $MinMaxScaler$ method by the Sklearn. After the scaling, the data was split into training 66% training set and 34% test set while maintaining the initial order of the dataset as time series data requires. As it can be seen in Section P, the following steps shown in were followed in order to fit and predict the CO_2 levels of the Swarm-CO2 dataset using LSTM model:

1. The next step was to split the normalized dataset into input features and target features. The columns named “temp”,“hum”,“noise”,“pres”, ’_Friday’, ’_Monday’,

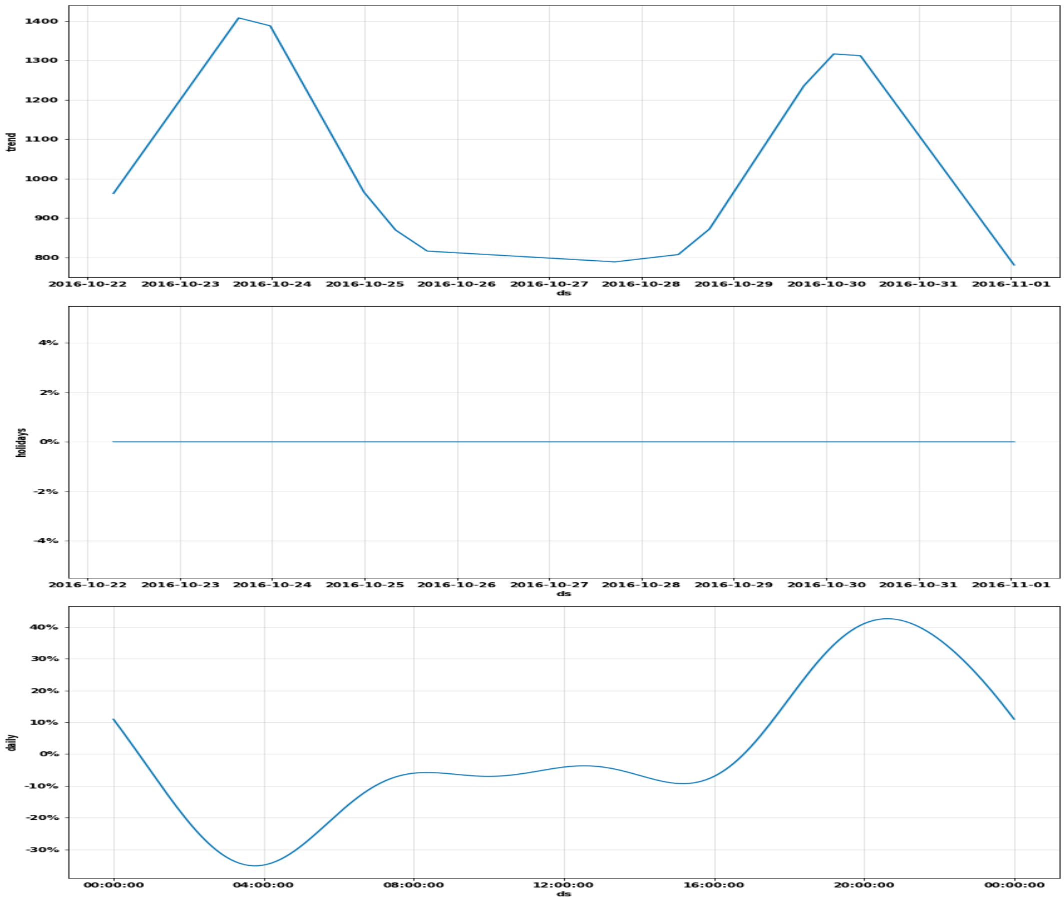


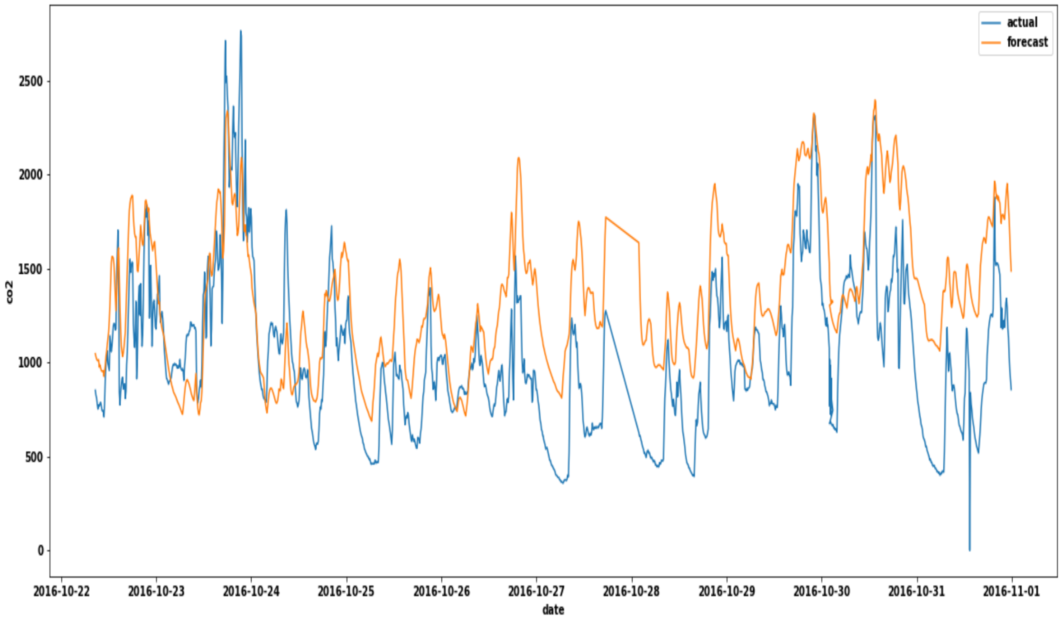
Fig. 5.14. Swarm-CO2 dataset seasonality plot for univariate CO_2 prediction with Prophet model

'_Saturday', '_Sunday', '_Thursday', '_Tuesday', '_Wednesday' are the input features making it a total of 11 input features, while the column named "co2" is the target feature. Please note that this first part of the analysis was treated as a multivariate time series.

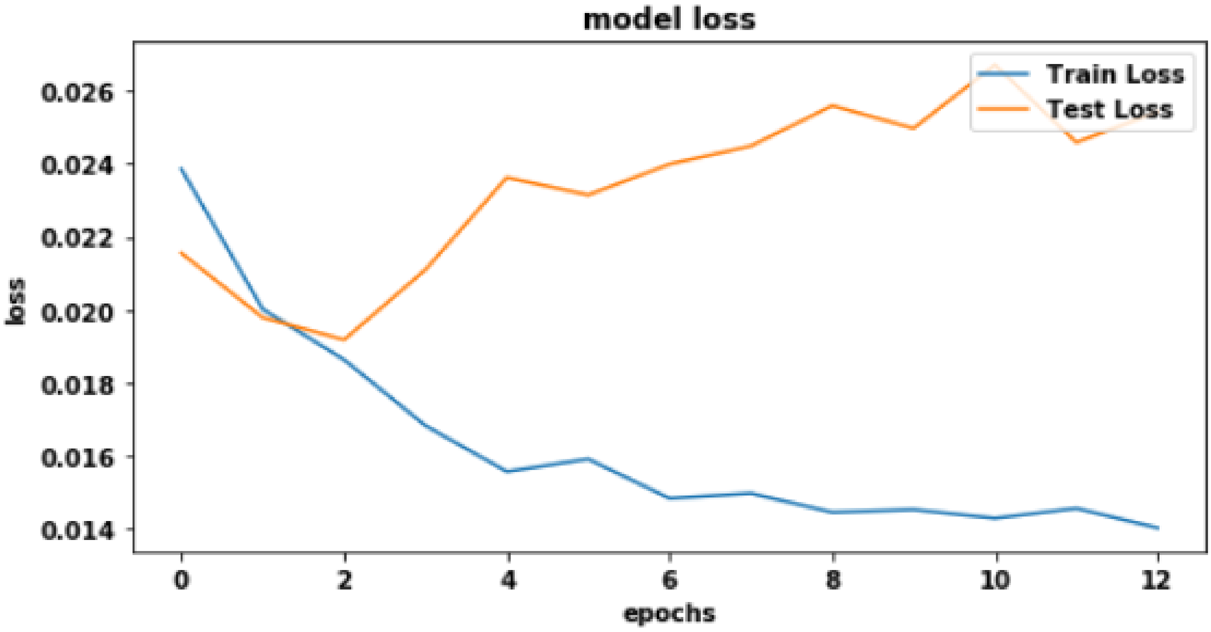
2. All the 11 input features and the target feature were converted into a supervised learning.

3. The 11 features were reshaped in to 3D array format of (number of samples, number of steps, number of features). The (3107, 24, 11) was for 2 hour look back and (3107, 12, 11) is for 1 hour look back. The reason for t=12 (5*12 = 60 minutes) and t=24 (5*24=120 minutes) was because the dataset has a 5 minutes frequency interval.

The LSTM model architecture was defined with 100 neurons in the first hidden layer and one neuron in the output layer for predicting indoor CO_2 levels. The two input shapes are 1) 12 time step with 11 features and 2) 24 time step with 11 features. All these are shown in Section P. The "Dropout" layer was 20%. The MSE was used as the loss function with efficient Adam version of stochastic gradient descent. The model was then fitted with 20 epochs and 35 batch size.



(a) 24 time steps Actual vs Predicted CO_2 Level with LSTM for multivariate time series



(b) Train vs Test loss plot for LSTM on Swarm-CO2 with 24 time steps

Fig. 5.15. 24 Time steps of actual vs predicted CO_2 and model Loss

After fitting the training dataset with the LSTM model for 24 and 12 time steps prediction and running 35 batch sizes, the two loss plot obtained from the two steps are shown in Figure 5.15b and Figure 5.16b respectively. The test loss is shown in orange while the train loss is shown in blue. For the 24 time steps shown in Figure 5.15b, the model loss (MSE) of the training set are lower (about 0.016) than that of the test set after running 2 epoch cycle. While in the Figure 5.16b the model loss for the test set were within 0.025 as it was in the 24 time-step.

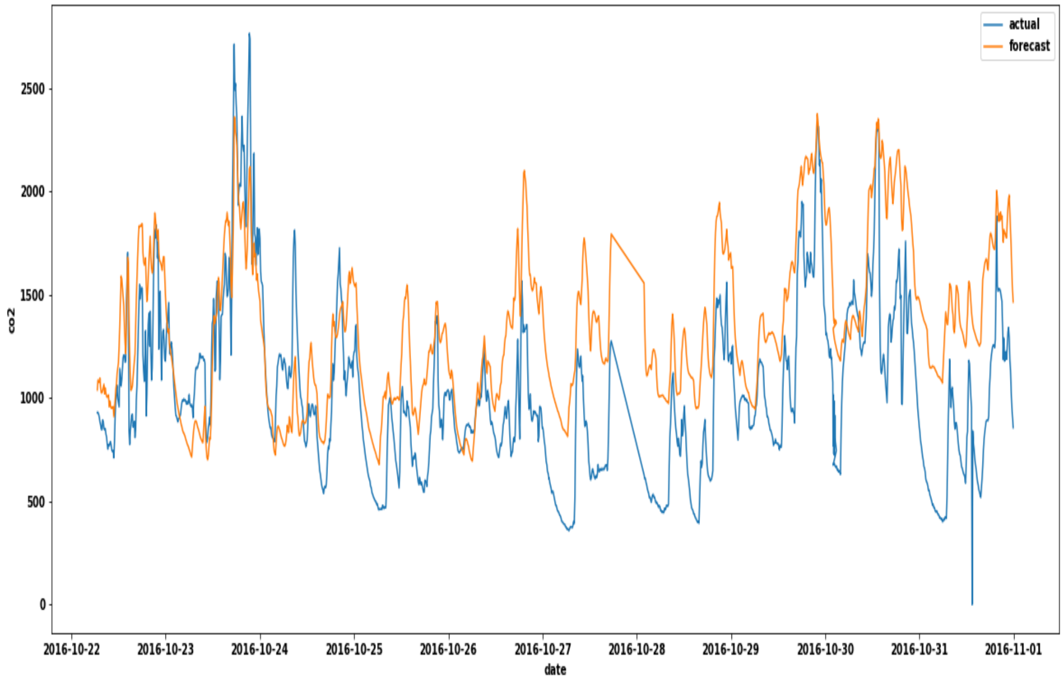
Conversely, the plots shown in Figure 5.15a and Figure 5.16a represents the actual versus predicted for the 24 time steps and 12 time steps respectively. The actual CO_2 is shown in blue line, while the predicted is shown in orange line. The pattern of the prediction plot for the two time steps are the same. In the two multivariate plots, the LSTM model made some wide predictions form 27/10/2016 to 29/10/2016. Other predictions were closely related to the actual CO_2 levels.
Additionally, the performance indices of the RMSE for 24 and 12 time steps are 0.0347 and 0.0341 respectively, while its loss value (mse) are 0.0024 and 0.0025 respectively. These closely related results is an indication that the LSTM model performance for the 12 and 24 time steps are almost the same. Moreover, the performance indices for the univariate time series of the indoor CO_2 prediction are indications that the LSTM performed better in the univariate series than the multivariate series.

5.4.3.2 Prophet Model for Multivariate Indoor CO_2 Prediction

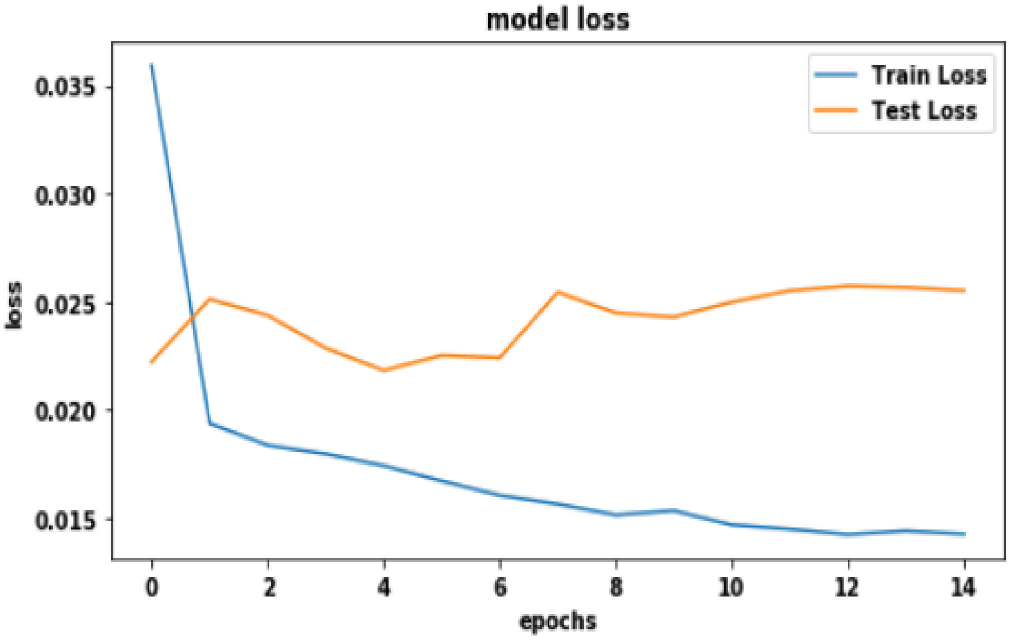
Prophet was applied to the Swarm-CO2 dataset with the same method used in Section 5.4.2.2 for the prediction of indoor CO_2 concentration. The full python code for the multivariate indoor CO_2 prediction using Prophet are shown in Section P. During the fitting of the dataset, the same changepoint scale of 0.05 and multiplicative data decomposition option were used. The same 33% test set and 67% training set that was used for univariate Prophet CO_2 prediction were also used for the multivariate series prediction.

Unlike LSTM, Prophet model does not require a compulsory preprocessing of the dataset into float or data scaling. As a result, in multivariate time series prediction of indoor CO_2 with Prophet, all the columns input features (temperature, humidity, pressure, noise and categorical variable (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday and Sunday) and target feature (CO_2) of the Swarm-CO2 were used as they were without any data scaling. Moreover, Prophet model uses a method known as additional regressors ($add\_regressor$) to compute a multivariate series.

Furthermore, Prophet has inbuilt method of dataset decomposition known as "multiplicative" seasonality, where non linear series with exponential trends are considered based on some compounding effects occasioned by seasonality effects such as country holidays, weekends etc.. Multiplicative data decomposing was used in this multivariate CO_2 prediction in order to make the quality of the forecasting model better. Figure 5.17 and Figure 5.18 are the plots of the 24 (2-hour) and 12 (1-hour) time steps with Prophet model respectively.
The performance indices; MSE of the 24 and 12 time steps using Prophet for multivariate



(a) Actual vs Predicted CO_2 Level with LSTM for multivariate 12 time steps



(b) Train versus Test Loss plot for LSTM on Swarm-CO2 with 12 time steps

Fig. 5.16. 12 Time Steps actual vs predicted and model Loss

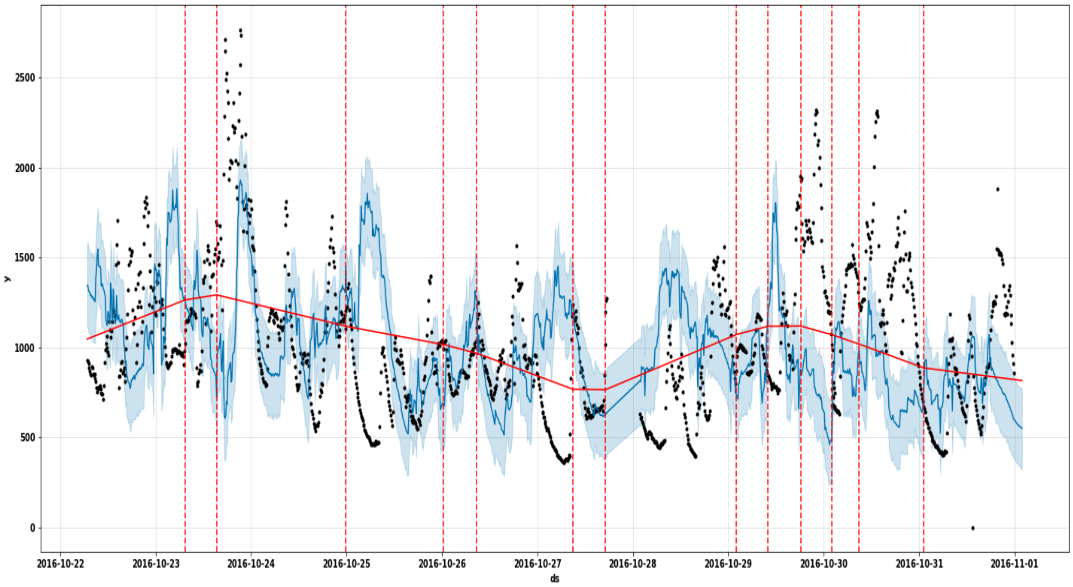


Fig. 5.17. 24 time step prediction of actual versus forecast for multivariate time series with Prophet and Swarm-CO2 dataset

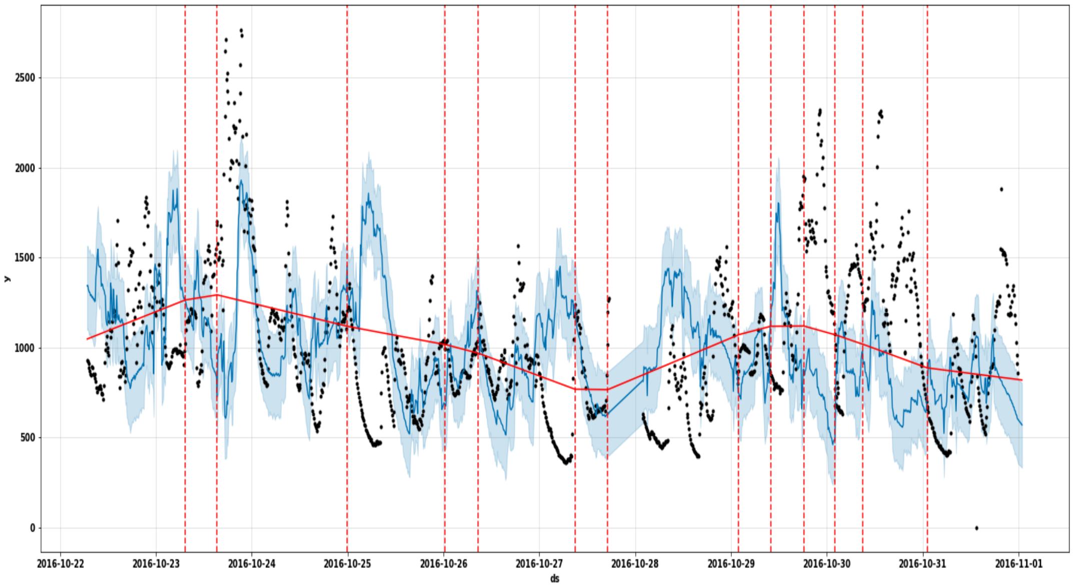


Fig. 5.18. 12 time steps actual versus forecast for multivariate time series with Prophet and Swarm-CO2 dataset

series prediction are 394.59 and 393.85 respectively, while its RMSE for the same 24 and 12 time steps are 513.87 and 514.280 respectively. Figure 5.18 and Figure 5.17 shows the multivariate forecast with Prophet model for 24 time steps and 12 time steps. The performance indices values of the two steps for either MSE and RMSE are very similar to each other with a difference of less than 1.

In the actual versus the predicted CO_2 plot shown in Figure 5.18, the actual CO_2 values are shown in black dots and the predicted values shown in blue line. The area with the light blue color is the confidence interval. The two plots for the two time steps with Prophet model are very similar, though in most area of the lines, the upward and the downward pattern of the actual CO_2 values differs from the predicted CO_2 values. But the area of the confidence interval light blue closely covered about 70% of both the actual and the predicted plot lines. One can assume that the performance of the model from the two plots are not bad looking at the two plots good.

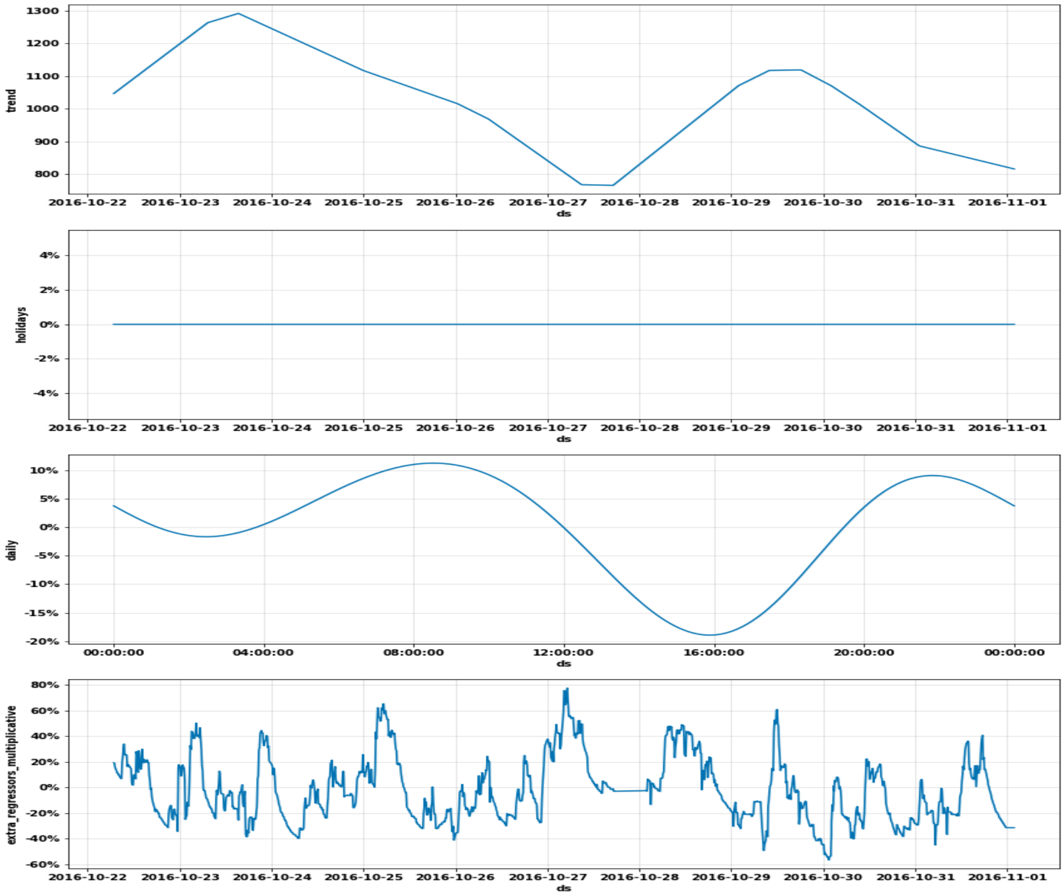


Fig. 5.19. Prophet seasonality plot for multivariate time series using Swarm-CO2 dataset

Furthermore, the *add_changepoints_to_plot* function was used to add the red lines; the vertical dashed lines are the changepoints that Prophet identified when there was change in the trend of the CO_2 levels. The bold red line is the trend after the removal of all the seasonality. The full Prophet seasonality component plots from the predicted test set are shown in Figure 5.19 and they were created with Fourier transforms.

In Figure 5.19, the seasonality component plots comprises of the trend, the holidays, the daily and extra_regressor_multiplicative plots. Here it is assumed that the absence of occupants in the room during certain hours of the day contributed to the trends in the recorded data. This trend plot is what the data is showing once you subtract every other component. The trend of this seasonality plot is not wavy but rather, it has a steep decline at the middle of the month and a gradual upward and downward trend at the beginning and at the end of the observations respectively. The holiday component plot shows the effects of all the holidays included in the model. Here there was no obvious holiday event that happened during the data recording. Unexpectedly, the daily component plot shows a daily repeated occurrence of lowest levels of CO_2 at 16.00pm in the afternoon, while the peak CO_2 concentration levels are early in the morning at 8.00am and 8.00pm in the evening. There is also gradual rise of the CO_2 concentration levels after 16.00pm and 4.00am.

Finally, the $extra\_regressor\_multiplicative$ plot shows the effect of the weather on the indoor CO_2 levels. That is the indoor CO_2 levels with respect to other variables has a cyclic and wavy pattern where CO_2 is increased in the middle of the day and goes down later in the night and a lot of that variability is accounted for by the weather.

5.4.4 Summary

Two time series models; LSTM and Prophet were used to predict indoor CO_2 concentration levels using univariate and multivariate time series approach with Swarm-CO2 dataset. Its result summary is shown in Table 5.3. The first column represents the models' name and its corresponding time-step, the second column represents the univariate time series results in terms of their performance indices and the third column represents the multivariate time series results in terms of their performance indices. The performance results of all the models are represented by the RMSE and the MSE values for the two models.

In general, the performance results shown in Table 5.3 indicates LSTM outperformed the Prophet model in both the univariate and multivariate indoor CO_2 prediction for both 12 and 24 time steps in the MSE and RMSE. Furthermore, all the 12 time steps prediction outperformed its counterpart in the 24 time-step prediction for both the MSE and RMSE. In order to answer the research question (RQ1) and the hypothesis stated in Section 4.1 and Section 5.1.2 respectively, the LSTM model will be used with the Post-Doc dataset so as to ensure that the research questions are answered.

5.5 Confirmatory Study Analysis

The preliminary study discussed in the previous section has shown that the best performed model among the two time series models used to forecast indoor CO_2 concentration levels is LSTM model. These results can be seen in Table 5.3. Therefore, this confirmatory study with Post-Doc dataset will be used to test and confirm the same LSTM model and its parameters as it was used for the Swarm-CO2 dataset. Testing the same model with the same parameters with another dataset recorded from another room will give credence to

Model name	Univariate Value	Multivariate Value
LSTM 12-time-step	MSE: 0.0025 RMSE: 0.0341	MSE: 0.0144 RMSE: 0.0859
LSTM 24 time-step	MSE: 0.0024 RMSE: 0.0347	MSE: 0.0147 RMSE: 0.0852
Prophet 12 time-step	MSE: 197.61 RMSE: 253.89	MSE: 393.85 RMSE: 514.28
Prophet 24 time-step	MSE: 199.23 RMSE: 255.85	MSE: 394.58 RMSE: 513.87

Table 5.3: Swarm-CO2 dataset Results for LSTM and Prophet model

the proposed model for advance prediction of indoor CO_2 concentration levels.

5.5.1 Stationarity Test for Post-Doc Dataset

In order to test for stationarity with the Post-Doc dataset, the *coint_jojansen* function was used. After the test, the eigenvalues returned from the Johansen test are all less than zero, which shows that the series (all the features) are stationary with its eigen values less than 0. The results can be seen in Section Q.

5.5.2 Univariate Indoor CO_2 Prediction with LSTM Model for Post-Doc Dataset

LSTM model was also fitted with the Post-Doc dataset using the same univariate approach stated on Section 5.4.2.2. The performance plots of the actual versus the predicted CO_2 concentration levels for the 24 and 12 time steps are shown in Figure 5.20 and Figure 5.21 respectively. The two plots are similar to each other. The predicted (blue line) and the actual (red line) CO_2 levels have the same downward and upward pattern. Though there are some places where the gaps between the actual CO_2 and the predicted values are large.

Similarly, the LSTM loss plots for the 12 and 24 time steps shown in Figure 5.22 and Figure 5.23 respectively is an evidence that the LSTM model performed very well with the Post-Doc dataset. The plot lines of the training loss (blue) and the test loss (orange) and intertwined with each other from 2 to 16 epoch running. But with a higher loss values at the beginning of the epoch runs.

The trend pattern with a bold red line shows similar pattern movement towards the end

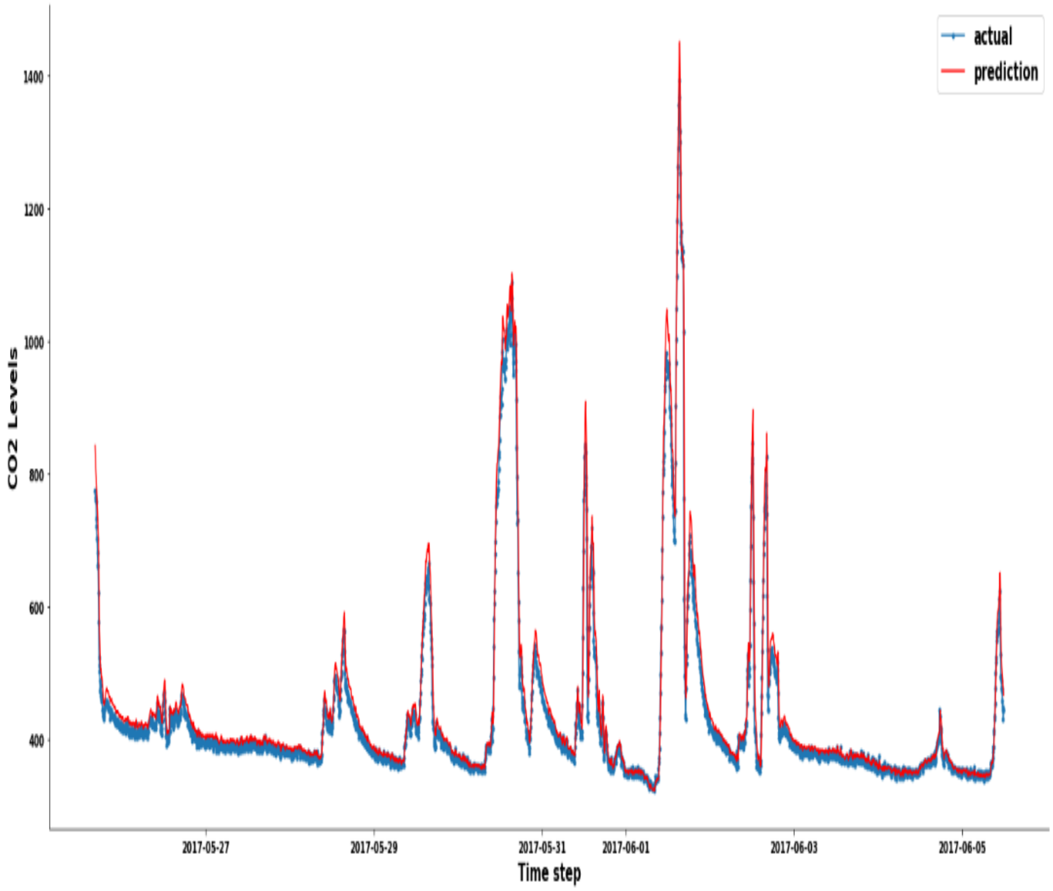


Fig. 5.20. 24 time step univariate actual versus predicted CO_2 levels with LSTM model using Post-Doc dataset

and the beginning of the observation. The performance indices for the Post-Doc dataset are MSE and RMSE obtained for the 12 and 24 time steps are approximately 3.072e-04 and 3.070e-04 and 0.0107 and 0.0104 respectively. The performance indices values are a good indication that the glsprophet model performed well on the Post-Doc dataset.

5.5.3 Multivariate Indoor CO_2 Prediction with LSTM Model for Post-Doc Dataset

Similar to the approach used in Section 5.4.3.2, the LSTM model was used with the Post-Doc dataset in order to confirm its performance of the LSTM model on another dataset. Therefore, the python codes for the multivariate prediction of indoor CO_2 levels where all the variables were considered before the prediction can be seen in Section Q.

The plots shown in Figure 5.24 and Figure 5.25 represents the multivariate CO_2 prediction using Post-Doc for 24 and 12 time steps respectively. In the first-half of the two performance plots, upto one-third of the plots for the multivariate CO_2 prediction, using LSTM made wide predictions between the actual and the predicted values. However, the

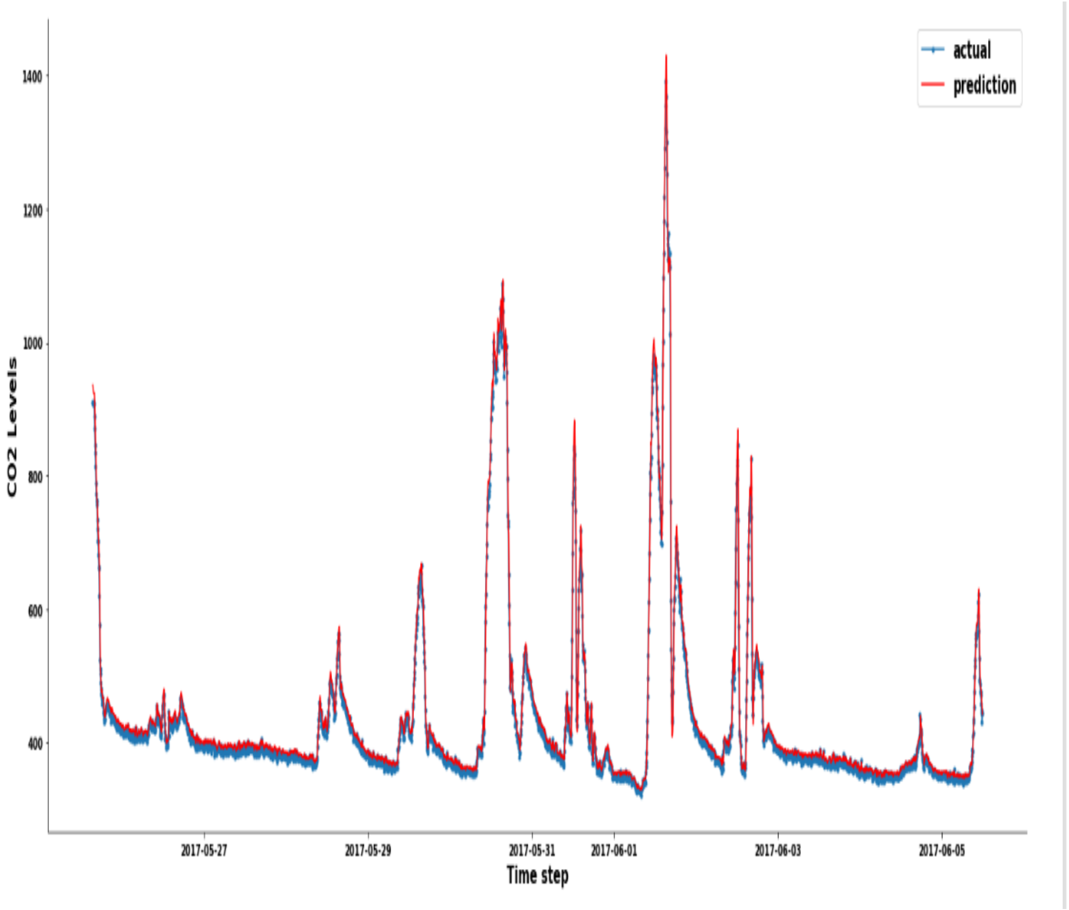


Fig. 5.21. 12 time step univariate actual versus predicted CO_2 levels with LSTM model using Post-Doc dataset

predictions were close to each other in the last half of the plot. The achieved performance indices for the multivariate series are MSE of 0.0089 and 0.0087 respectively while its RMSE are 0.0684 and 0.0682 respectively.

The loss plot for the two multivariate series shown in Figure 5.26 and Figure 5.27 shows wide gap in the achieved result with train set and test set. The loss of the train sets are higher with values above 0.030 after running 20 epochs. The same as the loss plot for the 12 time steps.

5.6 Findings and Discussions

Advance prediction of indoor CO_2 levels has the potential to help room occupants to change the way they live indoors, improve energy efficiency by encouraging demand-driven HVAC supply and also make indoor environments healthy. To the best of the author's knowledge, previous indoor CO_2 studies concentrated on using the CO_2 levels to forecast indoor air quality as such did not deal with future forecast of indoor CO_2 levels.

As a result of this drawback from the previous research, this study concentrated on using

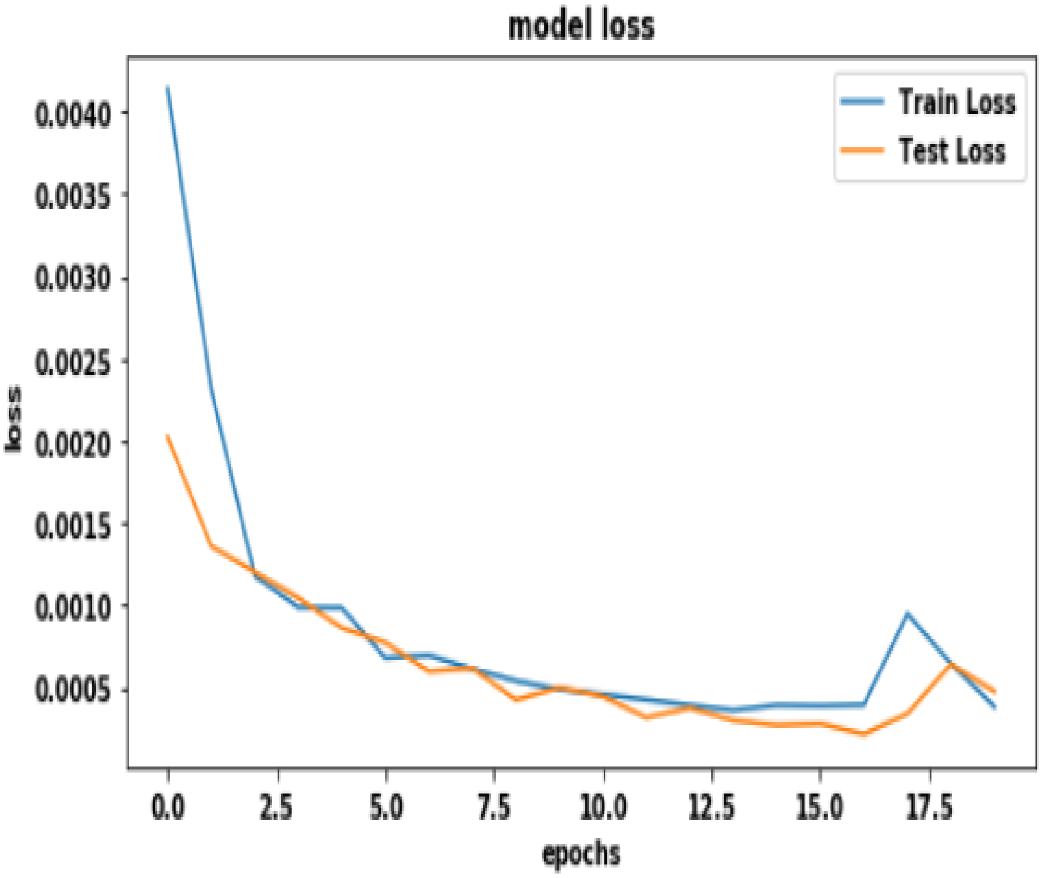


Fig. 5.22. 24 time step univariate actual versus predicted CO_2 levels with LSTM model using Post-Doc dataset

some indoor environmental variables to predict hourly and two-hourly forecast of indoor CO_2 levels with time series ML methods. This study has shown that there are ML models that could accurately predict indoor CO_2 levels against time and against some indoor environmental variables. However, these findings and the ML models used are obtainable in rooms with the environmental settings such 1) rooms with 1-15 seating capacity, 2) room with both constant and no HVAC supply, 3) rooms with no windows or where windows are constantly closed and 4) rooms where doors are constantly open.

The same ML model that outperformed the Swarm-CO2 dataset has proven to be effective with a different (Post-Doc) dataset recorded from another room. For instance, the LSTM model has outperformed the Prophet model with its RMSE and MSE values lower than the Prophet model. The same output was recorded for the 12 and 24 time-step multivariate prediction. The performance comparison can be seen in Table 5.3.

Furthermore, this study has shown that all the series (features) of the two (Swarm-CO2 and Post-Doc) datasets used in this study are stationary by the use of the Johansen and ADF test for stationarity. This is because the eigen values as shown in Section P and Section Q are less than zero. The stationarity properties of the two series helped in the understanding of the time series problem by the ML model used.

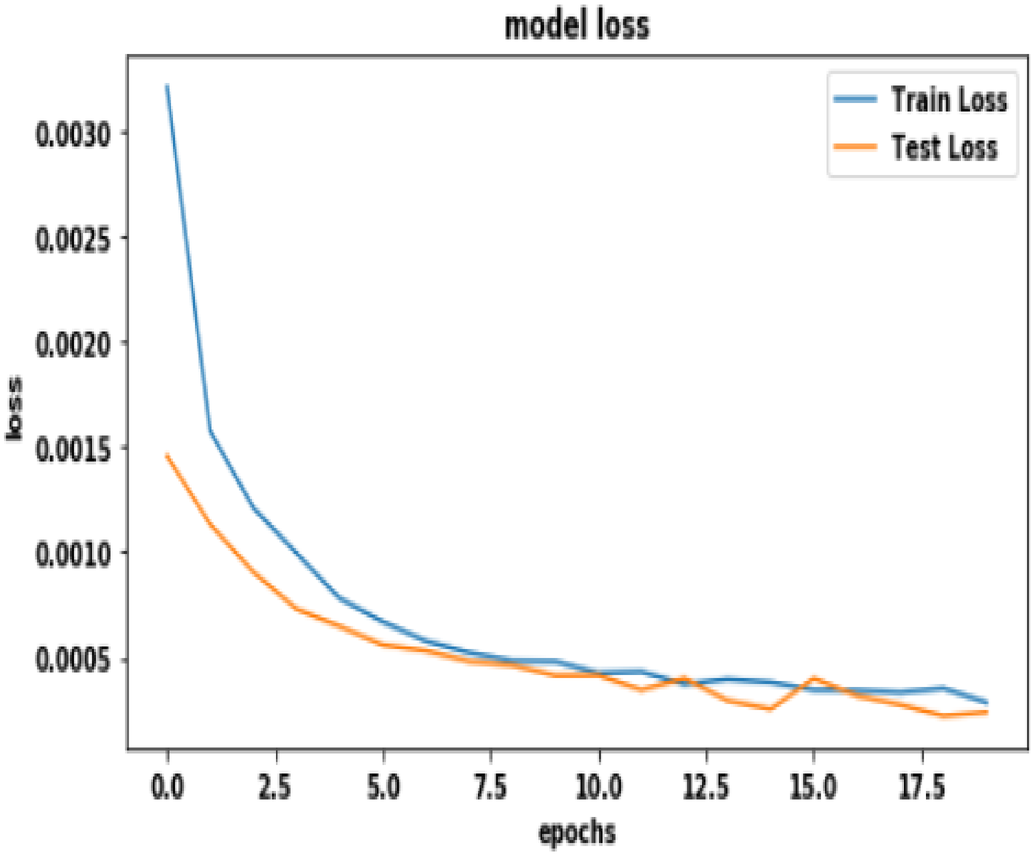


Fig. 5.23. 12 time step univariate actual versus predicted CO_2 levels with LSTM model using Post-Doc dataset

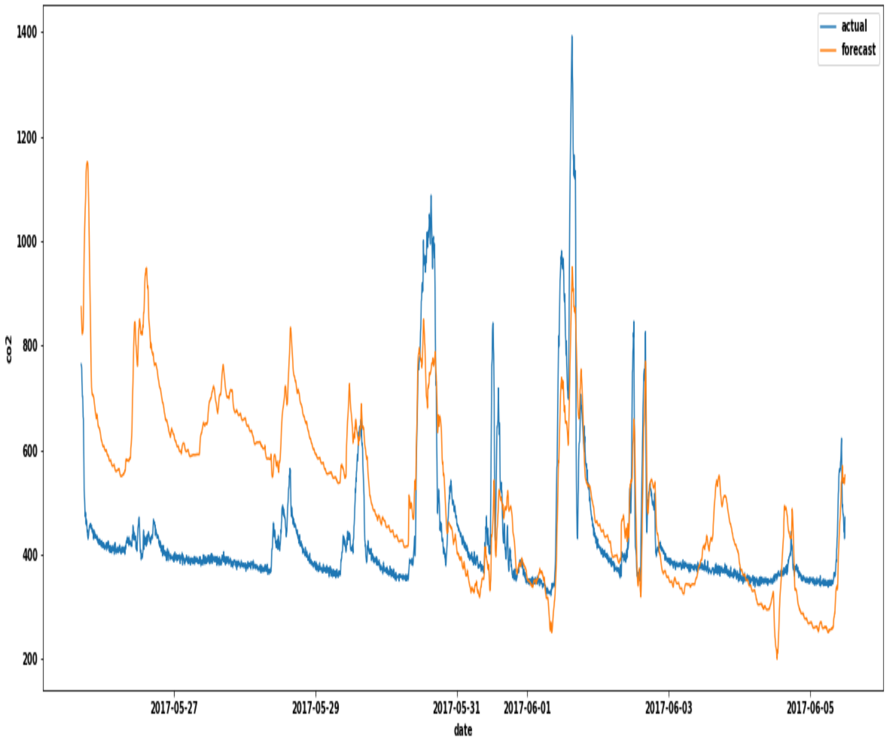


Fig. 5.24. 24 time step multivariate actual versus predicted CO_2 with LSTM model using Post-Doc dataset

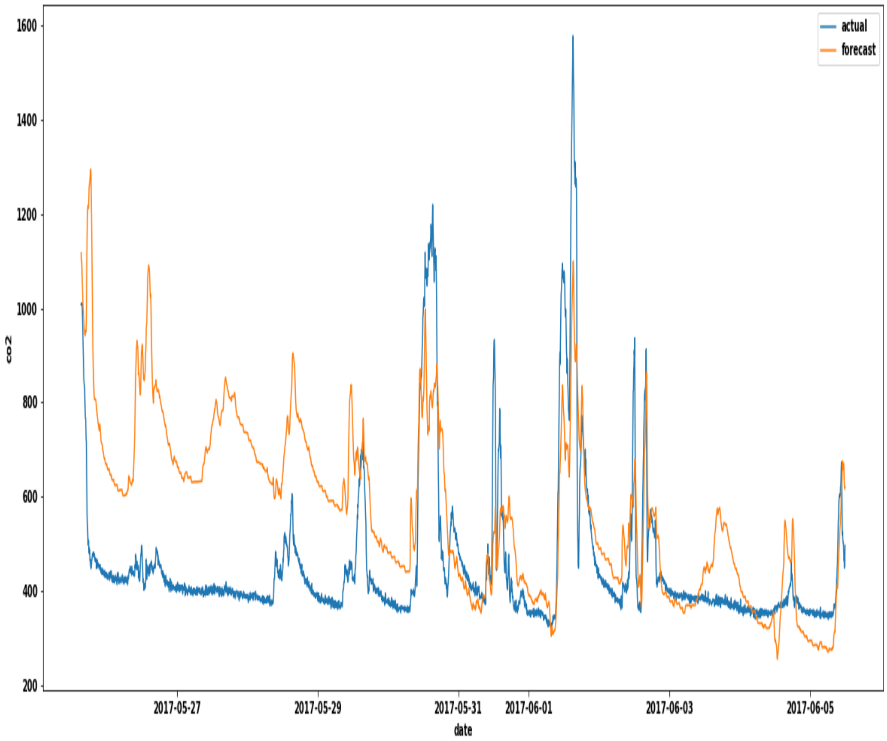


Fig. 5.25. 12 time step multivariate actual versus predicted CO_2 with LSTM model using Post-Doc dataset

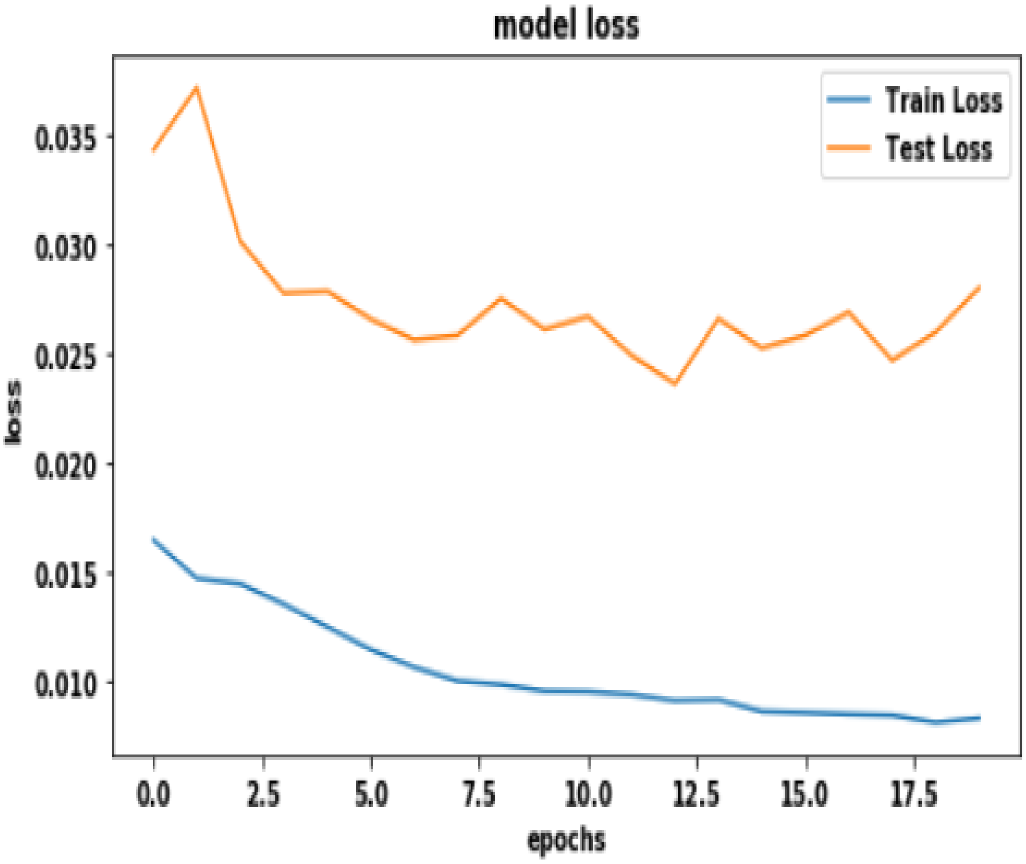


Fig. 5.26. 24 time step Loss plot for multivariate CO_2 prediction with LSTM model using Post-Doc dataset

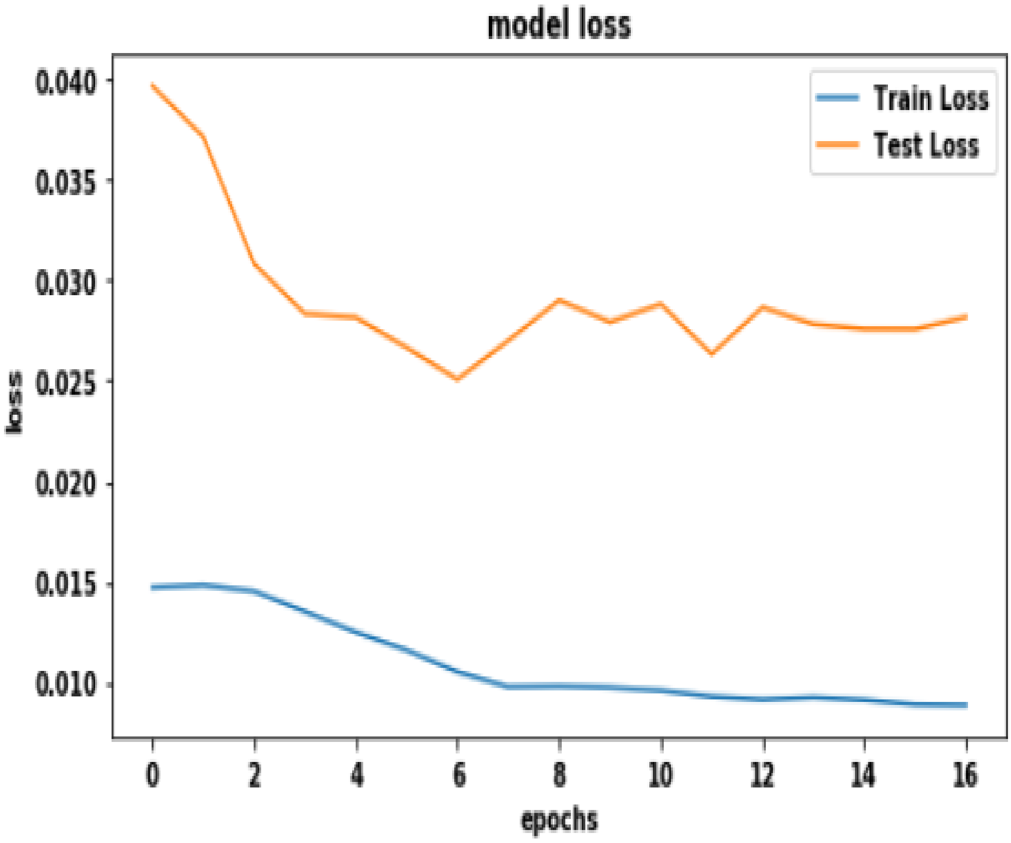


Fig. 5.27. 12 time step Loss plot for multivariate CO_2 prediction with LSTM model using Post-Doc dataset

6

Study Two: Estimating and Interpreting Room Occupancy

Chapter 5 showed that it is possible to forecast about three time-steps of indoor CO_2 concentration levels of a room using LSTM time series model. The hourly forecast was proposed in order to ensure that room occupants could have enough time to take preventive measures before any future indoor CO_2 concentration rise in a room. The study one was considered in this research because it consist of various ways of ensuring that our indoor environments are healthy especially as it relates to the kind of air that is been breathe.

Another way of ensuring that our indoor environment is healthy is been able to know the exact number of people in the room in order to improve energy efficiency, room management and reduce carbon emission. Therefore, this chapter will explore the possibility of estimating room occupancy using an ensemble ML method that is interpretable. The ensemble model will be interpreted using explainable ML for better understanding by non experts.

Before the commencement of the room occupancy study, semi-structured interviews were conducted with some interested participants that were either managing or using the rooms where the room occupancy study were conducted. The interview findings and how it helped in answering the research questions for the room occupancy estimation study will be discussed in this chapter. The study plan followed in order to answer the research question for this second study is shown in Figure 6.1.

6.1 Study Overview

In spite of all various machine learning methods used in estimating number of occupants in a room, there are three main problem that still remains completely or partly unresolved. These problems are as follows:

- No previous ML models used for research on room occupancy has been interpreted and evaluated (completely unresolved).

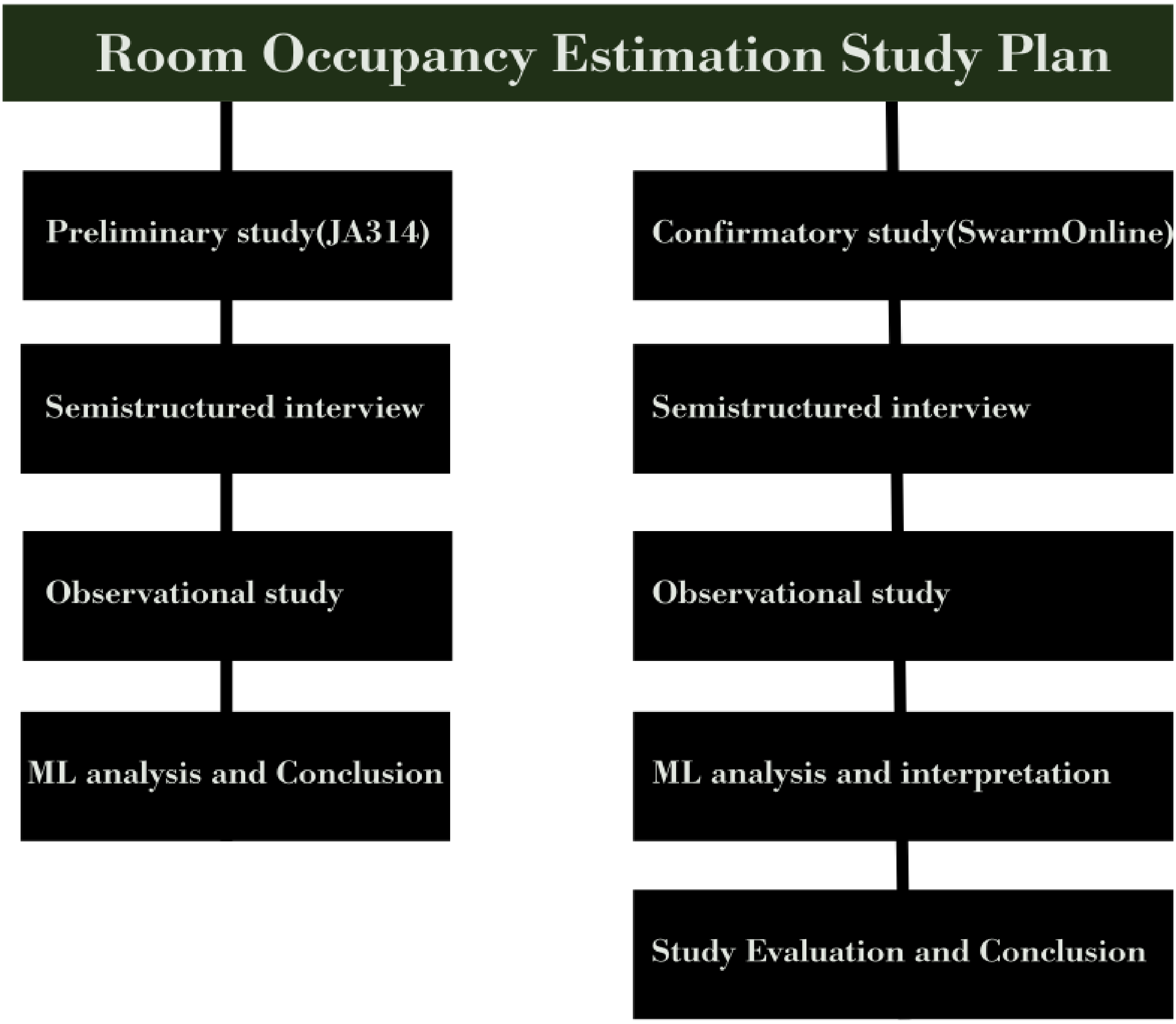

Room Occupancy Estimation Study Plan
Preliminary study(JA314)
Semistructured interview
Observational study
ML analysis and Conclusion
Confirmatory study(SwarmOnline)
Semistructured interview
Observational study
ML analysis and interpretation
Study Evaluation and Conclusion

Fig. 6.1. The study design plan for different phases of room occupancy estimation study

- Most of the previous studies uses a ML model that are room dependent (partly unresolved). This means that as the room for the experiment changes, the ML models or its parameters could change.

Therefore, as a means of acknowledging the uncertainty and limitations inherent during the presentation of the ML pipelines and the need to answer the second research question which states "**Which ML method can be used to estimate room occupancy which can also be interpreted with an interpretable ML.**" This research will use bootstrap prediction interval approach which provides range of values by approximation for its prediction rather than a specific assumption about sampling distribution Stine [1985].

Applying prediction interval in this analysis will help to bridge the gap in the literature where some room occupancy ML models used in the past were room dependent.

Furthermore, prediction interval application is used in regression problem in order to mitigate some limitations that often causes prediction uncertainties. One of such ML limitations that causes prediction uncertainties could be some imaginary features (such as actual HVAC readings, lighting information etc.) that could affect the target variable but were not captured during the data recording. Most times data scientists fail to notice these limitations when they give single exact value as their predicted result thereby giving an often false impression that the result is a source of truth. Additionally, interpreting the ensemble model used in this research will help to uncover the mechanism behind the model's prediction thereby, reducing the confusion often brought about by some uninterpreted ML models. Hence, understanding the ML models' prediction is an additional tool in deciding the worthiness of a model.

As the study for room occupancy estimation using ML methods continue to gain momentum, getting the right kind of interpretable ML model that can accurately predict the number of people in the room is still an academic challenge. In view of this, this study was able to generate a range of estimates in form of prediction intervals using (GB) ensemble by Sklearn, so as to show some degree of uncertainty in the capabilities of the results predicted. Thereby, bridging the gap in the literature and earning the trust of non-data scientists where often the room occupancy results predicted are far from reality.

6.1.1 Aims

There are two main aims of this room occupancy research. They are 1) to find out the best interpretable ML methods that can accurately estimate room occupancy irrespective of the room involve, and 2) to know which of the interpretable ML method could best explain the results predicted by the ML method for better understanding among non-experts

The objective is to use Netatmo sensor to record some indoor environmental variables and to analyse the dataset with GB ensemble ML method in order to find out how many occupants are in a room.

6.1.2 Hypothesis

The hypothesis formulated for this room occupancy study is **If some environmental indoor variables such as temperature, humidity, CO_2, pressure and noise are known, then room occupancy levels could be accurately estimated in intervals.**

6.2 Interview Study: Occupancy Study Interview and Analysis

6.2.1 Aims

This interview was to serve as a means of capturing individual stakeholders' interests and concerns before the room occupancy study commences. The objective of this interview was to find out directly from the people responsible for managing those rooms if there are any need that could be solved with the room occupancy study. The interview was in form of semi-structured interview and it created an opportunity to find out if there are challenges the participants face in managing their individual rooms when the number of occupants in those rooms are not known.

This investigation served as a stakeholders' requirement gathering that formed the main reason for conducting this room occupancy study. This interview data helped in formulation of research questions, hypothesis and final conclusion about the room occupancy research study. Therefore, the main aim of this qualitative study for room occupancy estimation was to find out what answers that could assist in the quantitative analysis being conducted. These aims are in the following:

- To supplement and demonstrate the benefits of knowing room occupancy levels of an indoor environment.
- To ascertain how, the staff responsible for managing these venues reflect on their management experience, whether they are aware of any challenge or benefits and mention them.
- To identify ways of determining the number of people in the room so that supply plant (HVAC) could provide varying inputs (chilled water, heat and ventilation) efficiently and reasonably based on the number of occupants without compromising the legislation.
- To identify ways of knowing the concentrations of CO_2 in advance and its benefits and connections to the supply plant.
- Identify any technique (in form of ML interpretability) that could aid in determining the number of occupants in the room so that lighting and other electrical equipment could be used only when they are needed.
- To explore ways in which real-time indoor CO_2 and occupancy levels could be communicated to the supply plant's automation system constantly and the format by which it could be communicated.

6.2.2 Methods

6.2.2.1 Study Participants

As part of the room occupancy estimation that comprises of quantitative and qualitative components, the findings of the qualitative part in this section is presented. The findings were obtained from semi-structured interviews with 5 people of the same workforce but from two different companies. They were asked to explain their estate (work) management experience with regards to people's indoor comfort, their reasons for being interested in management of room occupancy and indoor CO_2 levels, the advantages (if any) they derive from such activities and most importantly the format they would want the results to be presented to them. The experts who participated are described in Table 6.1. The participants were 1) estate management staff from the University of Strathclyde, 2) room booking staff of the University of Strathclyde and 3) three staff from SwarnOnline Ltd Glasgow. The interviews took place on the 22/02/2018, 22/03/2018 and 6/11/2019 respectively. Please see Table 4 for interview appointment schedules.

