Self-learning building HVAC control system based on dynamic occupancy patterns: A predictive approach using deep Q-networks and transfer learning

Mohammad Esrafilian-Najafabadi

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

Presented in Partial Fulfilment of the
Requirements for the Degree of
Doctor of Philosophy (Building Engineering)
at Concordia University
Montreal, Quebec, Canada

August 2022

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the Ph.D. thesis prepared

By: Mohammad Esrafilian-Najafabadi

Entitled: Self-learning building HVAC control system based on dynamic occupancy patterns: A predictive approach using deep Q-networks and transfer learning

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

Doctor of Philosophy (Building Engineering)

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

Signed by the final examining committee:

____ (Chair)
Dr. Sivakumar Narayanswamy

____ (External Examiner)
Dr. Luisa Fernanda Cabeza Fabra

____ (Examiner)
Dr. Yong Zeng

____ (Examiner)
Dr. Fuzhan Nasiri

____ (Examiner)
Dr. Leon Wang

____ (Thesis Supervisor)
Dr. Fariborz Haghighat

Approved by ____
Dr. Mazdak Nik-Bakht , Graduate Program Director

/ /2022 ____
Dr. Mourad Debbabi , Dean

Abstract

Self-learning building HVAC control system based on dynamic occupancy patterns: A predictive approach using deep Q-networks and transfer learning

Mohammad Esrafilian-Najafabadi, Ph.D. Concordia University, 2022

This dissertation reports the development of a self-learning control system that adjusts the building setpoint temperature to the dynamic occupancy schedules, aiming to maximize energy saving and thermal comfort. The controller interacts with the environment and learns the occupancy patterns and the lag time of heating, ventilation, and air-conditioning (HVAC) systems with no need for developing online models of occupancy and buildings. This process aims to minimize the runtime of HVAC systems during vacancy periods to save energy while providing thermal comfort conditions upon the occupants' arrival. This control framework also leverages the knowledge of the pre-trained controllers to accelerate the training process in unseen new buildings. This transfer learning method is performed based on an inter-building similarity analysis using unsupervised learning of the occupancy profiles. This process intends to minimize the thermal discomfort caused by the trial-and-error nature of the self-learning algorithm. The proposed system takes advantage of a double deep Q-network (DDQN) algorithm to find the optimal control policy. Moreover, an optimal feature selection algorithm is integrated into this framework for identifying irrelevancy and redundancy in the feature sets to further improve the training process. The merit of the controller is quantified by comparing its performance with that of a model-based predictive control (MPC), as a well-practiced occupancy-based control method. The results demonstrate that the control system provides superior thermal comfort for occupants by taking the occupancy forecasting uncertainty into account in the decision-making process. This ability improves thermal comfort by 7.87% on average with MPC as the benchmark. The use of the transfer learning method enhances thermal comfort by 68% during the training process of the algorithm.

Acknowledgement

First and foremost, I would like to express my deepest gratitude to Professor Fariborz Haghighat, my supervisor, for his invaluable feedback and comments. His guidance, suggestions, support, patience, and continuous encouragement made this degree possible. I am grateful for his trust in my abilities and for allowing me to join his research team.

I would like to thank my committee members, Dr. Yong Zeng, Dr. Fuzhan Nasiri, and Dr. Leon Wang, for providing me with their valuable perspectives and advice during my comprehensive exam and proposal.

I would also like to thank Concordia University for the support through Concordia Research Chair-Energy and Environment throughout my research.

My sincere thanks go to my friends and colleagues in the energy and environment group for their support and for sharing knowledge with me during my research. I am deeply indebted to Dr. Mohammad Sameti for providing me with his precious support and recommendations. I would also like to give my special thanks to Mr. Moein Dadollahi Sarab for his great expert advice and comments on different parts of my research. I would also like to express my gratitude to Mr. Mohamad Ghamangiz for his suggestions, encouragement, and for helping me to keep positive along the way.

I would like to dedicate this thesis to

My mother, Maryam, who has continuously provided her moral, spiritual, emotional, and financial support,

To my brother, Hamed,

For his endless encouragement and valuable advices,

AND

To my wonderful love, Mina,

For her patience, understanding, and support.

