ANALYZING WIFI CONNECTION COUNTS in COMMERCIAL/INSTITUTIONAL BUILDINGS to ESTIMATE/PREDICT OCCUPANCY for OPTIMIZING BUILDINGS' SYSTEMS OPERATION

Nastaran Alishahi

A Thesis
In the Department of
Building, Civil, and Environmental Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Master of Applied Science (Building Engineering)

at Concordia University
Montréal, Québec, Canada

July 2021

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Nastaran Alishahi

Entitled: **Analyzing WiFi connection counts in commercial/institutional buildings to estimate/predict occupancy patterns for optimizing buildings' systems operation**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Building Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final Examining Committee:

_____ Chair
Dr. Ursula Eicker

_____ External Examiner
Dr. Amin Hammad

_____ Thesis Co-supervisor
Dr. Mazdak Nik-Bakht

_____ Thesis Co-supervisor
Dr. Mohamed M. Ouf

Approved by

_____
Dr. Michelle Nokken, Graduate Program Director

Date August 10, 2021

_____
Dr. Mourad Debbabi, Dean of Faculty

Abstract

Analyzing WiFi connection counts in commercial/institutional buildings to estimate/predict occupancy patterns for optimizing buildings' systems operation

Nastaran Alishahi

Accurate occupancy information can help in optimizing the operation of building systems. To obtain this information, previous studies suggested using WiFi connection counts due to their strong correlation with occupancy counts. However, validating this correlation and investigating its variation have remained limited due to challenges regarding collecting ground-truth data. Moreover, the difficulty of integrating real-time WiFi traffic data in building automation systems hinders wide-scale deployment of this approach. This research addressed these gaps by proposing a methodology, including two modules focused on developing frameworks, for (i) validating the correlation between WiFi connection counts and actual building occupancy counts by using continuous ground-truth data collected from camera-based occupancy counters; and (ii) extracting occupancy indicators from WiFi connection count data which can then be used for updating control sequences.

The proposed research was implemented in two institutional buildings to validate the proposed methods in two case studies. Results of the first case study showed Hour of the day, Day of the week, as well as occupancy level, affect the correlation between WiFi and occupancy counts. Furthermore, the proposed models could successfully estimate real-time occupancy counts and predict day-ahead occupancy counts with an average accuracy (R^2) of 0.97 and 0.87, respectively. Moreover, the results of the second case study revealed that the proposed models could successfully predict weekly building occupancy patterns, with an average accuracy (R^2_D) of 0.90. Furthermore, the analysis identified peak occupancy timing, as well as arrival/departure times variations between different zones. These findings provided a proof-of-concept for the proposed methodology and demonstrated the potential of using WiFi connection count for estimating/forecasting occupancy counts at a large scale and extracting actionable information to optimize buildings' system operation based on buildings' unique occupancy patterns.

ACKNOWLEDGMENT

First and foremost, I would like to thank my supervisors Dr. Mazdak Nik-Bakht and Dr. Mohamed M. Ouf. I am very grateful for having the chance of working with these two knowledgeable and inspiring professors. They kindly guided me with their encouraging words, continuous support, and constructive feedback.

I would like to thank my committee members, Dr. Ursula Eicker and Dr. Amin Hammad for kindly agreeing to be part of this thesis committee. I am grateful for their time and insightful comments.

I would like to thank the National Science and Engineering Research Council of Canada (NSERC) and Concordia University for the grants they provided to support this research.

I would like to acknowledge the support of three teams in this research including the Instructional and Information Technology Services (IITS) team of Concordia University for providing the WiFi connection count data; the Information Systems and Technology team of Concordia University for providing access to the camera-based occupancy counters; and the Security and Facilities Management departments of Concordia University for providing the drawings of the building.

I would like to thank my friends and colleagues at COMPLECCiTY and IBCL (Intelligent Buildings and Cities Lab), two research labs at Concordia University, for their valuable comments and guidance as well as the good times we spent together virtually through the COVID-19 pandemic. Special thanks to Arash Hosseini, who helped me through the data collection process, as well as Abdelhady Hosny and Leila Rafati, who was always ready to help without any hesitation.

I would like to thank my beloved parents, Maryam and Saied, and my dear brother, Nima for their encouragement and support in all stages of my life. Finally, I could not have completed this thesis without the support of my partner, Mahyar, who was always there alongside me. I would like to thank him for his support and understanding throughout the course of this thesis, amid a pandemic.

PREFACE

This thesis is submitted for the Degree of Master of Applied Science (Building Engineering) at Concordia University. The research was conducted under the supervision of Dr. Mazdak Nik-Bakht and Dr. Mohamed M. Ouf, both in the Department of Building, Civil, and Environmental Engineering. In collaboration with both supervisors, the research conducted for this thesis formed three manuscripts as follows:

(i) Alishahi, N., Nik-Bakht, M., & Ouf, M. M. (2021). A framework to identify key occupancy indicators for optimizing building operation using WiFi connection count data. *Building and Environment, 200*, 107936. https://doi.org/10.1016/j.buildenv.2021.107936

(ii) Alishahi, N., Ouf, M. M., & Nik-Bakht, M. (2021). Using WiFi connection counts and camera-based occupancy counts to estimate and predict building occupancy. Manuscript submitted for publication.

(iii) Alishahi, N., Ouf, M. M., & Nik-Bakht, M. (2021). Investigating the correlation dynamism between WiFi connection counts and camera-based occupancy counts. Accepted Building Simulation 2021 Conference, Bruges, Belgium, 1-3 September 2021.

Major parts of the introduction and literature review in *CHAPTER 1* and *CHAPTER 2*, are used in the manuscripts. The methodology proposed in *CHAPTER 3*, is a combination of methodologies presented in these three manuscripts. The case study and results of the experiment on it in *CHAPTER 4* are presented in manuscripts (ii) and (iii) while the case study and results of the experiment on it in *CHAPTER 5* are presented in manuscripts (i).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

3D	Three Dimensional
AHU	Air Handling Units
ANN	Artificial Neural Networks
AP	Access Point
ARIMA	Autoregressive Integrated Moving Average
ARMA	Auto-Regressive Moving Average
BAS	Building Automation Systems
BIM	Building Information Model
BLE	Bluetooth Low Energy
CART	Classification and Regression Tree
CDBLSTM	Convolutional Deep Bidirectional Long Short-Term Memory
CO_2	Carbon Dioxide
CSI	Channel State Information
DBI	Davies Bouldin Index
DMTWI	Dynamic Markov Time-Window Inference
DT	Decision Tree
ELM	Extreme Learning Machine
FS-ELM	Feature Scaled Extreme Learning Machine
GLM	Generalized Linear Models
HVAC	Heating, Ventilation, and Air Conditioning
IITS	Instructional and Information Technology Services
IQR	Interquartile Range
KNN	K-Nearest Neighbor
LDA	Linear Discriminant Analysis
LEED	Leadership in Energy and Environmental Design

LSTM	Long-Term Short-Term Memory
MAC	Media Access Control
MAPE	Mean Absolute Percentage Error
M-FRNN	Markov based Feedback Recurrent Neural Network
MLE	Maximum Likelihood Estimation
MLR	Multiple Linear Regression
NB	Naive Bayesian
NRMSD	Normalized Root Mean Square Deviation
NRMSE	Normalized Root Mean Square Error
OLS	Ordinary Least Squares
PIR	Passive Infrared Sensors
RF	Random Forest
RFID	Radio Frequency Identification
RMSE	Root Mean Squared Error
RMSPE	Root Mean Squared Percentage Error
RNN	Recurrent Neural Networks
RSSI	Received Signal Strength Indicator
SD	Standard Deviation
SVM	Support Vector Machines
SVR	Support Vector Regression
TAN	Tree Augmented Naive Bayes Network
WiFi	Wireless Fidelity

CHAPTER 1: INTRODUCTION

1.1 Background

Buildings are reported to be responsible for a significant percentage of energy consumption all around the world. Although this percentage varies in different countries, in 2018, it was estimated around 30% in Canada (13% for commercial/institutional buildings and 17% for residential buildings), while around 70% of the energy consumed in each sector was allocated to the space heating and cooling (NRCan, 2018). The Heating, Ventilation, and Air Conditioning (HVAC) systems are recognized as the largest energy end-use in these buildings while accounting for about 40-50% of the energy consumption in buildings which is mostly due to inefficient operation of these systems (Gul & Patidar, 2015). In a study by Park et al. (2019), centralization and automation of building systems were identified as the main source of this inefficiency, since the building systems neglect occupancy variation and do not adapt to irregular or partial occupancy. Esrafilian-Najafabadi & Haghighat (2021) also reviewed user-defined schedules as well as other strategies including reactive and predictive control of building systems as strategies that were not fully successful in adapting to actual occupancy patterns. Following these problems, occupant-centric (or centered) controls (OCC) based on accurate occupancy information learned using data mining and machine learning algorithms has emerged recently.

Previous studies showed a significant reduction in building energy consumption (i.e., up to around 50%) through adapting HVAC systems to actual occupancy patterns (Balaji et al., 2013; Z. Yang et al., 2014; Z. Yang & Becerik-Gerber, 2014; F. Wang et al., 2017; Capozzoli et al., 2017; W. Wang, Wang, et al., 2018; Peng et al., 2018; W. Wang, Hong, Li, et al., 2019; Jung & Jazizadeh, 2019; Dai et al., 2020). These energy-saving potentials show the importance of obtaining accurate occupancy information for optimizing the operation of building systems.

Despite the previous findings, most buildings are still being operated under the assumption of full or nearly full occupancy, regardless of actual building occupancy; which can lead to significant energy waste and occupant discomfort (Hong et al., 2017; Ouf et al., 2019; O'Brien et al., 2020). These stereotypical schedules, recommended by standards based on the building type (NECB, 2015; ASHRAE, 2019), consider these conservative fixed schedules. Previous studies have proved considerable temporal and spatial deviations in building occupancy patterns, especially in large

commercial and institutional buildings (Gul & Patidar, 2015; J. Yang et al., 2016). In such buildings, occupants arrive and/or leave the spaces at different times of the day, while peak occupancy can be influenced by different events. However, many studies suggest that although significant variations exist in occupancy patterns of different buildings (even those of similar types), they tend to follow relatively repetitive patterns for individual buildings (D'Oca & Hong, 2015; Liang et al., 2016; Capozzoli et al., 2017; Hobson et al., 2020; Ding et al., 2021). In other words, once the occupancy patterns of a specific building are thoroughly investigated and identified, no significant changes should be expected in those patterns, unless exceptional events take place. On the other hand, some previous studies found that these repetitive patterns may also change over time due to transformations in space utilization patterns followed by technological advances and the expected increase in telecommuting (Ouf et al., 2019). Therefore, it is critical to obtain accurate occupancy information in order to optimize the operation of building systems based on the actual occupancy levels.

Adapting building systems' operation to occupancy variations can provide significant energy savings, but this is typically constrained by the unavailability of occupancy information. In recent years, researchers investigated various technologies for obtaining occupancy information, such as carbon dioxide (CO_2) or Passive Infrared Sensors – PIR (Z. Chen et al., 2018; Sun et al., 2020). These studies focused on developing HVAC control strategies to utilize occupancy information obtained from the adoption of these methods/technologies (Ouf et al., 2019). However, many of these technologies have limitations in terms of accuracy, cost, intrusiveness, and privacy. For example, while demand-controlled ventilation using CO_2 sensors has been successfully commercialized, its deployment in large buildings is still relatively limited due to the significant cost and maintenance requirements (Ouf et al., 2017). Among the technologies used for collecting the occupancy information, implicit occupancy sensing systems (e.g. WiFi connection history, keyboard and mouse activities, security card access systems, etc.) have attracted increasing attention due to their lower cost compared to conventional ones (e.g. PIR, CO_2 sensors, etc.) (Shen et al., 2017). WiFi networks are one of these implicit occupancy sensing systems proposed by several studies due to the availability of the infrastructure in most buildings, especially offices and educational buildings, as well as the popularity of WiFi-connected devices between occupants of such buildings.

1.2 Problem Definition

Although several previous studies investigated using WiFi connection count data as a reliable proxy for occupancy estimation, there are still challenges with employing WiFi data for optimizing the operation of buildings systems. Two of these challenges are addressed in this study. First, although the strong correlation between occupancy counts and WiFi connection counts was confirmed by several studies (Hobson et al., 2019; Mohottige et al., 2018; Ouf et al., 2017), WiFi connection counts have always shown a deviation from the actual number of occupants due to stationary devices and occupants carrying more or less than one WiFi-connected device. Therefore, some studies employed different methods to quantify this deviation (Mohottige et al., 2018; Y. Wang & Shao, 2018) and develop models to estimate real-time occupancy (Ouf et al., 2017; Jagadeesh Simma et al., 2019; Ashouri et al., 2019; Z. Wang et al., 2019; Hobson et al., 2020) or predict/forecast the number of occupants in the future (W. Wang et al., 2017; Ashouri et al., 2019; Hobson et al., 2020; Apostolo et al., 2021) using WiFi connection counts. However, due to the challenges of obtaining ground-truth data, i.e., the cost, inaccuracies, and the intermittent and time-consuming nature, they mostly relied on short-term manual counting of no more than one week at a room- or zone-level (Ouf et al., 2017; W. Wang et al., 2017; Jagadeesh Simma et al., 2019). This may not fully capture the dynamism in building occupancy and the underlying correlation patterns between WiFi connection and occupancy counts. Moreover, those studies focusing on building-level collected data from office buildings in which occupancy has less variation (Ashouri et al., 2019; Z. Wang et al., 2019) or from an academic office building with a maximum occupancy of around 600 people (Hobson et al., 2020).

Second, at the core of these studies, automated data exchange between WiFi networks and Building Automation Systems (BAS) is assumed as the path forward for utilizing WiFi connection counts data to optimize the HVAC system's operations. However, since WiFi networks and the BAS are typically managed by two different teams within most organizations, the widespread adoption of this approach in action has been hindered due to practical and liability issues, including data exchange, communication, and operators' reluctance to automation (Park et al., 2019). Furthermore, potential issues due to sudden WiFi interruptions or temporary system shutdowns may have significant negative consequences for the operation of HVAC systems and compromise occupants' health and safety (e.g., due to under ventilation).

1.3 Research Objectives

Following the above problems, the goal of this research is to improve the approaches used to leverage WiFi connection count data in optimizing the operation of building systems. To achieve this goal, two objectives are defined. These objectives along with the tasks to fulfill them are explained as follows:

1. Validating the correlation between WiFi connection counts and actual building occupancy counts using continuous ground-truth data, collected from camera-based occupancy counters;
   1.1. identifying the temporal variations in the relationship between WiFi connection counts and occupancy counts;
   1.2. identifying the influential features on changes in this relationship (which here is referred to, as WiFi-occupancy conversion factor or 'conversion factor' for short);
   1.3. investigating the effectiveness of the identified features through developing models for real-time occupancy estimation;
   1.4. forecasting day-ahead occupancy counts using WiFi connection counts.
2. Extracting key occupancy indicators relevant to HVAC operation offline, while using WiFi connection count data in existing buildings.
   2.1. learning and predicting weekly occupancy patterns;
   2.2. predicting peak occupancy as well as identifying its occurrence at the building-level;
   2.3. identifying earliest/latest arrival and departure times at the zone-level.

In order to achieve defined objectives, several machine learning algorithms and statistical analysis methods were used on WiFi connection count and camera-based occupancy count data. This study helps buildings' system operators with using WiFi connection count data to obtain practical information regarding the building's unique occupancy patterns. This information helps them in monitoring occupancy in real-time and modifying the sequences of operation of building systems in the day-ahead as well as routinely adjusting the sequences of operations of building systems and their scheduling offline.

1.4 Thesis Organization

This thesis is organized as follows:

CHAPTER 1: Introduction: In this chapter, firstly, an overview of the background and the significance of the problem are presented. Then, the problem statements, the research objectives, and the thesis organization are introduced.

CHAPTER 2: Literature Review: In this chapter, firstly, different resolutions of occupancy sensing, as well as technologies and models being used for this purpose, are reviewed. Then, previous studies using WiFi connection count for real-time occupancy estimation, occupancy pattern identification, and future occupancy prediction are reviewed.

CHAPTER 3: Research Methodology: In this chapter, an overview of the research methodology containing two main modules (i.e., created for developing two frameworks for achieving each of the main objectives) along with the detailed components of them are presented.

CHAPTER 4: Validating the Correlation between WiFi Connection Count and Occupancy Counts: In this chapter, the case study used for validating the first framework proposed in the methodology is introduced. Then, the results of applying different components of the module, including the analysis on conversion factors of WiFi connection counts to occupancy counts, and the prediction models to estimate real-time occupancy counts or to predict/forecast day-ahead occupancy counts, in the case study are presented.

CHAPTER 5: Extracting Key Occupancy Indicators from WiFi Connection Count Data: In this chapter, the case study used for validating the second framework proposed in the methodology is introduced. Then, the results of applying different components of the module, including analysis on occupancy patterns, the prediction models to predict/forecast week-ahead occupancy patterns, and analysis on peak occupancy and arrival/departure times, in the case study are presented.

CHAPTER 6: Summary, Conclusion, and Future Works: In this chapter, a conclusion of the research is provided by explaining the study contributions and limitations as well as proposing the future works.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

Previous studies focused on occupancy sensing from different perspectives. According to the objectives and scope of the present study, this chapter aims to firstly, introduce the occupancy information resolutions proposed in previous studies. Secondly, it discusses the pros and cons of different technologies used for occupancy counting. Thirdly, it reviews the models being used for occupancy counting (i.e., real-time estimation and future forecasting/prediction). Finally, it reviews the studies focusing on occupancy pattern identification, real-time occupancy count estimation, and future occupancy count prediction using WiFi connection counts.

2.2 Occupancy Sensing

The occupancy information can be obtained at different resolutions and different levels of detail. Teixeira, Thiago et al. (2010) categorized the information regarding the position and history of occupants as Spatio-temporal properties and defined five levels for that, including presence (i.e., indicating whether space is occupied or vacant), count (i.e., providing the number of occupants occupying a space), location (i.e., referring to the spatial coordinates of the occupant), track (i.e., indicating the movement of the occupant), and identity (i.e., identifying each of the occupants). Melfi et al. (2011) defined three resolutions for occupancy information including, temporal, spatial, and occupant knowledge. They defined occupant knowledge in four levels of occupancy (i.e., presence, count, identity, and activity). The spatial and temporal resolutions refer to the space structure (i.e., room, zone, floor, building, etc.) and the time spans (i.e., second, minute, hour, day, etc.). W. Wang, Chen, & Hong, (2018) used the figure presented by Melfi et al. (2011) and combined these definitions in a figure which is presented in Figure 2-1.

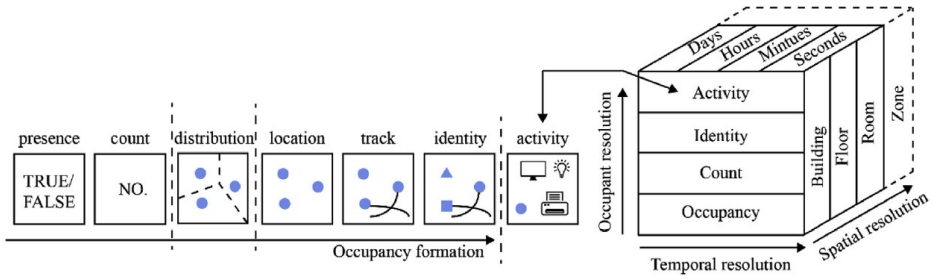


Figure 2-1. Occupancy resolution, including occupancy distribution levels (W. Wang, Chen, & Hong, 2018).

Beside different levels of details defined for occupancy information resolution, there are also other occupancy indicators that are practical in adjusting the schedules of buildings' system operation. Some of these occupancy indicators proposed by previous studies are earliest arrival time, latest departure time, latest arrival time, duration of presence and absence, peak occupancy, etc. (Z. Chen et al., 2016; Peng et al., 2017, 2018; S. Chen et al., 2021; Gunay et al., 2021). In a recent study by Gunay et al. (2021) the application of these indicators is discussed.

Obtaining occupancy information and indicators at different levels of detail is accompanied by different levels of computational complexity, privacy concerns, and cost. Therefore, it is important to obtain this information at the level which is applicable for the desired purpose. Previous studies focused on obtaining accurate occupancy information at different temporal and spatial resolutions using different technologies and methods to optimize the operation of HVAC systems and to maintain acceptable thermal comfort in buildings. Since this study concentrates on obtaining the occupancy counts for the purpose of optimizing the operation of building systems, the previous studies with a focus on occupancy counting were mostly reviewed. Therefore, in the following sub-sections, details on different technologies and methods adopted in previous studies to obtain occupancy counts are discussed.

2.2.1 Sensing Technologies for Occupancy Counting

Currently, the level of CO_2 concentration is one of the most commonly used indicators of occupancy levels in HVAC operations. Accordingly, many studies have focused on improving the accuracy of occupancy estimation using CO_2 sensors (Nassif, 2012; Ansanay-Alex, 2013; Gruber et al., 2014; Jiang et al., 2016; Arief-Ang et al., 2018; Jiang et al., 2020; Franco & Leccese, 2020). However, CO_2 concentration can be misleading since it is sensitive to situational factors including the location of sensors, occupants' activity, occupancy density, and open/closed state of the doors and windows. Moreover, there is always a delay between the change in the occupancy count in space and the variation in the level of CO_2. Other environmental sensors, measuring temperature, humidity, and pressure, can improve the accuracy of occupancy estimation, when combined to form a sensor fusion approach (Masood et al., 2015; Z. Chen, Zhu, et al., 2017; Szczurek et al., 2017; Weekly et al., 2018; Jin et al., 2018). However, it is still challenging to use environmental sensors for a high level of occupancy (Jiang et al., 2016). The most important step in using

environmental sensors is feature engineering since they are an indirect proxy for occupancy (Z. Chen et al., 2016; Ekwevugbe et al., 2017; Masood et al., 2017; Z. Chen et al., 2018).

Using cameras is another technology that measures occupancy counts more directly, through image and/or video processing (Liu et al., 2013; F. Wang et al., 2017; Meng et al., 2020; Chidurala & Li, 2021; Seghezzi et al., 2021). The cameras are also used for obtaining occupancy information at other occupancy resolutions such as presence detection, localization, tracking, and identification based on their types (i.e., depth or non-depth) and their installed locations (i.e., entrance overhead or room interior) (Sun et al., 2020). Compared to other technologies that are more applicable for occupancy counting at a small scale (i.e., rooms with low occupancy rate), several studies used some types of cameras for real-time crowd counting at a large scale (i.e., large areas with a dense occupancy) in complex scenes (Meng et al., 2020). In general, cameras can be used to estimate occupancy with comparatively high accuracy. However, the major issues associated with using cameras for occupancy estimation are high computational complexity and privacy concerns (Z. Chen et al., 2018). Hence, they are mostly used as a source of providing ground truth data for validating results from other technologies (Petersen et al., 2016).

Using motion sensors commonly PIR sensors is another direct approach, mostly applicable to detect occupants' presence (Shetty et al., 2017; Wu & Wang, 2019). However, there are also studies using PIR sensors in more complex set-ups to detect moving directions of occupants (Wahl et al., 2012) or identify their motion patterns (Raykov et al., 2016) and finally infer occupancy counts. An important consideration in using motion sensors is professional tuning and commissioning (i.e., changing the positions or angle of sensors, and tuning the sensitivity of the sensor), otherwise they may not achieve their highest claimed performance (Guo et al., 2010; Salimi & Hammad, 2019).

Two other common technologies proposed in previous studies are Radio Frequency Identification (RFID) tags (N. Li et al., 2012; H.-T. Wang et al., 2014) and Bluetooth Low Energy (BLE) beacons (Longo et al., 2019; Salimi et al., 2019; Tekler et al., 2020). Despite the high level of accuracy, both technologies have deployment limitations. They both need users to carry tags (or other forms of wearables), while the BLE can take advantage of the Bluetooth on users' devices (such as smartphones or smartwatches); the users must always keep the Bluetooth of their devices

on, which is not necessarily common. Therefore, they are not suitable, particularly for occupancy estimation in large areas with a high level of occupancy variation, dynamic and ad-hoc occupants, such as institutional or academic buildings. In general, their main application is for obtaining higher resolution of occupancy information including localization, tracking, and detecting the occupants' identity or activity in office buildings that have known occupants. Therefore, using them in large buildings with multi types of visitors such as institutional buildings might increase complexity and rise privacy issues (Naylor et al., 2018).

In general, all these technologies require installing sensors and an additional infrastructure, allocated to the occupancy counting, which will require extra installation and maintenance costs.

2.2.2 Occupancy Prediction Models

Previous studies utilized different types of methods to obtain occupancy count information with three main objectives, including (i) occupancy/occupants' patterns identification; (ii) real-time occupancy estimation; and (iii) future occupancy prediction/forecasting; (Dai et al., 2020). These studies commonly availed stochastic and machine learning methods while using data collected from different technologies.

In recent years, the application of machine learning algorithms for extracting occupancy information has increased (Dai et al., 2020). Artificial (Recurrent) Neural Networks (ANN/RNN) and Support Vector Machines (SVM) are recognized as the most frequently used data-driven methods for occupancy estimation (Dai et al., 2020; Rueda et al., 2020). Dai et al. (2020) also named other machine learning algorithms that were commonly used in different studies, including Decision Tree (DT), K-Nearest Neighbor (KNN), Random Forest (RF), Clustering techniques, etc. Among these algorithms, clustering was mostly used for learning and understanding occupancy patterns and then combined with DT for estimating or predicting occupancy patterns/counts. For example, D'Oca & Hong (2015) used DT to predict occupancy schedule based on four inputs (i.e., time of the day, day of the week, season, window change) and k-means clustering to identify occupancy profiles which were used as the input to building energy modeling programs. They recognized four working patterns varying between different days of the week from 16 single occupancy offices based on two years of historical data. Liang et al. (2016) proposed almost a similar approach where they used k-means clustering to learn occupancy profiles of an office

building with a maximum of 200 occupants and then used a decision tree to predict occupancy patterns based on two inputs (i.e., day of the week, season). Capozzoli et al. (2017) also utilized k-means clustering and decision tree to learn occupancy profiles and finally optimized the HVAC start/stop schedule in an office building. In another study, Ding et al. (2021) collected data from cameras as well as questionnaires for two weeks from three different buildings of campus including dormitory, offices, and classroom and clustered occupancy presence profiles using k-means clustering. They obtained various patterns for the same building types.

Many studies used machine learning algorithms for occupancy estimation/prediction while utilizing environmental data or taking the advantage of sensor fusion techniques with other data streams to increase the accuracy of developed models. Z. Yang et al. (2014) evaluated the performance of six different models, including SVM, KNN, ANN, Naive Bayesian (NB), Tree Augmented Naive Bayes Network (TAN), and DT in detecting binary occupancy as well as estimating occupancy counts. They assessed these models in single- and multi-occupancy offices while using different ambient sensor variables including light level, binary motion, CO_2 concentration, temperature, humidity, binary infrared, and door status (open/closed). They reported DT as the model with the best performance of all with an accuracy of around 97% in multi-occupancy offices with a maximum of 10 occupants and identified CO_2 concentration as the significant variable in these models. Amayri et al. (2016) also combined data collected from multiple sensors including motion detection, power consumption, CO_2 concentration, microphone, or window/door position detector. Based on the information gain values, they selected motion detector, acoustic pressure (microphone), and power consumption as the most informative features to develop DT based on. In a graduate office with around 4 occupants, they achieved an accuracy of 88% and 86% with DT and RF, respectively.

Z. Chen et al. (2016) used environmental sensors (i.e., CO_2 concentration, air temperature, relative humidity, and air pressure) to estimate occupancy levels in offices. They first selected the best features using an Extreme Learning Machine (ELM)-based wrapper method and then applied different models including ELM, SVM, ANN, KNN, Linear Discriminant Analysis (LDA), and Classification and Regression Tree (CART) for occupancy level estimation. In a graduate office with a maximum of 20 occupants, they reached an accuracy of about 70-74%. Jiang et al. (2016) improved the performance of ELM by adding a feature layer and called it the Feature Scaled

Extreme Learning Machine (FS-ELM) algorithm which they used for real-time occupancy estimation using CO_2 concentration. In an office with a maximum of 35 occupants, they reached the accuracy of 94% when up to four occupants error was allowed. Z. Chen, Zhao, et al. (2017) suggested another approach to automatically select significant features while using environmental sensors. They proposed Convolutional Deep Bidirectional Long Short-Term Memory (CDBLSTM) containing a convolutional network and a deep structure. In a similar testbed as the previous study, they reached an accuracy of 76%.

Zuraimi et al. (2017) developed ANN and SVM models to estimate occupancy from CO_2 concentration level data. In a theater hall with a maximum occupancy of 200 people, they reached an average accuracy of 70% and 76%, respectively. Kim et al. (2019) proposed a framework using long-term measured data to investigate the relationship between accuracy and seasonal changes. They evaluated the occupancy estimation performance of three different models including SVM, ANN, DT, while using a different combination of input data including indoor temperature and luminance, CO_2 concentration, the electricity consumption of lighting, HVAC, and electric appliances. Through a seven-month experiment in a single-occupancy office, they concluded that at different times of the year, different input variables might result in better estimation performance. Arief-Ang et al. (2018) proposed a semi-supervised domain adaptation approach, in which they trained a model to predict occupancy counts from CO_2 concentration in a small room and then adapted it with multiple different locations. In their experiments, they could reach an accuracy of about 60%.