ID	Profession	Sex
P1	Head of building services	M
P2	Timetabling manager	M
P3	Financial company manager	M
P4	Financial company staff	F
P5	Financial company staff	M

Table 6.1: Table showing interview participants and their domain expert

6.2.2.2 Research Design

The reason behind the qualitative part of this study was to give comprehensive understanding into how estate managers and company staff recognize the effects and advantages of managing indoor CO_2 and room occupancy levels, whereas the quantitative part after the dataset analysis with ML ensemble learning provides the degree to which the estimation of room occupancy is possible. The qualitative interviews provides what such (quantitative) results signify to them, how not knowing the indoor CO_2 and room occupancy levels in advance have adversely affected the energy efficiency of the rooms they manage.

6.2.2.3 Data Collection and Research Ethics

Before the commencement of this first qualitative study, there was an ethics application with number **Application ID: 951 and 676** that was approved by the department and the interview information sheet is shown in Section C. The qualitative interviews were carried out in the participants' offices at the University of Strathclyde and SwarmOnline limited before the quantitative component of this study. The semi-structured interview questions contains six closed-ended and seven open-ended questions described in Section B. The closed-ended questions asked the respondents whether they are staff or student, what they do in the organization where they work, if they are interested in the study and if they already have a mechanism in place that monitors indoor CO_2 and room occupancy levels in those halls. While the open-ended questions asked the respondents to elaborate more on the closed-ended questions they have already answered.

The semi-structured interview were conducted few weeks before the observational study. It was a face-to-face interview that lasted between 5 to 9 minutes with each of the participants. All the questions that the participants were asked are shown in Section B. Their answers were recorded using windows voice recorder and were later transferred to the university H-drive for security purposes. The topic guide for the interview was developed by the researcher and some minor adjustments were made by the chief investigator and the second supervisor. All the interviews were done and transcribed. The data body resulted into 8 pages of interview transcript for all the interviews.

All the participants (n=5) gave their full consent before been interviewed and their voices were recorded with their consents too. Before starting the interview, they were sensitized on the context of the study and what their data would be used for. The background interview questions were sent to them via email prior to the commencement of the interview. The interview data were anonymized in order to remove identifying information.

6.2.3 Data Analysis

The interviews were analysed using content analysis method Erlingsson and Brysiewicz [2017], which has also been discussed in Section 7.3.1.2. The process of using this method was started by first transcribing the interview data via oTranscribe software (free HTML App) Bentley [2019]. After which the obtained transcribed data and the voice recording were managed in line with the principle of University of Strathclyde's general data protection regulation (GDPR). Before the analysis begun, the transcribed data was anonymised for statistical purposes.

In order to provide solution to some of the research questions shown in Section 4.1, the interested participants report about how they manage their rooms was examined. The challenges involved in managing the rooms. The benefits they hope to get if they knew the real-time CO_2 and occupancy levels in advance. How they explain the difficulty or justify the efforts they encounter as space or estate managers and staff in their work experience.

By using thematic content analysis in analysing the transcribed interview data, it basically means that the transcribed data were first read several times in order to understand in general what the participants were talking about. After this initial step, the text were divided into meaningful components. Further condensation of the meaningful unit was done while retaining the core meaning. Next step was to label the finally condensed units with codes and then grouping the codes into categories so as to discover important information and pattern contained therein. In this analysis, categories were the highest level of abstraction. The coding scheme for this interview analysis is shown in Section D. Additionally, hybrid coding that was suggested by Fereday and Muir-Cochrane [2006], such as inductive and deductive style were also combined during the qualitative data analysis. The main categories for the code system are (i) benefits of the study and (ii) what format to develop the study results.

6.2.4 Findings

There were 7 key categories that emerged after using the thematic content analysis discussed in Section 7.3.1.2. These categories as shown in Section D were as follows:

- Management of Electrical and Mechanical services.
- Indoor Comfort and Well-being.
- Energy Savings.
- Reduce Carbon Emission.
- Utilization.
- Productivity.

- Information Usage and Format.

In this interview, the benefits of this study mentioned by the respondent to some degree covers all the advantages of the room occupancy estimation and advance CO_2 prediction earlier mentioned in this research. The answers from the study participants contributed to the formulation of some part of the thesis' questions which are **1) What time in advance will the indoor CO_2 levels be predicted so that the room occupants can have enough time to act in advance.** 2) **What is the best format for displaying the result of the room occupancy levels to the interested stakeholders.**

The results of the qualitative analysis shows that all the five participants indeed are conscious of the positive impact of knowing the room occupancy levels while one of the interviewee (P2) is not interested in knowing the CO_2 levels but rather only the room occupancy levels. A number of conclusions can be drawn from the answers provided by the participants as follows:

- Participants are aware of more than one benefits of knowing the exact number of indoor occupants in real-time. Though there are some benefits they became aware of when asked to explain their work experience.
- Participants can relate the benefits of the study to their individual work experience and they all spontaneously mention more than one benefits.

In view of the answers of the participants, the room occupancy study was thereafter designed in line with their responses. Adequate attention was also paid in the choice of the ML methods used in the room occupancy study. The previous methods in the literature were also studied for guidance in the design of this research study.

There are predominant benefits contained in the interview data. They are broken down into subcategories as follows:

- **Management of Electrical and Mechanical services:** The experts (P1 and P3) interviewed acknowledged that knowing the occupancy and the CO_2 levels will help to vary the input supply to the exact number of people in the space. These inputs are heating, lighting and cooling.

> *P1: "Yes the main reason will be that, we will like to be able to ensure that the plants is providing a reasonable environment to the occupants and therefore we are meeting the air obligations. But It is also useful to know, plants can be varied at its input in terms of heat input, chilled water input, ventilation input and plants can be matched to the number of people actually in a space. Then that is the more efficient way of operating plant. At the moment where we have rooms where there is no control, the plant will run at its designed capacity which might be for 50 people. It doesn't matter if there are 5 people in there or 50 people in there. The plant will run at the same rate. Therefore, a lot of the time we are over providing for the same space."*

> *P1: "I think that if we knew how many people in there, we could do that either by counting people or you could do by how much CO2 is being extracted from the room, that would allow us to carry out either for new buildings. We would then ask for that to be put in so that the plant and equipment would already provide us with the necessary information to vary the input."*

> *P3: "I also am interested in knowing whether or not there are any technique to it ... to save energy by perhaps increasing or decreasing the heating or the lighting in the room depending on the occupancy levels."*

- **Indoor Comfort and Well-being** P1, P3 and P5 agrees that knowing the exact number of people in the space and the CO_2 levels will guide the room managers in managing the occupants' comfort and well-being without compromising the thermal comfort law. An extract from the interview on this is below:

> *P1: "But my responsibility or our responsibility in the team I have is that we need to make sure that the spaces that people occupy are in the best we can make it fit for purpose and are complying with the legislation. Thermal comfort as well sort of.. Yes is to make sure that the people are studying and the people are working in our environment has to be of a reasonable quality".*

> *P3: "Yeah I think perhaps to add is our safety element as well, for instance high levels of CO2 there is actually those of safety threat which is something that we should know... about those threat.... and so on and so forth."*

> *P5: "I was only going to add to understand the rooms maximum"*

- **Energy Savings** P1 and P3 agrees that when the exact number of the people in the room are known, the plant supply will be decreased or increased based on that figure and no longer. This will save energy and reduce cost of the plant's operation.

> *P1: "And where there is existing equipment available, we don't have that facility at the moment, we could find a way of reasonably costed way of retrofitting that, ultimately, that would also save us money and would hopefully make the spaces for the occupants. It won't detract from that."*

> *P3: "...to save energy by perhaps increasing or decreasing the heating or the lighting "*

- **Reduce Carbon Emission** P1 agrees that operating the plant supply only on the need basis will reduce carbon emission in the environment.

> *P1: "... That ultimately would also save us money, save us carbon emissions..."*

- **Utilization** The participants (P2 and P4) believed that knowing the exact number of people in the hall will help them to know when the rooms were used or unused and when the rooms' capacity are under utilized.

  > *P2:"Am interested in curbing the differences between the actual usage and planned usage of central teaching rooms"*

  > *P4: "if it means us reducing the number of people in our room just to help efficiency then that's me."*

  > *P2: "Well as mentioned earlier I want to be able to compare it against the planned usage of the rooms and our process for not finding departments of the rooms that were booked or where an appropriately sized rooms have been booked and students which are attending.. this should lead to more efficient use of the space that we have. It should also lead to better information when new rooms are developed for central teaching spaces like size and usage."*

- **Productivity** The participants P3 and P4 believed that knowing the exact number of the people and the CO_2 information could improve productivity.

  > *P4: "... I think the study comes with a good result to help us to understand how to keep productivity high ... based on ... I think that will help us maintain that level of productivity ... so that we will know what to do."*

  > *P3: "I do agree with P4's comment."*

- **Information Usage and Format** All the participants were interested in using the results of the analysis to plan the efficient utilization of the rooms where they manage. For instance, P1, P2, P3, P4 and P5 expects the following formats of the results.

  > *P1: "Again if it was live data, we would want that to be taken straight into our automation systems. So the automation systems will read the sensor that says.. the $CO-2$ levels is currently it will automatically make changes to the ventilation or the heating or the cooling plant that is serving that space, to either vary up or down depending on that $CO-2$ levels. Ultimately, that would be the preferred option. At a lower level, unless responsibly will be to be told, to be given regular report or electronic report that says."*

  > *P2: "Just in a table format of time and number of students"*

6.3 Methods

The previous section presented the interview findings from the first interview conducted in this research. The findings were used to design the study conducted in this research. This section will now discuss various methods used for the study analysis. Therefore, in order to answer the research questions for room occupancy estimation, a Netatmo sensor was used to record some environmental variables such as temperature, humidity, CO_2, pressure and noise from the two offices; JA314 Figure 6.5 and Swarm Figure 6.6.

Dataset description used for the two room occupancy estimation study is explained in Section 6.4.1 and Section 6.5.1. The two datasets recorded will be analysed using ensemble trees methods known as gradient boosting (GB); an additive regression model that fits every iterated base learner drawn randomly by a least square methodsFriedman [2002]. This gradient boosting will be computed using a method known as prediction interval. Specifically, gradient boosting from Sklearn was used. The reasons for choosing gradient boosting are as follows:

- **It is effective when working with structured data.**
- **It is fast to train especially when the dataset are not too large.**
- **It does require lots of tuning during the training of the dataset with the model.**

ML model explainers (SHAP and LIME) discussed in Section 6.3.6.3 were applied so that they could aid in interpreting the GB ML models used for the estimation of room occupancy. The machine learning analysis work-flow shown in Figure 6.2 was followed in this research during the data analysis.

Figure 6.2 represents machine learning flowchart used for the estimation of room occupants. The diagram is interpreted as follows:

- The programme is started by loading the dataset from the “csv” file or excel sheet. This process will form a table of columns and rows. The columns and rows represent both the dependent and the independent variables.
- After the table has been formed, the complete dataset will be split into 66% training set and 34% test set. This method will help in the assessment of models’ performance. Splitting of the dataset is not compulsory, however it is good practice to do that.
- Machine learning model will be built for the regression analysis. The built model will be fitted with the training set and validated with the test set.
- The result output will be analysed/interpreted with SHAP and LIME methods for better understanding
- The programme terminated.

6.3.1 Model Testing Method

Correct division of dataset into training and test set is important for the performance of any ML model in order to avoid over training of the dataset and inadequate generalization. To ensure good model validation a cross-validation (K-fold) technique Mitchell [1997] is often used to split the dataset into two subsets; training and test set and the sizes of the resultant subsets are not restricted Reitermanova [2010]. Therefore, to ensure good generalization of the room occupancy model’s performance, this study proposed the use of K-fold cross-validation techniques with simple random sampling proposed by Lohr [2009]. This means that the obtained dataset was divided into 33% test set and 67% training set during ML analysis. The reason for this proposal is because it has proven to show encouraging results when applied on both the real-world and artificial datasets.

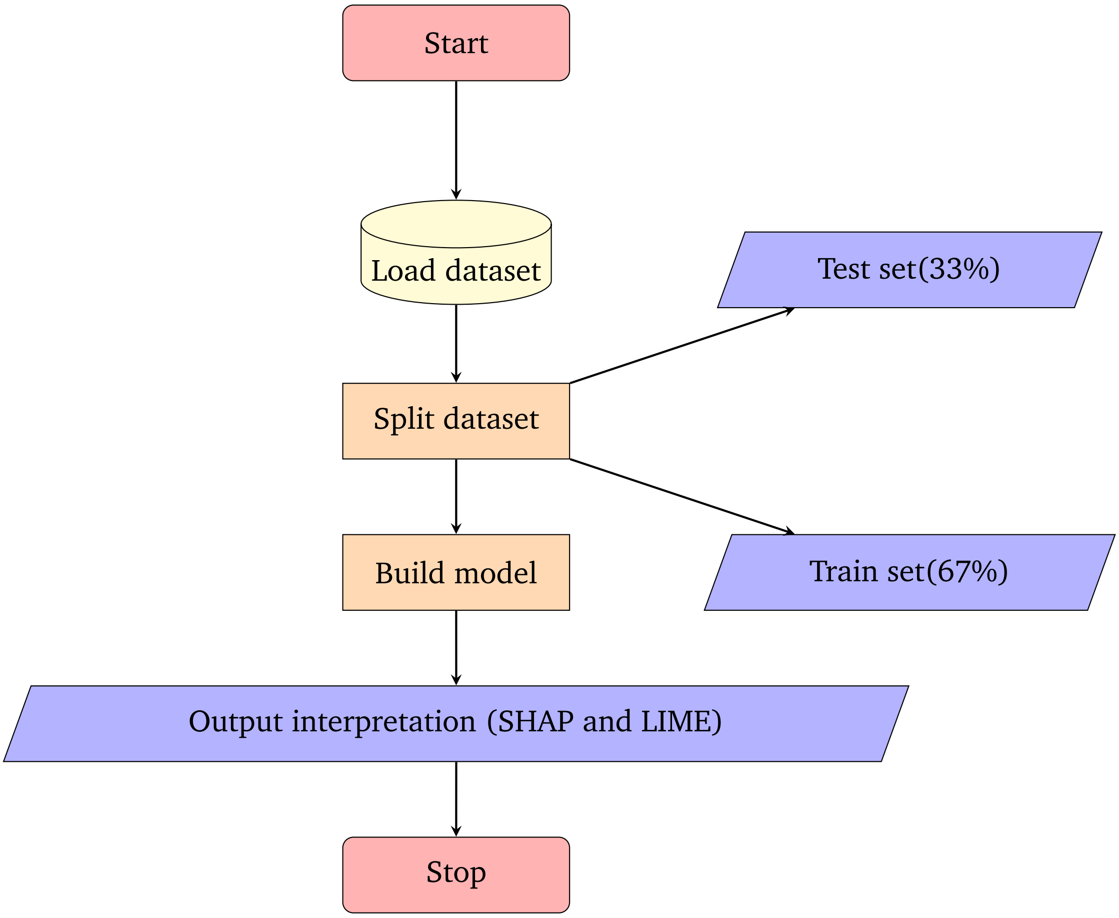


Fig. 6.2. Machine learning analysis workflow for room occupancy estimation

6.3.2 Camera Used for Study Two

During the confirmatory occupancy study, the ground truth occupancy information of the office was needed in order to conduct the room occupancy estimation study. A digital camera known as UniFi G3 video camera Online Repository [b], shown in Figure 6.3 was used to track the number of people in the office every 15 minutes. The unifi camera uses infrared for its surveillance management system.

The following characteristics are possessed by the camera:

- **Frequency of Recording and Software Management:** The video camera captures the live images of the room occupants at a frequency of one minute. It also uses both mobile and web application for easy camera management and live streaming. The Unifi software could be used on windows operating system, iOS and android based applications.

- **HD Image Quality:** The UniFi camera uses a superior effective focal length (EFL) of 3.6mm f/1.8 lens. Its EFL feature enables it to provide 1080p high definition (HD) resolution image at 30 frames per second (FPS). This characteristic enables for easy identification of the persons' images for manual counting.

- **Mounting Option:** The UniFi video camera is flexible when it comes to mounting and

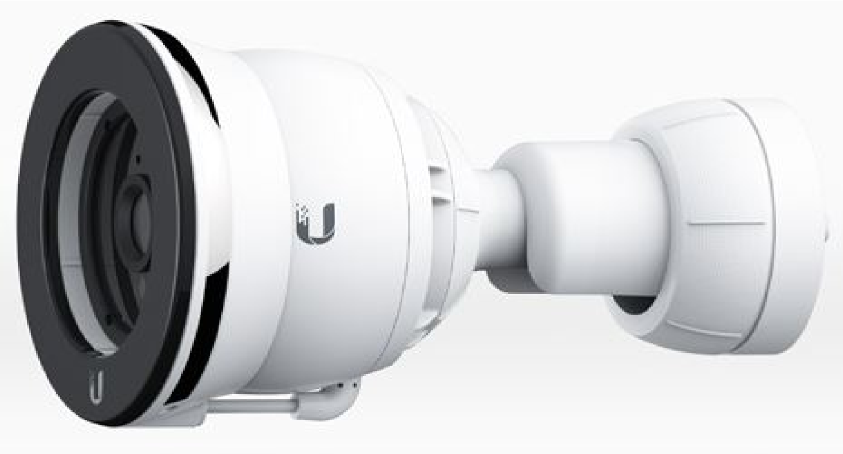


Fig. 6.3. Unifi Digital Camera Used for the Confirmatory Occupancy Study

installation. It could be mounted on the ceiling, wall or pole because of its flexible handle. The video camera was mounted on the wall for this confirmatory occupancy study as it can be seen on the image Figure 6.4.

- **Privacy and Access:** The recorded videos or images are kept privately on the person's local disk. This prevents unauthorized access of the images. The authorized user can access the videos from anywhere with the help of the UniFi hybrid cloud.
- **Infrared Range Extender and Storage:** The UniFi video camera uses infrared (IR) range extender which helps in capturing bright images at night time. The video camera has a storage capacity of about 1TB and could be extended to 5TB.

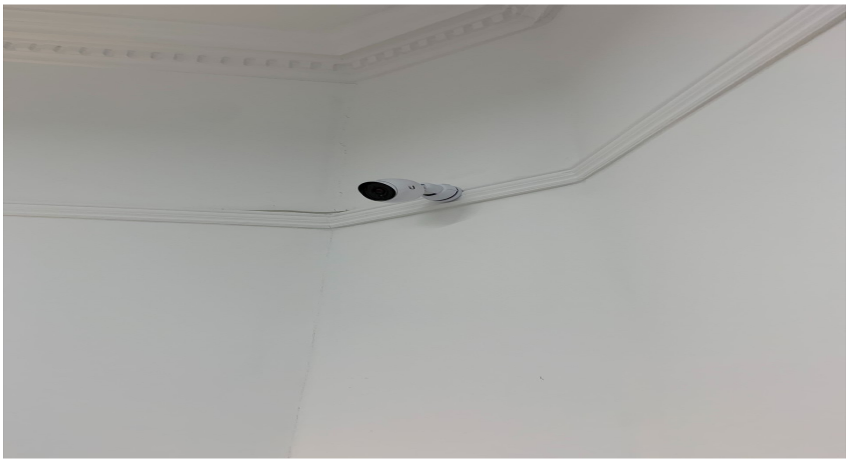


Fig. 6.4. Location of the mounted video camera

6.3.3 Choice of Locations for the Room Occupancy Study

The room occupancy estimation study was conducted using environmental variables in two different offices. The choice for the two offices used for the study was dependent on the following criteria:

- **Availability of the venue:** Because of concerns for occupants' privacy and interest in the project, some of the rooms that the research study would have benefited from were not permitted for use. As such other locations that were both readily available for the study and where the management in charge of the rooms were interested in the study were chosen.

- **Availability of the independent variables:** One major determinant for the locations used for the second part of the study (room occupancy estimation) was the possibility of obtaining some third-party independent variables' information for the study. These variables are ground occupancy information and HVAC information. The reason was because not all locations can assess the number of people that enters or leaves the room. Therefore, before considering any location to use for the room occupancy study, obtaining these information must be guaranteed.

- **Variability of the levels of independent variable:** Another determinant for an impartial analytic result of this research study is a location where its independent variable levels obtained for the study are frequently changing. One example is the study that estimates the number of room occupancy based on the CO_2 levels. If the number of occupants in the room are constant at every time of the day, the result of the analysis might be biased.

- **Capacity of the room:** Another criteria for the choice of study location for the room occupancy research study is the number of seating capacity in the room. For the purpose of improving on the previous studies done by other data scientist in the area of room occupancy estimation, using a venue that has more than ten seating capacity is very important.

- **Constant power and internet reception:** Due to the fact that the CO_2 sensor used for the study needs steady constant power and internet supply. Any location of choice must have these two requirements before the commencement of the study.

- **Security of the venue:** To avoid theft of the study materials or interruption of the study which often leads to missing data, the security of the study locations must be guaranteed.

The John Anderson (JA) venue Figure 6.5 has a seating capacity of between 100 to 140. The JA room has only one door for entrance and exit of the occupants and additional one door for only emergency exit. The JA room was used for first room occupancy study due to high variability of the number of occupants in the room. The same type of room setting with physical attributes was also used for the confirmatory room occupancy study.

The picture in Figure 6.6 is the development room at SwarmOnline company. The room has between 10 to 15 seating capacity. This room has two doors as it can be seen in Figure 6.7. Both of the doors are used for entrance and exit by the employees of the company.

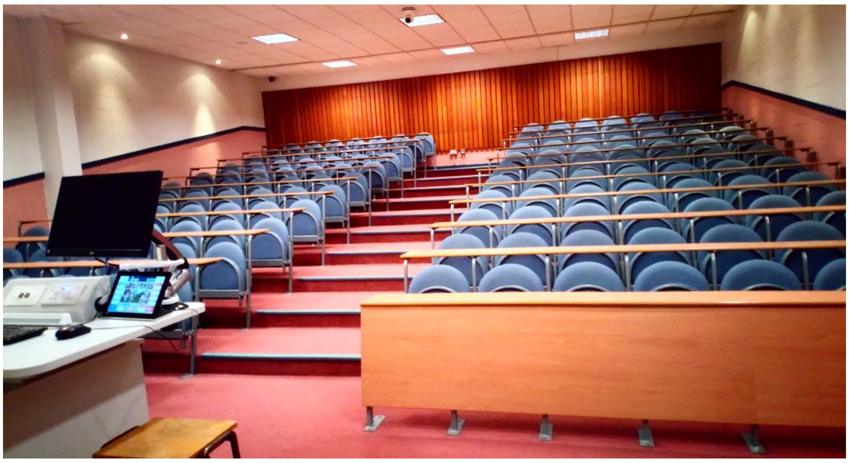


Fig. 6.5. John Anderson Venue for Preliminary Room Occupancy Study

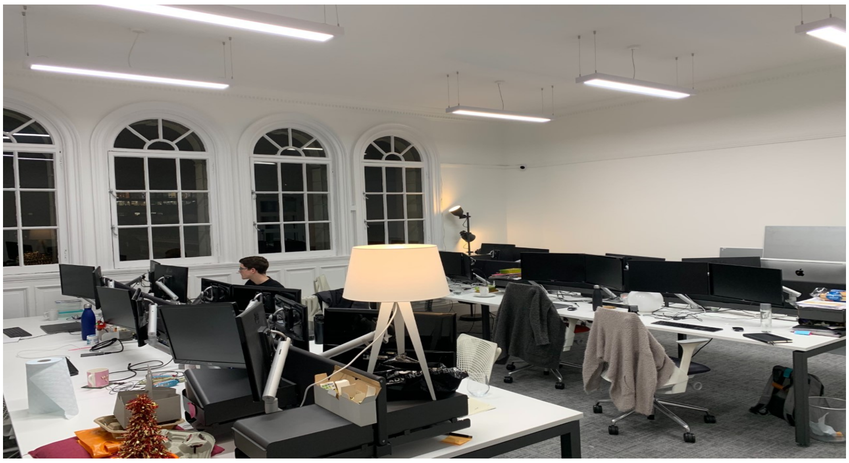


Fig. 6.6. SwarmOnline Venue for Confirmatory Occupancy Study and Testing phase

6.3.4 Ground Truth Occupancy Information for Room Occupancy Estimation Study

During the data recording of the two (JA314 and Swarm) room occupancy dataset, two different approaches were used to obtain the ground truth occupancy information of the two rooms where the room occupancy study were conducted. These approaches are stated below:

1. **JA314 Ground Truth Occupancy:** A digital camera was installed in the JA314 room. The camera automatically takes the picture of the room every 1 hour and stores the still pictures in a database for manual counting. Each of the pictures were manually examined, counted and recorded in order to obtain the number of occupants in the room at a frequency of 1 hour.

2. **Swarm Ground Truth Occupancy:** In the SwarmOnline office, a digital camera known as UniFi G3 video camera Online Repository [b], shown in Figure 6.3 was used to obtain the ground truth occupancy information of the office. The camera takes the pictures of people in the office every 15 minute and stores them in the

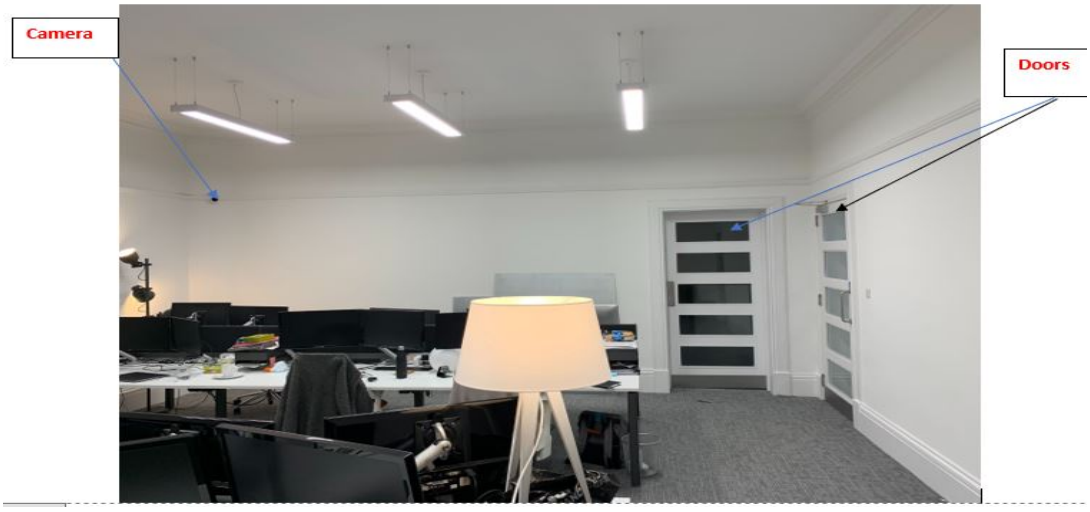


Fig. 6.7. The location of the doors and camera at the SwarmOnline Venue

owner's personal online account. Only authorized access is allowed to access the pictures for manual counting and recording. The picture of the mounted camera is shown in Figure 6.4.

Other parameters such as fresh air inlet in the room Ebadat et al. [2013] were ignored because there was already a mechanical air handling unit that supplies air at a constant interval to the rooms. This air handling unit is the HVAC which was either on (1) or off (0) during the day as it was programmed to work. Additional information of window and door events were recorded as closed (0) throughout the day and the internal doors are open (1) during the work days.

6.3.5 Ensemble Learning for Regression

The two most important problems in ML regression analysis are choosing the right model (model selection) and model improvement Rao and Tibshirani [1997]. During model selection, the analyst aims is to find the best model that can improve the accuracy of the prediction using a dataset. This fact is often daunting especially when the dataset is not linear. In some situations, using a single regression model on a dataset does not give the desired optimal solution during the ML analysis.

Many scientific data researchers Ali and Pazzani [1996], Dietterich [2000a], have observed that applying more than one model is better than any single classifier or regressor model. This process is known as ensemble method. Hence, it is believed that the use of ensemble method in ML analysis has so many advantages that results in the improvement of ML results. Such advantages are 1) It averages out biases, 2) It reduces high variance and 3) The method prevents over-fitting.

By definition, ensemble method also known as ensemble learning is a process that uses a set of different models obtained independently as a result of learning. These set of models is put together in some way to get the final model Strehl and Ghosh [2002]. There are two phases in ensemble method. The first is the learning phase, which is responsible for training some base estimators and the second phase is prediction phase, which combines the predictions of all regressors for new instances Zhao et al. [2009]. Therefore, to help achieve the objective of this research, Gradient Boosting ensemble model was used to estimate room occupancy in this research.

Simple illustration of the ensemble method is, for instance, if we have a regression problem with a likely boundless input space X (x_1, x_2, x_3,...), the aim therefore, will be to generate a function $\hat{f}$: X→ R that is roughly $\mathfrak{f}$ an unknown true function, where the standard of the approximation is given by the MSE mean square error (mse), otherwise known as generalization error. This can be represented in the following Equation (6.1). Though, the number of error being reduced depends on the type of domain being used Ali and Pazzani [1996]. This is to say that if the model domain is good, the quantity of error reduction would be high and vice versa.

$$mse(\hat{f}) = E[(\hat{f} - \mathfrak{f})^2] \tag{6.1}$$

Where $\hat{f}$ known as the predictor or model is derived by applying a learner on a limited set of n samples with the shape {(x_1, $\mathfrak{f}$(x_1)),.....,(x_n, $\mathfrak{f}$(x_n))}. Taking into account that it is impossible to ascertain the true error of a model $\hat{f}$ Mendes-Moreira et al. [2012]. The error can be estimated as in Equation (6.2) on a different n_{test} (test set) of dataset.

$$mse(\hat{f}) \approx \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} [(\hat{f}(x_i) - f(x_i)]^2 \tag{6.2}$$

Furthermore, there are three main steps of ensemble learning process Roli et al. [2001]. These steps are 1) the ensemble generation; where different sets of models are generated as shown in Equation (6.3), 2) the second step of the ensemble learning process is the pruning step represented in Equation (6.4). In this step, some sets of models generated earlier are eliminated before combination Zhao et al. [2009] and 3) the last stage is the integration stage shown in Equation (6.5) where all the base models are combined via a strategy in order to form new ensemble prediction for new cases Roli et al. [2001].

$$F_0 = \hat{f}_i, i = 1, 2, ..., K_0 \tag{6.3}$$

$$F \subseteq F_0 \tag{6.4}$$

$$\hat{f}_F(x) = \sum_{i=1}^{K} [h_i(x) * \hat{f}_i(x)] \tag{6.5}$$

The two major groups of ensemble methods popularly used in ML today are bagging and boosting methods Opitz and Maclin [1999]. However, only boosting method will be discussed in this research because it was used in this research study to estimate room occupancy.

6.3.5.1 Boosting method

Two well-known techniques for constructing ensemble methods (section 6.3.5) are (bagging) bootstrap aggregation by Breiman, L. and boosting by Freund Y. These two methods are used to improve the performance of any learning set Dietterich [2000b]. Empirically, experiments conducted in Freund and Schapire [1995], Freund et al. [1996] showed that boosting algorithm can be used to solve prediction problems by improving the accuracy of a weak learner and that the method can be applied to any real-world data set. Apart from bagging and cross-validation methods, boosting is also another form used in manipulating a training data set with an advantage over bagging. The advantage is that its iterative approach tends to reduce prediction error Dietterich [1997]. Applying boosting to a real-world problem will continue to prove how beneficial its practicability is in terms of problem solving, hence the reason for its use in this research.

6.3.5.2 Processes and Approaches to ensemble method

The concept of changing environment in ML was addressed by Rooney et al. [2004] where the authors proposed that for any regressor, be it single or ensemble, which is intended to be used for a real data application, that the regressor should be furnished with tools that will help it to adapt to the changes in the system. This is because the inherent changes in the environment constitutes to some of the main obstacles in its environments. The author's proposal could be adduced to the necessary strategies that should be put in place for building the regressor. These strategies depends on the type of model used which forms part of the steps and processes in ensemble methods.

Therefore, three different steps in ensemble learning method as shown in Figure 6.8 are 1) ensemble generation; which generates a set of models including redundant models, 2) ensemble pruning; which tries to eliminate some of the redundant models generated earlier and 3) ensemble integration; at this stage, base models are combined using a predefined strategy, and the strategy is used to obtain a prediction Roli et al. [2001].

The set of models generated in the first step of the ensemble process is represented in Equation (6.6). These set of models could be generated from the same initial algorithm applied to the dataset or with the combination of another algorithm. The methods to be applied differs in terms of application, which means that a general method could be applied to any kind of algorithm or that another method could also be considered because it is algorithm dependent Dietterich [1997]. However, in this research a general technique was applied and they are explained as follows:

- **Manipulating the input features:** In this analysis, random subspace was used so that for each base learning algorithm a subset of samples will generate diversity and transformation of test samples before regression. This means that, different independent variables were combined with the target variable (CO_2) for better regression outcome.

- **Injecting randomness into the learning method:** In this analysis, different algorithms were applied to the whole training samples while using different beginning weights to make the resulting regressor different.

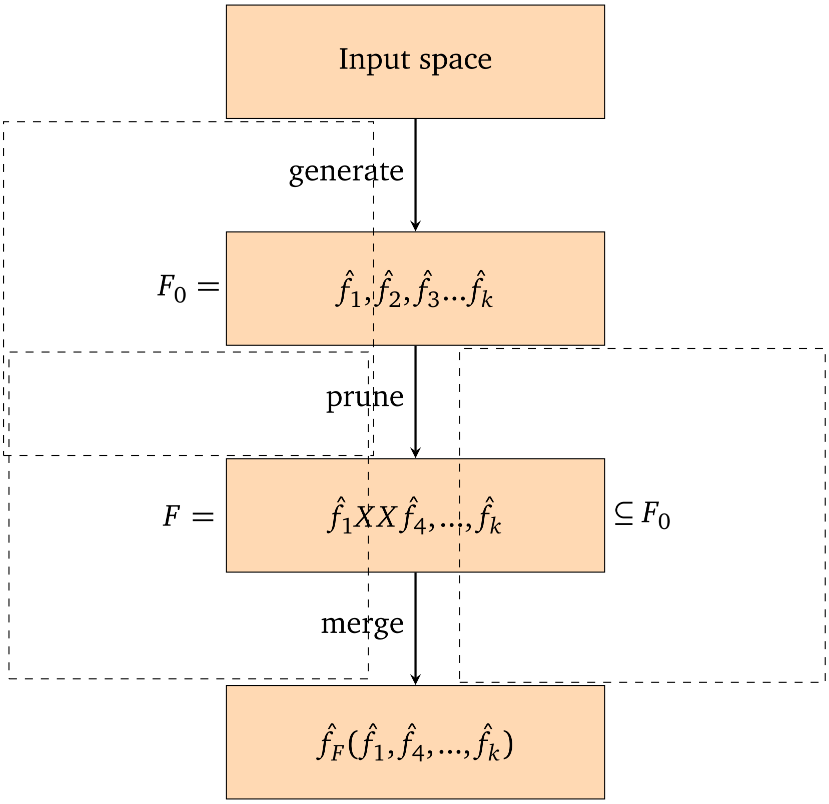

Input space
generate
$F_0 =$
$\hat{f}_1,\hat{f}_2,\hat{f}_3...\hat{f}_k$
prune
$F =$
$\hat{f}_1 XX \hat{f}_4,..,\hat{f}_k$
$\subseteq F_0$
merge
$\hat{f}_F(\hat{f}_1,\hat{f}_4,...,\hat{f}_k)$

Fig. 6.8. General model for ensemble regression architecture used in this research

- **Manipulating the training and the test samples:** As stated by Dietterich [1997], in this analysis the training and test samples were split with an arbitrary number using an inbuilt function from the machine learning toolkit (SciKit) Pedregosa et al. [2011] learn known as “train_test_split” for more efficient regression samples Friedman and Popescu [2008].

$$\mathfrak{F}_0 = \{\hat{f}_i, i = 1,..,K_0\} \tag{6.6}$$

In order to improve generalization and prediction accuracy, reducing the base regressors done at an intermediate phase as ensemble pruning have been shown by researchers to help in improving the accuracy of the ensemble and reducing the ensemble size Zhao et al. [2009]. Reducing the ensemble size is being done because previous researches have shown that some ensemble generated before combination have adverse effect on the accuracy of the regression and should be removed during the ensemble pruning process. Pruning these ensemble can help to prevent the problem arising as a result of multicollinearity (an event that makes a predictor variable predictable form others in multiple regression) and reduce operational cost Perrone and Cooper [1992]. However, identifying these bad regressors is not an easy task, because some of these ensemble contain large number of regressors and needs large computational overhead and memory space Zhao et al. [2009].

Thus, to select subensemble given the original ensemble size N, a space of 2^N-1 non-empty subset of M will be searched, which is exaustive and unfeasible Hernández-Lobato et al. [2006]. Hernández-Lobato, D. et al. proposed a greedy algorithm that selects subensemble that needs lower memory space, shorter response time and performs better than the complete ensemble. A pseudocode described by the author can be seen in Listing 8.

In Listing 8, the three parameters needed in ensemble generation stage are the dataset ($\mathfrak{L}$), the learning algorithm ($\mathfrak{B}$) and the number of models to be generated φ_0 (100 usually used). Due to the instability of the learning algorithm generated, it was obvious that the ensemble generated will take advantage of the complementariness of the algorithm, in order to improve the accuracy and diversity of methods used. This generation phase is represented in lines 1-4 of Listing 8. The aim of the pruning phase is to reduce the total number of low base estimators that does not have a positive impact on the learning algorithm, and the steps to achieve that is represented from lines 5-21 of Listing 8. Lines 5-9 represents the computation of the covariance matrix for φ_0 predictors. Lines 10-20 are steps for obtaining a subset of φ predictors with the use of forward search technique. During each iteration, regressor is selected from the list such that when added, the training error is brought to a minimum level. Note however, that all the steps in Listing 8 are not applied in the regression problems because regression problem uses the generalization error. The line 21 represents the integration function $\mathfrak{F}$

$$\mathfrak{F}\subseteq\mathfrak{F} \tag{6.7}$$

The last phase in ensemble learning process is the ensemble integration, which is represented in Listing 6.1 in form of a pseudocode. Integration is made for each prediction, and the average is obtained.

Require: $\mathfrak{F}$, the ensemble Require: x, the example to make a prediction about

```
K := Size($\mathfrak{F}$)
return $\frac{1}{K}\sum_{i=1}^{K}\hat{f}(x)$
```

Listing 6.1: Simple avearage of ensemble integration

These stages and processes of ensemble method mentioned were applied in this thesis in order to contribute towards achieving better results with the models.

6.3.6 Interpretable Machine Learning

The main benefits of using computers for computing ML models in data analytic are for ease and speedy computational processes, which facilitates the improvement of products and researches, especially when there are thousands of data involved. But one important disadvantage is that the computers often do not explain their results. As a result of that limitation, the ML models that its internal mechanisms were not explained are called the black box models Molnar [2019]. The concept of interpreting the ML black box model is known as interpretable ML, which helps to make the behaviour and prediction results of the ML systems easy for humans to understand.

Interpretability of ML does not have any mathematical explanation. According to Miller [2019], interpretability is the extent to which a human can regularly predict the ML model's result. If the interpretability of a ML model is high, it becomes easier for non-experts to understand why some certain decisions or predictions are made. This makes the model better in terms of interpretability than others. Doshi-Velez and Kim [2017] stated that what necessitated the need for model interpretability is because of incompleteness in the formalization of a problem. As such, the importance of interpretability is summarized as follows:

- **Scientific understanding:** Because of lack of knowledge of the models, humans are often compelled to ask questions so that they can gain knowledge of how certain results are obtained.

- **Safety:** Uncertainty and high cost often associated with the testing of complex systems or tasks makes the results of the models questionable.

- **Ethics:** In order to protect certain result decisions against bias which often leads to discrimination, humans wants to be treated fairly when certain decisions are made on their behalf.

- **Mismatched objectives:**There is likelihood that the agent's algorithm will focus on the objectives it is only interested in rather than a holistic objective.

- **Multi-objective tradeoffs:** Full specification of the dynamics of the trade-offs behind the objectives of the ML systems are necessary for competition with one another.

However, interpretability of a ML model depends on the level of impact the results of the model would have in real world Molnar [2019]. For example, interpretability is not needed when 1) the problem has been sufficiently researched and studied with enough practical experience and 2) the results of the interpretability will make human to manipulate the system. Example credit or loan systems used by the bank could be manipulated to favour those it does not intend to favour.

6.3.6.1 Taxonomy of Interpretability Evaluation

There exists some performance indices used for the evaluation of ML results that are considered appropriate. The same condition is applied to ML interpretability. Therefore the standard benchmark for evaluation of ML interpretability should demonstrate generalizability. Hence, Molnar classified the taxonomy of ML interpretability as either post hoc or intrinsic. In post hoc interpretability, models are applied after the data has been trained with a ML model. Whereas in intrinsic interpretability, the models are considered self-interpretable because of their simple nature or how simple its structure is presented. Example of intrinsic interpretability is decision trees Freitas [2014] or sparse linear models. Below are the types of evaluation metrics for ML interpretability as discussed by Dosh-Velez and Kim:

- **Application-based evaluation:** In order to ensure human computer interaction and user satisfaction in an application developed, evaluations are conducted based on human experiment. The experiments are conducted with respect to the task the application is expected to perform.

- **Human-based metrics:** In order to test the notion of general public about the quality of evaluation, experiments are conducted with lay humans that maintain the core of the target application.

- **Functionally-based evaluation:** In this case, no human experiment is conducted rather predefined functions and definitions are used to evaluate the quality of the interpretability of the model. This approach saves time and cost for the researcher.

The results obtained from the interpretation methods differ in different ways such as 1) through summary statistics for each feature which includes feature importance depending on the model used, 2) through visualization of the feature summary because some methods makes meaning when its curve are visualized rather than table. Example of such is partial dependence curves, 3) model internals such as weights of the linear models, 4) through models that return old data points or the newly created one. Example is image data, and 5) intrinsically interpretable models that is done by looking at the summary statistics of its features.

6.3.6.2 Scopes of Interpretability

According to Molnar, algorithm of models that predicts the result in ML analysis can be evaluated either in terms of transparency or interpretability. These assumptions brought the following scopes of interpretability:

- **Algorithm transparency:** Algorithm transparency depends on how the algorithm learns a model from the data. It is also about been able to explain the kind of relationship with the data it can learn. The transparency of an algorithm is only interested in the knowledge of the algorithm and not the data or the model to be learned. For example a least square linear algorithm is easy to understand compared to the deep learning algorithm.

- **Global holistic model interpretability:** If the entire model can be understood at once, the model is said to be interpretable Lipton [1990]. For the explanation of the output of a model globally, the data and the algorithm are essential in understanding how the model makes decisions. The features of the data needs to be known such as its weights, structures and other parameters. Trying to attain the goal of holistic model interpretability is difficult in practise because it usually involves many features which is difficult for average human brain to remember.

- **Global model interpretability on a modular level:** In this case only part of the model is expected to be understood on how it makes prediction unlike the global holistic model interpretability. A single weight/model can be understood or memorize on a modular level.

- **Local interpretability for single prediction:** In this case, individual prediction is considered by looking at the output of a particular input, in order to explain how the model works. Additionally, increasing or decreasing a particular instance locally can be more accurate than global explanation.

- **Local interpretability for a group of prediction:** In this case multiple instances are select as a subset of the whole dataset. The model explanation can be made based on the global model interpretation of the selected instances or as an individual instance. The individual explanation once made can be listed and aggregated for all the selected group.

There are many ML models in regression analysis, some of them are more difficult to interpret or explain either on a modular level or a holistic level. Most times this models are classified as monotonic or non-monotonic. For model to be monotonic, the model ensures that the independent and dependent variables move in the same direction over the entire feature range. This process makes it easier for the model to be interpreted because its relationship is easy to understand. Some of the interpretable ML regression models are linear, decision trees and k-nearest neighbours.

6.3.6.3 Model-agnostic Methods for Interpretability

Differentiating the ML models from the explanations is known as model-agnostic interpretability models. The advantage of model-agnostic over model specific is the ability for the ML developer to use any model, explanation and representation of choice Ribeiro et al. [2016]. It is crucial for developers to explain their models for non-expert to understand. However, most ML models that attain highest accuracy score are complex and examples of such models are ensemble and deep learning networks.

Consequently, various models have been proposed to help developers to interpret complex models, but there is still uncertainty in explaining how these methods are related and why one is preferable over another. Three popular methods for interpreting ML models are 1) SHAP (SHapley Additive exPlanation); a unified framework that assigns values to feature according to level of importance Lundberg and Lee [2017], 2) local interpretable model-agnostic explanations (LIME) Ribeiro et al. [2016]; a local surrogate model that can explain any single predictor, and 3) deep learning important features (DeepLIFT); a method for explaining the output prediction for deep learning neural network Shrikumar et al. [2016, 2017]. All the methods can be used in regression and classification problems.

Lundberg and Lee observed that the SHAP method improved computational performance and showed more constancy with human intuition more than LIME and DeepLIFT methods. LIME tends to feed different data points into the black-box model, then tests out what happens locally. In LIME Ribeiro et al. suggested that in order to provide a trusted explanation of the black-box model to users, inspecting the individual predictions and explanation will be a valuable answer to the problem. LIME also uses textual or visual artefact for the explanation of the model to the user. LIME is mathematically presented in the below equation:

$$\xi(x) = \underset{g \in G}{argmin}\, L(f, g, \pi_x) + \Omega(g) \tag{6.8}$$

In Equation (6.8), G comprises of interpretable models, g is a single model from G such as decision trees, linear models etc.. Note that any g might be difficult to interpret. As a result, $\Omega(g)$ will be the measure of the complexity of the interpretation. Let $\pi_x(z)$ be the proximity measure between an instance. f is the model being explained. Lastly, $L(f, g, \pi_x)$ is a measure of how untrue g is close to f as defined locally by π_x.

Moreover, SHAP values estimation aligns with Shapley regression values Equation (6.9), SHapley sampling values Equation (6.10) and DeepLIFT input influence Equation (6.11) which also connects with LIME.

$$\phi_i = \sum_{S \subseteq F\{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} [f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S)] \tag{6.9}$$

$$h_x(z') = E[f(z)\,|Z_S] \tag{6.10}$$

$$\sum_{i=1}^{n} C_{\Delta_{x_i}\Delta_t} = \Delta_t \tag{6.11}$$

The Equation (6.9) is feature importances for linear ML models taking into account the existence of multicollinearity. What happens in shapley regression values is that it retrains the model on all feature subsets $S \subseteq F$. $f_{S\cup\{i\}}$ is a model trained with the feature present, x_S is the values of the input features. In Equation (6.10), $h_x(z') = Z_S$ is an input mapping with Z_S having input values. Lastly, in Equation (6.11), the summation to delta property is defined by $\Delta_{x_i}\Delta_t$ where Δ_t is known as difference-from-reference. t is some target output neurons.

6.3.6.4 Local Model-agnostic Behaviour

Furthermore, most ML models gives a global interpretation of its behaviour but not a local interpretation. Hence the need for LIME or SHAP. Thinking of accuracy versus interpretability trade-off. Scientists believe that those ML models that are highly interpretable are usually less accurate, whereas those more accurate are less interpretable. Hence the need for ML explainers. They intend to produce interpretations of models as high as its accuracy. By interpretable explanations, it means which features the ML model picks on in order to make predictions. By model-agnostic it means that the explainers (LIME and SHAP) can be applied to any black-box model currently or in the future. Being local means that the model is observation specific and that it gives you explanation of all the features. The following are the steps that LIME uses to explain the ML model:

- Data permutation: LIME takes all the data or predictors and creates fake data from it through data permutation.
- Distance: LIME will calculate the distance between the fake data it created and the true observation.
- Make prediction: LIME will make its own predictions using the ML model with its new fake dataset.
- Feature selection: LIME will select few most informative features from its fake dataset that best describes the result that was predicted by ML model (black-box).
- Fit model: LIME will fit a simple model on the selected few features together with its similarity scores in order to derive new weights and coefficients which will serve as explainers to the complex model on a local scale.

6.4 Preliminary Study Analysis with JA314 Dataset

In this section, the JA314 was analysed with the GB model using a prediction interval approach. After the dataset analysis ML interpretable models known as SHAP and LIME were used to interpret the GB model after which the interpreted results were evaluated using human based approach.

6.4.1 JA314 Dataset Table and Room Information for Room Occupancy Study

The overall description of the JA314, the venue where the observation were recorded and reasons for the choice of venue for the study are enumerated as follows:

1. The first room occupancy study was carried out at Figure 6.5 in order to test the hypothesis mentioned in Section 6.1.2 above. The venue where the first occupancy study took place is called John Anderson lecture hall, Room 314 (JA314).

2. The Figure 6.5 room for the occupancy study was chosen for three main reasons 1) The room requires energy intensive ventilation to make its occupants comfortable. This could possibly cause high energy usage when the room is vacant, 2) The rooms can occupy average number of 100 students daily and 3) The number of occupants frequently varies because the room is usually booked in the frequency of 1 hour lecture slots.

3. The JA314 dataset has 64 observations(rows) with 7 variables (columns) recorded from the room. The frequency of this observation is 1-hourly.

4. The dataset were recorded every from from 09:30am to 16:30pm daily. This does not include weekend days (Saturday and Sunday) and public holidays. The first five rows of the JA314 dataset is shown in Table 6.2.

5. The recorded datasets are 1) indoor environmental data; CO_2, temperature, humidity, noise and pressure measured in parts per million (ppm), ^oC, %, decibel and millibar respectively, 2) HVAC data set and 3) occupancy data.

6. The environmental data was obtained via an environmental sensor shown in Figure 5.5, which is called Netatmo Online Repository [a].

7. The HVAC data information was recorded from an air handling unit (AHU) supply plant. The HVAC system uses a setting logic known as “night-time setback” Brooks et al. [2015]. This means that the control system is relaxed in the evening from 5pm when it is assumed that no one will be in the room. The HVAC data was recorded at the interval of 1 hour with a constant value.

8. The occupancy information of the lecture hall where the study was conducted was obtained through manual counting and totalling of the room occupants in each image by the university estates staff.

9. The images were captured by a digital camera and stored in a database after which the still images are counted and recorded. There were no counting errors because a 2-step counting verification was used during the recording.

10. All the data recorded for JA314 started from 03/12/2018 at 09:30am and ended on 03/22/2018 at 10:30am.

11. Also, the HVAC was turned on from 8:00am to 5:00pm daily, it was assumed that the values are 1 throughout the period of observation of the JA314 dataset.

12. The occupancy data is the dependent (target) variable while the temperature, humidity, noise, pressure and CO_2 are the independent variables.

13. All the dataset were recorded and extracted as a "csv" file. The environmental dataset was obtained from the Netatmo online account, while the HVAC and occupancy dataset were transferred from the estate manager and room booking departments' account. The data were recorded for 10 days.

14. Because HVAC has a constant value of 1, it was dropped among the independent variables.

	time	temp	hum	CO2	noise	press	Occ-314
0	09.30	20.7	41.0	1076.0	40.0	1009.4	29
2	10.30	21.6	41.0.05	1076.0	42.0	1009.9	29
4	11.30	20	43.0	1342.0	40.0	1013.1	41
3	11.00	21.2	42.10	939.0	43.0	1005.0	39
1	10.00	21.3	41.0	986.0	41.0	1009.7	0

Table 6.2: First-five Row of JA314 Dataset

Table 6.2 shows first five rows of the dataset obtained from John Anderson room 314. The column names and its corresponding variables names are temp, hum, CO_2, noise, press and Occ-314 are temperature, humidity, CO_2, noise, pressure and occupancy respectively.

6.4.2 JA314 Dataset Visualization

The same variable used in the CO_2 prediction study was used in the room occupancy study. The reason for choosing these same variable can be seen in Section 5.3.2. In Figure 6.9, the noise, CO_2 and occupancy variables have nearly the same upward and downward pattern. The temperature pattern has the same upward pattern at the beginning of the observation with humidity however, towards the end of the observation, the two observations show opposite pattern. Pressure has step pattern with the lowest value been recorded at the middle of the observation period.

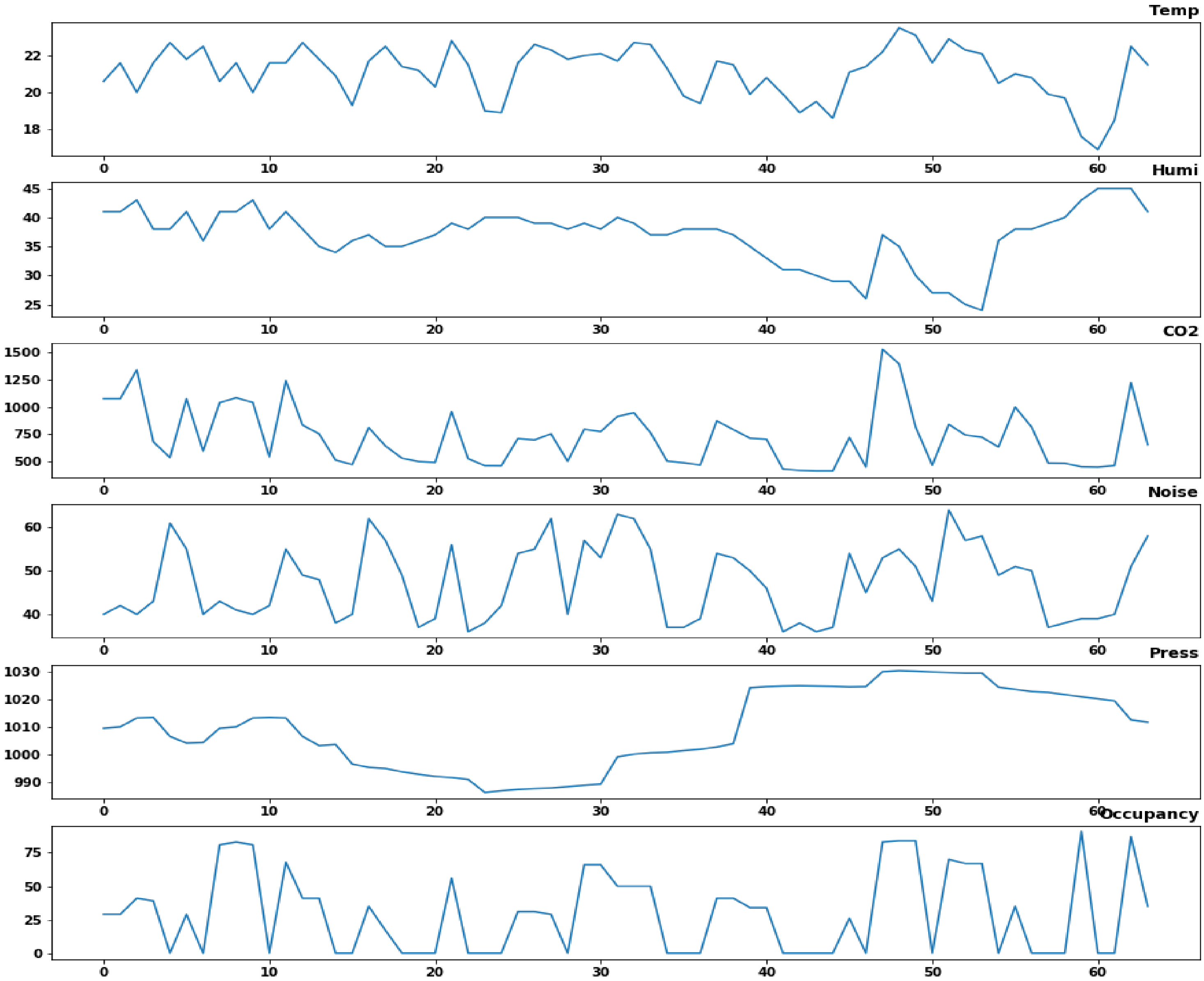


Fig. 6.9. JA314 dataset visualization. This measurement covers for the period of 12 March 2018 to 22nd March 2018.

6.4.3 Gradient Boosting Model Implementation for Prediction Interval

In Scikit-learn, GB ensemble trees is a tool developed for generating uncertainty interval because it is an additive model that uses forward stage-wise manner to optimize differential loss function of its model. Consequently, regression trees are fit at every stage of the given loss function. Certainly, there were some latent variables such as the actual HVAC readings, the lighting event, the door events (the actual time it was opened or closed) etc.. that were not captured in the dataset for room occupancy study because they were unavailable. These latent variables could help to (some extent) accurately estimate the exact number of people in the room. Therefore, this study will show some degree of uncertainty in its prediction by generating the upper, middle and lower bound of the GB model as the Figure 6.10 depicts.

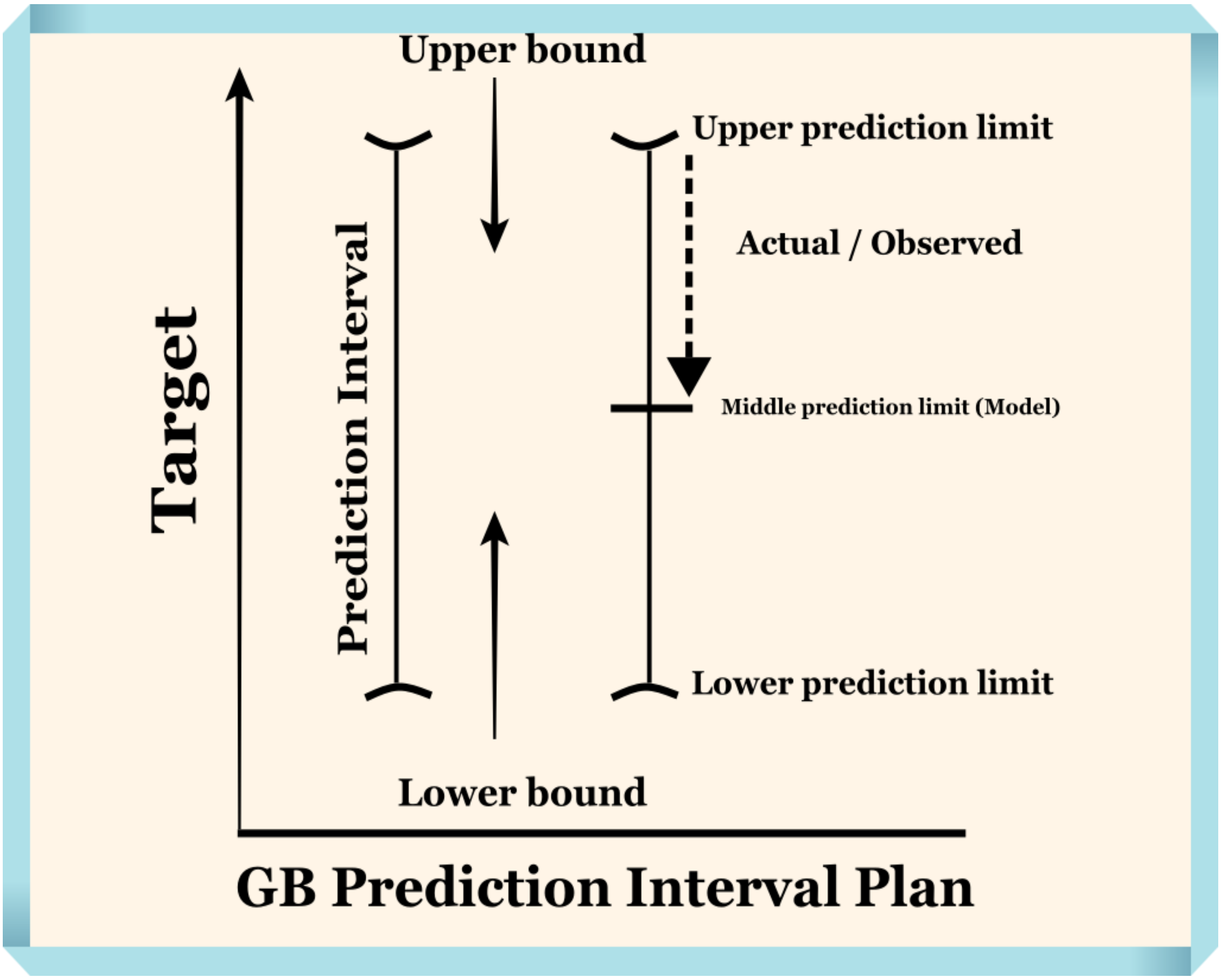


Fig. 6.10. Experimental set-up plan for GB prediction interval

In the Figure 6.10, the “Target” label indicates that the prediction intervals (upper prediction limit, middle prediction limit and the lower prediction limit) generated should be covered by it (Target). The prediction interval should stay within the lower and the upper bound and within he target bound too. Also, the figure indicates that most of the actual/observed values should be close to the middle prediction limit and far from the lower prediction limit (though not definite).

To implement the prediction interval, a $GradientBoostingRegressor$ class from the Sklearn was used. The changes made and strictly followed during the implementation were mostly based on two (alpha and loss) GB class parameters. The following were the steps followed in this research:

- **Lower Prediction:** For lower prediction the $GradientBoostingRegressor(loss = "quantile", alpha = lower\_quantile)$ where lower_quantile represents lower bound with value 0.05.

- **Middle Prediction:** For the middle (the same as GB model default parameters) $GradientBoostingRegressor(loss = "ls", alpha = 0.5)$, where 0.5 alpha value indicates that it predicts the median of the least square (LS) as the default loss option.

- **Upper Prediction:** For the upper prediction $GradientBoostingRegressor(loss = "quantile", alpha = upper\_quantile)$ where upper_quantile represents upper bound with value 0.95.

When quantile is chosen as loss and then the value for alpha is also chosen, the results obtained will correspond to percentile. With lower and upper alpha values specified, an estimated range is generated. The Section S shows the reusable python code that was defined and used for all the room occupancy estimation study in this research. In the defined $GradBoostPredInts$ class, its parameter (BaseEstimator) are defined in its constructor. The constructor takes a number of arguments (*arg) and keyword arguments (**kwargs).

There are also four methods defined within the $GradBoostPredInts$ class. These methods are 1) A fit method; that trains the training set of the features and the target variable. 2) A $predict$ method; that predicts the test set of the target variable using the test features. 3) A $plot\_intervals$ function; that calls the $plot\_intervals$ function defined solely for the plot of the performance of the GB model. The $plot\_intervals$ function code is shown in Section S. 4) A $calculate\_and\_show\_errors$ function that calls the $calc\_error$ function shown in Section S.

During the calculation of the prediction error on the test set of the GB model, the percentage of the time that the actual values falls within the prediction interval range was calculated. Though during optimization, this method could widen the interval, therefore an absolute error was calculated in order to account for that wide gap between the predicted and the actual values. The GB ensemble on the JA314 dataset was implemented in two versions. The first version is with no missing data; that means that all the missing data were deleted before the application of the GB model. The second approach is with missing data.

In view of the prediction interval plan shown in Figure 6.10, three (upper, middle and lower) different gradient boosting regressors were used because of the need to train the dataset separately for accurate optimization of the individual functions. The three separate regressors were already defined in the *GradBoostPredInts* class shown in Section S.

6.4.3.1 Implementing Gradient Boosting Ensemble Learning with JA314 Dataset

In this dataset, there are 64 observation/rows and 7 columns used for the ML analysis. Occupancy data was the dependent variable while temperature, CO_2, noise, humidity and pressure were independent variables. The additional column week day variable was converted from the date-time variable and was called “Day-of-week”. Furthermore, the newly created “Day-of-week” column was converted from categorical variable to dummy variable using OHE. Thereafter, all the variables were normalized into the range between 0 and 1 with *MinMaxScaler* method from Sklearn. The variables were later splitted into training set and test set. The training set contains 66% of the total number of observations in the dataset and 34% of the test set. All the python codes for data preprocessing techniques can be seen in Section R.

After the training of the dataset with GB model prediction, plots were generated with the help of *plot_intervals* method. The generated plots are shown in Figure 6.11 and Figure 6.12 and they are the prediction interval plot of the actual versus predicted values using middle prediction limit. In Figure 6.11, the main area of concern is the light grey area, which shows the area where the three interval results covered.

In the generated prediction interval plot, the orange line represents the upper prediction interval generated with the 0.95 alpha value (95 percentile). The dark violet line is the actual room occupancy values from the observed dataset. The dark blue line is the lower prediction interval results generated with 5 percentile. The green line is the median prediction interval generated with 0.5 alpha (50 percentile) value or by replacing loss parameter with least squares (LS). Similarly, the orange line in Figure 6.12 represents the predicted occupancy values with middle prediction limit, while the blue lines represents the actual occupancy values. In the two plots there was close lines between the actual and the predicted in the beginning and towards the end of the prediction. However, the GB made wide prediction values in some other areas.