Table of contents

List of figures xi

List of tables xv

Chapter 1 : Introduction 1

1.1. Motivation 1

1.2. Problem statement 2

1.3. Research objective 3

1.4. Organization 4

1.5. Type of the current dissertation 4

Chapter 2 : Occupancy-based HVAC control systems in buildings: A state-of-the-art review 6

2.1. Overview 6

2.2. Introduction 6

2.2.1. Motivation for occupancy-based HVAC control 6

2.2.2. Previous literature reviews 7

2.2.3. Contributions and structure 8

2.3. Research classification 9

2.4. HVAC operation based on user-defined occupancy schedules 9

2.5. Reactive control 11

2.6. Predictive control 14

2.6.1. Occupancy prediction models 15

2.6.1.2. Occupancy state/level prediction 17

2.6.2. Prediction of individuals' occupancy patterns 19

2.6.2.1. Rule-based control 20

2.6.2.2. RB control using preconditioning time and conservative setback 20

2.6.2.3. RB control using preconditioning time and deep setback 22

2.6.2.4. RB control using conditioning rate and occupancy probability 23

2.6.3. Optimal control 25

2.7. Discussion of previous research limitations 27

2.7.1. Performance indicators for evaluating control systems 27

2.7.2. Features used to develop occupancy models 29

2.7.3. Types of control strategies and evaluation methods 29

2.7.4. Building types 31

2.7.5. Occupancy prediction models 31
2.7.6. Performance indicators for occupancy models 32
2.8. Conclusion 33
Chapter 3 : Occupancy-based HVAC control using deep learning algorithms for estimating online preconditioning time in residential buildings 34
3.1. Overview 34
3.2. Introduction 34
3.2.1. Background 36
3.3. Methodology 38
3.3.1. System description 38
3.3.2. Occupancy database and model description 39
3.3.3. Rule-based control 39
3.3.4. Preconditioning-time model 40
3.4. Building simulation 42
3.5. Performance evaluation criteria 44
3.5.1. Financial indicators 44
3.5.2. Energy and peak-demand management indicators 46
3.5.3. Thermal comfort metrics 47
3.6. Results 47
3.6.1. Performance of the preconditioning-time model 47
3.6.2. Multi-criteria decision making 48
3.6.3. Impact of setback temperature 51
3.6.4. Peak-demand performance 53
3.7. Conclusion 54
Chapter 4 : Deep learning models for future occupancy prediction in residential buildings 56
4.1. Overview 56
4.2. Introduction 56
4.3. Method 57
4.3.1. Database description and data pre-processing 57
4.3.2. Deep learning models for occupancy prediction 57
.4.3.3 Model evaluation 58
4.4. Results 59
4.5. Discussion 60

4.6. Conclusions 61

Chapter 5 : Impact of occupancy prediction models on building HVAC control system performance: Application of machine learning techniques 62

5.1. Overview 62
5.2. Introduction 62
5.2.1. Related research works 63
5.2.2. Research gaps and contributions 65
5.3. Methodology 66
5.3.1. Development of the occupancy prediction models 66
5.3.1.1. k-nearest neighbors model 67
5.3.1.2. Decision Tree models 67
5.3.1.3. Multi-layer perceptron 67
5.3.1.4. Gated recurrent units 69
5.3.2. Occupancy database description and preprocessing 71
5.3.3. Control system and performance evaluation criteria 72
5.3.3.1. Case study description 72
5.3.3.2. Control system description 73
5.3.3.3. Evaluation criteria 74
5.3.3.4. Overall performance score 74
5.3.4. Correlation between the performance indicators 75
5.4. Results and discussion 76
5.4.1. Development of the occupancy models 76
5.4.1.1. Optimal hyperparameters 76
5.4.1.2. Performance of the machine learning techniques 78
5.4.2. Control system performance 79
5.4.3. Overall performance of the occupancy models 81
5.4.4. Correlation between the performance criteria 82
5.5. Conclusion 84