Beside machine learning algorithms, different types of Markov models were also popular in these studies (W. Wang et al., 2017). These types of models are commonly used for binary occupancy prediction and occupants' activity prediction (Salimi & Hammad, 2019). However, there are studies focusing on predicting the number of occupants using different types of Markov models. A Markov model is constructed based on the assumption that the future state depends on the current state and it is contained of states and transitions between these states (Wohlin et al., 2003). Different studies utilized different approaches for defining the states. In an early study by Page et al. (2008), binary states of presence and absence for each occupant within a zone (i.e., a residential unit, a single or multiple person offices) were considered for developing an inhomogeneous Markov model. They suggested calculating the total number of occupants in that

zone by multiplying the obtained occupant's pattern by the total number of occupants or accumulating all obtained occupants' patterns. The methodology proposed in this study was used or considered as the basis in many later studies (Mahdavi & Tahmasebi, 2015; Z. Li & Dong, 2018). Erickson et al. (2011) developed an inhomogeneous Markov chain and defined the states based on all observable occupancy numbers in each zone considering the maximum occupancy as the limit for the states. This approach gets exponentially complicated as the maximum number of occupants in a zone increases or more zones will be added to the calculations. Other studies used the same approach for defining the states in small-scale experiments, as well (Z. Chen & Soh, 2017). Compared to these studies, Chen et al. (2015) proposed an inhomogeneous Markov chain with states that are independent of the maximum number of occupants in the zone. They considered three states of "1", "-1", and "0" representing an occupant entering, leaving a zone and not moving, respectively. These states were considered based on the assumption that during a short interval, only one occupant can enter or leave the zone. They explained that the number of occupants moving during a specific interval can be extended to more than one which result in matrices with higher dimensions. W. Wang, Chen, Hong, et al. (2018) proposed a Markov based Feedback Recurrent Neural Network (M-FRNN) algorithm while they considered two simple states of "In" and "Out" for every identified occupant. In addition, different transitional matrices were created based on different times of the day.

Salimi et al. (2019) reviewed and compared recent studies that used different types of Markov chain models for occupancy modeling. They also developed an inhomogeneous Markov chain prediction model based on real occupancy data to obtain detailed information about occupancy. In this study occupants' activities were identified and referred to as work states. Based on the transitions between these work states, the occupants' profiles were predicted which were then used to calculate the number of occupants in each zone. In an experiment conducted on an office, they reached the R^2 value of 0.68 for 30-min-ahead occupancy prediction while the real occupancy data was collected using BLE.

There are studies that compared different machine learning algorithms with Markov models. For example, Z. Chen & Soh (2017) compared the performance of various models including inhomogeneous Markov chain (states being the number of occupants based on possible maximum occupancy counts), multivariate Gaussian, Autoregressive Integrated Moving Average (ARIMA),

ANN, Support Vector Regression (SVR) which were developed to predict regular occupancy level at different prediction horizons, including, 15-min, 30-min, 1-h, and 2-h. Based on an experiment conducted on a graduate office with a maximum of 8 occupants, ARIMA outperformed in short-horizon predictions (i.e., 15 min and 30 min) while in long-horizon predictions, SVR showed a higher performance. For these experiments, data was collected from video cameras.

Z. Li & Dong (2018), proposed a moving-window inhomogeneous Markov model to predict occupancy in commercial buildings. They compared the performance of the proposed model in predicting occupancy at different prediction horizons, including 15-min, 24-h with ANN, SVR, and two other Markov models proposed by previous studies. They showed the proposed model outperformed in short-term horizons (e.g., 15-min) with Root Mean Squared Error (RMSE) of 0.510, achieved in an experiment with a maximum of 6 occupants while had a significantly lower performance for a 24-h-ahead horizon.

The Markov models developed in these studies for occupancy count prediction were almost limited to a few numbers of occupants since the transitional matrices get exponentially complicated when the number of occupants increases. In some approaches, they needed access to the occupants' identity to develop a model specifically for each occupant which is mostly applicable in offices when there are known occupants. Therefore, in a large-scale application with a high occupancy variation, implementing these models would not be reasonable. In addition, they were mostly successful in predicting occupancy in short horizons (i.e., less than 1-h-ahead).

In the following sub-sections, the studies that developed models to identify occupancy patterns, estimate real-time occupancy count, or predict future occupancy count, while using WiFi data, are reviewed.

2.3 Occupancy Counting Using WiFi Data

WiFi networks are widely installed in modern buildings, especially in offices and educational buildings. Furthermore, using WiFi-enabled devices, including smartphones, tablets, and laptops, is very common among occupants of such buildings. These factors make WiFi connection count data a cost-efficient and reliable approach to obtain building occupancy information non-intrusively, and use it for different purposes, including HVAC operation optimization. For example, Balaji et al. (2013) used WiFi data for HVAC operation in a commercial building and

reached an electricity saving of 17.8%. W. Wang, Wang, et al., (2018) also used WiFi probe technology to estimate real-time occupancy for an occupancy-based ventilation strategy. They compared the energy consumption share of ventilation in their proposed strategy, to the fixed-rate ventilation, in an experiment on a graduate students' office. The proposed strategy showed a saving of about 44.26% and 55.5% on weekdays and weekends, respectively. In another study on a commercial building, almost 26.4% of energy consumption in cooling and ventilation demands was saved through inferring occupancy information from WiFi connections (W. Wang, Hong, Li, et al., 2019).

A statistically significant strong positive correlation between WiFi connection counts and actual occupancy counts has been confirmed by previous studies (Mohottige et al., 2018; Ouf et al., 2017). For example, Ouf et al. (2017) showed a correlation coefficient (R) of 0.84 between the two, which was stronger than the relationship of CO_2 concentration level and occupancy counts (Ouf et al., 2017). In another study, a stronger correlation was found between occupancy counts and WiFi connection counts (with an R of 0.84) compared to beam counters' log (Mohottige et al., 2018). Although these experiments were conducted at a room-level, they successfully identified the potential of using WiFi connection count data as a proxy to estimate occupancy counts at the building-level. These studies revealed that WiFi connection counts can be a strong representative of the occupancy pattern. Following this strong correlation, there are some recent studies that directly used WiFi connection count instead of occupancy counts for different purposes including space utilization (Oppermann & Munzner, 2020), emergency evacuation (Pasquel Mohottige et al., 2020), and evaluating building energy performance (Rafsanjani & Ghahramani, 2019; Hou et al., 2020; Chong et al., 2021).

Besides WiFi connection logs and WiFi connection counts, there are studies that benefited from other aspects of WiFi technologies, such as Received Signal Strength Indicator (RSSI) to estimate the position of occupants. In these studies, some included the occupancy counting as another occupancy resolution. For example, Zou et al. (2017) proposed a system for providing fine-grained occupancy information at different resolutions including occupancy detection, counting, and tracking. In an experiment conducted on an office with almost 20 occupants, their system obtained a Normalized Root Mean Square Deviation (NRMSD) of 0.096 in estimating the number of occupants. Yoo et al. (2020) proposed a station-oriented indoor localization system. In

an experiment on multiple offices with a maximum of 12 occupants, they obtained the overall Normalized Root Mean Square Error (NRMSE) of 0.0309 in estimating occupancy counts. Although using RSSI for occupancy counting at the building-level increases the complexity, it might be a better option than using the WiFi connection counts for estimating the number of occupants inside a room or zone. Since the connections that are out of the desired boundary can be filtered out based on a threshold defined for RSSI (Longo et al., 2019; Ravi & Misra, 2021). Longo et al. (2019) took this approach to estimate the number of occupants and achieved an RMSE ranging between 1.42 to 5.12 while experimenting on multiple academic spaces with a maximum of 132 occupants. However, this approach might not be a reasonable candidate for occupancy estimation at a large scale since it might invade occupants' privacy due to the need for Media Access Control (MAC) IDs. Moreover, the environment including building components can affect the signal strength.

On the other hand, Channel State Information (CSI), another potential of WiFi technology, describes the details of WiFi signal propagation from the transmitter to the receiver. Studies using CSI measured the impact of a certain number of occupants on signal propagation and estimated occupancy counts through classification techniques (Sobron et al., 2018). Studies that used CSI for occupancy counting could achieve an accuracy ranging between 81% to 96% (Di Domenico et al., 2016; Zou et al., 2018). However, this approach is mostly applicable in small spaces with a limited number of occupants since it is highly dependent on the environment and movements.

2.3.1 Real-time Occupancy Estimation Using WiFi Connection Count Data

The majority of studies with a focus on WiFi connection count as a proxy for occupancy count developed a model to estimate real-time occupancy counts. These studies mostly proposed a Linear Regression model using a short period of ground truth data to interpret the WiFi connection counts to occupancy counts. For example, Ouf et al. (2017) developed a Linear Regression model with one-week ground-truth data collected from a classroom. They achieved a coefficient of determination (R^2) of 0.703. (Jagadeesh Simma et al., 2019) introduced Multiple Linear Regression (MLR) models that were developed using six weeks of ground-truth data collected from a lecture room. The R^2 value for these models ranged between 0.90 to 0.96.

Some studies utilized these Linear Regression models developed with short-term ground-truth data to produce a time-series of occupancy counts which was used as ground-truth data for developing semi-supervised models that predict future occupancy. For example, in a study by Ashouri et al. (2019), a Linear Regression model was developed with an R^2 value of 0.90 while 19 hours of ground-truth data were collected from an office building. Before developing the model, 27 stationary devices were recognized during the night and deducted from WiFi connection counts. Hobson et al. (2020) also trained a Linear Regression model with eight-and-a-half weekdays of ground-truth data collected from an academic office building and reached an R^2 value of 0.85. The experiments were mainly conducted on two types of buildings, i.e., offices and academic buildings, which are summarized in Table 2-1. However, even in spaces of the same type, the coefficients of WiFi connection count were different, suggesting that this factor might be affected by the space type as well as other elements such as time and level of occupancy.

Table 2-1. Summary of studies with Linear Regression models developed for WiFi-based occupancy counting

Reference	Spatial Resolution	Case	Ground-truth Duration	Linear Regression Model[1]	Performance
(Jagadeesh Simma et al., 2019)**	Room-level	Lecture room (max 100 occupants)	6 weeks	Y = 1.01X + 0.54 Y = 1.04X - 0.55 Y = 1.19X + 0.61 Y = 1.20X - 0.64 Y = 1.12X - 0.18 Y = 1.33X - 0.52	R^2: 0.963 R^2: 0.958 R^2: 0.940 R^2: 0.916 R^2: 0.910 R^2: 0.905
(Ashouri et al., 2019)	Building-level	Office (max 80 occupants)	19 hours	Bias = 27 Y = 1.27X	R^2: 0.900
(Hobson et al., 2020)[2]	Building-level	Academic office (max 570 occupants)	~ 9 weekdays	Y = 0.83X	R^2: 0.845
(Ouf et al., 2017)	Room-level	Classroom (max 80 occupants)	1 week	Y = 0.79X + 1.4	R^2: 0.703

(1) In these models, the predictor feature (X) is the WiFi connection count and the target (Y) is the occupancy count.

(2) To report all models in the same format, the predictor and target are rearranged in the models reported from the reference (Jagadeesh Simma et al., 2019) and (Hobson et al., 2020).

Other studies focused on identifying the conversion factor without developing models. For example, Y. Wang & Shao (2018) collected 14 hours of ground-truth data from a typical study room. They reached the value of 1.16 by calculating the average of the quotient of WiFi connection counts and occupancy counts. Mohottige et al. (2018) took the same approach and reported this value to be 1.3. Their experiment was conducted on 37 samples collected from 4 classrooms. The number of stationary devices was also detected in studies that had access to devices' MAC address and connections history (W. Wang et al., 2017; Mohottige et al., 2018; Z. Wang et al., 2019), which is most of the time not feasible in practice, due to privacy concerns.

In addition to the study focusing on the total WiFi connection counts as the only feature to estimate occupancy counts based on, Z. Wang et al. (2019) split the total WiFi connection counts at each interval into eight categories based on the daily connection duration of each WiFi-connected device. Additionally, the hour of the day, the day of the week, and the day type (holiday or not) were included as the features in developing three models (Random forest, Deep learning neural network, and Long Term Short Term Memory networks (LSTMs) using five weeks of ground-truth data collected from two floors of an office building. They recognized the feature including the number of devices that were connected for 8 to 12 hours per day as the most important feature. By differentiating the types of devices based on their daily connection duration, they achieved a higher accuracy (i.e., the errors are within two people counts for more than 70% of estimations) compared to similar studies. Unlike most of the studies that used manual counting for collecting ground truth data, this study used camera-based sensors. However, this study was conducted in an office building with a peak occupancy of 48–74 occupants.

The main focus of WiFi-based occupancy sensing studies has been on real-time occupancy estimation. Among them, many have also compared WiFi data with other data streams, using different machine learning algorithms. W. Wang, Chen, & Hong (2018) employed ANN, KNN, and SVM to estimate occupancy counts using three datasets: an environmental dataset, a WiFi signal dataset, and a fused dataset of both. They showed that SVM and KNN trained on the WiFi data alone, provide fewer counting errors. Hobson et al. (2019) developed ANN and MLR models using data from WiFi APs, CO_2 sensors, PIR motion detectors, and plug and light electricity load meters, to estimate the occupancy counts. The developed models using sensor fusion containing WiFi data showed significantly higher accuracy than other datasets. The maximum and mean R^2

values resulted from models developed using WiFi data alone were 0.97 and 0.71, respectively in ANN models and 0.96 and 0.74, in MLR models. W. Wang, Hong, Xu, et al. (2019) used the adaptive lasso and ANN models to find the best feature set among twelve environmental features and WiFi dataset for real-time occupancy count estimation as well as four-level (i.e., zero, low, medium, high) occupancy estimation. In an experiment, they could achieve the best results (i.e., mean absolute error of 2.18 for real-time estimation and F1_accuracy of 84.36% for occupancy level estimation) with the fusion of three features, including indoor air temperature, CO_2 concentration, and WiFi dataset.

2.3.2 Future Occupancy Prediction Using WiFi Connection Count Data

Although adapting HVAC systems control to real-time occupancy counts can provide significant energy savings, for decreasing the response time and optimal operation of these systems, there is a need to acquire occupancy information ahead of time. Future occupancy prediction can be employed for proactive control of building systems' operation based on the actual occupancy patterns of a building. In addition, through this approach, potential technical faults, e.g., WiFi temporary downtime or sudden interruptions can be mitigated. However, due to the lack of access to long-term ground truth data for training the prediction models, future occupancy prediction using WiFi connection count data is rarely investigated.

As mentioned before, some studies have overcome this challenge by developing semi-supervised models by producing the required ground-truth data from the models developed for real-time occupancy estimation. For example, Ashouri et al. (2019) predicted day-ahead occupancy counts in two steps. They firstly converted WiFi connection counts to occupancy counts through a Linear regression model; and then used it to predict the day-ahead occupancy counts while using the 9-week synthetic ground-truth data. They used MLR and ANN for the second step (i.e., developing the day-ahead prediction models) and achieved R^2 values of 0.88 and 0.96, respectively, indicating a superior performance for ANN. However, the authors proposed MLR as a more reasonable candidate for future occupancy prediction due to its lower computational complexity. Hobson et al. (2020) also predicted day-ahead occupancy day type using 7-month synthetic ground-truth data (i.e., data generated using prediction models developed for estimating occupancy counts). During an experiment on an academic office building, they fed the lighting

and plug load profiles (which follow the same patterns of WiFi data in 84.5% of days) into a classification tree and reached a successful classification rate of 70.4%.

Considering the strong correlation between WiFi connection count and occupancy counts, some studies developed models to predict WiFi connection counts without collecting ground truth data. Apostolo et al. (2021) used WiFi connection data at Access Point (AP)-level in order to predict the Wi-Fi network demand for energy-efficient smart buildings. They developed multiple classification and regression models based on a combination of different parameters such as features, machine learning algorithms, etc. to find the best model. In an experiment on a classroom building, they reached the highest performance, in terms of the lowest Root Mean Squared Percentage Error (RMSPE), Mean Absolute Percentage Error (MAPE), and RMSE for connection count prediction, with the values of 0.29, 0.41, and 8.41, respectively.

Another approach typically being proposed for future occupancy prediction is the Markov models which do not require a long duration of ground-truth data. W. Wang et al. (2017) used WiFi probes to predict occupancy in the following time window based on previous time windows, through a Dynamic Markov Time-Window Inference (DMTWI) model. They conducted an experiment with a time window length of 20 minutes on a research student office room with less than 20 occupants. They achieved a prediction accuracy of 80% with a tolerance of four (4) occupants on weekdays; three (3) on holidays, and two (2) on weekends. Their proposed model was also compared versus Auto-Regressive Moving Average (ARMA) model and SVR, and showed slightly higher accuracy. A summary of major research works focusing on WiFi-based occupancy counting is presented in Table 2-2.

Table 2-2. Summary of major WiFi-based occupancy counting studies.

Problem	Reference	Data	Spatial Resolution	Case	Experiment Duration	Method (Technique)	Performance
Occupancy estimation using WiFi data	(Di Domenico et al., 2016)	CSI	Room-level	Office room (max 7 occupants)	Not mentioned	Linear discriminant classifier	2-Accuracy: 81-91%
	(Zou et al., 2017)	RSSI	Zone/Room level	Multi-functional office and lab (max 20 occupants)	4 weeks	WiFi-based non-intrusive Occupancy Sensing System (WinOSS)	NRMSD: 0.096 Mean error: 0.196 SD[1] error: 0.578
	(Ouf et al., 2017)	Device counts, CO_2	Room-level	Classroom (max 80 occupants)	1 week	MLR; Pearson's product-moment correlation	R^2: 0.703 R = 0.839
	(Mohottige et al., 2018)	Device counts	Room-level	Classroom (max 200 occupants)	1 week	Pearson's product-moment correlation	R = 0.85 SMAPE: 12.1%
	(Zou et al., 2018)	CSI	Room-level	Meeting room (max 11 occupants)	2 days	Transfer kernel learning	Classification accuracy: 90.2-96%
	(W. Wang, Chen, & Hong, 2018b)	RSSI	Room-level	Graduate student office (max 20 occupants)	3 days	KNN; SVM; and ANN	MAE: 2.1-2.5 MAPE: 34.3-37 RMSE: 2.8-3.3
	(Longo et al., 2019)	RSSI	Room-level	Multiple academic spaces (max 132 occupants)	25 hours	Regularized linear regression; Regularized multinomial logistic regression	RMSE: 1.42-5.12 MAE: 1.05-4.25 Classification accuracy: 68-95%
	(Azam et al., 2019)	RSSI	Room-level	Open office (max 20 occupants)	9 weeks	DT; RF; Gradient Boosting; Extremely Randomized Trees	Classification accuracy: 95%
	(Z. Wang et al., 2019)	Device counts	Zone-level	Private and cubicle offices (max 74 occupants)	5 weeks	Long Term Short Term Memory (LSTM) networks; RF; ANN	2-Accuracy[2]: 70-72% RMSE: 3.95-4.62
	(Hobson et al., 2019)	Device counts	Floor-level	Academic office (max 72 occupants)	208 hours	MLR; ANN	Max R^2: 0.96-0.97 Mean R^2: 0.71-0.74
	(Jagadeesh Simma et al., 2019)	Device counts	Room-level	Lecture room (max 100 occupants)	6 weeks	Linear regression	R^2: 0.86 - 0.96
	(Ashouri et al., 2019)	Device counts	Building-level	Office (max 80 occupants)	19 hours	Linear model	R^2: 0.9

Table 2-2. Continued

Problem	Reference	Data	Spatial Resolution	Case	Experiment Duration	Method (Technique)	Performance
Occupancy pattern clustering	(Y. Wang & Shao, 2018)	Connection data	Room-level	Study room (max 20 occupants)	14 hours	K-means clustering	Not reported
	(Hobson et al., 2020)	Device counts	Building-level	Academic office (max 570 occupants)	7 months	K-means clustering	Not reported
Future occupancy prediction using WiFi data	(W. Wang et al., 2017)	Connection data	Room-level	Research students' office rooms (max 68 occupants)	3 days	DMTWI; ARMA; SVR	5-Accuracy[2]: 85%
	(Ashouri et al., 2019)	Device counts	Building-level	Office (max 80 occupants)	9 weeks	MLR; ANN	R^2: 0.88-0.96 RMSE: 2.9-5.0 Accuracy[3]: 83.1-90.1%
	(Hobson et al., 2020)	Device counts, plug and light load	Building-level	Academic office (max 570 occupants)	7 months	Decision Tree	Classification rate: 70.4% Error: 47 ± 69 occupants
	(Apostolo et al., 2021)	Connection data	AP-level	A classroom building (max 30 occupants)	6 months	RF; DT; KNN; XGBoost	Min RMSE: 8.4130 Min RMSPE: 0.29 Min MAPE: 0.41

(1) Standard Deviation

(2) X-tolerance accuracy proposed by Jiang et al. (2016) is a metric reporting the percentage of the estimations with errors less than X (i.e., when up to X occupants error is allowed)

(3) $Accuracy = 100 \times (1 - \frac{RMSE}{\bar{O}})$, where $\bar{O}$ is the mean of actual occupancy during the entire experiment. (Ashouri et al., 2019)

2.4 Summary and Conclusion

Although several previous studies investigated using WiFi data as a reliable proxy for occupancy estimation, relatively few studies focused on using WiFi data for extracting occupancy indicators relevant to HVAC operation. Moreover, the models developed for occupancy estimation/prediction have limitations, since they were mostly developed based on short-term ground truth data which ignores the variation of WiFi connection counts deviation from actual occupancy counts over time. These studies mostly focused on occupancy at testbeds with a limited number of occupants or with regular working schedules. The occupancy is less nuanced in these cases compared to the occupancy of a building with multiple functions and a high level of peak occupancy. Furthermore, although the temporal and spatial variation of the relationship between WiFi connection count and actual occupancy count were discussed in many studies, a fixed value was mostly suggested as the conversion factor.

The present study aims to bridge some of the identified gaps in the literature by firstly, leveraging longitudinal ground-truth data to investigate the relationship between WiFi connection counts and occupancy counts; Secondly, extracting practical occupancy indicators from WiFi connection count data, that can be later used by building operators to adjust HVAC systems' schedules. This approach can overcome various challenges related to the automation of data exchange between WiFi networks and BAS.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction

As illustrated in Figure 3-1, the proposed methodology consists of two main modules, including multiple components, to obtain building occupancy information using WiFi connection count data. The first module focused on developing a framework to validate the correlation between WiFi connection counts and the actual building occupancy counts, using a continuous stream of ground-truth data. This module started by, firstly, investigating the relationship between WiFi connection counts and camera-based occupancy counts (as a measure of true occupancy) over a comparatively long period, as well as identifying the features influencing this relationship. The effectiveness of each of these features was then studied and eventually prediction models were developed to estimate real-time occupancy counts (i.e., translate WiFi connection counts to occupancy counts) at the building-level. Secondly, it focused on developing prediction models to forecast day-ahead occupancy counts at the building-level using WiFi connection counts. These studies highlight the important considerations that need to be addressed while using WiFi connection count as a proxy for occupancy counts to optimize buildings' system operation based on real-time occupancy.

The second module focused on developing a framework to extract key occupancy indicators using WiFi connection count data. This module started by, firstly, learning and predicting weekly occupancy patterns at the building-level. Secondly, the analysis focused on predicting peak occupancy and identifying its occurrence interval on different days of the week. Finally, earliest/latest arrival and departure times were identified at a zone-level. Extracting these metrics can provide practical information for building operators, especially for adjusting ventilation schedules based on weekly occupancy patterns and the peak occupancy occurrence, as well as adjusting temperature setpoints and scheduling setbacks for different building zones. It is worth noting that this module focused on identifying indicators that characterize occupancy patterns (e.g., arrival/departure times, or occurrence of peak occupancy) rather than the exact number of occupants by considering that there is a strong correlation between WiFi connection counts and occupancy counts. Accordingly, the times at which the first and last WiFi connections are observed are considered as the earliest arrival and latest departure times of occupants, respectively.

Moreover, the time at which the maximum WiFi connection count is observed is considered as the peak occupancy time.

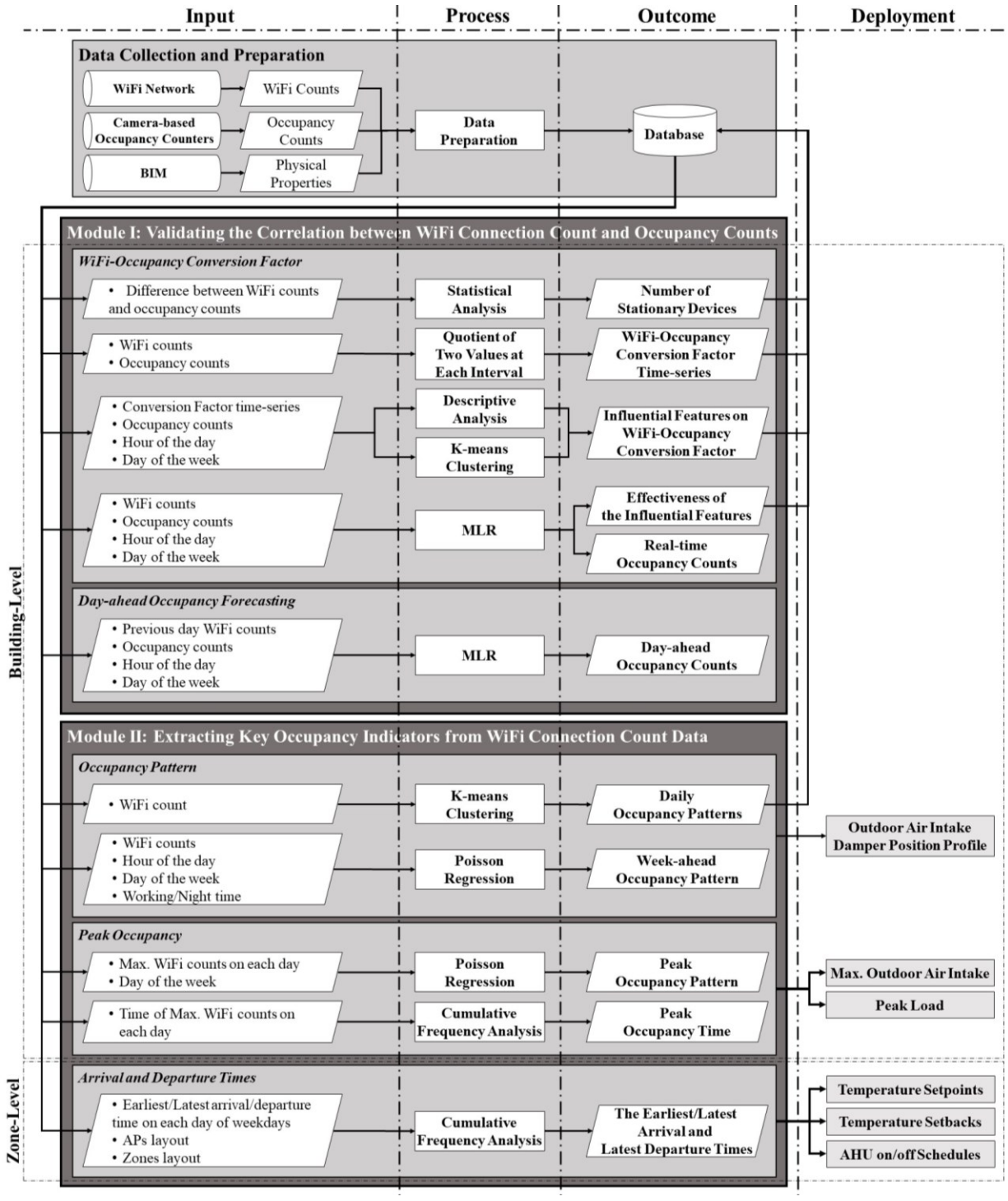


Figure 3-1. The high-level methodology of the research

The components of the methodology, including machine learning algorithms, were implemented with Python libraries including scikit-learn v0.24.2[1] and statsmodels v0.10.2[2].

3.1.1 Scope of the Case Studies

To validate the proposed methodology, modules are implemented in two different institutional buildings of Concordia university campus, in Montreal, Canada, as a proof-of-concept. These buildings include (i) a library building with an occupancy level up to around 2,000, and (ii) an academic building with a WiFi connection level up to around 3,000. They consist of multiple space types, such as classrooms, study rooms, meeting rooms, offices, etc. The required data for this research was collected through the WiFi network administration platform of the University as well as the cameras installed at one of the buildings. The data collection period started from mid-January 2020 and stopped in mid-March 2020 due to the COVID-19 pandemic. More detailed explanations about the case study buildings and data collection and preparation process are presented in the following chapters.

3.1.2 Data Collection and Preparation

Two main sources of data were acquired in this study, i.e., WiFi connection count data-stream, and camera-based occupancy count data-stream. For the former dataset, the total number of devices connected or trying to connect to each of the available Service Set Identifiers (SSIDs) in the building were retrieved from the WiFi network administration platform. Typically, most WiFi network administration platforms can provide hourly or sub-hourly reports on the number of 'authenticated' and 'associated' device counts. While the former refers to the connected devices, the latter provides the total number of devices trying to connect to the network (successfully and unsuccessfully) (Cisco, 2017). The number of 'associated' accounts, which include those devices that have been successfully authenticated as well as those who failed to connect, provides a more realistic view of the actual number of occupants, present in the building. Therefore, for the purpose of this study, the total number of 'associated' accounts was used due to its additional coverage of occupants (e.g., visitors who are not authenticated in the WiFi network). To limit privacy concerns,

[1] https://scikit-learn.org/
[2] https://www.statsmodels.org/

only the aggregated device counts, i.e., the number of 'associated' device counts at each WiFi AP, were collected and no information regarding the devices' MAC addresses was being collected.