In this research the error of the prediction range was quantified in two ways. They are by 1) calculating the percentage of the time that the observed values fall within the range (lower, mid and upper) of values predicted. Thereby, penalizing the GB model for making much wide predictions. This is known as absolute error 2) calculating the RMSE and MSE. The *calc_error* function shown in Section R was used to calculate absolute errors of the lower, upper, interval and middle respectively. The box plot of the absolute errors are shown in Figure 6.13.

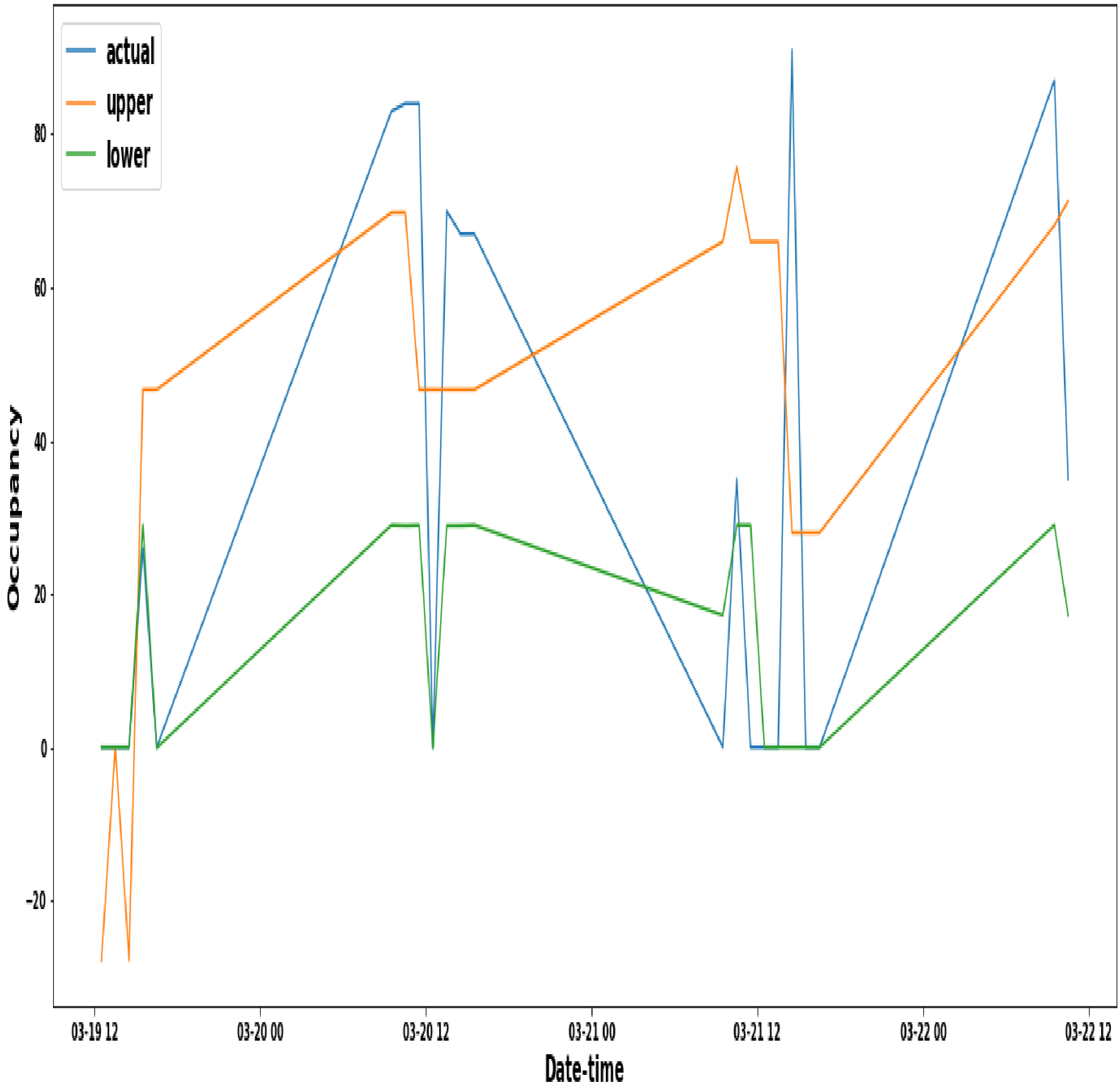


Fig. 6.11. Prediction interval plot with Swarm dataset

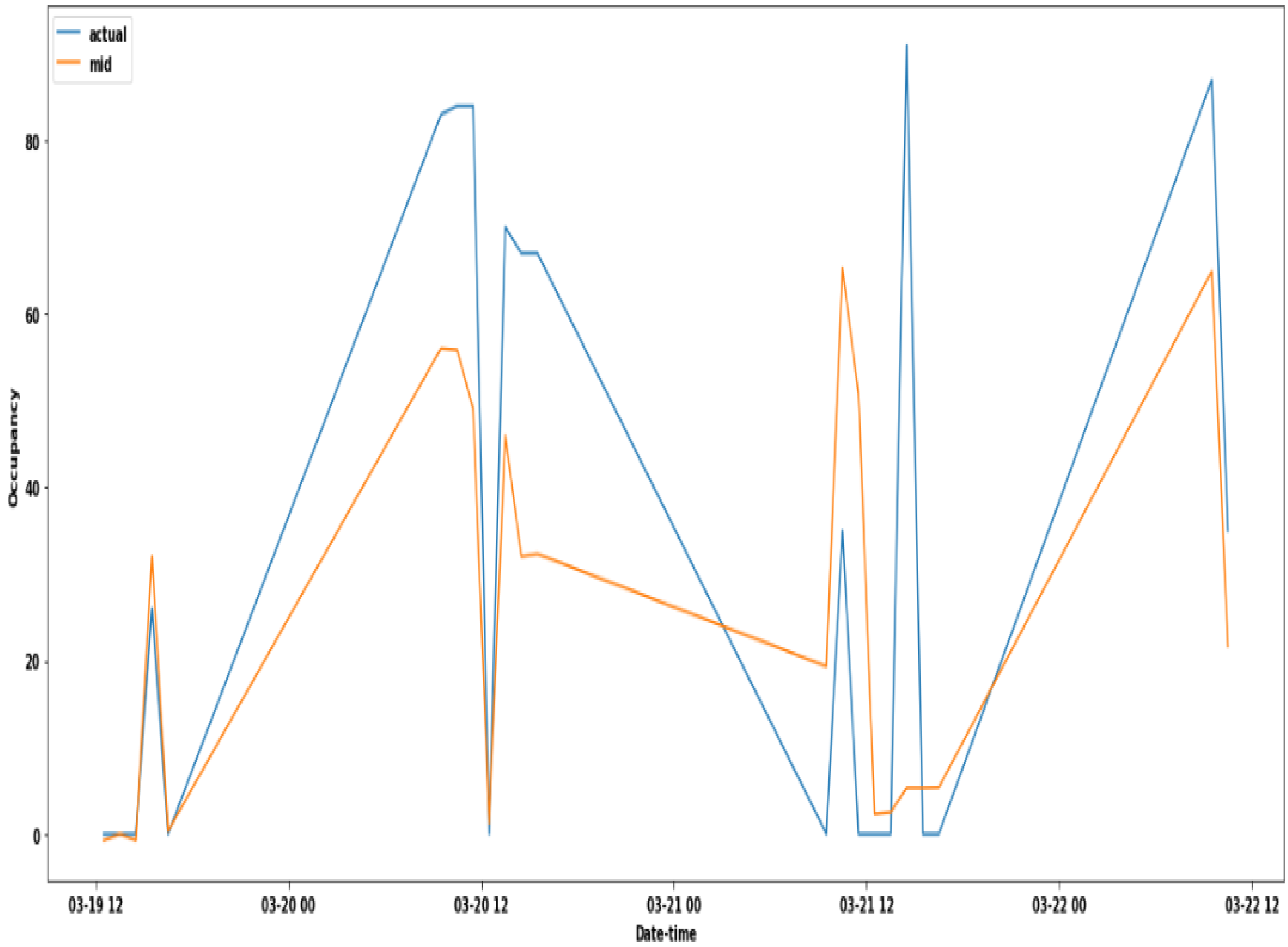


Fig. 6.12. GB prediction plot for actual versus predicted room occupancy using middle prediction limit

The absolute error metrics table for JA314 shown in Table 6.3, shows that the lower model prediction error has a lower value of 16.63519 in terms of median followed by the middle model prediction error with 17.033, which is actually surprising because of the obvious difference in their alpha values. The box plot in Figure 6.13 shows the same and also shows the in bounds value of 69.05%. This means that the predicted values that were between the lower and the upper bounds are slightly more than half of the time (entire test set). The MSE and RMSE performance indices were calculated with the prediction results obtained from GB model's alpha medium range and they are 19.534 and 28.385 respectively.

6.4.3.2 Implementing Lime and SHAP Methods

Training of JA314 dataset with 1000 GB trees and different ranges (upper, lower and middle) of alpha values resulted in the about 54.55% of the test set comfortably falling within the prediction interval as shown in Figure 6.13. This model could somehow be

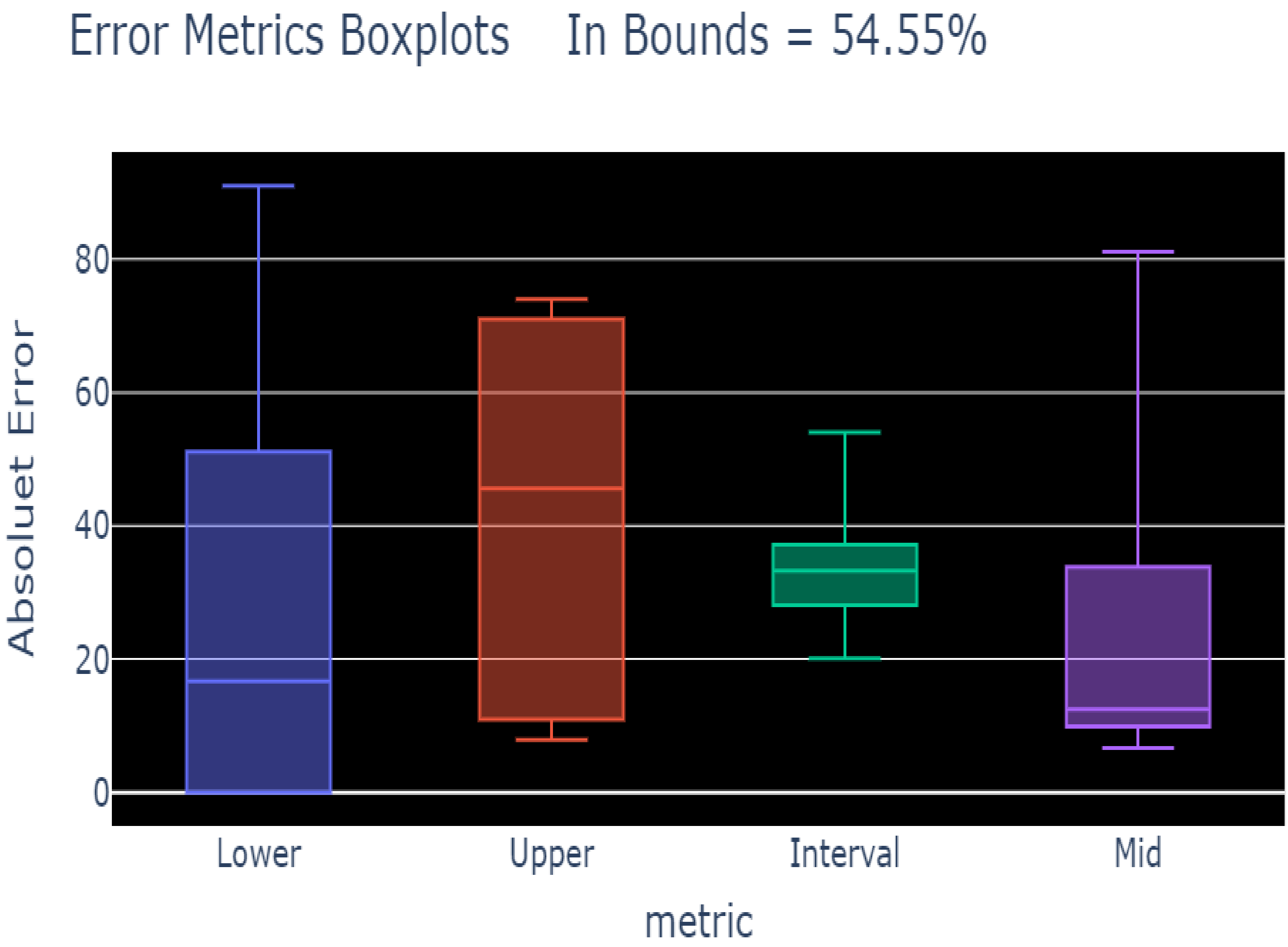


Fig. 6.13. Mean Absolute Inbound Plot for JA314 Dataset

	abs_error_lower	abs_error_upper	abs_error_interval	abs_error_mid
count	2.200000e+01	22.000000	22.000000	22.000000
mean	2.634436e+01	43.329310	34.836836	19.421354
std	3.002057e+01	27.033155	7.414702	21.386684
min	3.833854e-47	1.000000	23.517597	0.000555
25%	3.833854e-47	17.165440	29.890352	0.494378
50%	1.663519e+01	53.236206	33.000000	17.033167
75%	5.105000e+01	66.000000	40.711224	30.155514
max	9.100000e+01	75.472412	53.263794	85.794288

Table 6.3: Absolute errors metrics for predicted intervals for JA314

trusted if accuracy is the only measure of trust in terms of ML computing. Regrettably, this is not always the case in the real world practical applications. Therefore, it is important for the interested participants and non-experts to understand the rationale behind the model's behaviour as it was used by the data scientist. Understanding the reason why the model made such prediction will help non experts to make decisions or predictions on their own if need be.

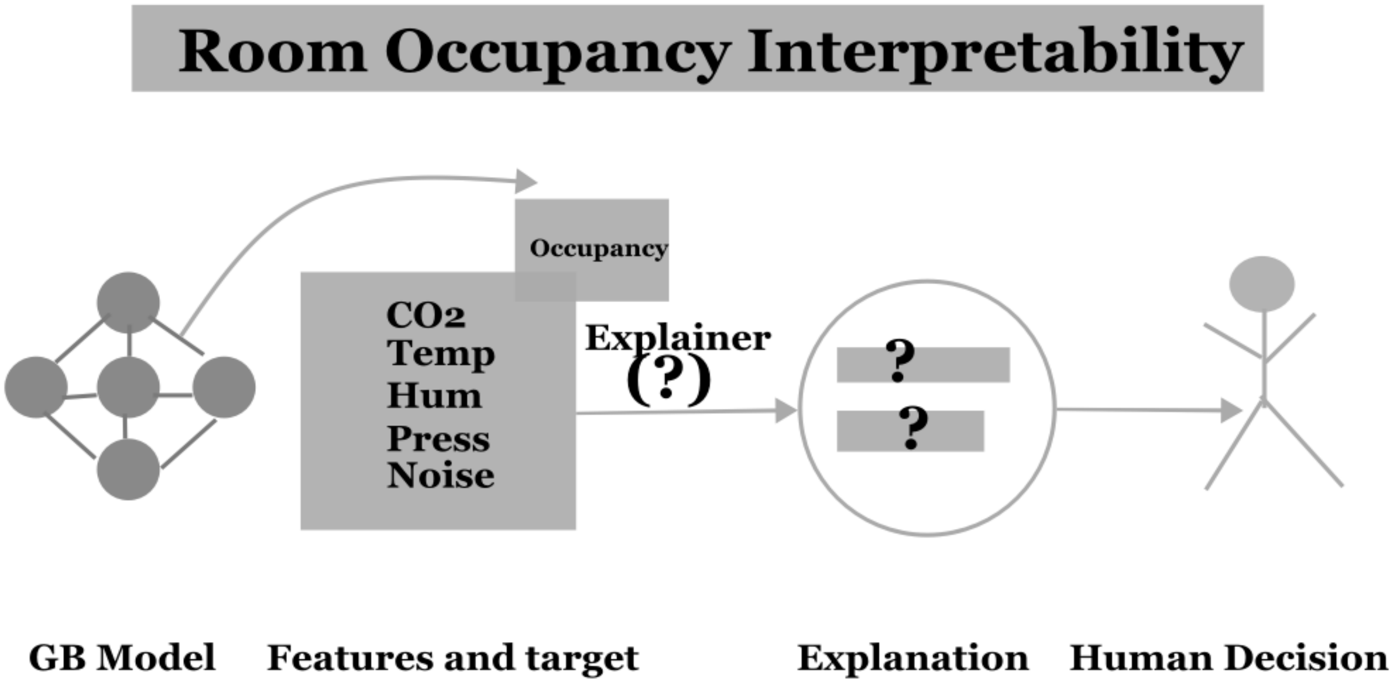


Fig. 6.14. GB ensemple model explainability for room occupancy study

To illustrate what this section is all about, the diagram shown in Figure 6.14 represents a concept where a model (GB) that was used to train the dataset (features and target) in order to make a prediction (occupancy) needs to be explained to a participant (human). Invariably, the participant wants to know (by asking question) about the logic behind the model's prediction results.

Therefore, in this section the LIME and SHAP values were used to explain the prediction results made by the GB ensemble model. These two additional tools were used in order to answer the research question “ **Which interpretable ML method is easy for non-expert to understand during the evaluation of the model’s interpretability.”**. More detailed explanation of the origin, differences between SHAP and LIME are discussed in the Section 6.3.6. The algorithm behind LIME method is discussed in Section 6.3.6.4. In order to be able to explain the GB model with LIME and SHAP, the following procedures were followed:

- **Build SHAP/LIME explainer with the middle GB model(alpha=0.5).**
- **For LIME, treat features with less than 10 unique values as categorical features.**
- **Use the SHAP tree explainer to predict SHAP values of the instance of the test set or LIME tabular explainer to predict one instance from the test features.**
- **For SHAP, plot an individual force plot to show the features pushing the prediction to and from the baseline (expected value).**
- **For LIME, use *show_in_notebook* method from LIME class to see the features that are either positively or negatively influencing the prediction (intercept).**

Explaining SHAP Values with Individual Force Plots:

In individual force plot by SHAP, there are two important values; the output (GB prediction) and the base (SHAP expected) value. The plot shown in Figure 6.15, has 0.00 as output value and 28.14 as base value. SHAP’s base value simply means what GB model is expected to predict if no feature is known from the current observation, whereas “output value” means what GB model predicted with known features. Therefore the results means that the model predicted that there are no occupants in the room, however SHAP expected approximately 24 people in the room. The red and blue colours are the features that push the prediction higher (towards the base value) and lower (away from the base value) respectively. In other words, for this particular instance, SHAP values explained that the CO_2 value of 414 and humidity value of 31 contributed in moving the SHAP prediction away from the base value. Whereas noise (38) and pressure (1025) positively contributed in pushing the output towards 24.4; the expected (base) value. These results account for only the first row of the JA314 test set.

The same explanation applies to Figure 6.16 for 15th row of the test set. Looking at the result of the 15th instance shown in another SHAP’s force plot; Figure 6.16, the output (GB model) value is 32 and the base (SHAP) value is 3.13, and the base value is 28.4. In this case the CO_2 value of 482 (as a major contributor) was lowering the output value by the GB model, while pressure value of 1022 was pushing the occupancy value high.

6.4.3.3 SHAP Feature Importance:

In the summary plot for SHAP’s feature ranking shown in Figure 6.17, the regressors are shown in a way that they are either positively or negatively related to the target variable (room occupancy). The first plot shown in Figure 6.17a shows the average impact of all the features and how it affects the target values. Whereas the second plot shown in Figure 6.17b shows SHAP values for every instance of the prediction. This means that the summary

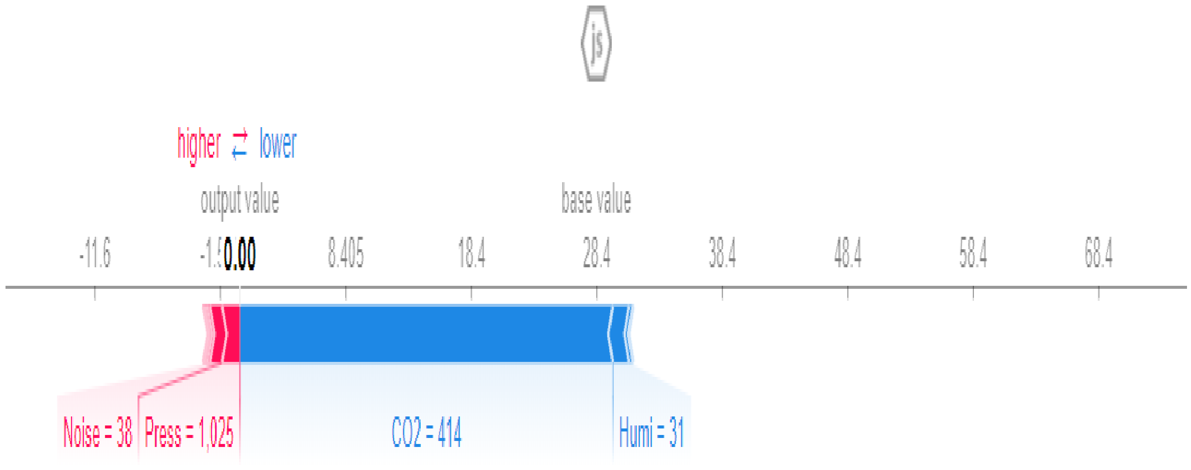


Fig. 6.15. SHAP Individual force plot for JA314 dataset

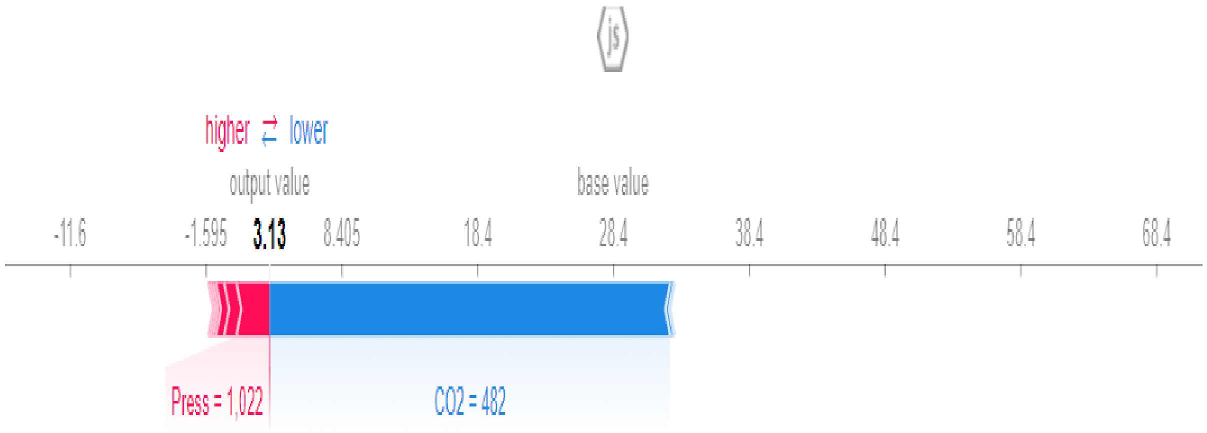
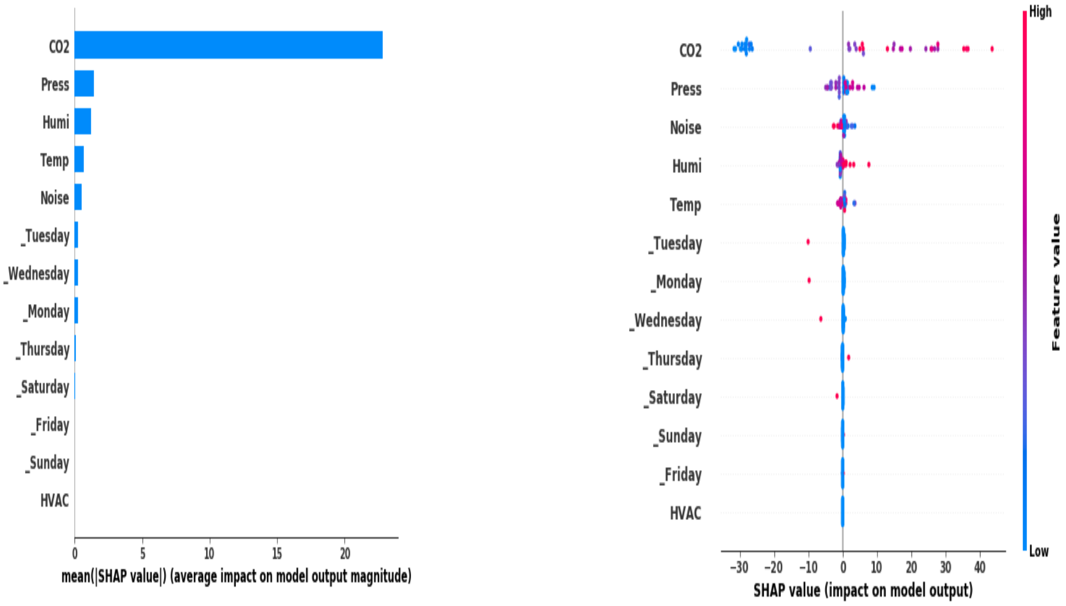


Fig. 6.16. Second SHAP Individual force plot for JA314 dataset

plot of SHAP can be likened to feature importance in other ML model. The two plots are both ranked in the descending order. The dots in the individual summary plot are from the training set. The horizontal position shows the variables that are associated with the higher or lower prediction.

From the correlation angle in the Figure 6.17b, the red dot colors show high value of the variable while blue dot colors show low value. High CO_2 value has high and positive impact on the room occupancy. High comes from the red color and positive impact is on the x axis. Similarly, it can be said that noise is both positively and negatively correlated to the room occupancy of JA314 dataset. Above all, from the SHAP explanation, CO_2 is the highest contributor to the room occupancy estimation.



(a) SHAP summary mean impact

(b) SHAP individual summary impact

Fig. 6.17. SHAP feature impact ranking for JA314

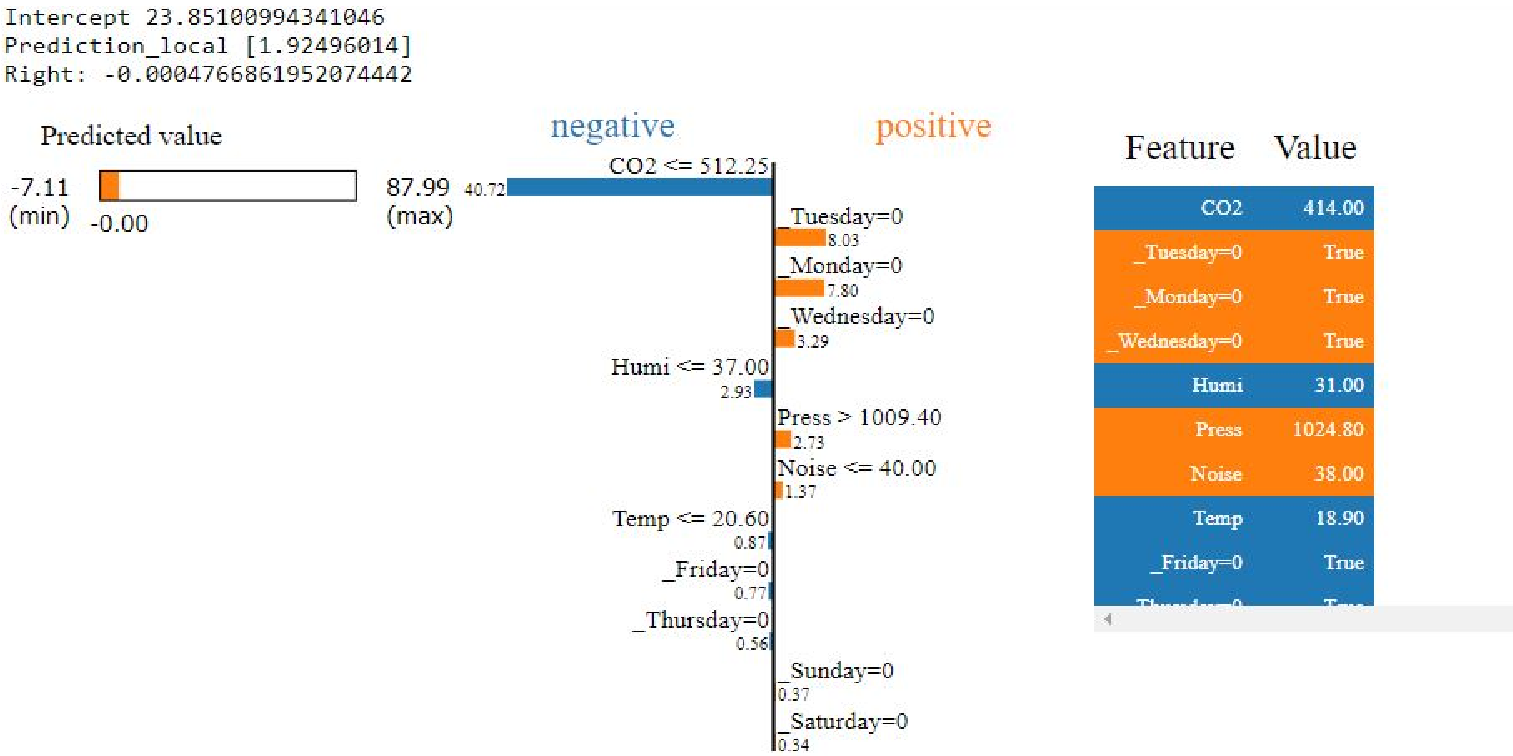
LIME Plot:

The steps of LIME method has been discussed in Section 6.3.6.4 for more understanding of the principle behind LIME. LIME's feature importance differs from SHAP because SHAP has a more theoretical background for its explanation and it is best with decision tree ML models, while LIME is best for classification problems and linear models. The plot shown in Figure 6.18 represents the LIME model for explaining the GB ensemble model used with the JA314 (with no missing data) dataset for the estimation of room occupancy.

In the LIME, the explanation is done by approximating the primary (GB) model locally by an interpretable one. The orange color indicates positive influence to the local prediction of that particular instance, while the blue color indicates negative influence to the intercept of the same instance. In this instance of the LIME plot shown in Figure 6.18, the global prediction is -0.0. The global prediction in LIME is the same as the "output value" in SHAP. Also the intercept in LIME is the same as the base value for the same instance (row) in SHAP. In this LIME plot, the intercept is approximately 23.85. Similar to SHAP, the top contributor to this model for this instance is CO_2 that is between 512.25 and 414. In the same vein, pressure between 1009.40 and 1024.80 has a positive influence. If the day is not Tuesday, Wednesday or Monday and temperature is between 18.90 and 20.60, the number of people in the room are positively affected.

6.5 Confirmatory Study Analysis with Swarm Dataset

In the preliminary room occupancy study discussed in the previous section, it has shown that about 54.55% room occupancy can be accurately estimated with GB using prediction interval method. In order to test the same GB model on a dataset recorded from another



```
Intercept 23.85100994341046
Prediction_local [1.92496014]
Right: -0.0004766861952074442
```

Feature	Value
CO2	414.00
_Tuesday=0	True
_Monday=0	True
_Wednesday=0	True
Humi	31.00
Press	1024.80
Noise	38.00
Temp	18.90
_Friday=0	True

Fig. 6.18. LIME model explainability for JA314 dataset

room. A confirmatory room occupancy study was conducted. The physical properties of the room were discussed in Section 6.5.1 and the reason for choosing this second office is explained in Section 6.3.3. The properties of the camera used for this confirmatory study has been discussed in Section 6.3.2. This confirmatory study analysis used exactly the same methods, parameters and approaches used in the study analysis of JA314 dataset. All the plots, the explainers (SHAP and LIME) used to interpret the GB model were also applied on this Swarm dataset.

6.5.1 Summary Statistics for Swarm Dataset and Room Information

The Table 6.4 shows the first five rows of the Swarm dataset. During the dataset recording windows and doors were closed and open respectively hence the reason for 0 and 1 values. This can be seen in(Figure 6.7) throughout the time of the dataset recording. The dataset information for Swarm room is enumerated below:

1. The second experiments was carried out at the room shown in Figure 6.6 in order to support the test for the hypothesis (please see Section 6.1.2) already tested with JA314 dataset. The venue where the second occupancy study took place was the development room at SwarnOnline Ltd.

2. The SwarnOnline Ltd room was chosen as a live environment testing for the GB model used in the preliminary study. This room has between 10 to 15 seating capacity. The Heating of this room is constantly turned on from 9.00am in the morning to 17.30pm in the evening, Mondays through Fridays.

	Date-Time	Temp	Hum	CO2	Noise	Press	Occupancy	Window	Door
0	2020-01-22 09:00:00	21.9	48	604	55.0	1040.1	6.0	0	1
1	2020-01-22 09:15:00	22.2	49	730	46.0	1040.	7.0	0	1
2	2020-01-22 09:30:00	22.5	48	790	47.0	1040.1	6.0	0	1
3	2020-01-22 09:45:00	22.8	48	808	48.0	1040.1	8.0	0	1
4	2020-01-22 10:00:00	22.9	48	839	63.0	1040.2	6.0	0	1

Table 6.4: First-five Row of Swarm Dataset

3. The datasets were recorded only during work days;

4. For the Swarm dataset, it has 627 rows with 8 (variables) columns. The first five rows of the Swarm dataset is shown in Table 6.4. The first five columns consists of temperature, humidity, CO_2, noise and pressure measured in ^oC, %, parts per million (ppm), decibel and millibar respectively. Additional three more columns are occupancy, window and door.

5. The occupancy is the target (dependent) variable, while the temperature, humidity, CO_2, noise and pressure are the independent variables.

6. For Swarm dataset, the window values are 0 throughout the period study was being observed because it was closed during work days. The door values are 1 throughout because the internal door connecting the study room with the other rooms are left open during work days. These two columns were later dropped because their values do not vary.

7. All the data recorded from the Swarm office were for approximately four weeks. The recording started on 22/01/2020 at 9.00am and ended on 14/02/2020 at 17.32pm.

8. The dataset were recorded at a frequency of 15 minutes, Monday to Friday daily except bank holidays and weekends.

There are 627 instances and 6 features after dropping the two columns. The features are shown in Figure 6.19 as Temp, Humi, CO_2, Noise, Press and Occupancy for Temperature, Humidity, CO_2, Noise, Pressure and Occupancy respectively. In the Swarm dataset feature plot, noise and occupancy have the same pattern of movement which shows that its values vary simultaneously with time. Pressure features shows gradual declining and rising pattern throughout the period of observation. The CO_2 has the same upward and downward pattern with occupancy in most data points. The temperature and humidity have similar pattern in most of its data points.

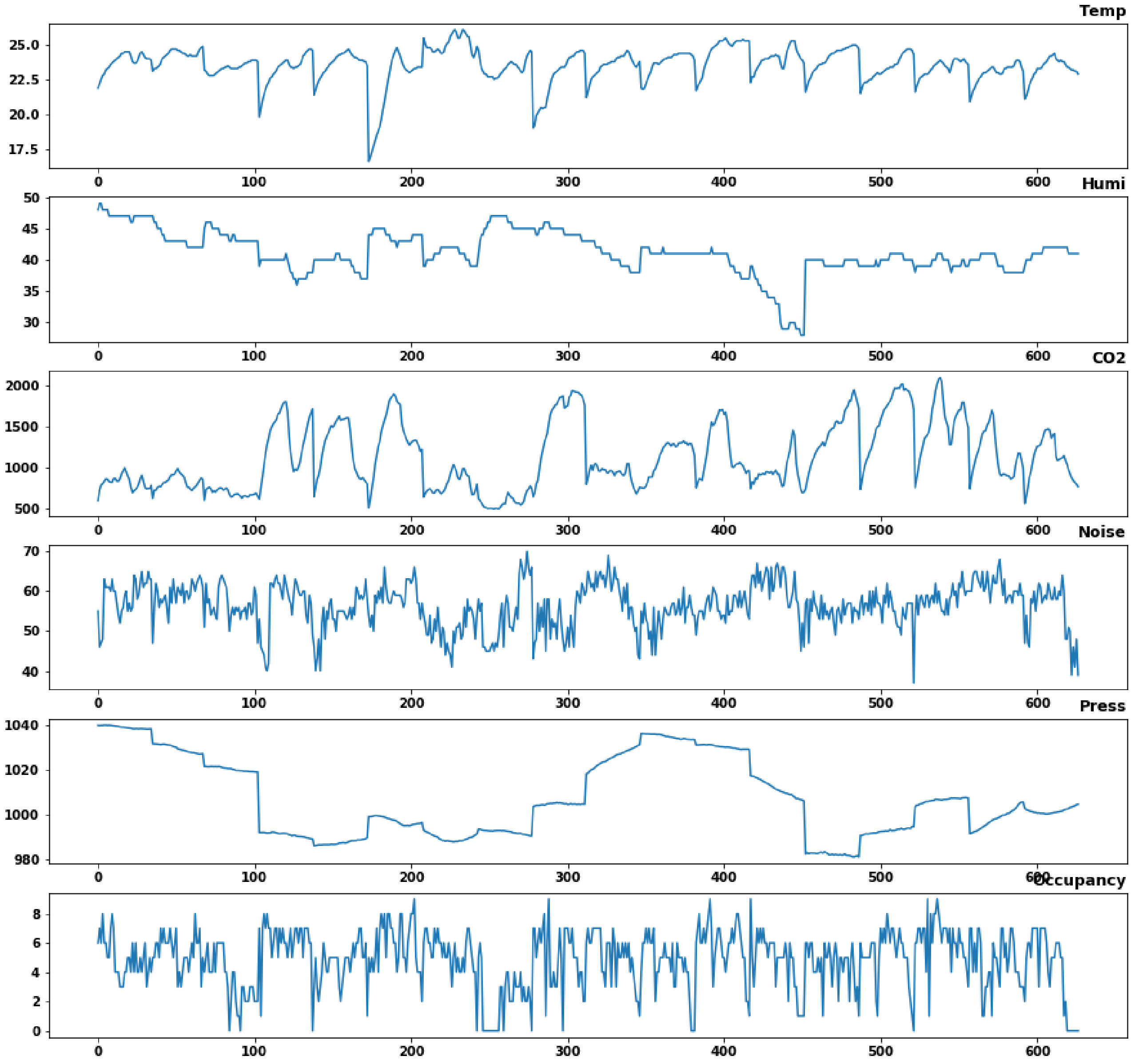


Fig. 6.19. Swarm dataset visualization plot for all the features

6.5.2 Implementing GB Model on Swarm Dataset

Before the application of GB ensemble models on the Swarm dataset, all the data preprocessing technique used on JA314 dataset analysis were done. All the codes can be seen in Section S.

The same GB model with the same alpha and quantile values used previously in the JA314 dataset were used to train (66%) and test (34%) the Swarm dataset. The prediction interval plot for Swarm dataset shown in Figure 6.20 has the dark violet color representing the actual observations, the orange colour representing the upper prediction with alpha (0.95), the dark blue colour representing the lower prediction with alpha (0.05) and the green colour representing the middle prediction (default GB model parameters) with alpha (0.5).

Similarity, the same prediction values obtained from the GB default alpha parameter (0.5) was used to plot the Figure 6.21. In this plot, the actual occupancy values are shown in blue while the predicted occupancy values are shown in orange. The performance of the model from the plot shows that GB model performed well with the plot lines of the actual versus predicted closely aligned to each other.

6.5.2.1 Error Metrics and Plots for Swarm Dataset

The performance score of all the predicted interval results were calculated in terms of absolute error as can be seen in Section S. It was observed that about 82.24% of the predicted test set were within the bounds of the prediction interval. The absolute error plot is shown in Figure 6.22. Also, the RMSE and MSE for the GB's model performance with the Swarm dataset are 2.227 and 1.709 respectively, which is an indication that the model performed well with the Swarm dataset.

	abs_error_lower	abs_error_upper	abs_error_interval	abs_error_mid
count	214.000000	2.140000e+02	214.000000	214.000000
mean	2.808021	2.878505e+00	2.843263	1.690696
std	1.608440	2.021993e+00	0.708782	1.403007
min	0.039950	3.552714e-15	1.048665	0.021898
25%	1.347784	1.000000e+00	2.316335	0.636031
50%	2.829748	2.000000e+00	2.836370	1.303768
75%	3.761486	4.000000e+00	3.220195	2.520425
max	7.351750	8.000000e+00	4.889179	5.854884

Table 6.5: Metrics table for Swarm dataset

The absolute error score can be buttressed with the prediction table metrics shown in Table 6.5, where the middle (alpha=0.5 or loss=quantile) GB model consistently has less

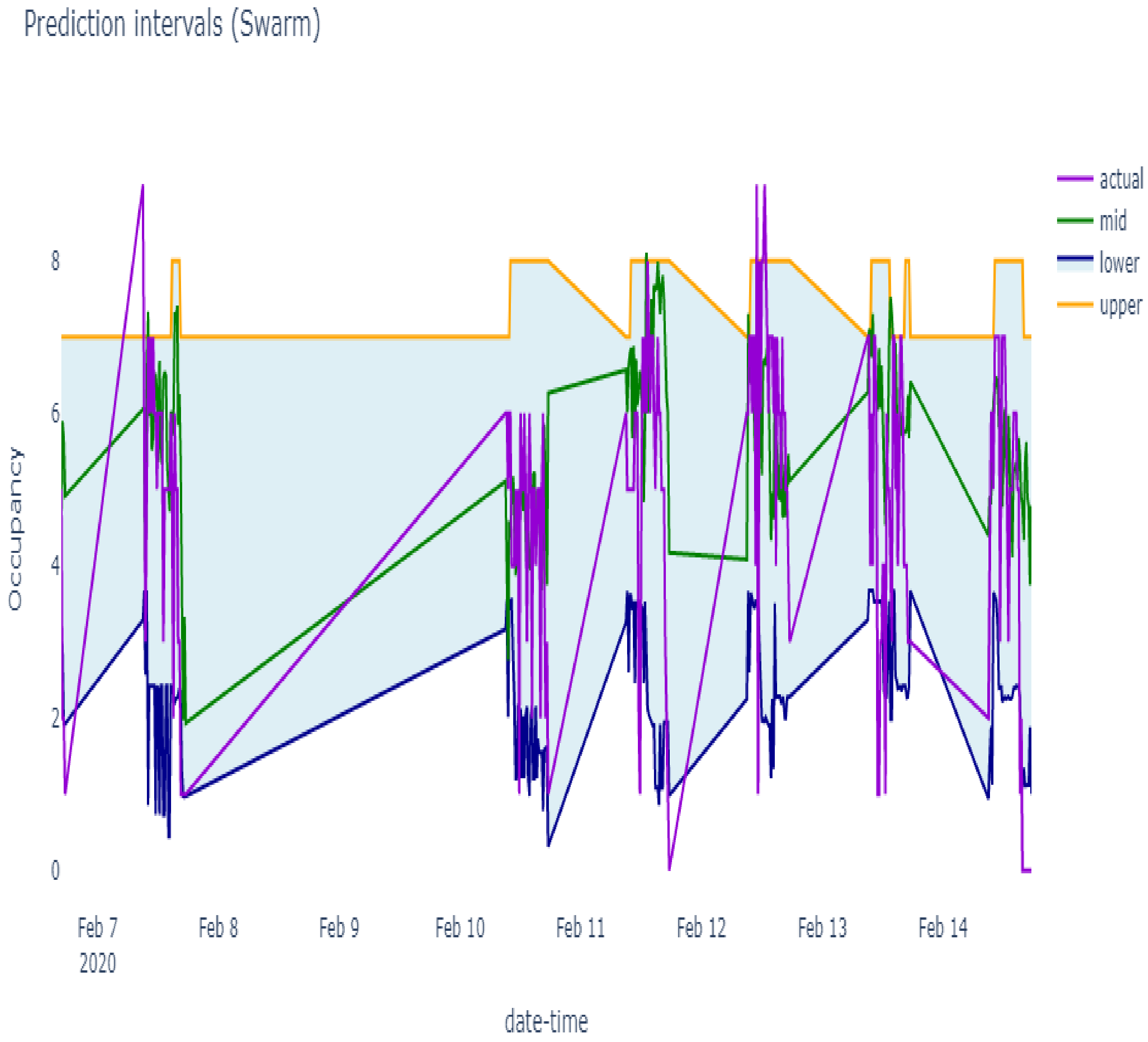


Fig. 6.20. Prediction interval plot with Swarm dataset

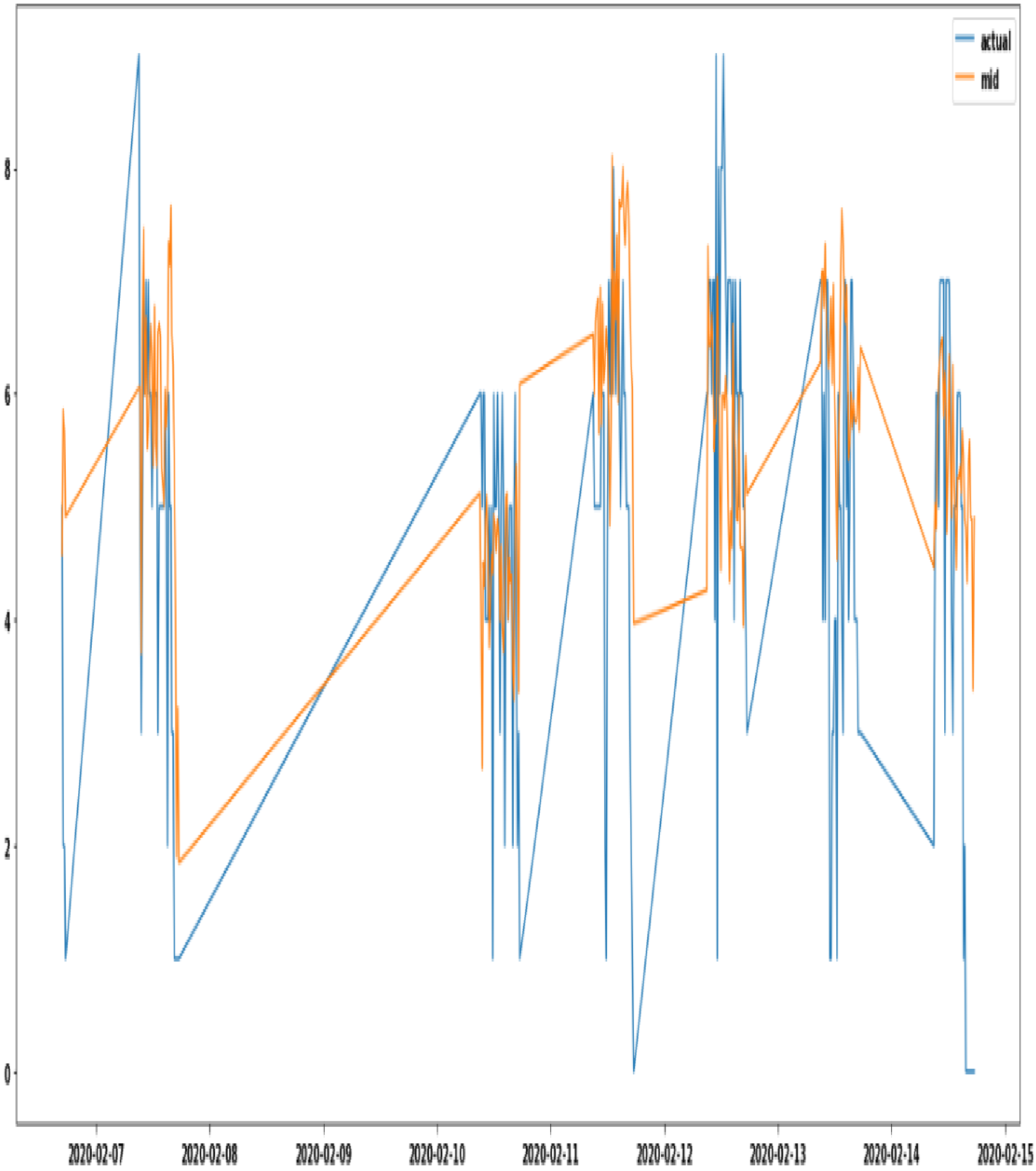


Fig. 6.21. Prediction plot of actual versus predicted occupancy with Swarm

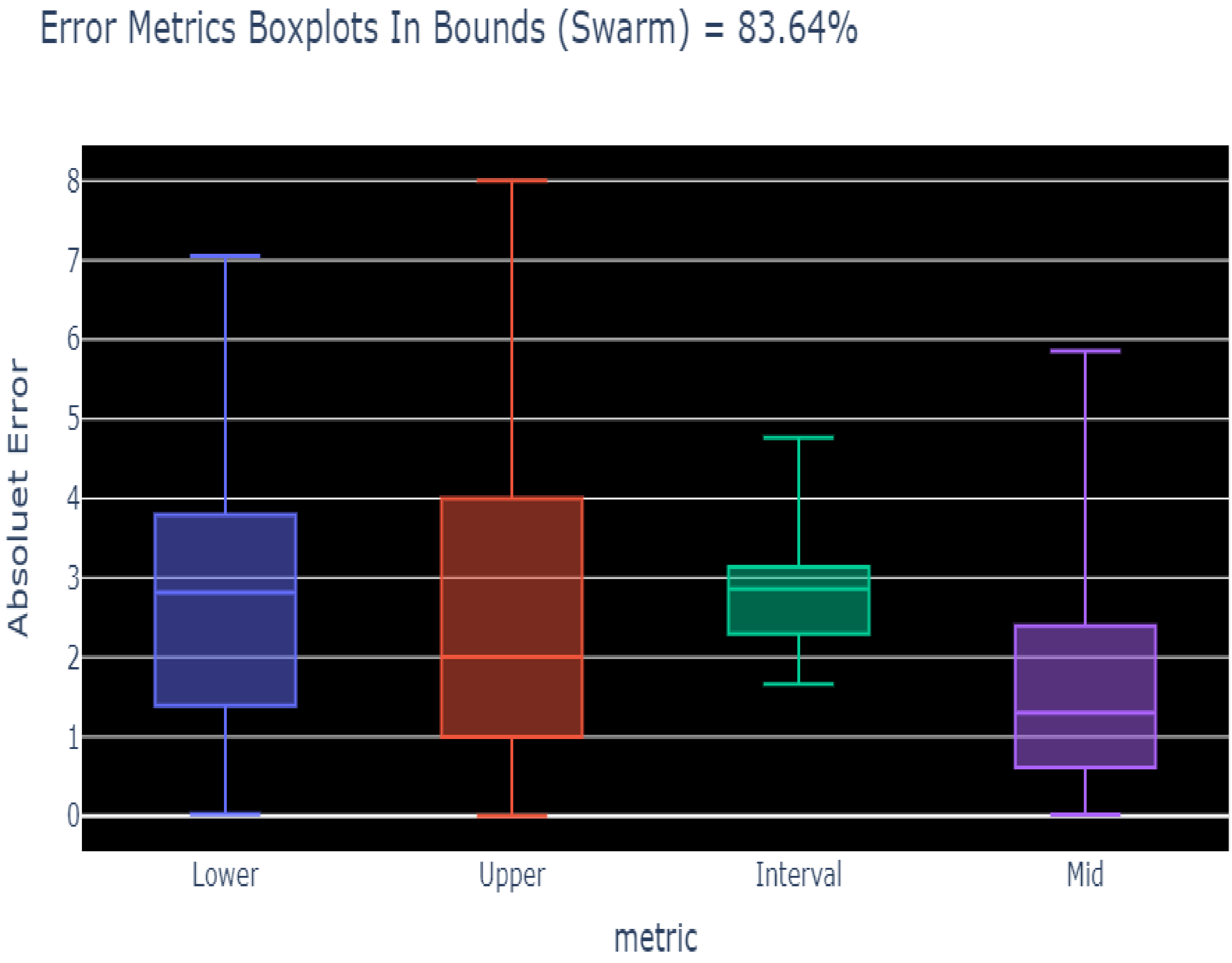


Fig. 6.22. Mean Absolute Inbound plot with Swarm dataset

error value in terms of 25%, median (50%), 75% and max quantiles. This means that with the complete Swarm dataset, the middle model outperformed the upper and the lower GB model. The column abs_error_interval represents the values (that penalizes the errors) of the upper absolute error and the lower absolute error divided by two.

6.5.2.2 Explaining the Prediction Results of Swarm Dataset with Lime and SHAP

SHAP Values:

The GB model was interpreted with LIME and SHAP methods that were discussed in Section 6.4.3.2 in order to understand the features that contributed to its prediction results. The plot in Figure 6.23 represents the individual SHAP plot with an instance of a test set from the Swarm dataset. In this plot the output (GB prediction) value for the 47th instance is 3.76, while the base value is 4.877. The noise (54), pressure (982.9) and temperature 25.3 contributed in pushing the output value lower (away from the base value), whereas the CO_2 (1237) and humidity (40) contributed to pushing the prediction higher towards the base value.

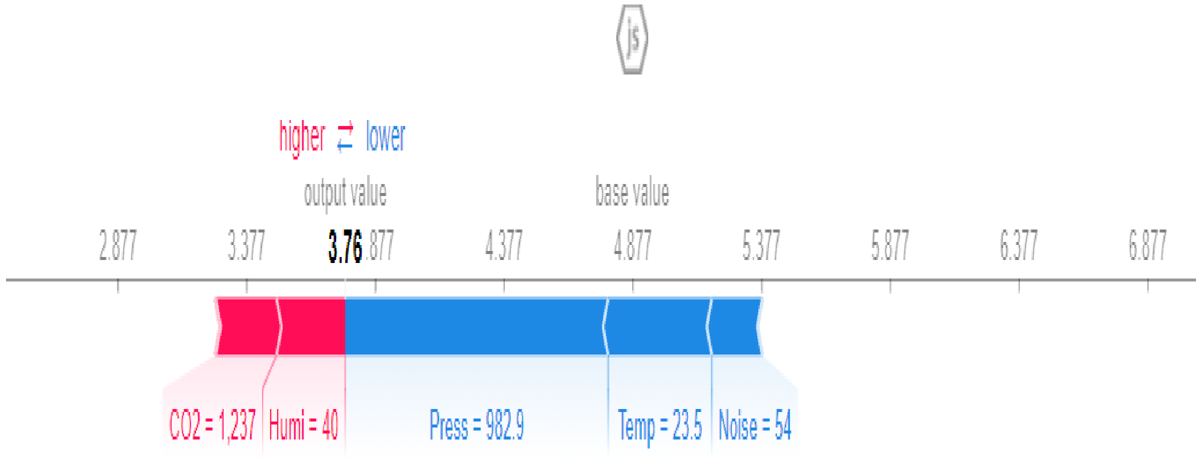


Fig. 6.23. First SHAP values for individual force plot with Swarm dataset

Specifically, the general impact of CO_2 in the entire test set as it was predicted by the GB model is shown in Figure 6.24. The CO_2 impact plot shows that the majority of CO_2 contributed in pushing the output (GB prediction) value on the y axis above the base (SHAP values) value (4.877).

LIME Plot:

The LIME values shown in Figure 6.25 can be compared with the SHAP values shown in Figure 6.23 because the two outputs are from the same instance (47th row). In LIME values, the global prediction (Right value) is 3.76 which matches the SHAP output value in Figure 6.23. LIME's intercept value is 5.8 which is higher than the SHAP base value of the same instance. The CO_2 and noise values positively contributed in the LIME's global output, while temperature less than 23.80 negatively contributed to the prediction result.

6.6 Results Summary

There were two results obtained from two versions of the room occupancy study analysis conducted. The two versions are 1) JA314 and 2) the Swarm dataset with its results shown in Table 6.6. These results are of three categories such as 1) the performance indices, 2) LIME values and 3) SHAP values.

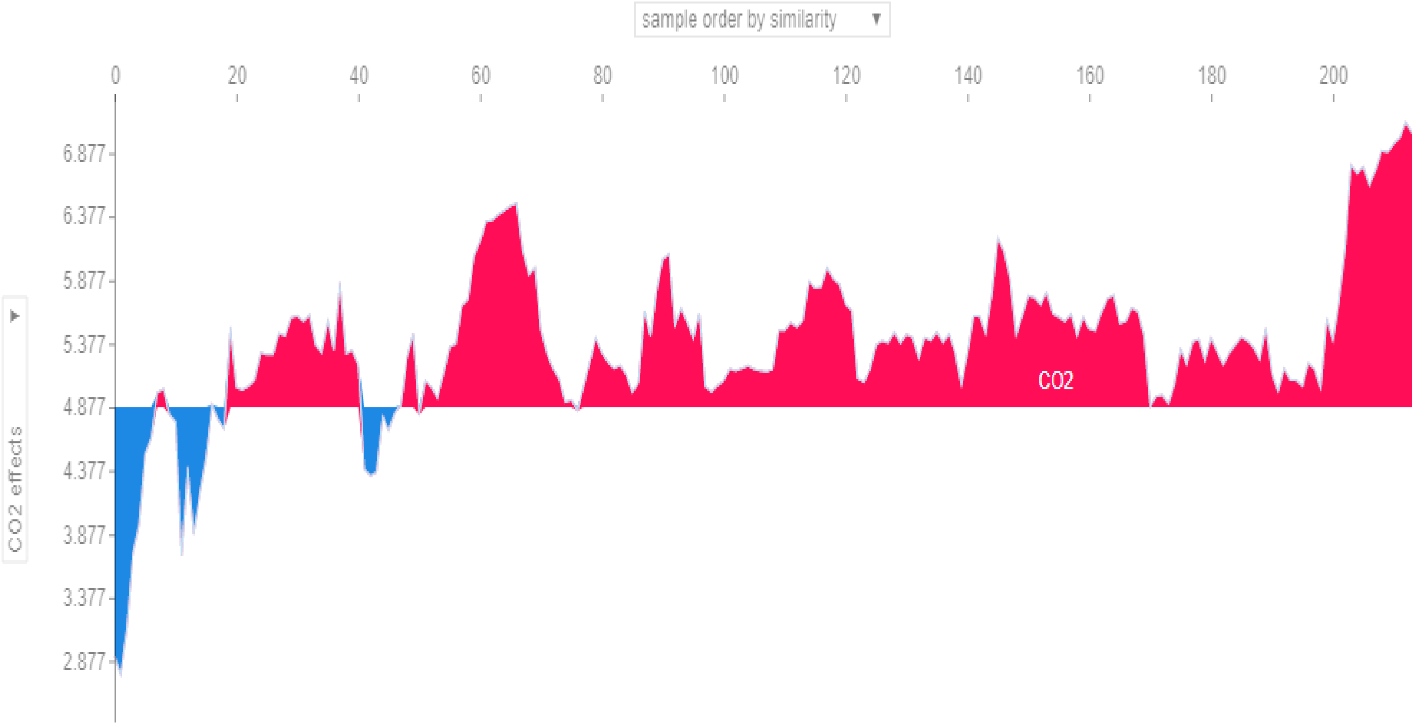
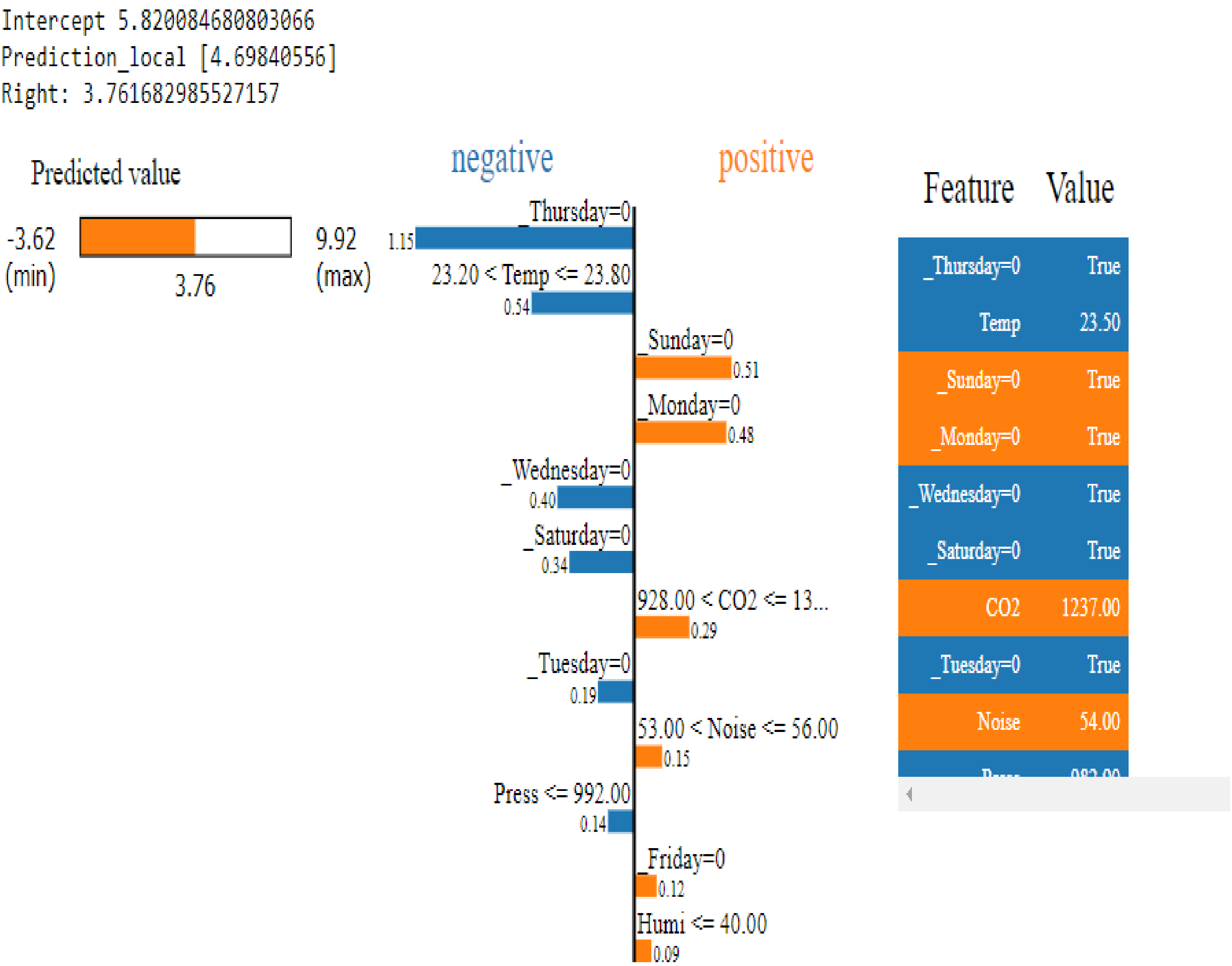


Fig. 6.24. SHAP CO_2 impact on the entire Swarm test set

JA314 dataset								
Performance Indices	LIME Values				SHAP Values			
	Individual Stats		Summary Stats		Individual Stats		Summary Stats	
Inbound: 54.55 % MSE: 19.545 RMSE: 28.385	Intercept	32.7	CO_2-	414	Base	33.1	CO_2-	414
	Global	-0.15	Temp+	18.8	Output	-0.15	Press+	1024.8
			Press+	1024.80			Temp+	18.9
			Humi-	31			Humi-	31
			Noise+	40			Noise	38
Swarm dataset								
Performance Indices	LIME Values				SHAP Values			
	Individual Stats		Summary Stats		Individual Stats		Summary Stats	
Inbound: 82.24% MSE: 1.709 RMSE: 2.227	Intercept	3.26	CO_2+	981	Base	4.877	CO_2-	981
	Global	4.58	Temp-	25.30	Output	4.58	Press-	1029
			Press+	1029.40			Temp+	25.3
			Humi+	37			Humi+	37
			Noise+	60			Noise+	60

Table 6.6: Results for room occupancy estimation study

The results shown in Table 6.6, are divided into two sections. The first section of the table represents the results of the JA314 dataset. The second section represents the results of



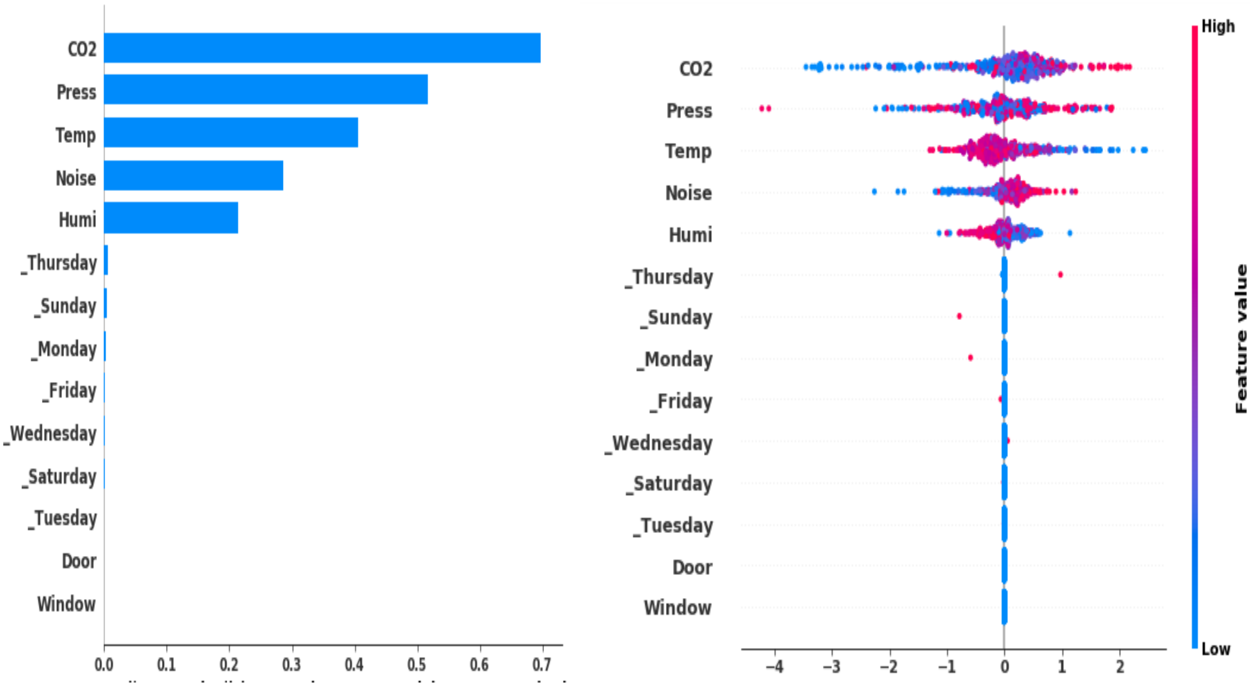
```
Intercept 5.820084680803066
Prediction_local [4.69840556]
Right: 3.761682985527157
```

Feature	Value
_Thursday=0	True
Temp	23.50
_Sunday=0	True
_Monday=0	True
_Wednesday=0	True
_Saturday=0	True
CO2	1237.00
_Tuesday=0	True
Noise	54.00
Press	992.00

Fig. 6.25. LIME interpretation plot with Swarm dataset

the Swarm dataset with imputed missing occupancy data. Furthermore, each of the sections were divided into three columns. The first column (performance Indices) represents the percentage of the actual observations that were correctly predicted within the prediction intervals by the GB model. Additional performance indices that were used in this study are the MSE and the RMSE. The second and third columns represent the explainers statistics from the LIME and SHAP model respectively.