Chapter 6 : Impact of predictor variables on the performance of future occupancy prediction: Feature selection using genetic algorithms and machine learning 86

6.1. Overview 86
6.2. Introduction 86
6.2.1. Related research works 87
6.2.2. Research gaps and contributions 89

6.3. Methodology 90
6.3.1. Database description and preprocessing 90
6.3.1.1. Candidate features 92
6.3.2. Optimal feature selection 96
6.3.2.1. Genetic algorithm for feature selection 97
6.3.2.2. Multi-criteria decision making 99
6.3.3. Sequential feature selection 100
6.4. Results and discussion 102
6.4.1. Multi-objective genetic algorithm 102
6.4.2. Performance of feature selection methods 104
6.4.3. Optimal feature types 107
6.4.4. Optimal number of features 109
6.5. Conclusion 110
Chapter 7 : Towards self-learning control of HVAC systems with the consideration of dynamic occupancy patterns: Application of model-free deep reinforcement learning 112
7.1. Overview 112
7.2. Methodology 112
7.2.1. Reinforcement learning (RL) 112
7.2.1.1. Markov decision process (MDP) 112
7.2.1.2. Q-learning algorithm 114
7.2.1.3. Deep Q-network 115
7.2.1.4. Double deep Q-networks (DQN) 116
7.2.1.5. Policy 116
7.2.1.6. Hyperparameter tuning 117
7.2.1.7. Employment of DDQN 118
7.2.2. Model-based predictive control 120
7.2.2.2. Building and occupancy prediction models 121
7.2.2.3. Optimization algorithm 122
7.2.2.4. MPC employment 122
7.2.3. Building model 124
7.2.4. Performance evaluation metrics 125
7.3. Case study 126
7.3.1. Occupancy database preprocessing 126
7.3.2. Building testbed 126

7.4. Results and discussion 127
7.4.1. Building model selection 127
7.4.2. DDQN control system........ 128
7.4.2.1. Hyperparameter tuning 128
7.4.2.2. Training process and convergence 128
7.4.3. Test performance evaluation 131
7.4.3.1. Intra-day performance 131
7.4.3.2. Overall performance 133
7.5. Conclusion 135
Chapter 8 : Transfer learning for model-free HVAC control based on unsupervised learning of occupancy profiles and deep reinforcement learning 137
8.1. Overview 137
8.2. Methodology 137
8.2.1. General description of the control framework 137
8.2.2. Database description and preprocessing 138
8.2.3. Control system 139
8.2.4. Unsupervised learning 141
8.2.4.2. Database preprocessing 141
8.2.4.3. K-means clustering using dynamic time warping 142
8.2.5. Transfer learning 143
8.2.5.1. Similarity analysis 144
8.2.5.2. Performance evaluation 145
8.3. Results 146
8.3.1. Occupancy patterns extraction 146
8.3.2. Similarity between the occupancy patterns 148
8.3.3. Transfer learning performance 148
8.4. Conclusion 150
Chapter 9 : Summary, conclusion and future directions 152
9.1. Summary and conclusion 152
9.2. Contributions 153
9.3. Limitations and directions for future work 154

List of figures:

Fig. 2.1. Distribution of reviewed papers based on publication year. 8

Fig. 2.2. Classification of occupancy-based HVAC control systems based on types of control strategies and the integration of occupancy information in systems. 9

Fig. 2.3. The proportion of U.S. households using different types of thermostats with central or individual air conditioning system [43]. 10

Fig. 2.4. The frequency of different performance criteria utilized to assess the merits of occupancy-based control strategies in the reviewed studies. 28

Fig. 2.5. Frequency of different features used for developing occupancy models in the reviewed papers. 29

Fig. 2.6. Distribution of the papers based on control type and evaluation methods. 30

Fig. 2.7. The distribution of testbeds used to evaluate occupancy-based control systems in the reviewed papers. 31

Fig. 2.8. Frequency of various occupancy models used in the literature. 32

Fig. 2.9. Performance criteria used for evaluation of occupancy models. 32

Fig. 3.1. Schematic diagram of the proposed RB control framework. 38

Fig. 3.2. The flow diagram of the RB control system proposed in this study. 40

Fig. 3.3. The diagram of a sample multi-layer perceptron network implemented for preconditioning time estimation. 42