In addition to WiFi data, the building's contextual information including physical properties and floor areas of the building zones, as well as the HVAC system's layout were retrieved from the 3D BIM (Building Information Model). By integrating the location of APs in the BIM, APs' coverage can be estimated based on the floor layouts and building components' materials, if available, then it can be attributed to zones served by individual Air Handling Units (AHU).

The ground-truth dataset, i.e., the camera-based occupancy count data, was generated using camera-based counters. These counters were installed at the main entrances/exits of the studied zone/room and use image recognition to detect the number of people (heads) going in and out of a building/zone/room without any facial recognition to avoid privacy issues. Based on the values reported by these counters, the number of occupants inside the building was calculated in each interval. To validate the synchronization of the two data-streams, the normalized daily profiles (i.e., normalized through min-max normalization) of WiFi connection counts and camera-based occupancy counts were clustered. Following the assumption of a strong correlation between their counts, similar patterns and peak times were expected between these two datasets. For this purpose, k-means clustering with Euclidian distance similarity metric was selected and Davies Bouldin Index (DBI) was calculated to determine the optimal number of clusters.

To identify the days that may be considered outliers among daily profiles and investigate the possibility of grouping various days based on the similarity of their occupancy patterns, the Kruskal-Wallis H test was used, which is a non-parametric statistical test used to determine whether statistically significant differences exist between different subgroups. Moreover, for each dataset, the outliers among values of each hour were detected through Interquartile Range (IQR) method which identifies outliers as values that are more than 1.5 IQR (i.e., the difference between third quartile (Q_3) and first quartile (Q_1)) below Q_1 or above Q_3. The hourly outliers were then replaced with the mean or median of values ranging in each hour. Finally, both collected data were cleaned, synchronized, and transformed to a proper format for further analyses.

3.2 Module I: Validating the Correlation between WiFi Connection Count and Occupancy Counts

This module, consisting of multiple components, aims to introduce a framework for validating the correlation between WiFi connection counts and the actual building occupancy counts by leveraging longitudinal ground-truth data. The details of components are discussed in the following sub-sections.

3.2.1 Investigating the Relationship between WiFi Connection Count and Camera-Based Occupancy Count Data

Investigating the relationship between WiFi connection counts and camera-based occupancy count data entailed three steps. The first step consisted of identifying the number of connections that were attributed to stationary devices such as printers, servers, etc., which cannot be representative of occupants. Although the number of these devices can vary throughout the day (due to switching to off or idle modes), the level of dynamism of their behavior is considerably lower than the counts from the occupants' WiFi-connected devices. It was assumed that during the time at which the number of occupants is at its lowest level, the difference between WiFi connection counts and actual occupancy counts is minimized. Therefore, the hour at which the minimum occupancy counts occurred on each day was selected and the difference between WiFi connection counts and occupancy counts at those times were attributed to the number of stationary devices on each day. Due to the slight variations in these values over various days, their minimum was considered as the number of stationary devices. This value was then deducted from WiFi connection counts at all hours throughout the entire time-series.

The next step entailed calculating the conversion factor of WiFi connection counts to occupancy counts. A time-series of conversion factors was produced as the quotient of camera-based occupancy and WiFi connection counts on an hourly basis. The dynamism of these conversion factors throughout the day as well as between different days was then investigated using statistical analysis techniques. Following the variation of the conversion factors, their daily profiles were clustered, using k-means clustering, to study the dominant daily patterns for conversion factors and evaluate the influence of time and occupancy level on these patterns.

The last step entailed investigating the effectiveness of each identified influential feature on the relationship between WiFi connection counts and occupancy counts through extrinsic evaluation, i.e., by developing prediction models. Multiple prediction models were trained based on different combinations of identified features as predictors to estimate the real-time number of occupants. MLR was selected to develop the prediction models, due to (i) the suitability of the model according to the literature; and (ii) the white-box nature of the model, which allows for the interpretation of features' importance. The performance of the developed models was evaluated based on RMSE, MAPE, and R^2. For developing the MLR, the categorical features were converted into numerical features using one-hot encoding, and numerical features were normalized using min-max normalization.

3.2.2 Day-ahead Occupancy Counts Prediction

This step consisted of developing prediction models that forecast day-ahead occupancy using WiFi connection counts. In these models, WiFi connection counts collected on each day were used for forecasting the occupancy counts of the next day. This helped the models to account for the occupancy variation of the next day influenced by the occupancy variation of the present day. Moreover, time-related features, including Hour of the day and Day of the week, were used as inputs to the models to introduce historical patterns of occupancy into the models.

The models were developed using MLR and three metrics, including RMSE, MAPE, and R^2, were selected for assessing the performance improvement of the models while extending the training window. For developing the MLR, the categorical features were converted into numerical features using one-hot encoding, and numerical features were normalized using min-max normalization. The process starts by training the model on one week while forecasting different days of the following week, then the training window was extended to two, three, and more weeks, each time forecasting the immediately following week's occupancy counts. Monitoring the accuracy gain/losses in forecasting the days of the next week, opting for maximum R^2 and minimum errors, will provide the preferred prediction window. To avoid overfitting, the models were evaluated through time-based n-fold cross-validation employed on days-of-the-week that followed similar patterns.

3.3 Module 2: Extracting Key Occupancy Indicators from WiFi Connection Count

This module, consisting of multiple components, aims to introduce a framework for extracting key occupancy indicators using WiFi connection count data. The details of components are discussed in the following sub-sections.

3.3.1 Occupancy Pattern Prediction

Occupancy pattern prediction entailed two steps, i.e., identifying day type groups that follow similar patterns; and developing prediction models for each group. For the first step, 'day type' can be decided extrinsically (e.g., weekdays vs weekends) or intrinsically (i.e., through clustering of observed patterns on different days). For the latter, the centroid-based clustering algorithm (k-means clustering) was selected after evaluating k-shape and k-means clustering techniques. The clustering was applied to normalized daily occupancy profiles (i.e., normalized through min-max normalization) and the similarity between data-points is measured through Euclidian distance. To find the optimal k value in k-means clustering, DBI was used as the mathematical performance measure besides two other factors, namely the number of clusters and the cluster size (i.e., the number of items in each cluster).

The next step entailed developing prediction models, considering the results of clustering daily occupancy patterns to improve the prediction's performance. The features included the values of WiFi connection counts, the Hour of the day, and the Day of the week, all of which were extracted from the collected time-series WiFi connection count data. In addition, comparing the performance of models developed for predicting the entire day, versus models predicting the working time period only (i.e., 8:00 a.m. to 9:00 p.m.) suggested that introducing an additional Boolean feature to distinguish the working time (i.e., 8:00 a.m. to 9:00 p.m.) against nighttime (i.e., before 8:00 a.m. and after 9:00 p.m.), can improve the prediction performance despite its correlation with the Hour of the day feature. All these categorical features were converted into numerical features using one-hot encoding and then used for training prediction models. Evaluating prediction models developed on different datasets showed slightly better performance in predicting typical days when using separate models for specific day types, rather than having one model for all.

To develop the prediction model, Poisson regression, which is a type of Generalized Linear Model (GLM), was selected. Unlike MLR, which is the common modeling technique for occupancy estimation, Poisson regression uses the Maximum Likelihood Estimation (MLE) technique to identify regression coefficients. Poisson regression assumes the response variable follows a Poisson distribution; hence it is ideal for modeling event-based and discrete WiFi connection count data. The performance of the developed models was then evaluated based on the error, using RMSE and MAPE, and Pseudo R-squared (R^2_D) – which is also called 'percentage of deviance'. $R^2_D$is a generalization of regular R^2 in the Ordinary Least Squares (OLS) method and is calculated through the following equation (Heinzl & Mittlböck, 2003):

$$\boldsymbol{R_D^2 = 1 - \frac{\sum_i^n \left[y_i \log\left(\frac{y_i}{\hat{\mu}_i}\right) - (y_i - \hat{\mu}_i) \right]}{\sum_i^n \left[y_i \log\left(\frac{y_i}{\bar{y}}\right) - (y_i - \bar{y}) \right]}}, \quad (1)$$

where $\boldsymbol{y} = (y_1, \ldots, y_n)$ and $\hat{\boldsymbol{\mu}} = (\hat{\mu}_1, \ldots, \hat{\mu}_n)$ are the actual values of the dependent variable and the corresponding predicted values, respectively; and

$$\bar{y} = \frac{\sum_{i=1}^n y_i}{n} \quad (2)$$

3.3.2 Peak Occupancy Analysis

Peak occupancy was investigated in two levels including the maximum WiFi connection counts and the time of its occurrence. Firstly, the maximum WiFi connection counts on each day and day of the week were identified as features for training prediction models through Poisson regression. R^2_D was also used as the measure to evaluate the impact of extending the training period, on improving the performance of peak occupancy prediction. The process started by predicting the peak occupancy of each week from the previous week's data, then introducing data from additional past weeks until reaching an acceptable accuracy of prediction. Through this process, the R^2_Dvalue was calculated by averaging the performance of prediction models trained with the same number of training weeks.

Secondly, to investigate the occurrence time of peak occupancy, cumulative relative frequency distributions of the time when the maximum WiFi connection counts occurred on each day were used. These plots can identify the probability of peak occupancy occurrence at different times of the day, for each day of the week. The identified occupancy profile on different days, as well as the most likely time of peak occupancy occurrence, can then be used by building operators to adjust ventilation schedules accordingly to mimic these profiles.

3.3.3 Arrival and Departure Times Analysis

Arrival/departure times analysis needed zone-level WiFi connection counts which firstly entailed mapping APs to AHUs' zones; and secondly involved aggregating WiFi connection counts of all APs assigned to each zone. Then, on each day, the time at which the first WiFi connection was detected was classified as arrival time, while the time at which the number of WiFi counts dropped to zero was classified as departure time. Cumulative relative frequency distributions of arrival and departure times in each zone were plotted to identify the probabilities of earliest/latest arrival and departure times in different zones. These indicators can then be used by building operators to schedule temperature setpoints

3.4 Summary

Following the identified gaps in the literature regarding the variation in the relationship of WiFi connection counts and occupancy counts as well as the integration of WiFi connection count data in BAS, a methodology was proposed in this chapter. The methodology consisted of two main modules including various components in order to introduce two frameworks for (i) validating the correlation between WiFi connection count and occupancy count, and (ii) extracting occupancy indicators using WiFi connection counts. The python codes for prediction models in modules are presented in Appendix A and Appendix B, respectively. Several machine learning algorithms and statistical analysis methods were utilized in these components in order to achieve the objectives based on which the modules were proposed. In the following chapters, the results of implementing each module in a case study are discussed in detail.

CHAPTER 4: VALIDATING THE CORRELATION BETWEEN WIFI CONNECTION COUNT AND OCCUPANCY COUNTS

4.1 Introduction

The first module, introduced in the research methodology, was proposed to address the gap in the literature regarding the variation in the correlation between WiFi connection counts and actual building occupancy counts. In this module, continuous ground-truth data, collected from camera-based occupancy counters was utilized to validate this correlation at building-level and investigate the influencing features affecting the variation of this correlation. Furthermore, multiple prediction models were developed to study the effectiveness of each of these features. Finally, the prediction models were proposed to estimate real-time occupancy count and predict/forecast day-ahead occupancy count at building-level.

To validate this framework, it was applied in a library building in Montreal, Canada with data collected between January and March 2020 (i.e., almost nine weeks) before the closure of the university due to COVID-19 pandemic. The case study building consisted of multiple space types with a high variation of occupancy level (i.e., occupancy counts ranged between 1 and 2,180). For the purpose of this study, the WiFi connection counts were collected from WiFi network administration platforms while the occupancy counts were collected from camera-cased occupancy counters located at the main entrances to the testbed. The case study building and the results of applying the module to it are presented in the following sub-sections.

4.2 Case Study

The proposed framework was applied in a university library building, located in Montreal, Canada. The library consists of four stories (total area of 19,180 m^2) within a 13-story university building, operated separately from the rest of the building. The layout and space types of these floors were similar with 40% of the total area being allocated to reading rooms, group study rooms, and lecture rooms, while offices covered 14% of the total area. A summary of the area of the different spaces identified in the selected four floors is presented in Table 4-1. The combination of multiple space types in these floors provided the opportunity to show a proof-of-concept of the proposed framework on a dynamic environment with a high variation of occupancy.

Table 4-1. The total area of different spaces in the investigated four floors of the case study building

Space type	Area (m^2)
Office	2,671
Corridor/Lobby	6,087
Reading room	7,110
Group study room/ Lecture room	630
Collection spaces	2,680

4.2.1 Building-Level Data Collection and Preparation

For this study, nine weeks of data, extended from January 13, 2020, to March 12, 2020, were collected from two sources, WiFi network management platform and camera-based occupancy counters. The anonymized reports of WiFi connection counts in 3-min intervals were generated by 152 Cisco Aironet AIR-CAP3702I and Cisco Aironet 2802I APs that were located throughout the four investigated floors of the case study building. For the purpose of this study, the data was aggregated hourly and flattened into one dataset for building-level analysis.

The counts of occupants entering or leaving the library in 5-min intervals were provided by two cameras, one optical and one thermal (manufactured by SenSource company[3]), which covered the two main gateways of the library section on the first floor (i.e., the primary entrance/exit point for the investigated four floors). The accuracy of these counter cameras was initially tested through manual counting at random points over time. Table 4-2 reports the average percentage error of these cameras (i.e., optical and thermal) in counting the number of occupants entering or leaving the library are reported, based on two hours of manual counting. According to the reported values, except for the low accuracy of the thermal camera in counting occupants leaving the library, the rest had a level of error under 15%. These two gateways, covered by the two cameras, are the only entrances/exits for visitors of the library section of the building. However, there are some other access points that are infrequently used, only by the staff and librarians for entering or leaving the library section.

[3] https://sensourceinc.com/

Table 4-2. The average percentage error of optical and thermal camera-based occupancy counters in counting the number of occupants entering (In) or leaving (Out) the library

Camera-based occupancy counter	In[1]	Out[1]	Total[1]
Optical	5%	2%	3%
Thermal	14%	40%	34%

(1) $Percentage\ error = \frac{|Approximate\ Value\ (camera) - Exact\ Value\ (observed)|}{Exact\ Value\ (observed)} \times 100$

The occupancy counts data was produced based on the collected values from both counters and was then aggregated hourly to match the WiFi connection count data's timestep. However, the calculations were imbalanced, i.e., the total number of occupants leaving the library during each day was greater than the total number of occupants entering the library. This mismatch, which may have been a result of the thermal camera's inaccuracy in counting outgoing occupants, or due to the staff taking other exit points, resulted in a downward trend in the produced camera-based occupancy counts time-series. These differences between ins and outs ranged from 0 to 250 occupants, i.e., around 11% of the maximum occupancy. They were adjusted by setting the minimum occupancy value of each day to zero and recounting occupancy each day accordingly. Figure 4-1 depicts the pseudocode showing how the camera-based occupancy count data was calibrated on each day. For the rest of the study, the preprocessed camera-based occupancy count data will be used as the ground-truth.

Pseudocode

Set d = day of the experiment, D = the total days of the experiment, h = hour of the day, H = the total hours of the day (i.e., 24 hours), $O_{(h)}$ = number of occupants at time h, $O_{(d)}$ = maximum error between Ins and Outs on each day

For each d in D

    $O_{(d)}$ = the absolute value of the minimum occupancy on that d

    For each h in H

        $O_{(h)} = O_{(h)} + O_{(d)}$

Figure 4-1. Pseudocode for calibrating camera-based occupancy count data on each day

As an example, the location of APs and optical and thermal cameras for the second floor of the case study building is presented in Appendix D.

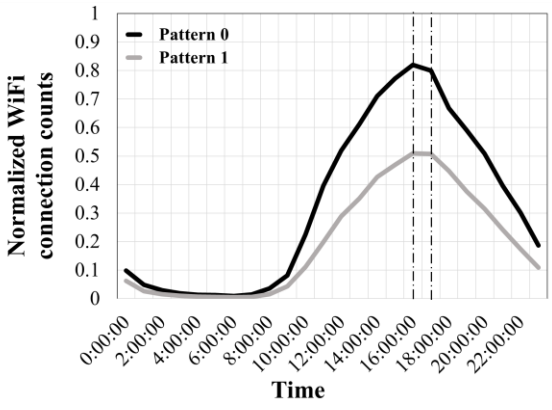
4.3 Results

This section presents the descriptive data analysis and discusses the results of implementing the first module in the case study building.

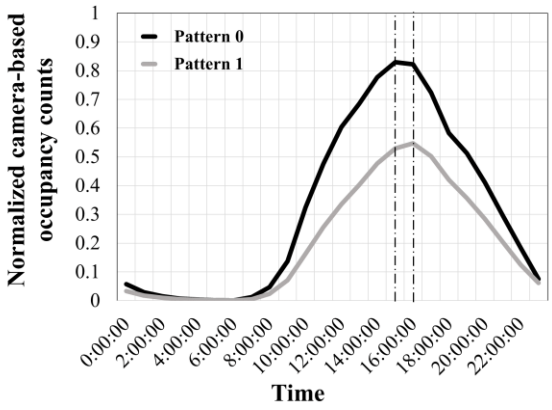
4.3.1 Descriptive Analysis

Normalized daily profiles of WiFi connection counts and camera-based occupancy counts were clustered to validate similarities between the daily patterns of the two datasets. Two clusters were identified, through k-means clustering, for each of these datasets which are plotted in Figure 4-2. Based on the visualization of the two clusters in each dataset, two similar shapes with different ranges of magnitude were observed. For camera-based occupancy, counts of both clusters rose to 10% at around 9:00 a.m., then reached a peak between 3:00 and 4:00 p.m. whereas they decreased to 10%, once again, around 10:00 p.m. Despite the shape similarity, clusters were different in terms of the average level of occupancy and the day they usually happened. Cluster 0, representing the higher level of counts, mostly happened during weekdays; while Cluster 1, showing the lower level of counts, typically occurred during weekends. Figure 4-3 represents the distribution of clusters between different days of the week, for each dataset.

Although the same patterns were observed in WiFi connection count data, dissimilarity can be seen between the peak times of this dataset which happened between 4:00 and 5:00 p.m. while camera-based peaks occurred between 3:00 and 4:00 p.m. The synchronization of timesteps of camera-based occupancy counters with actual time was validated through on-site observations. Therefore, it appears that WiFi connection count records show a one-hour lag. This error might be due to the WiFi network management platform reporting previous hour counts. To validate this assumption, another dataset was also created from WiFi connection counts, with values being shifted backward by one hour. The differences between camera-based occupancy count data and the modified as well as original WiFi datasets were compared based on two metrics of the RMSE and average absolute error. Based on the results, presented in Table 4-3, shifting WiFi connection counts backward by one hour resulted in significant error reduction, suggesting the one-hour shifted data to be a better representative of actual occupancy patterns.

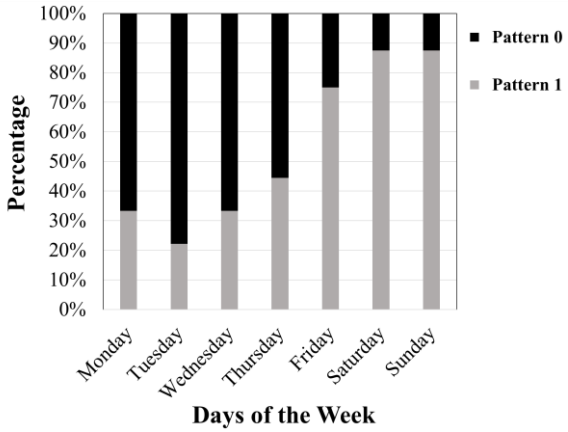


(a)

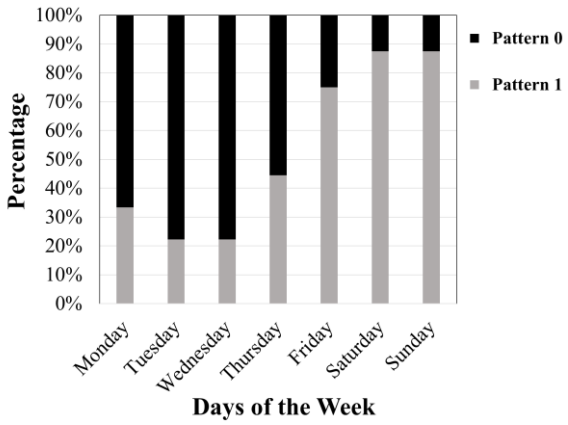


(b)

Figure 4-2. Two clusters of daily patterns for (a) WiFi connection counts, (b) camera-based occupancy counts.



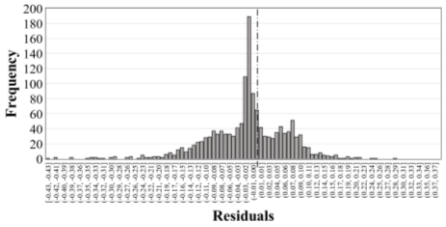
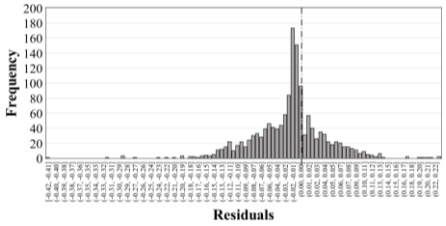
(a)



(b)

Figure 4-3. Membership of days of the week to each cluster for (a) WiFi connection counts, (b) camera-based occupancy counts

Table 4-3. Descriptive statistics of the two comparisons between camera-based occupancy count vs. WiFi connection count data and camera-based occupancy count vs. one-hour shifted WiFi connection counts

Descriptive statistics	**Normalized camera-based occupancy count data vs. Normalized WiFi connection count data**	**Normalized camera-based occupancy count data vs. Normalized one-hour backward shifted WiFi connection count data**
RMSE	0.086	0.062
Average absolute error +/- SD	(0.002, 0.122)	(0.000, 0.088)
Residual distribution		

Therefore, the shifted WiFi connection count data was used for the rest of the study. The statistically significant positive correlation between this data and camera-based occupancy count data is depicted in Figure 4-4, with a Pearson product-moment correlation coefficient (r) of 0.987.

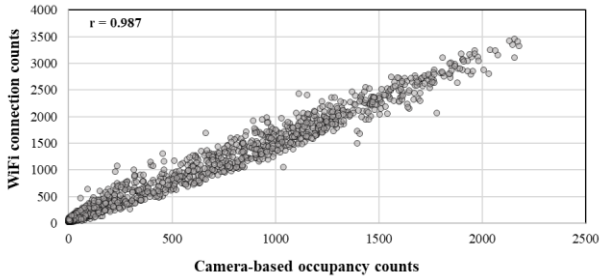


Figure 4-4. The correlation of WiFi connection counts and camera-based occupancy counts

The time-series of the two datasets are plotted in Figure 4-5 for the entire duration of the experiment. A slight upward trend was observed in both datasets during the first six weeks, which declined at week no. 7 (the week of the university's winter break) and again jumped up at week no. 8. Since the testbed was a library, it was expected that the number of occupants using the space grow from the beginning of the semester, up to the exams' week (week no. 8). However, since no classes took place during week no. 7, i.e., the break week, fewer students visited the campus and used the library.

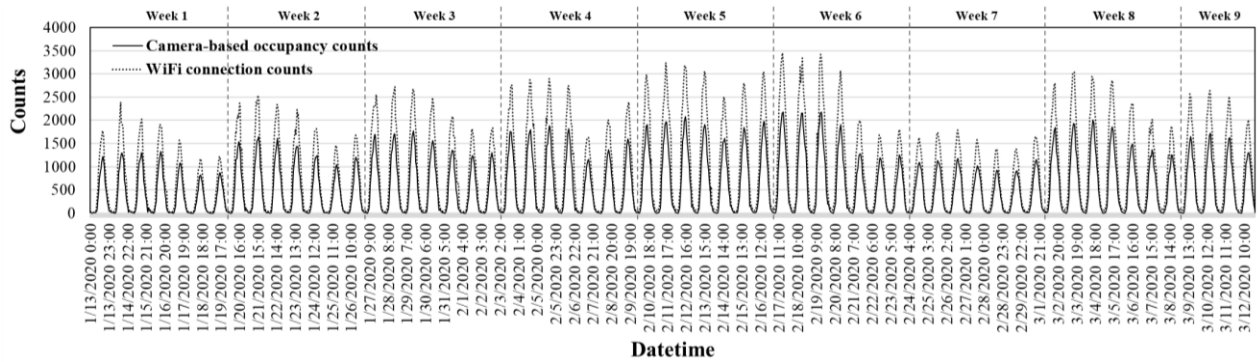


Figure 4-5. Nine weeks of WiFi connection counts and camera-based occupancy counts data in four floors of the case study building

Beside these intuitive observations, in order to quantitatively evaluate the variations, a Kruskal-Wallis H test was performed, which showed no statistically significant differences were observed among the days of each week with other weeks. Descriptive statistics of WiFi connection count and camera-based occupancy count data are presented in Table 4-4.

Table 4-4. Descriptive statistics of WiFi connection count and camera-based occupancy count data

Descriptive statistics	Camera-based occupancy count data	WiFi connection count data
Min	1	14
Max	2,180	3,449
Median	347.5	624
Mean +/- SD	(4.58, 1,119.92)	(18.16, 1,781.84)

4.3.2 Investigating the Relationship Between WiFi Connection Count and Camera-Based Occupancy Count Data

Although the investigated four floors of the case study building were operated 24 hours on all days of the week, the parts allocated to librarians' offices were more actively used during weekdays. As some stationary devices were located in these offices, the process of identifying stationary devices was conducted separately for weekdays and weekends. Identifying stationary devices focused on early morning periods due to the lower level of occupancy during these hours. The minimum values attributed to stationary devices identified for weekdays and weekends were 17 and 14, respectively, which were subtracted from WiFi connection counts at each interval of weekdays and weekends. Figure 4-6 shows the distribution of stationary device counts for weekdays and weekends.

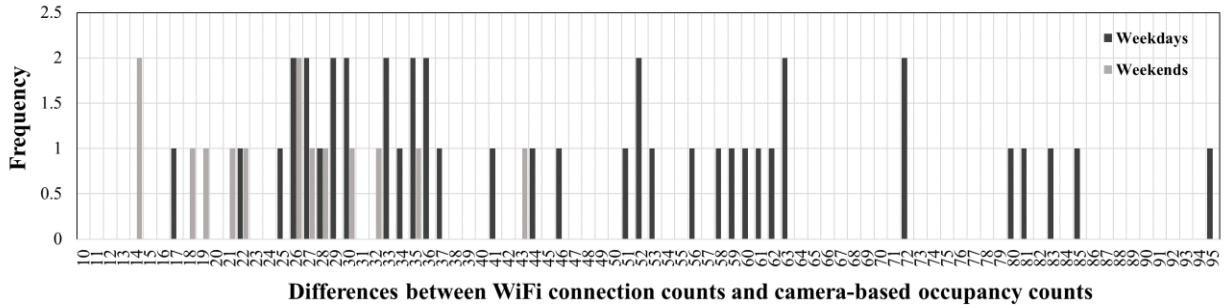


Figure 4-6. Distribution of stationary device counts for weekdays and weekends

The low values of WiFi connections and camera-based occupancy counts at the early morning times increased the conversion factor's fluctuation during this time. Therefore, for the rest of the study, including the calculation procedure of conversion factor, the study was limited to the hours when occupancy was more than 10% and space was more actively used (i.e., 9:00 a.m. to 10:00 p.m.). Given the fact that implementing occupancy-based active control systems and optimizing the mechanical system's schedule are mostly targeted during the same hours, this assumption is not expected to create limitations for this study. The standard deviation of the conversion factor on hours between 9:00 a.m. and 10:00 p.m. was less than 0.2. It was expected to improve the predictability by minimizing the time window to these hours with lower variation, compared to highly fluctuating early morning times. The time-series of the conversion factors was then produced for the selected hours on all days of the entire experiment duration. Figure 4-7 presents the range of these conversion factors on each of these selected hours. The variation of conversion factor between different hours of the day showed the significant influence of time on these values.

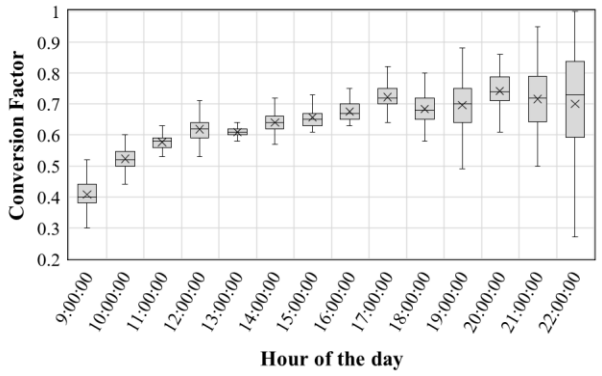


Figure 4-7. The hourly box plot of WiFi-occupancy conversion factors

K-means clustering was then used to investigate the dominant patterns among the daily profiles of conversion factors (i.e., between 9:00 a.m. and 10:00 p.m.). These profiles, along with the centroids of identified clusters are plotted in Figure 4-8. Considering the membership of days of the week to each cluster, as shown in Figure 4-9, the characteristics of clusters can be summarized as follows.