The performance indices results were assessed based on the 33% of the test set from the original observation. The results were for the two different dataset recorded from two independent observations. Also the results shown in Table 6.6 can be said that the GB ensemble model showed better improvement with the Swarm dataset in all of its performance indices. That is more than 82% accuracy within the prediction range. This improvement could be because more (627 observations) dataset were used compared to the 64 observations that were used in JA314. The JA314 dataset achieved 54.55% of



(a) SHAP value (mean) impact **(b) SHAP value (individual) impact**

Fig. 6.26. SHAP feature impact ranking for Swarm dataset

accuracy score. This could indicate that the more the number of the observations the better the performance of the GB ensemble ML model for the estimation of indoor room occupancy.

Similarly, the MSE and RMSE of the two datasets confirmed the same improved performance as it was observed in the accuracy results of the two datasets. For instance, the Swarm dataset MSE and RMSE results is one tenth of the same indices for the JA314 dataset. The MSE and RMSE performance indices were obtained by comparing the actual occupancy values with the predicted occupancy values by the GB model.

Furthermore, in the individual stats columns of the two explainers (LIME and SHAP), the base (SHAP predicted/expected values) and the intercept (LIME predicted/expected) values have the same interpretation, likewise the Output (GB model prediction for SHAP) and the Global (GB model prediction for LIME) values. These similarities in meaning can be seen in the values shown in the individual stats column where there are only slight difference in the intercept (for LIME) and base (for SHAP) values, however the same values for global and output.

The reason for slight difference could be as a result of two different models with different algorithms. Additionally, the feature values in the summary stats column represents individual explainers from the LIME and SHAP models. The signs (+/-) attached to each feature represents the features that were positively or negatively pushing the GB model predicted values for that particular instance (for any row being explained) respectively. The results shown here are only for the first row respectively.

Moreover, feature importance plots of LIME and SHAP agrees that the CO_2 is most important (either positively or negatively) in the estimating room occupancy study. Its positive contribution depends on the combination with other variables (temperature, pressure, humidity and noise). However, the two explainers did not harmoniously agree on the importance of weekdays towards the prediction results.

In conclusion, ML models are prone to making wrong predictions especially outside the range of the data they were trained on, so it is important to test the results (especially in real world practical examples) explicitly so as to avoid unpleasant surprises. Therefore data scientist should worry more about prediction stability rather than parameter stability because parameters are often regularized.

6.7 Human-Based Evaluation of Interpretability for Room Occupancy Estimation Model

The previous Section 6.6 has shown that estimating room occupancy using GB model via prediction interval can achieve up to 82% accuracy. This section will evaluate the interpretability of the proposed GB ML model via two interpretatble ML methods known as SHAP and LIME. This research considered evaluating the model's interpretability because the commercial sponsor of this research study raised it as a key issue. Also because this problem (interpreting the ML models used for room occupancy estimation study) has not been sufficiently researched in the past.

According to Molnar [2019], already sufficiently researched problems with enough practical experience should not be interpreted. Therefore, in addition to a Post-hoc taxonomy (where models are applied after the data has been trained with a ML model) interpretable approach already used in Section 6.5.2.2, this research will now use human-based evaluation metrics discussed in Doshi-Velez and Kim [2017] in order to evaluate the interpretability of the proposed GB model. This means that non-experts will be invited for an experiment to test and see if they are able to trust and understand how the results of the model were achieved. Therefore, four non-data science student were invited for the evaluation study. The interpretable models were explained to them, after which they were asked to predict some values themselves. There results were graded and the percentage of model's interpretability were calculated based on the identified metrics.

6.7.1 Aims and Objectives

According to Miller [2019], interpretability is the extent to which a non-expert (human) can regularly predict the ML model's result. Therefore, the aims and objectives of this evaluation is of two parts. 1) to get people to try and predict some (rows) of the model's result without seeing the actual result. This will help to determine how often they make accurate predictions in order to evaluate the interpretability of the GB model. 2) to get the participants to choose the easiest method from the two interpretable methods. Doing so will help to answer the following research questions:

- How best can the ML model interpretability be evaluated in order to assess the interpretability of the GB model used for the estimation of room occupancy.
- What is the best explainable ML method to be used to interpret the results predicted

by the GB model so that non experts can make sense of the model's results and be able to predict the results by themselves.

6.7.2 Interpretability Data Collection and Evaluation

The model evaluation interview was done via Zoom video conferencing. The participants were four postgraduate students. Three of the students are in the computer and information science (Computer and Information Science Department, University of Strathclyde (CIS)) department at the University of Strathclyde. One is in the University of Glasgow. The video interviews took place on the 22/06/2020, 23/06/2020, 25/06/2020, 26/06/2020, 24/08/2020 and 29/08/2020. Please see Table 7 for participants interview appointment schedules. The approved ethics application number by the CIS department for the model evaluation is **Application ID: 1172**. The profile of the people who participated in the model evaluation video interview are stated in Table 7.7.

ID	Occupation	Sex	Time taken
E1	University of Glasgow (UoG) Student	F	60 minutes
E2	CIS Student	F	45 minutes
E3	CIS Student	F	34 minutes
E4	CIS Student	F	67 minutes

Table 6.7: Table showing interview participants for GB model evaluation and their occupation.

Before the commencement of the evaluation interview, a participant's information sheet detailing what the study was all about and what was expected of the participants during the interview was first sent to the participants via email. The consent form is shown in Section O. Among the two datasets used for the room occupancy estimation, only the Swarm dataset was considered for this evaluation. The reason for choosing Swarm dataset was because it has a complete dataset without missing data and it performed better than the JA314 dataset. Furthermore, only the results obtained by the middle interval (alpha = 0.5) was chosen for the evaluation.

At the start of the evaluation, different rows of the dataset were selected at random for explanation. The interpretable methods that were explained with the selected rows are SHAP and LIME. The plots generated with the selected rows by the interpretable methods are similar to the ones shown in Figure 6.18 and Figure 6.18 respectively. Two major phases were used for the evaluation. They are the explanation and the testing phase. The following steps were followed during the evaluation process of the proposed GB model by all the participants:

Explanation Steps:

1. In the explanation phase, five random rows were chosen by the interviewer from the test set of the Swarm dataset. In these randomly selected rows, all the values

corresponding to the independent variables and the corresponding GB model's results target variable (occupancy results) were shown to the participants.

2. The meaning of the selected rows and its results were then explained to the participants via SHAP and LIME graphical representation.
3. The participants were allowed to ask questions pertaining the explanation of the results where necessary.
4. Steps one and two above are repeated for every rows for every participants.

Testing Steps:

1. During the testing phase, another five rows were selected at random by either the interviewer or the participants. Only the independent values of the rows were revealed to the participants.
2. The reason for the step one above was done for the participants to try to predict the number of people they think might be in the room based on their understanding of the explanation phase above.
3. The testing of the rows were done one after the other. The participants guessed answers (occupancy results) were assumed for the SHAP and the LIME methods.
4. The participants were then asked to say which of the interpretable method was easy; in terms of readability for them to understand based on the interpretation of its diagram.

The table that the participants were asked to complete is shown in Section N.3. The "Guess-Occupancy" column was for the occupancy values predicted by the participants, while the "Occupancy-actual" column is the actual middle interval values predicted by the GB model. The same steps of evaluation was followed for all the participants.

6.7.3 Interpretability Outcome and Conclusion

The results shown in Table 6.8 represents the results of the explanation and the testing phase for the GB model evaluation. The participants are named Participants E1 to E4. The first column named "Expl row numbers" refers to the randomly selected rows for the explanation phase of all participants. The column named "Test row number" refers to the randomly selected rows for the testing phase. The column named "Part pred val" refers to the predicted values by the participants. The column named "GB model val" refers to the actual values predicted by the GB model for the occupancy. The column named "Eval" refers to the evaluation results.

The evaluation of this model was done by considering error of +/- 1. This means that if the values predicted by the participants is within the range of +/- 1 the actual results predicted by the GB model, it will be counted as correct (1), but if the participants guessed results that is not within the range of +/- 1 it will be counted as incorrect (). shown in Table 6.8 0. In Table 6.8, E1 predicted 80% of the model's results correctly and 20% incorrectly during the testing phase. Participants E2, E3 and E4 predicted 60% of the model's result correctly and 40% incorrectly.

Expl row numbers	Test row number	Part pred val	GB model val	Eval
Participant E1				
1	5	5	6.5	1
7	23	5	5.24	1
33	50	3	4.37	1
35	42	4	4.42	1
51	70	2	4.91	0
Participant E2				
2	10	5	6.56	1
10	15	5	6.21	1
25	50	5	4.37	1
60	100	4	7.37	0
70	80	2	6.88	0
Participant E3				
5	11	3	5.51	0
22	35	4	4.13	1
40	52	4	4.80	1
90	92	5	7.07	0
87	102	7	7.66	1
Participant E4				
8	17	4	6.66	0
18	27	2	5.70	0
35	47	4	3.77	1
73	95	6	6.34	1
100	107	5	6.02	1

Table 6.8: Participants' answers and the GB models results for Interpretability evaluation

Therefore, the total number of results predicted by all the participants correctly is 65% and incorrectly is 35%. It can be concluded that the interpretability of GB ML model for the estimation of room occupancy is above average based on the randomly selected twenty (n=20) rows of the test set. This evaluation has answered the research question which states that **"How best can the ML model interpretability be evaluated in order to assess the interpretability of the GB model used for the estimation of room occupancy."**

In terms of the choice of the easier interpretable methods based on readability, all the participants agreed that SHAP is better that LIME. The following were their statements:

- **E1:** "I understand the concept... It will be more easy for people to predict the results if they have the app installed in their room..." , "SHAP is still simple..."
- **E2:** "It seems that LIME has a different output... I prefer SHAP" .... "Though not too much of a difference"

- **E3:** "I think is because you explained SHAP to me better than the LIME... So I prefer SHAP."
- **E4:** "The first one (SHAP) is easier to read.... ", "but this (LIME) is interesting because they show you data of a time", "for simplicity purposes I like the first one (SHAP)... but this (LIME) is also bringing more details which is also valuable..."

The research question which states that **"What is the best explainable ML method to be used to interpret the results predicted by the GB model so that non experts can make sense of the model's results and be able to take predict the results by themselves."** can be answered by saying that SHAP is easy to read than LIME.

6.8 Findings and Discussions

Interest in room occupancy study has continued to advance among room and estate managers because of its great potentials such as 1) energy savings, 2) fight against climate change and 3) room capacity management. In this room occupancy research, some problems were noted from the interview analysis and literature review. Some of the problems are 1) **making sure that the ML model used for the room occupancy study is not room dependent** and 2) answering a question from the participant (P3) who said **"I also am interested in knowing whether or not there are any technique to it ... to save energy by perhaps increasing or decreasing the heating or the lighting in the room depending on the occupancy levels"**. Safely answering two of these questions and at the same time achieving the main aim of this research needs a simple and clear method such as the prediction interval approach.

The data analysis of the JA314 dataset with 64 observations achieved a performance score of 54.55% while the Swarm dataset with 627 observations achieved a performance score of 82.24%. This performance improvement suggests that the more the number of observations the better the prediction interval result.

The absolute error score of the lower and the middle bound prediction interval continues to be lower than the upper bound in terms of median (50 percentile), which suggests that among the three different room occupancy analysis done in this research, the prediction made by the lower and the middle bound closely aligned with the actual observations hence the reason for the lesser error scores. The metrics tables can be seen in Table 6.3 and Table 6.5.

Currently, it is believed that most non-experts who make use of the real-world applications developed or powered by ML methods don't know how most ML models make their predictions as such, those individuals are limited by the kind of decisions they take when using those applications Ahmad et al. [2018]. This problem is compounded by the fact that most accurate models are not interpretable and most interpretable models are less accurate Marco Tulio Ribeiro [2021]. Therefore, data scientist suggests establishing a trade-off between the interpretability and accuracy of ML models.

Hence, this room occupancy research used two known ML model interpretable methods

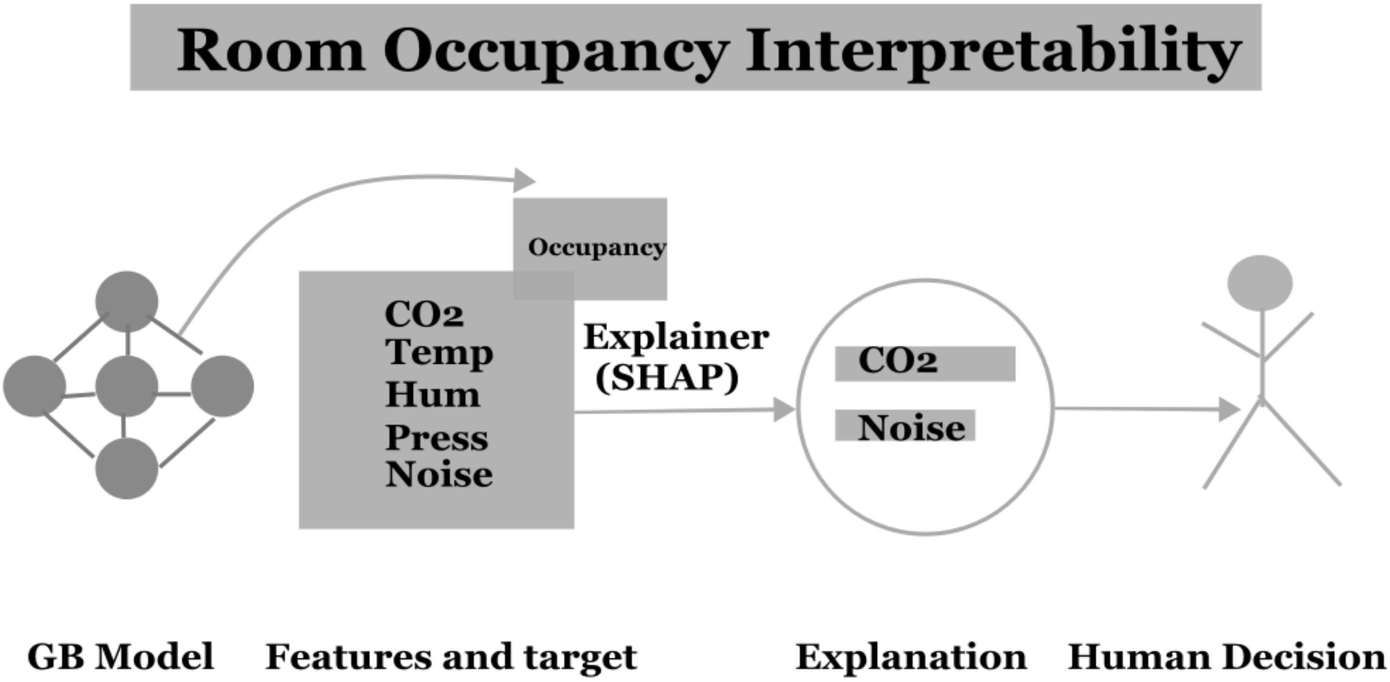


Fig. 6.27. GB ensemple model explainability for room occupancy study

(SHAP and LIME) to interpret how the GB model made its room occupancy predictions. From the SHAP and LIME value plots, the two explainers believed that CO_2 feature is the regressor to the estimation of room occupancy levels. Other features such as noise, temperature, pressure and humidity contribution levels interchanges depending on the dataset. However, the mean SHAP values impact for Swarm dataset with better prediction accuracy showed CO_2, pressure, temperature, noise and humidity in order of their contribution respectively. Furthermore, SHAP summary plots agrees that the days of the week plays an insignificant contribution to the estimation of room occupancy with GB ensemble model. Figure 6.27 depicts the empirical findings in terms of diagram of the interpretability of the GB model used in this research.

In terms of model performance outside the experimental dataset, this model is not guaranteed to perform optimally in different environmental setting different from the experimental dataset. As such, more studies are needed for more investigation and better conclusion. Furthermore, from the evaluation result shown in Table 6.8, it was observed

Previous Study Comparison					
Research name	Variables considered	ML method used	Accuracy result	Interpretability considered	Study Venues
Jiang et al. [2016]	Indoor CO_2	Feature-scaled extreme ML	94%	No	One Venue
This research	CO_2, temperature, humidity, pressure, noise	GB	82%	Yes	Two test centres and one real world office

Table 6.9: Result Comparison with Previous Study

that the four participants were able to predict the results of the model correctly for about 65% of the time please see Table 6.8, making the interpretability of the GB model above average.

Also, according to all the participants, it was observed that the SHAP interpretable method is easier to understand better than the LIME. In conclusion, the GB ensemble tree model used in this research has provided the opportunity for interpretability of its model behaviour by the use of the suitable model explainer (SHAPley values) best know for fairly interpreting ensemble tree models. The proposed approach in this research was compared in Table 6.9 so as to show a clear difference and the importance of this method. In the Table 6.9, it could be observed that though our methods defer, there is a clear improvement with our method from the current method. This approach could increase the credibility of the GB model used and could be applied to more real-world practical application for further validations.

7

Study Three: Dashboard Design for Room Monitoring

Chapter 5 has shown that how indoor CO_2 concentration levels can be predicted accurately with LSTM time series model. Also Chapter 6 shows that room occupancy of different rooms can be better estimated using GB ensemble ML model with approximately 82% accuracy. Therefore, in this chapter we will explore a type of system that would allow room managers to monitor their various rooms and be able to make informed decisions that will ensure occupants health and well-being. This chapter presents the design approach and the methods taken, and the resulting dashboard interface.

7.1 Problem Statement

Residential landlords, estate managers and letting agents are faced with new rules and regulations such as Homes act 2018 (fitness for habitation) and minimum energy efficiency act Kivimaa and Martiskainen [2018]. This means that the act was designed to ensure that all properties privately rented by tenants either for domestic or commercial purpose was free from any health hazards that could risk the safety and health of tenants/employees/room occupants. In addition to the act, those properties are expected to reach a minimum EPC (Energy Performance Certificate) rating of E in terms of energy performance before the landlords are granted tenancy renewal.

Some environmental factors that are known to hinder the required standard of these indoor places are certain levels of indoor temperature, CO_2 and humidity. Therefore, the means for overcoming part of this problem lies on 1) the adequate and timely supply of HVAC into the room when the current environmental condition is below the required level and 2) been able to know which of the environmental variable is responsible for the problem and ensuring that there is a permanent solution to the problem. This approach will prevent damp and mould from building up on the walls. Therefore, it has become necessary for a development of a simple application that helps room occupants to know when these values are below the required standard by a means of data visualization, record keeping and report generation.

7.2 Aims

The aim of this dashboard design is to design a user-centred dashboard that would be used to monitor our indoor environment with the aid of the dashboard interview findings from the participants. The interview themes will be converted into a prototype system using a photographic drawing tool known as PhotoScape X team [2019], a free online image design tool. This dashboard will be called O-EMA. The following sections will list the software requirements for this dashboard application designed in this research.

7.3 User Requirements Interviews

In order to ensure a user-friendly centred system a semi-structured interview was conducted before the design of this system. An overview of the qualitative analysis of a group semi-structured interview conducted for the dashboard design is presented in this thesis. The participants were two (n=2) staff from SwarmOnline Glasgow; financial technology company. The interviews took place on 27/02/2020. Please see Table 6 for the interview schedule. The interview data was transcribed via oTranscribe software (free HTML App) after which it was analysed using content analysis method discussed in Section 7.3.1.2. The voice recording obtained were managed in line with the principle of University of Strathclyde's general data protection regulation (GDPR). Before the qualitative analysis begun, the transcribed data was anonymised to prevent revealing the identity of the participants.

In order to provide solution to some of the interview questions that the participants were asked, the interviews were examined based on what the participants reported about how helpful the dashboard would be to their company (if they have about 50 staff) or to other companies for reasons such as 1) making their indoor environment healthy and fit for purpose, 2) curtailing indoor environmental conditions of their work place and 3) the benefits they could get if they have the report of the actual statistics of different environmental variables been measured. Moreover, this approach helped in the concluding and formulating software requirement specifications (SRS) of the dashboard being designed and contributed in answering the research question which states the following: By designing the dashboard system for room management:

- How will the dashboard be designed so that it can help in visualizing the levels of indoor environmental variables, the predicted indoor CO_2 levels and the estimated room occupancy.
- What software design will help to make the usability of the system easy.
- How will the stakeholders' contributions be identified and incorporated in the design phase of the system.
- What are the functional requirements that will help the system users to perform their task effectively.
- What software tool will be used to design the system.
- What software tool will be used to test and evaluate the software design.

- How will the designed system be evaluated in order to determine if the system can perform its required tasks when developed.

7.3.1 Methods

7.3.1.1 Study Participants

The characteristics of the participants that took part in the qualitative study of this research are presented as follows:

- Two people of the same workforce and from the same company were interviewed using semi-structured interview format.
- They were asked to mention the importance of environmental room monitoring system to their organization.
- They were asked what variables they could be particularly interested in viewing.
- They were asked what design/format do they want to view the variables' readings on the system.
- They were asked if they have any functional requirements or system features in mind that they would want the system to have.
- They were asked of any colour scheme or push notification that they would want the system to have.

The experts who participated are described in Table 7.1

ID	Profession	Sex
P1	Financial company staff	F
P2	Financial company staff	M

Table 7.1: Table showing software development interview participants and their domain expert

7.3.1.2 Content Analysis Methods Used for Generating User Requirements

Qualitative research such as semi-structured interviews are used to explore various aspect of research study especially when the research is intended for real-world application. Though, there are some interviews processes that are easy and straightforward but analysing the transcript for meaningful understanding is often a problem for interviewers Burnard [1991]. One of the methods for analysing interview data is thematic content analysis by Glaser et al. [1968] which basically uses "grounded theory" approach as a way of arriving at theory meant for its uses. Thus grounded theory approach means a strategy for manipulating theoretical data by providing ways of conceptualization and explanations.

The theory is expected to be made clear with as many categories and hypothesis as possible for future and current research verification Glaser and Strauss [2017]. Glaser and Strauss proposes that the data analyst should first systematically discover the theory in the data by a means of data examination in order to understand the underlying meaning of the data. Burnard observed that the thematic content analysis is suitable for semi-structured, open-ended interviews that have been recorded in full and transcribed. The theory to be developed from the qualitative research has to have great description with data synthesis and abstraction in order for the reader to understand Morse and Field [1995].

According to Morse and Field, types of theory are 1) deductive theory; a method of deducing unknown truths from the already known concept thereby drawing logical inference to the established research standard and 2) evaluating theory; evaluation of theories involves some characteristics such as extensiveness, validity, consistency and usefulness. These characteristics when fulfilled, the theory is assumed to be tested. Burnard suggested fourteen stages of interview analysis as follows:

- The researcher should write short notes either during the course of the research or after the interview about the main topics or ideas to be discussed. The researcher can revisit these notes during the qualitative data analysis which can help a lot during categorization of the interview data.

- After transcribing the interview recordings, the researcher should first read through the transcribed data and make important notes at the same time. This approach will give the researcher an opportunity interpret the data in the respondents' real-life perspective Rogers [2012].

- Further reading of the transcript is done by the researcher. At this stage, several names of headings and categories are written down and the process in known as open coding scheme [Lune and Berg, 2016].

- The various categories obtained from the transcripts are disintegrated similar categories into higher categories in order to reduce the total number of categories already obtained.

- New categories and subheadings are reviewed and duplicate removed before the final list of categories.

- Other investigator(s) is invited at this stage to independently read the transcript and to personally categorize the interview data from their perspective. At the end their views are considered and the initial researchers' categories validated. This approach will reduce bias during data analysis.

- Transcripts are further read at the same time with the final list of categories in order to make adjustment where it is needed.

- The transcript and all categories and sub categories are differentiated from one another via coding. Pen highlighters are used at this stage for proper identification.

- After coding and highlighting of the transcripts, several duplicates of the transcripts are made in order to cut out the body of the transcript with its highlighted code. Before this stage a complete interview transcript is duplicated and kept for future reference purposes.

- A file or folder is created in order to paste the cut out sections of the coded transcript.

- At this point, it might be necessary to go back to the respondents to show them what the final interview analysis looks like. They are allowed to make their inputs and corrections if there is need for that. This stage validates the researchers categories.

- This stage is called the writing up stage, where all sections are joined together and referenced during the final writing. The interview recording and transcribed interview data are often referred back to for more understanding in the case of confusion.

- At this stage writings are categorized into sections. The direct word from the respondent are linked to each example. Referring back to the original tape recording is also allowed and practised at this stage.

- At this final stage, uncodable transcripts are discarded while the coded ones are used. The researcher will also decide if the raw verbatim text from the respondent should be used in the writing up stage or a summarized version of the answers from the respondents.

7.3.1.3 Data Collection and Research Ethics

Before the commencement of this second qualitative study, the departmental approved ethics' application with the number **Application ID: 951 and 516** and the interview information sheet is shown in Section J. The interview was done in the participants' office at SwarnOnline Ltd limited before the O-EMA design component of this study and was conducted a week after the confirmatory room occupancy study and one month before the environmental monitoring system design. The semi-structured interview questions contains seven open-ended questions described in Section H. The open-ended questions asked the respondents whether they think the software could be of benefit to their office, what features would they want the features to have if they are to have one, what kind of design would they wish that the system have assuming they are the company interested in the dashboard, the likely variable that could be more important to them if they are to make a choice and the functional requirement will they want to system to have if they are to use it.

The interview was a face-to-face group interview that lasted for a total of 22 minutes 52 seconds. All questions that the participants were asked are shown in Section H. Their answers were recorded using windows voice recorder and were later transferred to the university H-drive for security purposes. The topic guide for the interview was developed my the researcher and some minor adjustments were made by the chief investigator. All the interviews were done and transcribed. The data body resulted into 5 pages of interview transcript.

The full consent of the participants (n=2) were sought and obtained before the interview date. They also permitted for their voices to be recorded during the interview. Before starting the commencement of the interview, the interview information sheet was sent to them via email in order to sensitized the participants on the context of the interview, what their data would be used for and how their data will be handled during and after the interview. The interview data were anonymized to remove identifying information.

7.4 Data Analysis

The same content analysis method used to analyse the first part of the interview discussed in Section 6.2.3 was used to analyse this second interview. Firstly, the transcribed interview data were read multiple times so as to understand all the interview response from the participants point of view. After which the text in the interview data were divided into important components while retaining the main meaning of the interview response.

Another step that was taken was that the final component units were labelled with codes and then the codes were grouped into categories so as to discover important information and pattern contained therein. In this analysis, categories were the highest level of abstraction. Additionally, there was an element of hybrid coding as it was done in the previous qualitative analysis Section 6.2.3, such as inductive and deductive style were also combined during the qualitative data analysis. The main categories for the code system that emerged from the questions they were asked and what was intended to find out about the O-EMA design are (i) Who might benefit from O-EMA (ii) benefits of O-EMA to landlords, tenants etc and (iii) the functional and non-functional requirement of the indoor environmental monitoring software system.

7.4.1 Themes

There are 13 main categories and 28 subcategories (explanations) that were discovered in this interview data as shown in Section I. These categories are 1) Beneficiaries of O-EMA, 2) Benefits of the O-EMA to landlords, 3) O-EMA impact in the control of moulds and damps in a building, 4) O-EMA potential in provision of energy efficient and good working environment, 5) Push notifications or warning as a functional requirement, 6) Login as a functional requirement, 7) Report generation as a Functional requirement of O-EMA, 8) Design pattern as a functional requirement of O-EMA, 9) Environmental variables current levels as a Functional requirements of O-EMA, 10) Traffic light colours as different indicators, 11) Colour scheme as non-functional requirement, 12) Non-functional requirements 13) Platform and operating system.

The benefits mentioned by the respondents are mostly the known benefits of O-EMA in work places. The interview data analysis contributed to the formulation of research question such as **1) What software design and the functional requirements will make the usability of the system easy.** and **2) How will the stakeholders' contributions be identified and incorporated in the design phase of the dashboard.** The answers mentioned by the two participants can be summarized as follows:

- Most companies with large number of employees that share a common office will benefit more from the use of O-EMA. Landlords who have the obligation of making sure that the rooms they rent to tenants are about standards set by the government are likely beneficiaries of O-EMA.
- O-EMA can assist room occupants in the control of sick building syndrome often caused by moulds and damps thereby ensuring that the indoor environment is healthy.
- O-EMA can reduce the quantity of energy been used by the building by ensuring that the heating or cooling are done when it is needed.
- When designing the O-EMA dashboard the following should be considered as its functional and non functional requirements:
  - The O-EMA design should be simple with all the rooms interconnected as a centralized unit system with each other.
  - The main colour of the O-EMA background should be dark, while the warning colours for hotspot rooms should reflect that of the traffic lights at different scenarios.
  - The fonts of the O-EMA should be meaningful and readable. The pages should be easy to navigate.
  - There should be single login.
  - The O-EMA dashboard should have a graphical interface showing the pattern of past variable readings.
  - The O-EMA should show readings of Humidity and CO_2.

7.4.2 Findings

There are predominant benefits contained in the interview data. They are broken down into categories and subcategories as follows:

- **Beneficiaries of O-EMA:** Both P1 and P2 agrees that O-EMA will be beneficial to their company if they have more employees making use of an open office space in their company. They however believed that the O-EMA will more likely be beneficial to landlords and big companies with larger businesses.

  > *P2: Talking of Swarm, our size I think is probably of limited .. for constantly to always be on dashboard however I can see the benefit , I have two scenario on my head. One is for larger business who have bigger office and the second is for landlords.*
  >
  > *P1: Yeah Perhaps not for us.*
  >
  > *P2: ... That's said if it is something for us to have easy access..for us to do...at no cost then that will be useful for us as well.*

> *P2: ... Also it will be good where you have about 50/60 people around an office that's where I can see the use case of this.*

- **Benefits of the O-EMA to landlords:** P2 agrees that landlords could use O-EMA to manage their houses, offices or rooms they build in accordance with the legislation.

> *P2: ...So if you imagine landlord of this building who will want to monitor all these things especially for new legislation coming in for landlords.*

- **O-EMA impact in the control of moulds and damps in a building:** P1 and P2 also believed that if the software system could generate weekly report of these indoor variables, it will be used to identify rooms that are prone to cause sick building syndromes which will ensure that the owners acts when the need arises.

> *P1: The historic level, maybe a period of a week and being able to generate a report from there.*

> *P2: .... there is a wall damp ... as we speak it has been repaired but not has been painted, before we moved in that was bad damp. I think it goes back to the landlord discussion again.*

> *P2: The report will be good. Because again it goes back to these companies' landlord...*

> *P2: .... for us to have a meeting with the report 15:31 saying here is the reading of the last month... or whatever that will be...*

> *P2: Think about a sort of report, imagine we have a system,.. maybe something in the building that we can't but we could give that report to a landlord and say look there is a problem here and you need to fix this two problems 16:10 as a landlord and here is the report that proves what the last three months ... or he could take it to finance to sign off the money for the work we have done if we do the work by ourselves.*

- **O-EMA potential in provision of energy efficient and good working environment:** P2 believes that if landlords could have access to O-EMA that it will help them in ensuring that their properties are providing energy efficient and good working environments for their tenants.

> *P2:landlords are getting more legislation around being energy efficient and more focus on providing good working environment for employees.*

- **Push notifications or warning as a functional requirement:** P2 agrees that in addition to the report, the O-EMA should be able to send a warning to the user via email of any potential problem.

> *P1: if you could get an email to go and turn the report.*

> *P2: I think having something that could set the threshold or you could brainstorm your findings and find out to say that the CO_2 with this colour... and you could send an email*

- **Login as a functional requirement:**P2 believes that single login will be enough for O-EMA except if there are important security task that the software will be performing.

> *P2: I think maybe ...logins if people will use it, but I think if you don't have to install stuff single account login ...*

- **Report generation as a Functional requirements of O-EMA:** P1 and P2 agrees that the O-EMA should be able to generate report and a graph of the environmental variables either weekly or monthly.

> *P1: ...the historic level, maybe a period of a week and being able to generate a report from there.*
>
> *P2: The report will be good...In summary of the dashboard there will be a graph and that will be quite interesting.*
>
> *P2: Think about a sort of report, imagine we have a system,.. maybe something in the building that we can't but we could give that report to a landlord and say look there is a problem here and you need to fix this two problems as a landlord and here is the report that proves what the last three months... or he could take it to finance to sign off the money for the work we have done if we did the work by ourselves*

- **Design pattern as a Functional requirements of O-EMA:** P1 and P2 agrees that O-EMA should have a map-like design where all the rooms in a floor are connected to one page with different situations of each room showing on that front page.

> *P2: So I think having a dashboard such as this will be cool...We gone out to draw the four corners of our office and that is one of our hot spots*
>
> *P2: ...You could show there are six people sitting in that room.....and it is a visual representation.*
>
> *P1: again in the view of the office plan....*

- **Environmental variables current levels as a Functional requirements of O-EMA:** P1 agrees that the O-EMA should be able to show in real-time the current levels of each variables it measures. P2 agrees that CO_2 levels is a very important feature for O-EMA to have because it will help them to check the productivity levels of their employees.

> *P1: the current levels as well.*

> *P2: For example we could say that certain levels of CO_2 we use it to identify the productivity depth assume there will be further studies. ... you can pop that up and say that room there the development room , the CO_2 there has been high and as a management team we can go actually, .. maybe people are complaining of a lot of headaches or people not productive or whatever...that's the kind of stuff I think...*

- **Traffic light colours as different indicators:** P1 and P2 believes that the warning signs on the dashboard should be indicated with the kind of traffic colours used by the driver and vehicle licensing agency (DVLA).

> *P1: We need a kind of values with traffic light system colours , you know what is too high, too low..*

> *P2: you imagine that there is a TV in the wall that room goes amber ..you know that there is something to deal with that, that's the kind of thing in my head ..*

- **Colour scheme as non-functional requirement:** P1 and P2 agrees that the colour of the dashboard is not really a priority as long as the texts are eligible and easily readable.

> *P1: but as developers we go for darker I struggle to understand the reason.*

> *P2: I don't think colour is such a big deal...Just any colour that can show.. I think you should keep it simple, keep it light... and simple read*

- **Non-functional requirement:** P2 agrees that the dashboard should be easy to use and navigate with sensible meaning for words and adequate font size.

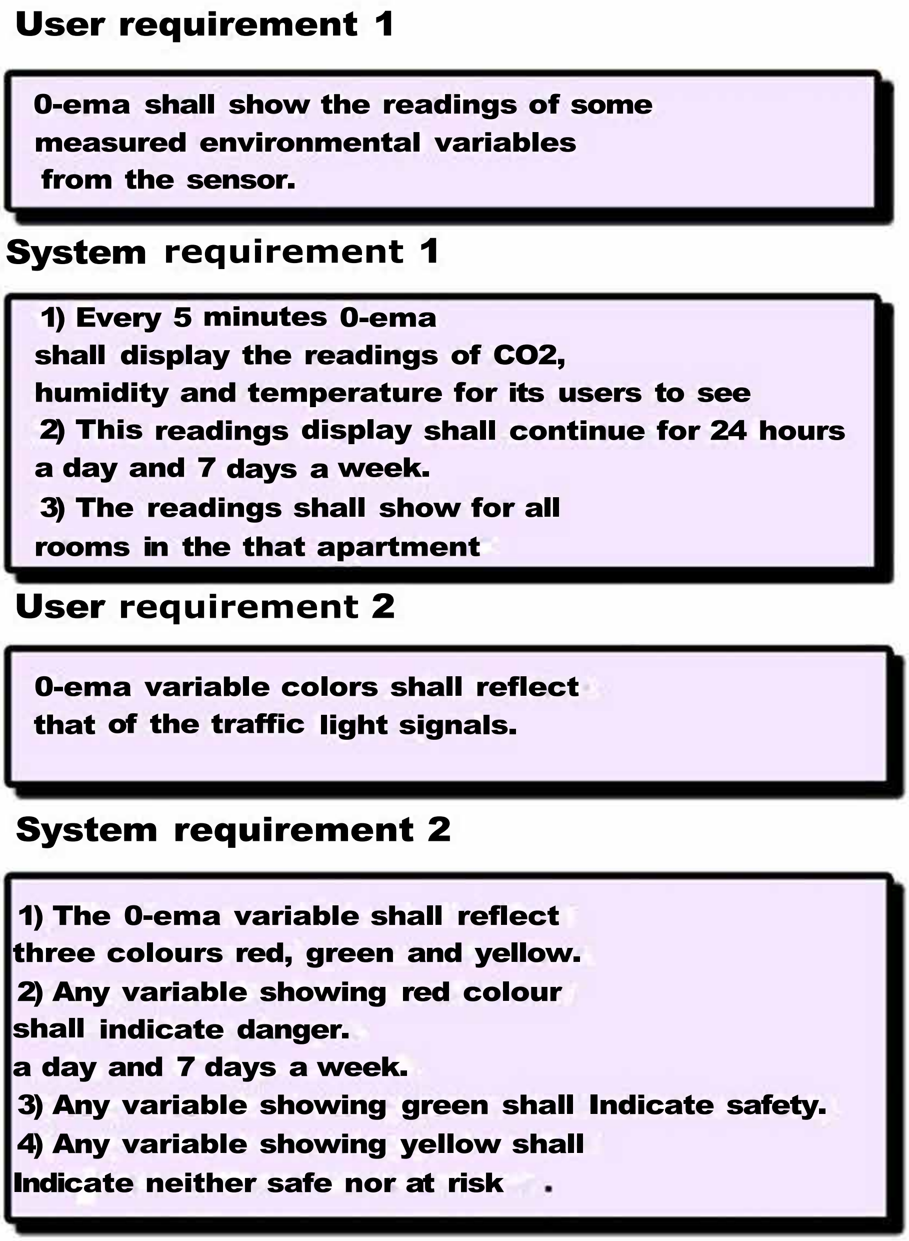
> *P2: maybe we should be looking at non functional requirement which should be easily readable, easy to navigate, words should be as possible sensible, thinks about the font sizes ... you know colour contrast, you are just playing colours ...*

- **Platform and operating system:** P2 believes that any operating system and platform is okay but that O-EMA should be designed to work on iPad.

> *P2: if do those dashboard to have something that is responsive that could work on iPad and even some web app you can load up an iPad....we have built dashboard application in the past, it worked on iPad but not on the phone but it was displaying a graph.*

7.4.3 User-system Requirement

In the table below, the term O-EMA was used as the name of the dashboard that was designed in this research.



User requirement 1

0-ema shall show the readings of some
measured environmental variables
from the sensor.

System requirement 1

1) Every 5 minutes 0-ema
shall display the readings of CO2,
humidity and temperature for its users to see
2) This readings display shall continue for 24 hours
a day and 7 days a week.
3) The readings shall show for all
rooms in the that apartment

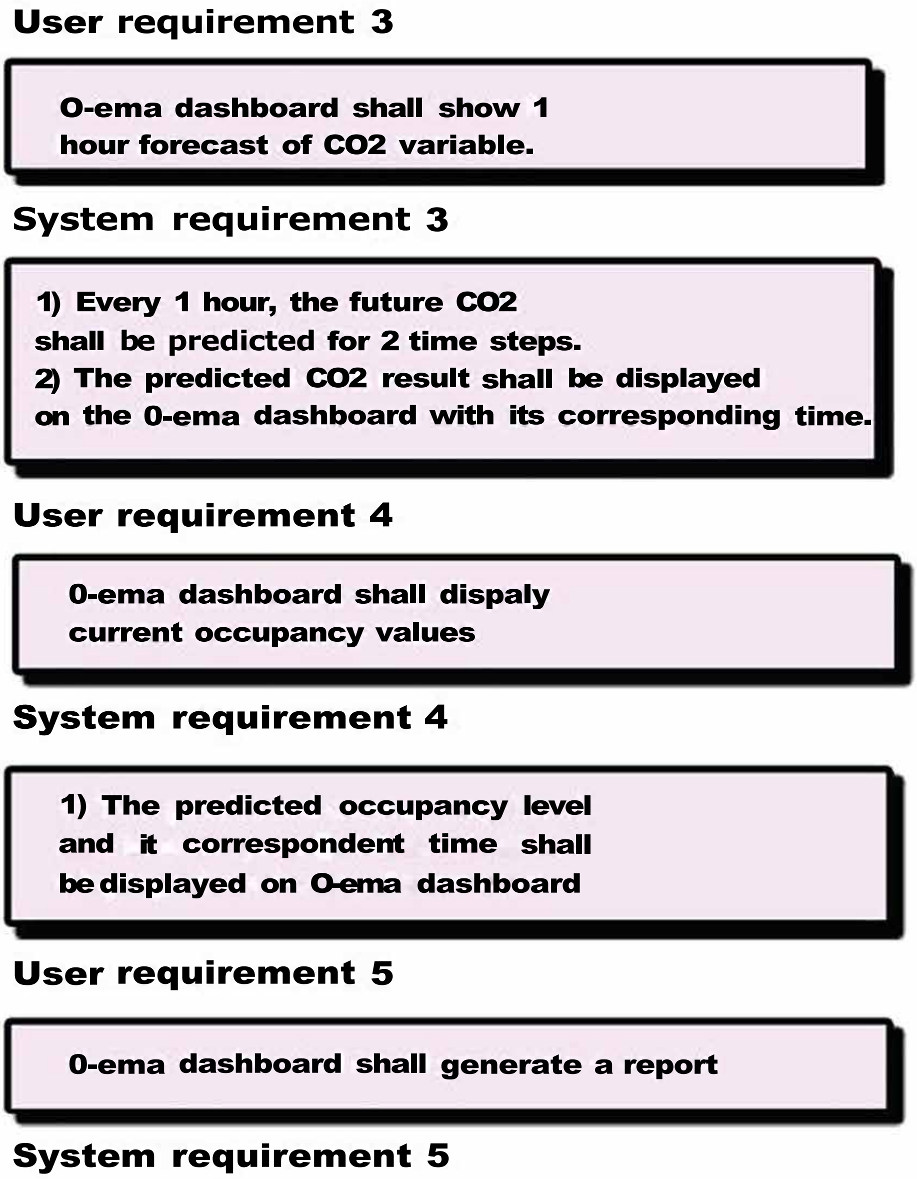
User requirement 2

0-ema variable colors shall reflect
that of the traffic light signals.

System requirement 2

1) The 0-ema variable shall reflect
three colours red, green and yellow.
2) Any variable showing red colour
shall indicate danger.
a day and 7 days a week.
3) Any variable showing green shall Indicate safety.
4) Any variable showing yellow shall
Indicate neither safe nor at risk .

Table 7.3: User-system requirement for O-EMA



User requirement 3

O-ema dashboard shall show 1
hour forecast of CO2 variable.

System requirement 3

1) Every 1 hour, the future CO2
shall be predicted for 2 time steps.
2) The predicted CO2 result shall be displayed
on the 0-ema dashboard with its corresponding time.

User requirement 4

0-ema dashboard shall dispaly
current occupancy values

System requirement 4

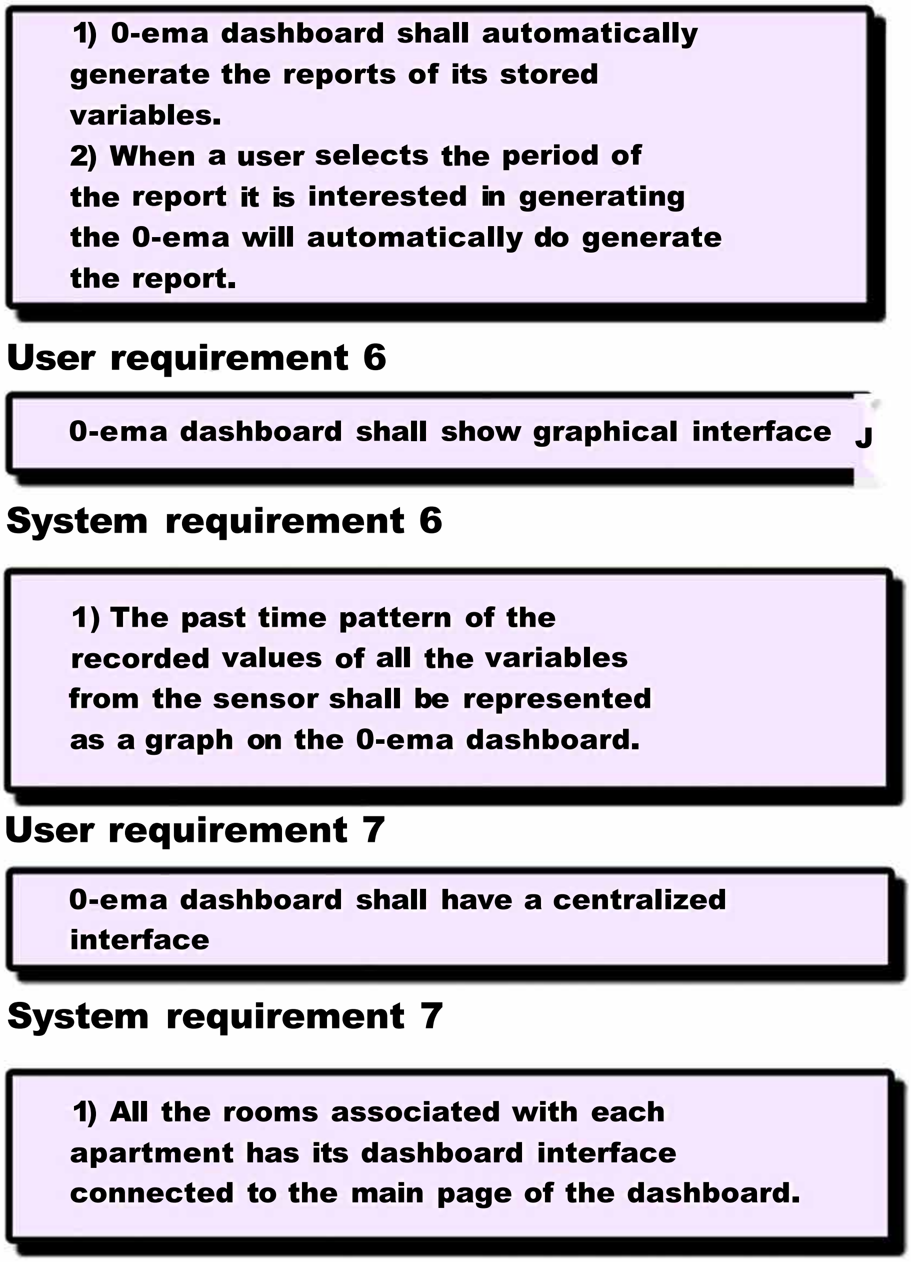
1) The predicted occupancy level
and it correspondent time shall
be displayed on O-ema dashboard

User requirement 5

0-ema dashboard shall generate a report

System requirement 5

Table 7.4: User-system requirement for O-EMA



1) 0-ema dashboard shall automatically generate the reports of its stored variables.

2) When a user selects the period of the report it is interested in generating the 0-ema will automatically do generate the report.

User requirement 6

0-ema dashboard shall show graphical interface

System requirement 6

1) The past time pattern of the recorded values of all the variables from the sensor shall be represented as a graph on the 0-ema dashboard.

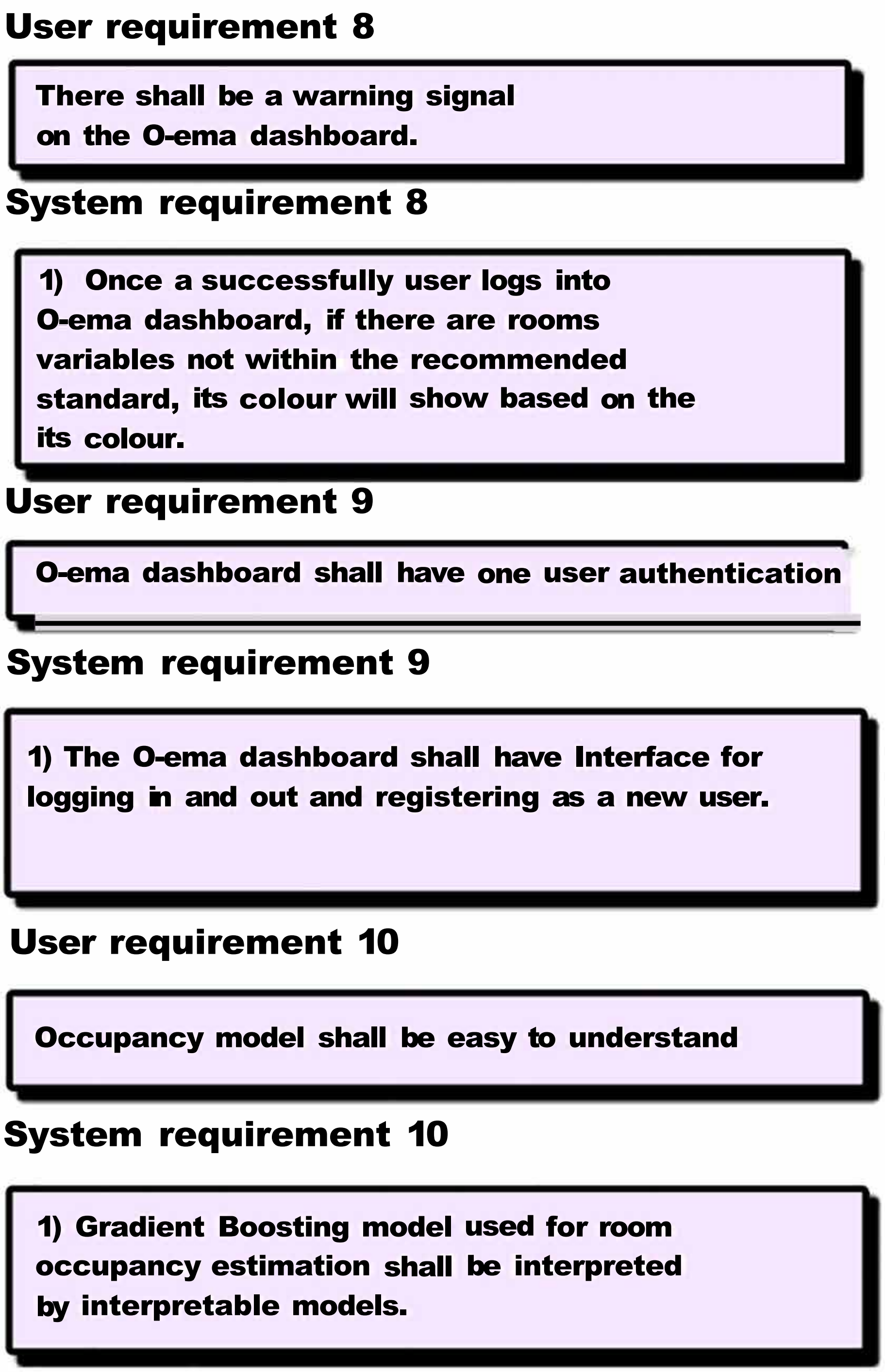
User requirement 7

0-ema dashboard shall have a centralized interface

System requirement 7

1) All the rooms associated with each apartment has its dashboard interface connected to the main page of the dashboard.

Table 7.5: User-system requirement for O-EMA



User requirement 8

There shall be a warning signal
on the O-ema dashboard.

System requirement 8

1) Once a successfully user logs into
O-ema dashboard, if there are rooms
variables not within the recommended
standard, its colour will show based on the
its colour.

User requirement 9

O-ema dashboard shall have one user authentication

System requirement 9

1) The O-ema dashboard shall have Interface for
logging in and out and registering as a new user.

User requirement 10

Occupancy model shall be easy to understand

System requirement 10

1) Gradient Boosting model used for room
occupancy estimation shall be interpreted
by interpretable models.

Table 7.6: User-system requirement for O-EMA

Occupancy Study Interview		
Participants	Participants Interests	Considered Interests
P1, P2, P3, P4, P5	CO_2 values	Considered
P1, P2, P3, P4, P5	Occupancy levels	Considered
P3	Interpretability of the occupancy model	Considered
P1, P2, P3, P4, P5	Analytic result	Considered
Dashboard Requirement Interview		
P1, P2	Report generation	Considered
P2	Variable's level of risk	Considered
P2	Single Login	Considered
P1, P2	Centralized system	Considered
P1, P2	Traffic colours for warning sign	Considered
P1	Real-time system	Considered
P1, P2	Background colour	Considered
P2	Operating system	Not Considered

Table 7.2: Table showing system requirements as they were suggested by the participants and the selected system requirements

The result summary and the prioritization can be seen in Table 7.2 and Section 7.4.3 respectively. In the summary table, there are two sections; occupancy study interview section (for the summary of the interview results conducted during occupancy and CO_2) and dashboard requirement interview section (for the summary of the dashboard interview). From the Table 7.2, the rows marked "Considered" were all included during the research study analysis and results formulation. The row marked "Not Considered" were not considered because the dashboard design has not been developed. But this feature certainly will be considered if the dashboard design will go through the development stage.

This research implemented 91.7% of the requirements suggested by the interview participants. Doing this step helped in formulating all the research questions necessary to conduct this research. Furthermore, some of the user requirements that were eventually converted into system requirements are shown in Section 7.4.3-4.4. The table shows 10 user requirements with its corresponding 10 system requirements which mainly shows the most important functional requirements considered in this research. More explanation of these functional requirements are discussed in Section E.1. The dashboard implementation is fully discussed in Section F.

7.5 Testing and Evaluation

Section F has discussed how a high fidelity prototype dashboard known as O-EMA, for monitoring indoor environment was designed in order to help occupants to stay healthy in their various indoor environment. This section will then test and evaluate the implemented O-EMA dashboard design in order to assess its feasibility of doing the work it was meant to do. In this implementation and testing process, people who were interviewed before the dashboard was designed and new set of participants who are interested in the dashboard design took part in this study. The participants will be engaged in a series of structured task by using the prototype system via A free online web application for testing a high fidelity App. (InVision) team [2020].

7.5.1 Data Collection and Evaluation

The dashboard testing was conducted in two phases with six participants. First was an interactive walk-through of the system with four participants via Zoom video conferencing and then a follow up survey. The second was through a follow up survey with the two staffs of the SwarnOnline Ltd who already took part in the initial dashboard design interview. Four postgraduate students studying computer science degree at University of Strathclyde and University of Glasgow. Three of the students are in the computer and information science (CIS) department at the University of Strathclyde. One is in the University of Glasgow. The other two are staffs of SwarnOnline Ltd. The video interviews took place on the 22/06/2020, 23/06/2020, 25/06/2020, 26/06/2020 24/08/2020 and 29/08/2020. Please see Table 7 for participants interview and survey appointment schedules. The approved ethics application number by the CIS department for the dashboard evaluation is **Application ID: 1172**. The profile of the people who participated in the dashboard evaluation video interview are stated in Table 7.7. The corresponding time taken by each participants during the interview is a combined time for the model and dashboard evaluation.

The interactive prototype was designed in order to answer the research question which states that: **"By designing a dashboard system for room management, How will the designed system be evaluated in order to determine if the system can perform its required tasks when developed".**

Before the commencement of the video interview and survey, participants were sent the dashboard user guide shown in Section K and the link to the interactive fidelity prototype (please see: O-EMA App) was first sent to the participants. This was done for easy understanding of the expected functions of the dashboard by the participants, thereby making the interview and survey process easy and fast. Furthermore, the questionnaire for the dashboard evaluation was also sent to the participants which was completed after the interview. Please see is Section L which is the questionnaire.

ID	Occupation	Sex	Time taken
E1	UoG Student	F	60 minutes
E2	CIS Student	F	45 minutes
E3	CIS Student	F	34 minutes
E4	CIS Student	F	67 minutes
E5	SwarnOnline Ltd staff	F	N/A
E6	SwarnOnline Ltd staff	M	N/A

Table 7.7: Table showing interview participants for dashboard evaluation and their occupation.

During the dashboard evaluation interview, there are steps that were followed by this study for better evaluation of the dashboard by all the four participants. The steps were as follows:

- The participant dashboard evaluation user guide was first explained to the participants.
- After the user guide explanation, the link to the prototype dashboard was opened on the interviewer's laptop and shown to the participants via screen sharing.
- All the stated functions in the dashboard user guide was then walk-through step by step together with the participants for easy understanding.
- At the end of the interview, the participants were then asked to fill out the dashboard evaluation questionnaire sheet based on their experience of the prototype system.
- All the steps were repeated for all the four participants.

7.5.2 Evaluation

There were 6 key themes that emerged after the interview analysis. These themes were as follows:

- Design.
- Appearance.
- Clarity.
- Meaningful names.
- Usefulness.
- Risk prevention.

The grading of this dashboard were divided into two. The first nine questions (Q1-Q9) used grading metrics of 3, where 1 represents some what easy to understand, 2 represents easy to understand and 3 represents very easy to understand. The Q10-Q11 used grading metrics of 5 where 1 represents least likely, 2 represents unlikely, 3 represents neither, 4 represents likely and 5 represents very likely. During the grading of the participants' answers according to the above stated categories, five of the participants agreed that the design, appearance, meaningfulness and usefulness of the dashboard were very easy to understand by giving it the scale of 3 out of 3. However, some of the participants agreed that more information about the colours of the different levels of the environmental variables and the chart could still be improved on during the development of the application. Furthermore, all the participants agreed that the dashboard is likely going to help occupants to monitor their rooms and thereby preventing any feature health risk.

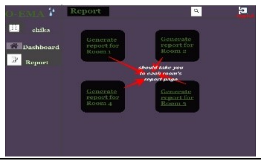
Requirement 1: Centralized Interface

<table>
<tr><th>User requirement</th><th>Corresponding system requirement</th><th>System design
Showing requirement</th></tr>
<tr>
<td>O-ema shall have a centralized user interface.

<u>P2</u>: “So I think having a dashboard such as this will be cool...We gone out to draw the four corners of our office and that is one of our hot spots”

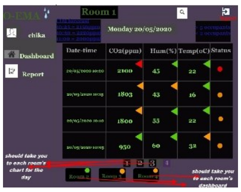
<u>P1</u>: “..again in the view of the office plan....”</td>
<td>All the rooms associated with each apartment has its dashboard interface connected to the main page of the dashboard.</td>
<td>This Image shows that all 0-EMA rooms' Interface are centralized and could be reached from the main page

This image shows that all 0-EMA rooms' interface are centralized and could be reached from the main page


O-EMA
Report
chika
Dashboard
Report
Generate report for Room 1
Generate report for Room 2
should take you to each room's report page
Generate report for Room 4
Generate report for Room 3

</td>
</tr>
<tr><td></td><td><b>Requirement 2: Warning Sign</b></td><td></td></tr>
<tr>
<td>O-ema shall alert its user with a warning via email.

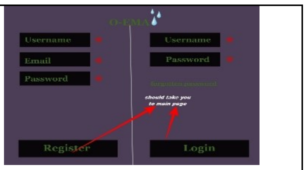
<u>P1</u>: “if you could get an email to go and turn the report.”
<u>P2</u>: “I think having something that could set the threshold or you could brainstorm your findings and find out to say that the CO2 with this colour... and you could send an email”</td>
<td>Once a user receives a warning email of their room situation. They will be able to view what is happening on O-ema dashboard. If there are rooms variables not within the recommended standard, its colour will show based on its colour.</td>
<td>Image showing the status of each room and its warning in traffic light colours



<table>
<tr><th>Date-time</th><th>CO2(ppm)</th><th>Hum(%)</th><th>Temp(oC)</th><th>Status</th></tr>
<tr><td>20/05/2020 10:20</td><td>2100</td><td>45</td><td>22</td><td>●</td></tr>
<tr><td>20/05/2020 10:15</td><td>1803</td><td>43</td><td>16</td><td>●</td></tr>
<tr><td>20/05/2020 10:10</td><td>1800</td><td>55</td><td>22</td><td>●</td></tr>
<tr><td>20/05/2020 10:05</td><td>950</td><td>60</td><td>32</td><td>●</td></tr>
</table></td>
</tr>
</table>

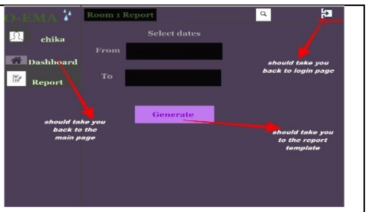
Requirement 3: Single Authentication

Table 7.8: Requirement table and Its Dashboard images

O-ema shall have a single authentication page. P2: “I think maybe ...logins if people will use it, but I think if you don't have to install stuff single account login ...”	The O-ema dashboard shall have interface for logging in and out and registering a new user	 O-EMA Username Email Password Username Password forgotten password should take you to main page Register Login The images of the 0-EMA dashboard sign-in and registration page

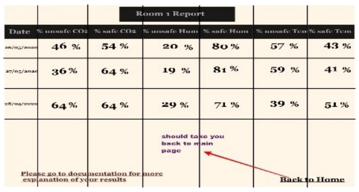
Requirement 4: Report Generation

<table>
<tr>
<td>O-ema shall generate report for its users.