Fig. 3.4. The designed model of the simulated building as the case study. 43

Fig. 3.5. Interactions between different applications and libraries used in this study. 44

Fig. 3.6. Off-, mid-, and on-peak periods considered for evaluating the peak-demand performance of the systems [143]. 47

Fig. 3.7. The performance of the final MLP model after each epoch of the training process. 48

Fig. 3.8. The distribution of NPV for the RB control systems in different setback bounds. 49

Fig. 3.9. The distribution of MissTime for RB control systems in different setback temperature bounds. 50

Fig. 3.10. Average MissTime and NPV for the selected RB control systems based on the allowed setback temperature. 52

Fig. 3.11. Average hourly energy consumption of the PRB system using the medium setback. . 54

Fig. 4.1. The accuracy of the deep learning models based on prediction horizon. 59

Fig. 4.2. Computational time required for training the deep learning models. 60

Fig. 5.1. Sample MLP model used for a) arrival time and b) occupancy-state prediction with two hidden layers (N=12 and M=12). 69

Fig. 5.2. Many-to-one architecture of the GRU model for arival time prediction (M=12). 70

Fig. 5.3. Many-to-many architecture of the GRU method for occupancy-state prediction (N=12 and M=12) 70

Fig. 5.4. The floor plan of the virtual test case. 72
Fig. 5.5. The RB control system using occupancy models for regulating setpoint/setback temperature. 74
Fig. 5.6. The performance of deep learning models during the training process for (a) classificatoin and (b) regression tasks. 77
Fig. 5.7. The distribution of the accuracy calculated for the occupancy-state prediction models. 78
Fig. 5.8. The distribution of the MAE calculated for the arrival time prediction models. 79
Fig. 5.9. The ditribution of *MissTime* performance for using different occupancy models in the RB control system. 80
Fig. 5.10. The distribution of energy-efficiency performance of using different occupancy models in the RB control system. 81
Fig. 5.11. The average relative closeness for the occupancy-state and arrival-time prediction models utilized in the RB control system. 82
Fig. 5.12. Average relative closeness and the corresponding normalized accuracy and MAE associated with each model. 83
Fig. 5.13. Pearson correlation coefficient for evaluating the relationship between the machine learning performance metrics and relative closeness. 84
Fig. 6.1. The floor plan of the apartments utilized to collect the occupancy database. 92
Fig. 6.2. The average occupancy ratios in different hours of day and days of week for three sample a) T4, b) T3, and c) T2 apartments. 93
Fig. 6.3. The lagged and current occupancy states utilized to forecast future occupancy. 94
Fig. 6.4. The plug load patterns from a sample apartment a) before and b) after addressing the outliers. 95
Fig. 6.5. The schematic diagram of the MOGA. 98
Fig. 6.6. The schematic diagram of the FSS process. 101
Fig. 6.7. The schematic diagram of the BSS process. 102
Fig. 6.8. A sample Pareto frontier solution including the PIS, NIS, and the selected solutions based on the TOPSIS and MA methods. 103
Fig. 6.9. The convergence of the hypervolume quality indicator based on the number of function evaluations. 104
Fig. 6.10. Distribution of the test accuracy based on different FS methods. 105
Fig. 6.11. Distribution of the mean CV accuracy based on different FS methods. 106
Fig. 6.12. Distribution of the number of features selected using the MOGA and SOGA. 107
Fig. 6.13. The frequency of the selected features based on the MA approach. 108
Fig. 6.14. The frequency of the selected features based on the TOPSIS method. 109
Fig. 6.15. The average number of features selected based on the MA approach. 109
Fig. 6.16. The average number of features selected according to the TOPSIS method. 110

Fig. 7.1. The working diagram of the implemented DDQN algorithm. 119

Fig. 7.2. The general overview of the interactions between libraries and software packages utilized to develop the DDQN controller. 120