Firstly, both clusters generally followed ascending patterns with similar centroid values between 9:00 a.m. and 5:00 p.m., when they started to diverge. Secondly, Cluster 0, which was the dominant cluster during the weekends, continued rising until 10:00 p.m., suggesting that the number of occupants and WiFi connections got closer through the end of the day during these days. These were the days that typically experienced the peak occupancy around 4:00 or 5:00 p.m. with a lower average occupancy rate. However, Cluster 1 mostly happened during weekdays with a peak occupancy, occurring around 3:00 or 4:00 p.m. In this cluster, the conversion factor started to decline after 5:00 p.m. Although weekends were almost clustered together, the weekdays were split among the two clusters, with almost equal membership percentages. This means that the 'day' for the weekdays, had a small effect on daily conversion factors.

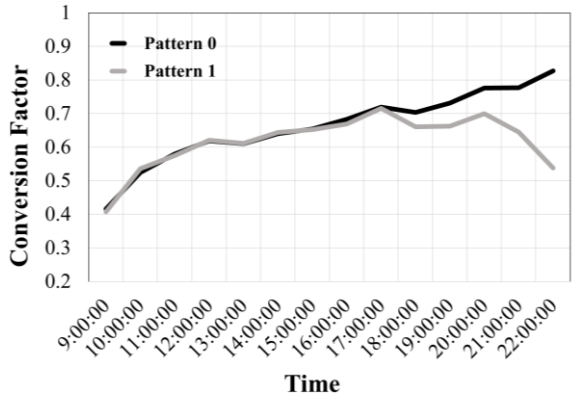


Figure 4-8. Two clusters of daily conversion factor

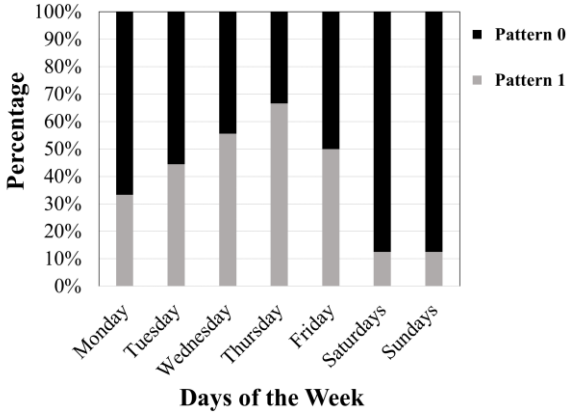


Figure 4-9. Membership of days of the week to each cluster of daily conversion factor profiles

The level of occupancy was another feature that influenced the conversion factor's variations. Figure 4-10, illustrates the relationship between camera-based occupancy counts, the conversion factor, and the hour of the day. Based on this figure, the conversion factor experienced high variations when the occupancy counts were lower; while it converged to a steady value, i.e.,

around 0.65, as the occupancy counts increased toward the peak level. A virtual horizontal line at conversion factor equal to 0.65 splits the records into two almost distinct areas, above and below the line. Those records below the line were mostly happening before the peak time., i.e., before 3:00 or 4:00 p.m., and had a lower conversion factor, i.e., between 0.2 and 0.65. However, the records above the line were mostly happening after the peak time and had higher values of conversion factor, i.e., between 0.65 and 1. Comparing two random records from above and below the line (i.e., after and below peak time, respectively) with the same level of occupancy counts (e.g. at occupancy count equal to 300), revealed that the difference between these two records resulted from the difference in their WiFi connection counts. Fewer WiFi connections were typically detected after (compared to before) the peak time. This might be due to the operation of more stationary devices before peak times, compared to after peak time when the offices are closed. More importantly, this difference might have resulted from the difference in the profile and/or behavior of occupants who use the library after the peak time, manifested in the use of several devices connected to the WiFi.

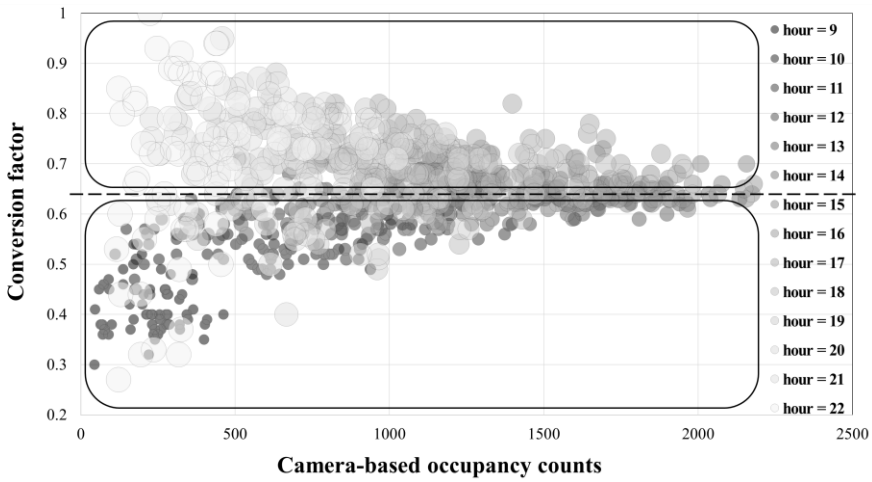


Figure 4-10. The relationship between camera-based occupancy counts, the conversion factor, and the hour of the day

Based on these observations, three features were recognized as influencing factors on the variation of conversion factor, including, Hour of the day, Day of the week, and occupancy level. Besides this intrinsic study of features' importance, an extrinsic analysis of these features' impact was performed through developing prediction models to estimate occupancy counts. To investigate the effect of each of these features in improving the performance of the prediction model, five

different combinations of these features were considered, as follows: (i) Hour of the day, (ii) Hour of the day, and Day of the week, (iii) WiFi connection count, (iv) Hour of the day and WiFi connection count, and (v) Hour of the day, Day of the week and WiFi connection count. For each set of the input features, two separate prediction models were developed for weekdays and weekends using MLR. The performance of all created models was evaluated by averaging the score of R^2, RMSE, and MAPE in all folds of time-based k-fold cross-validation. In each fold, the model was tested on one day alone while it was trained on the rest of the days. Therefore, based on the available data for weekdays and weekends, *k* was set to 44 and 16, respectively. The results are presented in Table 4-5. The analyses of features significance for all developed models are presented in Appendix C.

Among the models developed based on different combinations of features, the models including WiFi connection count as the input showed better performance than those with time-related features alone. Although the fifth set of models was built based on all the three features, they had almost similar performance to the fourth set that did not include the Day of the week as a predictor and showed the best performance of all (with an R^2 of 0.96 and 0.98, RMSE of 81 and 48, and MAPE of 9% and 7% for Weekdays' and Weekends' model, respectively). This stressed the low influence of the feature, Day of the week, on the relationship variation between WiFi connection counts and occupancy counts.

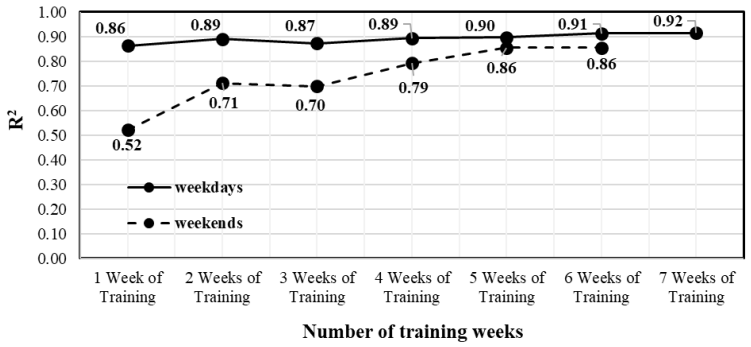
Table 4-5. Performance of the prediction models to estimate real-time occupancy counts

Models set No.	**Combinations of input features**	**Datasets**	**R^2**	**RMSE**	**MAPE**	**Residuals (**Mean+/-SD**)**
1	Hour of the day	Weekdays	0.50	230	27%	0.54 +/- 263.89
		Weekends	0.54	193	31%	-0.6 +/- 244.98
2	Hour of the day, Day of the week	Weekdays	0.53	225	27%	3.5 +/- 257.73
		Weekends	0.52	200	32%	-2.62 +/- 253.9
3	WiFi	Weekdays	0.92	117	14%	-0.12 +/- 120.5
		Weekends	0.94	87	17%	0.51 +/- 89.33
4	Hour of the day, WiFi	Weekdays	0.96	81	9%	0.12 +/- 90.38
		Weekends	0.98	48	7%	0.47 +/- 49.55
5	Hour of the day, Day of the week, WiFi	Weekdays	0.96	82	9%	-1.29 +/- 90.11
		Weekends	0.98	48	7%	0.1 +/- 49.93

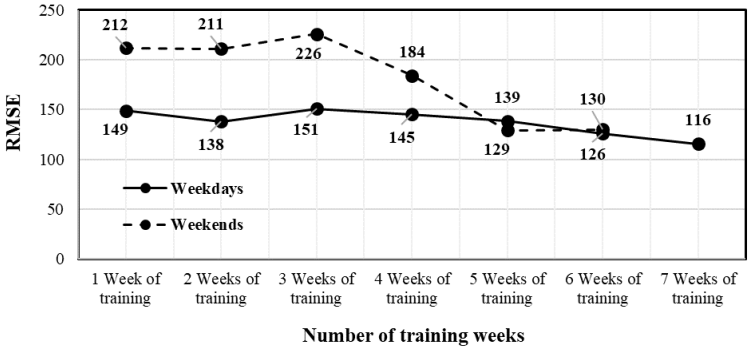
4.3.3 Day-ahead Occupancy Counts Prediction

Comparing the performance of prediction models developed for all days, versus prediction models for weekdays and weekends, suggested that having two separate models for weekdays and weekends can improve the prediction performance. Therefore, two sets of prediction models were trained to forecast day-ahead occupancy counts on weekdays and weekends using normalized WiFi connection counts of the previous day; Despite the low significance of Day of the week in the real-time prediction model, including this feature besides Hour of the day improved the performance of day-ahead prediction model. Therefore, these two features, Hour of the day and Day of the week were also used as inputs to the model. While two separate sets of models were developed for weekdays and weekends, still the Weekdays' model used the WiFi connection counts of Sundays to forecast actual occupancy counts for Mondays. This is also true for forecasting Saturdays from the WiFi connection counts of Fridays, in the Weekends' model. For training these models, although according to the Kruskal-Wallis test results, the data of week no. 7 did not show a significant difference with days of other weeks; the developed models could not predict this week accurately, due to their slightly different occupancy patterns. Therefore, week no. 7 was considered an anomaly, hence was removed from the dataset and the prediction models were developed based on eight weeks of data.

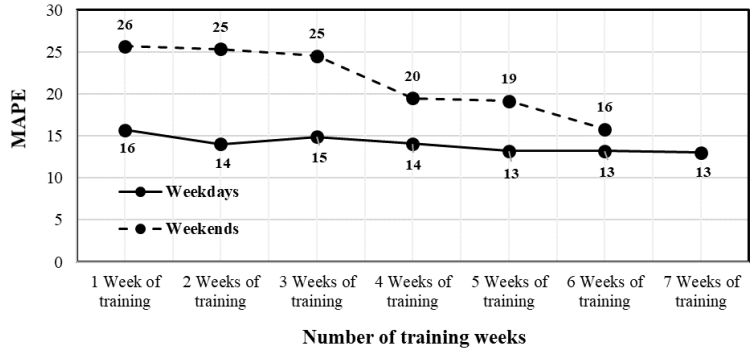
The accuracy of day-ahead forecasting models was improved by introducing more data for training the model. This was investigated through availing more data in the Weekdays' and Weekends' prediction models, starting from one week of training data up to seven weeks of training data. Figure 4-11 shows the results of average R^2, RMSE, and MAPE values of models trained based on various training windows. The accuracy of Weekdays' model was marginally improved by increasing the number of training weeks since the trained weeks followed almost similar patterns. This improvement was, however, more significant for Weekends' model. Since weekends experienced higher variations of occupancy patterns, by extending the training window, the model captured more information about the occupancy and could better learn and forecast the day-ahead's behavior.



(a)



(b)



(c)

Figure 4-11. The average performance of day-ahead occupancy prediction with prediction models trained on different numbers of weeks assessed based on (a) R^2, (b) RMSE, and (c) MAPE

Although the study was limited to seven weeks of training data, a significant improvement was shown by training the Weekends' model based on seven weeks of data. Therefore, the performance of the final Weekends' model was evaluated by averaging the values of R^2, RMSE, and MAPE in cross-validation, presented in Table 4-6. Despite the high performance of the Weekdays' model trained even based on one or two weeks of data, the final Weekends' model was also trained on seven weeks of data. The average R^2, RMSE, and MAPE values from cross-validation for this model are also reported in the same table. For both models, the statistical characteristics of residuals show a mean near zero in all two prediction models, indicating that the models have not been biased. Moreover, due to the drastically wide range of values for the dependent variable (from 44 to around 2,180 counts), the resulting RMSE (ranging between 116 and 149) is deemed acceptable. According to the table, Weekdays' model achieved a significantly higher R^2 value (0.92), compared to the Weekends' model (with an R^2 of 0.82). Furthermore, the results of MAPEs showed an acceptable level of error in the Weekdays' as well as Weekends' models (with 14% and 22% error, respectively). The breakdown performance of the Weekdays' and Weekends' models for each day of the week can be compared in the same table. The analyses of features significance for all developed models are presented in Appendix C.

Although the performance of day-ahead occupancy count forecasting models was lower than the performance of models developed for estimating real-time occupancy count, they could successfully forecast daily building occupancy counts with relatively high accuracy. As an example, Figure 4-12 shows the result of the Weekdays' and Weekends' models, for one weekday and one weekend. To forecast these two days, Weekdays' and Weekends' models were trained with the rest of the weekdays and weekends, respectively. These future prediction models provide occupancy information ahead of time which can be practical for proactive control of building systems' operation. Using these models compared to real-time occupancy estimation models would also decrease the response time of systems.

Table 4-6. Performance of prediction models to predict day-ahead occupancy counts for Weekdays and Weekends

Prediction Model	Days of the Week	R^2	RMSE	MAPE	Residuals (Mean+/-SD)
Weekdays	Mondays	0.94	113	14%	-2.47 +/- 114.90
	Tuesdays	0.95	110	13%	5.12 +/- 113.45
	Wednesdays	0.93	122	12%	0.06 +/- 128.36
	Thursdays	0.94	96	11 %	1.03 +/- 105.77
	Fridays	0.82	140	21%	7.91 +/- 148.14
	All weekdays	0.92	116	14%	2.31 +/- 122.56
Weekends	Saturdays	0.76	175	26%	-1.92 +/- 188.81
	Sundays	0.89	122	19%	-6.56 +/- 124.29
	All weekends	0.82	149	22%	-4.24 +/- 159.86

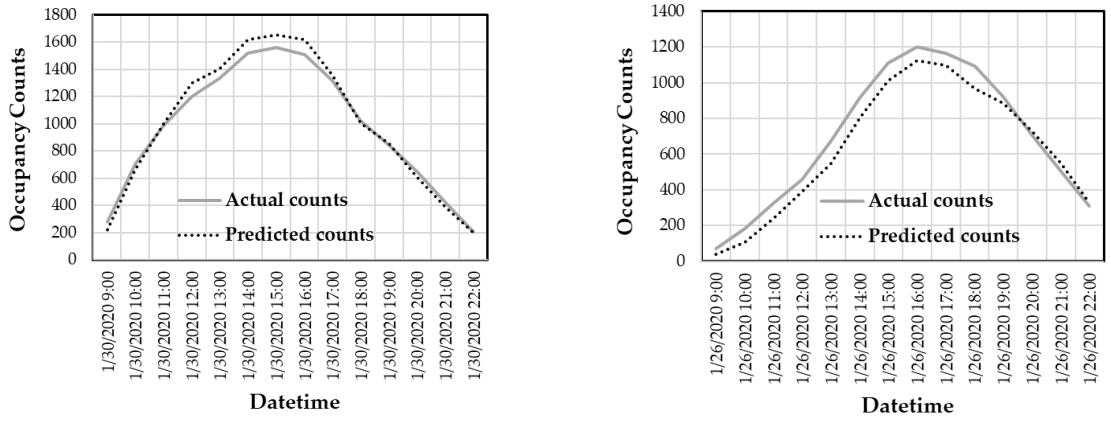


Figure 4-12. The prediction result of weekdays and weekends models, for (a) one weekday and (b) one weekend

4.4 Summary and Conclusions

Using camera-based occupancy counters as a source of continuous ground-truth data, to investigate the correlation between WiFi connection counts and actual occupancy counts showed dynamism in the conversion factor over time, which cannot be ignored. Furthermore, the study of influencing features on this relationship in different situations including real-time occupancy estimation as well as future occupancy forecasting provided key insights that can be summarized as follows.

- The results showed the variability of WiFi-occupancy conversion factors throughout the day, as well as between different days. Identification/quantification of this dynamism is one of the main contributions of the present study. This variation must be considered in models using WiFi connection counts as a proxy for the occupancy counts, especially when they are used by building systems' operators to assess the actual demand for ventilation.
- The WiFi-occupancy conversion factor was affected by the time of day, day of the week, and occupancy level. Therefore, considering a fixed value for the conversion factor in translating WiFi connection count to real occupancy counts, at different times and occupancy levels (as formerly done in the literature) can result in accuracy losses.
- The WiFi-occupancy conversion factor tended to converge to a fixed threshold, as the occupancy level increased to the peak level on different days; suggesting that during peak hours, the WiFi-occupancy conversion factor experienced the lowest variation between different days.
- WiFi connection count was the most significant variable in prediction models developed to estimate real-time occupancy counts, while other features had low significance levels. Since the variation of WiFi connection counts was already affected by time, this feature alone reflects the temporal variation. Therefore, the model developed based on this feature alone, achieved considerably high performance. This performance was slightly improved by introducing a time-related feature, i.e., Hour of the day which added more information about occupancy patterns that were not captured by the WiFi connection count. That said, adding the feature Day of the week did not improve the performance of real-time occupancy estimation models due to the slight difference between days of the week, in this case study.

- Although the WiFi connection count was recognized as the most significant feature for estimating real-time occupancy counts; this feature was unable to fully capture occupancy count variations for future occupancy forecasting. Therefore, in the trained models, the feature Day of the week (i.e., a time-related feature) was added to introduce the temporal variation to the model. In day-ahead forecasting models, the significance of this feature improved compared to the real-time occupancy estimation models.
- Comparing the models estimating real-time occupancy counts with those forecasting future occupancy counts showed that for future forecasting, a longer duration of previous data including more irregular patterns was needed. However, testing the increase in data was limited due to the COVID-19 pandemic and lockdown of the case study building.
- The day-ahead forecasting model could successfully provide accurate information regarding the next day building occupancy counts which can be practical in identifying the actual demand of the building for ventilation.

CHAPTER 5: EXTRACTING KEY OCCUPANCY INDICATORS FROM WIFI CONNECTION COUNT DATA

5.1 Introduction

The second module, introduced in the research methodology, was proposed to address the gap in the literature regarding the integration of WiFi connection count data in BAS. In this module, WiFi connection count data was utilized alone to learn daily occupancy patterns and predict week-ahead occupancy patterns at the building level. Moreover, models were developed to predict peak occupancy while the analyses were performed to identify its occurrence interval on different days of the week. Finally, analyses were performed to identify earliest/latest arrival and departure times at a zone-level.

To validate this framework, it was applied in an academic building in Montreal, Canada with data collected between January and March 2020 (i.e., almost nine weeks) before the closure of the university due to the COVID-19 pandemic. The case study consisted of multiple space types with a high variation of occupancy level (i.e., WiFi connection counts ranged between 3 and 3,076). For the purpose of this study, the WiFi connection counts were collected from WiFi network administration platforms. The case study building and the results of applying the module to it are presented in the following sub-sections.

5.2 Case Study

The proposed framework was tested on a 17-story academic building with a gross floor area of 37,000-m^2, located in Montreal, Canada. The building is LEED silver certified, and its floors vary in layout and space type. Classrooms are mostly located on the lower eight floors, starting at sub-basement two through the sixth floor. Therefore, these eight floors were selected for this case study. The specifications of each floor retrieved from the BIM are summarized in Table 5-1.

Table 5-1. Specifications of the first eight floors of the case study building

Floors	Classroom/ Auditorium (m^2)	Study/ Meeting room (m^2)	Corridor/ Lobby (m^2)	Service (m^2)	Office (m^2)	Number of APs
Sub-basement 2	1050	80	870	450	0	14
Sub-basement 1	800	100	1100	400	0	10
Ground floor	600	0	1600	250	0	14
Floor 2	1100	260	840	200	0	11
Floor 3	1000	260	940	200	0	12
Floor 4	150	50	920	200	680	8
Floor 5	700	170	850	180	100	9
Floor 6	380	160	940	180	340	11
Total	5780	1080	8060	2060	1120	89

5.2.1 Building-Level Data Collection and Preparation

A total of 89 Cisco Aironet AIR-CAP3702I and AIR-CAP3602I APs are located throughout the first eight floors of the case study building; mostly concentrated in classrooms and auditoriums. According to the Instructional and Information Technology Services (IITS) team's verification, these APs cover the entire eight floors of the building and there are no dead zones on these floors. For this study, the anonymized reports of WiFi connection counts (generated by the APs) were provided in 3-min intervals for the period of nine weeks, from January 13, 2020, to March 12, 2020, including one week of winter break at which no classes took place. The collected data from the APs of the entire eight floors were aggregated into hourly counts and were flattened into one dataset for the building-level analysis.

5.2.2 Zone-Level Data Collection and Preparation

Four zones with two different space types, i.e., classroom and office spaces, were selected for investigating early and late arrival/departure times. The number of the APs assigned to these zones, as well as the space type that they support are presented inTable 5-2. Considering the position of these APs in the selected zones, and their coverage which is about 10 to 12 meters (Cisco, 2014), these APs reported the WiFi connection counts that were within the desired zones with minimum error. The collected data from these APs were aggregated into 30-minute intervals. For the purpose of arrival/departure times analysis, WiFi connection counts on weekdays were selected.

As an example, the location of APs for the second floor of the case study building as well as the zone of class1 are presented in Appendix D.

Table 5-2. Description of selected APs and corresponding spaces

Zone name	Space type	Number of APs in each zone
Class1	Classroom	2
Class2	Classroom	2
Office1	Office	1
Office2	Office	1

5.3 Results

This section presents the descriptive data analysis and discusses the results of implementing the second module in the case study building.

5.3.1 Descriptive Analytics

Figure 5-1 shows the WiFi connection count data, over the entire eight floors of the building, within the study period. The time-series contained almost nine weeks with similar weekly patterns, except for week no. 7, which coincided with the university's winter break. The total number of WiFi connection counts during weekdays of this week was significantly lower than other weekdays, therefore it was considered an outlier and was removed from the analysis.

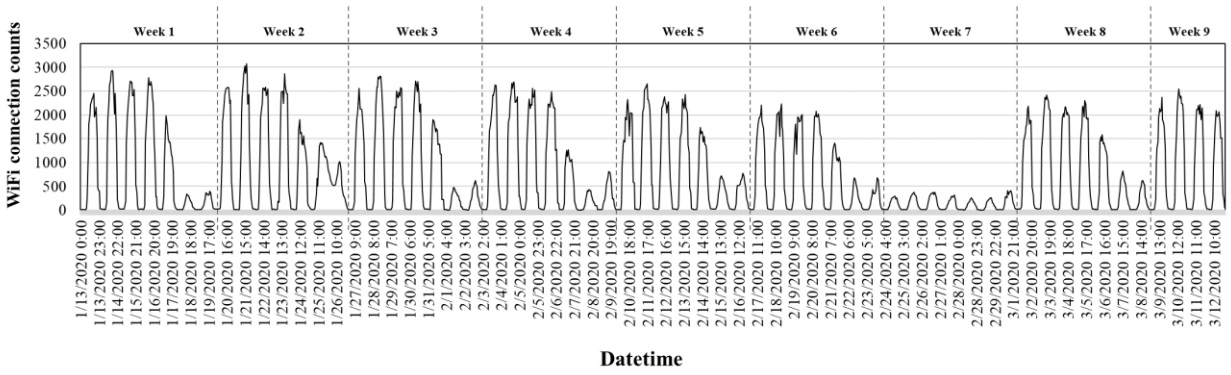


Figure 5-1. Nine weeks of WiFi connection counts in eight floors of the case study building

As expected, similar patterns were observed among weekdays, which were different from the weekends. This difference suggested developing separate models for weekdays and weekends. Despite variations in the patterns of Mondays through Fridays, based on the results of the Kruskal-Wallis H test, no statistically significant differences were observed among them, indicating that it is acceptable to group weekdays as one dataset for the next step. Moreover, Saturdays and Sundays

can be grouped, except for week no. 2, in which some special events may have taken place. Descriptive statistics of weekdays and weekends datasets are presented in Table 5-3.

Table 5-3. Descriptive statistics of weekdays and weekends datasets

	Weekdays	**Weekends**
Min	7	3
Max	3,076	823
Median	708	130
Mean +/- SD	(31, 1,967)	(-16, 422)

The range, quantiles, median, and mean of the building's peak WiFi connection counts for each day of the week during the experiment are plotted in Figure 5-2. Three distinct levels of peak values were observed, which were correlated with the days of the week as follows, (i) Monday through Thursday; (ii) Friday; and (ii) Saturday, Sunday; showing an average of about 2455, 1650, and 610 WiFi connections, respectively.

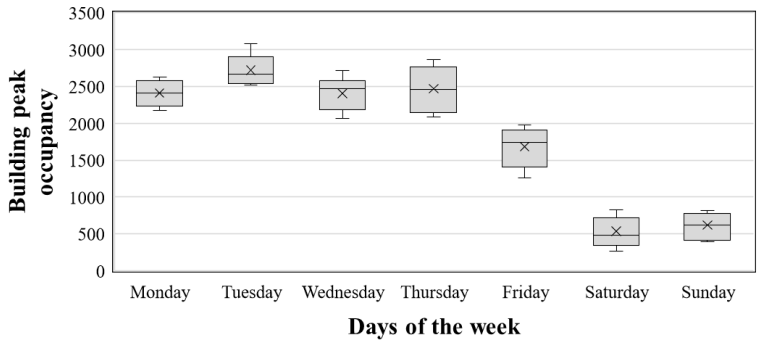


Figure 5-2. Building peak WiFi connection counts (occupancy) for different days of the week within the study period

5.3.2 Occupancy Pattern Prediction

In order to identify the governing behavioral regimes, representative day types were extracted using *k*-means clustering. The range of values tested for *k* was limited to 3-10 to avoid the relatively small or large number of clusters. On the one hand, a high value for *k* may lead to over-fitting and might not help to improve the accuracy of prediction. On the other hand, a low value might overshadow some frequent patterns. Moreover, a cluster with a low number of items is not acceptable as a representative of frequently repeated daily occupancy patterns. According to the plot in Figure 5-3, *k* equals both 3 and 5 showed low DBI values. However, setting *k* equal to 5

resulted in having one cluster with only two data-points, which was not acceptable. Therefore, *k* equal to 3 was chosen as the optimal number of clusters for k-means clustering.

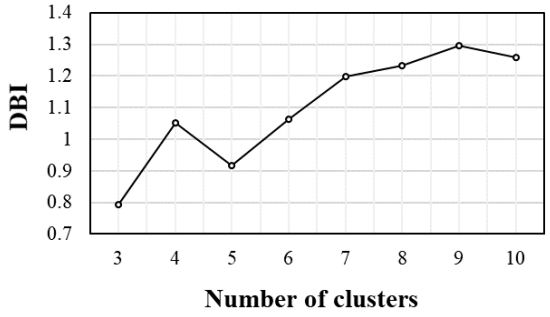


Figure 5-3. DBI graph of different values of *k* in k-means clustering of daily occupancy patterns

Figure 5-4 shows the three clusters of daily occupancy patterns. In all three patterns, the counts start to increase at 7:00 a.m. and continue with a sharp growth between 8:00 a.m. and 10:00 a.m., with a rapid decrease after 7:00 p.m. The major difference between the three clusters is the average level of peak WiFi connection counts, and the time of its occurrence, which can be summarized as follows.

- Pattern 0 showed an average peak value of about 1,700, by around 11:00 a.m.
- Pattern 1 showed an average peak value of about 2,200, by around 4:00 p.m.
- Pattern 2 showed an average peak value of about 2,700, by around 4:00 p.m.

The membership of days of the week to each cluster is shown in Figure 5-5. Pattern 1, which represented the lowest average of WiFi connection counts with a distinct shape, was the only fully homogeneous cluster, entirely allocated to Fridays. This can be associated with Fridays' characteristics; not only fewer classes are usually scheduled during Friday evenings, but it is also less common to organize events (such as presentations, talks, or workshops) on Friday evenings. The other two patterns happened on the other four days, with an almost similar rate on Mondays, Wednesdays, and Thursdays. However, the dominant pattern for Tuesdays was pattern 3, which represented the highest average of WiFi connection counts, suggesting that more classes and events happen during Tuesdays in this building.