P1: “The historic level, maybe a period of a week and being able to generate a report from there.”
P2: “The report will be good. Because again it goes back to these companies' landlord...”
P2: “.... for us to have a meeting with the report 15:31 saying here is the reading of the last month... or whatever that will be...”
P2: “Think about a sort of report, imagine we have a system.. maybe something in the building that we can't but we could give that report to a landlord and say look there is a problem here and you need to fix this two problems 16:10 as a landlord and here is the report that proves what the last three months ... or he could take it to finance to sign off the money for the work we have done if we do the work by ourselves.”</td>
<td>1) 0-ema dashboard shall automatically generate the reports of its stored variables.
2) When a user selects the period of the report it is interested in generating the 0-ema will automatically do generate the report.</td>
<td>

O-EMA
Room 1 Report
chika
Dashboard
Report
Select dates
From
To
Generate
should take you back to login page
should take you back to the main page
should take you to the report template

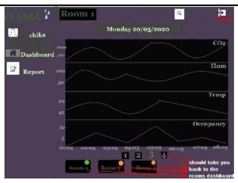
Images of clickable link for 0-EMA report generation


Room 1 Report
<table>
<tr><th>Date</th><th>% unsafe CO2</th><th>% safe CO2</th><th>% unsafe Hum</th><th>% safe Hum</th><th>% unsafe Tem</th><th>% safe Tem</th></tr>
<tr><td>26/05/2020</td><td>46 %</td><td>54 %</td><td>20 %</td><td>80 %</td><td>57 %</td><td>43 %</td></tr>
<tr><td>27/05/2020</td><td>36 %</td><td>64 %</td><td>19 %</td><td>81 %</td><td>59 %</td><td>41 %</td></tr>
<tr><td>28/05/2020</td><td>64 %</td><td>64 %</td><td>29 %</td><td>71 %</td><td>39 %</td><td>51 %</td></tr>
</table>

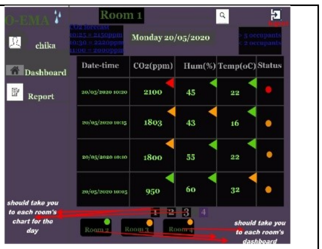
Please go to documentation for more explanation of your results
Back to Home
Image of final report generated by 0-EMA</td>
</tr>
</table>

Requirement 5: Graphical User Interface

Table 7.9: Requirement table and Its Dashboard images

0-ema dashboard shall show graphical interface P2: "The report will be good...In summary of the dashboard there will be a graph and that will be quite interesting".	1) The past time pattern of the recorded values of all the variables from the sensor shall be represented as a graph on the 0-ema dashboard.	 0-EMA Room 1 Monday 20/05/2020 chika Dashboard Report CO2 Hum Temp Occupancy 01:05 02:05 03:05 04:05 05:05 06:05 07:05 08:05 1 2 3 4 Room 2 Room 3 Room 4 should take you back to the rooms dashboard Image of 0-EMA showing graphical Interface for past-time

Requirement 6: Display of Sensor Readings

0-ema shall show the readings of some measured environmental variables from the sensors P2: "For example we could say that certain levels of CO2 we use it to identify the productivity depth assume there will be further studies. ... you can pop that up and say that room there the development room , the CO2 there has been high and as a management team we can go actually, .. maybe people are complaining of a lot of headaches or people not productive or whatever...that's the kind of stuff I think..."	1) Every 5 minutes 0-ema shall display the readings of CO2, humidity and temperature for its users to see 2) This readings display shall continue for 24 hours a day and 7 days a week. 3) The readings shall show for all rooms in the that apartment	 0-EMA Room 1 Monday 20/05/2020 chika Dashboard Report CO2 forecast 10:25 = 2150ppm 10:30 = 2220ppm 11:00 = 2000ppm > 5 occupants < 2 occupants 1 2 3 4 Room 2 Room 3 Room 4 should take you to each room's chart for the day should take you to each room's dashboard Date-time · CO2(ppm) · Hum(%) · Temp(oC) · Status 20/05/2020 10:20 · 2100 · 45 · 22 · ● 20/05/2020 10:15 · 1803 · 43 · 16 · ● 20/05/2020 10:10 · 1800 · 55 · 22 · ● 20/05/2020 10:05 · 950 · 60 · 32 · ● Image showing the display of the CO2, Humidity, Occupancy readings.

Requirement 7: Variable Colours

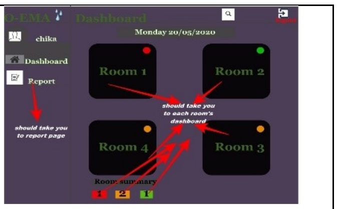
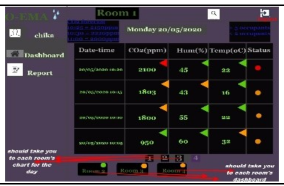
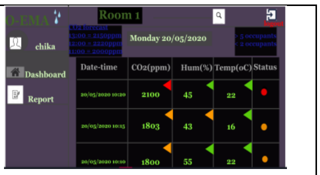
0-ema variable colours shall reflect that of the traffic light signals. P1: "We need a kind of values with traffic light system colours, you know what is too high, too low.." P2: "..you imagine that there is a TV in the wall that room goes amber ..you know that there is something to deal with that,	1) The 0-ema variable shall reflect three colours red, green and yellow. 2) Any variable showing red colour shall indicate danger a day and 7 days a week. 3)Any variable showing green shall Indicate safety. 4) Any variable showing yellow shall Indicate neither safe nor at risk	 0-EMA Dashboard Monday 20/05/2020 chika Dashboard Report Room 1 Room 2 Room 4 Room 3 should take you to each room's dashboard should take you to report page Room summary 1 2 1 O-ema showing the red, green, orange colours from on its main and dashboard page.

Table 7.10: Requirement table and Its Dashboard images

that's the kind of thing in my head ..”		O-EMA Room 1 chika Dashboard Report CO2 forecast 10:25 = 2150ppm 10:30 = 2220ppm 11:00 = 2000ppm Monday 20/05/2020 > 5 occupants < 2 occupants Date-time, CO2(ppm), Hum(%), Temp(oC), Status 20/05/2020 10:20, 2100, 45, 22, ● 20/05/2020 10:15, 1803, 43, 16, ● 20/05/2020 10:10, 1800, 55, 22, ● 20/05/2020 10:05, 950, 60, 32, ● 1 2 3 4 Room 2 Room 3 Room 4 should take you to each room's chart for the day should take you to each room's dashboard

Requirement 8: CO2 Forecast

O-ema dashboard shall show 1 hour forecast of CO2 variable. P1: “..the current levels as well”	1) Every 1 hour, the future CO2 shall be predicted for 2, time steps. 2) The predicted CO2 result shall be displayed on the 0-ema dashboard with its corresponding time	O-EMA Room 1 logout chika Dashboard Report CO2 forecast 13:00 = 2150ppm 12:00 = 2220ppm 11:00 = 2000ppm Monday 20/05/2020 > 5 occupants < 2 occupants Date-time, CO2(ppm), Hum(%), Temp(oC), Status 20/05/2020 10:20, 2100, 45, 22, ● 20/05/2020 10:15, 1803, 43, 16, ● 20/05/2020 10:10, 1800, 55, 22, ● O-ema showing the results of the CO2 forecast on the Room’s dashboard

Requirement 9: Occupancy Estimation display

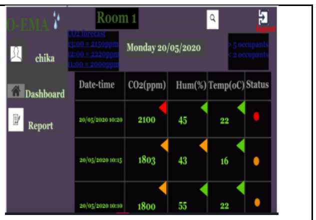
O-ema shall display the results of the estimated Occupancy levels with its corresponding time. P2: ...You could show there are six people sitting in that room.....and it is a visual representation.	1) The predicted occupancy level and its corresponding time shall be displayed on O-ema dashboard	O-EMA Room 1 logout chika Dashboard Report CO2 forecast 13:00 = 2150ppm 12:00 = 2220ppm 11:00 = 2000ppm Monday 20/05/2020 > 5 occupants < 2 occupants Date-time, CO2(ppm), Hum(%), Temp(oC), Status 20/05/2020 10:20, 2100, 45, 22, ● 20/05/2020 10:15, 1803, 43, 16, ● 20/05/2020 10:10, 1800, 55, 22, ● O-ema showing the results of the current occupancy estimate and Its corresponding time on the Room’s dashboard

Table 7.11: Requirement table and Its Dashboard images

Section 7.5.2 shows some very important functional system requirements in verbatim statements by the user and its corresponding physical image as it appears on the prototype dashboard. They constitute the functional requirements stated by the participants who were interviewed before the dashboard was designed. Also, the table shows how the user requirements were implemented in the dashboard design. Among the six participants who took part in the testing and evaluation of the functional requirements of the dashboard shown in Section 7.5.2 were the same participants (E5 and E6) who gave the requirement. They took part in the study via online interview and a follow up survey.

All the participants were asked if they have further comment about the functions of the dashboard that could be improved on as stated in Q12, the following answers were given by six of the participants:

- **E1:** “The prototype looks easy to navigate through the different pages. However, for the main design if the content could be reduced on the room page. For example, the CO_2 forecast, and number of occupants could be on a separate page. Overall, it is a straight forward application to use”.

- **E3:** “Because this is still a prototype, I really cannot say much about the dashboard design because I hoped it’d still be improved on. The concept of using traffic light colours are okay ...”

- **E4:** “The dashboard was simple and easy to navigate. The user experience (UX) might be improved on e.g. color/ background...”.

- **E4:** “Some of the forecasting data jumps out more...”

- **E5:** “Overall the UI looks easy to understand and I like the use of the traffic light colours which would help understand health hazards at a glance. It would perhaps be good to use something similar in the charts so you can tell where the thresholds are for green, amber and red.”

- **E6:**“Think about the colour scheme – purple is usually not a good background colour for a functional web application. You need to design the screens so that the data and navigation is the focus. Consider adding hover overs to acronyms e.g ”ppm“. While a traffic light system is good at indicating status, it’s not clear what I should do with a red/amber status. Consider adding alerts. Consider adding more info in reports so it explains what the report will show before you generate it.”.

Based on the above comment, all the participants greed that more work should be done on the dashboard design to help improve some of its flaws such as User Experience (UX), some colours, organization of its table and charts. Participant E1 suggested that the content in the main page can be reduce in order to give room for more clarity of the dashboard’s function. E6 also suggested that the use of acronym should be accompanied with a meaningful hover element for clarity. E6 also suggested that the purple colour of the dashboard should be changed to a better darker colour such as black and that alert should also be included during the development phase of the dashboard. Therefore above all, it would be fair to conclude that the dashboard when developed with all the suggestions been included, it would be

Participants number	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11
E1	3	3	3	3	3	3	3	3	3	4	5
E2	3	3	3	3	3	3	3	3	3	5	5
E3	3	3	3	3	3	3	3	3	3	5	5
E4	3	3	2	3	3	3	2	3	2	4	4
E5	3	3	2	3	3	3	2	2	2	4	4
E6	2	1	2	2	3	3	3	2	2	4	4

Table 7.12: O-EMA Dashboard Evaluation Results

helpful for its users in terms of preventing any health risk often caused by poor indoor environmental conditions.

7.5.3 Findings

The interview themes obtained from the interview data were six and are represented in form of participants' questionnaire answers as shown in Table 7.12. Participants were asked questions that can be categorized into six. The emerged twelve categories and its corresponding question codes are; 1) design Q1, 2) appearance Q2, Q3, 3) clarity Q4, Q5, Q6, Q9, 4) meaningful Q7), 5) usefulness Q8 and 6) risk prevention Q10, Q11.
Table 5 represents the lists of all the questions that were asked the participants during the O-EMA evaluation study. In Table 7.12, the column named "Participants number" refers to all the participants that took part in the interview. The columns named Q1 to Q11 refers to the corresponding questions that each participants were asked during the dashboard evaluation study. The answers obtained from the participants in column Q1, Q2 and Q3 shows that n=5 out of six of the participants agreed that the design, appearance and clarity of O-EMA is very easy to understand, while one participant agreed that the appearance is some-what easy to understand. In Q7 and Q8, more than half of the participants agreed that there is a potential for the O-EMA to be useful for the management of occupants' indoor environment when developed. In columns Q10 and Q11, more than half of the participants agreed that the O-EMA is likely going to serve as an indoor environmental risk prevention if eventually developed.

7.6 Summary and Recommendations

Constant monitoring of our indoor environmental conditions scientifically, is a big step towards achieving our overall well-being and making our indoor environment healthy. However, most commercially available environmental monitoring systems are faced with some challenges such as 1) high cost, 2) lack of privacy, 3) non-comprehensive in terms of functionality and 4) complexity. As a result, intended users are dissuaded from acquiring it despite its great potentials. Part of the reason is because, it is time consuming and academically challenging to develop what works well for every indoor environment because of differences in weather conditions and expected standards by the legislature of different countries.

Therefore, this research concentrated more on designing a simple system that will be cost effective if implemented. In addition, this system is designed to contain some vital information such as 1) number of room occupants (which can improve energy efficiency) and 2) future CO_2 forecast (that can help room occupants take timely decisions). All these additional, all-in-one functionalities are lacking in the current environmental room occupancy research in the literature.

The design of this system was achieved by firstly conducting a semi-structured interview with some participants who suggested the likely functional and non-functional requirement the indoor environmental monitoring system should have. The interview findings was discussed in Section 7.4.1. and the recommendations of the participants were considered during the design of this O-EMA. Four most important recommendations by the participants were as follows:

1. That the O-EMA should have a main page showing all the rooms and their corresponding hotspots. This can be seen in Figure 4 where there are four rooms in a building and the one room is at risk with its variables and one room is safe with all its variables.

2. The participants suggested that the design should reflect traffic light colours based on the conditions of the room. This proposal was considered with the three traffic light colours showing green (safe), red (danger) and yellow (warning) as can be seen in Figure 5.

3. The participants proposed a dark colour for the dashboard and it was considered.

4. The participants proposed a dashboard that can generate report in order to enable the room occupants, estate managers and landlords to be up to date with the legislature where it is needed. This can be seen in Figure 8 and the sample report is shown in Figure 9.

Furthermore, the O-EMA designed in this research has meaningful and readable text as its non-functional requirement. The system was designed to have single login for any user and it is shown in Figure 1. This system was designed to prevent redundancy of the stored database data in such a way that the database can only store and retrieve a maximum of one year data after which it will be deleted.

In conclusion, from the results obtained from the participants who evaluated the system design, it shows that room occupants could be helped to change the way they live indoors and at the same time help estate or room managers to keep abreast with the government's building regulations. Therefore, an O-EMA that is non intrusive, low cost with additional backend development of ML model that predicts indoor CO_2 and estimates room occupancy has been designed. It is expected that this O-EMA when developed will assist home owners, large business owners and room occupants in making their indoor environment healthy and fit for purpose for any functions that it is expected to be used for.

8

Discussion and Future Work

The objectives of this research were to 1) analyse recorded environmental data from Netatmo sensor using ensemble regression and time series ML models in order to estimate indoor room occupancy in real time and forecast indoor CO_2 levels respectively, 2) use explainable machine learning method to interpret the results of the ensemble ML model so that non experts could be able to understand how the ML model predicted its results and 3) design a prototype (high fidelity) dashboard for indoor environmental monitoring and room occupancy management. Achieving the above mentioned objectives was an academic challenge that required holistic approach. The holistic approach involves, knowing the current methods used and identifying the existing problems in order to set out the actual solution that needed immediate attention.

Part of the parameters that guided this holistic approach of making our indoor environment suitable for any purpose were 1) knowing the users' or stakeholders' thoughts, feelings and work pattern by a way of interview and 2) incorporating these information into the research study design/plan. This ensured a user centred research study that encourages practicability of the proposed research aims and solutions. As a result of this proposed approach, research methodology were set out and followed in order to tackle this problem holistically and to answer the research questions.

In order to understand the underlying research problems, a background study of how our indoor places were affected by some environmental conditions was conducted. These includes 1) IAQ (its benefits and problems), 2) thermal comfort of occupants and 3) room management. It was observed that it was indeed important for occupants to be concious of the type of indoor environmental conditions they spend most of their time either working or living in and take proactive steps to mitigate any circumstances that could affect their health in the long run. Furthermore, background study of various ML methods currently used in this type of research study was conducted in order to guide the researcher during ML method selection for the research data analysis.

This research went further to conduct a scoping literature review in order to supplement the identified information obtained from the already conducted background study. This scoping literature review was done by a way of identifying what was already done by the previous researchers in the same area of this research, how they were done and what still needs to be done. The information obtained from both the literature review and the background study helped in the formulation of part of the research questions and boosted the confidence needed to be convinced that this research study is feasible.

Following the research methodological approach outlined, a qualitative analysis was conducted for room occupancy estimation study and indoor environmental monitoring system (dashboard) design. From the interview analysis result, it can be concluded that 1) knowing the exact number of people in the room could lead to demand driven energy supply, thereby improving energy efficiency, 2) knowing the number of people in the room can help in the planning of the utilization capacity of that room, 3) advance prediction of indoor CO_2 concentration levels can determine when and at what level the HVAC can be supplied to the room and 4) having a dashboard that shows all these results can help landlords, estate managers and room occupants to change the way they live indoors for their overall well-being. These same conclusions were also justified in the previous studies Candanedo and Feldheim [2016], Jiang et al. [2016], Nguyen and Aiello [2013] conducted in room occupancy.

Therefore, from the findings in the results of the qualitative analysis conducted in this research, two different quantitative studies were conducted for indoor CO_2 forecast and room occupancy estimation. Based on the quantitative analysis results, it can be concluded that 1) LSTM time series ML model can accurately forecast in advance about two-hour of the indoor CO_2 concentration levels once the indoor temperature, humidity, pressure and noise values were known, 2) GB ensemble ML model through prediction interval can comfortably estimate approximately 83% of the exact range of number of people in the room once its indoor CO_2, temperature, pressure, noise and humidity values are known, 3) The proposed models performance for CO_2 prediction and room occupancy study is limited to the type of rooms with the same environmental setting as earlier mentioned in the previous study findings. Please see Section 5.6 and Section 6.8 receptively. Typical indoor settings for the optimal performance of the proposed models are room with 1) windows constantly closed, 2) doors constantly open, 3) HAVC constantly on at specified time of the day and 4) rooms with between 10-140 seating capacity.

In addition to the use of GB ensemble ML for the estimation of room occupancy, this research further considered interpreting the results obtained by GB ML model so that non-experts who might be interested in the result could understand the rationale behind certain results predicted by the model and comfortably take decisions on their own without expert guidance. This model interpretability was more achieved and understood by SHAP model values according to its evaluation analysis results obtained from the participants.

Furthermore, the interpretability evaluation was done using human-based metrics in order to assess its local interpretability for single prediction of how the model works. The model evaluation was carried out via a video conferencing interview where four postgraduate, non-data science students participated. It was observed that the interpretability of the GB model was above average with 65% accuracy. The participants also observed that SHAP was more readable and easy to understand than the LIME. This particular evaluation observation contributed to the naming of some part of the title of this thesis.

Obtaining these results was one approach, but making the room occupants and managers to constantly become aware of these results is what makes the difference. As a result, this research went ahead to design an indoor environmental monitoring system (web application) known as O-EMA. This system design was done with the help of the potential stakeholders. They were first asked by a way of interview to make their suggestions on how they think the O-EMA should look like, the main characteristics (functional and non functional requirements) it should posses and any unique function it should have so that any potential user could benefit from it.

Most of the participants contributions were 1) the type of design for the O-EMA, 2) the signals and colour signs it should exhibit for any kind of environmental situation and 3) the availability of its report to its users. All these suggestions and more were ranked according to preference by interview participants and then incorporated into the system design and they formed part of the research questions. O-EMA system design was in form of a high fidelity prototype designed for easy evaluation, interaction and simulation of the design details. Both the system design and the computer based interaction were done with PhotoScape X and InVision software respectively.

Furthermore, the same set of people that evaluated the GB model and those who took part in the dashboard design interview were asked to test and evaluate the O-EMA dashboard. Based on the results of their evaluations, the participants all agreed that indeed the O-EMA design will be valuable to its users in terms of ensuring that the occupants indoor environment was healthy and fit for any purpose it was going to be meant for. Additionally, the participants agreed that the dashboard was very likely going to be easy to operate and simple in terms of design when developed. It was assumed that when O-EMA is developed with all the characteristics suggested by the participants who evaluated its prototype, that some of the existing problems in the literature about complex room monitoring systems would be solved.

The major limitations of this research were in three folds. First was in the area of having enough dataset and test locations for more firmer conclusions. As it can be seen that the GB ensemble model used for the estimation of room occupancy performed better with Swarm unlike with the JA314 because the Swarm was larger in size. Therefore, it would be good to suggest that this problem can be further investigated in more different test locations with dataset that has more frequency and should be recorded for at least six months or one year in order to account more for all the potential factors that might be at play. Such factor could be seasonality trend (holidays that happens yearly in every country) which could disrupt the pattern of the environmental variable readings. The second limitation of this research is the optimal performance of the model in a different room outside the experimental dataset. Therefore more studies is needed for further investigations and better conclusion. The third limitation of this research was lack of been able to know how many minutes it takes for the drop in the environmental variable to reflect the drop in the number of occupants and vice versa. Knowing the time frame between the drop/increase in CO_2 and the number of people in the room could be a game changer in the accuracy rate for room occupancy estimation.

There were four main future work been proposed for the continuation of this research. They were 1) more dataset should be recorded for atleast one year and with regular frequency. This can improve the accuracy of the room occupancy prediction, 2) during this further study, the time interval between when the number of occupants reduces and when it increases with respect to each environmental variables' rise and fall should be thoroughly investigated 3) during this further study, there should be connections to HVAC and room management systems in order to test levels of energy savings, cost savings and improved productivity based on the use of ML systems 4) more rigorous interpretability techniques such as a combination of application and human-based evaluation metrics should be considered. Combining application-based and human-based metrics for the interpretability evaluation of the model will ensure self-assessed interaction between the computed model's output and its user against the core target of the ML model and its stated objectives.

Other future works are 1) the development and testing of the O-EMA system for public use. This will reduce the health related issues encountered in any room with poor IAQ and lack of any means of knowing when these variables were at risk to room occupants. 2) more rigorous testing of the GB model using a blind-testing approach. Blind testing of the model performance will prove the accuracy of the model and remove bias.

The importances of this research output were in two phases. Firstly, to address key issue where commercial sponsor and other potential users were interested in understanding how the room occupancy model works. Secondly, with the design of the dashboard system, room occupants, landlords, estate managers, companies etc.. could be encouraged to ensure that the indoor places they manage are devoid of any potential health risk caused by the indoor environmental variables thereby, ensuring that they abide by the stipulated legislation.

The results achieved in this study has answered the research questions stated in this thesis which re 1) using sensed indoor environmental data, indoor CO_2 can be accurately and reliably predicted with LSTM, 2) using sensed indoor environmental data, room occupancy can be reliably estimated by GB and interpreted by SHAP, 3) users can see potential benefits for a dashboard for managing rooms that is based on interpretable ML methods from environmentally sensed data.

In conclusion, being aware of ones indoor environmental conditions should be encouraged by all. The means of creating this concious awareness should be affordable, simple, fast in terms of response, energy efficient and easily accessible because of its advantages in improving people's health and saving energy. The room managers should take ownership of this type of method and ensure its availability. The room occupants or employees should demand for this from their employers to avoid any kind of illness associated with sick building syndrome symptoms caused by unhealthy work places (sick buildings). Ensuring adequate indoor environmental conditions will improve everyone's well-being, increase revenues for employers and reduce unnecessary cost that is often caused by lack of it.

9

Conclusions

Constant monitoring, visualizing and reviewing our indoor environment is gaining momentum recently. This research is proposing a two way solution approach such as data analytics and visualization via multiple platforms in order to aid in adequate monitoring and awareness. This could serve as a way of curbing the health risks posed by high CO_2 concentration levels when there is poor ventilation.

This research has not only incorporated this two-way approach, but it went further to interpret and evaluate the GB model used for predicting these results (room occupancy). This research considered this approach because of its proven records from the literature of improving environmental awareness and encouraging proactive measures by the room occupants themselves towards their overall health and well-being.

The evaluation results of the studies conducted in this research helped to show that our chosen approach could help in some areas such as 1) model interpretability by non-experts and 2) approximately 83% of room occupancy estimation accuracy. The approaches used in this research were chosen based on the interview answers that were analysed in this research study. Which also contributed in making the design of this research more user-centred design. The combination of our chosen approach is an improvement from the current method been used as shown in the comparison, please see (Table 6.9).

It is hoped that future research work that were recommended in the previous chapter should be considered for greater results to be achieved in the area of room monitoring and management as it relates to cost and efficiency. Doing so will contribute greatly in bridging some of the gaps currently existing in the literature and at the same time make our indoor environment healthy and fit for purpose for all occupants.

Acknowledgments

Firstly, I would like to express my sincere appreciation to my first supervisor Dr Marilyn Lennon for her support, patience and advice throughout my PhD study. Her guidance helped me during the research and writing and structuring of this thesis. I could not have wished to have another supervisor and mentor for my PhD study.

My sincere thanks also goes to my second supervisor who is Dr Richard Bellingham for his support and advice throughout my PhD research. His intuitive suggestions and corrections provided me with the opportunity of conducting a better research.

My heartfelt thanks also goes to the Chief executive officer of SwarmOnline limited who is Andrew Duncan. His financial contribution towards the research grant of my PhD research is invaluable. In addition to the grant, he provided the venue and some of the materials used for the successful conduct of this PhD study. Without this generous act of his, this PhD research would not have been complete.

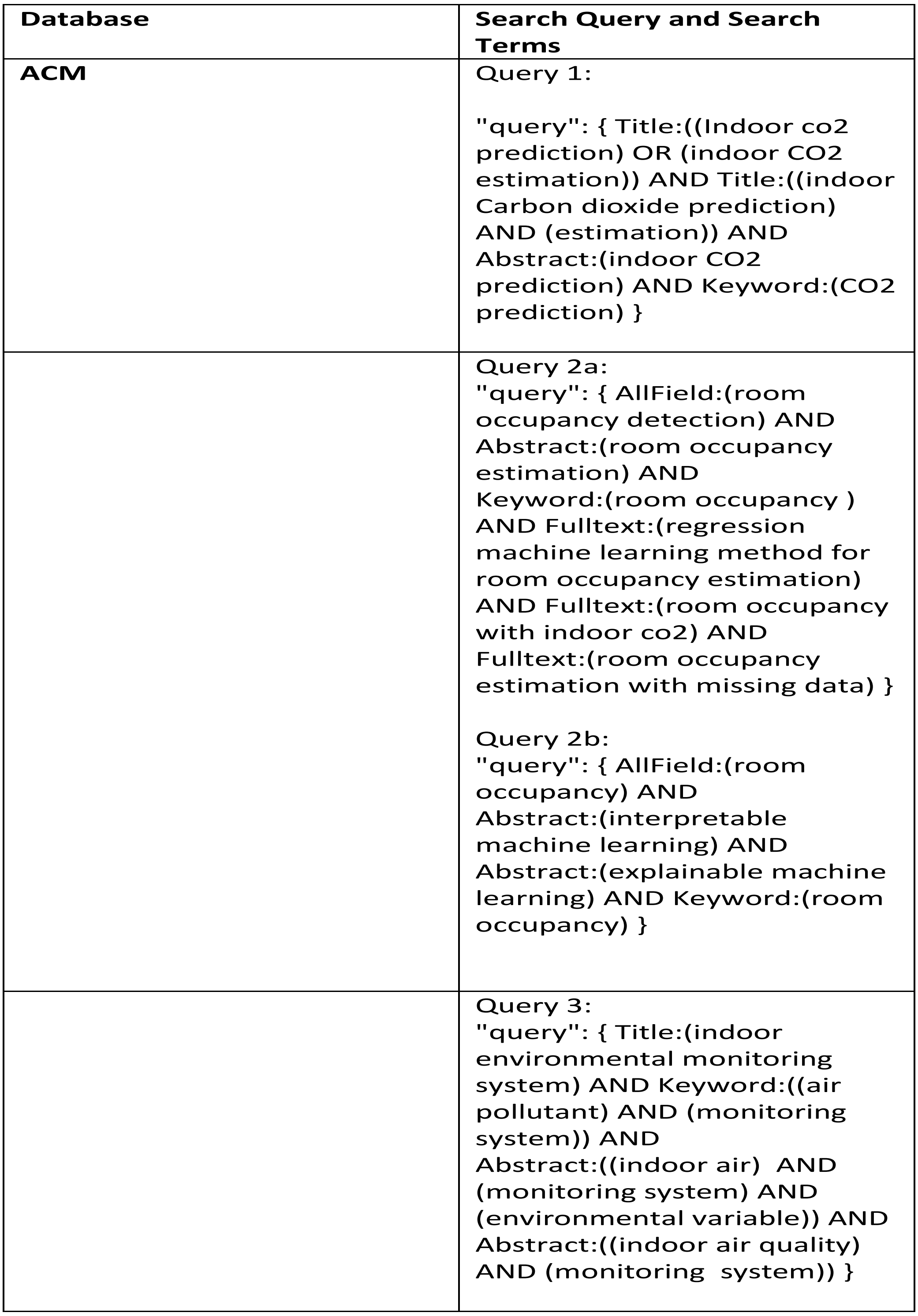
In addition to my supervisors, I would like to thank Dr Marc Roper for his insightful comments and his immense knowledge which has contributed towards achieving the desired result in this PhD study.

I thank my colleagues for the discussions and the fun we had throughout the four years of this PhD journey. I have indeed made professional friends for life.

The last but not the least goes to my family, especially my dear husband who have had to take care of home and our children while I study for my PhD. His words of encouragement, love and spiritual support contributed to finishing this PhD on time. I would not have wished for another better half other than him.

Appendices

A Literature Review Search Sources

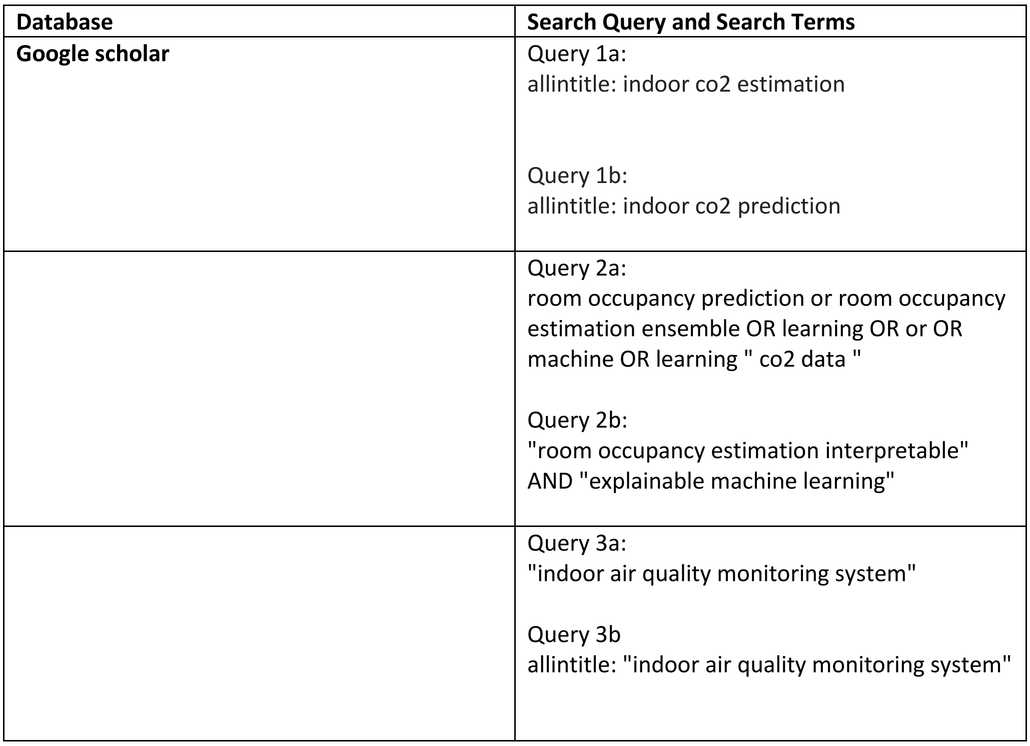


Database	Search Query and Search Terms
ACM	Query 1: "query": { Title:((Indoor co2 prediction) OR (indoor CO2 estimation)) AND Title:((indoor Carbon dioxide prediction) AND (estimation)) AND Abstract:(indoor CO2 prediction) AND Keyword:(CO2 prediction) }
	Query 2a: "query": { AllField:(room occupancy detection) AND Abstract:(room occupancy estimation) AND Keyword:(room occupancy) AND Fulltext:(regression machine learning method for room occupancy estimation) AND Fulltext:(room occupancy with indoor co2) AND Fulltext:(room occupancy estimation with missing data) } Query 2b: "query": { AllField:(room occupancy) AND Abstract:(interpretable machine learning) AND Abstract:(explainable machine learning) AND Keyword:(room occupancy) }
	Query 3: "query": { Title:(indoor environmental monitoring system) AND Keyword:((air pollutant) AND (monitoring system)) AND Abstract:((indoor air) AND (monitoring system) AND (environmental variable)) AND Abstract:((indoor air quality) AND (monitoring system)) }

Table 1: Table showing ACM databases' search queries

Database	Search Query and Search Terms
IEEE	Query 1: "Document Title":"Indoor co2 estimation" AND "Full Text .AND. Metadata":"Indoor co2 estimation" AND "methods" "Author Keywords":"Indoor co2 " AND "sensor" OR "Document Title":"Indoor co2 " AND "problems" OR "Full Text .AND. Metadata":"indoor co2 problems" OR "causes"
	Query 2a: ((("Full Text Only":room occupancy estimation) OR "Full Text & Metadata":room occupancy prediction) AND "Abstract":Machine learning) Query 2b: All Metadata":room occupancy estimation) AND "Abstract":interpretable machine learning) OR "Abstract":explainable machine learning) AND "Abstract":room occupancy detection)
	Query 3a: "Full Text .AND. Metadata":"environmental monitoring system" AND "Document Title":"air pollutant monitoring system" AND "monitoring system" OR "Author Keywords":"air pollutant monitoring system" OR "Full Text .AND. Metadata":"air monitoring system" OR "Full Text .AND. Metadata":"indoor air quality" AND "monitoring system"

Table 2: Table showing IEEE databases' search queries



Database	**Search Query and Search Terms**
Google scholar	Query 1a: allintitle: indoor co2 estimation Query 1b: allintitle: indoor co2 prediction
	Query 2a: room occupancy prediction or room occupancy estimation ensemble OR learning OR or OR machine OR learning " co2 data " Query 2b: "room occupancy estimation interpretable" AND "explainable machine learning"
	Query 3a: "indoor air quality monitoring system" Query 3b allintitle: "indoor air quality monitoring system"

Table 3: Table showing Google scholar databases' search queries

B Interview Guide for Room Occupancy Estimation Study

B.1 Interview Presentation

Title: Room Occupancy Estimation Study Using indoor Environmental Variables.

Aim: This interview study was aimed to aid in the estimation the number of people in a room when the indoor temperature, humidity, noise, pressure and CO_2 concentrations are known.

Methods: This interview study was conducted in two phases; qualitative and quantitative analysis.

Investigators:
Researcher contact details: Name: Chika Ugwuanyi,
Email: chika.ugwuanyi@strath.ac.uk Contact details: Livingstone Tower, room 1206
Telephone: 01415483705 University of Strathclyde

Chief Investigator details:
Dr Marilyn Lennon Senior Lecturer Department of Computer and Information Science
University of Strathclyde Rm 1311a Livingstone Tower
Email marilyn.lennon@strath.ac.uk, Telephone: 01415483098

Dr Richard Bellingham Director Institute for Future Cities, Senior Research Fellow
Strathclyde Business School
University of Strathclyde
Email richard.bellingham@strath.ac.uk

B.2 Background information and Introduction

The main aim of this study is to test our proposed model in a different live environment so as to ascertain to the extent to which the proposed ensemble models can be relied upon by the interested stakeholders. As a means of capturing individual stakeholder interests and concerns before the commencement of the study, people directly responsible for managing and using these rooms will be asked some questions via interview about any challenges they face in managing the rooms when the number of occupants in those rooms are not known. The result of this investigation will serve as a stakeholders' requirement gathering that will form the main reason for conducting this study Please answer the following questions as briefly as possible.

1. Are you a staff member?

2. Are you responsible for, or interested in the management of rooms?

3. Are you interested in knowing the room occupancy information of the rooms you control?

4. Are you also interested in knowing the CO_2 levels of the rooms you manage?

5. Do you currently control CO_2 in rooms that you look after: Yes/No.

6. Do you currently manage or wish to be managing room occupancy levels in rooms that you look after: Yes/No.

B.3 Benefits of monitoring indoor CO_2 and estimation of room occupancy

B.4 Introduction

1. Please provide reason/(s) for your answer in Q3 in the box below.

2. If you answered "yes" in Q4 above please provide reason/(s) in the box below otherwise write N/A.

3. If you answered "yes" in Q6 above please provide how in the box below otherwise write N/A.

4. If you answered "yes" in Q8 above please provide how in the box below otherwise write N/A.

5. What features in the room do you wish you could manage better knowing the room occupancy information and why.

6. If you could estimate room occupancy and CO2 levels more accurately in real time would it help anything.

B.5 Conclusion of any Other benefit

- In what format (the way it is displayed, how it is sent or communicated to you) would you want to receive recommendation about room occupancy and $CO-2$ information.

B.6 Conclusion of the interview

Thank you for taking part in this interview. Your answers will be considered during this study. Please print your name and signature in the space below. The data recorded will be used for the purposes of this study and then disposed of. No real names of the participants will be disclosed.

B.7 Statistical data

1. Organization Name

2. Address

3. City, Post-code

4. interview Date

C Information Sheet and Consent Form for Room Occupancy Study

Participant Information Sheet for Room Occupancy Estimation Using Indoor CO2 Concentrations

1) **Name of department: Department of Computer and Information Science**
   Title of the study: Room Occupancy Estimation using Indoor CO_2 Concentrations**.**

Introduction:
My name is Chika Ugwuanyi, I am a PhD student at the University of Strathclyde.

What is the purpose of this investigation?

The reason for this study is to use machine-learning (the study of models that computer uses to perform a task) models to improve the estimation of room occupancy of an office room from CO_2 sensor data captured via a commercially available environmental sensor device

The initial phase is a qualitative (gathering of non-numerical data) analysis which involves finding out (via interviews and surveys) from the staff responsible for managing/using the office if there are any interests/challenge they face while using the room especially when the number of occupants' in those rooms are not known. The result of this investigation will serve as a stakeholders' requirement gathering activity that will inform the main study.

The second phase is a quantitative (gathering of numerical data) analysis which involves obtaining environmental (noise, temperature, CO_2, humidity, pressure) and occupancy data from the offices where the study will be conducted for final analysis. This data collection period will last for a minimum period of one month and a maximum of three months

Do you have to take part?

It is not compulsory that you participate in this interview, if you wish to take part and later decide not to be involved, it is not going to affect the way you would be treated or regarded. In other words, participation is voluntary.

What will happen in this project?

1) We will conduct a simple semi-structured interview with the Swarmonline employees either managing or using those rooms via one on one or focused group meeting.
2) We will send out the participants' information sheet containing a consent form so that people who are interested could indicate by signing the consent form. During this process, the participants will be given the option of choosing between the two methods (one-on-one/focused group) for the interview.
3) The method (one-on-one/focused group) for the semi-structured interview will be chosen based on what the participants agreed that would be most suitable for them.
4) We will install an environmental sensor device called Netatmo (https://www.netatmo.com/en-gb/weather/weatherstation) in the office spaces to allow for the recording of variables like temperature, humidity, noise, pressure and carbon dioxide. No personal or identifiable information will be recorded at any time using this sensor. This will last for between one to three months.

5) We will obtain room occupancy levels from the staff of the company (SwarmOnline) where the study will be conducted. This ground truth occupancy data will be obtained via a webcam installed by the company. The images will be counted and managed by a member of the staff of the company where the interview will be conducted. The data obtained via numerical count of these images can only be used for analytical purposes.
6) Because these images will be managed by an employee of SwarmOnline, we will not have access to the identifiable images, rather only the numerical counts will be sent to us for the analysis. After which the employee of SwarmOnline with the images will destroy the images as agreed by the participants.
7) We will analyse the data using a machine learning (regression and time series) method to determine how accurate the model is for predicting room occupancy using CO_2 concentrations**.**
8) The same process applied in 2-3 above will be followed before a post-study interview that will be conducted with the same people to see if they find the results interesting (the quantitative data, the models and the results of the analysis will be shown to them).

Why have you been invited to take part?

You are invited to take part in this investigation either as a stakeholder that uses the offices where this study will be conducted or as someone who might be interested in knowing the real-time occupancy information of these offices.

There is no special skill needed for you to participate. No screening procedure is required. The names of the participants providing us with their data will be, kept secret (anonymized). Your work environment will not change as a result of this study directly, but ongoing findings might contribute to improving the overall quality of your working environment.

What are the potential risks to you in taking part?

There is no potential risk associated with this study. Firstly, the environmental sensor will only record the current environmental variables, and the webcam/digital camera will be able to track the number of people entering and leaving the room.

A member of the staff of the company where the study will be conducted will be responsible for managing and recording the occupancy data in order to avoid breach of security. The images to be taken by the device will be only for the purposes of updating the number of occupants

No personal or office information will be tracked or recorded.

What happens to the information in the project?

The voice data recorded on computer device will be used for the purposes of this study and then disposed of. No real names of the participants will be disclosed during the analysis.

The University of Strathclyde has a Data Protection Policy which sets out the roles and responsibilities in relation to data protection within the University.

All the data obtained will be protected based on the core principles of University of Strathclyde's GDPR as stated in this link https://www.strath.ac.uk/professionalservices/media/ps/strategyandpolicy/GDPR_Principles_Poster.pdf .

The guidance on how long to retain the audio records is stated on the University of Strathclyde information and records management guidance notes via the link below

https://www.strath.ac.uk/media/ps/cs/foi/recordsmanagement/Information_and_Records_Guidance_9_Retention_and_Disposal_v2.1.pdf.

The audio record will be held for maximum of 1 year or at the end of the PhD research, after which it will be deleted/destroyed from the secured server where it is stored.

Thank you for reading this information – please ask any questions if you are unsure about what is written here.

What happens next?

If you are happy to participate, I will send out a consent form for you to sign and confirm this.

At the end of the investigation, we will send a feedback to you and publish the results of our findings.

If you are not interested thank you for your attention.

Researcher contact details:

Name is Chika Ugwuanyi, Email: chika.ugwuanyi@strath.ac.uk

My contact details: Livingstone Tower, room 1206 Telephone: 01415483705
University of Strathclyde

Chief Investigator details:

1) Dr Marilyn Lennon
Senior Lecturer Department of Computer and Information Science
University of Strathclyde
Rm 1311a Livingstone Tower

Email marlin.lennon@strath.ac.uk, Telephone: 01415483098

2) Dr Richard **Bellingham**
Director Institute for Future Cities, Senior Research Fellow
Strathclyde Business School
University of Strathclyde

This investigation was granted ethical approval by the Department of Computer and Information Science Ethics Committee.

If you have any questions/concerns, during or after the investigation, or wish to contact an independent person to whom any questions may be directed or further information may be sought from, please contact:

Department of Computer and Information Science Ethics Committee
University of Strathclyde
Livingstone Tower
26 Richmond street
Glasgow
G1 1XH

Telephone: 0141 548 3189
Email: ethics@cis.strath.ac.uk

Consent Form for Room Occupancy Estimation Using Indoor CO2 Concentrations.

Name of department: Department of Computer and Information Science.
Title of the study: Room Occupancy Estimation Study with Indoor Carbon Dioxide Concentrations.

- I confirm that I have read and understood the information sheet for the above project and the researcher has answered any queries to my satisfaction.
- I understand that my participation is voluntary and that I am free to withdraw from the project at any time, up to the point of completion, without having to give a reason and without any consequences. If I exercise my right to withdraw and I don't want my data to be used, any data which have been collected from me will be destroyed.
- I understand that I can withdraw from the study any personal data (i.e. data which identify me personally) at any time.
- I understand that anonymised data (i.e. .data which do not identify me personally) cannot be withdrawn once they have been included in the study.
- I understand that any information recorded in the investigation will remain confidential and no information that identifies me will be made publicly available.
- I consent to be a participant in the project

(PRINT NAME)	
Signature of Participant:	Date:

D Coding Scheme for Room Occupancy Study Interview

Categories	Description	Example
Management of Electrical and Mechanical services	The participants acknowledged that knowing the occupancy and the CO_2 levels will help to vary the input supply to the exact number of people in the space. These inputs are heating, lighting and cooling	P1: "Yes the main reason will be that, we will like to be able to ensure that the plants is providing a reasonable environment to the occupants and therefore we are meeting the air obligations. But It is also useful to know, plants can be varied at its input in terms of heat input, chilled water input, ventilation input and plants can be matched to the number of people actually in a space. Then that is the more efficient way of operating plant. At the moment where we have rooms where there is no control, the plant will run at its designed capacity which might be for 50 people. It doesn't matter if there are 5 people in there or 50 people in there. The plant will run at the same rate. Therefore, a lot of the time we are over providing for the same space."
		P3: "I also am interested in knowing whether or not there are any technique to it ... to save energy by perhaps increasing or decreasing the heating or the lighting in the room depending on the occupancy levels."
Indoor Comfort and Well-being	P1, P3 and P5 agrees that knowing the exact number of people in the space and the CO_2 levels will guide the room managers in managing the occupants' comfort and well-being without compromising the thermal comfort law. An extract from the interview on this is below	P1: "But my responsibility or our responsibility in the team I have is that we need to make sure that the spaces that people occupy are in the best we can make it fit for purpose and are complying with the legislation. Thermal comfort as well sort of.. Yes is to make sure that the people are studying and the people are working in our environment has to be of a reasonable quality"
		P3: "Yeah I think perhaps to add is our safety element as

		well, for instance high levels of CO_2 there is actually those of safety threat which is something that we should know... about those threat.... and so on and so forth."
		P5: "I was only going to add to understand the rooms maximum"}
Energy Savings	P1 and P3 agrees that when the exact number of the people in the room are known, the plant supply will be decreased or increased based on that figure and no longer. This will save energy and reduce cost of the plant's operation.	P1: "And where there is existing equipment available, we don't have that facility at the moment, we could find a way of reasonably costed way of retrofitting that, ultimately, that would also save us money and would hopefully make the spaces for the occupants. It won't detract from that."}
		P3: "...to save energy by perhaps increasing or decreasing the heating or the lighting "
Reduce Carbon Emission	P1 agrees that operating the plant supply only on the need basis will reduce carbon emission in the environment	P1: "... That ultimately would also save us money, save us carbon emissions..."
Utilization	The participants (P2 and P4) believed that knowing the exact number of people in the hall will help them to know when the rooms were used or unused and when the rooms' capacity are under utilized.	P2:"Am interested in curbing the differences between the actual usage and planned usage of central teaching rooms"
		P4: "if it means us reducing the number of people in our room just to help efficiency then that's me."
		P2: "Well as mentioned earlier I want to be able to compare it against the planned usage of the rooms and our process for not finding departments of the rooms that were booked or where an appropriately sized rooms have been booked and students which are attending.. this should lead to more efficient use of the space that we have. It should also lead to better information when new

		rooms are developed for central teaching spaces like size and usage."
Productivity	The participants P3 and P4 believed that knowing the exact number of the people and the CO_2 information could improve productivity.	P4: "... I think the study comes with a good result to help us to understand how to keep productivity high ... based on ... I think that will help us maintain that level of productivity ... so that we will know what to do."
Information Usage and Format	All the participants are interested in using the results of the analysis to plan the efficient utilization of the rooms where they manage. For instance, P1, P2, P3, P4 and P5 expects the following formats of the results.	P1: "Again if it was live data, we would want that to be taken straight into our automation systems. So the automation systems will read the sensor that says.. the CO_2 levels is currently it will automatically make changes to the ventilation or the heating or the cooling plant that is serving that space, to either vary up or down depending on that CO_2 levels. Ultimately, that would be the preferred option. At a lower level, unless responsibly will be to be told, to be given regular report or electronic report that says."

D.1 Interview Schedule for Room Occupancy study

Table 4: Participants Data for Room Occupancy Interview

Participants	Status	Interview Date
P1	Estate Staff	22/02/2018
P2	Estate Staff	20/03/2018
P3	Swarm Staff	6/11/2019
P4	Swarm Staff	6/11/2019
P5	Swarm Staff	6/11/2019

E Formal Requirements Specification

E.1 Functional Requirement

The O-EMA functional requirements will mainly make use of the interview themes shown in section 7.4.1 obtained from the interview data and will be based for answering the research question which states that **"How will the stakeholders' contributions be identified and incorporated in the design phase of the dashboard."**. Furthermore, O-EMA application maintains information about the levels of the indoor temperature, CO_2, humidity, occupancy and the information about when the levels are above or below the required limit as they were required by the interview participants. More of the functional requirements of O-EMA application would be discussed during the implementation phase.

E.2 Performance Requirements

This system will be designed in a way to reduce redundancy, which means that no same data will be added twice to the database. The database will be created to store only 1 year data to avoid data redundancy. There will be frequent normalization of the data stored in the database in order to prevent data redundancy which often leads to wastage of storage space in the database. The system will be normalized based on 3 normal form (3NF) Atzeni and De Antonellis [1993], because it is enough for most practical purposes. Another non-functional requirement of this system will be the time of execution of the functional requirements. Because this system is meant to be a real-time system, the execution of the process on the function should not be more than 10 seconds.

E.3 Safety and Security Requirements

In the case of fatal damage to the database often caused by disk crash, data recovery method will be used to restore the most recent backup copy of the database and more current state will be reconstructed by redoing the committed transactions from the backed up logged up to the point of crash. This will ensure that all the lost data is recovered. One Only authorized users can access this system.

E.4 Software Quality Attribute

- **Availability:** To ascertain if all rooms in a flat is fit for purpose and safe for occupants health, the users will have high dependency on the functionality of O-EMA application. As a result, when O-EMA webpage is down the room occupants health could be adversely affected unknowingly. Therefore, to maintain constant awareness of the rooms' indoor environmental condition, O-EMA webpage must be hosted on a high performance website.

- **Correctness:** The real time data and the information being displayed by the system must be accurate always.

- **Usability:** The users of this system should be able to use it without any difficulty.

E.5 Language and Visual Problems

Any information or message being displayed by O-EMA application must have clear and simple meaning to enable non-technical users to understand the system. All the text on the

O-EMA dashboard must be eligible with a font size of atleast 20 Sans serif fonts in case of users with visual impairment. In addition to font size, all the click-able links must be large in size as well as with a clear colour.

E.6 Accessibility Guide

The developers will adhere to the Web Content Accessibility Guidelines 2.0

F Implementation

F.1 Description and Priority

In order to mitigate any unforeseen circumstances of each room that could result in the unhealthy indoor place, the O-EMA system will have a function that generates daily reports via email to its users. This system if developed and utilized could save cost and serve as a guide to landlords and estate managers in keeping up with the legislation.

F.2 System Security

Though, there won't be any security challenge associated with this desktop application, there will be a single login as suggested by the participants. Therefore, O-EMA will have a registration page for new users and a login page for authentication of already registered users as shown in Figure 1. The decision for this type of security measure was copied from the proposed architectural style in Fielding and Taylor [2000] and the need of the users from the analysed interview data. Other security measures for O-EMA are 1) the link for resetting the users' forgotten password and it is shown in Figure 2 and 2) the link for resending the password link and it is shown in Figure 3

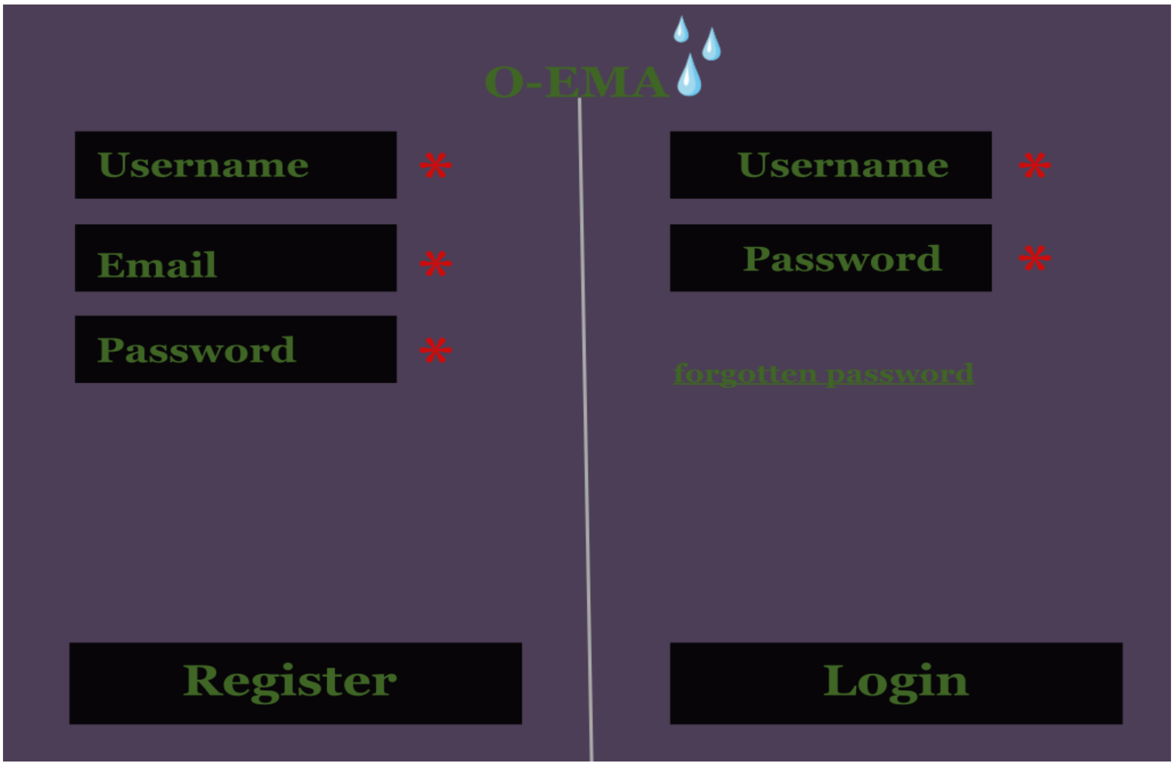


Fig. 1. O-EMA registration and login page

Actors: Room manager
Trigger: The user clicks the O-EMA website and the login and registration page loads.
Preconditions: The user creates an account with the registration form if they are new to the website or fills in the login page if they have already registered.

Post conditions: The account details of the user will be stored in the O-EMA database for subsequent authentications.
Response Sequence:

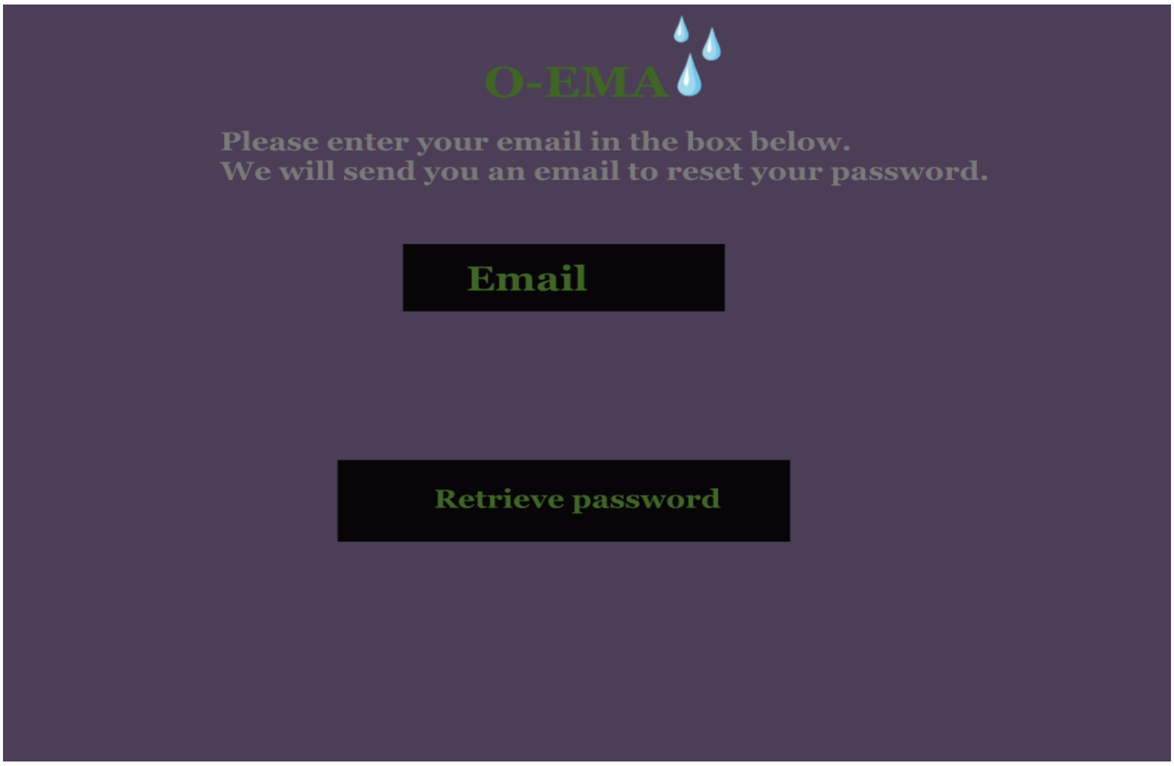
1. The user needs their user name, email and password of their choice to be able to create an account.
2. If the account is successfully created, an automatic message will be sent to the new user via email with a link to verify their account.
3. Once their account is verified, they will be able to view the main page of O-EMA application and see the summary of all the variables being displayed on the screen for timely decision.
4. If the account has been previously created, the user will be able to login with their official email and the registered password.
5. If the user has forgotten their password, then the forgotten password link should be clicked for password rest.
6. The user will be asked to enter their email address for the password reset link to be sent to them.
7. The user will click the link that was sent to their email for password reset.
8. If the user hasn't received the password reset link for more than 5 minutes, another "Resend email" link will be clicked. Steps 6 - 7 should be repeated.
9. Steps 1 - 3 will be repeated for new users and step 4 will be repeated for already registered users.

Exceptions: The dashboard will continue to be open for each successful login.

F.3 Messages and its Meanings

As described in Section 7.1, if O-EMA application could not show hotspot areas at the point of login, the user or occupants might not be aware of which room is at risk. For instance, if the indoor CO_2 levels is consistently above 2000ppm, occupants could be at risk of poor indoor air quality which could result to nose itching, eye scratching, sneezing etc. So to curb this type of scenario, O-EMA will be able to show a distinct colour sign on the dashboard for immediate recognition of the real time condition of the each room. This suggestions comes from 1) one of the interview themes and 2) answering the research questions which states **"What is the best format for displaying the result of the room occupancy levels to the interested stakeholders".**

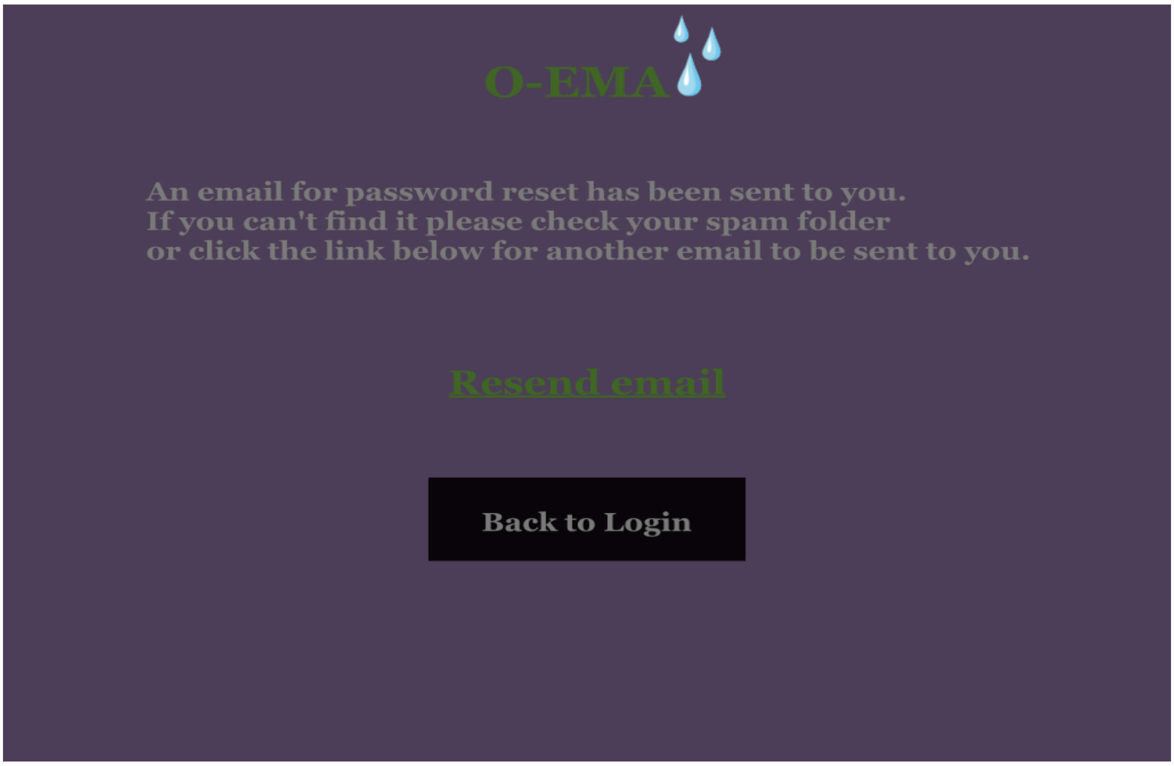
Figure 4 provides a story board of how O-EMA is expected to achieve its aims and objectives. On the O-EMA dashboard page, the curved black squares represents each room



O-EMA

Please enter your email in the box below.
We will send you an email to reset your password.

Fig. 2. Link for resetting forgotten password



O-EMA

An email for password reset has been sent to you.
If you can't find it please check your spam folder
or click the link below for another email to be sent to you.

Fig. 3. Link for resending unrecovered password

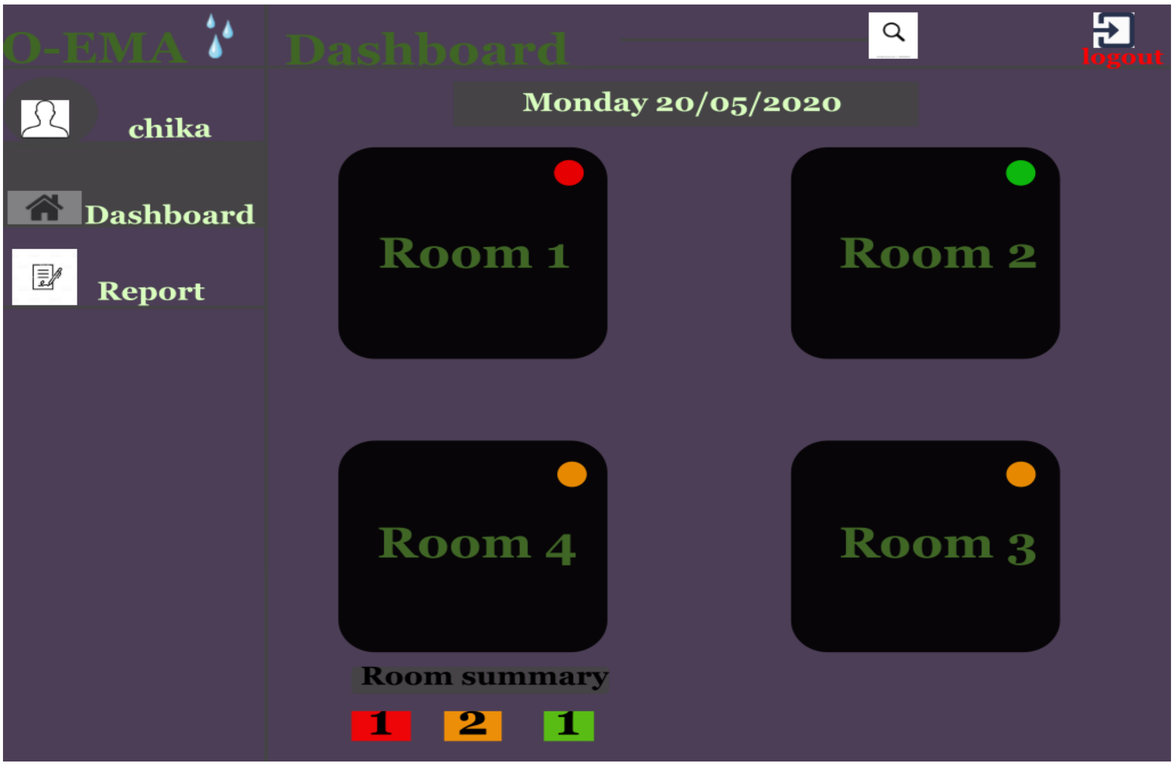


Fig. 4. O-EMA dashboard showing all rooms that are hotspots and safe in real-time

in the apartment with its name on each square for "Monday 20/05/2020". The round colour signs on each square represents the worst room condition of those rooms. This means that in "Room 1" for instance, there is a variable that would likely put the occupants' health at risk. Whereas in "Room 2", all the current variables' levels are safe for its occupants. At the bottom of the dashboard page shows the "Room summary" which represents the total number of the rooms that are at risk, near safe and safe respectively. There is also a logout button that enables the user to sign out of their account when they wish. The dashboard room colour signs will continue to store and update its colours on the screen based on the latest room reading. This means that previous danger (red) variable level could change to green (safe) once the necessary steps has been taken to correct the situation or when the environmental room condition changes.

Actors: Room manager
Trigger: The user clicks the "Dashboard" menu
Preconditions: Netatmo sensor records and stores the temperature, CO_2 and the humidity levels of each of the rooms.
Post conditions: The recorded data will be obtained and stored by the O-EMA database for display on the O-EMA.
Response Sequence:

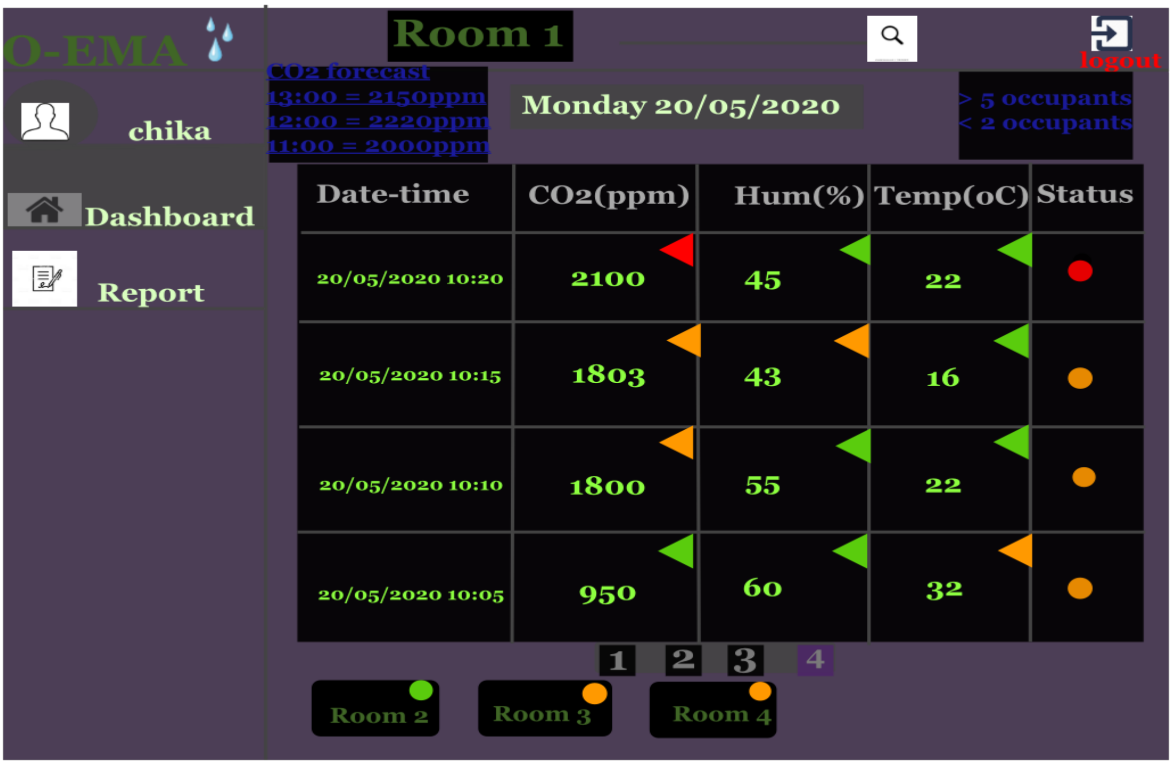
1. All the rooms' variables will be updated and their colour signs will be updated based on the current levels.

2. The room summary will also be updated based on the current levels of the variables in every room.
3. Necessary action is then taken to resolve the issue.
4. The highlighted red colour (danger) will change to green (good) once the issue has been resolved or remain red until the issue is resolved.
5. Steps 1 - 3 will be repeated once another issue is found with any of the variables.

Exceptions: The dashboard will continue to show all the variables in green highlight if all the rooms variables have normal levels.

F.4 Daily Room Reading

In order for the O-EMA user to quickly check which of the variables is not within the indoor accepted range, a full variables' display of each room is shown on a dashboard with its colour signal attached to it. As shown in Figure 5, The "Status" column in Room 1 dashboard is the summary of the variables' levels and its impact to occupants. In that status column the current level of CO_2 as at 10:20am on 20/05/2020 is above the required standard hence the red colour circular sign. This red colour symbol was currently displayed on the main page of the dashboard for room 1 as shown in Figure 4.



Room 1

CO2 forecast
13:00 = 2150ppm
12:00 = 2220ppm
11:00 = 2000ppm

Monday 20/05/2020

> 5 occupants
< 2 occupants

Date-time	CO2(ppm)	Hum(%)	Temp(oC)	Status
20/05/2020 10:20	2100	45	22	●
20/05/2020 10:15	1803	43	16	●
20/05/2020 10:10	1800	55	22	●
20/05/2020 10:05	950	60	32	●

Fig. 5. Room 1 dashboard showing the last five readings in the real-time

Actors: Room manager
Trigger: The user clicks the "Room 1" link.
Preconditions: The O-EMA database table for "Room 1" loads the sensor data it obtained from Netatmo sensor.
Post conditions: The stored data for room 1 loads and the last 5 records of temperature, CO_2 and humidity levels displays on the dashboard.
Response Sequence:

1. All the rooms' variables will be updated and their colour signs will be updated based on the last 5 records.
2. The expected future CO_2 level is also displayed with its corresponding time.
3. The number of occupants is also shown on the dashboard for room 1.

Exceptions: The dashboard will continue to update every 5 minutes interval with the current readings for each database table.

F.5 Daily Analytics Plot

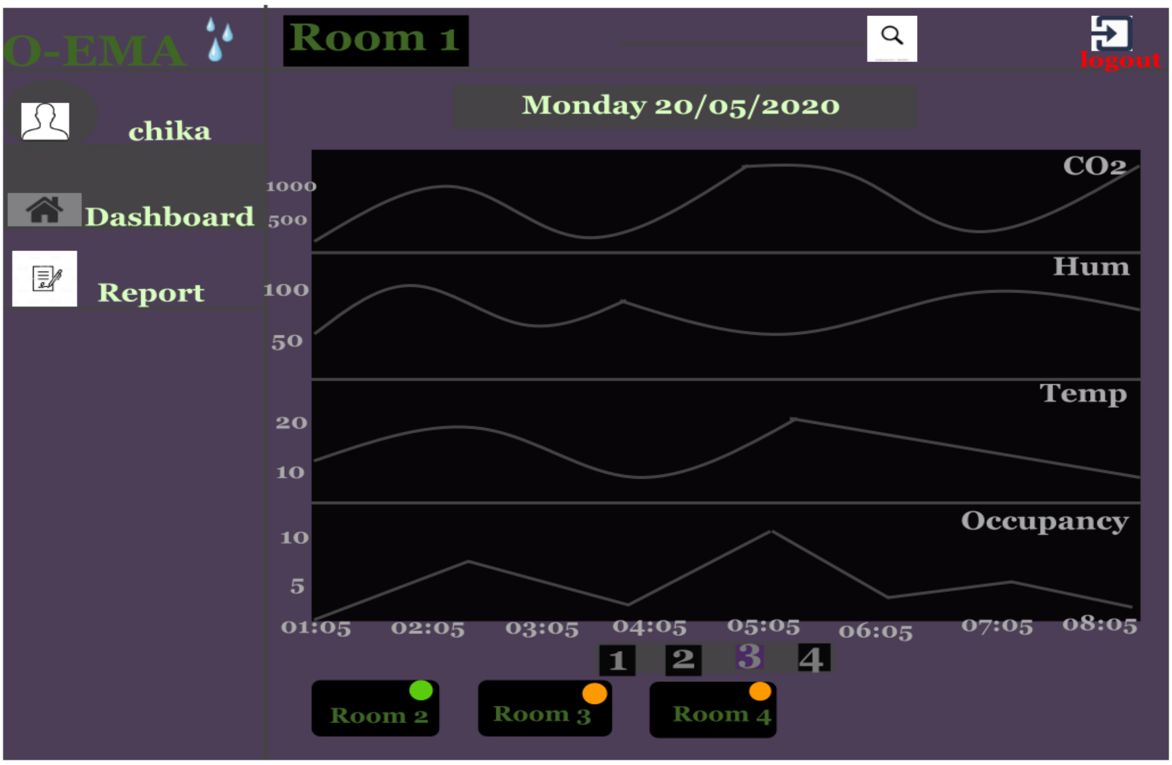


Fig. 6. O-EMA Analytics page showing the daily plots of temperature, CO_2 and humidity levels recorded by the sensor.

It has been observed that plots are effective in data analytics. This is because it simplifies the true situation or results of the problems for better interpretation by non experts. Therefore, O-EMA will provide suitable line plot in order to simplify the daily sensor recordings of temperature, CO_2 and humidity values. This will be made available in the pagination pages of each room as shown in Figure 6.

Actors: Room manager
Trigger: The user clicks the pagination link.
Preconditions: The current day page has successfully loaded.
Post conditions: O-EMA loads the stored data for the requested room and its subsequent plot. O-EMA also shows the link to other rooms at the bottom of the page.

F.6 System Report

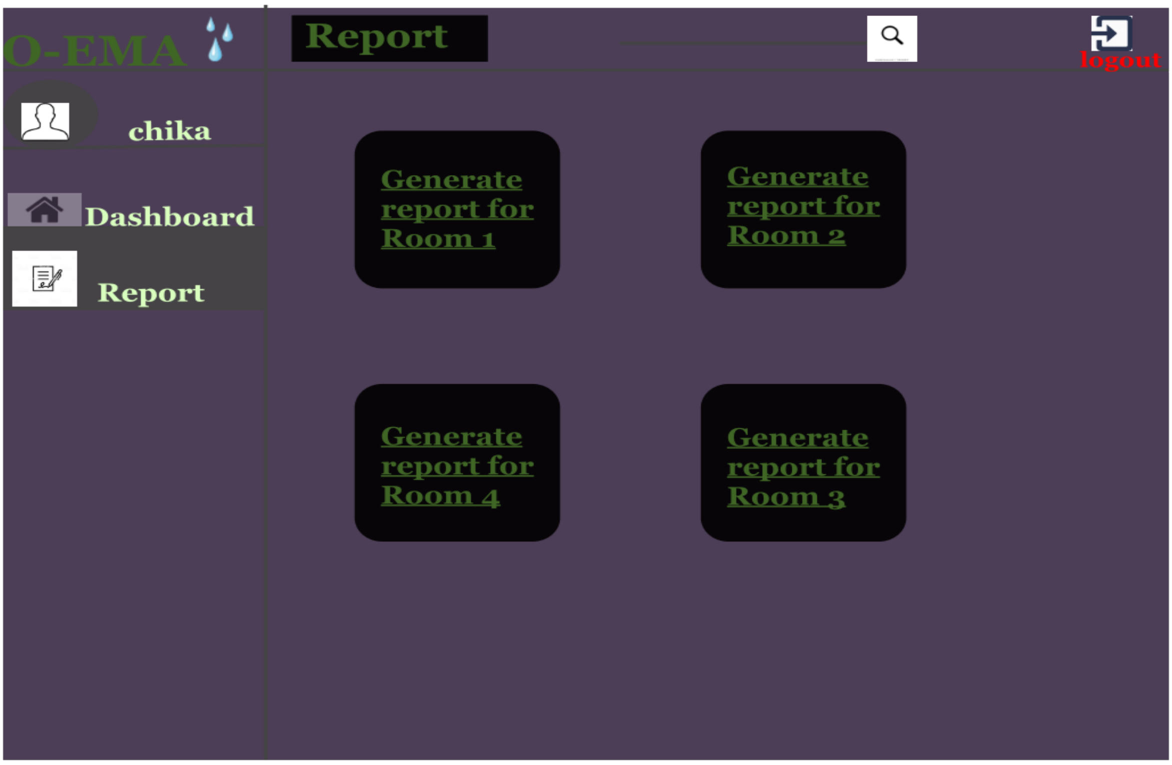

O-EMA
Report
logout
chika
Dashboard
Report
Generate report for Room 1
Generate report for Room 2
Generate report for Room 4
Generate report for Room 3

Fig. 7. O-EMA Report page for all rooms

For record purposes and for answering the research question which states **"What software design and the functional requirements will effectively help the users to perform their task and make the usability of the system easy"**, it is important to keep the track record of daily readings. This will help the room mangers to have accurate report of the daily happenings of each of the room. The reports can be calculated in terms of percentage of the time the variables' levels are below or above standard. The report page is shown in Figure 7.

Actors: Room manager
Trigger: The user clicks the "Report" menu.
Preconditions: The current page has successfully loaded.
Post conditions: O-EMA loads the already stored data incidents from O-EMA database.

F.7 Generate Report

Actors: Room manager
Trigger: The user clicks the “Generate report for Room 1” link.
Preconditions: The current page has successfully loaded.
Post conditions: O-EMA loads the already stored past data (humidity, temperature and CO_2) for each room from history table in O-EMA database. The history data comprises of percentage of the time in a day that the variables’ levels were either substandard or standard in each room.

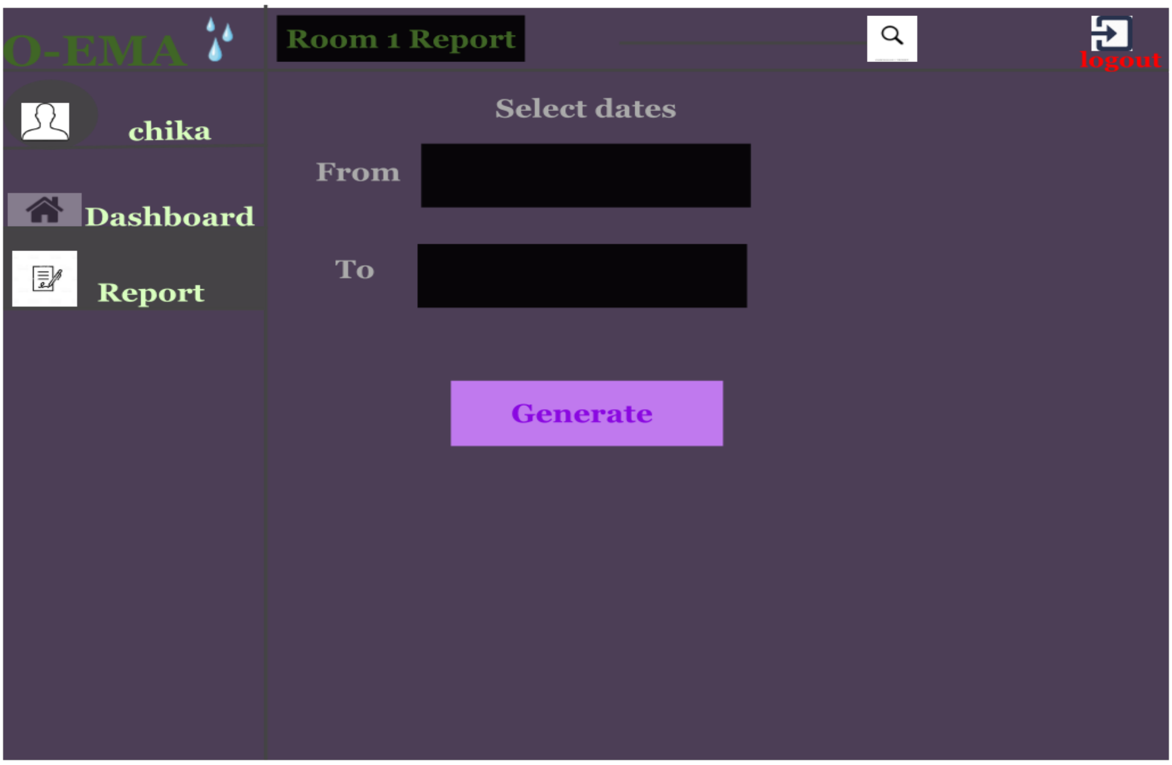


Fig. 8. O-EMA **report generation page for Room 1**

Response Sequence:

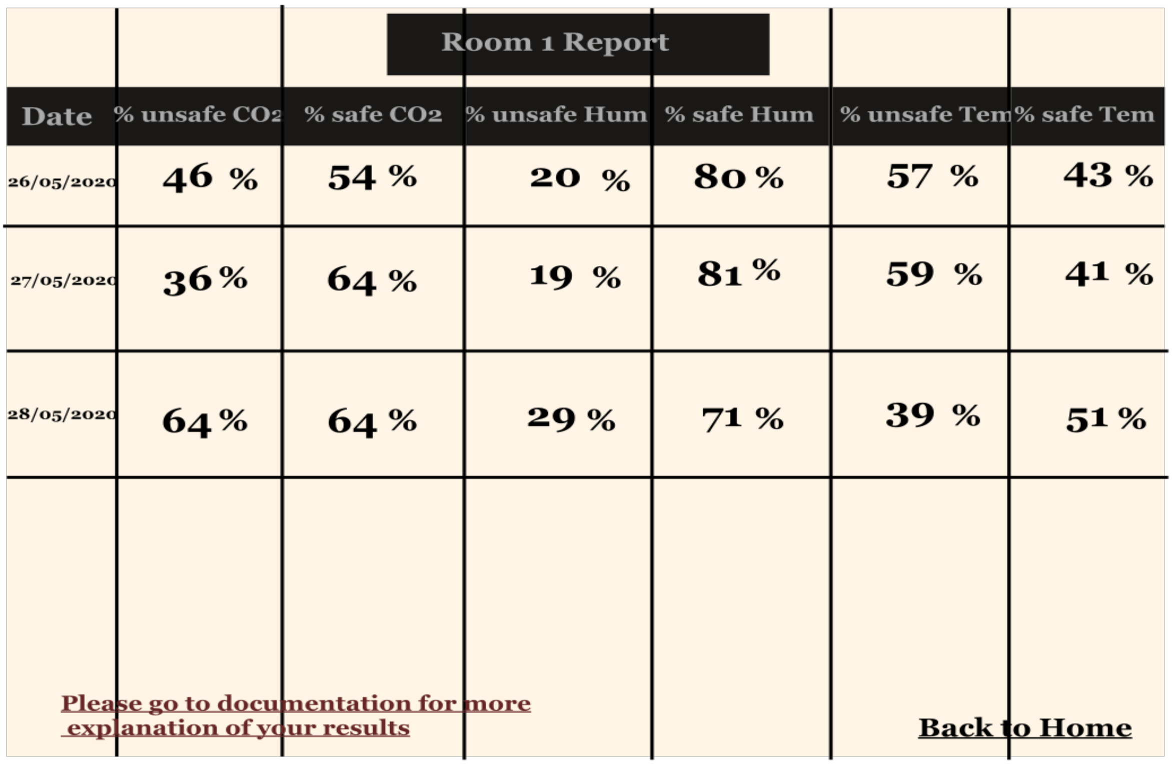
1. The user clicks on any of the room report link, the recorded calculated history event will open on another page.

2. The user will be asked to choose the dates they are interested in generating its report as shown in Figure 8.

3. The user may choose to print out any of the report as CSV file or readonly as shown in Figure 9.

4. The user will click another link below for explanation of the results.

Exceptions: If there are no recorded events for the date selected by the user, O-EMA will trigger a warning that states “No recordings yet for some of your selected dates”.



Room 1 Report

Date	% unsafe CO2	% safe CO2	% unsafe Hum	% safe Hum	% unsafe Tem	% safe Tem
26/05/2020	46 %	54 %	20 %	80 %	57 %	43 %
27/05/2020	36 %	64 %	19 %	81 %	59 %	41 %
28/05/2020	64 %	64 %	29 %	71 %	39 %	51 %

Please go to documentation for more explanation of your results

Back to Home

Fig. 9. O-EMA sample report sheet for Room 1

G Design

G.1 Conventions

In this document, the term “occupancy” and “environmental monitoring” maybe referred to as

- Estimating number of people in the room.
- Monitoring and recording of the indoor temperature, humidity and CO_2 levels.
- The acronym relational DB will be referred to as Database.
- The acronym ER will be referred to as entity relationship.

The O-EMA will have three colours as a warning sign. These colours will be typical of the traffic light colours as suggested by one of the participants. They are green(safe), yellow(warning), red(danger). Safe means that all the levels(humidity, temperature and CO_2) are within the required limit. Warning means that the levels are above the required limit but has not entered the dangerous phase. Danger means that the levels are above the required limit and at the same time poses risk to occupants health. All the three variables will reflect each of these colours at each point in time. The occupancy levels shown on O-EMA will serve as indicator to the user when the room is occupied or not so as to know when to regulate the electrical appliances. This method has been shown to reduce the cost of energy in buildings.

The required levels of the indoor environmental variables shown on O-EMA will be based on the recommended levels by glsashrae UK because O-EMA will be used in the UK. However, these levels are not perfect for every situation. They were as follows:

- Humidity levels between 45% and 55% will reflect green colour. Humidity below 45% and above 65% will reflect yellow colour.
- CO_2 levels less than 1000ppm will reflect green colour. CO_2 levels between 1000ppm and 2000ppm will reflect yellow colour. CO_2 levels above 2000ppm will reflect red colour.
- Temperature between 18°C and 23 °C it will reflect green colour. If the temperature becomes less than 16°C and greater than 25°C it will reflect yellow colour.

G.2 Intended Users and Beneficiaries

This project is a prototype application for environmental monitoring towards energy management of offices and rooms. Landlords or estate managers can use it to monitor their individual properties and employers can use it to monitor their offices.

G.3 Project scope

O-EMA will create a simple, easy-to-use application for the beneficiaries mentioned in Section G.2 who wants to know when the temperature, humidity and CO_2 levels of their rooms are not within the desired levels, which will ensure that proper, timely or corrective action is taken for overall well-being of the occupants. This approach will prevent the complete breakdown of the indoor places.

O-EMA is based on a relational database of the stored sensor data with its notification functions. The system will have a database server storing thousands of indoor temperature, humidity and CO_2 levels. These variable levels will be shown on the screen permanently in addition with the number of people in those rooms. O-EMA will be able to update once the variables' levels changes or are normalized. Most importantly, O-EMA hopes to provide good user experience along with timely data communication to the its users.

G.4 Additional Sources for the Software Monitoring Design

More information on the previous monitoring design systems (health and environment), web application, database architecture, the programming language to be used and the characteristics of environmental monitoring systems may be found in the articles below:

- Effective requirements practices by Young Ralph R (2001): This articles helps the researcher to follow the best practices in writing detailed software requirements and full documentation of each stage of the requirements which the development teams will consult or use during the system development.

- Enabling technologies for smart city services and applications by Balakrishna Chitra (2012): This papers discusses examples of architecture used in smart city to create real world awareness via embedded sensors on mobile devices in order for people to live smartly. This paper also covers some obtainable framework that makes information available on mobile devices via knowledge engineering of real world data.

- Requirements Engineering: A Roadmap by Nuseibeh, Bashar and Easterbrook Steve (2000): This paper deals with requirement gathering step by step approach.

- RESTful web services by Richardson, Leonard and Ruby Sam (2008): This book discussed how to develop RESTFul API with regards to client-server protocol over HTTP protocol.

- Architectural styles and the design of network-based software architectures by Fielding Roy T and Taylor Richard N (2000): This paper discusses model standard for the architecture design of network-based web application. These standards discussed in this book will guide in the development of scalable application that meets some demand such as 1) how the partition of the systems work, 2) how different component part of the system communicates with each other with its information, etc..

- Software systems architecture: working with stakeholders using viewpoints and perspectives by Rozanski Nick and Woods Eoin (2011): This book discussed how to ensure that both the design and the development process that meets the needs of the stakeholders. This includes how views can be used for better understanding by the stakeholders.

- An embeddable dashboard for widget-based visual analytics on scientific communities by Derntl Michael and Erdtmann Stephan and Klamma Ralf (2012): This paper helps the researcher to understand how embeddable and personalized dashboards can be developed so that the SQL queries from the database can be visualized using widgets.

- The MyESnet Portal: Making the Network Visible by Dugan Jon and Engineer Network (2012): This paper explains how large dataset can be made visible through a network and user-centred interface. The paper also discuses how every feature added to the system needs be tested by the users so that they can give their opinion about the feature.

- Environmental monitoring system by Mayer, John and Van't Slot, James E and Rawlings, Daniel (2004): This paper discussed the various patents of how some of the previous software monitoring systems were developed including how some push notifications were implemented for alert purposes.

- Pulmonary diagnostic system by Snow, Michael G and Tyler, William R and Hsu, Sungpeng and Fallat, Robert J (1989): This paper discussed the information obtained from a software in kind of notification or warning helped in the diagnosis and treatment of pulmonary illnesses.

- Heart-related parameters monitoring apparatus by Warner, Glenfield (1989): This paper discussed how early warning from a software helped in the diagnosis and treatment of heart related illnesses.

- Monitoring system for producing patient status indicator by Nevo, Igal and Guez, Allon (1994): This paper discusses monitoring methods based on patients' measurement which shows different levels such as normal, least normal and critical. These levels are later transformed into baseline such as maximum and minimum which shows the every physician the conditions of patients.

G.5 System features with Relational Database

A free online photo editor known as PhotoScape X team [2019] will be used to design the O-EMA. Before using PhotoScape X for the system design, another free online software tool for drawing complicated diagrams known as Lucidchart Faulkner and Contributor [2018] was used for the drawing of the entity relationship (ER) diagram of O-EMA such that it can reflect the views of the participants about having a kind of software design where different rooms' variables will be connected to one interface. The ER diagram is shown in Figure 10. Some of the decision for this type of ER diagram was based on the previous studies done in the monitoring system which are listed in Section G.4. There will be different sensors for each room each measuring the current environmental variables being displayed.

The ER diagram of O-EMA will be defined based on some assumptions such as 1) there are 3 rooms in the flat or company, 2) there will be single login, 3) the O-EMA user is interested in only CO_2, humidity, temperature and occupancy level and 4) there are three sensors for the three rooms. Based on these assumptions, there will be six tables in O-EMA database. These tables are Sensor, Room 1, Room 2, Room 3, Variable-hist and Admin. The "Variable-history" will responsible for storing the results of the report that will be generated. The temperature, CO_2 and humidity values represents the percentage of the time the values of each variables

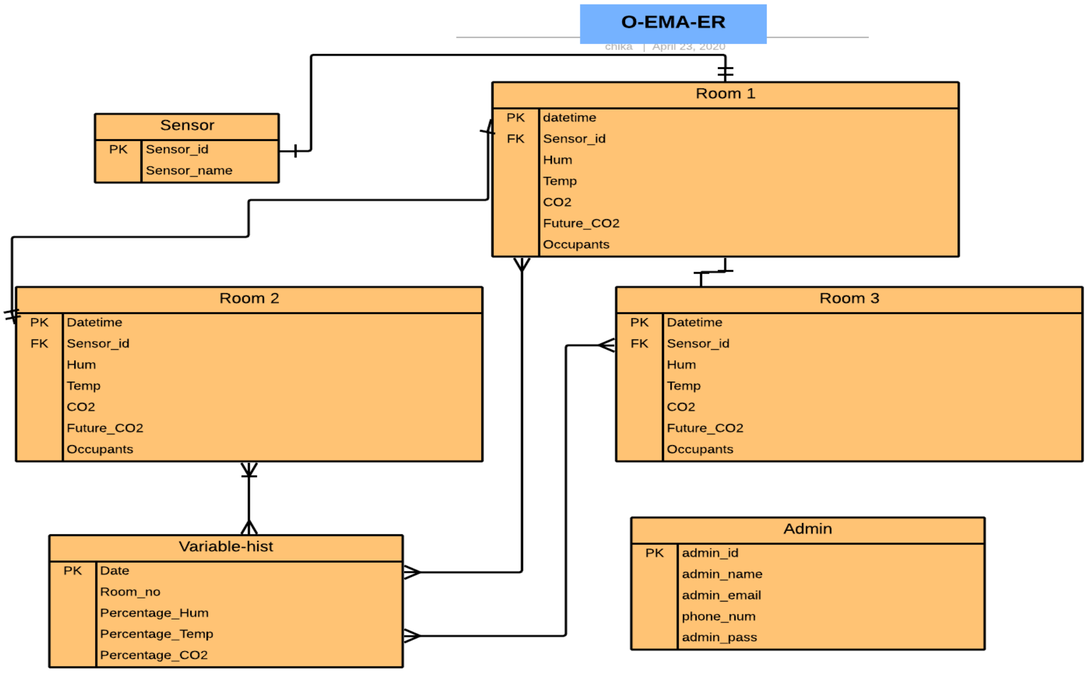


Fig. 10. ER diagram representing the O-EMA database relationships

in a dangerous level per day. The Admin table will contain the user information for the authentication before privilege is granted to access the O-EMA.

G.6 System Characteristics

The main function of this system is to provide more timely opportunity for someone who manages the rooms to ensure that rooms are fit for habitation, employees are functioning at the desire capacity and that the rooms are energy efficient. This will be achieved by the O-EMA users been able to visualize the humidity, CO_2 and temperature information of the rooms they manage in real-time, daily and regularly. The frequency of the data will be every 5 minutes, 24 hourly and 7 days a week. The system will grant only one type of privilege; user privilege. The O-EMA user will have access to the energy management functions as follows:

- View the room variable indices after login in.
- Navigate to different menu to see changes in the recordings.
- See notification or warning colours and messages on the screen once any of the variables is not within the expected levels needed as was predetermined.
- download daily report of the values.
- If the issues are not yet resolved, the notification messages will continue to show until it is acted upon.
- View on the screen when all variable levels are normal.

G.7 The Solution

Because O-EMA application will largely depend on a third party application programming interface (API) called Netatmo, a 4-tier web application is being proposed to solve the indoor environmental challenges discussed in Section 7.1 which are often encountered in most indoor places for the overall well-being of the occupants.

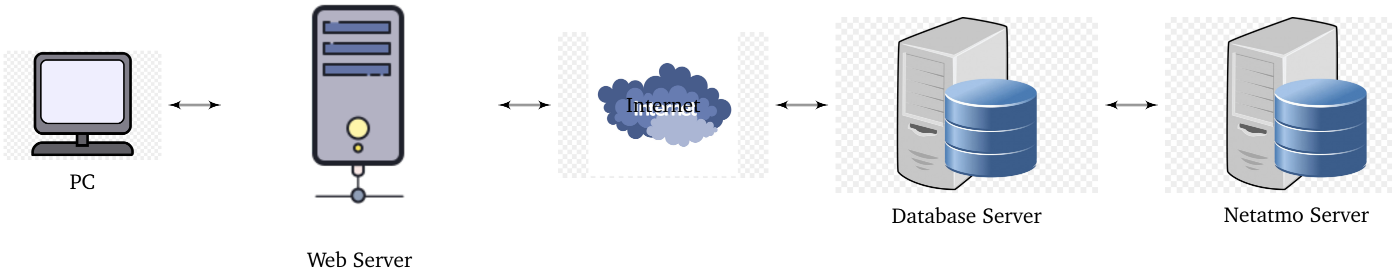
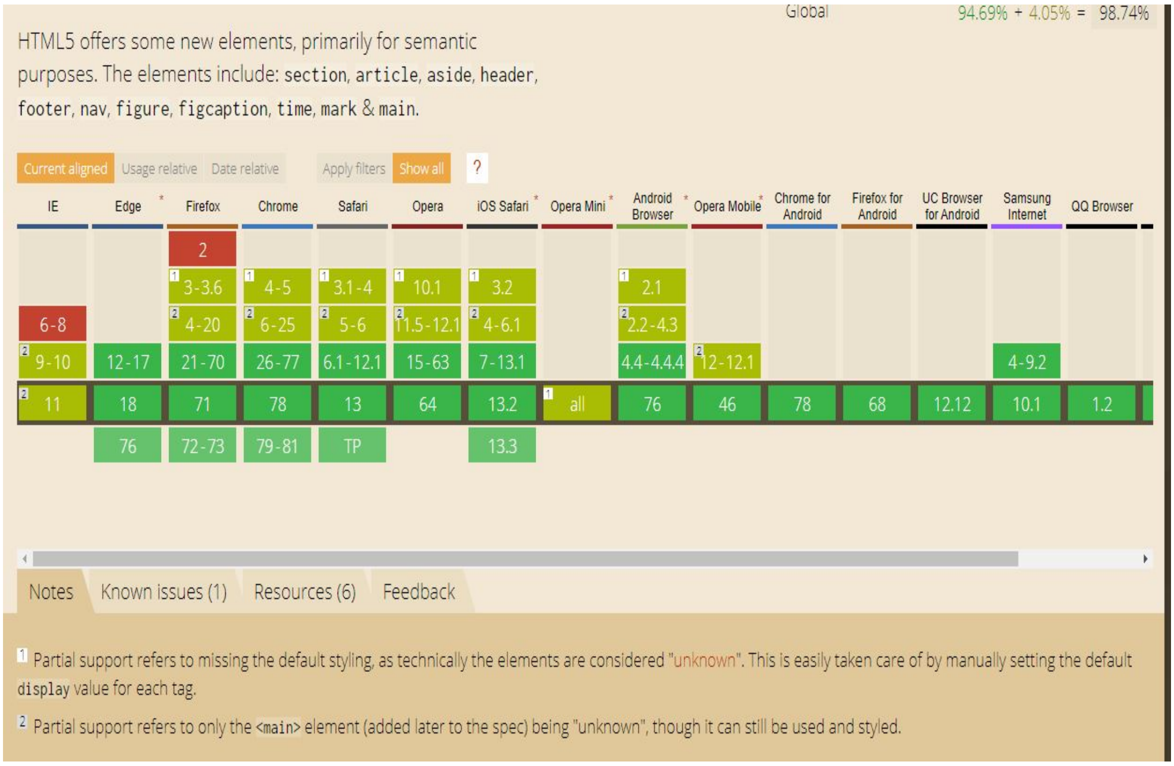


Fig. 11. Proposed 4-tier Architecture for O-EMA

Figure 11 represents the interactions between the 4-tier architecture of the O-EMA. The resource tier (Netatmo server) will be responsible for storing persistence temperature, CO_2 and humidity metadata managed by Netatmo company. The enterprise tier (Database server) will be responsible for hosting the system logics, applying of security rules and retrieving the metadata from the resource tier. The web tier (Web server) will be responsible for handling user request, processing logon request, processing all the functions that will aid in the management of the indoor environmental variables. The client tier (PC) will be responsible for displaying the output pages requested by the user. The internet will be a wide area network (WAN) that interconnects the web server and the database for the O-EMA. Furthermore, the O-EMA will have a backend that will be responsible for computing most of he functions that will be displayed in the frontend. One of those functions is the use of ML GB model discussed in Section 6.4.3 to compute the number of people in each room.

G.8 Assumptions and Dependencies

O-EMA will be developed using HTML5, CSS3, SQL, PHP7 and JavaScript Frain [2012], Yank [2004]. A prerequisite for O-EMA application to work well is to update the browser with the current version regularly. Users should desist from using legacy internet explorer browser because of its compatibility issues with most of the HTML5 syntax. Figure 12 shows the lists of browsers that will be compatible with O-EMA application.
Only the browser version showing green colour will be fully compatible with O-EMA application while the versions showing red and lemon green colours will be incompatible and partially compatible respectively. O-EMA will largely depend on the a third party application known as Netatmo sensor app. As a web application O-EMA can be accessed through the internet via any operating systems such as Windows, Linux and Mac.



Global 94.69% + 4.05% = 98.74%

HTML5 offers some new elements, primarily for semantic purposes. The elements include: `section`, `article`, `aside`, `header`, `footer`, `nav`, `figure`, `figcaption`, `time`, `mark` & `main`.

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser
		2												
		[1] 3-3.6	[1] 4-5	[1] 3.1-4	[1] 10.1	[1] 3.2		[1] 2.1						
6-8		[2] 4-20	[2] 6-25	[2] 5-6	[2] 11.5-12.1	[2] 4-6.1		[2] 2.2-4.3						
[2] 9-10	12-17	21-70	26-77	6.1-12.1	15-63	7-13.1		4.4-4.4.4	[2] 12-12.1				4-9.2	
[2] 11	18	71	78	13	64	13.2	[1] all	76	46	78	68	12.12	10.1	1.2
	76	72-73	79-81	TP		13.3								

[1] Partial support refers to missing the default styling, as technically the elements are considered "unknown". This is easily taken care of by manually setting the default `display` value for each tag.

[2] Partial support refers to only the `<main>` element (added later to the spec) being "unknown", though it can still be used and styled.

Fig. 12. List of compatible browsers that the O-EMA application can function properly
Source: Daveria and Schoors [2019]

Question number	Description
Q1	Answer in the scale of 1-3 Is the user interface design of the dashboard easy to understand? For instance, the way all the room numbers are appearing on the main screen of the dashboard
Q2	Answer in the scale of 1-3 Do the colours showing on the main page of the dashboard give a clue of what room is at risk of safe
Q3	Answer in the scale of 1-3 Are the prediction results shown on the dashboard easy to understand
Q4	Answer in the scale of 1-3 Is the analytic part of the dashboard easy to understand
Q5	Answer in the scale of 1-3 Is the report generation format request clear
Q6	Answer in the scale of 1-3 Are the icons that are used to assist in the navigation of the page clear and intelligible? Eg back to main page, logout
Q7	Answer in the scale of 1-3 Is the sample for report generated by the dashboard clear and easy to understand
Q8	Answer in the scale of 1-3 Is the dashboard user offered useful feedback on the reports generated by the dashboard
Q9	Answer in the scale of 1-3 Are the texts in the dashboard meaningful
Q10	In the scale of 1 to 5, please score this dashboard design based on its usefulness for monitoring rooms. (1: least likely, 2: unlikely, 3: neither, 4: likely, 5: most likely)
Q11	In the scale of 1 to 5, please score this dashboard design based on its usefulness in preventing any future health hazards to room occupants. (1: least likely, 2: unlikely, 3: neither, 4: likely, 5: most likely
Q12	Please add any additional comment regarding the system. Example any suggestion on how the usability of the system can be improved

Table 5: O-EMA Evaluation Questions and its Descriptions

H Interview Guide for the Indoor Environmental Monitoring System Design

H.0.1 Question Description for O-EMA

H.1 Background information and Introduction

The use of sensors to monitor and record indoor environmental parameters are well known in the literature. For example, many wireless environmental sensors such as

Netatmo weather station have shown some degree of accuracy in reporting different parameters of variables such as CO2, temperature, humidity, pressure and noise. Sometimes, due to the strategic importance of a room or office, some indoor environmental parameters are specified at a specific range (such as minimum indoor temperature should be 16OC). These specified ranges are expected to be maintained for proper functioning of that indoor environment. When these specified ranges are not met, there could be human or monetary cost often associated with such problems.

In order to prevent this kind of issue from arising in our indoor environment, a dashboard (web application) is being proposed to help the room occupants to be able to monitor and react as necessary to their indoor environment in real-time. This system will serve several purposes such as; 1) to create different specific parameters for environmental variables, 2) to alert the occupants should any environmental parameter fail to meet the required specifications and 3) to provide reports to the occupants about the current parameters, which will help them to act promptly for their overall well-being.

H.2 Benefits and Important Features for the Environmental Room Monitoring with Software Application

- Of what importance will this environmental room monitoring system be to your organization?
- What variables do you wish to view on the system?
- What design/format do you want to view on the system?
- What are the functional requirements or system features that you want this system to have?
- Do you prefer any colour scheme?
- Do you want any push notification or warnings?
- What other important features do you want this system to have?

H.3 Conclusion of the second interview

Thank you for taking part in this interview. Your answers will be considered during this study. Please print your name and signature in the space below. The voice recorded will be used for the purposes of this study and then disposed of. No real names of the participants will be disclosed.

I Coding Scheme for Indoor Environmental Monitoring System Interview

Categories	Description	Example
Beneficiaries of IEMS	Both P1 and P2 agrees that IEMS will be beneficial to their company if they have more employees making use of an open office space in their company. They however believed that the IEMS will more likely be beneficial to landlords and big companies with larger businesses.	P2: Talking of Swarm, our size I think is probably of limited .. for constantly to always be on dashboard however I can see the benefit, I have two scenario on my head. One is for larger business who have bigger office and the second is for landlords.
		P2: ... Also it will be good where you have about 50/60 people around an office that's where I can see the use case of this.
		P1: Yeah Perhaps not for us
		P2: ... That's said if it is something for us to have easy access..for us to do...at no cost then that will be useful for us as well.
Benefits of the IEMS to landlords	P2 agrees that landlords could use IEMS to manage their houses, offices or rooms they build in accordance with the legislation.	P2: ...So if you imagine landlord of this building who will want to monitor all these things especially for new legislation coming in for landlords.
IEMS impact in the control of moulds and damps in a building:	P1 and P2 also believed that if the software system could generate weekly report of these indoor variables, it will be used to identify rooms that are prone to cause sick building syndromes which will ensure that the owners acts when the need arises	P1: The historic level, maybe a period of a week and being able to generate a report from there.
		P2: .... there is a wall damp ... as we speak it has been repaired but not has been painted, before we moved in that was bad damp. I think it goes back to the landlord discussion again.
		P2: The report will be good. Because again it goes back to this companies landlord...
		P2: .... for us to have a meeting with the report 15:31 saying here is the reading of the last

		month... or whatever that will be...
		P2: Think about a sort of report, imagine we have a system,.. maybe something in the building that we can't but we could give that report to a landlord and say look there is a problem here and you need to fix this two problems 16:10 as a landlord and here is the report that proves what the last three months ... or he could take it to finance to sign off the money for the work we have done if we do the work by ourselves.
IEMS potential in provision of energy efficient and good working environment:	P2 believes that if landlords could have access to IEMS that it will help them in ensuring that their properties are providing energy efficient and good working environments for their tenants.	P2: landlords are getting more legislation around being energy efficient and more focus on providing good working environment for employees.
Push notifications or warning as a functional requirement:	P2 agrees that in addition to the report, the IEMS should be able to send a warning to the user via email of any potential problem.	P1: if you could get an email to go and turn the report.
		P2: I think having something that could set the threshold, or you could brainstorm your findings and find out to say that the CO_2 with this colour... and you could send an email
Login as a functional requirement	P2 believes that single login will be enough for IEMS except if there are important security task that the software will be performing.	P2: I think maybe ...logins if people will use it, but I think if you don't have to install stuff single account login ...
Report generation as a Functional requirement of IEMS	P1 and P2 agrees that the IEMS should be able to generate report and a graph of the environmental variables either weekly or monthly.	P1: ...the historic level, maybe a period of a week and being able to generate a report from there.
		P2: The report will be good...In summary of the dashboard there will be a graph and that will be quite interesting.

		P2: Think about a sort of report, imagine we have a system,.. maybe something in the building that we can't but we could give that report to a landlord and say look there is a problem here and you need to fix this two problems as a landlord and here is the report that proves what the last three months... or he could take it to finance to sign off the money for the work we have done if we did the work by ourselves
Design pattern as a functional requirement of IEMS	P1 and P2 agrees that IEMS should have a map-like design where all the rooms in a floor are connected to one page with different situations of each room showing on that front page.	P2: So I think having a dashboard such as this will be cool...We gone out to draw the four corners of our office and that is one of our hot spots
		P2: ...You could show there are six people sitting in that room....and it is a visual representation.
		P1: Again, in the view of the office plan....
Environmental variables current levels as a Functional requirements of IEMS	P1 agrees that the IEMS should be able to show in real-time the current levels of each variables it measures. P2 agrees that CO_2 levels is a very important feature for IEMS to have because it will help them to check the productivity levels of their employees	P1: the current levels as well
		P2: For example we could say that certain levels of CO_2 we use it to identify the productivity depth assume there will be further studies. ... you can pop that up and say that room there the development room , the CO_2 there has been high and as a management team we can go actually, .. maybe people are complaining of a lot of headaches or people not

		productive or whatever...that's the kind of stuff I think...
Traffic colours as different indicators:	P1 and P2 believes that the warning signs on the dashboard should be indicated with the kind of traffic colours used by the driver and vehicle licensing agency (DVLA)	P1: We need a kind of values with traffic light system colours , you know what is too high, too low..
		P2: you imagine that there is a TV in the wall that room goes amber ..you know that there is something to deal with that, that's the kind of thing in my head ..
Colour scheme as non-functional requirement:	P1 and P2 agrees that the colour of the dashboard is not really a priority as long as the texts are eligible and easily readable.	P1: but as developers we go for darker I struggle to understand the reason.
		P2: I don't think colour is such a big deal...Just any colour that can show.. I think you should keep it simple, keep it light... and simple read
Non-functional requirement:	P2 agrees that the dashboard should be easy to use and navigate with sensible meaning for words and adequate font size	P2: maybe we should be looking at non functional requirement which should be easily readable, easy to navigate, words should be as possible sensible, thinks about the font sizes ... you know colour contrast, you are just playing colours ...
Platform and operating system:	P2 believes that any operating system and platform is okay but that IEMS should be designed to work on iPad	P2: if do those dashboard to have something that is responsive that could work on iPad and even some web app you can load up an iPad....we have built dashbaord application in the past, it worked on iPad but not on the phone but it was displaying a graph.

J Information Sheet and Consent Form for IEMS

Stakeholders’ Interview Sheet for Environmental Room Monitoring System Design Specification.

Instructions:

1) This interview will be recorded for transcription purposes

Organization Name: ……………………………………………………………...

Address: ……………………………………………………………………………..

City, Post-code: …………………………………………………………………

Interview Date: …………………………………………………

Background Information:

The use of sensors to monitor and record indoor environmental parameters are well known in the literature. For example, many wireless environmental sensors such as Netatmo weather station have shown some degree of accuracy in reporting different parameters of variables such as CO2, temperature, humidity, pressure and noise. Sometimes, due to the strategic importance of a room or office, some indoor environmental parameters are specified at a specific range (such as minimum indoor temperature should be 16^{O}C). These specified ranges are expected to be maintained for proper functioning of that indoor environment. When these specified ranges are not met, there could be human or monetary cost often associated with such problems.

In order to prevent this kind of issue from arising in our indoor environment, a dashboard (web application) is being proposed to help the room occupants to be able to monitor and react as necessary to their indoor environment in real-time. This system will serve several purposes such as; 1) to create different specific parameters for environmental variables, 2) to alert the occupants should any environmental parameter fail to meet the required specifications and 3) to provide reports to the occupants about the current parameters, which will help them to act promptly for their overall well-being.

Reason for the Interview:

The main aim of this interview is to find out from the interested stakeholders (the people directly responsible for managing these rooms), whether there are any requirements of interest to them, that needs to be included in the design phase of the environmental monitoring system. The answers to the interview questions will be interpreted verbatim and it will serve as a means of capturing individual stakeholder’s interests and concerns before the writing of the software requirement specification document.

How will the Interview be conducted?

This interview will be in form of an open-ended questions and answers. Where the participants will be engaged in the following topics. Answers will be audio recorded, transcribed and interpreted.

The main topics of the discussion in this interview section will be the following:

1) Of what importance will this environmental room monitoring system be to your organization?

2) What variables do you wish to view on the system?

3) What design/format do you want to view on the system?

4) What are the functional requirements or system features that you want this system to have?

5) Do you prefer any colour scheme?

6) Do you want any push notification or warnings?

7) What other important features do you want this system to have?

Thank you for taking part in this interview. Your answers will be considered during this study. Please print your name and signature in the space below. The voice recorded will be used for the purposes of this study and then disposed of. No real names of the participants will be disclosed.

(PRINT NAME)	
Signature of Participant:	Date:

K The Evaluation User Guide for IEMS

Stakeholders' Interview Sheet for Environmental Room Monitoring System Design Specification.

Instructions:

1) This interview will be recorded for transcription purposes

Organization Name: ………………………………………………………………...

Address: ……………………………………………………………………………..

City, Post-code: ………………………………………………………………………

Interview Date: …………………………………………………

Background Information:

The use of sensors to monitor and record indoor environmental parameters are well known in the literature. For example, many wireless environmental sensors such as Netatmo weather station have shown some degree of accuracy in reporting different parameters of variables such as CO2, temperature, humidity, pressure and noise. Sometimes, due to the strategic importance of a room or office, some indoor environmental parameters are specified at a specific range (such as minimum indoor temperature should be 16^{O}C). These specified ranges are expected to be maintained for proper functioning of that indoor environment. When these specified ranges are not met, there could be human or monetary cost often associated with such problems.

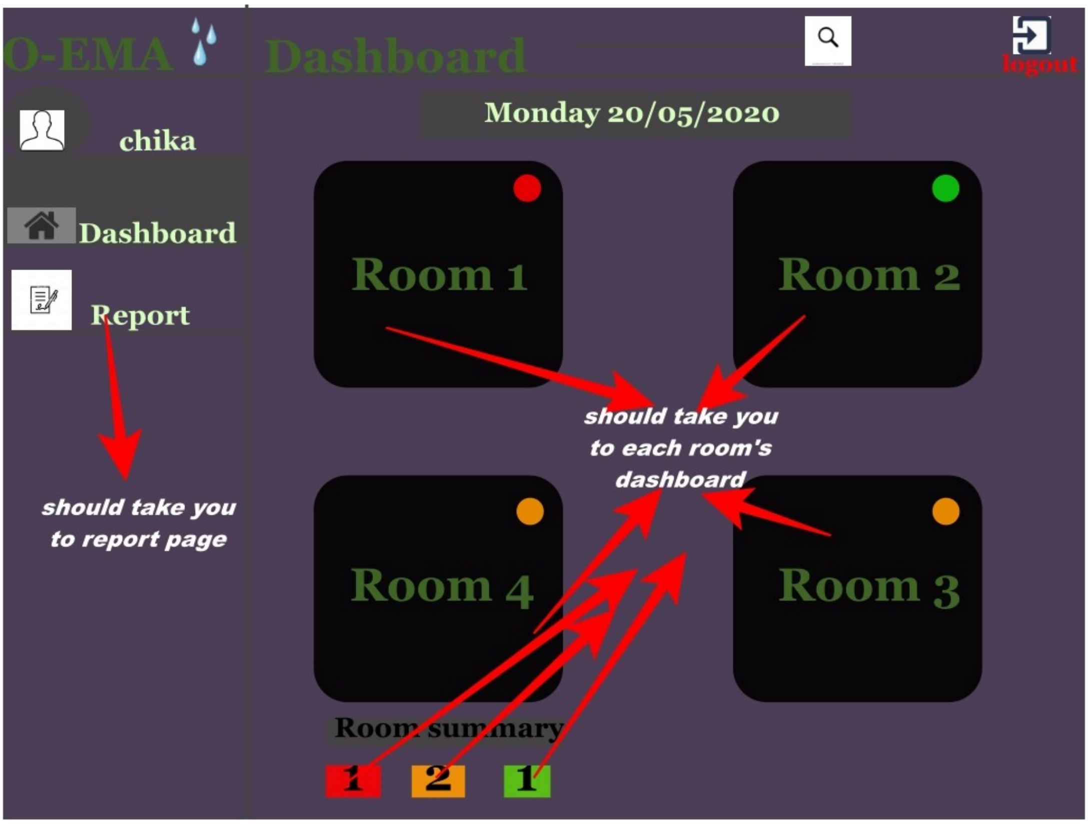
In order to prevent this kind of issue from arising in our indoor environment, a dashboard (web application) is being proposed to help the room occupants to be able to monitor and react as necessary to their indoor environment in real-time. This system will serve several purposes such as; 1) to create different specific parameters for environmental variables, 2) to alert the occupants should any environmental parameter fail to meet the required specifications and 3) to provide reports to the occupants about the current parameters, which will help them to act promptly for their overall well-being.

Reason for the Interview:

The main aim of this interview is to find out from the interested stakeholders (the people directly responsible for managing these rooms), whether there are any requirements of interest to them, that needs to be included in the design phase of the environmental monitoring system. The answers to the interview questions will be interpreted verbatim and it will serve as a means of capturing individual stakeholder's interests and concerns before the writing of the software requirement specification document.

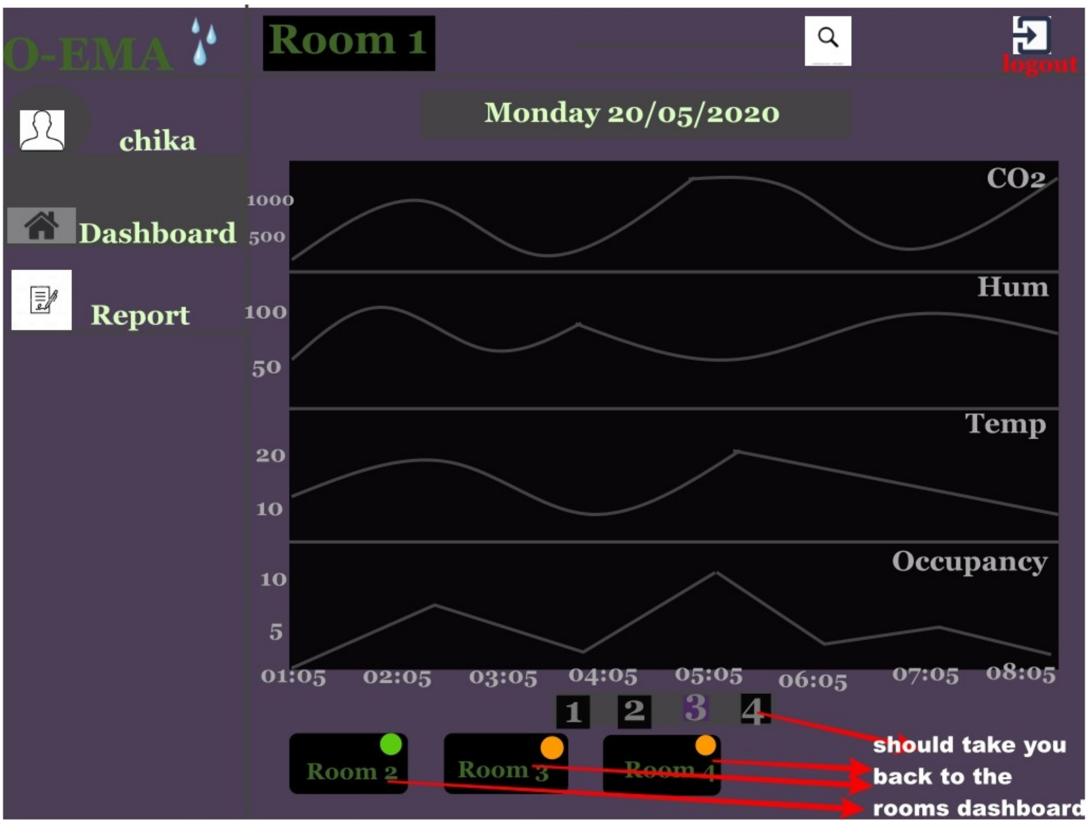
How will the Interview be conducted?

This interview will be in form of an open-ended questions and answers. Where the participants will be engaged in the following topics. Answers will be audio recorded, transcribed and interpreted.

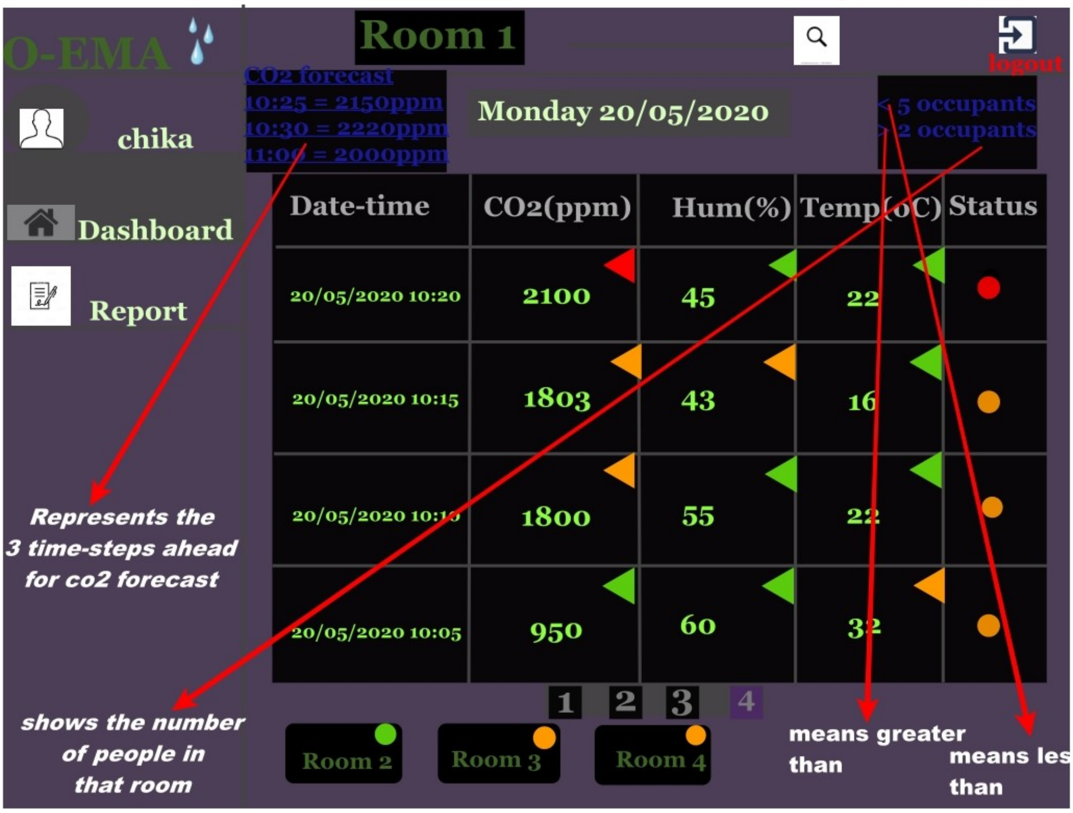


should take you to each room's dashboard

should take you to report page



should take you back to the rooms dashboard



CO2 forecast
10:25 = 2150ppm
10:30 = 2220ppm
11:00 = 2000ppm

< 5 occupants
> 2 occupants

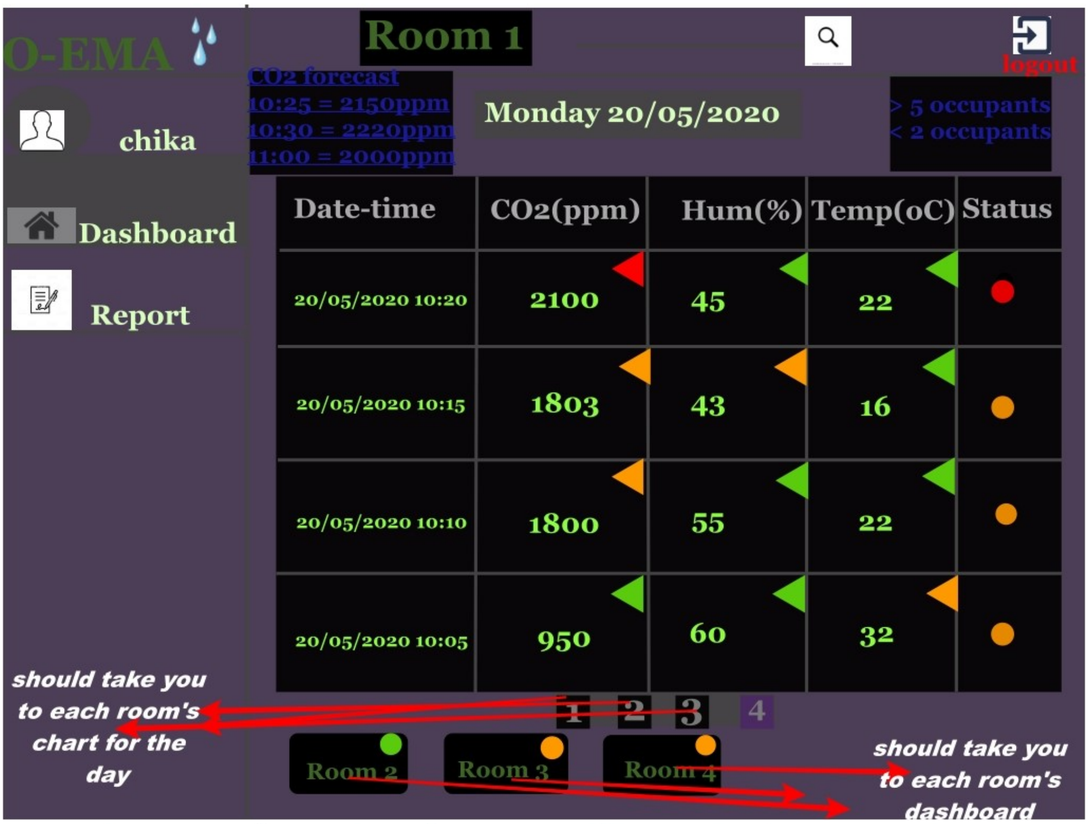
Date-time	CO2(ppm)	Hum(%)	Temp(oC)	Status
20/05/2020 10:20	2100	45	22	●
20/05/2020 10:15	1803	43	16	●
20/05/2020 10:10	1800	55	22	●
20/05/2020 10:05	950	60	32	●

Represents the 3 time-steps ahead for co2 forecast

shows the number of people in that room

means greater than

means les than



Room 1

CO2 forecast
10:25 = 2150ppm
10:30 = 2220ppm
11:00 = 2000ppm

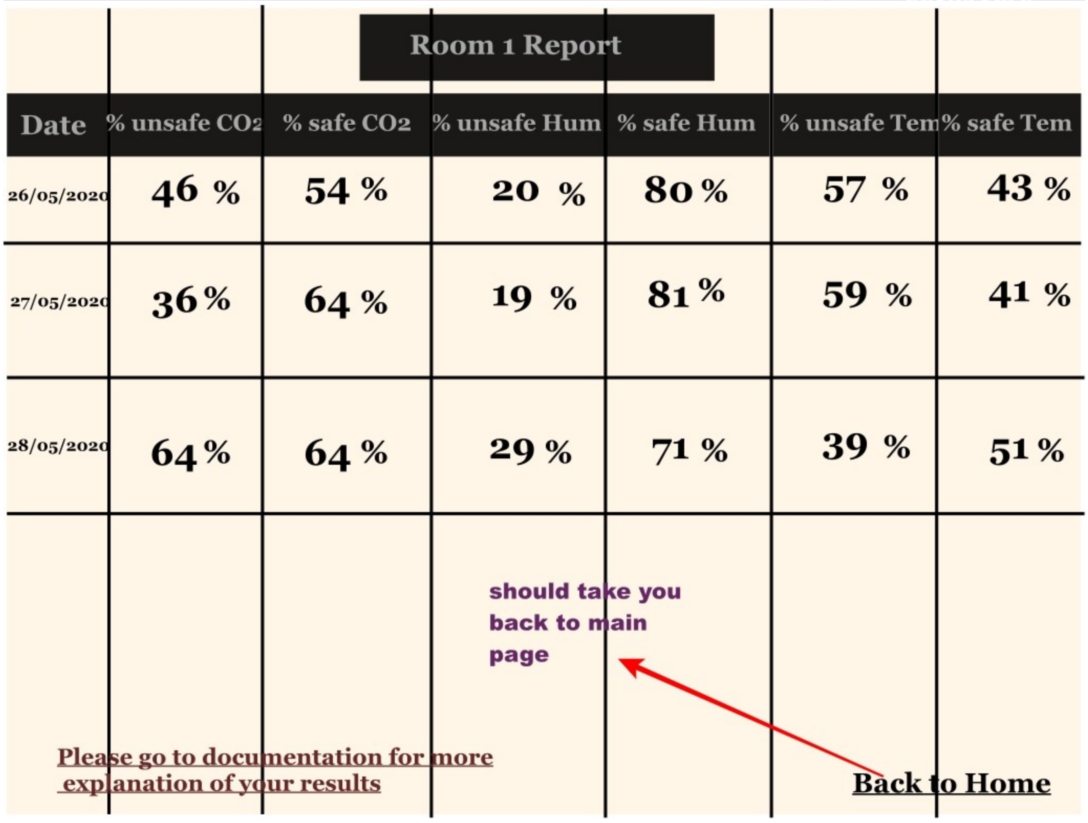
Monday 20/05/2020

> 5 occupants
< 2 occupants

Date-time	CO2(ppm)	Hum(%)	Temp(oC)	Status
20/05/2020 10:20	2100 ◀	45 ◀	22 ◀	●
20/05/2020 10:15	1803 ◀	43 ◀	16 ◀	●
20/05/2020 10:10	1800 ◀	55 ◀	22 ◀	●
20/05/2020 10:05	950 ◀	60 ◀	32 ◀	●

should take you to each room's chart for the day

should take you to each room's dashboard



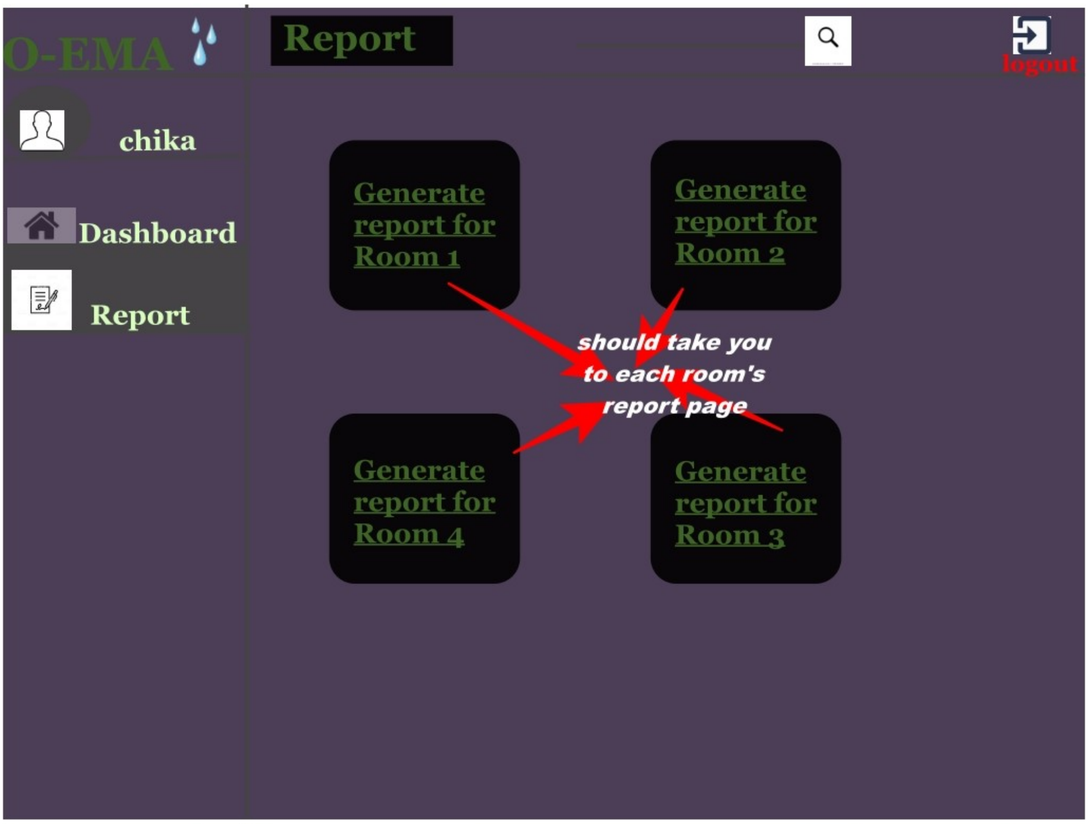
Room 1 Report

Date	% unsafe CO2	% safe CO2	% unsafe Hum	% safe Hum	% unsafe Tem	% safe Tem
26/05/2020	46 %	54 %	20 %	80 %	57 %	43 %
27/05/2020	36 %	64 %	19 %	81 %	59 %	41 %
28/05/2020	64 %	64 %	29 %	71 %	39 %	51 %

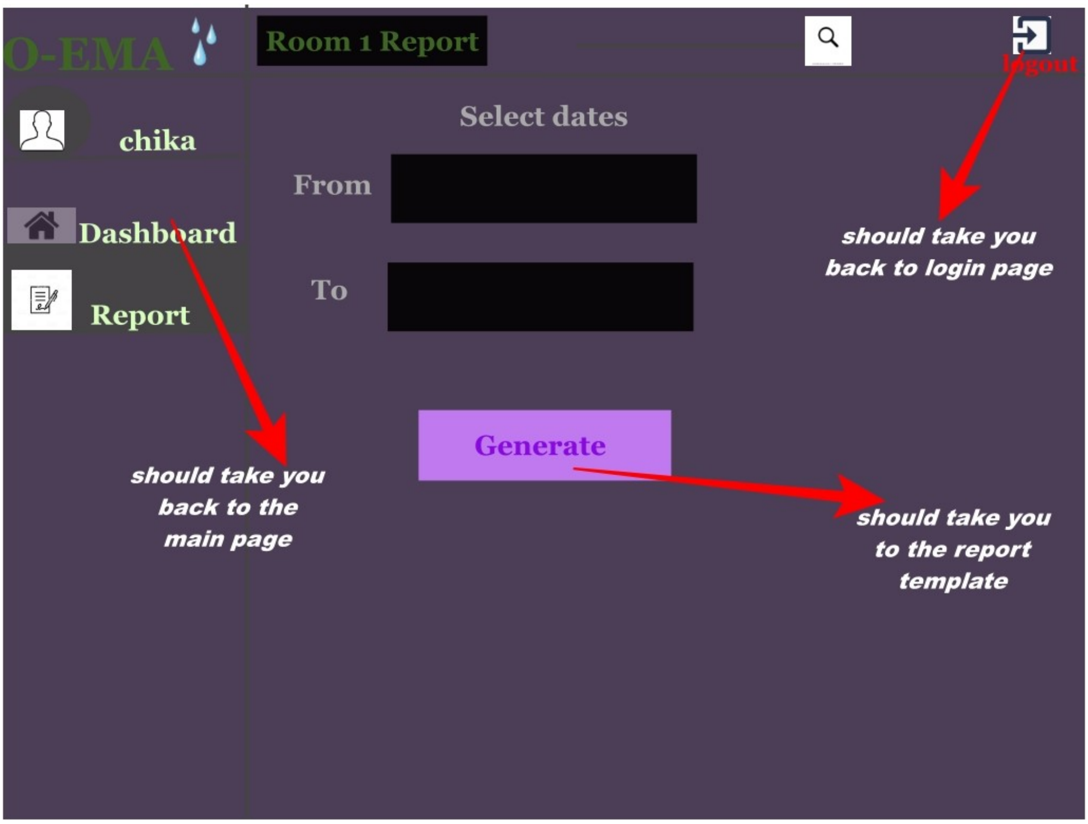
should take you back to main page

Please go to documentation for more explanation of your results

Back to Home


O-EMA
Report
logout
chika
Dashboard
Report
Generate report for Room 1
Generate report for Room 2
Generate report for Room 4
Generate report for Room 3

should take you to each room's report page



should take you back to login page

should take you back to the main page

should take you to the report template

L The Participants Dashboard Evaluation Questionnaire

Participants' Interview Sheet for Dashboard Evaluation

Organization Name: …………………………………………………………………...