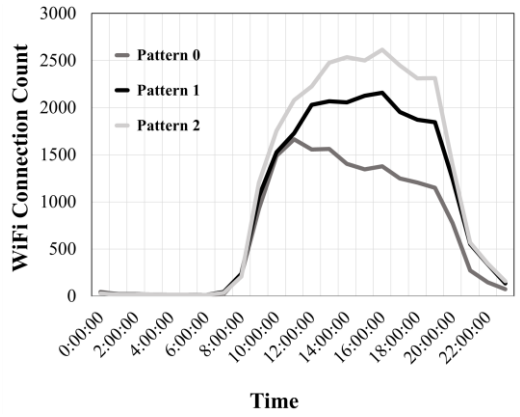


Figure 5-4. Three clusters of daily occupancy patterns of weekdays

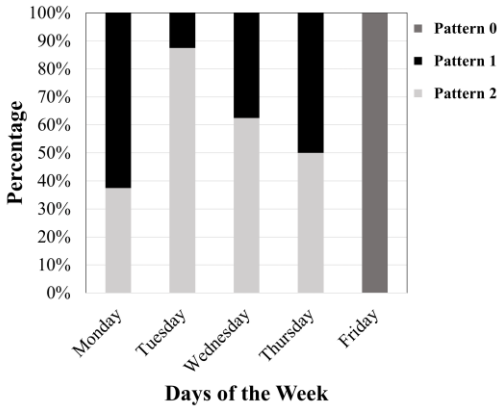


Figure 5-5. Membership of days of the week to each cluster

These findings, including (i) the significant difference between weekdays' and weekends' occupancy; and (ii) having Fridays as a separate cluster, suggested dividing the complete dataset into three independent subsets, i.e., (i) Weekdays (Monday through Thursday); (ii) Fridays; and (iii) Weekends. Three separate prediction models were then developed for each subset, to improve the performance of week-ahead occupancy pattern prediction. In these three models, the hours around peak times had higher coefficients which was expected, given that the WiFi connection count was of a considerably larger order of magnitude during these hours.

The performance of the three prediction models was evaluated through time-based *n*-fold cross-validation employed based on the available data of the three subsets, with *n* equals 8, 7, and 7 for 'Weekdays', 'Fridays', and 'Weekends', respectively. As an example, Figure 5-6 shows the combined results of all three models, for one fold of the *n*-fold validation. In this fold, week no. 8 was considered as the test dataset, while all three models were trained with the rest of the weeks.

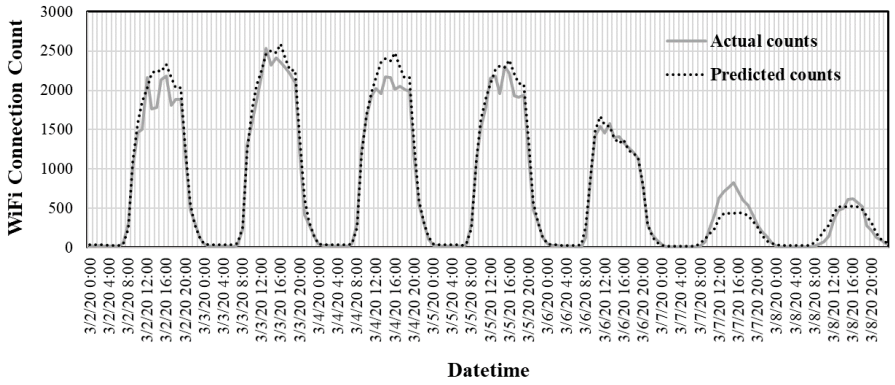


Figure 5-6. The combination of all three prediction models' results for predicting test dataset, week no.8, versus actual values, starting from Monday, ending on Sunday

The performance of each of the three prediction models in predicting week-ahead occupancy patterns was evaluated by averaging the score of R^2_D, RMSE, and MAPE in all folds of cross-validation. The results are reported in Table 5-4. The statistical characteristics of deviance residuals show a mean near zero in all three prediction models, indicating that the models have not been biased. Moreover, since the dependent variable ranged from 3 to around 3,000 counts, the resulted RMSE (ranging between 110 and 178) is deemed acceptable. According to the table, 'Weekdays' and 'Fridays' models achieved significantly higher R^2_Dvalues (0.98 and 0.97, respectively), compared to the 'Weekends' model (with an $R^2_D$of 0.81). Furthermore, the results of MAPEs showed a high level of error in the 'Weekends' model (83.99%) compared to 'Weekdays' and 'Fridays' models (with 42.79% and 38.33% errors, respectively). The lower performance of the 'Weekends' model can be justified due to the considerably higher frequency of temporary events, such as short-term workshops and special meetings, during the weekends. Such events make the occupancy patterns far less predictable, while the university schedule during weekdays can help to regularize the occupancy patterns to some extent. On the other hand, comparing the MAPEs of working time and nighttime prediction, revealed that due to the small number of WiFi connections during nighttime as well as the significant discrepancy between working time and nighttime WiFi connection counts, the nighttime prediction has been the main contributor to the overall error. Since the present study was mainly aiming at the prediction of occupancy patterns during working time, the MAPEs calculated for this period will be of the main concern. Those

errors are of a considerably lower magnitude, and range between 12.49% to 16.61%. Figure 5-7 presents the breakdown of errors for days of the week and hours of the day. A higher level of errors was observed for nighttime, which resulted from the lower number of connection counts during this time and likely more sporadic occupancy outside the school's operation hours. During the working time when occupants use the space more actively, the level of error was lower and more stable (around 20%). Comparing the prediction errors during different days of the week also showed lower performance of the 'Weekends' model which was mainly due to the lower number of counts as well as irregular events such as workshops. The analyses of features significance for all developed models are presented in Appendix C.

Table 5-4. Performance of three prediction models developed on data subsets (i) Weekdays, (ii) Fridays, and (iii) Weekends

Prediction model	$\mathbf{R^2_D}$	**RMSE**	**MAPE**			**Deviance Residual** (Mean+/-SD)
			All predictions	*working time prediction*	*nighttime prediction*	
Weekdays	0.98	178	42.79%	12.49%	85.22%	-0.104 +/- 4.077
Fridays	0.97	143	38.33%	16.61%	68.74%	- 0.106 +/- 3.956
Weekends	0.81	110	83.99%	60.50%	116.87%	- 0.289 +/- 5.233

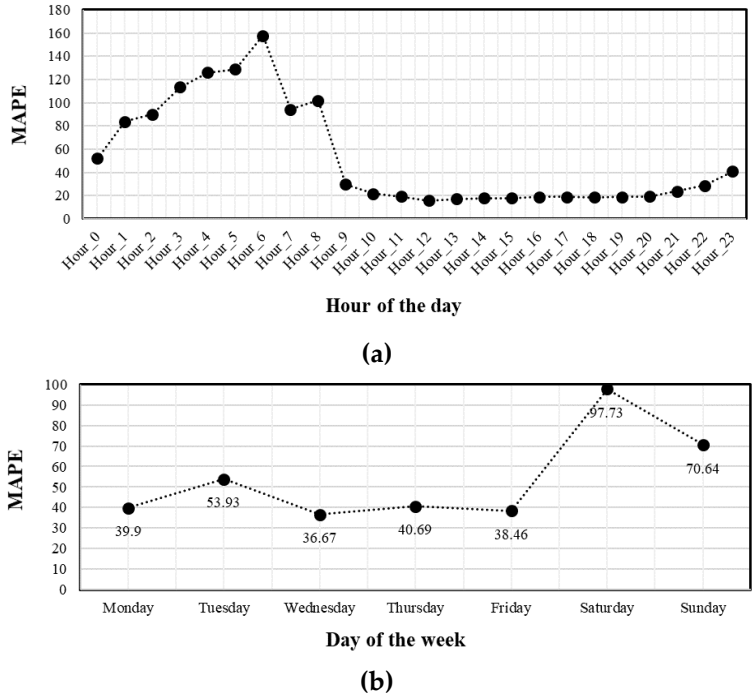


Figure 5-7. The breakdown of errors over (a) hours of the day, (b) days of the week

5.3.3 Peak Occupancy Prediction

Multiple prediction models were developed and tested to predict week-ahead peak WiFi connection counts which is a proxy for peak occupancy. This process relied on using a range of training sets, starting from one week of training data up to seven weeks of training. As plotted in Figure 5-8, the resulting average R^2_D value from cross-validation for each certain number of weeks of training was slightly improved by increasing the number of training weeks from one to two, where it reached its highest value (of 0.94). Extending the training window to beyond two weeks caused the R^2_D to decline. This can be due to introducing more variation in counts to the model which causes more noise. Therefore, two weeks of training prior to the week of the target was determined as the optimum number of weeks required for peak occupancy prediction in this building for the winter semester.

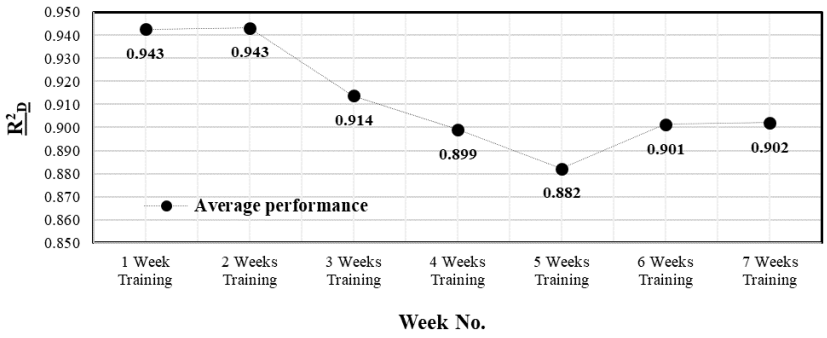


Figure 5-8. The average performance of peak occupancy prediction with prediction models trained on a different number of weeks

To further investigate the occurrence of peak building occupancy, the cumulative relative frequency distributions of the time at which the WiFi connection counts reach their daily maximum were plotted for each day of the week (see Figure 5-9). Based on the plot, the most apparent finding was the significant difference between the time at which peak occupancy occurs on different days. In 90% of the time, peak occupancy occurs on or before 4:00 p.m. on Mondays, Tuesdays, and Thursdays; while Fridays and Wednesdays experience peak occupancy on or before 1:00 p.m. and 6:00 p.m., respectively. These variations in the probable peak occupancy time were also observed on weekends, which can help the building operators with adjusting ventilation schedules and ensuring maximum outdoor air intake coincides with the timing of peak occupancy on different days.

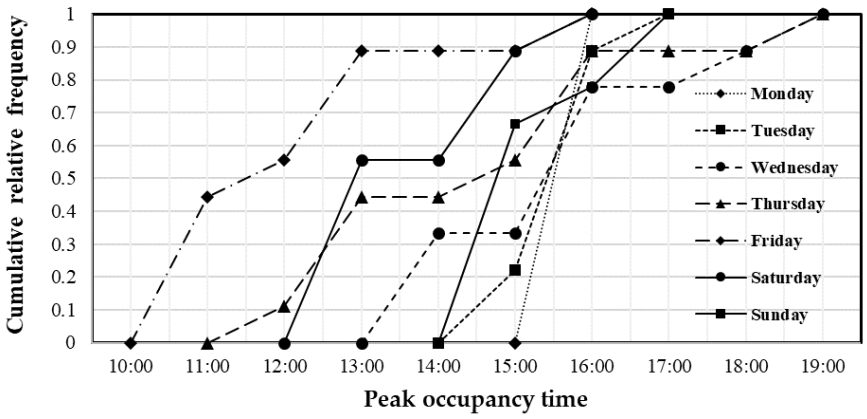


Figure 5-9. Cumulative relative frequency distribution of building peak occupancy time

5.3.4 Arrival and Departure Times Analysis

A similar approach was followed to identify arrival/departure times at the four selected zones. The cumulative relative frequency distributions in Figure 5-10 showed considerable variation in arrival/departure times of zones, regardless of their space types. For example, the earliest arrival time observed in Office 1 was 5:30 a.m. while it was 7:00 a.m. in Office 2. Based on these observations, the heating setback schedule in Office 2 could be extended relative to Office 1.

In 99% of the time, it was observed that the latest arrival times for Offices 1 and 2 were 7:30 a.m. and 1:00 p.m., respectively, which were also different for classrooms. Building operators could use this information to schedule earlier temperature set-backs in these rooms if no occupancy is detected after the observed latest arrival time, which would avoid heating these empty rooms for several hours when standard operational schedules are used.

The same analysis was extended to departure times in the selected four rooms which also showed large variations. In 90% of the time, it was observed that the latest departure times were 10:00 p.m. and 1:00 a.m. in Offices 1 and 2, respectively. This information could also be used to adjust the timing of initiating temperature setbacks in different rooms. In many cases, building operators resort to keeping the temperature setpoint for 24 hours if building occupancy is irregular to avoid occupant complaints. However, this practice can be avoided by gaining insights into the occupancy patterns of specific rooms and zones using the proposed approach.

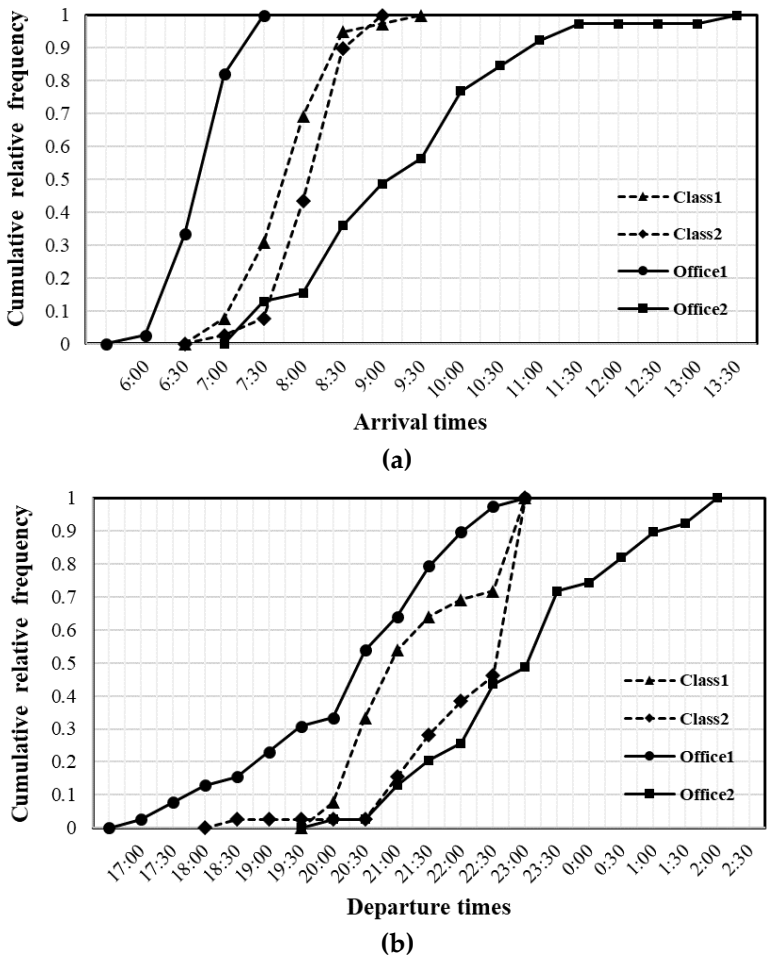


Figure 5-10. Cumulative relative frequency distributions of (a) arrival times of occupants in four zones (b) departure times of occupants in four zones

It must be noted that the results of this analysis are only meant to show a proof of concept of the proposed methodology and the observed occupancy pattern variations in different rooms. However, data collection would have to be extended beyond the duration of this case study for at least one year to capture seasonal variations and ensure results accuracy, before making any changes to the control sequences of building systems.

5.4 Summary and Conclusions

The proposed framework aims to identify practical occupancy indicators using WiFi connection count data. Applying it to the case study provided a proof-of-concept and indicated that occupancy indicators derived from the analytics can provide a simple, yet more accurate understanding of the building's occupancy schedule, compared to standard schedules. The wide differences between occupancy attributes, such as the peak time, arrival, and departure times, even in spaces of the same type, are clear indicators of the need for building-specific sequences of operations.

Although the case study demonstrated that WiFi can be a low-cost approach to analyze building-specific occupancy patterns with an acceptable level of accuracy, it must be noted that the results are only meant to show a proof of concept of the proposed methodology. Data collection would have to be extended beyond the duration of this case study for at least one year to capture seasonal variations and ensure results accuracy, before making any changes to the control sequences of building systems. Key insights drawn from the case study are summarized as follows.

- Clustering daily WiFi connection count profiles extracted representative day types of occupancy patterns that were significantly different from standard operational schedules. Although the presented results may not identify or predict the exact occupancy counts (due to the duality of WiFi traffic and occupant numbers), they clearly show the deviation between the standard and actual schedules, especially in terms of peak occupancy, arrival, and departure times.
- The high accuracy of the developed prediction models showed that Poisson regression, which was rarely used in this field, improved model performance while still relying only on simple input data (i.e., aggregate WiFi connection counts). This algorithm was capable of learning and predicting the variations in WiFi counts and obtained higher prediction accuracy than the typically used MLR models.
- Although Hour of the day was considered as a feature for developing prediction models, the high discrepancy between WiFi connection counts of working time and nighttime necessitated introducing a new Boolean feature to the model to represent these two states. Using this approach in developing the prediction model, considerably improved the accuracy of

prediction, compared to prediction models developed for working time and nighttime separately.

- The comparison between training a single prediction model to cover all days of the week and having separate prediction models for each subset derived from day types (e.g., weekends/weekdays) showed that although the latter slightly outperforms in predicting typical weeks; the former approach can better capture irregularities of different days in some weeks. Hence, using a single model can be helpful to predict occupancy in atypical weeks (such as exam times in academic buildings or holidays in retail and commercial buildings) which results from being trained by more diverse daily profiles.
- A notable variation was demonstrated in the peak occupancy level, as well as the timing of its occurrence, for different days of the week. This can significantly influence the outdoor air ventilation demand on different days.
- The study of arrival/departure times showed that the diversity of zone-level occupancy patterns can be considerable, which may result in inefficient operation of building systems in unoccupied rooms. Analyzing this diversity can be practical in adjusting zone-level sequences of operation with a high level of confidence, particularly in buildings with more variable occupancy patterns.
- Unlike most of the technologies proposed for occupancy counting, simple data of aggregate WiFi connection counts provided the opportunity of extracting high-level occupancy indicators, such as building occupancy patterns and building peak occupancy time. However, analysis of WiFi data can also result in more detailed insights regarding occupants' behavioral patterns, which requires access to individuals' identifier attributes such as the MAC address.

CHAPTER 6: SUMMARY, CONCLUSION, AND FUTURE WORKS

6.1 Summary of Research

In this study, a review of related studies followed by the current research gaps in the field was presented. Then, the methodology including two modules to bridge some of the identified gaps as well as the results of implementing the proposed frameworks in two case studies as a proof-of-concept were presented. In the literature review, occupancy resolutions, different technologies, and models proposed to obtain occupancy information were discussed. Furthermore, a more in-depth review was performed on studies focusing on occupancy counting using WiFi connection counts for different purposes, including occupancy pattern identification, real-time occupancy estimation, and future occupancy prediction/forecasting. The research methodology, consisting of two main modules, introduced two frameworks to validate the correlation between WiFi connection counts and the actual building occupancy counts and identify key occupancy indicators using WiFi connection count data. Finally, these two frameworks were applied in two different case studies for validation and providing a proof-of-concept.

In the first module, the proposed framework started with investigating the variation of correlation between WiFi connection counts and the actual building occupancy counts, using a continuous stream of ground-truth data. Then, the influential features on this variation and the effectiveness of each of the identified features, including Hour of the day, Day of the week, and occupancy level were studied. Finally, the prediction models were developed that could successfully estimate real-time occupancy counts (i.e., translate WiFi connection counts to occupancy counts) and forecast day-ahead occupancy counts using WiFi connection counts. In the case study, they could achieve an average accuracy (R^2) of 0.97 and 0.87, respectively.

In the second module, the proposed framework started with studying weekly occupancy patterns at the building-level. Then, models were developed to predict the week-ahead occupancy patterns that achieved an average accuracy (R^2_D) of 0.90 in the case study. Analyses were then performed for predicting peak occupancy, identifying its occurrence interval on different days of the week, and finally identifying earliest/latest arrival and departure times at a zone-level which all demonstrated the potential of using WiFi connection count data for extracting occupancy indicators.

6.2 Limitations

Despite the benefits of the proposed approach, some limitations should also be acknowledged and addressed in future work. In the first framework, although a minimum value was concluded as the number of stationary devices since these devices sometimes change into idle mode, the chance exists that this value did not cover all the stationary devices connected to the WiFi network. Furthermore, data collected from the thermal camera-based occupancy counter showed a slight inaccuracy compared to the data collected through manual counting which needs to be further investigated throughout a longer duration of manual counting or employing more accurate cameras such as the optical one. Moreover, the assumption of resetting daily profiles of camera-based occupancy count data to zero every day might have affected the accuracy of this data. In addition, although there were other entrances used by some staff and librarians, it was assumed that the two gateways monitored by two cameras covered all the occupants entering or leaving the library. Besides the features identified as the top influential ones on the variation of conversion factor, this study showed that there might be other features including space type and occupants' behavior in using the space through the day, which need to be investigated in future studies.

In the second framework, WiFi connection count data was used without being calibrated using ground-truth data. Although previous studies as well as the studies in the first module, showed a strong statistically significant correlation between WiFi connection counts and actual occupancy counts, the proposed models should be calibrated with ground-truth data to account for stationary devices and occupants carrying more or less than one WiFi-connected device. The importance of having ground-truth data increases when the proposed methodology for occupancy pattern prediction is applied at a smaller scale such as room-level or zone-level instead of building-level. At these levels, it is more challenging to recognize the number of connections attributed to each zone without having access to the details of each connection, such as the RSSI.

In addition to the limitations in each framework, there were also limitations regarding the entire study. A longer period of data collection could not only help with better identification of training window requirements, but it could capture the impact of seasonality as well as different events on occupancy variations, especially for future occupancy forecasting. However, this was not feasible within this study due to occupancy restrictions that took place as of March 2020 at the

onset of the COVID-19 pandemic. Moreover, the transferability of the developed models to other buildings with different space types, as well as their scalability to other zones, is always a concern that needs to be further validated.

6.3 Research Contributions and Conclusions

Although previous studies employed ground-truth data to validate the use of WiFi connection counts as a proxy for occupancy count with the aim of adapting HVAC systems operation to occupancy variations, there are still challenges regarding this adaptation. First, the limitations of the ground-truth data (including duration, scope, and size) typically led to reporting a single (fixed) value as the conversion factor of WiFi connection counts to actual occupancy counts. Accordingly, the temporal variations in such a conversion factor and the effect of occupancy levels on it remained unknown. Second, in adapting HVAC systems operation to occupancy variations, logistical, cost, and integration challenges remained a key issue.

This study addressed these gaps by developing two frameworks for (i) collecting longitudinal ground-truth data via camera-based occupancy counters for validating the correlation between WiFi connection counts and actual building occupancy counts; This can advance the knowledge of training models to predict the actual occupancy count from the proxy data of WiFi connection counts in order to optimize buildings' system operation based on the actual needs of the building; (ii) analyzing WiFi traffic data to identify key information about buildings' unique occupancy patterns, which can then be used to adjust sequences of operations.

In this regard, the contributions of this study to the literature can be categorized based on two modules of the methodology as follows:

1. Developing a framework for validating the correlation between WiFi connection counts and actual building occupancy counts over a longer duration by using continuous ground-truth data, collected from camera-based occupancy counters;
    1.1. Using clustering, as well as statistical analysis techniques, to firstly identify stationary devices, and then investigate the correlations between WiFi connection counts and occupancy counts;
    1.2. Identifying Hour of the day, Day of the week, and occupancy level as the influencing features affecting this correlation;

1.3. Assessing the importance of these influential features through developing multiple prediction models, using MLR, to estimate real-time occupancy counts;
1.4. Training prediction models through MLR to forecast day-ahead occupancy counts based on WiFi connection count data with comparatively high accuracy;
1.5. Suggesting WiFi connection count as a reliable indicator of occupancy by implementing the proposed methodology in a library building, as a proof-of-concept;
1.6. Demonstrating WiFi connection count potential to accurately predict actual occupancy counts upon addressing the temporal correlations;
1.7. Achieving higher accuracy in the prediction models developed to estimate real-time occupancy counts and forecasting day-ahead occupancy counts compared to the models created in previous studies.

2. Developing a framework for extracting key occupancy indicators relevant to HVAC operation offline, while using WiFi connection count data in existing buildings.
2.1. Offering prediction models that predict occupancy patterns in a longer horizon (i.e., week-ahead) and larger scale (i.e., building-level) with comparatively high accuracy;
2.2. Identifying and capturing dynamisms of peak occupancy occurrence as well as arrival and departure times at the zone-level;
2.3. Showing that Poisson regression can predict occupancy patterns with a higher level of accuracy, outperforming MLR, through implementing the methodology in a case study;
2.4. Introducing more features to differentiate between daytime and nighttime WiFi connection counts which notably improved the performance of prediction models;
2.5. Demonstrating the potential for extracting practical occupancy indicators to adjust zone-level sequences of operations based on typical arrival and departure times, which varied significantly even between rooms of similar functions in the case study building.

6.4 Future Works

Future works will focus on mitigating some of these limitations by extending the data collection period, as well as investigating the influence of space type, space usage, and other potential features on the variation of the WiFi-occupancy conversion factor. Moreover, automated integration between collected data and BIM is suggested to apply and compare the proposed methodology at different spatial resolutions, including room- or zone-level. Finally, developing a comprehensive system is suggested to integrate WiFi connection counts as well as data from supporting technologies into BAS. Through such a system, occupancy information including occupancy patterns, occupancy counts in real-time or future horizons as well as other occupancy indicators of a building will be extracted from the collected data. All these buildings' unique occupancy patterns would be employed to control building systems operation based on the actual demand of the building. The major contribution of this system would be providing sufficient ventilation especially in situations like the COVID-19 pandemic. Furthermore, quantifying energy savings resulting from implementing this system will be investigated. These findings will further highlight the benefits of using the proposed approach to provide actionable information to modify sequences of operation and reduce energy consumption in large commercial and institutional buildings.

REFERENCES

Alishahi, N., Nik-Bakht, M., & Ouf, M. M. (2021). A framework to identify key occupancy indicators for optimizing building operation using WiFi connection count data. *Building and Environment, 200*, 107936. https://doi.org/10.1016/j.buildenv.2021.107936

Amayri, M., Arora, A., Ploix, S., Bandhyopadyay, S., Ngo, Q.-D., & Badarla, V. R. (2016). Estimating occupancy in heterogeneous sensor environment. *Energy and Buildings*, *129*, 46–58. https://doi.org/10.1016/j.enbuild.2016.07.026

Ansanay-Alex, G. (2013). *Estimating Occupancy Using Indoor Carbon Dioxide Concen- trations Only in an Office Building: A Method and Qualitative Assessment,*. 1–8.

Apostolo, G. H., Bernardini, F., Magalhaes, L. C. S., & Muchaluat-Saade, D. C. (2021). A Unified Methodology to Predict Wi-Fi Network Usage in Smart Buildings. *IEEE Access*, *9*, 11455–11469. https://doi.org/10.1109/ACCESS.2020.3048891

Arief-Ang, I. B., Hamilton, M., & Salim, F. D. (2018). A Scalable Room Occupancy Prediction with Transferable Time Series Decomposition of CO_2 Sensor Data. *ACM Transactions on Sensor Networks*, *14*(3–4), 1–28. https://doi.org/10.1145/3217214

Ashouri, A., Newsham, G. R., Shi, Z., & Gunay, H. B. (2019). Day-ahead Prediction of Building Occupancy using WiFi Signals. *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, 1237–1242. https://doi.org/10.1109/COASE.2019.8843224

ASHRAE. (2019). *ANSI/ASHRAE Standard 90.1-2016 Energy Standard for Buildings except Low-Rise Residential Buildings*.

Azam, M., Blayo, M., Venne, J.-S., & Allegue-Martinez, M. (2019). Occupancy Estimation Using Wifi Motion Detection via Supervised Machine Learning Algorithms. *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 1–5. https://doi.org/10.1109/GlobalSIP45357.2019.8969297

Balaji, B., Xu, J., Nwokafor, A., Gupta, R., & Agarwal, Y. (2013). Sentinel: Occupancy based HVAC actuation using existing WiFi infrastructure within commercial buildings. *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems - SenSys '13*, 1–14. https://doi.org/10.1145/2517351.2517370

Capozzoli, A., Piscitelli, M. S., Gorrino, A., Ballarini, I., & Corrado, V. (2017). Data analytics for occupancy pattern learning to reduce the energy consumption of HVAC systems in office buildings. *Sustainable Cities and Society*, *35*, 191–208. https://doi.org/10.1016/j.scs.2017.07.016

Chen, S., Zhang, G., Xia, X., Chen, Y., Setunge, S., & Shi, L. (2021). The impacts of occupant behavior on building energy consumption: A review. *Sustainable Energy Technologies and Assessments*, *45*, 101212. https://doi.org/10.1016/j.seta.2021.101212

Chen, Z., Jiang, C., & Xie, L. (2018). Building occupancy estimation and detection: A review. *Energy and Buildings*, *169*, 260–270. https://doi.org/10.1016/j.enbuild.2018.03.084

Chen, Z., Masood, M. K., & Soh, Y. C. (2016). A fusion framework for occupancy estimation in office buildings based on environmental sensor data. *Energy and Buildings*, *133*, 790–798. https://doi.org/10.1016/j.enbuild.2016.10.030

Chen, Z., & Soh, Y. C. (2017). Comparing occupancy models and data mining approaches for regular occupancy prediction in commercial buildings. *Journal of Building Performance Simulation*, *10*(5–6), 545–553. https://doi.org/10.1080/19401493.2016.1199735

Chen, Z., Xu, J., & Soh, Y. C. (2015). Modeling regular occupancy in commercial buildings using stochastic models. *Energy and Buildings*, *103*, 216–223. https://doi.org/10.1016/j.enbuild.2015.06.009

Chen, Z., Zhao, R., Zhu, Q., Masood, M. K., Soh, Y. C., & Mao, K. (2017). Building Occupancy Estimation with Environmental Sensors via CDBLSTM. *IEEE Transactions on Industrial Electronics*, *64*(12), 9549–9559. https://doi.org/10.1109/TIE.2017.2711530

Chen, Z., Zhu, Q., Masood, M. K., & Soh, Y. C. (2017). Environmental Sensors-Based Occupancy Estimation in Buildings via IHMM-MLR. *IEEE Transactions on Industrial Informatics*, *13*(5), 2184–2193. https://doi.org/10.1109/TII.2017.2668444

Chidurala, V., & Li, X. (2021). Occupancy Estimation Using Thermal Imaging Sensors and Machine Learning Algorithms. *IEEE Sensors Journal*, *21*(6), 8627–8638. https://doi.org/10.1109/JSEN.2021.3049311

Chong, A., Augenbroe, G., & Yan, D. (2021). Occupancy data at different spatial resolutions: Building energy performance and model calibration. *Applied Energy*, *286*, 116492. https://doi.org/10.1016/j.apenergy.2021.116492

Cisco. (2017). *Cisco Prime Infrastructure 3.4 User Guide*. http://www.cisco.com

Cisco, E. (2014). *Cisco Aironet Series 1700/2700/3700 Access Point Deployment Guide*. http://www.cisco.com

Dai, X., Liu, J., & Zhang, X. (2020). A review of studies applying machine learning models to predict occupancy and window-opening behaviours in smart buildings. *Energy and Buildings*, *223*, 110159. https://doi.org/10.1016/j.enbuild.2020.110159

Di Domenico, S., De Sanctis, M., Cianca, E., & Bianchi, G. (2016). A Trained-once Crowd Counting Method Using Differential WiFi Channel State Information. *Proceedings of the 3rd International on Workshop on Physical Analytics - WPA '16*, 37–42. https://doi.org/10.1145/2935651.2935657

Ding, Y., Chen, W., Wei, S., & Yang, F. (2021). An occupancy prediction model for campus buildings based on the diversity of occupancy patterns. *Sustainable Cities and Society*, *64*, 102533. https://doi.org/10.1016/j.scs.2020.102533

D'Oca, S., & Hong, T. (2015). Occupancy schedules learning process through a data mining framework. *Energy and Buildings*, *88*, 395–408. https://doi.org/10.1016/j.enbuild.2014.11.065

Ekwevugbe, T., Brown, N., Pakka, V., & Fan, D. (2017). Improved occupancy monitoring in non-domestic buildings. *Sustainable Cities and Society*, *30*, 97–107. https://doi.org/10.1016/j.scs.2017.01.003

Erickson, V., Carreira-Perpinan, M. A., & Cerpa, A. (2011). *OBSERVE: Occupancy-based system for efficient reduction of HVAC energy*. 258–269.

Esrafilian-Najafabadi, M., & Haghighat, F. (2021). Occupancy-based HVAC control systems in buildings: A state-of-the-art review. *Building and Environment*, *197*, 107810. https://doi.org/10.1016/j.buildenv.2021.107810

Franco, A., & Leccese, F. (2020). Measurement of CO2 concentration for occupancy estimation in educational buildings with energy efficiency purposes. *Journal of Building Engineering*, *32*, 101714. https://doi.org/10.1016/j.jobe.2020.101714

Gruber, M., Trüschel, A., & Dalenbäck, J.-O. (2014). CO 2 sensors for occupancy estimations: Potential in building automation applications. *Energy and Buildings*, *84*, 548–556. https://doi.org/10.1016/j.enbuild.2014.09.002

Gul, M. S., & Patidar, S. (2015). Understanding the energy consumption and occupancy of a multi-purpose academic building. *Energy and Buildings*, *87*, 155–165. https://doi.org/10.1016/j.enbuild.2014.11.027

Gunay, B., Nagy, Z., Miller, C., Ouf, M. M., & Dong, B. (2021). Using Occupant-Centric Control for Commercial HVAC Systems. *ASHRAE Journal*, *63*(5), 30-32,34-36,38-40.

Guo, X., Tiller, D., Henze, G., & Waters, C. (2010). The performance of occupancy-based lighting control systems: A review. *Lighting Research & Technology*, *42*(4), 415–431. https://doi.org/10.1177/1477153510376225

Heinzl, H., & Mittlböck, M. (2003). Pseudo R-squared measures for Poisson regression models with over- or underdispersion. *Computational Statistics & Data Analysis*, *44*(1–2), 253–271. https://doi.org/10.1016/S0167-9473(03)00062-8

Hobson, B. W., Gunay, H. B., Ashouri, A., & Newsham, G. R. (2020). Clustering and motif identification for occupancy-centric control of an air handling unit. *Energy and Buildings*, *223*, 110179. https://doi.org/10.1016/j.enbuild.2020.110179

Hobson, B. W., Lowcay, D., Gunay, H. B., Ashouri, A., & Newsham, G. R. (2019). Opportunistic occupancy-count estimation using sensor fusion: A case study. *Building and Environment*, *159*, 106154. https://doi.org/10.1016/j.buildenv.2019.05.032

Hong, T., Yan, D., D'Oca, S., & Chen, C. (2017). Ten questions concerning occupant behavior in buildings: The big picture. *Building and Environment*, *114*, 518–530. https://doi.org/10.1016/j.buildenv.2016.12.006

Hou, H., Pawlak, J., Sivakumar, A., Howard, B., & Polak, J. (2020). An approach for building occupancy modelling considering the urban context. *Building and Environment*, *183*, 107126. https://doi.org/10.1016/j.buildenv.2020.107126

Jagadeesh Simma, K. C., Mammoli, A., & Bogus, S. M. (2019). Real-Time Occupancy Estimation Using WiFi Network to Optimize HVAC Operation. *Procedia Computer Science*, *155*, 495–502. https://doi.org/10.1016/j.procs.2019.08.069

Jiang, C., Chen, Z., Su, R., Masood, M. K., & Soh, Y. C. (2020). Bayesian filtering for building occupancy estimation from carbon dioxide concentration. *Energy and Buildings*, *206*, 109566. https://doi.org/10.1016/j.enbuild.2019.109566

Jiang, C., Masood, M. K., Soh, Y. C., & Li, H. (2016). Indoor occupancy estimation from carbon dioxide concentration. *Energy and Buildings*, *131*, 132–141. https://doi.org/10.1016/j.enbuild.2016.09.002

Jin, M., Bekiaris-Liberis, N., Weekly, K., Spanos, C. J., & Bayen, A. M. (2018). Occupancy Detection via Environmental Sensing. *IEEE Transactions on Automation Science and Engineering*, *15*(2), 443–455. https://doi.org/10.1109/TASE.2016.2619720

Jung, W., & Jazizadeh, F. (2019). Human-in-the-loop HVAC operations: A quantitative review on occupancy, comfort, and energy-efficiency dimensions. *Applied Energy*, *239*, 1471–1508. https://doi.org/10.1016/j.apenergy.2019.01.070

Kim, S., Sung, Y., Sung, Y., & Seo, D. (2019). Development of a Consecutive Occupancy Estimation Framework for Improving the Energy Demand Prediction Performance of Building Energy Modeling Tools. *Energies*, *12*(3), 433. https://doi.org/10.3390/en12030433

Li, N., Calis, G., & Becerik-Gerber, B. (2012). Measuring and monitoring occupancy with an RFID based system for demand-driven HVAC operations. *Automation in Construction*, *24*, 89–99. https://doi.org/10.1016/j.autcon.2012.02.013

Li, Z., & Dong, B. (2018). Short term predictions of occupancy in commercial buildings—Performance analysis for stochastic models and machine learning approaches. *Energy and Buildings*, *158*, 268–281. https://doi.org/10.1016/j.enbuild.2017.09.052

Liang, X., Hong, T., & Shen, G. Q. (2016). Occupancy data analytics and prediction: A case study. *Building and Environment*, *102*, 179–192. https://doi.org/10.1016/j.buildenv.2016.03.027

Liu, D., Guan, X., Du, Y., & Zhao, Q. (2013). Measuring indoor occupancy in intelligent buildings using the fusion of vision sensors. *Measurement Science and Technology*, *24*(7), 074023. https://doi.org/10.1088/0957-0233/24/7/074023

Longo, E., Redondi, A. E. C., & Cesana, M. (2019). Accurate occupancy estimation with WiFi and bluetooth/BLE packet capture. *Computer Networks*, *163*, 106876. https://doi.org/10.1016/j.comnet.2019.106876

Mahdavi, A., & Tahmasebi, F. (2015). Predicting people's presence in buildings: An empirically based model performance analysis. *Energy and Buildings*, *86*, 349–355. https://doi.org/10.1016/j.enbuild.2014.10.027

Masood, M. K., Chai Soh, Y., & Chang, V. W.-C. (2015). Real-time occupancy estimation using environmental parameters. *2015 International Joint Conference on Neural Networks (IJCNN)*, 1–8. https://doi.org/10.1109/IJCNN.2015.7280781

Masood, M. K., Soh, Y. C., & Jiang, C. (2017). Occupancy estimation from environmental parameters using wrapper and hybrid feature selection. *Applied Soft Computing*, *60*, 482–494. https://doi.org/10.1016/j.asoc.2017.07.003

Melfi, R., Rosenblum, B., Nordman, B., & Christensen, K. (2011). Measuring building occupancy using existing network infrastructure. *2011 International Green Computing Conference and Workshops*, 1–8. https://doi.org/10.1109/IGCC.2011.6008560

Meng, Y., Li, T., Liu, G., Xu, S., & Ji, T. (2020). Real-time dynamic estimation of occupancy load and an air-conditioning predictive control method based on image information fusion. *Building and Environment*, *173*, 106741. https://doi.org/10.1016/j.buildenv.2020.106741

Mohottige, I. P., Sutjarittham, T., Raju, N., Gharakheili, H. H., & Sivaraman, V. (2018). Role of Campus WiFi Infrastructure for Occupancy Monitoring in a Large University. *2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*, 1–5. https://doi.org/10.1109/ICIAFS.2018.8913341

Nassif, N. (2012). A robust CO2-based demand-controlled ventilation control strategy for multi-zone HVAC systems. *Energy and Buildings*, *45*, 72–81. https://doi.org/10.1016/j.enbuild.2011.10.018

Natural Resources Canada (NRCan). (2018). *Energy Efficiency Trends Analysis Tables*. https://oee.nrcan.gc.ca/corporate/statistics/neud/dpa/data_e/databases.cfm?attr=0

Naylor, S., Gillott, M., & Lau, T. (2018). A review of occupant-centric building control strategies to reduce building energy use. *Renewable and Sustainable Energy Reviews*, *96*, 1–10. https://doi.org/10.1016/j.rser.2018.07.019

NECB. (2015). *National Energy Code of Canada for Buildings*.

O'Brien, W., Wagner, A., Schweiker, M., Mahdavi, A., Day, J., Kjærgaard, M. B., Carlucci, S., Dong, B., Tahmasebi, F., Yan, D., Hong, T., Gunay, H. B., Nagy, Z., Miller, C., & Berger, C. (2020). Introducing IEA EBC annex 79: Key challenges and opportunities in the field of occupant-centric building design and operation. *Building and Environment*, *178*, 106738. https://doi.org/10.1016/j.buildenv.2020.106738

Oppermann, M., & Munzner, T. (2020). Ocupado: Visualizing Location-Based Counts Over Time Across Buildings. *Computer Graphics Forum*, *39*(3), 127–138. https://doi.org/10.1111/cgf.13968

Ouf, M. M., Issa, M. H., Azzouz, A., & Sadick, A.-M. (2017). Effectiveness of using WiFi technologies to detect and predict building occupancy. *Sustainable Buildings*, *2*, 7. https://doi.org/10.1051/sbuild/2017005

Ouf, M. M., O'Brien, W., & Gunay, B. (2019). On quantifying building performance adaptability to variable occupancy. *Building and Environment*, *155*, 257–267. https://doi.org/10.1016/j.buildenv.2019.03.048

Page, J., Robinson, D., Morel, N., & Scartezzini, J.-L. (2008). A generalised stochastic model for the simulation of occupant presence. *Energy and Buildings*, *40*(2), 83–98. https://doi.org/10.1016/j.enbuild.2007.01.018

Park, J. Y., Ouf, M. M., Gunay, B., Peng, Y., O'Brien, W., Kjærgaard, M. B., & Nagy, Z. (2019). A critical review of field implementations of occupant-centric building controls. *Building and Environment*, *165*, 106351. https://doi.org/10.1016/j.buildenv.2019.106351

Pasquel Mohottige, I., Gharakheili, H. H., Vishwanath, A., Kanhere, S. S., & Sivaraman, V. (2020). Evaluating Emergency Evacuation Events Using Building WiFi Data. *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*, 116–127. https://doi.org/10.1109/IoTDI49375.2020.00018

Peng, Y., Rysanek, A., Nagy, Z., & Schlüter, A. (2017). Occupancy learning-based demand-driven cooling control for office spaces. *Building and Environment*, *122*, 145–160. https://doi.org/10.1016/j.buildenv.2017.06.010

Peng, Y., Rysanek, A., Nagy, Z., & Schlüter, A. (2018). Using machine learning techniques for occupancy-prediction-based cooling control in office buildings. *Applied Energy*, *211*, 1343–1358. https://doi.org/10.1016/j.apenergy.2017.12.002

Petersen, S., Pedersen, T. H., Nielsen, K. U., & Knudsen, M. D. (2016). Establishing an image-based ground truth for validation of sensor data-based room occupancy detection. *Energy and Buildings*, *130*, 787–793. https://doi.org/10.1016/j.enbuild.2016.09.009

Rafsanjani, H. N., & Ghahramani, A. (2019). Extracting occupants' energy-use patterns from Wi-Fi networks in office buildings. *Journal of Building Engineering*, *26*, 100864. https://doi.org/10.1016/j.jobe.2019.100864

Ravi, A., & Misra, A. (2021). Practical server-side WiFi-based indoor localization: Addressing cardinality & outlier challenges for improved occupancy estimation. *Ad Hoc Networks*, *115*, 102443. https://doi.org/10.1016/j.adhoc.2021.102443

Raykov, Y. P., Ozer, E., Dasika, G., Boukouvalas, A., & Little, M. A. (2016). Predicting room occupancy with a single passive infrared (PIR) sensor through behavior extraction. *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 1016–1027. https://doi.org/10.1145/2971648.2971746

Rueda, L., Agbossou, K., Cardenas, A., Henao, N., & Kelouwani, S. (2020). A comprehensive review of approaches to building occupancy detection. *Building and Environment*, *180*, 106966. https://doi.org/10.1016/j.buildenv.2020.106966

Salimi, S., & Hammad, A. (2019). Critical review and research roadmap of office building energy management based on occupancy monitoring. *Energy and Buildings*, *182*, 214–241. https://doi.org/10.1016/j.enbuild.2018.10.007

Salimi, S., Liu, Z., & Hammad, A. (2019). Occupancy prediction model for open-plan offices using real-time location system and inhomogeneous Markov chain. *Building and Environment*, *152*, 1–16. https://doi.org/10.1016/j.buildenv.2019.01.052

Seghezzi, E., Locatelli, M., Pellegrini, L., Pattini, G., Di Giuda, G. M., Tagliabue, L. C., & Boella, G. (2021). Towards an Occupancy-Oriented Digital Twin for Facility Management: Test Campaign and Sensors Assessment. *Applied Sciences*, *11*(7), 3108. https://doi.org/10.3390/app11073108

Shen, W., Newsham, G., & Gunay, B. (2017). Leveraging existing occupancy-related data for optimal control of commercial office buildings: A review. *Advanced Engineering Informatics*, *33*, 230–242. https://doi.org/10.1016/j.aei.2016.12.008

Shetty, S. S., Chinh, H. D., Gupta, M., & Panda, S. K. (2017). User Presence Estimation in Multi-Occupancy Rooms Using Plug-Load Meters and PIR Sensors. *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 1–6. https://doi.org/10.1109/GLOCOM.2017.8255036

Sobron, I., Del Ser, J., Eizmendi, I., & Velez, M. (2018). Device-Free People Counting in IoT Environments: New Insights, Results, and Open Challenges. *IEEE Internet of Things Journal*, *5*(6), 4396–4408. https://doi.org/10.1109/JIOT.2018.2806990

Sun, K., Zhao, Q., & Zou, J. (2020). A review of building occupancy measurement systems. *Energy and Buildings*, *216*, 109965. https://doi.org/10.1016/j.enbuild.2020.109965

Szczurek, A., Maciejewska, M., & Pietrucha, T. (2017). Occupancy determination based on time series of CO2 concentration, temperature and relative humidity. *Energy and Buildings*, *147*, 142–154. https://doi.org/10.1016/j.enbuild.2017.04.080

Teixeira, Thiago, Dublon, Gershon, & Savvides, Andreas. (2010). *A Survey of Human-Sensing: Methods for Detecting Presence, Count, Location, Track, and Identity*. *5*.

Tekler, Z. D., Low, R., Gunay, B., Andersen, R. K., & Blessing, L. (2020). A scalable Bluetooth Low Energy approach to identify occupancy patterns and profiles in office spaces. *Building and Environment*, *171*, 106681. https://doi.org/10.1016/j.buildenv.2020.106681

Wahl, F., Milenkovic, M., & Amft, O. (2012). A Distributed PIR-based Approach for Estimating People Count in Office Environments. *2012 IEEE 15th International Conference on Computational Science and Engineering*, 640–647. https://doi.org/10.1109/ICCSE.2012.92

Wang, F., Feng, Q., Chen, Z., Zhao, Q., Cheng, Z., Zou, J., Zhang, Y., Mai, J., Li, Y., & Reeve, H. (2017). Predictive control of indoor environment using occupant number detected by video data and CO 2 concentration. *Energy and Buildings*, *145*, 155–162. https://doi.org/10.1016/j.enbuild.2017.04.014

Wang, H.-T., Jia, Q.-S., Song, C., Yuan, R., & Guan, X. (2014). Building occupant level estimation based on heterogeneous information fusion. *Information Sciences*, *272*, 145–157. https://doi.org/10.1016/j.ins.2014.02.080

Wang, W., Chen, J., & Hong, T. (2018a). Modeling occupancy distribution in large spaces with multi-feature classification algorithm. *Building and Environment*, *137*, 108–117. https://doi.org/10.1016/j.buildenv.2018.04.002

Wang, W., Chen, J., & Hong, T. (2018b). Occupancy prediction through machine learning and data fusion of environmental sensing and Wi-Fi sensing in buildings. *Automation in Construction*, *94*, 233–243. https://doi.org/10.1016/j.autcon.2018.07.007

Wang, W., Chen, J., Hong, T., & Zhu, N. (2018). Occupancy prediction through Markov based feedback recurrent neural network (M-FRNN) algorithm with WiFi probe technology. *Building and Environment*, *138*, 160–170. https://doi.org/10.1016/j.buildenv.2018.04.034

Wang, W., Chen, J., & Song, X. (2017). Modeling and predicting occupancy profile in office space with a Wi-Fi probe-based Dynamic Markov Time-Window Inference approach. *Building and Environment*, *124*, 130–142. https://doi.org/10.1016/j.buildenv.2017.08.003

Wang, W., Hong, T., Li, N., Wang, R. Q., & Chen, J. (2019). Linking energy-cyber-physical systems with occupancy prediction and interpretation through WiFi probe-based ensemble classification. *Applied Energy*, *236*, 55–69. https://doi.org/10.1016/j.apenergy.2018.11.079

Wang, W., Hong, T., Xu, N., Xu, X., Chen, J., & Shan, X. (2019). Cross-source sensing data fusion for building occupancy prediction with adaptive lasso feature filtering. *Building and Environment*, *162*, 106280. https://doi.org/10.1016/j.buildenv.2019.106280

Wang, W., Wang, J., Chen, J., Huang, G., & Guo, X. (2018). Multi-zone outdoor air coordination through Wi-Fi probe-based occupancy sensing. *Energy and Buildings*, *159*, 495–507. https://doi.org/10.1016/j.enbuild.2017.11.041

Wang, Y., & Shao, L. (2018). Understanding occupancy and user behaviour through Wi-Fi-based indoor positioning. *Building Research & Information*, *46*(7), 725–737. https://doi.org/10.1080/09613218.2018.1378498

Wang, Z., Hong, T., Piette, M. A., & Pritoni, M. (2019). Inferring occupant counts from Wi-Fi data in buildings through machine learning. *Building and Environment*, *158*, 281–294. https://doi.org/10.1016/j.buildenv.2019.05.015

Weekly, K., Jin, M., Zou, H., Hsu, C., Soyza, C., Bayen, A., & Spanos, C. (2018). Building-in-Briefcase: A Rapidly-Deployable Environmental Sensor Suite for the Smart Building. *Sensors*, *18*(5), 1381. https://doi.org/10.3390/s18051381

Wohlin, C., Höst, M., Per, R., & Wesslén, A. (2003). Software Reliability In R. Meyers (Eds.). In *Encyclopedia of Physical Science and Technology* (Third, pp. 25–39). Academic Press. https://doi.org/10.1016/B0-12-227410-5/00858-9

Wu, L., & Wang, Y. (2019). A Low-Power Electric-Mechanical Driving Approach for True Occupancy Detection Using a Shuttered Passive Infrared Sensor. *IEEE Sensors Journal*, *19*(1), 47–57. https://doi.org/10.1109/JSEN.2018.2875659

Yang, J., Santamouris, M., & Lee, S. E. (2016). Review of occupancy sensing systems and occupancy modeling methodologies for the application in institutional buildings. *Energy and Buildings*, *121*, 344–349. https://doi.org/10.1016/j.enbuild.2015.12.019

Yang, Z., & Becerik-Gerber, B. (2014). The coupled effects of personalized occupancy profile based HVAC schedules and room reassignment on building energy use. *Energy and Buildings*, *78*, 113–122. https://doi.org/10.1016/j.enbuild.2014.04.002

Yang, Z., Li, N., Becerik-Gerber, B., & Orosz, M. (2014). A systematic approach to occupancy modeling in ambient sensor-rich buildings. *SIMULATION*, *90*(8), 960–977. https://doi.org/10.1177/0037549713489918

Yoo, W., Kim, H., & Shin, M. (2020). Stations-oriented indoor localization (SOIL): A BIM-Based occupancy schedule modeling system. *Building and Environment*, *168*, 106520. https://doi.org/10.1016/j.buildenv.2019.106520

Zou, H., Jiang, H., Yang, J., Xie, L., & Spanos, C. (2017). Non-intrusive occupancy sensing in commercial buildings. *Energy and Buildings*, *154*, 633–643. https://doi.org/10.1016/j.enbuild.2017.08.045

Zou, H., Zhou, Y., Yang, J., & Spanos, C. J. (2018). Device-free occupancy detection and crowd counting in smart buildings with WiFi-enabled IoT. *Energy and Buildings*, *174*, 309–322. https://doi.org/10.1016/j.enbuild.2018.06.040

Zuraimi, M. S., Pantazaras, A., Chaturvedi, K. A., Yang, J. J., Tham, K. W., & Lee, S. E. (2017). Predicting occupancy counts using physical and statistical Co2-based modeling methodologies. *Building and Environment*, *123*, 517–528. https://doi.org/10.1016/j.buildenv.2017.07.027

APPENDICES

Appendix A – Python code for real-time occupancy estimation OR day-ahead occupancy prediction/forecasting

A-1. Function for cross-validating prediction models

```
# This function is similar for realtime occupancy estimation as well as
future occupancy prediction. The difference is in the dataset used for
training

def lb_occupancy_prediction(x_train, y_train, n):

  kf = KFold(n_splits = n)

# defining lists to keep the results of each fold
  rmses = []
  mapes = []
  r2s = []
  residual = []
  predictions = []

# dividing data into n folds and train and test the model on each fold
  for train_index, test_index in kf.split(x_train):
      x_tr = x_train.iloc[train_index]
      x_te = x_train.iloc[test_index]
      y_tr = y_train.iloc[train_index]
      y_te = y_train.iloc[test_index]

      regression = linear_model.LinearRegression()
      regression.fit(x_tr, y_tr)

      prediction = regression.predict(x_te)
      predictions.append(prediction)

# calculating r2 of each fold
      r2 = regression.score(x_te, y_te)
      r2s.append(r2)

# calculating rmse of each fold
      rmse = mean_squared_error(y_te, prediction, squared=False)
      rmses.append(rmse)

# calculating mape of each fold
      mape = mean_absolute_percentage_error(y_te, prediction)
      mapes.append(mape)
```

```
# calculating residuals of each fold
      pred_vs_act = pd.concat([y_te,
                               pd.DataFrame(prediction,
                                            index=y_te.index)], axis=1)
      pred_vs_act.columns = ["Actual", "Predicted"]
      resid = pred_vs_act["Predicted"] - pred_vs_act["Actual"]
      residual.append(resid)

# plotting the prediction vs actual values for each fold
      pred_vs_act.plot(figsize=(20,7))
      plt.xlabel("Time")
      plt.ylabel("Clients")

# printing the results of each fold and also the average of the results
for each metrics
  dates = sorted(list(set(y_train.index.date)))
  r2s = pd.DataFrame(r2s, index = dates)
  print("\n\nr2:", r2s)
  print("\n\nAve r2:", np.mean(r2s))
  rmses = pd.DataFrame(rmses, index = dates)
  print("\n\nrmse:", rmses)
  print("\n\nAve rmse:", round(np.mean(rmses),2))
  mapes = pd.DataFrame(mapes, index = dates)
  print("\n\nmape:", mapes)
  print("\n\nAve mape:", round(np.mean(mapes),2))
  print("\n\nAve residual:", np.round(np.mean(list(flatten(residual))),2))
  print("\n\nStd residual:", np.round(np.std(list(flatten(residual))),2))

# calculating the R2 values for each day of the week
  r2s.columns = ["r2"]
  r2s['Weekday'] = r2s.index.map(lambda t: t.weekday())
  print("\n\nR2 values for each day of the week",
        r2s.groupby(by = 'Weekday').mean())

# calculating the RMSE values for each day of the week
  rmses.columns = ["RMSE"]
  rmses['Weekday'] = rmses.index.map(lambda t: t.weekday())
  print("\n\nRMSE values for each day of the week",
        rmses.groupby(by = 'Weekday').mean())

# calculating the MAPE values for each day of the week
  mapes.columns = ["MAPE"]
  mapes['Weekday'] = mapes.index.map(lambda t: t.weekday())
  print("\n\nMAPE values for each day of the week",
        mapes.groupby(by = 'Weekday').mean())
```

A-2. Data preparation for real-time occupancy counts estimation

```
# concatenating camera and shifted wifi count data
cam_shwifi_lib = pd.concat([camera_lib_h.drop(["Time"], axis = 1),
                            wifi_lib_h_shifted.drop(["Time"], axis = 1)],
                            axis=1).sort_index()

# adding a new column "adj_occ_wifi" for keeping the wifi counts after
deducting stationary devices from them
cam_shwifi_lib["adj_occ_wifi"] = cam_shwifi_lib["occ_wifi_1h_shifted_backw
ard"]
cam_shwifi_lib['Weekday'] = cam_shwifi_lib.index.map(lambda t: t.date().we
ekday())

for row in cam_shwifi_lib.index:
  if ((cam_shwifi_lib["Weekday"][row] == 5) |
      (cam_shwifi_lib["Weekday"][row] == 6)):
    cam_shwifi_lib.loc[row,"adj_occ_wifi"] = cam_shwifi_lib.loc[row,"adj_o
cc_wifi"] - 14
# deducting 14 from wifi counts on weekends
  else:
    cam_shwifi_lib.loc[row,"adj_occ_wifi"] = cam_shwifi_lib.loc[row,"adj_o
cc_wifi"] - 17
# deducting 17 from wifi counts on weekends

# replacing camera counts value for those hours that wifi counts become
less than zero (after deducting stationary devices)
for row in cam_shwifi_lib.index:
  if (cam_shwifi_lib["adj_occ_wifi"][row] <= 0):
    cam_shwifi_lib.loc[row,"adj_occ_wifi"] = cam_shwifi_lib.loc[row,"occ_c
am"]

# normalizing wifi counts
scaler = MinMaxScaler()
scaled = scaler.fit_transform(cam_shwifi_lib[["adj_occ_wifi"]])
scaled = pd.DataFrame(scaled, index = cam_shwifi_lib.index,
                      columns = ["adj_occ_wifi"])
cam_shwifi_lib[["adj_occ_wifi"]] = scaled

# adding required attributes
cam_shwifi_lib['Date'] = cam_shwifi_lib.index.map(lambda t: t.date())
cam_shwifi_lib['hour'] = cam_shwifi_lib.index.map(lambda t: t.time().hour)
cam_shwifi_lib['Weekday'] = cam_shwifi_lib.index.map(lambda t: t.date().we
ekday())
cam_shwifi_lib['Time'] = cam_shwifi_lib.index.map(lambda t: t.time())
```

```
# selecting daytime period (9 am to 10 pm)
for row in cam_shwifi_lib.index:
  if (cam_shwifi_lib['hour'][row] < 9):
    cam_shwifi_lib = cam_shwifi_lib.drop(index = row)
  elif (cam_shwifi_lib['hour'][row] > 22):
    cam_shwifi_lib = cam_shwifi_lib.drop(index = row)


#spliting dataset into weekdays/weekends
cam_shwifi_lib_weekdays = cam_shwifi_lib[
      (cam_shwifi_lib.index.map(lambda t: t.date().weekday()) == 0) |
      (cam_shwifi_lib.index.map(lambda t: t.date().weekday()) == 1) |
      (cam_shwifi_lib.index.map(lambda t: t.date().weekday()) == 2) |
      (cam_shwifi_lib.index.map(lambda t: t.date().weekday()) == 3) |
      (cam_shwifi_lib.index.map(lambda t: t.date().weekday()) == 4) ]

cam_shwifi_lib_weekends = cam_shwifi_lib[
      (cam_shwifi_lib.index.map(lambda t: t.date().weekday()) == 5) |
      (cam_shwifi_lib.index.map(lambda t: t.date().weekday()) == 6) ]

cam_shwifi_lib
```