Address: ………………………………………………………………………………..

City, Post-code: ……………………………………………………………………….

Interview Date: …………………………………………………

Background Information:

This study will evaluate the usability and the functionality of the high fidelity protype room monitoring system. This system evaluation will involve the following:

- To check if the format used in displaying the prediction result on the dashboard is good and easy to understand.
- To check if the design plan of the dashboard is simple and well understood.
- To know if the colours being displayed by the variables' readings are meaningful.
- To check if the format for the report generation is meaningful.
- To check if the report generated by the system will be helpful to the dashboards.
- To give the overall rating on the usefulness, usability, and the design of the dashboard.

As one of the ways of evaluating the usability of a software system, it is therefore important to re-interview those participants that took part in the initial interviews that were conducted before the commencement of this research study. The participants will be asked if their interests were met in the final output and results of the studies. Their answers will serve as information gatherings which will help to in the testing and evaluation of the designed dashboard.

Therefore, because the system is not yet functional, this interview will only concentrate of the main functions of the dashboard as it relates to the answers the participants provided during the initial interviews.

A room monitoring dashboard system called Occupancy-Environmental Monitoring Application (O-EMA) has been designed. The backend of this system uses machine learning method to predict the number of people in that room and to predict indoor CO_2 in advance. In order to evaluate the usability of this system, please answer the following questions:

	Evaluation Questions for O-EMA dashboard	**Scales to answer:** 1: Somewhat easy to understand

		2: Easy to understand 3: Very easy to understand
1	**Answer in the scale of 1-3** Is the user interface design of the dashboard easy to understand? For instance, the way all the room numbers are appearing on the main screen of the dashboard.	
2	**Answer in the scale of 1-3** Do the colours showing on the main page of the dashboard give a clue of what room is at risk of safe?	
3	**Answer in the scale of 1-3** Are the prediction results shown on the dashboard easy to understand?	
4	**Answer in the scale of 1-3** Is the analytic part of the dashboard easy to understand?	
5	**Answer in the scale of 1-3** Is the report generation format request clear?	
6	**Answer in the scale of 1-3** Are the icons that are used to assist in the navigation of the page clear and intelligible? Eg back to main page, logout etc..	
7	**Answer in the scale of 1-3** Is the sample for report generated by the dashboard clear and easy to understand?	
8	**Answer in the scale of 1-3** Is the dashboard user offered useful feedback on the reports generated by the dashboard?	
9	**Answer in the scale of 1-3**	

	Are the texts in the dashboard meaningful?	
10	In the scale of 1 to 5, please score this dashboard design based on its usefulness for monitoring rooms. **(1: least likely, 2: unlikely, 3: neither, 4: likely, 5: most likely**.	
11	In the scale of 1 to 5, please score this dashboard design based on its usefulness in preventing any future health hazards to room occupants. **(1: least likely, 2: unlikely, 3: neither, 4: likely, 5: most likely.**	

12) Please add any additional comment regarding the system. Example any suggestion on how the usability of the system can be improved.

Thank you for taking part in this online interview. Your answers will be considered during the evaluation of the thesis study. Please print your name and signature in the space below. The data recorded will be used for the purposes of this study and then disposed of. No real names of the participants will be disclosed.

(PRINT NAME)	
Signature of Participant:	Date:

L.1 Interview Schedule for O-EMA Design

Table 6: P1 and P2 interview schedule for O-EMA design

Participants	Status	Interview Date
P1	Swarm Staff	27/02/2020
P2	Swarm Staff	27/02/2020

M Citation Table for Literature Review

Title	Database	Abstract summary	Year of publication
Indoor air quality control of HVAC system	Google scholar	control strategy used for the control of ventilation system (HVAC)	2010
Time series prediction of CO2, TVOC and HCHO based on machine learning at different sampling points	Google scholar	Prediction of indoor co2 and VOCs	2018
Ventilation monitoring and control system for high rise historical buildings	Google scholar	Use of graphical interface to display indoor environmental factors such as temperature, humidity and CO_2	2017
A Scalable Room Occupancy Prediction with Transferable Time Series Decomposition of CO2 Sensor Data	ACM	room occupancy prediction (binary and counting) with indoor co2 concentration and timeseries method. This uses netatmo sensor	2018
A Multi-sensor Based Occupancy Estimation Model for Supporting Demand Driven HVAC Operations	ACM	occupancy estimation using non-intrusive sensor data like temp, humidity, co2,light etc..	2012
Indoor occupancy estimation from carbon dioxide concentration	Google scholar	Room occupancy estimation with smoothed indoor co2 data for energy efficiency	2016
Real-time occupancy estimation using environmental parameters	Google scholar	Occupancy estimation for the control of mechanical ventilation and Air conditioning	2015
Building occupancy estimation and detection: A review	Google scholar	A complete review of different methods of estimating room occupancy and comparison of best methods	2018

Uncertainties in neural network model based on carbon dioxide concentration for occupancy estimation	Google scholar	Neural network was used to estimate room occupancy based on co2 data	2017
Smell Pittsburgh: Community-Empowered Mobile Smell Reporting System	ACM	Local community with poor air pollution are visualized on the map	2019
Design and application of a VOC-monitoring system based on a ZigBee wireless sensor network	IEEE	Sensor network and internode data reception for the monitoring of indoor pollutant.	2014
Environmental monitoring systems: A review	IEEE	Review of environmental monitoring system	2012
Environmental monitoring system	Google scholar	Use of plurarity sensor for the measurement and control of the environment	2004
AirCloud: a cloud-based air-quality monitoring system for everyone	ACM	Low cost PM2.5 air quality monitoring system.	2014
DA-HOC: semi-supervised domain adaptation for room occupancy prediction using CO2 sensor data	Google scholar	Domain based occupancy counter	2017
Using machine learning techniques for occupancy-prediction-based cooling control in office buildings	Google scholar	Analysis of occupants stochastic behaivour for demand driven energy supply.	2018

MAQS: A personalized mobile sensing system for indoor air quality monitoring.	Google scholar	personal system for sensing IAQ with analytics engine	2011
InAir: sharing indoor air quality measurements and visualizations	Google scholar	A tool for charing measurements of IAQ on social network	2010
Estimating occupancy in heterogeneous sensor environment	Google scholar	Using machine learning to estimate occupancy	2016
Advances in research and applications of energy-related occupant behavior in buildings	Google scholar	Energy related occupants behaiviour in the building	2016
Occupants' window opening behaviour: A literature review of factors influencing occupant behaviour and models	Google scholar	Simple task of opening and closing of windows by the occupants and how it helps with energy efficiency.	2012
A sort of CO2 concentration monitoring system based on energy efficiency for terminal	Google scholar	CO2 Monitoring sytem used at a terminal for passengers	2016
Indoor air quality control for energy-efficient buildings using CO 2 predictive model	Google scholar	Control system for indoor air quality control	2012

Outcome	Publisher	Limitation
Indoor CO2 was predicted using occuancy data	IEEE	No advance prediction in terms of time was done and ML was not used.
Prediction of indoor co2 with SVM and nueral network with 86% accuracy. Mean Absolute Percentage Error of 1.87%	Elsevier	Though this method reduced mathematical cost, but a lot of calibration is needed to achiebve the desired result
real-time information from exhaust fan are displayed by the GUI developed with visual studio	IEEE	No prediction or machine learning analysis was done. No backend prediction model was used in the system.
SVR regression technique and seasonal decompositionfor human occupancy counting (SD-HOC) was used	ACM	MATLAB and R were used. Binary counting achieved between 56 - 63% of accuracy. While occupancy counting achieved between 50-53% using SVR method.
RBF neural net was used to achieve accuracy of about 66%. RMSE was 2.310	ACM	The model is not universal for every offices. New calibration are needed for new test.
feature scaled extreme machine learning (FS-EML) and was used to achive accuracy score of 94% with tolerance of 4 occupants	Elsevier	Not a universal model (the ML model is room dependent). Different office for different model calibration
Uses extreme learning machine (ELM) achieves accuracy of 74% using co2, temp, relative humidity	Elsevier	Analysis of larger values are needed
	Elsevier	Performance is still far froms satisfactory, hence future work needs to be done. Placement of sensors has to be explored as well as combination of sensors. For more reliable result and coverage.

Demand control ventilation using occupancy estimation	Springer	Lack of data sampling interval for the used data
This visualization includes air quality data and push notification being sent to the local authorities . Prediction of upcoming smell event.	ACM	Unreliable machine learning method for the smell prediction result
End device sensor that communicates with the computer for VOCs monitoring. The measured data are displayed on a computer monitor.	IEEE	No prediction or machine learning analysis was done. No backend prediction model was used in the system.
A review of different approaches and methods used to monitor indoor environment	IEEE	Most systems cannot measure both air quality and environmental parameters simultaneously. No energy efficient system. Robust and easy to use systems are still needed
System for clean room operation and contamination control. Patents and standards used in various studies for designing and developing monitoring systems	Google books	The study in research dependent.
The system uses analytics engine that is comnnected to a frontend that monitors the particulate matter. Application systems were also built on top of it for easy monitoring	ACM	Complex system.
The trained data from a small room was able to predict room occupancy of a larger room if the model is adapted to a larger room	ACM	The model has not been tested in other locations
Improving HVAC efficiency via occupancy behaiviour such as demand driven control strategy. This reduces the human input while controling the energy input.	Elsevier	Only identifies when occupants will be present in a room

Uses co2 sensor to measure indoor air quality and allows for data and information sharing.	ACM	No prediction analysis of interested variable and occupancy data was made available for overall information of the room.
This tool includes vizualization screen for displaying line gaphs of the data measured in one's household	ACM	No analytical engine built in this tool
Approximates the number of people that might be in the room using co2 sensors, microphone etc	Elsevier	No interpretable model and the model is room dependent
Occupants behaiviour in the control of energy usage	Elsevier	Lack of model verification
	Elsevier	Lack of explainable ML model
The CO2 monitoring system was used to improve passengers air quality	IEEE	No predictive model
	IEEE	There was only one step forecast of CO2

N Interview Guide for the Gradient Boosting Model Evaluation

N.1 Background information and Introduction

An initial study that estimates the room occupancy with environmental variables was conducted. This study will evaluate the interpretability of the ensemble machine learning model used for the room occupancy estimation study earlier conducted in this research. By interpretability, it means to check if non-expert participants can be able to understand how the Gradient Boosting (GB) ensemble machine learning model used for the room occupancy estimation, makes its prediction based on the input (environmental) variables. This model interpretability evaluation will involve the following:

- To check if the participants can make at least five predictions (based on prediction interval approach used in the quantitative analysis) if some specific input variables are given.

- To ask the participants to choose from the two interpretable models which one is easier for them to understand how the room occupancy model works.

As one of the ways of evaluating the interpretability of the black-box of a machine learning model, it is therefore important to obtain first-hand information from non-experts themselves. The information that will be obtained from this study will serve as information gatherings which will help to evaluate the interpretability of the model.

N.2 Interview Schedule for Room Occupancy study

Table 7: Participants interview schedule for Study Evaluation

Participants	Status	Interview Date
E1	Student	22/06/2020
E2	Student	23/06/2020
E3	Student	27/02/2020
E4	Student	25/06/2020
E5	Student	24/08/2020
E6	Student	29/08/2020

N.3 Questions for Model Interpretability Evaluation

1) Please fill in the following table for SHapley Additive exPlanations (SHAP) model:

2) Based on the answers you gave in the two tables above, which of the interpretable models (SHAP or LIME) made it easier for you to understand how the GB model works.

CO2	Humidity	Pressure	Noise	Temperature	Guess-Occupancy	Occupancy-actual

N.4 Statistical data

1. Organization Name
2. Address
3. City, Post-code
4. interview Date

O Information Sheet and Consent Form for Gradient Boosting Model Evaluation

Dashboard and Room Occupancy Model Evaluation User Study.

1) **Name of department: Department of Computer and Information Science**
Title of the study: Dashboard and Room Occupancy Model Evaluation User Study.

Introduction:
My name is Chika Ugwuanyi, I am a PhD student at the University of Strathclyde.

What is the purpose of this investigation?

There are three studies conducted as part of this research.

These studies are:

1) Advanced indoor CO_2 prediction with environmental variables,

2) Room occupancy estimation with environmental variables and

3) Prototype design of room monitoring system.

Before the commencement of these three studies, semi-structured interviews were conducted. The interview findings from the transcribed interview data were considered during the analysis and the design of the three studies.

Therefore, this study (Part 3 above) will now focus on user testing and evaluation of the final and room monitoring system.

The two main reasons for this study are as follows:

1) To evaluate the usability and the functionality of the high fidelity protype room monitoring system. This system evaluation will explore the following:

- To check if the format used in displaying the prediction result on the dashboard is good and easy to understand.
- To check if the design plan of the dashboard is simple and well understood.
- To know if the colours being displayed by the variables' readings are meaningful.
- To check if the format for the report generation is meaningful.
- To check if the report generated by the system will be helpful to the user of the system.
- To give the overall rating on the usefulness, usability, and the design of the dashboard.

2) The second reason for this study is to evaluate the interpretability of the ensemble machine learning model used for the room occupancy estimation. By interpretability, we mean to check if the non-expert participants can be able to understand how the Gradient Boosting ensemble machine learning model, used for the room occupancy estimation, makes its prediction based on the input (environmental) variables. Therefore, this model interpretability evaluation will explore the following:

- To check if the participants can make at least five predictions (based on prediction interval approach used in the quantitative analysis) if some specific input variables are given.
- To ask the participants to choose from the two interpretable models which one is easier for them to understand how the room occupancy model works.

These two evaluations will involve two different sets of people namely, 1) the participants for the system evaluation will be the same participants that took part in the semi-structured interviews that was carried out before the commencement of the quantitative studies and 2) the participants for the model evaluation can be any adult.

Do you have to take part?

No. It is not compulsory that you participate in this online interview, if you wish to take part and later decide not to be involved, it is not going to affect the way you would be treated or regarded. In other words, participation is voluntary.

What will happen in this project?

1) We will conduct an online interview for the two evaluations. This interview should not last more than one hour. The online interviews will be via video conferencing or video calls of any suitable online media such as Skype or Zoom.
2) You will be sent an information sheet containing a consent form where you will indicate interest by signing of the consent form.
3) The media for the online interview will be chosen by you depending on what is most suitable for you.
4) For the system evaluation, an interactive online link (InVision) for the dashboard design will be sent to you via your email address.
5) Some annotated screenshots of the dashboard and a script will be sent to you in addition as a user guide.
6) The interviewer will explain (with the help of the screenshots) what the system was meant to do to you during the video call. After the explanations, you will be asked to evaluate the usability and the functionality of the system based on some functions and appearance of the dashboard.
7) For the model evaluation, no link will be sent to you. Rather, during the video call, the model will be explained to you. After which you will be asked to make 5 predictions yourself based on your understanding of the mode and the values that will be shown to you. In addition, you will be asked to comment on the understandability of each of the interpretable models.

Why have you been invited to take part?

You are invited to take part in this investigation either as a someone that initially participated in the previous semi-structured interviews conducted in the early part of this research or as someone who might be interested in knowing how the black box model of machine learning works.

There is no special skill needed for you to participate. No screening procedure is required. No identifiable information will be recorded. Your work environment will not change because of this study directly, but ongoing findings might contribute to improving the overall quality of your working environment.

What are the potential risks to you in taking part?

There is no potential risk associated with this study. Firstly, the video call will only record your answers to the questions you were asked. Your photographic images will not be taken or stored during this video call. None of your personal data will be recorded during this study. The voice recording will be stored on the personal H-drive of the University server. The storage will be deleted immediately at the end of this PhD research.

What happens to the information in the project?

The voice data recorded on computer device will be used for the purposes of this study and then disposed of. No real names of the participants will be disclosed during the analysis.

The University of Strathclyde has a Data Protection Policy which sets out the roles and responsibilities in relation to data protection within the University.

All the data obtained will be protected based on the core principles of University of Strathclyde's GDPR as stated in this link https://www.strath.ac.uk/professionalservices/media/ps/strategyandpolicy/GDPR_Principles_Poster.pdf .

The guidance on how long to retain the audio records is stated on the University of Strathclyde information and records management guidance notes via the link below

https://www.strath.ac.uk/media/ps/cs/foi/recordsmanagement/Information_and_Records_Guidance_9_Retention_and_Disposal_v2.1.pdf.

The audio record will be held for maximum of 1 year or at the end of the PhD research, after which it will be deleted/destroyed from the secured server where it is stored.

Thank you for reading this information – please ask any questions if you are unsure about what is written here.

What happens next?

If you are happy to participate, the consent form for you to sign and confirm is on the next page of this document.

At the end of the investigation, we will include the results of our findings in the research thesis.

If you are not interested thank you for your attention.

Researcher contact details:

Name is Chika Ugwuanyi, Email: chika.ugwuanyi@strath.ac.uk

My contact details: Livingstone Tower, room 1206 Telephone: 01415483705
University of Strathclyde

Chief Investigator details:

1) Dr Marilyn Lennon
   Senior Lecturer Department of Computer and Information Science

University of Strathclyde
Rm 1311a Livingstone Tower

Email marlin.lennon@strath.ac.uk, Telephone: 01415483098

2) Dr Richard **Bellingham**
Director Institute for Future Cities, Senior Research Fellow
Strathclyde Business School
University of Strathclyde

This investigation was granted ethical approval by the Department of Computer and Information Science Ethics Committee.

If you have any questions/concerns, during or after the investigation, or wish to contact an independent person to whom any questions may be directed or further information may be sought from, please contact:

Department of Computer and Information Science Ethics Committee
University of Strathclyde
Livingstone Tower
26 Richmond street
Glasgow
G1 1XH

Telephone: 0141 548 3189
Email: ethics@cis.strath.ac.uk

University of
Strathclyde
Glasgow

Consent Form for Dashboard and Room Occupancy Model Evaluation.

Name of department: Department of Computer and Information Science.
Title of the study: Dashboard and Room Occupancy Model Evaluation User Study.

- I confirm that I have read and understood the information sheet for the above project and the researcher has answered any queries to my satisfaction.
- I understand that my participation is voluntary and that I am free to withdraw from the project at any time, up to the point of completion, without having to give a reason and without any consequences. If I exercise my right to withdraw and I do not want my data to be used, any data which have been collected from me will be destroyed.
- I understand that I can withdraw from the study any personal data (i.e. data which identify me personally) at any time.
- I understand that anonymised data (i.e. .data which do not identify me personally) cannot be withdrawn once they have been included in the thesis.
- I understand that any information recorded in the investigation will remain confidential and no information that identifies me will be made publicly available.
- I consent to be a participant in the project

(PRINT NAME)	
Signature of Participant:	Date:

P Python Codes for Indoor CO_2 Prediction with Swarm-CO2 Dataset

```python

import matplotlib.pyplot as plt
import pandas as pd
import datetime as dt
import numpy as np
import seaborn as sns

import tensorflow_docs as tfdocs
import tensorflow_docs.plots
import tensorflow_docs.modeling
import tensorflow as ctf # This code has been tested with TensorFlow 1.6
import keras

from sklearn.metrics import mean_squared_error, mean_absolute_error
from math import sqrt
from tensorflow.keras import models
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense, Dropout, LSTM
#from tensorflow.keras.layers import CuDNNLSTM
from tensorflow.compat.v1.keras.layers import CuDNNLSTM
from sklearn.preprocessing import MinMaxScaler

from statsmodels.tsa.vector_ar.vecm import coint_johansen
from sklearn.model_selection import cross_val_score
from fbprophet import Prophet
from fbprophet.plot import add_changepoints_to_plot
from keras.callbacks import EarlyStopping




# import dataset as a csv file
file_url = "C:/Users/chikabrown/Documents/Netatmo-docs/preli-swarm.csv"
column_name = ["date-time",
               "temp",
               "hum",
               "noise",
               "pres",
                "co2"
               ]

def import_file(url, columns):
    df = pd.read_csv(url, names=columns, skiprows=1, parse_dates=['date-time'], date_parser=lambda col: pd.to_datetime(col, utc=True),
                        encoding='utf-8-sig', sep=',', infer_datetime_format=True)
    df['date-time'] = df['date-time'].astype(str).str[:-6]
    df['date-time'] = pd.to_datetime(df['date-time'], format = '%Y-%m-%d %H:%M:%S', errors='coerce')
    # Creates more time series features from datetime index.
    cats = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]
    df['day-of-week'] = df['date-time'].dt.day_name().astype('category'
   , cats)
    df = df.dropna()
    df = df.reset_index(drop=True)

    return df

data = import_file(file_url, column_name)
#data[-1351:]
data

# plot each series as a serate subplot
# load dataset
data_with_time_ndex = data.set_index('date-time')
values = data_with_time_ndex.values
# specify columns to plot
groups = [0, 1, 2, 3,4, 5]
i = 1
# plot each column
plt.figure(figsize=(15,15))
for group in groups:
    plt.subplot(len(groups), 1, i )
    plt.plot(values[:, group])
    plt.title(data_with_time_ndex.columns[group], loc='right')
    i += 1
plt.savefig("swarm-all-column.png")
plt.show()


# check for stationarity with coint johansen
coint_johansen(data_with_time_ndex.iloc[:, :4],-1,1).eig

# create new dataframe with the only co2 and newly added features
def create_features(df, label=None):
    X = df[['day-of-week']]
    if label:
        y = df[label]
        return X, y
    return X

X, y = create_features(data, label='co2')


features_and_target = pd.concat([X, y], axis=1)


# Plotting the Features to see trends
# CO2 conc has strong daily and seasonal properties.
# Day of week also seems to show differences in peaks

sns.pairplot(features_and_target,
             hue='day-of-week',
             x_vars=['day-of-week'],
             y_vars='co2',
             height=7,
             aspect=2,
             plot_kws={'alpha':1.0, 'linewidth':0}
```

```
                )
plt.suptitle('CO2 concentration levels by Day of Week')
plt.savefig("swarm-co2-season-co2.png")
plt.show()

# convert day of the week column to one hot encoded

# The "Day-of-week" column is really categorical, not numeric. So convert that to a one-hot:
data['day-of-week'] = pd.DataFrame({'day-of-week':['Monday','Tuesday','Wednesday','Thursday', 'Friday', 'Saturday', 'Sunday']})
data = pd.concat([data, pd.get_dummies(data['day-of-week'], prefix='')], axis=1)

# now drop the original 'Day-of-week' column
data.drop(['day-of-week'],axis=1, inplace=True)
print(data.columns)
new_data = data.copy()

new_data = new_data.set_index('date-time')

# split into 34% test and 66% training set
train_size = int(len(new_data)   0.66)
test_size = int(len(new_data)  0.34)

train_arr = new_data[0:train_size]
test_arr = new_data[train_size:]

# split into target and features for training and test set
y_train = train_arr['co2']
y_test = test_arr['co2']
X_train = train_arr[["temp","hum","noise","pres", '_Friday',
        '_Monday', '_Saturday', '_Sunday', '_Thursday', '_Tuesday',
        '_Wednesday']]
X_test = test_arr[["temp","hum","noise","pres", '_Friday',
        '_Monday', '_Saturday', '_Sunday', '_Thursday', '_Tuesday',
        '_Wednesday']]
X_test.shape

# ensure all data is float
X_train_values = X_train.values.astype('float32')
X_test_values = X_test.values.astype('float32')
y_train_values = y_train.values.astype('float32')
y_test_values = y_test.values.astype('float32')

# normalize all the features to be within a range of 0 and 1
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_trainX = scaler.fit_transform(X_train_values)
scaled_testX = scaler.fit_transform(X_test_values)
scaled_trainy = scaler.fit_transform(y_train_values.reshape(-1,1))
scaled_testy = scaler.fit_transform(y_test_values.reshape(-1,1))

# convert multivare times series to supervise learning using 24 time steps

def temporalize(X, y, lookback=1):
    output_X = []
    output_y = []
    for i in range(len(X)-lookback-1):
        t = []
        for j in range(1,lookback+1):
            # Gather past records upto the lookback period
            t.append(X[[(i+j+1)], :])
        output_X.append(t)
        output_y.append(y[i+lookback+1])
    return np.array(output_X), np.array(output_y)
lookback = 24
trainX, train_y = temporalize(scaled_trainX, scaled_trainy, lookback)
testX, test_y = temporalize(scaled_testX, scaled_testy, lookback)

# Reshape the input data into n_samples X timesteps X n_features
n_features = 11
reshapeXtrain = trainX.reshape(trainX.shape[0], lookback, n_features)
reshapeXtest = testX.reshape(testX.shape[0], lookback, n_features)
reshapeXtest.shape

# define model and print summary
model = Sequential()
model.add(LSTM(100, input_shape=(reshapeXtrain.shape[1], reshapeXtrain.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')

history = model.fit(reshapeXtrain, train_y, epochs=20, batch_size=35, validation_data=(reshapeXtest, test_y),
                        callbacks=[EarlyStopping(monitor='val_loss', patience=10)], verbose=1, shuffle=False)

model.summary()

# predict test set using lstm for lstm model for 12 time steps
yhat = model.predict(reshapeXtest, verbose=0)

# def custom rmse
#this unction is the same code as MSE with the addition of the sqrt() wrapping the result.

def rmse(test_y, yhat):
  return backend.sqrt(backend.mean(backend.square(yhat - test_y), axis=-1))

#For example, below is the code for the mean_squared_error loss function and metric in Keras.

def mean_squared_error(test_y, yhat):
    return K.mean(K.square(yhat - test_y), axis=-1)

#We can test this in our regression example as follows.
#Note that we simply list the function name directly rather
# than providing it as a string or alias for Keras to resolve.


```

```python
# define model and print the value of rsme with the custom rsme function
model = Sequential()
model.add(LSTM(100, input_shape=(reshapeXtrain.shape[1], reshapeXtrain.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam', metrics=[rmse])

history = model.fit(reshapeXtrain, train_y, epochs=20, batch_size=35, validation_data=(reshapeXtest, test_y),
                    callbacks=[EarlyStopping(monitor='val_loss', patience=10)], verbose=1, shuffle=False)

model.summary()

# plot the line plot of custom rsme value
plt.plot(history.history['rmse'])
plt.show()

reset_time = data.reset_index(drop=True)

# invert predictions and test set
yhat = scaler.inverse_transform(yhat)
test_y = scaler.inverse_transform(test_y)

test_y


yhat.shape

# plot predicted and actual values

plt.figure(figsize=(20, 8))
plt.plot(reset_time['date-time'][-1339:], test_y.reshape(-1,1), label='actual')
plt.plot(reset_time['date-time'][-1339:], yhat.reshape(-1,1), label='forecast')
plt.ylabel('co2')
plt.xlabel('date')
plt.legend()
plt.savefig("24-lstm-multi-actvspred.png")
plt.show()

# plot loss
plt.figure(figsize=(8,4))
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Test Loss')
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epochs')
plt.legend(loc='upper right')
plt.show();

# Univeriate time series Prediction with lstm
# create dataframe with only co2 and date time
co2_data = data.co2.values
```

```

# ensure they are float all float and reshape
co2_data = co2_data.astype('float32')
co2_data = np.reshape(co2_data, (-1, 1))
co2_data.shape

# normalize the co2 data
scaler = MinMaxScaler(feature_range=(0, 1))
co2_data = scaler.fit_transform(co2_data)
train_size = int(len(co2_data)   0.66)
test_size = len(co2_data) - train_size
train, test = co2_data[0:train_size,:], co2_data[train_size:len(
    co2_data),:]

# convert the newly formed co2 dataframe to supervised learning using
    lstm look back
def create_dataset(dataset, look_back=1):
    X, Y = [], []
    for i in range(len(dataset)-look_back-1):
        a = dataset[i:(i+look_back), 0]
        X.append(a)
        Y.append(dataset[i + look_back, 0])
    return np.array(X), np.array(Y)

look_back = 24
X_train, Y_train = create_dataset(train, look_back)
X_test, Y_test = create_dataset(test, look_back)

# reshape input to be [samples, time steps, features]
X_train = np.reshape(X_train, (X_train.shape[0], 1, X_train.shape[1]))
X_test = np.reshape(X_test, (X_test.shape[0], 1, X_test.shape[1]))
Y_test.shape

model = Sequential()
model.add(LSTM(100, input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')

history = model.fit(X_train, Y_train, epochs=20, batch_size=70,
    validation_data=(X_test, Y_test),
                    callbacks=[EarlyStopping(monitor='val_loss',
    patience=10)], verbose=1, shuffle=False)

model.summary()


# make predictions with the test set
test_predict = model.predict(X_test)


# define custom rsme for univariate series
def rmse(Y_test, test_predict):
    return backend.sqrt(backend.mean(backend.square(test_predict -
    Y_test), axis=-1))


#define model and print summary with the custom rsme
```

```
model = Sequential()
model.add(LSTM(100, input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam', metrics=[
    rmse])

history = model.fit(X_train, Y_train, epochs=20, batch_size=70,
    validation_data=(X_test, Y_test),
                    callbacks=[EarlyStopping(monitor='val_loss',
    patience=10)], verbose=1, shuffle=False)

model.summary()

# plot the line plot of the rmse value
plt.plot(history.history['rmse'])
plt.show()

# invert predictions and the test set into original form
test_predict = scaler.inverse_transform(test_predict)
Y_test = scaler.inverse_transform([Y_test])
print(test_predict.shape)
print(Y_test.shape)



#plot loss function
plt.figure(figsize=(8,4))
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Test Loss')
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epochs')
plt.legend(loc='upper right')
plt.show();

# plot subplot for univariate time series
plt.figure(figsize=(20,8))
plt.plot(reset_time['date-time'][-1339:], Y_test.reshape(-1,1),  marker
    ='.', label='actual')
plt.plot( reset_time['date-time'][-1339:], test_predict.reshape(-1,1),
    'r', label="prediction")
plt.tight_layout()
sns.despine(top=True)
plt.subplots_adjust(left=0.07)
plt.ylabel('CO2 Levels', size=15)
plt.xlabel('Time step', size=15)
plt.legend(fontsize=15)
plt.show();

# Univariate Prophet Model
new_data = new_data.reset_index()

#initialize the prophet model based on the daily 5 minutes forecast
model = Prophet(changepoint_prior_scale=0.05, changepoint_range=0.9,
    seasonality_mode='multiplicative')
model.add_country_holidays(country_name='UK')
```

```python
# fit 65% train set
model.fit(df[df['ds'] > pd.to_datetime('2016-10-22 06:36:58')])
# ask prophet to make forcast for the next 2 hrs
future = model.make_future_dataframe(periods=1, freq='H')
# predict 35% test set
forecast = model.predict(future)
fig = model.plot(forecast, figsize=(20, 8))
a = add_changepoints_to_plot(fig.gca(), model, forecast)
plt.savefig('swarm-12-prophet.png')

# show daily seasonality of the components plot
fig = model.plot_components(forecast, figsize=(20, 8))
plt.savefig('prophet-seasonality-swarm.png')

#calculte error
y_forecast = forecast[:len(df)]
#MAE
print('The MAE:  ', mean_absolute_error(y_true=df['y'][-1351:],
                    y_pred=y_forecast['yhat']))
#RMSE
print('The RMSE:  ', sqrt(mean_squared_error(y_true=df['y'][-1351:],
                    y_pred=y_forecast['yhat'])))

# Multivariate Prophet model
# add additional regressors
df['temp'] = data['temp']
df['pres'] = data['pres']
df['noise'] = data['noise']
df['hum'] = data['hum']
df['_Monday'] = new_data['_Monday']
df['_Tuesday'] = new_data['_Tuesday']
df['_Wednesday'] = new_data['_Wednesday']
df['_Thursday'] = new_data['_Thursday']
df['_Friday'] = new_data['_Friday']
df['_Saturday'] = new_data['_Saturday']
df['_Sunday'] = new_data['_Sunday']

model = Prophet(changepoint_prior_scale=0.05, changepoint_range=0.9,
    seasonality_mode='multiplicative')
# add holidays event
model.add_country_holidays(country_name='UK')
# add additional regressor
model.add_regressor('temp')
model.add_regressor('hum')
model.add_regressor('pres')
model.add_regressor('noise')
model.add_regressor('_Monday')
model.add_regressor('_Tuesday')
model.add_regressor('_Wednesday')
model.add_regressor('_Thursday')
model.add_regressor('_Friday')
model.add_regressor('_Saturday')
model.add_regressor('_Sunday')
# fit 65% train set
model.fit(df[df['ds'] > pd.to_datetime('2016-10-22 06:36:58')])
# ask prophet to make forcast for the next 2 hrs
future_with_reg_sing = model.make_future_dataframe(periods=2, freq='H')
```

```
# initialize the columns
future_with_reg_sing['temp'] = df['temp']
future_with_reg_sing['hum'] = df['hum']
future_with_reg_sing['pres'] = df['pres']
future_with_reg_sing['noise'] = df['noise']
future_with_reg_sing['_Monday'] = df['_Monday']
future_with_reg_sing['_Tuesday'] = df['_Tuesday']
future_with_reg_sing['_Wednesday'] = df['_Wednesday']
future_with_reg_sing['_Thursday'] = df['_Thursday']
future_with_reg_sing['_Friday'] = df['_Friday']
future_with_reg_sing['_Saturday'] = df['_Saturday']
future_with_reg_sing['_Sunday'] = df['_Sunday']
# make prediction
# predict with 35% test set
future_with_reg_sing = model.predict(future_with_reg_sing)
# plot the result
fig = model.plot(future_with_reg_sing, figsize=(20, 8))
# plot fexibility of the trend
a = add_changepoints_to_plot(fig.gca(), model, future_with_reg_sing)
plt.savefig('swarm-24-multi-prophet.png')


#calculte error
true_y = df['y']
y_true_len = len(future_with_reg_sing)

future_24hr = future_with_reg_sing[len(df):]
y_forecast_with_reg_sing = future_with_reg_sing[:len(
    future_with_reg_sing)]
#MAE
print('MAE:', mean_absolute_error(y_true=true_y[-y_true_len:],
                         y_pred=y_forecast_with_reg_sing['yhat']))
#RMSE
print('RSME:', sqrt(mean_squared_error(y_true=true_y[-y_true_len:],
                         y_pred=y_forecast_with_reg_sing['yhat'])))

# 24steps(2hrs) Prophet model
df['ds'] = pd.to_datetime(data['date-time'])
df['y'] = data['co2']
# add additional regressors
df['temp'] = data['temp']
df['pres'] = data['pres']
df['noise'] = data['noise']
df['hum'] = data['hum']
df['_Monday'] = new_data['_Monday']
df['_Tuesday'] = new_data['_Tuesday']
df['_Wednesday'] = new_data['_Wednesday']
df['_Thursday'] = new_data['_Thursday']
df['_Friday'] = new_data['_Friday']
df['_Saturday'] = new_data['_Saturday']
df['_Sunday'] = new_data['_Sunday']

model = Prophet(changepoint_prior_scale=0.05, changepoint_range=0.9,
    seasonality_mode='multiplicative')
# add holidays event
```

```python
model.add_country_holidays(country_name='UK')
# add additional regressor
model.add_regressor('temp')
model.add_regressor('hum')
model.add_regressor('pres')
model.add_regressor('noise')
model.add_regressor('_Monday')
model.add_regressor('_Tuesday')
model.add_regressor('_Wednesday')
model.add_regressor('_Thursday')
model.add_regressor('_Friday')
model.add_regressor('_Saturday')
model.add_regressor('_Sunday')
# fit 65% train set
model.fit(df[df['ds'] > pd.to_datetime('2016-10-22 06:36:58')])
# ask prophet to make forcast for the next 2 hrs
future_with_reg_sing = model.make_future_dataframe(periods=1, freq='H')
# initialize the columns
future_with_reg_sing['temp'] = df['temp']
future_with_reg_sing['hum'] = df['hum']
future_with_reg_sing['pres'] = df['pres']
future_with_reg_sing['noise'] = df['noise']
future_with_reg_sing['_Monday'] = df['_Monday']
future_with_reg_sing['_Tuesday'] = df['_Tuesday']
future_with_reg_sing['_Wednesday'] = df['_Wednesday']
future_with_reg_sing['_Thursday'] = df['_Thursday']
future_with_reg_sing['_Friday'] = df['_Friday']
future_with_reg_sing['_Saturday'] = df['_Saturday']
future_with_reg_sing['_Sunday'] = df['_Sunday']
# make prediction
# predict with 35% test set
future_with_reg_sing = model.predict(future_with_reg_sing)
# plot the result
fig = model.plot(future_with_reg_sing, figsize=(20, 8))
# plot fexibility of the trend
a = add_changepoints_to_plot(fig.gca(), model, future_with_reg_sing)
plt.savefig('swarm-12-multi-prophet.png')

# show daily seasonality of the components plot
fig = model.plot_components(future_with_reg_sing, figsize=(20, 8))
# save the plot
plt.savefig('prophet-dhi-hols.png')

#calculte error
true_y = df['y']
y_true_len = len(future_with_reg_sing)

future_24hr = future_with_reg_sing[len(df):]
y_forecast_with_reg_sing = future_with_reg_sing[:len(
    future_with_reg_sing)]
#MAE
print('MAE: ', mean_absolute_error(y_true=true_y[-y_true_len:],
                        y_pred=y_forecast_with_reg_sing['yhat']))
#RMSE
print('RSME: ', sqrt(mean_squared_error(y_true=true_y[-y_true_len:],
                                   y_pred=y_forecast_with_reg_sing['yhat']))
```

Listing 1: Python Codes for Indoor CO_2 Prediction with Swarm-CO2 Dataset

```

# DHI dataset test for stationarity using  Augmented Dickey-Fuller test
# implemented in statsmodels as smt.adfuller. The return type is
# wrap it in a namedtuple

y = data_with_time_ndex['co2']

from collections import namedtuple
import statsmodels.formula.api as smf
import statsmodels.tsa.api as smt
import statsmodels.api as sm

ADF = namedtuple("ADF", "adf pvalue usedlag nobs critical icbest")
ADF( smt.adfuller(y))._asdict()

# H_A (alternative hypothesis): y is stationary, doesn't need to be differenced
# critical values should be more than the test statistics

OrderedDict([('adf', -7.543949272495876),
             ('pvalue', 3.321445299790804e-11),
             ('usedlag', 26),
             ('nobs', 9182),
             ('critical',
              {'1%': -3.4310623860906317,
               '5%': -2.861854829186942,
               '10%': -2.5669375785149895}),
             ('icbest', 50979.739766571176)])
```

Listing 2: ADF Test for Stationarity with DHI dataset

```

# check for stationarity with coint johansen
coint_johansen(data_with_time_ndex.iloc[:, :4],-1,1).eig

array([8.05061243e-02, 7.55511545e-03, 1.26592243e-03, 5.84347791e-07])
```

Listing 3: Johansen Test for Stationarity using DHI Dataset

```
# import dataset as a csv file
file_url = "C:/Users/chikabrown/Documents/Netatmo-docs/preli-swarm.csv"
column_name = ["date-time",
               "temp",
               "hum",
               "noise",
               "pres",
                "co2"
               ]

def import_file(url, columns):
    df = pd.read_csv(url, names=columns, skiprows=1, parse_dates=['date-time'], date_parser=lambda col: pd.to_datetime(col, utc=True),
                            encoding='utf-8-sig', sep=',',
   infer_datetime_format=True)
    df['date-time'] = df['date-time'].astype(str).str[:-6]
    df['date-time'] = pd.to_datetime(df['date-time'], format = '%Y-%m-%
   d %H:%M:%S', errors='coerce')
    # Creates more time series features from datetime index.
    df['day-of-week'] = df['date-time'].dt.day_name()
    df = df.dropna()
    #df = df.drop(['time-stamp'], axis=1)
    df = df.reset_index(drop=True)

    return df

data = import_file(file_url, column_name)
data[-1351:]
```

Listing 4: Dataset Visualization for Swarm-Co2 Dataset code

```

file_url = "C:/Users/chikabrown/Documents/Netatmo-docs/dhi-room.csv"
column_name = ["time-stamp",
                "date-time",
               "temp",
               "hum",
               "co2",
               "noise",
               "pres",
               ]

def import_file(url, columns):
     df = pd.read_csv(url, names=columns, skiprows=1, parse_dates=True,
    encoding='utf-8-sig', sep=',')
     df['date-time'] = pd.to_datetime(df['date-time'], format = '%d/%m/%
    Y %H:%M', errors='coerce')
     # Creates more time series features from datetime index.
     df['day-of-week'] = df['date-time'].dt.day_name()
     df = df.dropna()
     df = df.drop(['time-stamp'], axis=1)
     df = df.reset_index(drop=True)
     # reorder the table for better readability

     return df


data = import_file(file_url, column_name)

# set date-time column as index
data_with_time_ndex = data.set_index("date-time")
data_with_time_ndex.head()

# plot each series as a serate subplot
# load dataset
values = data_with_time_ndex.values
# specify columns to plot
groups = [ 1, 2, 3,4, 5]
i = 1
# plot each column
```

```python
plt.figure(figsize=(15,15))
for group in groups:
    plt.subplot(len(groups), 1, i )
    plt.plot(values[:, group])
    plt.title(data_with_time_ndex.columns[group], loc='right')
    i += 1
plt.show()

# create new dataframe with the only co2 and newly added features
def create_features(df, label=None):
    X = df[['day-of-week']]
    if label:
        y = df[label]
        return X, y
    return X

X, y = create_features(data, label='co2')


features_and_target = pd.concat([X, y], axis=1)


# Plotting the Features to see trends
# CO2 conc has strong daily and seasonal properties.
# Day of week also seems to show differences in peaks

sns.pairplot(features_and_target,
             hue='day-of-week',
             x_vars=['day-of-week'],
             y_vars='co2',
             height=7,
             aspect=2,
             plot_kws={'alpha':1.0, 'linewidth':0}
            )
plt.suptitle('CO2 concentration levels by Day of Week')
plt.savefig("post-season-co2.png")
plt.show()
```

Listing 5: Dataset Visualization for DHI Dataset code

The codes in Listing 1 are python codes used for the Swarm-CO2. After the importation of the necessary modules for the computation, it started with dataset cleansing. After the dataset cleansing, full dataset plot was done in order to show the patterns of the recorded dataset (variables). Another plot of CO_2 concentration levels against day of the week was done in order to identify how its concentration could help with the study analysis.
Next was the dataset preprocessing which was spiting into test and training set in order to minimize bias in the training data. Additional dataset cleansing done was scaling of the data between 0-1 in order to form a uniform data scales for better result accuracy.
The final steps were the application of our machine learning models LSTM and Prophetwhich have already been discussed extensively in Section 5.4.

Q Python Codes for Indoor CO_2 Prediction with Post-Doc Dataset

```python

# create interactive plot to see
import plotly.graph_objs as go
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)
import plotly_express as px
# cufflinks is a wrapper on plotly
import cufflinks as cf
cf.go_offline(connected=True)

# import dataset as a csv file
file_url = "C:/Users/chikabrown/Documents/Netatmo-docs/post-doc.csv"
column_name = ["time-stamp",
               "date-time",
              "temp",
              "hum",
              "noise",
              "pres",
               "co2"
              ]

def import_file(url, columns):
    df = pd.read_csv(url, names=columns, skiprows=1, parse_dates=True, encoding='utf-8-sig', sep=',')
    df['date-time'] = pd.to_datetime(df['date-time'], format = '%d/%m/%Y %H:%M', errors='coerce')
     # Creates more time series features from datetime index.
    cats = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"]
    df['day-of-week'] = df['date-time'].dt.day_name().astype('category', cats)
    df = df.dropna()
    df = df.drop(['time-stamp'], axis=1)
    df = df.reset_index(drop=True)

    return df
    data = import_file(file_url, column_name)
print(data.shape)
data.head(3000)

# plot each series as a serate subplot
# load dataset
data_with_time_ndex = data.set_index('date-time')
values = data_with_time_ndex.values
# specify columns to plot
groups = [ 1, 2, 3,4, 5]
i = 1
# plot each column
plt.figure(figsize=(15,15))
for group in groups:
    plt.subplot(len(groups), 1, i )
    plt.plot(values[:, group])
    plt.title(data_with_time_ndex.columns[group], loc='right')
    i += 1
plt.savefig("swarm-all-column.png")
plt.show()

# check for stationarity with coint johansen
coint_johansen(data_with_time_ndex.iloc[:, :4],-1,1).eig

# create new dataframe with the only co2 and newly added features
def create_features(df, label=None):
    X = df[['day-of-week']]
    if label:
        y = df[label]
        return X, y
    return X

X, y = create_features(data, label='co2')


features_and_target = pd.concat([X, y], axis=1)


# Plotting the Features to see trends
# CO2 conc has strong daily and seasonal properties.
# Day of week also seems to show differences in peaks

sns.pairplot(features_and_target,
             hue='day-of-week',
             x_vars=['day-of-week'],
             y_vars='co2',
             height=7,
             aspect=2,
             plot_kws={'alpha':1.0, 'linewidth':0}
            )
plt.suptitle('CO2 concentration levels by Day of Week')
#plt.savefig("swarm-co2-season-co2.png")
plt.show()

# convert day of the week to one hot encoded

# The "Day-of-week" column is really categorical, not numeric. So convert that to a one-hot:
data['day-of-week'] = pd.DataFrame({'day-of-week':['Monday','Tuesday','Wednesday','Thursday', 'Friday', 'Saturday', 'Sunday']})
data = pd.concat([data, pd.get_dummies(data['day-of-week'], prefix='')], axis=1)

# now drop the original 'Day-of-week' column
data.drop(['day-of-week'],axis=1, inplace=True)
print(data.columns)

new_data = data.set_index('date-time')

# split into test and training set
train_size = int(len(new_data)  0.66)
test_size = int(len(new_data)  0.34)

```

```python
train_arr = new_data[0:train_size]
test_arr = new_data[train_size:]

y_train = train_arr['co2']
y_test = test_arr['co2']
X_train = train_arr[["temp","hum","noise","pres", '_Friday',
        '_Monday', '_Saturday', '_Sunday', '_Thursday', '_Tuesday',
        '_Wednesday']]
X_test = test_arr[["temp","hum","noise","pres", '_Friday',
        '_Monday', '_Saturday', '_Sunday', '_Thursday', '_Tuesday',
        '_Wednesday']]
X_test.shape

# ensure all data is float
X_train_values = X_train.values.astype('float32')
X_test_values = X_test.values.astype('float32')
y_train_values = y_train.values.astype('float32')
y_test_values = y_test.values.astype('float32')

# normalize all the features to be within a range of 0 and 1
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_trainX = scaler.fit_transform(X_train_values)
scaled_testX = scaler.fit_transform(X_test_values)
scaled_trainy = scaler.fit_transform(y_train_values.reshape(-1,1))
scaled_testy = scaler.fit_transform(y_test_values.reshape(-1,1))

# convert multivare times series to supervise learning using 24 time steps

def temporalize(X, y, lookback=1):
    output_X = []
    output_y = []
    for i in range(len(X)-lookback-1):
        t = []
        for j in range(1,lookback+1):
            # Gather past records upto the lookback period
            t.append(X[[(i+j+1)], :])
        output_X.append(t)
        output_y.append(y[i+lookback+1])
    return np.array(output_X), np.array(output_y)
lookback = 12
trainX, train_y = temporalize(scaled_trainX, scaled_trainy, lookback)
testX, test_y = temporalize(scaled_testX, scaled_testy, lookback)

# Reshape the input data into n_samples X timesteps X n_features
n_features = 11
reshapeXtrain = trainX.reshape(trainX.shape[0], lookback, n_features)
reshapeXtest = testX.reshape(testX.shape[0], lookback, n_features)
reshapeXtest.shape

# define model and print summary for mse
model = Sequential()
model.add(LSTM(100, input_shape=(reshapeXtrain.shape[1], reshapeXtrain.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')

history = model.fit(reshapeXtrain, train_y, epochs=20, batch_size=35, validation_data=(reshapeXtest, test_y),
                    callbacks=[EarlyStopping(monitor='val_loss', patience=10)], verbose=1, shuffle=False)

model.summary()

# predict test set using lstm for lstm model for 12 time steps
yhat = model.predict(reshapeXtest, verbose=0)

# def custom rmse
#this unction is the same code as MSE with the addition of the sqrt() wrapping the result.

def rmse(test_y, yhat):
  return backend.sqrt(backend.mean(backend.square(yhat - test_y), axis=-1))

#For example, below is the code for the mean_squared_error loss function and metric in Keras.

def mean_squared_error(test_y, yhat):
    return K.mean(K.square(yhat - test_y), axis=-1)

#We can test this in our regression example as follows.
#Note that we simply list the function name directly rather
# than providing it as a string or alias for Keras to resolve.


# define model and print the value of rsme with the custom rsme function
model = Sequential()
model.add(LSTM(100, input_shape=(reshapeXtrain.shape[1], reshapeXtrain.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam', metrics=[rmse])

history = model.fit(reshapeXtrain, train_y, epochs=20, batch_size=35, validation_data=(reshapeXtest, test_y),
                    callbacks=[EarlyStopping(monitor='val_loss', patience=10)], verbose=1, shuffle=False)

model.summary()

# plot the line plot of custom rsme value
plt.plot(history.history['rmse'])
plt.show()

# reset time index
reset_time = data.reset_index(drop=True)

# invert predictions and test set
yhat = scaler.inverse_transform(yhat)
```

```python
test_y = scaler.inverse_transform(test_y)


time_reset = reset_time['date-time'][-3096:]

# plot predicted and actual values

plt.figure(figsize=(20, 8))
plt.plot(reset_time['date-time'][-3108:], test_y.reshape(-1,1), label='actual')
plt.plot(reset_time['date-time'][-3108:], yhat.reshape(-1,1), label='forecast')
plt.ylabel('co2')
plt.xlabel('date')
plt.legend()
plt.savefig("12-lstm-post-multi-actvspred.png")
plt.show()

plt.figure(figsize=(8,4))
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Test Loss')
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epochs')
plt.legend(loc='upper right')
plt.show();

# Univeriate time series Prediction with lstm
# create dataframe with only co2 and date time
co2_data = data.co2.values

# ensure they are float all float and reshape
co2_data = co2_data.astype('float32')
co2_data = np.reshape(co2_data, (-1, 1))
co2_data.shape

# normalize the co2 data
scaler = MinMaxScaler(feature_range=(0, 1))
co2_data = scaler.fit_transform(co2_data)
train_size = int(len(co2_data)   0.66)
test_size = len(co2_data) - train_size
train, test = co2_data[0:train_size,:], co2_data[train_size:len(co2_data),:]

# convert the newly formed co2 dataframe to supervised learning using lstm look back
def create_dataset(dataset, look_back=1):
    X, Y = [], []
    for i in range(len(dataset)-look_back-1):
        a = dataset[i:(i+look_back), 0]
        X.append(a)
        Y.append(dataset[i + look_back, 0])
    return np.array(X), np.array(Y)

look_back = 12
X_train, Y_train = create_dataset(train, look_back)
X_test, Y_test = create_dataset(test, look_back)

```

```python
# reshape input to be [samples, time steps, features]
X_train = np.reshape(X_train, (X_train.shape[0], 1, X_train.shape[1]))
X_test = np.reshape(X_test, (X_test.shape[0], 1, X_test.shape[1]))
Y_test.shape

# define model and print summary based on mse
model = Sequential()
model.add(LSTM(100, input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')

history = model.fit(X_train, Y_train, epochs=20, batch_size=70,
    validation_data=(X_test, Y_test),
                    callbacks=[EarlyStopping(monitor='val_loss',
    patience=10)], verbose=1, shuffle=False)

model.summary()

# make predictions with the test set
test_predict = model.predict(X_test)

# define custom rsme for univariate series
def rmse(Y_test, test_predict):
    return backend.sqrt(backend.mean(backend.square(test_predict -
    Y_test), axis=-1))


# define model and print summary of rsme
model = Sequential()
model.add(LSTM(100, input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam', metrics=[
    rmse])

history = model.fit(X_train, Y_train, epochs=20, batch_size=70,
    validation_data=(X_test, Y_test),
                    callbacks=[EarlyStopping(monitor='val_loss',
    patience=10)], verbose=1, shuffle=False)

model.summary()

# plot line plot of rmse value
plt.plot(history.history['rmse'])
plt.show()

# invert predictions and the test set into original form
test_predict = scaler.inverse_transform(test_predict)
Y_test = scaler.inverse_transform([Y_test])

print(test_predict.shape)
print(Y_test.shape)


plt.figure(figsize=(8,4))
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Test Loss')
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epochs')
plt.legend(loc='upper right')
plt.show();

plt.figure(figsize=(20,8))
plt.plot(reset_time['date-time'][-3108:], Y_test.reshape(-1,1),  marker='.', label='actual')
plt.plot( reset_time['date-time'][-3108:], test_predict.reshape(-1,1), 'r', label="prediction")
plt.tight_layout()
sns.despine(top=True)
plt.subplots_adjust(left=0.07)
plt.ylabel('CO2 Levels', size=15)
plt.xlabel('Time step', size=15)
plt.legend(fontsize=15)
plt.show();
```

Listing 6: Python Codes for Indoor CO_2 Prediction with Post-Doc Dataset

The codes in Listing 6 are python codes used for the Post-Doc. Just like the previous section, this It started with dataset cleansing. After the dataset cleansing, full dataset plot was done in order to show the patterns of the recorded dataset (variables). Another plot of CO_2 concentration levels against day of the week was done in order to identify how its concentration could help with the study analysis.

Next was the dataset preprocessing which was spiting into test and training set in order to minimize bias in the training data. Additional dataset cleansing done was scaling of the data between 0-1 in order to form a uniform data scales for better result accuracy.

Unlike in the previous section, the Post-Doc dataset was analysed with LSTM only because it was identified to have out-performed the Prophet in the Swarm-CO2. The methods and the results have already been discussed extensively in Section 5.5.3.