A-3. Real-time occupancy counts estimation (Model set 1)

```
# Weekdays' model

# Splitting dataset into predictors and response features
x_train = pd.concat([pd.get_dummies(cam_shwifi_lib_weekdays.index.hour),
                     pd.DataFrame(cam_shwifi_lib_weekdays.index)
                     ], axis=1)

x_train.columns = ['h9', 'h10', 'h11', 'h12', 'h13', 'h14', 'h15',
                   'h16', 'h17', 'h18', 'h19', 'h20', 'h21', 'h22',
                   'datetime']

x_train.set_index('datetime', inplace = True)

y_train = cam_shwifi_lib_weekdays["occ_cam"]


# Calling occupancy prediction function with 44 fold cross-validation
lb_occupancy_prediction(x_train, y_train, 44)

est = sm.OLS(y_train.values.reshape(-1,1), sm.add_constant(x_train))
est2 = est.fit()
print(est2.summary())


# Weekends' model

# Splitting dataset into predictors and response features
x_train = pd.concat([pd.get_dummies(cam_shwifi_lib_weekends.index.hour),
                     pd.DataFrame(cam_shwifi_lib_weekends.index)
                     ], axis=1)

x_train.columns = ['h9', 'h10', 'h11', 'h12', 'h13', 'h14', 'h15',
                   'h16', 'h17', 'h18', 'h19', 'h20', 'h21', 'h22',
                   'datetime']

x_train.set_index('datetime', inplace = True)

y_train = cam_shwifi_lib_weekends["occ_cam"]

# Calling occupancy prediction function with 16 fold cross-validation
lb_occupancy_prediction(x_train, y_train, 16)

est = sm.OLS(y_train.values.reshape(-1,1), sm.add_constant(x_train))
est2 = est.fit()
print(est2.summary())
```

A-4. Real-time occupancy counts estimation (Model set 2)

```
# Weekdays' model

# Splitting dataset into predictors and response features
x_train = pd.concat([pd.get_dummies(cam_shwifi_lib_weekdays.index.hour),
                  pd.get_dummies(cam_shwifi_lib_weekdays.index.dayofweek),
                     pd.DataFrame(cam_shwifi_lib_weekdays.index)
                     ], axis=1)

x_train.columns = ['h9', 'h10', 'h11', 'h12', 'h13', 'h14', 'h15',
                   'h16', 'h17', 'h18', 'h19', 'h20', 'h21', 'h22',
                   'mon', 'tue', 'wed', 'thu', 'fri',
                   'datetime']

x_train.set_index('datetime', inplace = True)

y_train = cam_shwifi_lib_weekdays["occ_cam"]

# Calling occupancy prediction function with 44 fold cross-validation
lb_occupancy_prediction(x_train, y_train, 44)

est = sm.OLS(y_train.values.reshape(-1,1), sm.add_constant(x_train))
est2 = est.fit()
print(est2.summary())

# Weekends' model

# Splitting dataset into predictors and response features
x_train = pd.concat([pd.get_dummies(cam_shwifi_lib_weekends.index.hour),
                 pd.get_dummies(cam_shwifi_lib_weekends.index.dayofweek,
                                drop_first=True),
                     pd.DataFrame(cam_shwifi_lib_weekends.index)
                     ], axis=1)

x_train.columns = ['h9', 'h10', 'h11', 'h12', 'h13', 'h14', 'h15',
                   'h16', 'h17', 'h18', 'h19', 'h20', 'h21', 'h22',
                   'sat/sun',
                   'datetime']

x_train.set_index('datetime', inplace = True)

y_train = cam_shwifi_lib_weekends["occ_cam"]


# Calling occupancy prediction function with 16 fold cross-validation
lb_occupancy_prediction(x_train, y_train, 16)

est = sm.OLS(y_train.values.reshape(-1,1), sm.add_constant(x_train))
est2 = est.fit()
print(est2.summary())
```

A-5. Real-time occupancy counts estimation (Model set 3)

```
# Weekdays' model

# Splitting dataset into predictors and response features
x_train = pd.concat(
      [cam_shwifi_lib_weekdays.reset_index()['adj_occ_wifi'],
       pd.DataFrame(cam_shwifi_lib_weekdays.index)], axis=1)

x_train.columns = ['wifi', 'datetime']

x_train.set_index('datetime', inplace = True)

y_train = cam_shwifi_lib_weekdays["occ_cam"]

# Calling occupancy prediction function with 44 fold cross-validation
lb_occupancy_prediction(x_train, y_train, 44)

est = sm.OLS(y_train.values.reshape(-1,1), sm.add_constant(x_train))
est2 = est.fit()
print(est2.summary())

# Weekends' model

# Splitting dataset into predictors and response features
x_train = pd.concat(
      [cam_shwifi_lib_weekends.reset_index()['adj_occ_wifi'],
      pd.DataFrame(cam_shwifi_lib_weekends.index)], axis=1)

x_train.columns = ['wifi', 'datetime']

x_train.set_index('datetime', inplace = True)

y_train = cam_shwifi_lib_weekends["occ_cam"]

# Calling occupancy prediction function with 16 fold cross-validation
lb_occupancy_prediction(x_train, y_train, 16)

est = sm.OLS(y_train.values.reshape(-1,1), sm.add_constant(x_train))
est2 = est.fit()
print(est2.summary())
```

A-6. Real-time occupancy counts estimation (Model set 4)

```
# Weekdays' model

# Splitting dataset into predictors and response features
x_train = pd.concat([pd.get_dummies(cam_shwifi_lib_weekdays.index.hour),
                     cam_shwifi_lib_weekdays.reset_index()['adj_occ_wifi'],
                       pd.DataFrame(cam_shwifi_lib_weekdays.index)
                       ], axis=1)

x_train.columns = ['h9', 'h10', 'h11', 'h12', 'h13', 'h14', 'h15',
                   'h16', 'h17', 'h18', 'h19', 'h20', 'h21', 'h22',
                   'wifi', 'datetime']

x_train.set_index('datetime', inplace = True)

y_train = cam_shwifi_lib_weekdays["occ_cam"]

# Calling occupancy prediction function with 44 fold cross-validation
lb_occupancy_prediction(x_train, y_train, 44)

est = sm.OLS(y_train.values.reshape(-1,1), sm.add_constant(x_train))
est2 = est.fit()
print(est2.summary())

# Weekends' model

# Splitting dataset into predictors and response features
x_train = pd.concat([pd.get_dummies(cam_shwifi_lib_weekends.index.hour),
                     cam_shwifi_lib_weekends.reset_index()['adj_occ_wifi'],
                       pd.DataFrame(cam_shwifi_lib_weekends.index)
                       ], axis=1)

x_train.columns = ['h9', 'h10', 'h11', 'h12', 'h13', 'h14', 'h15',
                   'h16', 'h17', 'h18', 'h19', 'h20', 'h21', 'h22',
                   'wifi', 'datetime']

x_train.set_index('datetime', inplace = True)

y_train = cam_shwifi_lib_weekends["occ_cam"]

# Calling occupancy prediction function with 16 fold cross-validation
lb_occupancy_prediction(x_train, y_train, 16)

est = sm.OLS(y_train.values.reshape(-1,1), sm.add_constant(x_train))
est2 = est.fit()
print(est2.summary())
```

A-7. Real-time occupancy counts estimation (Model set 5)

```
# Weekdays' model

# Splitting dataset into predictors and response features
x_train = pd.concat([pd.get_dummies(cam_shwifi_lib_weekdays.index.hour),
                     pd.get_dummies(cam_shwifi_lib_weekdays.index.dayofweek),
                     cam_shwifi_lib_weekdays.reset_index()['adj_occ_wifi'],
                        pd.DataFrame(cam_shwifi_lib_weekdays.index)
                        ], axis=1)

x_train.columns = ['h9', 'h10', 'h11', 'h12', 'h13', 'h14', 'h15',
                   'h16', 'h17', 'h18', 'h19', 'h20', 'h21', 'h22',
                   'mon', 'tue', 'wed', 'thu', 'fri',
                   'wifi', 'datetime']

x_train.set_index('datetime', inplace = True)

y_train = cam_shwifi_lib_weekdays["occ_cam"]

# Calling occupancy prediction function with 44 fold cross-validation
lb_occupancy_prediction(x_train, y_train, 44)

est = sm.OLS(y_train.values.reshape(-1,1), sm.add_constant(x_train))
est2 = est.fit()
print(est2.summary())

# Weekends' model

# Splitting dataset into predictors and response features
x_train = pd.concat([pd.get_dummies(cam_shwifi_lib_weekends.index.hour),
                     pd.get_dummies(cam_shwifi_lib_weekends.index.dayofweek,
                                    drop_first=True),
                     cam_shwifi_lib_weekends.reset_index()['adj_occ_wifi'],
                        pd.DataFrame(cam_shwifi_lib_weekends.index)
                        ], axis=1)

x_train.columns = ['h9', 'h10', 'h11', 'h12', 'h13', 'h14', 'h15',
                   'h16', 'h17', 'h18', 'h19', 'h20', 'h21', 'h22',
                   'sat/sun',
                   'wifi', 'datetime']

x_train.set_index('datetime', inplace = True)

y_train = cam_shwifi_lib_weekends["occ_cam"]

# Calling occupancy prediction function with 16 fold cross-validation
lb_occupancy_prediction(x_train, y_train, 16)

est = sm.OLS(y_train.values.reshape(-1,1), sm.add_constant(x_train))
est2 = est.fit()
print(est2.summary())
```

A-8. Data preparation for day-ahead occupancy counts prediction/forecasting

```
# concatenating camera and shifted wifi count data
cam_shwifi_lib = pd.concat([camera_lib_h.drop(["Time"], axis = 1),
                            wifi_lib_h_shifted.drop(["Time"], axis = 1)],
                            axis=1).sort_index()

# adding a new column "adj_occ_wifi" for keeping the wifi counts after ded
ucting stationary devices from them
cam_shwifi_lib["adj_occ_wifi"] = cam_shwifi_lib["occ_wifi_1h_shifted_backw
ard"]
cam_shwifi_lib['Weekday'] = cam_shwifi_lib.index.map(lambda t: t.date().we
ekday())

for row in cam_shwifi_lib.index:
  if ((cam_shwifi_lib["Weekday"][row] == 5) |
      (cam_shwifi_lib["Weekday"][row] == 6)):
    cam_shwifi_lib.loc[row,"adj_occ_wifi"] = cam_shwifi_lib.loc[row,"adj_o
cc_wifi"] - 14
# subtracting 14 from wifi counts on weekends
  else:
    cam_shwifi_lib.loc[row,"adj_occ_wifi"] = cam_shwifi_lib.loc[row,"adj_o
cc_wifi"] - 17
# subtracting 17 from wifi counts on weekends

# replacing camera counts value for those hours that wifi counts become le
ss than zero (after deducting stationary devices)
for row in cam_shwifi_lib.index:
  if (cam_shwifi_lib["adj_occ_wifi"][row] <= 0):
    cam_shwifi_lib.loc[row,"adj_occ_wifi"] = cam_shwifi_lib.loc[row,"occ_c
am"]

# removing week no.7
cam_shwifi_lib = cam_shwifi_lib[((cam_shwifi_lib.index.weekofyear)-2)!=7]

# shifting wifi counts backward for one day since wifi counts of today is
needed for predicting camera counts of tomorrow
sel_cam = pd.DataFrame(cam_shwifi_lib.loc['2020-01-14 00:00:00' : '2020-
03-12 23:00:00'].occ_cam.values, index = cam_shwifi_lib.loc['2020-01-
14 00:00:00' : '2020-03-12 23:00:00'].index)

sel_wifi = pd.DataFrame(cam_shwifi_lib.loc['2020-01-13 00:00:00' : '2020-
03-11 23:00:00'].adj_occ_wifi.values, index = cam_shwifi_lib.loc['2020-01-
14 00:00:00' : '2020-03-12 23:00:00'].index)

future_dataset = pd.concat([sel_cam, sel_wifi], axis=1).sort_index()
future_dataset.columns = ["cam", "wifi"]
future_dataset = future_dataset
future_dataset.head()
```

```
# normalizing wifi counts
scaler = MinMaxScaler()
scaled = scaler.fit_transform(future_dataset[["wifi"]])
scaled = pd.DataFrame(scaled, index = future_dataset.index, columns = ["wi
fi"])
future_dataset[["wifi"]] = scaled

# adding required attributes
future_dataset['Date'] = future_dataset.index.map(lambda t: t.date())
future_dataset['hour'] = future_dataset.index.map(lambda t: t.time().hour)
future_dataset['Weekday'] = future_dataset.index.map(lambda t: t.date().we
ekday())
future_dataset['Time'] = future_dataset.index.map(lambda t: t.time())

# selecting daytime period (9 am to 10 pm)
for row in future_dataset.index:
  if (future_dataset['hour'][row] < 9):
    future_dataset = future_dataset.drop(index = row)

  elif (future_dataset['hour'][row] > 22):
    future_dataset = future_dataset.drop(index = row)

#spliting dataset into weekdays/weekends
future_dataset_weekdays = future_dataset[
      (future_dataset.index.map(lambda t: t.date().weekday()) == 0) |
      (future_dataset.index.map(lambda t: t.date().weekday()) == 1) |
      (future_dataset.index.map(lambda t: t.date().weekday()) == 2) |
      (future_dataset.index.map(lambda t: t.date().weekday()) == 3) |
      (future_dataset.index.map(lambda t: t.date().weekday()) == 4) ]

future_dataset_weekends = future_dataset[
      (future_dataset.index.map(lambda t: t.date().weekday()) == 5) |
      (future_dataset.index.map(lambda t: t.date().weekday()) == 6) ]

future_dataset
```

A-9. Day-ahead occupancy counts prediction/forecasting

```
# Weekdays' model

# Splitting dataset into predictors and response features
x_train = pd.concat([pd.get_dummies(future_dataset_weekdays.index.hour),
                  pd.get_dummies(future_dataset_weekdays.index.dayofweek),
                     future_dataset_weekdays.reset_index()['wifi'],
                     pd.DataFrame(future_dataset_weekdays.index)
                     ], axis=1)

x_train.columns = ['h9', 'h10', 'h11', 'h12', 'h13', 'h14', 'h15',
                   'h16', 'h17', 'h18', 'h19', 'h20', 'h21', 'h22',
                   'mon', 'tue', 'wed', 'thu', 'fri',
                   'wifi', 'datetime']

x_train.set_index('datetime', inplace = True)

y_train = future_dataset_weekdays["cam"]

# Calling occupancy prediction function with 38 fold cross-validation
lb_occupancy_prediction(x_train, y_train, 38)

est = sm.OLS(y_train.values.reshape(-1,1), sm.add_constant(x_train))
est2 = est.fit()
print(est2.summary())

print("*day-ahead model\n",
      "*weekdays' model\n",
      "*tvalues:\n\n", est2.tvalues.abs().round(2).sort_values())
```

```
# Weekends' model

# Splitting dataset into predictors and response features
x_train = pd.concat([pd.get_dummies(future_dataset_weekends.index.hour),
                     pd.get_dummies(future_dataset_weekends.index.dayofweek,
                                    drop_first=True),
                        future_dataset_weekends.reset_index()['wifi'],
                        pd.DataFrame(future_dataset_weekends.index)
                        ], axis=1)

x_train.columns = ['h9', 'h10', 'h11', 'h12', 'h13', 'h14', 'h15',
                   'h16', 'h17', 'h18', 'h19', 'h20', 'h21', 'h22',
                   'sat/sun',
                   'wifi', 'datetime']

x_train.set_index('datetime', inplace = True)

y_train = future_dataset_weekends["cam"]

# Calling  occupancy prediction function with 14 fold cross-validation
lb_occupancy_prediction(x_train, y_train, 14)

est = sm.OLS(y_train.values.reshape(-1,1), sm.add_constant(x_train))
est2 = est.fit()
print(est2.summary())

print("*day-ahead model\n",
      "*weekends' model\n",
      "*tvalues:\n\n", est2.tvalues.abs().round(2).sort_values())
```

Appendix B – Python code of week-ahead occupancy pattern prediction/forecasting

B-1. Function for cross-validating prediction models

```
def jmsb_occupancy_pattern_prediction(x_train, y_train, n):

  kf = KFold(n_splits=n)

# defining lists to keep the results of each fold
  rmses = []
  mapes = []
  day_mapes = []
  night_mapes = []
  d2s = []
  residual = []
  predictions = []

# dividing data into n folds and train and test the model on each fold
  for train_index, test_index in kf.split(x_train):
    x_tr = x_train.iloc[train_index]
    x_te = x_train.iloc[test_index]
    y_tr = y_train.iloc[train_index]
    y_te = y_train.iloc[test_index]

    glm = PoissonRegressor(max_iter=1000)
    glm.fit(x_tr, y_tr)

    prediction = glm.predict(x_te)
    predictions.append(prediction)

# calculating d2 of each fold
    d2 = np.round(glm.score(x_te, y_te),2)
    d2s.append(d2)

# calculating rmse of each fold
    rmse = mean_squared_error(y_te, prediction, squared=False)
    rmses.append(rmse)

# calculating mape of each fold
    mape = mean_absolute_percentage_error(y_te, prediction)
    mapes.append(mape)

# calculating residuals of each fold
    pred_vs_act = pd.concat([y_te,
                             pd.DataFrame(prediction, index=y_te.index)],
                             axis=1)
    pred_vs_act.columns = ["Actual", "Predicted"]
    resid = pred_vs_act["Predicted"] - pred_vs_act["Actual"]
    residual.append(resid)
```

```
# calculating mape of each fold for only daytime period
    day = pred_vs_act.copy()
    day['Time'] = day.index.map(lambda t: t.time().hour)

    for row in day.index:
      if day["Time"][row] < 8:
        day = day.drop(index = row)
      elif day["Time"][row] > 21:
        day = day.drop(index = row)

    day_mape = mean_absolute_percentage_error(day.Actual, day.Predicted)
    day_mapes.append(day_mape)

# calculating mape of each fold for only nighttime period
    night = pred_vs_act.copy()
    night['Time'] = night.index.map(lambda t: t.time().hour)

    for row in night.index:
      if (night["Time"][row] > 7):
        if (night["Time"][row] < 22):
          night = night.drop(index = row)

    night_mape = mean_absolute_percentage_error(night.Actual,
                                                night.Predicted)
    night_mapes.append(night_mape)

# plotting the prediction vs actual values for each fold
    pred_vs_act.plot(figsize=(20,7))
    plt.xlabel("Time")
    plt.ylabel("Clients")

# plotting the histogram of residuals for each fold
    fig = plt.figure(figsize=(20,7))
    plt.hist(resid, bins=50)
    plt.xlabel("Residuals")
    plt.ylabel("Frequency")

# plotting the scatter plot of residuals vs predictions for each fold
    plt.figure(figsize=(20,7))
    sns.regplot(prediction, resid, scatter=True, lowess=True,
                line_kws={'color': 'red', 'lw': 1, 'alpha': 0.8})
    plt.ylabel("Standarized Residuals")
    plt.xlabel("Fitted value")

# training the a model on the entire dataset for studying the coefficients
  print('\nTrain on All: \n')

  glm.fit(x_train, y_train)

  print('\nCoefficients: \n', glm.coef_)
  print('\nIntercept: \n', glm.intercept_)
```

```
poisson_training_results = sm.GLM(
            pd.DataFrame(y_train.tolist(), index=y_train.index),
            sm.add_constant(x_train.set_index(y_train.index)),
            family=sm.families.Poisson()).fit()
  print(poisson_training_results.summary())

# plotting the histogram of standardized deviance residuals
  fig, ax = plt.subplots()
  resid = poisson_training_results.resid_deviance.copy()
  print("\ndeviance residual mean:\n",
        resid.mean(),
        "\ndeviance residual std:\n",
        resid.std())
  resid_std = stats.zscore(resid)
  ax.hist(resid_std, bins=25)
  ax.set_title('Histogram of standardized deviance residuals');

# Printing the results of each fold and also the average of the results fo
r each metrics

  print("\n\nd2:", d2s)
  print("\n\nAve d2:", np.mean(d2s))
  print("\n\nrmse:", rmses)
  print("\n\nAve rmse:", round(np.mean(rmses),2))
  print("\n\nmape:", mapes)
  print("\n\nAve mape:", round(np.mean(mapes),2))
  print("\n\nday_mape:", day_mapes)
  print("\n\nAve day_mape:", round(np.mean(day_mapes),2))
  print("\n\nnight_mape:", night_mapes)
  print("\n\nAve night_mape:", round(np.mean(night_mapes),2))
```

B-2. Week-ahead occupancy pattern prediction/forecasting

```
# Fridays' model

# Creating the Fridays' dataframe
jmsb_hourly_fridays = jmsb_hourly_weekdays[
                     (jmsb_hourly_weekdays.Weekday == 4)].drop(['Weekday',
                                                               'WeekNo',
                                                               'Date',
                                                               'Time'],
                                                                axis=1)
jmsb_hourly_fridays.plot(figsize=(25, 5))

# Defining a new attribute "night" to distinguish the working time (i.e. 8
:00 a.m. to 9:00 p.m.) against nighttime (i.e. before 8:00 a.m. and after
9:00 p.m.)
jmsb_hourly_fridays['hour'] = jmsb_hourly_fridays.index.map(lambda t: t.ti
me().hour)

jmsb_hourly_fridays["night"] = 0

for row in jmsb_hourly_fridays.index:
  if jmsb_hourly_fridays["hour"][row] < 8:
    jmsb_hourly_fridays.loc[row,"night"] = 1

  elif jmsb_hourly_fridays["hour"][row] > 21:
    jmsb_hourly_fridays.loc[row,"night"] = 1

# Separating the predictors (x) from response attribute (y)
fri_x_train = pd.concat([pd.get_dummies(jmsb_hourly_fridays.index.hour),
                         pd.Series(jmsb_hourly_fridays.night.tolist())],
                         axis=1)

fri_y_train = jmsb_hourly_fridays["clients"]

fri_x_train.columns = ["h_0", "h_1", "h_2", "h_3", "h_4", "h_5", "h_6",
                       "h_7", "h_8", "h_9", "h_10", "h_11", "h_12",
                       "h_13", "h_14", "h_15", "h_16", "h_17", "h_18",
                       "h_19", 'h_20', "h_21", "h_22", "h_23",
                        "Day/Night"]

# Running the prediction model
jmsb_occupancy_pattern_prediction(fri_x_train, fri_y_train, 7)
```

```
# Mondays'-Thursdays' model

# Creating the Mondays'-Thursdays' dataframe
jmsb_hourly_mondays_thursdays = jmsb_hourly_weekdays[
            (jmsb_hourly_weekdays.Weekday == 0) |
            (jmsb_hourly_weekdays.Weekday == 1) |
            (jmsb_hourly_weekdays.Weekday == 2) |
            (jmsb_hourly_weekdays.Weekday == 3) ].drop(['Weekday',
                                                        'WeekNo',
                                                        'Date',
                                                        'Time'],
                                                        axis=1)

jmsb_hourly_mondays_thursdays.plot(figsize=(25, 5))

# Defining a new attribute "night" to distinguish the working time (i.e. 8
:00 a.m. to 9:00 p.m.) against nighttime (i.e. before 8:00 a.m. and after
9:00 p.m.)
jmsb_hourly_mondays_thursdays['hour'] = jmsb_hourly_mondays_thursdays.inde
x.map(lambda t: t.time().hour)

jmsb_hourly_mondays_thursdays["night"] = 0

for row in jmsb_hourly_mondays_thursdays.index:
  if jmsb_hourly_mondays_thursdays["hour"][row] < 8:
    jmsb_hourly_mondays_thursdays.loc[row,"night"] = 1

  elif jmsb_hourly_mondays_thursdays["hour"][row] > 21:
    jmsb_hourly_mondays_thursdays.loc[row,"night"] = 1

# Separating the predictors (x) from response attribute (y)
mon_thu_x_train = pd.concat([
            pd.get_dummies(jmsb_hourly_mondays_thursdays.index.hour),
            pd.get_dummies(jmsb_hourly_mondays_thursdays.index.dayofweek),
            pd.Series(jmsb_hourly_mondays_thursdays.night.tolist())],
            axis=1)

mon_thu_y_train = jmsb_hourly_mondays_thursdays["clients"]

mon_thu_x_train.columns =["h_0", "h_1", "h_2", "h_3", "h_4", "h_5", "h_6",
                          "h_7", "h_8", "h_9", "h_10", "h_11", "h_12",
                          "h_13", "h_14", "h_15", "h_16", "h_17", "h_18",
                          "h_19", 'h_20', "h_21", "h_22", "h_23",
                          "Mon", "Tue", "Wed", "Thu", "Day/Night"]

# Running the prediction model
jmsb_occupancy_pattern_prediction(mon_thu_x_train, mon_thu_y_train, 8)
```

```
# Saturdays',Sundays' model

# Creating the Saturdays',Sundays' dataframe
jmsb_hourly_weekends = jmsb_hourly_weekends.drop(['Weekday',
                                                  'WeekNo',
                                                  'Date',
                                                  'Time'],
                                                  axis=1)

jmsb_hourly_weekends.plot(figsize=(25, 5))

# Defining a new attribute "night" to distinguish the working time (i.e. 8
:00 a.m. to 9:00 p.m.) against nighttime (i.e. before 8:00 a.m. and after
9:00 p.m.)
jmsb_hourly_weekends['hour'] = jmsb_hourly_weekends.index.map(lambda t: t.
time().hour)

jmsb_hourly_weekends["night"] = 0

for row in jmsb_hourly_weekends.index:
  if jmsb_hourly_weekends["hour"][row] < 8:
    jmsb_hourly_weekends.loc[row,"night"] = 1

  elif jmsb_hourly_weekends["hour"][row] > 21:
    jmsb_hourly_weekends.loc[row,"night"] = 1

# Separating the predictors (x) from response attribute (y)
weekends_x_train = pd.concat([
     pd.get_dummies(jmsb_hourly_weekends.index.hour),
     pd.get_dummies(jmsb_hourly_weekends.index.dayofweek,
                    drop_first=True),
     pd.Series(jmsb_hourly_weekends.night.tolist())], axis=1)

weekends_y_train = jmsb_hourly_weekends["clients"]

weekends_x_train.columns = ["h_0","h_1","h_2", "h_3", "h_4", "h_5", "h_6",
                            "h_7", "h_8", "h_9", "h_10", "h_11", "h_12",
                            "h_13","h_14","h_15", "h_16", "h_17", "h_18",
                            "h_19", 'h_20', "h_21", "h_22", "h_23",
                            "sat/sun", "Day/Night"]

# Running the prediction model
jmsb_occupancy_pattern_prediction(weekends_x_train, weekends_y_train, 7)
```

Appendix C – Prediction models coefficients

Following are the notes related to the tables in this Appendix.

- Description of features used as input to the models is presented in Table C- 1.

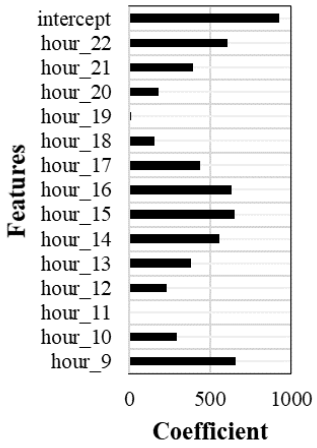
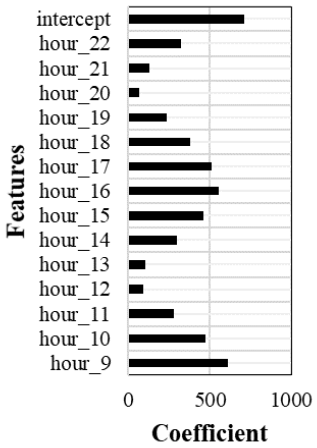
Table C- 1. Description of features used in the prediction models

Original feature	Feature in the model	Notes
Hour of the day	hour_0 hour_1 hour_2 hour_3 hour_4 hour_5 hour_6 hour_7 hour_8 hour_9 hour_10 hour_11 hour_12 hour_13 hour_14 hour_15 hour_16 hour_17 hour_18 hour_19 hour_20 hour_21 hour_22 hour_23	Feature "Hour of the day" is one-hot encoded into 24 binary features. Selected hours were used in different prediction models.
Day of the week	mon tue wed thu fri sat/sun	Feature "Day of the week" is entailed of five weekdays and two weekends. Weekdays are one-hot encoded into 5 binary features. Weekends are one-hot encoded into 1 binary feature.
Day/Night	Day/Night	Feature "Day/Night" is a binary feature differentiating the working hours from nighttime
WiFi connection count	Normalized WiFi	Feature "WiFi connection count" is normalized.

- The significance of the features is shown using p-value and t-value. The features are sorted based on the absolute value of features' t-value.
- Bar charts are plotted based on the absolute value of features' coefficient.

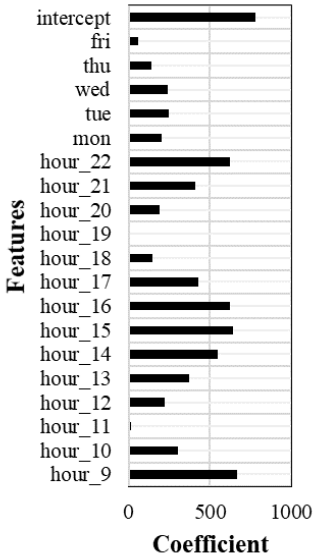
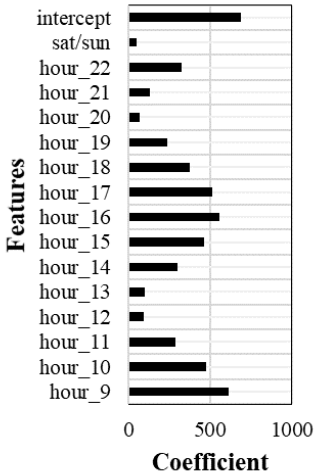
C-1. Real-time occupancy counts estimation model (Model set 1)

Table C- 2. Coefficients, p-values, t-values, and bar chart of the features in the real-time occupancy count estimation model (model set 1)

Model	Feature	Coefficient	p-value	t-value	Bart chart
Weekdays' model	intercept	925.53	0.00	94.30	
	hour_9	-655.49	0.00	-17.28	
	hour_15	651.83	0.00	17.19	
	hour_16	634.88	0.00	16.74	
	hour_22	-610.24	0.00	-16.09	
	hour_14	555.67	0.00	14.65	
	hour_17	440.49	0.00	11.62	
	hour_21	-397.83	0.00	-10.49	
	hour_13	380.17	0.00	10.03	
	hour_10	-293.49	0.00	-7.74	
	hour_12	231.38	0.00	6.10	
	hour_20	-182.24	0.00	-4.81	
	hour_18	159.92	0.00	4.22	
	hour_19	14.92	0.69	0.39	
	hour_11	-4.44	0.91	-0.12	
Weekends' model	intercept	713.86	0.00	49.10	
	hour_9	-609.23	0.00	-10.85	
	hour_16	556.52	0.00	9.91	
	hour_17	513.52	0.00	9.14	
	hour_10	-475.61	0.00	-8.47	
	hour_15	463.33	0.00	8.25	
	hour_18	379.33	0.00	6.75	
	hour_22	-323.67	0.00	-5.76	
	hour_14	299.70	0.00	5.34	
	hour_11	-282.42	0.00	-5.03	
	hour_19	237.58	0.00	4.23	
	hour_21	-126.61	0.03	-2.25	
	hour_13	103.83	0.07	1.85	
	hour_12	-90.23	0.11	-1.61	
	hour_20	67.83	0.23	1.21	

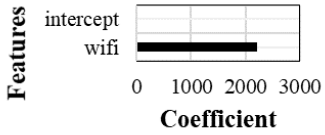
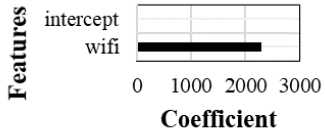
C-2. Real-time occupancy counts estimation model (Model set 2)

Table C- 3. Coefficients, p-values, t-values, and bar chart of the features in the real-time occupancy count estimation model (model set 2)

Model	Feature	Coefficient	p-value	t-value	Bart chart
Weekdays' model	intercept	776.15	0.00	103.15	
	hour_9	-666.16	0.00	-19.33	
	hour_15	641.16	0.00	18.61	
	hour_16	624.21	0.00	18.11	
	hour_22	-620.91	0.00	-18.02	
	hour_14	545.00	0.00	15.82	
	tue	247.74	0.00	13.04	
	wed	241.95	0.00	12.74	
	hour_17	429.82	0.00	12.47	
	hour_21	-408.50	0.00	-11.85	
	hour_13	369.50	0.00	10.72	
	mon	202.53	0.00	10.66	
	hour_10	-304.16	0.00	-8.83	
	thu	141.12	0.00	7.43	
	hour_12	220.71	0.00	6.41	
	hour_20	-192.91	0.00	-5.60	
	hour_18	149.25	0.00	4.33	
	fri	-57.19	0.00	-2.87	
	hour_11	-15.11	0.66	-0.44	
	hour_19	4.25	0.90	0.12	
Weekends' model	intercept	690.11	0.00	33.70	
	hour_9	-610.93	0.00	-10.92	
	hour_16	554.82	0.00	9.92	
	hour_17	511.82	0.00	9.15	
	hour_10	-477.31	0.00	-8.53	
	hour_15	461.63	0.00	8.25	
	hour_18	377.63	0.00	6.75	
	hour_22	-325.37	0.00	-5.81	
	hour_14	298.01	0.00	5.33	
	hour_11	-284.12	0.00	-5.08	
	hour_19	235.88	0.00	4.22	
	hour_21	-128.31	0.02	-2.29	
	hour_13	102.13	0.07	1.83	
	hour_12	-91.93	0.10	-1.64	
	sat/sun	50.89	0.10	1.64	
	hour_20	66.13	0.24	1.18	

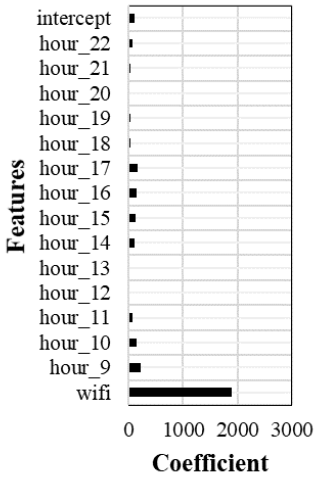
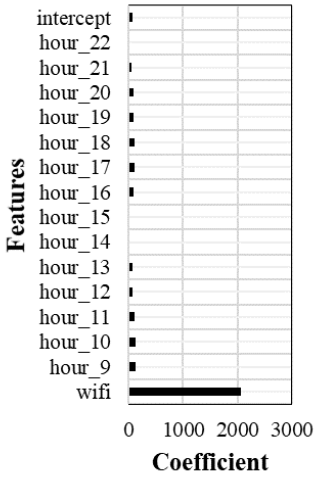
C-3. Real-time occupancy counts estimation model (Model set 3)

Table C- 4. Coefficients, p-values, t-values, and bar chart of the features in the real-time occupancy count estimation model (model set 3)

Model	Feature	Coefficient	p-value	t-value	Bart chart
Weekdays' model	normalized wifi	2212.38	0.00	100.25	
	intercept	-15.15	0.18	-1.36	
Weekends' model	normalized wifi	2287.65	0.00	70.73	
	intercept	6.73	0.58	0.55	

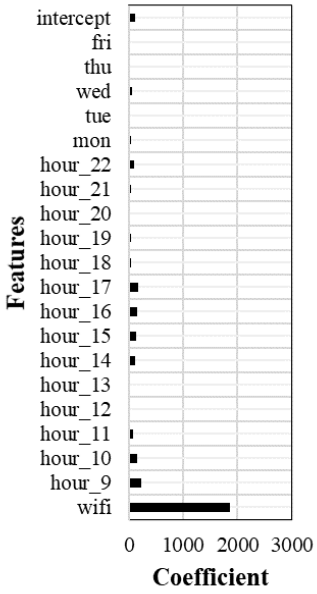
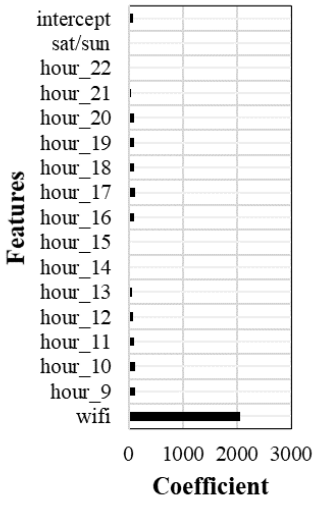
C-4. Real-time occupancy counts estimation model (Model set 4)

Table C- 5. Coefficients, p-values, t-values, and bar chart of the features in the real-time occupancy count estimation model (model set 4)

Model	Feature	Coefficient	p-value	t-value	Bart chart
Weekdays' model	normalized wifi	1899.35	0.00	67.75	
	hour_9	-221.49	0.00	-15.36	
	hour_17	168.19	0.00	12.43	
	hour_10	-143.31	0.00	-10.94	
	hour_16	143.39	0.00	9.68	
	intercept	118.81	0.00	9.61	
	hour_15	127.18	0.00	8.45	
	hour_14	100.56	0.00	6.91	
	hour_22	-79.97	0.00	-5.30	
	hour_11	-67.38	0.00	-5.20	
	hour_18	41.59	0.00	3.19	
	hour_21	-37.89	0.01	-2.71	
	hour_19	33.51	0.01	2.59	
	hour_12	24.41	0.07	1.84	
	hour_20	18.47	0.16	1.39	
	hour_13	11.54	0.41	0.82	
Weekends' model	normalized wifi	2058.01	0.00	69.94	
	hour_10	-122.44	0.00	-9.82	
	hour_9	-122.08	0.00	-9.14	
	hour_11	-106.35	0.00	-9.11	
	hour_17	114.69	0.00	9.00	
	hour_18	105.00	0.00	8.71	
	hour_19	96.15	0.00	8.31	
	hour_20	94.02	0.00	8.24	
	intercept	77.31	0.00	8.08	
	hour_16	91.03	0.00	6.90	
	hour_12	-76.38	0.00	-6.70	
	hour_13	-67.93	0.00	-5.83	
	hour_21	46.54	0.00	3.99	
	hour_22	22.49	0.07	1.81	
	hour_15	19.29	0.14	1.48	
	hour_14	-16.72	0.17	-1.36	

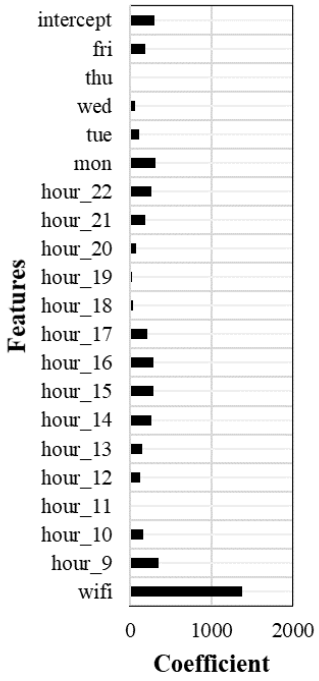
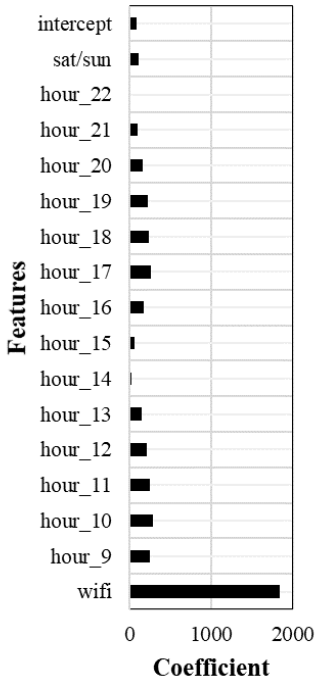
C-5. Real-time occupancy counts estimation model (Model set 5)

Table C- 6. Coefficients, p-values, t-values, and bar chart of the features in the real-time occupancy count estimation model (model set 5)

Model	Feature	Coefficient	p-value	t-value	Bart chart
Weekdays' model	normalized wifi	1861.59	0.00	61.58	
	hour_9	-231.67	0.00	-15.93	
	hour_17	172.05	0.00	12.85	
	hour_10	-147.84	0.00	-11.40	
	hour_16	151.62	0.00	10.21	
	intercept	113.22	0.00	10.18	
	hour_15	136.07	0.00	8.99	
	hour_14	108.06	0.00	7.42	
	wed	49.18	0.00	6.40	
	hour_22	-92.06	0.00	-6.00	
	hour_11	-67.67	0.00	-5.31	
	mon	34.91	0.00	4.64	
	hour_21	-46.59	0.00	-3.33	
	hour_18	42.40	0.00	3.30	
	hour_19	31.60	0.01	2.48	
	tue	18.50	0.02	2.33	
	hour_12	26.98	0.04	2.06	
	thu	11.57	0.11	1.58	
	hour_13	17.33	0.22	1.24	
	hour_20	12.94	0.33	0.98	
	fri	-0.95	0.90	-0.13	
Weekends' model	normalized wifi	2057.98	0.00	69.31	
	hour_10	-122.45	0.00	-9.78	
	hour_9	-122.09	0.00	-9.09	
	hour_11	-106.36	0.00	-9.08	
	hour_17	114.69	0.00	8.97	
	hour_18	105.01	0.00	8.69	
	hour_19	96.15	0.00	8.29	
	hour_20	94.01	0.00	8.22	
	intercept	77.29	0.00	7.90	
	hour_16	91.04	0.00	6.87	
	hour_12	-76.38	0.00	-6.68	
	hour_13	-67.93	0.00	-5.81	
	hour_21	46.53	0.00	3.98	
	hour_22	22.48	0.07	1.80	
	hour_15	19.29	0.14	1.47	
	hour_14	-16.72	0.18	-1.36	
	sat/sun	0.06	0.99	0.01	

C-6. Day-ahead occupancy counts prediction/forecasting model

Table C- 7. Coefficients, p-values, t-values, and bar chart of the features in the day-ahead occupancy count prediction/forecasting model

Model	Feature	Coefficient	p-value	t-value	Bart chart
Weekdays' model	normalized wifi (previous day)	1374.553	0	31.638	
	mon	319.637	0	30.423	
	intercept	304.609	0	18.339	
	fri	-187.526	0	-17.03	
	hour_9	-345.403	0	-16.409	
	hour_15	288.667	0	13.29	
	hour_16	282.687	0	13.219	
	hour_14	260.662	0	12.624	
	hour_22	-261.112	0	-11.928	
	hour_17	219.633	0	11.445	
	tue	118.988	0	11.251	
	hour_21	-182.634	0	-9.275	
	hour_10	-158.987	0	-8.609	
	hour_13	153.923	0	7.853	
	hour_12	130.222	0	7.107	
	wed	62.134	0	5.528	
	hour_20	-78.143	0	-4.234	
	hour_18	32.419	0.075	1.784	
	hour_19	-19.625	0.273	-1.098	
	hour_11	-17.701	0.323	-0.989	
	thu	-8.624	0.424	-0.8	
Weekends' model	normalized wifi (previous day)	1835.234	0	18.679	
	hour_10	-290.555	0	-7.711	
	hour_11	-248.162	0	-6.846	
	hour_17	264.697	0	6.83	
	hour_18	240.785	0	6.514	
	hour_19	228.715	0	6.322	
	hour_9	-249.813	0	-6.042	
	hour_12	-206.274	0	-5.617	
	sat/sun	109.023	0	5.339	
	hour_20	166.119	0	4.534	
	hour_16	174.718	0	4.151	
	hour_13	-154.228	0	-3.969	
	hour_21	101.077	0.009	2.632	
	intercept	81.767	0.025	2.254	
	hour_15	66.627	0.118	1.571	
	hour_14	-27.812	0.493	-0.686	
	h22	15.874	0.698	0.389	

C-7. Week-ahead occupancy pattern prediction/forecasting model

Table C- 8. Coefficients, p-values, t-values, and bar chart of the features in the week-ahead occupancy pattern prediction/forecasting model

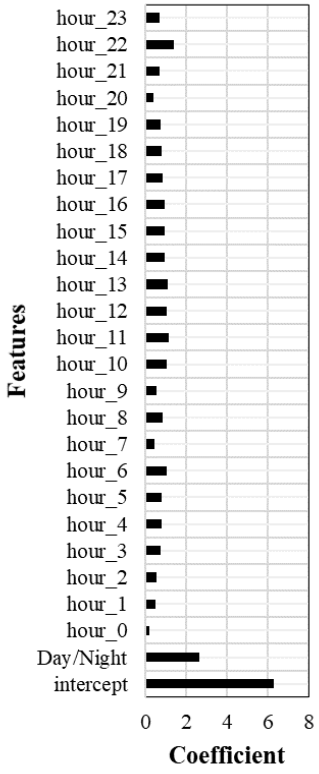
Model	Feature	Coefficient	p-value	z	Bart chart
Fridays' model	intercept	6.28	0.00	1815.54	
	Day/Night	-2.67	0.00	-124.22	
	hour_11	1.14	0.00	121.69	
	hour_13	1.07	0.00	111.71	
	hour_12	1.07	0.00	111.19	
	hour_10	1.02	0.00	104.64	
	hour_14	0.96	0.00	95.98	
	hour_16	0.95	0.00	93.37	
	hour_15	0.92	0.00	90.08	
	hour_17	0.85	0.00	80.14	
	hour_18	0.81	0.00	75.44	
	hour_19	0.76	0.00	69.56	
	hour_9	0.55	0.00	45.36	
	hour_22	1.38	0.00	38.34	
	hour_8	-0.85	0.00	-35.96	
	hour_21	-0.68	0.00	-31.15	
	hour_20	0.37	0.00	28.22	
	hour_23	0.69	0.00	15.00	
	hour_6	-1.05	0.00	-10.78	
	hour_5	-0.82	0.00	-9.35	
	hour_4	-0.80	0.00	-9.23	
	hour_3	-0.76	0.00	-8.92	
	hour_2	-0.55	0.00	-7.13	
	hour_1	-0.51	0.00	-6.74	
	hour_7	-0.46	0.00	-6.22	
	hour_0	0.21	0.00	3.80	

Table C- 8. Continued

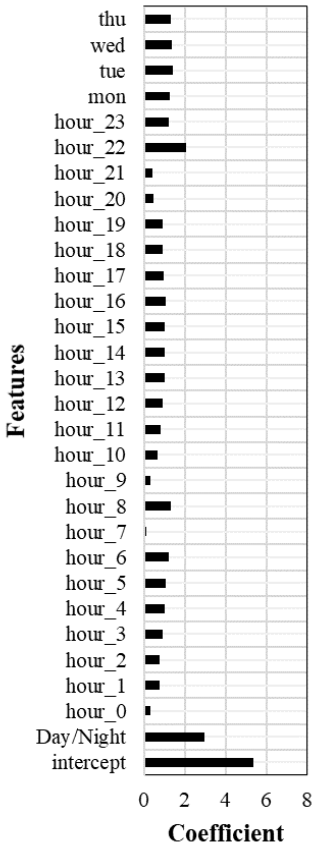
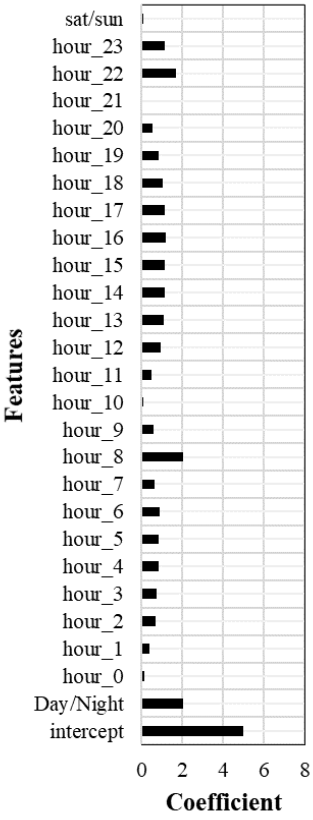
Model	Feature	Coefficient	p-value	z	Bart chart
Weekdays' model	intercept	5.39	0.00	4723.87	
	tue	1.41	0.00	744.94	
	wed	1.36	0.00	707.14	
	thu	1.33	0.00	679.39	
	mon	1.30	0.00	657.83	
	Day/Night	-2.96	0.00	-297.11	
	hour_16	1.06	0.00	289.63	
	hour_14	1.02	0.00	276.05	
	hour_15	1.02	0.00	275.90	
	hour_13	1.01	0.00	270.15	
	hour_17	0.98	0.00	258.41	
	hour_12	0.93	0.00	243.55	
	hour_18	0.92	0.00	239.85	
	hour_19	0.92	0.00	238.74	
	hour_11	0.83	0.00	206.68	
	hour_10	0.68	0.00	157.38	
	hour_22	2.06	0.00	150.90	
	hour_8	-1.35	0.00	-121.06	
	hour_20	0.45	0.00	95.51	
	hour_23	1.23	0.00	73.11	
	hour_9	0.30	0.00	60.18	
	hour_21	-0.41	0.00	-57.55	
	hour_6	-1.20	0.00	-26.33	
	hour_4	-1.04	0.00	-24.96	
	hour_5	-1.06	0.00	-24.89	
	hour_3	-0.94	0.00	-23.98	
	hour_2	-0.79	0.00	-21.31	
	hour_1	-0.79	0.00	-21.25	
	hour_0	-0.33	0.00	-11.06	
	hour_7	-0.11	0.00	-3.81	

Table C- 8. Continued

Model	Feature	Coefficient	p-value	z	Bart chart
Weekends' model	intercept	4.99	0.00	752.98	
	Day/Night	-2.03	0.00	-98.15	
	hour_16	1.17	0.00	94.09	
	hour_15	1.16	0.00	92.84	
	hour_14	1.13	0.00	90.08	
	hour_17	1.12	0.00	89.04	
	hour_13	1.09	0.00	85.66	
	hour_18	1.05	0.00	81.18	
	hour_12	0.94	0.00	69.32	
	hour_19	0.82	0.00	57.60	
	hour_22	1.71	0.00	54.58	
	hour_8	-2.04	0.00	-36.64	
	hour_20	0.53	0.00	32.91	
	hour_11	0.52	0.00	31.48	
	hour_23	1.13	0.00	30.28	
	hour_9	-0.58	0.00	-21.28	
	sat/sun	0.12	0.00	15.29	
	hour_6	-0.89	0.00	-10.38	
	hour_5	-0.83	0.00	-9.97	
	hour_4	-0.82	0.00	-9.85	
	hour_3	-0.72	0.00	-9.08	
	hour_2	-0.71	0.00	-8.99	
	hour_7	-0.65	0.00	-8.48	
	hour_1	-0.37	0.00	-5.52	
	hour_10	0.11	0.00	5.35	
	hour_0	0.14	0.01	2.58	
	hour_21	-0.01	0.59	-0.54	

Appendix D – Location of Access Points and Cameras on Floors

The location and distribution of 43 APs and two optical and thermal camera-based occupancy counters located on the second floor of the case study building (for module I) are shown in Figure D- 1.

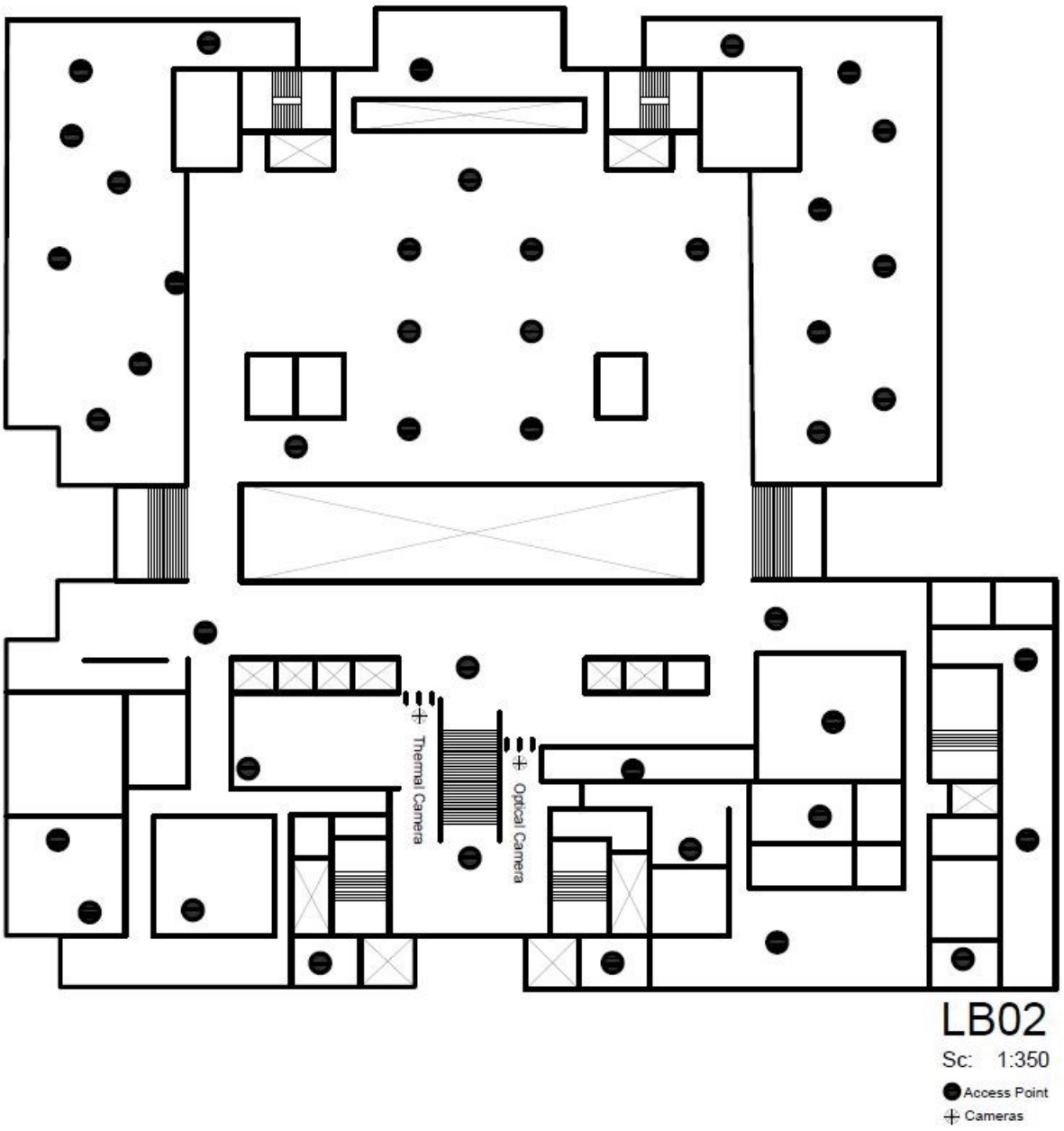


Figure D- 1. Location and distribution of 43 APs and two optical and thermal camera-based occupancy counters located on the second floor of the case study building (for module I)

The location and distribution of 11 APs located on the second floor of the case study building (for module II) are shown in Figure D- 2.

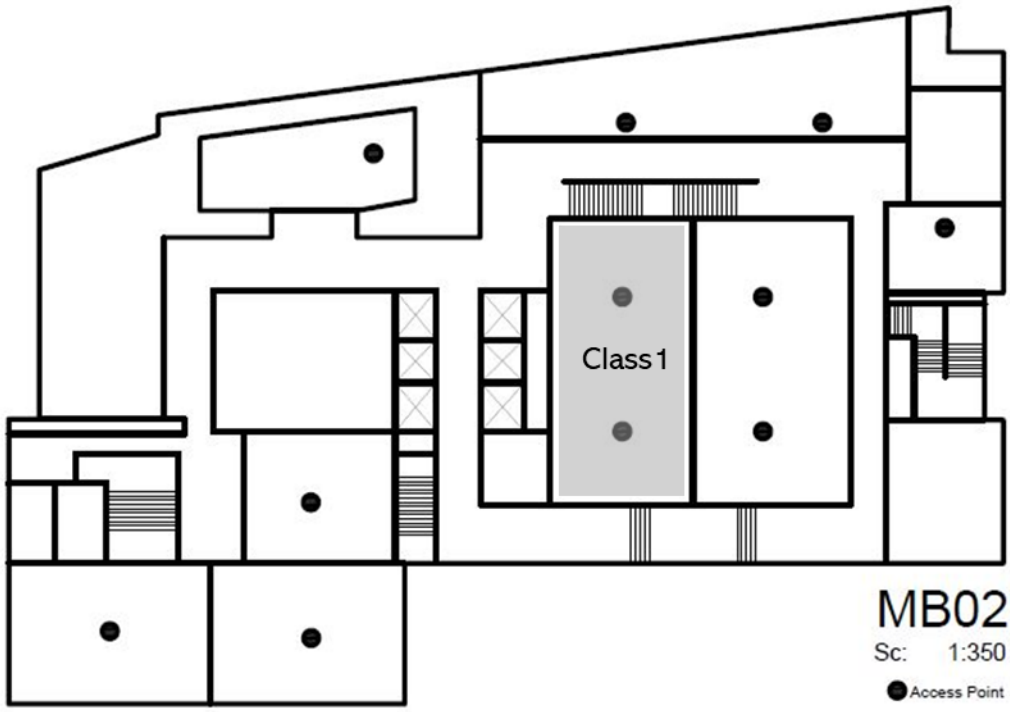


Figure D- 2. Location and distribution of 11 APs located on the second floor of the case study building (for module II)