R Python Codes for Machine learning Models used for the Room Occupancy Study with JA314 Dataset

```

JA314_url ="C:\\Users\\chikabrown\\Documents\\Netatmo-docs\\occupancy\\JA314_env.csv"

# Rename the column names
column_name = ["Time-stamp",
               "Date-Time",
               "Temp",
               "Humi",
               "CO2",
               "Noise",
               "Press",
               "HVAC",
               "Occupancy"]


# Import and read files

def import_file(url, columns):
    df = pd.read_csv(url, names=columns, skiprows=1, parse_dates=True, sep=',')
    df['Date-Time'] = pd.to_datetime(df['Date-Time'], format = '%m/%d/%Y %H:%M:%S')
    df['Day-of-week'] = df['Date-Time'].dt.day_name()
    df = df.drop(['Time-stamp','HVAC'], axis=1)
    df = df.dropna()
    df = df.reset_index(drop=True)
    return df

# import the two dataset and drop two columns
df = import_file(JA314_url, column_name)
df

# find the percentage missing data on each column
percent_missing = df.isna().sum()   100 / len(df)
missing_value_df = pd.DataFrame({'column_name': df.columns,
                                 'percent_missing': percent_missing})

# plot each series as a serate subplot
# load dataset
values = df.values
# specify columns to plot
groups = [ 1, 2, 3, 4, 5, 6]
i = 1
# plot each column
plt.figure(figsize=(15,15))
for group in groups:
    plt.subplot(len(groups), 1, i )
    plt.plot(values[:, group])
    plt.title(df.columns[group], loc='right')
    i += 1
```

```python
plt.savefig("all-314.png")
plt.show()
df.describe()

# convert day of the week to one hot encoded

# The "Day-of-week" column is really categorical, not numeric. So convert that to a one-hot:
df['Day-of-week'] = pd.DataFrame({'Day-of-week':['Monday','Tuesday','Wednesday','Thursday', 'Friday', 'Saturday', 'Sunday']})
df = pd.concat([df, pd.get_dummies(df['Day-of-week'], prefix='')], axis=1)

# now drop the original 'Day-of-week' column
df.drop(['Day-of-week'],axis=1, inplace=True)
print(df.columns)
new_data = df.copy()

# set date-time index
df = df.set_index('Date-Time')

# get only co2 and occupancy from imputed dataset
X_features = df[[ 'Temp', 'Humi', 'CO2', 'Noise', 'Press',
          '_Friday', '_Monday', '_Saturday', '_Sunday', '_Thursday',
         '_Tuesday', '_Wednesday']]
y_target = df['Occupancy']


# split training and test features
train_size = int(len(df)  0.66)
X_train, X_test = X_features[0:train_size], X_features[train_size:len(X_features)]
print('No of Observations: %d' % (len(df)))
print('Training Observations: %d' % (len(X_train)))
print('Testing Observations: %d' % (len(X_test)))

# split training and test target
y_train, y_test = y_target[0:train_size], y_target[train_size:len(y_target)]
print('No of Observations: %d' % (len(df)))
print('Training Observations: %d' % (len(y_train)))
print('Testing Observations: %d' % (len(y_test)))

# normalize the training and the test set
scaler = MinMaxScaler(feature_range=(0, 1))

normalized_X_train = pd.DataFrame(scaler.fit_transform(X_train), columns = X_train.columns)

normalized_X_test = pd.DataFrame(scaler.transform(X_test), columns = X_test.columns)

# plot the prediction interval
def plot_intervals(predictions, mid=False, start=None, stop=None, title=None):
    """
    Function for plotting prediction intervals as filled area chart.

    :param predictions: dataframe of predictions with lower, upper, and actual columns
    :param start: optional parameter for subsetting start of predictions
    :param stop: optional parameter for subsetting end of predictions
    :param title: optional string title

    :return fig: plotly figure
    """
    # Subset if required
    predictions = (
        predictions.loc[start:stop].copy()
        if start is not None or stop is not None
        else predictions.copy()
    )
    data = []
    # Lower trace will fill to the upper trace
    trace_low = go.Scatter(
        x=predictions.index,
        y=predictions["lower"],
        fill="tonexty",
        line=dict(color="darkblue"),
        fillcolor="rgba(173, 216, 230, 0.4)",
        showlegend=True,
        name="lower",
    )
    # Upper trace has no fill
    trace_high = go.Scatter(
        x=predictions.index,
        y=predictions["upper"],
        fill=None,
        line=dict(color="orange"),
        showlegend=True,
        name="upper",
    )
    # Must append high trace first so low trace fills to the high trace
    data.append(trace_high)
    data.append(trace_low)

    if predictions['mid'].all():
        trace_mid = go.Scatter(
        x=predictions.index,
        y=predictions["mid"],
        fill=None,
        line=dict(color="green"),
        showlegend=True,
        name="mid",
    )
    data.append(trace_mid)

    # Trace of actual values
    trace_actual = go.Scatter(
        x=predictions.index,
        y=predictions["actual"],
        fill=None,
        line=dict(color="darkviolet"),
        showlegend=True,
        name="actual",
    )
    data.append(trace_actual)

    layout = go.Layout(
    title = "Prediction intervals (JA314)",
    xaxis = {'title' : "date−time"},
    yaxis = {'title' : "Occupancy"},
    paper_bgcolor = "black",
    plot_bgcolor= "black"
    )

    fig = go.Figure(data=data, layout=layout)

    return fig

# build a repeatable Predictive class that can fit and test all the models in one call

class GradBoostPredInts(BaseEstimator):
    """
    These models will be using Scikit−Learn inteface for its prediction intervl

    :param lower_alpha: lower quantile for prediction, default=0.1
    :param upper_alpha: upper quantile for prediction, default=0.9
    :param   kwargs: additional keyword arguments for creating a GradientBoostingRegressor model
    """
    def __init__(self, lower_alpha=0.2, upper_alpha=0.8,   kwargs):
        self.lower_alpha = lower_alpha
        self.upper_alpha = upper_alpha

        # Three separate models
        self.lower_model = GradientBoostingRegressor(
            loss="quantile", alpha=self.lower_alpha,   kwargs
        )
        self.mid_model = GradientBoostingRegressor(loss="ls",   kwargs)
        self.upper_model = GradientBoostingRegressor(
            loss="quantile", alpha=self.upper_alpha,   kwargs
        )
        self.predictions = None
    def fit(self, X, y):
        """
        Fit all three models

        :param X: train features
        :param y: train targets

        TODO: parallelize this code across processors
        """
        self.lower_model.fit(X, y)
        self.mid_model.fit(X, y)
        self.upper_model.fit(X, y)
    def predict(self, X, y):
        """
        Predict with all 3 models

        :param X: test features
        :param y: test targets
        :return predictions: dataframe of predictions

        TODO: parallelize this code across processors
        """
        predictions = pd.DataFrame(data={"actual":y})
        predictions["lower"] = self.lower_model.predict(X)
        predictions["mid"] = self.mid_model.predict(X)
        predictions["upper"] = self.upper_model.predict(X)
        self.predictions = predictions

        return predictions
    def plot_intervals(self, mid=False, start=None, stop=None):
        """
        Plot the prediction intervals

        :param mid: boolean for whether to show the mid prediction
        :param start: optional parameter for subsetting start of predictions
        :param stop: optional parameter for subsetting end of predictions

        :return fig: plotly figure
        """

        if self.predictions is None:
            raise ValueError("This model has not yet made predictions.")
            return
        fig = plot_intervals(predictions, mid=mid, start=start, stop=stop)

        return fig
    def calculate_and_show_errors(self):
        """
        Calculate and display the errors associated with a set of prediction intervals

        :return fig: plotly boxplot of absolute error metrics
        """
        if self.predictions is None:
            raise ValueError("This model has not yet made predictions.")
            return

        calc_error(self.predictions)
        fig = show_metrics(self.predictions)
        return fig

model = GradBoostPredInts(
    lower_alpha=0.05, upper_alpha=0.95, n_estimators=1000, max_depth=3)

# Fit and make predictions
_ = model.fit(normalized_X_train, y_train)
predictions = model.predict(normalized_X_test, y_test)

#print performance metrics
```

```
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
print('Test Mean Absolute Error:', mean_absolute_error(y_test, predictions['mid'].values))
print('Test Root Mean Squared Error:',np.sqrt(mean_squared_error(y_test, predictions['mid'].values)))

fig = model.plot_intervals(predictions)
iplot(fig)

plt.figure(figsize=(20, 8))
plt.plot(predictions['actual'])
plt.plot(predictions['mid'])
plt.ylabel('Occupancy')
plt.xlabel('Date-time')
plt.legend()
plt.show()

# calculate the prediction error, the percentage of the time that the actual value falls in the range
# penalize the model for making too wide prediction intervals.

def calc_error(predictions):
    """
    Calculate the absolute error associated with prediction intervals

    :param predictions: dataframe of predictions
    :return: None, modifies the prediction dataframe

    """
    predictions['abs_error_lower'] = (predictions['lower'] - predictions["actual"]).abs()
    predictions['abs_error_upper'] = (predictions['upper'] - predictions["actual"]).abs()

    predictions['abs_error_interval'] = (predictions['abs_error_lower'] + predictions['abs_error_upper']) / 2
    predictions['abs_error_mid'] = (predictions['mid'] - predictions["actual"]).abs()
  # check if the actual values fall between the lower and the upper predicted occupancy values
    predictions['in_bounds'] = predictions["actual"].between(left=predictions['lower'], right=predictions['upper'])
# calculate the errors for lower, upper, mid and interval
calc_error(predictions)
metrics = predictions[['abs_error_lower', 'abs_error_upper', 'abs_error_interval', 'abs_error_mid', 'in_bounds']].copy()
metrics.describe()

#We see the mid prediction has a smaller absolute error (in terms of the median).
#It's interesting the absolute error for the mid bound is actually less than that for the upper prediction!
# We can write a short function to display the metrics.
def count_substring(string, sub_string):
    true_count=string.astype(str).str.contains(sub_string).sum()
    false_count = len(string) - true_count
    if true_count>0:
        print("There are {t} Trues and {f} Falses".format(t=true_count, f=false_count))
    else:
         print("There are no Trues")


count_substring(metrics["in_bounds"], "True")

# box plot of the metrics
# We can write a short function to display the metrics.
def show_metrics(metrics):
    """
    Make a boxplot of the metrics associated with prediction intervals

    :param metrics: dataframe of metrics produced from calculate error
    :return fig: plotly figure
    """
    percent_in_bounds = metrics['in_bounds'].mean()    100
    metrics_to_plot = metrics[[c for c in metrics if 'abs_error' in c]]

    # Rename the columns
    metrics_to_plot.columns = [column.split('_')[-1].title() for column in metrics_to_plot]

    # Create a boxplot of the metrics
    fig = px.box(
        metrics_to_plot.melt(var_name="metric", value_name='Absolute Error'),
        x="metric",
        y="Absolute Error",
        color='metric',
        #title=f"Error Metrics Boxplots    In Bounds = {percent_in_bounds:.2f}%",
        height=800,
        width=1000,
        points=False,
    )

    # Create new data with no legends
    d = []

    for trace in fig.data:
        # Remove legend for each trace
        trace['showlegend'] = False
        d.append(trace)
    # Simple plot layout
    layout = go.Layout(
        title=f"Error Metrics Boxplots    In Bounds = {percent_in_bounds:.2f}%",
        plot_bgcolor= "black",
        paper_bgcolor= "black",
        xaxis = {'title' : "metric"},
        yaxis = {'title' : "Absoluet Error"},
    )

    # Make the plot look a little better
    fig = go.Figure(data=d, layout=layout)
    #fig.data = d
    fig['layout']['font'] = dict(size=20)
    return fig

iplot(show_metrics(metrics))

def calculate_quantile_loss(quantile, actual, predicted):
    """
    Quantile loss for a given quantile and prediction
    """
    return np.maximum(quantile   (actual - predicted), (quantile - 1)   (actual - predicted))

def plot_quantile_loss(actual, prediction_list, quantile_list, plot_ls=False):
    """
    Shows the quantile loss associated with predictions at different quantiles.
    Figure shows the loss versus the error

    :param actual: array-like of actual values
    :param prediction_list: list of array-like predictions
    :param quantile_list: list of float quantiles corresponding to the predictions
    :param plot_ls: whether to plot the least squares loss

    :return fig: plotly figure
    """
    data = []

    # Iterate through each combination of prediction and quantile
    for predictions, quantile in zip(prediction_list, quantile_list):
        # Calculate the loss
        quantile_loss = calculate_quantile_loss(quantile, actual, predictions)

        errors = actual - predictions
        # Sort errors and loss by error
        idx = np.argsort(errors)
        errors = errors[idx]; quantile_loss = quantile_loss[idx]

        # Add data to plot
        data.append(go.Scatter(mode="lines", x=errors, y=quantile_loss, line=dict(width=4), name=f"{quantile} Quantile"))

    if plot_ls:
        loss = np.square(predictions - actual)
        errors = actual - predictions

        # Sort errors and loss by error
        idx = np.argsort(errors)
        errors = errors[idx]; loss = loss[idx]

        # Add data to plot
        data.append(go.Scatter(mode="lines", x=errors, y=loss, line=
        dict(width=4), name="Least Squares"))

    # Simple plot layout
    layout = go.Layout(
        title="Quantile Loss vs Error",
        yaxis=dict(title="Loss"),
        xaxis=dict(title="Error"),
        width=1000, height=600,
        paper_bgcolor = "black",
        plot_bgcolor= "black"
    )

    fig = go.Figure(data=data, layout=layout)
    fig['layout']['font'] = dict(size=18)
    return fig

# Create a plot showing the same predictions at different quantiles
fig = plot_quantile_loss(predictions['actual'], [predictions['lower'],
    predictions['mid'], predictions['upper']], [0.05,0.5,0.95], False)
iplot(fig)

fig = plot_quantile_loss(predictions['actual'], [predictions['lower'],
    predictions['mid'], predictions['upper']], [0.05,0.5,0.95], True)
iplot(fig)

# define seperate intervals based on the GB model used for SHAP testing
lower_alpha = 0.05
upper_alpha = 0.95

N_estimators = 1000
MAX_DEPTH = 3

# Each model has to be separate

lower_model = GradientBoostingRegressor(
    loss="quantile", alpha=lower_alpha, n_estimators=N_estimators,
    max_depth=MAX_DEPTH
)
# The mid model will use the default
mid_model = GradientBoostingRegressor(loss="ls", n_estimators=
    N_estimators, max_depth=MAX_DEPTH)

upper_model = GradientBoostingRegressor(
    loss="quantile", alpha=upper_alpha, n_estimators=N_estimators,
    max_depth=MAX_DEPTH
)

lower_model.fit(X_train, y_train)
mid_model.fit(X_train, y_train)
upper_model.fit(X_train, y_train)
# use SHAP explainer "Tree" on Gradient boosting regressor
explainerGB = shap.TreeExplainer(mid_model)
shap_values_GB_test = explainerGB.shap_values(X_test)
shap_values_GB_train = explainerGB.shap_values(X_train)

# put the SHAP results in a dataframe
df_shap_GB_test = pd.DataFrame(shap_values_GB_test, columns=X_test.
    columns.values)
```

```python
df_shap_GB_train = pd.DataFrame(shap_values_GB_train, columns=X_train.columns.values)
df_shap_GB_test.tail()

# if a feature has 10 or less unique values then treat it as categorical
categorical_features = np.argwhere(np.array([len(set(X_train.values[:,x]))
for x in range(X_train.values.shape[1])]) <= 10).flatten()
categorical_features

# LIME has one explainer for all models
explainer = lime.lime_tabular.LimeTabularExplainer(X_train.values,
feature_names=X_train.columns.values.tolist(),
class_names=['Occupancy'],
categorical_features=categorical_features,
verbose=True, mode='regression')

X_test.iloc[1,:]
explainerGB.expected_value

shap_values_GB_test[j]

#The plot below is called a force plot. It shows features contributing
# to push the prediction from the base value.
# The base value is the average model output over the training dataset we passed.
# Features pushing the prediction higher are shown in red
# Features pushing it lower appear in blue

# j will be the record we explain
j = 15
# initialize js for SHAP
shap.initjs()
# plot the shap values
shap.force_plot(explainerGB.expected_value, shap_values_GB_test[j], X_test.iloc[[j]])

# initialize js for SHAP
shap.force_plot(explainerGB.expected_value, shap_values_GB_test, X_test, show=True)

# we know that Noise had a positive influence on this predicted occupancy levels
# because its value was above 60.00
However, LIMEs feature importance differs from SHAPs.
# Since SHAP has a more solid theoretical foundation
# in LIME humi, co2 did not indicate positive influence to the occupancy levels
lime\_gb = explainer.explain\_instance(X\_test.values[j], mid_model.predict, num\_features=13)
lime_gb.show_in_notebook(show_table=True, )
#lime_gb.save_to_file("limeJA314-no-missing.html")

lime_gb.as_list()

#Variable importance graphs are useful tools for understanding the
```

```
model in a global sense.
# SHAP provides a theoretically sound method for evaluating variable importance. This is important,
# given the debate over which of the traditional methods of calculating variable
# importance is correct and that those methods do not always agree.

shap.summary_plot(shap_values_GB_test, X_test, plot_type="bar")

# Similar to a variable importance plot, SHAP also offers a summary plot
# showing the SHAP values for every instance from the training dataset.
# This can lead to a better understanding of overall
# patterns and allow discovery of pockets of prediction outliers

shap.summary_plot(shap_values_GB_train, X_train, show=False)
plt.savefig("shap-impact.png")

# Variable influence or dependency plots have long been a favorite of statisticians
# for model interpretability. SHAP provides these as well, and I find them quite useful.

shp_plott = shap.dependence_plot("CO2", shap_values_GB_train, X_train)
```

Listing 7: Python Codes for Machine Learning Models Used for the Room Occupancy Study with JA314 Dataset

Require: $\mathfrak{L} = \{(x_i, y_i)$, i=1,2, ..., n$\}$, the data set

Require: $\mathfrak{B}$, the base learning algorithm

Require: φ_0, the intended number of models

Require: PK, percentage of the models from the pool that will be used for prediction

```
 for i := 1 to $\varphi_0$ do
   $\mathfrak{L}_i$ = Bootstrap($\mathfrak{L}$)
   $\hat{f}_i = \mathfrak{B}(\mathfrak{L}_i)$
 end for
 for i := 1 to $\varphi_0$ do
   for j := 1 to $\varphi_0$ do
     $C_{ij} := \frac{1}{n}\sum_{p=1}^{n}[(\hat{f}_i(x_p) - f(x_p))(\hat{f}_j(x_p) - f(x_p))]$
   end for
 end for
 s := empty vector
 for i := 1 to $\varphi$ do
   minimum := $+\infty$
   for j $\in$ {1 to $\varphi_0$}$\backslash\{s_i, \ldots s_n\}$ do
     value := $i^{-2}(\sum_{p=1}^{i-1}\sum_{q=1}^{i-1} C_{sp,sq} + 2\sum_{p=1}^{i-1} C_{sp,j} + C_{j,j})$
     if(value < minimum) then
       $x_i$ := j
       minimum := value
     end if
   end for
 end for
 $\mathfrak{F} = \{\hat{f}|i = s_1, s_2, \ldots s_k\}$
 return $\mathfrak{F}$
```

Listing 8: Pseudo-code of bagging with ordererd pruning

The codes in Listing 7 are python codes used for the JA314 dataset for room occupancy estimation and interpretation. This section began with dataset cleansing and elimination of missing data. After the dataset cleansing, full dataset plot of all the variables was done in order to show the patterns of the recorded dataset.

Next was the dataset preprocessing which was spiting into test (34%) and training set (66%) in order to minimize bias in the training data. Additional dataset cleansing done was scaling of the data between 0-1 in order to form a uniform data scales for better result accuracy.

Before the application of the GB, three (low, mid and high) levels of prediction intervals were obtained and they have been discussed in Section 6.4.3.1. Further was the application of the ML interpretability methods and more plots showing how the interpretability works. All these were discussed in full in Section 6.4.3.2, Section 6.4.3.2, Section 6.4.3.3 and Section 6.4.3.3.

S Python Codes for Machine learning Models used for the Room Occupancy Study with Swarm Dataset

```python

# Read the data
# documents url

four_week ="C:\\Users\\chikabrown\\Documents\\Netatmo-docs\\occupancy-live\\four-week-occ.csv"

# table column names
column_name = ["Time-stamp",
               "Date-Time",
               "Temp",
               "Humi",
               "CO2",
               "Noise",
               "Press",
               "Occupancy",
               "Window",
               "Door",
               "Day-of-week"]
# read file, create dataframe and add additional columns

def import_file(url, columns):
    df = pd.read_csv(url, names=columns, skiprows=1, parse_dates=True, encoding='utf-8-sig', sep=',')
    df['Date-Time'] = pd.to_datetime(df['Date-Time'], format = '%d/%m/%Y %H:%M', errors='coerce')
    df['Day-of-week'] = df['Date-Time'].dt.day_name()
    df['Window'] = 0
    df['Door'] = 1
    df = df.dropna()
    df = df.drop(['Time-stamp','Window','Door'], axis=1)
    df = df.reset_index(drop=True)
    return df



df = import_file(four_week, column_name)

print(df.shape)
print(df.head(5))

# plot each series as a serate subplot
# load dataset
values = df.values
# specify columns to plot
groups = [ 1, 2, 3, 4, 5, 6]
i = 1
# plot each column
plt.figure(figsize=(15,15))
for group in groups:
    plt.subplot(len(groups), 1, i )
    plt.plot(values[:, group])
    plt.title(df.columns[group], loc='right')
    i += 1
plt.savefig("swarm-all.png")
plt.show()
df.describe()

# convert day of the week to one hot encoded

# The "Day-of-week" column is really categorical, not numeric. So convert that to a one-hot:
df['Day-of-week'] = pd.DataFrame({'Day-of-week':['Monday','Tuesday','Wednesday','Thursday', 'Friday', 'Saturday', 'Sunday']})
df = pd.concat([df, pd.get_dummies(df['Day-of-week'], prefix='')], axis=1)

# now drop the original 'Day-of-week' column
df.drop(['Day-of-week'],axis=1, inplace=True)
print(df.columns)
print(df.shape)
new_data = df.copy()

# set date-time index
df = df.set_index('Date-Time')

# get only co2 and occupancy from imputed dataset
X_features = df[[ 'Temp', 'Humi', 'CO2', 'Noise', 'Press',
        '_Friday', '_Monday', '_Saturday', '_Sunday', '_Thursday',
       '_Tuesday', '_Wednesday']]
y_target = df['Occupancy']

# split training and test features
train_size = int(len(df)  0.66)
X_train, X_test = X_features[0:train_size], X_features[train_size:len(X_features)]
print('No of Observations: %d' % (len(df)))
print('Training Observations: %d' % (len(X_train)))
print('Testing Observations: %d' % (len(X_test)))

# split training and test target
y_train, y_test = y_target[0:train_size], y_target[train_size:len(y_target)]
print('No of Observations: %d' % (len(df)))
print('Training Observations: %d' % (len(y_train)))
print('Testing Observations: %d' % (len(y_test)))

# normalize the training and the test set
scaler = MinMaxScaler(feature_range=(0, 1))

normalized_X_train = pd.DataFrame(scaler.fit_transform(X_train), columns = X_train.columns)

normalized_X_test = pd.DataFrame(scaler.transform(X_test), columns = X_test.columns)

# plot the prediction interval
def plot_intervals(predictions, mid=False, start=None, stop=None, title=None):
    """
    Function for plotting prediction intervals as filled area chart.

    :param predictions: dataframe of predictions with lower, upper, and actual columns
    :param start: optional parameter for subsetting start of predictions
    :param stop: optional parameter for subsetting end of predictions
    :param title: optional string title

    :return fig: plotly figure
    """
    # Subset if required
    predictions = (
        predictions.loc[start:stop].copy()
        if start is not None or stop is not None
        else predictions.copy()
    )
    data = []
    # Lower trace will fill to the upper trace
    trace_low = go.Scatter(
        x=predictions.index,
        y=predictions["lower"],
        fill="tonexty",
        line=dict(color="darkblue"),
        fillcolor="rgba(173, 216, 230, 0.4)",
        showlegend=True,
        name="lower",
    )
    # Upper trace has no fill
    trace_high = go.Scatter(
        x=predictions.index,
        y=predictions["upper"],
        fill=None,
        line=dict(color="orange"),
        showlegend=True,
        name="upper",
    )
    # Must append high trace first so low trace fills to the high trace
    data.append(trace_high)
    data.append(trace_low)

    if predictions['mid'].all():
        trace_mid = go.Scatter(
        x=predictions.index,
        y=predictions["mid"],
        fill=None,
        line=dict(color="green"),
        showlegend=True,
        name="mid",
    )
    data.append(trace_mid)

    # Trace of actual values
    trace_actual = go.Scatter(
        x=predictions.index,
        y=predictions["actual"],
        fill=None,
        line=dict(color="darkviolet"),
        showlegend=True,
        name="actual",
    )
    data.append(trace_actual)

    layout = go.Layout(
    title = "Prediction intervals (Swarm)",
    xaxis = {'title' : "date-time"},
    yaxis = {'title' : "Occupancy"},
    paper_bgcolor = "black",
    plot_bgcolor= "black"
    )

    fig = go.Figure(data=data, layout=layout)

    return fig


# build a repeatable Predictive class that can fit and test all the models in one call

class GradBoostPredInts(BaseEstimator):
    """
    These models will be using Scikit-Learn inteface for its prediction intervl

    :param lower_alpha: lower quantile for prediction, default=0.1
    :param upper_alpha: upper quantile for prediction, default=0.9
    :param   kwargs: additional keyword arguments for creating a GradientBoostingRegressor model
    """
    def __init__(self, lower_alpha=0.2, upper_alpha=0.8,   kwargs):
        self.lower_alpha = lower_alpha
        self.upper_alpha = upper_alpha

        # Three separate models
        self.lower_model = GradientBoostingRegressor(
            loss="quantile", alpha=self.lower_alpha,   kwargs
        )
        self.mid_model = GradientBoostingRegressor(loss="ls",   kwargs)
        self.upper_model = GradientBoostingRegressor(
            loss="quantile", alpha=self.upper_alpha,   kwargs
        )
        self.predictions = None
    def fit(self, X, y):
        """
        Fit all three models

        :param X: train features
        :param y: train targets

        TODO: parallelize this code across processors
        """
        self.lower_model.fit(X, y)
        self.mid_model.fit(X, y)
        self.upper_model.fit(X, y)
    def predict(self, X, y):
        """
        Predict with all 3 models

        :param X: test features
        :param y: test targets
        :return predictions: dataframe of predictions

        TODO: parallelize this code across processors
        """
        predictions = pd.DataFrame(data={"actual":y})
        predictions["lower"] = self.lower_model.predict(X)
        predictions["mid"] = self.mid_model.predict(X)
        predictions["upper"] = self.upper_model.predict(X)
        self.predictions = predictions

        return predictions
    def plot_intervals(self, mid=False, start=None, stop=None):
        """
        Plot the prediction intervals

        :param mid: boolean for whether to show the mid prediction
        :param start: optional parameter for subsetting start of predictions
        :param stop: optional parameter for subsetting end of predictions

        :return fig: plotly figure
        """

        if self.predictions is None:
            raise ValueError("This model has not yet made predictions.")
            return
        fig = plot_intervals(predictions, mid=mid, start=start, stop=stop)

        return fig
    def calculate_and_show_errors(self):
        """
        Calculate and display the errors associated with a set of prediction intervals

        :return fig: plotly boxplot of absolute error metrics
        """
        if self.predictions is None:
            raise ValueError("This model has not yet made predictions.")
            return

        calc_error(self.predictions)
        fig = show_metrics(self.predictions)
        return fig

model = GradBoostPredInts(
    lower_alpha=0.05, upper_alpha=0.95, n_estimators=1000, max_depth=3)

# Fit and make predictions
_ = model.fit(normalized_X_train, y_train)
```

```
predictions = model.predict(normalized_X_test, y_test)

#print performance metrics

from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
print('Test Mean Absolute Error:', mean_absolute_error(y_test, predictions['mid'].values))
print('Test Root Mean Squared Error:',np.sqrt(mean_squared_error(y_test, predictions['mid'].values)))

fig = model.plot_intervals(predictions)
iplot(fig)

plt.figure(figsize=(20, 8))
plt.plot(predictions['actual'])
plt.plot(predictions['mid'])
plt.legend()
plt.show()


# calculate the prediction error, the percentage of the time that the actual value falls in the range
# penalize the model for making too wide prediction intervals.

def calc_error(predictions):
    """
    Calculate the absolute error associated with prediction intervals

    :param predictions: dataframe of predictions
    :return: None, modifies the prediction dataframe

    """
    predictions['abs_error_lower'] = (predictions['lower'] - predictions["actual"]).abs()
    predictions['abs_error_upper'] = (predictions['upper'] - predictions["actual"]).abs()

    predictions['abs_error_interval'] = (predictions['abs_error_lower'] + predictions['abs_error_upper']) / 2
    predictions['abs_error_mid'] = (predictions['mid'] - predictions["actual"]).abs()
  # check if the actual values fall between the lower and the upper predicted occupancy values
    predictions['in_bounds'] = predictions["actual"].between(left=predictions['lower'], right=predictions['upper'])
# calculate the errors for lower, upper, mid and interval
calc_error(predictions)
metrics = predictions[['abs_error_lower', 'abs_error_upper', 'abs_error_interval', 'abs_error_mid', 'in_bounds']].copy()
metrics.describe()

# box plot of the metrics
# We can write a short function to display the metrics.
def show_metrics(metrics):
    """
    Make a boxplot of the metrics associated with prediction intervals

    :param metrics: dataframe of metrics produced from calculate error
    :return fig: plotly figure
    """
    percent_in_bounds = metrics['in_bounds'].mean()    100
    metrics_to_plot = metrics[[c for c in metrics if 'abs_error' in c]]

    # Rename the columns
    metrics_to_plot.columns = [column.split('_')[-1].title() for column
    in metrics_to_plot]

    # Create a boxplot of the metrics
    fig = px.box(
        metrics_to_plot.melt(var_name="metric", value_name='Absolute
    Error'),
        x="metric",
        y="Absolute Error",
        color='metric',
        #title=f"Error Metrics Boxplots    In Bounds = {
    percent_in_bounds:.2f}%",
        height=800,
        width=1000,
        points=False,
    )

    # Create new data with no legends
    d = []

    for trace in fig.data:
        # Remove legend for each trace
        trace['showlegend'] = False
        d.append(trace)
    # Simple plot layout
    layout = go.Layout(
        title=f"Error Metrics Boxplots In Bounds (Swarm) = {
    percent_in_bounds:.2f}% ",
        plot_bgcolor= "black",
        paper_bgcolor= "black",
        xaxis = {'title' : "metric"},
        yaxis = {'title' : "Absoluet Error"},
    )

    # Make the plot look a little better
    fig = go.Figure(data=d, layout=layout)
    #fig.data = d
    fig['layout']['font'] = dict(size=16)
    return fig

iplot(show_metrics(metrics))

def calculate_quantile_loss(quantile, actual, predicted):
    """
    Quantile loss for a given quantile and prediction
    """
    return np.maximum(quantile   (actual - predicted), (quantile - 1)
    (actual - predicted))
```

```python

def plot_quantile_loss(actual, prediction_list, quantile_list, plot_ls=False):
    """
    Shows the quantile loss associated with predictions at different quantiles.
    Figure shows the loss versus the error

    :param actual: array-like of actual values
    :param prediction_list: list of array-like predictions
    :param quantile_list: list of float quantiles corresponding to the predictions
    :param plot_ls: whether to plot the least squares loss

    :return fig: plotly figure
    """
    data = []

    # Iterate through each combination of prediction and quantile
    for predictions, quantile in zip(prediction_list, quantile_list):
        # Calculate the loss
        quantile_loss = calculate_quantile_loss(quantile, actual, predictions)

        errors = actual - predictions
        # Sort errors and loss by error
        idx = np.argsort(errors)
        errors = errors[idx]; quantile_loss = quantile_loss[idx]

        # Add data to plot
        data.append(go.Scatter(mode="lines", x=errors, y=quantile_loss, line=dict(width=4), name=f"{quantile} Quantile"))

    if plot_ls:
        loss = np.square(predictions - actual)
        errors = actual - predictions

        # Sort errors and loss by error
        idx = np.argsort(errors)
        errors = errors[idx]; loss = loss[idx]

        # Add data to plot
        data.append(go.Scatter(mode="lines", x=errors, y=loss, line=dict(width=4), name="Least Squares"))

    # Simple plot layout
    layout = go.Layout(
        title="Quantile Loss vs Error (Swarm)",
        yaxis=dict(title="Loss"),
        xaxis=dict(title="Error"),
        width=1000, height=600,
        paper_bgcolor = "black",
        plot_bgcolor= "black"
    )

    fig = go.Figure(data=data, layout=layout)
    fig['layout']['font'] = dict(size=18)
    return fig

    # Create a plot showing the same predictions at different quantiles
fig = plot_quantile_loss(predictions['actual'], [predictions['lower'],
    predictions['mid'], predictions['upper']], [0.05,0.5,0.95], False)
iplot(fig)

fig = plot_quantile_loss(predictions['actual'], [predictions['lower'],
    predictions['mid'], predictions['upper']], [0.05,0.5,0.95], True)
iplot(fig)

# define seperate intervals based on the GB model used for SHAP testing
lower_alpha = 0.05
upper_alpha = 0.95

N_estimators = 1000
MAX_DEPTH = 3

# Each model has to be separate

lower_model = GradientBoostingRegressor(
     loss="quantile", alpha=lower_alpha, n_estimators=N_estimators,
    max_depth=MAX_DEPTH
)
# The mid model will use the default
mid_model = GradientBoostingRegressor(loss="ls", n_estimators=
    N_estimators, max_depth=MAX_DEPTH)

upper_model = GradientBoostingRegressor(
     loss="quantile", alpha=upper_alpha, n_estimators=N_estimators,
    max_depth=MAX_DEPTH
)

lower_model.fit(X_train, y_train)
mid_model.fit(X_train, y_train)
upper_model.fit(X_train, y_train)
# use SHAP explainer "Tree" on Gradient boosting regressor
explainerGB = shap.TreeExplainer(mid_model)
shap_values_GB_test = explainerGB.shap_values(X_test)
shap_values_GB_train = explainerGB.shap_values(X_train)
X_test.shape

# put the SHAP results in a dataframe
df_shap_GB_test = pd.DataFrame(shap_values_GB_test, columns=X_test.
    columns.values)
df_shap_GB_train = pd.DataFrame(shap_values_GB_train, columns=X_train.
    columns.values)
#df_shap_GB_test.tail()

# if a feature has 10 or less unique values then treat it as
    categorical
categorical_features = np.argwhere(np.array([len(set(X_train.values[:,x
    ]))
for x in range(X_train.values.shape[1])]) <= 10).flatten()
categorical_features

# LIME has one explainer for all models
explainer = lime.lime_tabular.LimeTabularExplainer(X_train.values,
feature_names=X_train.columns.values.tolist(),
```

```python
class_names=['Occupancy'],
categorical_features=categorical_features,
verbose=True, mode='regression')

explainerGB.expected_value

df_shap_GB_test.iloc[0,:]

X_test.iloc[23,:]

# j will be the record we explain
j = 47
# initialize js for SHAP
shap.initjs()
# plot the shap values
shap.force_plot(explainerGB.expected_value, shap_values_GB_test[j], X_test.iloc[[j]])

# initialize js for SHAP
shap.force_plot(explainerGB.expected_value, shap_values_GB_test, X_test, show=True)

# in LIMe humi, co2 did not indicate positive influence to the occupancy levels
lime_gb = explainer.explain_instance(X_test.values[j], mid_model.predict, num_features=13)
lime_gb.show_in_notebook(show_table=True, )

# importance is correct and that those methods do not always agree.

shap.summary_plot(shap_values_GB_train, X_train, plot_type="bar")

# Similar to a variable importance plot, SHAP also offers a summary plot
# showing the SHAP values for every instance from the training dataset.
# This can lead to a better understanding of overall
# patterns and allow discovery of pockets of prediction outliers

shap.summary_plot(shap_values_GB_train, X_train, show=False)
plt.savefig("shap-impact.png")

# Variable influence or dependency plots have long been a favorite of statisticians
# for model interpretability. SHAP provides these as well, and I find them quite useful.

shp_plott = shap.dependence_plot("CO2", shap_values_GB_train, X_train)
```

Listing 9: Python Codes for Machine learning Models used for the Room Occupancy Study with Swarm Dataset

The codes in Listing 9 are python codes used for the Swarm dataset for room occupancy estimation and interpretation. Similar to the previous section, it began with dataset cleansing and elimination of missing data and they are discussed in Section 6.5.2. After the dataset cleansing, full dataset plot of all the variables was done in order to show the patterns of the recorded dataset.

Next was the dataset preprocessing which was spiting into test (34%) and training set (66%) in order to minimize bias in the training data. Additional dataset cleansing done was scaling of the data between 0-1 in order to form a uniform data scales for better result accuracy. Error metrics were also plotted and they were discussed in Section 6.5.2.1.

Before the application of the GB, three (low, mid and high) levels of prediction intervals were obtained and they have been discussed in Section 6.4.3.1. Further was the application of the ML interpretability methods and more plots showing how the interpretability works. All these were discussed in full in Section 6.5.2.2, Section 6.4.3.2 and Section 6.5.2.2.

Bibliography

Netatmo helpcenter. [online], 2020.

Muhammad Aurangzeb Ahmad, Carly Eckert, and Ankur Teredesai. Interpretable machine learning in healthcare. In *Proceedings of the 2018 ACM international conference on bioinformatics, computational biology, and health informatics*, pages 559–560, 2018.

Luai Al Shalabi, Zyad Shaaban, and Basel Kasasbeh. Data mining: A preprocessing engine. *Journal of Computer Science*, 2(9):735–739, 2006.

Azimil Gani Alam, Haolia Rahman, Jung-Kyung Kim, and Hwataik Han. Uncertainties in neural network model based on carbon dioxide concentration for occupancy estimation. *Journal of Mechanical Science and Technology*, 31(5):2573–2580, 2017.

Kamal M Ali and Michael J Pazzani. Error reduction through learning multiple descriptions. *Machine learning*, pages 173–202, 1996.

Hussain Alkharusi. Categorical variables in regression analysis: A comparison of dummy and effect coding. *International Journal of Education*, 4(2):202, 2012.

Joseph G Allen, Piers MacNaughton, Usha Satish, Suresh Santanam, Jose Vallarino, and John D Spengler. Associations of cognitive function scores with carbon dioxide, ventilation, and volatile organic compound exposures in office workers: a controlled exposure study of green and conventional office environments. *Environmental health perspectives*, page 805, 2016.

Manar Amayri, Abhay Arora, Stephane Ploix, Sanghamitra Bandhyopadyay, Quoc-Dung Ngo, and Venkata Ramana Badarla. Estimating occupancy in heterogeneous sensor environment. *Energy and Buildings*, 129:46–58, 2016.

Refrigerating American Society of Heating and Air conditioning Engineers. Standard 62.1-2016 – ventilation for acceptable indoor air quality (ansi approved). *Environment*, 2016. URL http://www.techstreet.com/ashrae/standards/ashrae-62-1-2016?product_id=1912838.

Aynur O Aptula, Nina G Jeliazkova, Terry W Schultz, and Mark TD Cronin. The better predictive model: high q2 for the training set or low root mean square error of prediction for the test set? *QSAR & Combinatorial Science*, 24(3):385–396, 2005.

Irvan B Arief-Ang, Flora D Salim, and Margaret Hamilton. Da-hoc: semi-supervised domain adaptation for room occupancy prediction using co2 sensor data. In *Proceedings of the 4th*

ACM International Conference on Systems for Energy-Efficient Built Environments, pages 1–10, 2017.

Irvan B. Arief-Ang, Margaret Hamilton, and Flora D. Salim. A scalable room occupancy prediction with transferable time series decomposition of co2 sensor data. *ACM Trans. Sen. Netw.*, 14(3–4), November 2018a. ISSN 1550-4859. doi: 10.1145/3217214. URL https://doi.org/10.1145/3217214.

Irvan B Arief-Ang, Margaret Hamilton, and Flora D Salim. A scalable room occupancy prediction with transferable time series decomposition of co 2 sensor data. *ACM Transactions on Sensor Networks (TOSN)*, 14(3-4):21, 2018b.

J Asha, S Rishidas, S SanthoshKumar, and P Reena. Analysis of temperature prediction using random forest and facebook prophet algorithms. In *International Conference on Innovative Data Communication Technologies and Application*, pages 432–439. Springer, 2019.

Paolo Atzeni and Valeria De Antonellis. *Relational database theory*. Benjamin/Cummings Redwood City, CA, 1993.

Jorge Luis Bacca Acosta, Silvia Margarita Baldiris Navarro, Ramon Fabregat Gesa, Sabine Graf, et al. Augmented reality trends in education: a systematic review of research and applications. *Journal of Educational Technology and Society, 2014, vol. 17, núm. 4, p. 133-149*, 2014.

Paramvir Bahl and Venkata N Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *Proceedings IEEE INFOCOM 2000. Conference on computer communications. Nineteenth annual joint conference of the IEEE computer and communications societies (Cat. No. 00CH37064)*, volume 2, pages 775–784. Ieee, 2000.

Simon Bell, Dan Cornford, and Lucy Bastin. The state of automated amateur weather observations. *Weather*, 68(2):36–41, 2013.

E Bentley. otranscribe, 2019. URL https://otranscribe.com/. Last accessed on 10 Nov. 2019.

Kenneth J Berry, Paul W Mielke Jr, and Hariharan K Iyer. Factorial designs and dummy coding. *Perceptual and motor skills*, 87(3):919–927, 1998.

Peter J Brockwell, Richard A Davis, and Matthew V Calder. *Introduction to time series and forecasting*, volume 2. Springer, 2002.

Jonathan Brooks, Saket Kumar, Siddharth Goyal, Rahul Subramany, and Prabir Barooah. Energy-efficient control of under-actuated hvac zones in commercial buildings. *Energy and Buildings*, pages 160–168, 2015.

Philip Burnard. A method of analysing interview transcripts in qualitative research. *Nurse education today*, 11(6):461–466, 1991.

Luis M Candanedo and Véronique Feldheim. Accurate occupancy detection of an office room from light, temperature, humidity and co 2 measurements using statistical learning models. *Energy and Buildings*, 112:28–39, 2016.

Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 2017.

Patricio Cerda, Gaël Varoquaux, and Balázs Kégl. Similarity encoding for learning with dirty categorical variables. *Machine Learning*, 107(8-10):1477–1494, 2018.

Tianfeng Chai and Roland R Draxler. Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, 7(3):1247–1250, 2014.

Heather Chappells and Elizabeth Shove. Debating the future of comfort: environmental sustainability, energy consumption and the indoor environment. *Building Research & Information*, pages 32–40, 2005.

Dong Chen, Sean Barker, Adarsh Subbaswamy, David Irwin, and Prashant Shenoy. Non-intrusive occupancy monitoring using smart meters. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, pages 1–8, 2013.

Jingjie Chen, Peng Du, and Daoxian Ren. A sort of co2 concentration monitoring system based on energy efficiency for terminal. In *Control and Decision Conference (CCDC), 2016 Chinese*, pages 6814–6817. IEEE, 2016.

Shisheng Chen, Kuniaki Mihara, and Jianxiu Wen. Time series prediction of co2, tvoc and hcho based on machine learning at different sampling points. *Building and Environment*, 146:238–246, 2018a.

Xiaojun Chen, Yunming Ye, Graham Williams, and Xiaofei Xu. A survey of open source data mining systems. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 3–14. Springer, 2007.

Zhenghua Chen, Chaoyang Jiang, and Lihua Xie. Building occupancy estimation and detection: A review. *Energy and Buildings*, 169:260–270, 2018b.

Yun Cheng, Xiucheng Li, Zhijun Li, Shouxu Jiang, Yilong Li, Ji Jia, and Xiaofan Jiang. Aircloud: a cloud-based air-quality monitoring system for everyone. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, pages 251–265. ACM, 2014.

Yin-Wong Cheung and Kon S. Lai. Lag order and critical values of the augmented dickey–fuller test. *Journal of Business & Economic Statistics*, pages 277–280, 1995.

Jacob Cohen, Patricia Cohen, Stephen G West, and Leona S Aiken. *Applied multiple regression/correlation analysis for the behavioral sciences*. Routledge, 2013.

Alexis Daveria and Lennart Schoors. Can i use... browser support tables for modern web technologies, Nov 2019. URL https://caniuse.com.

David A Dickey and Wayne A Fuller. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American statistical association*, 74(366a):427–431, 1979.

Thomas G Dietterich. Machine-learning research. *AI magazine*, page 97, 1997.

Thomas G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, pages 1–15. Springer-Verlag, 2000a. URL http://dl.acm.org/citation.cfm?id=648054.743935.

Thomas G Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, pages 139–157, 2000b.

Jiagen Ding, S-Y Cheung, C-W Tan, and Pravin Varaiya. Signal processing of sensor node data for vehicle detection. In *Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No. 04TH8749)*, pages 70–75. IEEE, 2004.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

Yanjie Duan, Yisheng Lv, and Fei-Yue Wang. Travel time prediction with lstm neural network. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1053–1058. IEEE, 2016.

Wlodzislaw Duch, Karol Grudzinski, and G Stawski. Symbolic features in neural networks. In *In Proceedings of the 5th Conference on Neural Networks and Their Applications*. Citeseer, 2000.

Afrooz Ebadat, Giulio Bottegal, Damiano Varagnolo, Bo Wahlberg, and Karl H Johansson. Estimation of building occupancy levels through environmental signals deconvolution. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, pages 1–8. ACM, 2013.

Energy repository. International energy statistics, 2014. URL https://www.eia.gov/beta/international/data/browser/#/?vs=INTL.44-1-AFRC-QBTU.A&vo=0&v=H&start=1980&end=2014.

Varick L Erickson and Alberto E Cerpa. Occupancy based demand response hvac control strategy. In *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, pages 7–12, 2010.

Christen Erlingsson and Petra Brysiewicz. A hands-on guide to doing content analysis. *African Journal of Emergency Medicine*, 7(3):93–99, 2017.

Valentina Fabi, Rune Vinther Andersen, Stefano Corgnati, and Bjarne W Olesen. Occupants' window opening behaviour: A literature review of factors influencing occupant behaviour and models. *Building and Environment*, 58:188–198, 2012.

Wen-Xiang Fang, Po-Chao Lan, Wan-Rung Lin, Hsiao-Chen Chang, Hai-Yen Chang, and Yi-Hsien Wang. Combine facebook prophet and lstm with bpnn forecasting financial markets: the morgan taiwan index. In *2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pages 1–2. IEEE, 2019.

Poul O Fanger et al. Thermal comfort. analysis and applications in environmental engineering. *Thermal comfort. Analysis and applications in environmental engineering.*, 1970.

M Fasiuddin and I Budaiwi. Hvac system strategies for energy conservation in commercial buildings in saudi arabia. *Energy and Buildings*, pages 3457–3466, 2011.

Autumn Faulkner and Contributor. Lucidchart for easy workflow mapping. *Serials Review*, 44(2):157–162, 2018.

Jennifer Fereday and Eimear Muir-Cochrane. Demonstrating rigor using thematic analysis: A hybrid approach of inductive and deductive coding and theme development. *International journal of qualitative methods*, 5(1):80–92, 2006.

Shiaw-Fen Ferng and Lee Li-Wen. Indoor air quality assessment of daycare facilities with carbon dioxide, temperature, and humidity as indicators. *Journal of Environmental Health*, 65(4):14, 2002.

Roy T Fielding and Richard N Taylor. *Architectural styles and the design of network-based software architectures*, volume 7. University of California, Irvine Irvine, 2000.

Department for Trade and Industry. Energy white paper–our energy future: Creating a low carbon economy, 2003.

Ben Frain. *Responsive web design with HTML5 and CSS3*. Packt Publishing Ltd, 2012.

Ray J Frank, Neil Davey, and Stephen P Hunt. Input window size and neural network predictors. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, volume 2, pages 237–242. IEEE, 2000.

Alex A Freitas. Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter*, 15(1):1–10, 2014.

Yoav Freund and Robert Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory*, pages 23–37. Springer, 1995.

Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *icml*, pages 148–156, 1996.

Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

Jerome H Friedman and Bogdan E Popescu. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, pages 916–954, 2008.

Elliott T. Gall, Toby Cheung, Irvan Luhung, Stefano Schiavon, and William W. Nazaroff. Real-time monitoring of personal exposures to carbon dioxide. *Building and Environment*, pages 59 – 67, 2016. URL //www.sciencedirect.com/science/article/pii/S0360132316301421.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *IET digital library*, 1999.

Barney G Glaser and Anselm L Strauss. *Discovery of grounded theory: Strategies for qualitative research*. Routledge, 2017.

Barney G Glaser, Anselm L Strauss, and Elizabeth Strutzel. The discovery of grounded theory; strategies for qualitative research. *Nursing research*, 17(4):364, 1968.

Jessica B Hamrick. Creating and grading ipython/jupyter notebook assignments with nbgrader. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, pages 242–242. ACM, 2016.

Andrew C Harvey and Simon Peters. Estimation procedures for structural time series models. *Journal of Forecasting*, 9(2):89–108, 1990.

Daniel Hernández-Lobato, Gonzalo Martínez-Muñoz, and Alberto Suárez. Pruning in ordered regression bagging ensembles. In *Neural Networks, 2006. IJCNN'06. International Joint Conference on*, pages 1266–1273. IEEE, 2006.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Tianzhen Hong, Sarah C Taylor-Lange, Simona D'Oca, Da Yan, and Stefano P Corgnati. Advances in research and applications of energy-related occupant behavior in buildings. *Energy and buildings*, 116:694–702, 2016.

Yen-Chia Hsu, Jennifer Cross, Paul Dille, Michael Tasota, Beatrice Dias, Randy Sargent, Ting-Hao (Kenneth) Huang, and Illah Nourbakhsh. Smell pittsburgh: Community-empowered mobile smell reporting system. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, page 65–79, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362726. doi: 10.1145/3301275.3302293. URL https://doi.org/10.1145/3301275.3302293.

Chaoyang Jiang, Mustafa K. Masood, Yeng Chai Soh, and Hua Li. Indoor occupancy estimation from carbon dioxide concentration. *Energy and Buildings*, pages 132 – 141, 2016. URL http://www.sciencedirect.com/science/article/pii/S0378778816308027.

Yifei Jiang, Kun Li, Lei Tian, Ricardo Piedrahita, Xiang Yun, Omkar Mansata, Qin Lv, Robert P Dick, Michael Hannigan, and Li Shang. Maqs: a personalized mobile sensing system for indoor air quality monitoring. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 271–280, 2011.

D Johnston, R Lowe, and M Bell. An exploration of the technical feasibility of achieving co 2 emission reductions in excess of 60% within the uk housing stock by the year 2050. *Energy Policy*, 33(13):1643–1659, 2005.

L Kajtar, L Herczeg, E Lang, T Hrustinzky, and L Banhidi. Influence of carbon-dioxide pollutant on human well-being and work intensity. In *Proc Healthy Buildings Conf*, pages 85–90, 2006.

Mohamed Khalifa. Reducing emergency department crowding using health analytics methods: designing anevidence based decision algorithm. *Procedia Computer Science*, 63:409–416, 2015.

Sunyoung Kim and Eric Paulos. Inair: Sharing indoor air quality measurements and visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1861–1870. ACM, 2010. ISBN 978-1-60558-929-9.

Barbara Kitchenham. Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004):1–26, 2004.

Paula Kivimaa and Mari Martiskainen. Dynamics of policy change and intermediation: the arduous transition towards low-energy homes in the united kingdom. *Energy research & social science*, 44:83–99, 2018.

SB Kotsiantis, Dimitris Kanellopoulos, and PE Pintelas. Data preprocessing for supervised leaning. *International Journal of Computer Science*, 1(2):111–117, 2006.

Ben Kröse, Ben Krose, Patrick van der Smagt, and Patrick Smagt. An introduction to neural networks. 1993.

Anuj Kumar, Hiesik Kim, and Gerhard P Hancke. Environmental monitoring systems: A review. *IEEE Sensors Journal*, 13(4):1329–1339, 2012.

HM Künzel, A Holm, D Zirkelbach, and AN Karagiozis. Simulation of indoor temperature and humidity conditions including hygrothermal interactions with the building envelope. *Solar Energy*, 78(4):554–561, 2005.

Rob Law. Initially testing an improved extrapolative hotel room occupancy rate forecasting technique. *Journal of Travel & Tourism Marketing*, 16(2-3):71–77, 2004.

Corinne Le Quéré, Robert J Andres, T Boden, Thomas Conway, Richard A Houghton, Joanna I House, Gregg Marland, Glen Philip Peters, Guido Van der Werf, Anders Ahlström, et al. The global carbon budget 1959–2011. *Earth System Science Data Discussions*, pages 1107–1157, 2012.

Jiaming Li, Josh Wall, and Glenn Platt. Indoor air quality control of hvac system. In *Proceedings of the 2010 International Conference on Modelling, Identification and Control*, pages 756–761. IEEE, 2010.

Michael J Lipsett, DJ Shusterman, and RR Beard. Inorganic compounds of carbon, nitrogen, and oxygen. *Patty's Industrial Hygiene and Toxicology. New York, NY: John Wiley & Sons*, pages 4523–4643, 1994.

Peter Lipton. Contrastive explanation. *Royal Institute of Philosophy Supplements*, 27:247–266, 1990.

Sharon L Lohr. *Sampling: design and analysis*. Nelson Education, 2009.

Wen Long, Zhichen Lu, and Lingxiao Cui. Deep learning-based feature engineering for stock price movement prediction. *Knowledge-Based Systems*, 164:163–173, 2019.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774, 2017.

Howard Lune and Bruce L Berg. *Qualitative research methods for the social sciences*. Pearson Higher Ed, 2016.

Helmut Lütkepohl. *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.

James G MacKinnon. Critical values for cointegration tests. In *Eds.), Long-Run Economic Relationship: Readings in Cointegration*. Citeseer, 1991.

James G MacKinnon. Approximate asymptotic distribution functions for unit-root and cointegration tests. *Journal of Business & Economic Statistics*, 12(2):167–176, 1994.

O'Reilly Media. title= Local Interpretable Model-Agnostic Explanations (LIME): An Introduction. [online] website= <https://www.oreilly.com/content/introduction-to-local-interpretable-model-agnostic-explanations-lime/> [Accessed 23 March 2021] Marco Tulio Ribeiro, C., 2021.

Gonçalo Marques and Rui Pitarma. Health informatics for indoor air quality monitoring. In *Information Systems and Technologies (CISTI), 2016 11th Iberian Conference on*, pages 1–6. IEEE, 2016.

Mustafa K Masood, Yeng Chai Soh, and Victor W-C Chang. Real-time occupancy estimation using environmental parameters. In *Neural Networks (IJCNN), 2015 International Joint Conference on*, pages 1–8. IEEE, 2015.

Mustafa K Masood, Yeng Chai Soh, and Chaoyang Jiang. Occupancy estimation from environmental parameters using wrapper and hybrid feature selection. *Applied Soft Computing*, 60:482–494, 2017.

John Mayer, James E Van't Slot, and Daniel Rawlings. Environmental monitoring system, April 20 2004. US Patent 6,725,180.

Melissa L McCarthy, Dominik Aronsky, Ian D Jones, James R Miner, Roger A Band, Jill M Baren, Jeffrey S Desmond, Kevin M Baumlin, Ru Ding, and Robert Shesser. The emergency department occupancy rate: a simple measure of emergency department crowding? *Annals of emergency medicine*, 51(1):15–24, 2008.

João Mendes-Moreira, Carlos Soares, Alípio Mário Jorge, and Jorge Freire De Sousa. Ensemble approaches for regression: A survey. *ACM Comput. Surv.*, pages 10:1–10:40, 2012. URL http://doi.acm.org/10.1145/2379776.2379786.

Abigail M Methley, Stephen Campbell, Carolyn Chew-Graham, Rosalind McNally, and Sudeh Cheraghi-Sohi. Pico, picos and spider: a comparison study of specificity and sensitivity in three search tools for qualitative systematic reviews. *BMC health services research*, 14 (1):579, 2014.

Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

Donald K Milton, P Mark Glencross, and Michael D Walters. Risk of sick leave associated with outdoor air supply rate, humidification, and occupant complaints. *Indoor air*, pages 212–221, 2000.

Tom M Mitchell. Does machine learning really work? *AI magazine*, 18(3):11–11, 1997.

Babak Moatamed, Farhad Shahmohammadi, Ramin Ramezani, Arash Naeim, Majid Sarrafzadeh, et al. Low-cost indoor health monitoring system. In *Wearable and Implantable Body Sensor Networks (BSN), 2016 IEEE 13th International Conference on*, pages 159–164. IEEE, 2016.

Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2019.

Janice M Morse and Peggy Anne Field. *Nursing research: The application of qualitative approaches*. Nelson Thornes, 1995.

Harvey Motulsky and Arthur Christopoulos. *Fitting models to biological data using linear and nonlinear regression: a practical guide to curve fitting*. Oxford University Press, 2004.

Tuan Anh Nguyen and Marco Aiello. Energy intelligent buildings based on user activity: A survey. *Energy and buildings*, 56:244–257, 2013.

Lionel M Ni, Yunhao Liu, Yiu Cho Lau, and Abhishek P Patil. Landmarc: Indoor location sensing using active rfid. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003.(PerCom 2003).*, pages 407–415. IEEE, 2003.

J Fergus Nicol and Michael A Humphreys. Adaptive thermal comfort and sustainable thermal standards for buildings. *Energy and buildings*, 34(6):563–572, 2002.

Frauke Oldewurtel, Alessandra Parisio, Colin N. Jones, Dimitrios Gyalistras, Markus Gwerder, Vanessa Stauch, Beat Lehmann, and Manfred Morari. Use of model predictive control and weather forecasts for energy efficient building climate control. *Energy and Buildings*, pages 15 – 27, 2012.

Bjarne W Olesen and KC Parsons. Introduction to thermal comfort standards and to the proposed new version of en iso 7730. *Energy and buildings*, pages 537–548, 2002.

Online Repository. Netatmo weather station, 2017a. URL https://www.netatmo.com/product/weather/. Last accessed on 7th March 2017.

Online Repository. Ubiquiti - unifi® video camera g3, 2020b. URL https://www.ui.com/unifi-video/unifi-video-camera-g3/. Last accessed on 20 March 2020.

David Opitz and Richard Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, pages 169–198, 1999.

Alexandros Pantazaras, Siew Eang Lee, Mattheos Santamouris, and Junjing Yang. Predicting the co2 levels in buildings using deterministic and identified models. *Energy and Buildings*, 127:774–785, 2016.

Oren Patashnik. Designing bibtex styles, 1988.

Neal Patwari, Alfred O Hero, Matt Perkins, Neiyer S Correal, and Robert J O'dea. Relative location estimation in wireless sensor networks. *IEEE Transactions on signal processing*, 51(8):2137–2148, 2003.

Theis Heidmann Pedersen, Kasper Ubbe Nielsen, and Steffen Petersen. Method for room occupancy detection based on trajectory of indoor climate sensor data. *Building and Environment*, 115:147–156, 2017.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, pages 2825–2830, 2011.

Changhai Peng, Kun Qian, and Chenyang Wang. Design and application of a voc-monitoring system based on a zigbee wireless sensor network. *IEEE Sensors Journal*, 15(4):2255–2268, 2014.

Yuzhen Peng, Adam Rysanek, Zoltán Nagy, and Arno Schlüter. Using machine learning techniques for occupancy-prediction-based cooling control in office buildings. *Applied energy*, 211:1343–1358, 2018.

Luis Pérez-Lombard, José Ortiz, and Christine Pout. A review on buildings energy consumption information. *Energy and buildings*, pages 394–398, 2008.

Michael P Perrone and Leon N Cooper. When networks disagree: Ensemble methods for hybrid neural networks. Technical report, DTIC Document, 1992.

Andrew Persily. Challenges in developing ventilation and indoor air quality standards: The story of {ASHRAE} standard 62. *Building and Environment*, pages 61 – 69, 2015. URL //www.sciencedirect.com/science/article/pii/S0360132315000839. Fifty Year Anniversary for Building and Environment.

Steffen Petersen, Theis Heidmann Pedersen, Kasper Ubbe Nielsen, and Michael Dahl Knudsen. Establishing an image-based ground truth for validation of sensor data-based room occupancy detection. *Energy and Buildings*, 130:787–793, 2016.

Samuel Privara, Jan Širokỳ, Lukáš Ferkl, and Jiří Cigler. Model predictive control of a building heating system: The first experience. *Energy and Buildings*, 43(2):564–572, 2011.

Sutharshan Rajasegarar, Peng Zhang, Yang Zhou, Shanika Karunasekera, Christopher Leckie, and Marimuthu Palaniswami. High resolution spatio-temporal monitoring of air pollutants using wireless sensor networks. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, pages 1–6. IEEE, 2014.

J Sunil Rao and Robert Tibshirani. The out-of-bootstrap method for model averaging and selection. *University of Toronto*, 1997.

Zuzana Reitermanova. Data splitting. In *WDS*, volume 10, pages 31–36, 2010.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

M Rocca. Health and well-being in indoor work environments: a review of literature. In *Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.

Carl Rogers. *Client Centred Therapy (New Ed)*. Hachette UK, 2012.

Fabio Roli, Giorgio Giacinto, and Gianni Vernazza. Methods for designing multiple classifier systems. In *International Workshop on Multiple Classifier Systems*, pages 78–87. Springer, 2001.

P Romeo. Strange time upon us. *Restaurant Business*, 96(14):6, 1997.

Kay Römer, Philipp Blum, and Lennart Meier. Time synchronization and calibration in wireless sensor networks. *Handbook of sensor networks: Algorithms and architectures*, page 199, 2005.

Niall Rooney, David Patterson, Sarab Anand, and Alexey Tsymbal. Dynamic integration of regression models. In *International Workshop on Multiple Classifier Systems*, pages 164–173. Springer, 2004.

SN Rudnick and DK Milton. Risk of indoor airborne infection transmission estimated from carbon dioxide concentration. *Indoor air*, 13(3):237–245, 2003.

Thomas Henry Runge, Bruce Martin Downie, and Michael Henry Runge. Wireless environmental sensor system, September 17 2002. US Patent 6,452,499.

Shaharil Mad Saad, Abdul Rahman Mohd Saad, Azman Muhamad Yusof Kamarudin, Ammar Zakaria, and Ali Yeon Md Shakaff. Indoor air quality monitoring system using wireless sensor network (wsn) with web interface. In *Electrical, Electronics and System Engineering (ICEESE), 2013 International Conference on*, pages 60–64. IEEE, 2013.

Said E Said and David A Dickey. Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika*, 71(3):599–607, 1984.

RJ Salway, R Valenzuela, JM Shoenberger, WK Mallon, and A Viccellio. Emergency department (ed) overcrowding: evidence-based answers to frequently asked questions. *Revista Médica Clínica Las Condes*, 28(2):213–219, 2017.

UNEP Sbci. Buildings and climate change: Summary for decision-makers. *United Nations Environmental Programme, Sustainable Buildings and Climate Initiative, Paris*, pages 1–62, 2009.

O Seppanen and WJ Fisk. Relationship of sbs-symptoms and ventilation system type in office buildings. *Building*, 2002.

O. A. Seppänen and W. J. Fisk. Summary of human responses to ventilation. *Indoor Air*, pages 102–118, 2004. ISSN 1600-0668. doi: 10.1111/j.1600-0668.2004.00279.x. URL http://dx.doi.org/10.1111/j.1600-0668.2004.00279.x.

Helen Shen. Interactive notebooks: Sharing the code. *Nature News*, 515(7525):151, 2014.

Derek G Shendell, Richard Prill, William J Fisk, Michael G Apte, David Blake, and David Faulkner. Associations between classroom co2 concentrations and student attendance in washington and idaho. *Indoor air*, pages 333–341, 2004.

LD Shorrock, J Henderson, JI Utley, and GA Walters. Carbon emission reductions from energy efficiency improvements to the uk housing stock. *Building Research Establishment, Report BR435*, 2001.

Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3145–3153. JMLR. org, 2017.

Abhishek Singh, Yadvendra Pandey, Ashok Kumar, Manoj Kumar Singh, Anuj Kumar, and Subhas Chandra Mukhopadhyay. Ventilation monitoring and control system for high rise historical buildings. *IEEE Sensors Journal*, 17(22):7533–7541, 2017.

Joanna Smith and Jill Firth. Qualitative data analysis: the framework approach. *Nurse researcher*, 18(2):52–62, 2011.

Donald F Specht. A general regression neural network. *IEEE transactions on neural networks*, 2(6):568–576, 1991.

Robert A Stine. Bootstrap prediction intervals for regression. *Journal of the American Statistical Association*, 80(392):1026–1031, 1985.

Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, pages 583–617, 2002.

Jan Sundell. On the history of indoor air quality and health. *Indoor air*, 14(s7):51–58, 2004.

Jan Sundell. Reflections on the history of indoor air science, focusing on the last 50 years. *Indoor Air*, 2017. ISSN 1600-0668. URL http://dx.doi.org/10.1111/ina.12368.

Sean J Taylor and Benjamin Letham. Forecasting at scale. *PeerJ Preprints*, 5:e3190v2, September 2017. ISSN 2167-9843. doi: 10.7287/peerj.preprints.3190v2. URL https://doi.org/10.7287/peerj.preprints.3190v2.

InVision team. Digital product design, workflow & collaboration, 2020. URL https://www.invisionapp.com/.

Photoscape team. Photoscape x for mac and windows 10, 2019. URL http://x.photoscape.org/.

George R Terrell. The maximal smoothing principle in density estimation. *Journal of the American Statistical Association*, 85(410):470–477, 1990.

Howell Tong. *Threshold models in non-linear time series analysis*, volume 21. Springer Science & Business Media, 2012.

Henry Tsai and Zheng Gu. Optimizing room capacity and profitability for hong kong hotels. *Journal of Travel & Tourism Marketing*, 29(1):57–68, 2012.

James C Tyhurst. Non-intrusive occupancy detection methods and models. Technical report, AIR FORCE INSTITUTE OF TECHNOLOGY WRIGHT-PATTERSON AFB OH WRIGHT-PATTERSON . . . , 2019.

Sebastian Uziel, Thomas Elste, Wolfram Kattanek, Danilo Hollosi, Stephan Gerlach, and Stefan Goetze. Networked embedded acoustic processing system for smart building applications. In *Design and Architectures for Signal and Image Processing (DASIP), 2013 Conference on*, pages 349–350. IEEE, 2013.

Tommi Vehviläinen, Harri Lindholm, Hannu Rintamäki, Rauno Pääkkönen, Ari Hirvonen, Olli Niemi, and Juha Vinha. High indoor co2 concentrations in an office environment increases the transcutaneous co2 level and sleepiness during cognitive work. *Journal of Occupational and Environmental Hygiene*, pages 19–29, 2016. URL http://dx.doi.org/10.1080/15459624.2015.1076160.

Hao Wang, Rohit P Ojha, Richard D Robinson, Bradford E Jackson, Sajid A Shaikh, Chad D Cowden, Rath Shyamanand, JoAnna Leuck, Chet D Schrader, and Nestor R Zenarosa. Optimal measurement interval for emergency department crowding estimation tools. *Annals of emergency medicine*, 70(5):632–639, 2017.

Zhu Wang and Lingfeng Wang. Indoor air quality control for energy-efficient buildings using co 2 predictive model. In *Industrial Informatics (INDIN), 2012 10th IEEE International Conference on*, pages 133–138. IEEE, 2012.

Pawel Wargocki, David P Wyon, Yong K Baik, Geo Clausen, and P Ole Fanger. Perceived air quality, sick building syndrome (sbs) symptoms and productivity in an office with two different pollution loads. *Indoor air*, 9(3):165–179, 1999.

Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 1113–1120, 2009.

Cort J Willmott and Kenji Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, 30(1):79–82, 2005.

CL Wu, KW Chau, and YS Li. Predicting monthly streamflow using data-driven models coupled with data-preprocessing techniques. *Water Resources Research*, 45(8), 2009.

DP Wyon and Pawel Wargocki. The sbs symptoms and environmental perception of office workers in the tropics at two air temperatures and two ventilation rates. In *Procedings of the Healthy Buildings 2003 Conference*, pages 182–188, 2003.

Junjing Yang, Mattheos Santamouris, and Siew Eang Lee. Review of occupancy sensing systems and occupancy modeling methodologies for the application in institutional buildings. *Energy and Buildings*, 121:344–349, 2016.

Zheng Yang, Nan Li, Burcin Becerik-Gerber, and Michael Orosz. A multi-sensor based occupancy estimation model for supporting demand driven hvac operations. Society for Computer Simulation International, 2012. URL http://dl.acm.org/citation.cfm?id=2339453.2339455.

Kevin Yank. *Build your own database driven website using PHP & MySQL*. SitePoint Pty Ltd, 2004.

Bowen Zhang, Pavankumar Murali, MM Dessouky, and David Belson. A mixed integer programming approach for allocating operating room capacity. *Journal of the Operational Research Society*, 60(5):663–673, 2009.

Zhiqiang Zhang, Xuebin Gao, Jit Biswas, and Jian Kang Wu. Moving targets detection and localization in passive infrared sensor networks. In *Information Fusion, 2007 10th International Conference on*, pages 1–6. IEEE, 2007.

Hai-xiang Zhao and Frédéric Magoulès. A review on the prediction of building energy consumption. *Renewable and Sustainable Energy Reviews*, pages 3586–3592, 2012.

Hengyang Zhao, Qi Hua, Hai-Bao Chen, Yaoyao Ye, Hai Wang, Sheldon X-D Tan, and Esteban Tlelo-Cuautle. Thermal-sensor-based occupancy detection for smart buildings using machine-learning methods. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 23(4):1–21, 2018.

Qiang-Li Zhao, Yan-Huang Jiang, and Ming Xu. A fast ensemble pruning algorithm based on pattern mining process. *Data Mining and Knowledge Discovery*, pages 277–292, 2009.

Jianhong Zou, Qianchuan Zhao, Wen Yang, and Fulin Wang. Occupancy detection in the office by analyzing surveillance videos and its application to building energy conservation. *Energy and Buildings*, 152:385–398, 2017.