Fig. 7.3. The overview of the workflow of the MPC. 121

Fig. 7.4. The schematic diagram of the MPC implemented in this study. 123

Fig. 7.5. The performance of the building models in terms of prediction performance and computational time. 127

Fig. 7.6. The episodic rewards achieved by the DDQN agent during the training process. 129

Fig. 7.7. Average temperature deviation during the training period of the DDQN controller. .. 130

Fig. 7.8. Deviation period during the training period of the DDQN controller. 130

Fig. 7.9. The energy consumption per timestep during the training period of the DDQN controller. 131

Fig. 7.10. The intra-day operative temperature and energy consumption based on perfect occupancy prediction for a sample apartment on February 1st. 132

Fig. 7.11. The intra-day operative temperature and energy consumption based on actual occupancy prediction for a sample apartment on February 1st. 133

Fig. 7.12. The distribution of average temperature deviation associated with the control systems based on perfect and actual occupancy models. 134

Fig. 7.13. The distribution of deviation period associated with the control systems based on perfect and actual occupancy models. 134

Fig. 7.14. The distribution of energy saving associated with the control systems based on perfect and actual occupancy models. 135

Fig. 8.1. The general framework of the proposed control system. 138

Fig. 8.2. The schematic diagram of the data collection and preprocessing process. 138

Fig. 8.3. The general description of the RL agent via MDP. 140

Fig. 8.4. The unsupervised learning process to find the dominant occupancy profiles in buildings. 141

Fig. 8.5. Daily occupancy patterns for a random day in a sample apartment. 142

Fig. 8.6. The occupancy patterns of two sample days from the same apartment. 143

Fig. 8.7. The schematic diagram of the transfer learning process. 144

Fig. 8.8. The distribution of the number of clusters for use in k-means clustering algorithm.... 146

Fig. 8.9. The centroids of a sample household with two clusters. 146

Fig. 8.10. The centroids of a sample apartment with four dominant cluster centroids. 147

Fig. 8.11. The occupancy patterns of a) a sample test case and b) its most similar train apartment. 148

Fig. 8.12. The median jumpstart, asymptotic, and total performance of the conventional control (RL) and the proposed transfer learning method (TL). 149

Fig. 8.13. The distribution of the average temperature deviation, deviation period, and energy consumption in the first episode of training for the conventional RL control and the proposed TL. 150

List of tables:

Table 2.1. Previously published review papers on the topics related to occupant behavior-based control in buildings. 8

Table 2.2. Summarization of occupancy-based control strategies proposed in the literature....... 11

Table 2.3. Summarization of occupancy models used in the reviewed papers........ 15

Table 3.1. Pearson correlation coefficients between the input variables and the output. 41

Table 3.2. The economic parameters utilized for the financial analysis in this study. 46

Table 3.3. The range of hyperparameters used in the tuning process as well as the optimal values obtained........ 48

Table 3.4. The mean NPV and MissTime for RB control systems used in the decision-making process as well as the relative closeness index achieved from the TOPSIS method. 51

Table 3.5. The average performance indices for the selected RB control systems in different setback bounds. 53

Table 3.6. On-peak and mid-peak energy consumption using RB control with different setback temperatures. 54

Table 5.1. The summary of the parameters utilized for developing the MLP and GRU models. 68

Table 5.2. The mean, standard deviation, and ranges of hyperparameters utilized for developing the occupancy models. 76

Table 5.3. The average of the MAE, accuracy, and the overall performance score (relative closeness) for each occupancy model. 82

Table 6.1. The attributes utilized for developing the occupancy prediction models in the literature. 90

Table 6.2. Information about the type of apartments and occupancy ratio based on 30-min intervals. 91

Table 6.3. The candidate features utilized in the FS process. 93

Table 6.4. The parameters utilized in the MOGA and SOGA. 97

Table 6.5. Mean and standard deviation of the occupancy prediction test accuracy for the MA and TOPSIS methods........ 103

Table 7.1. The features utilized to represent the state-space for the agent. 113

Table 7.2. The structural parameters of the function approximator. 116

Table 7.3. Hyperparameters of the DDQN model with the associated search ranges for the tuning process........ 118

Table 7.4. The parameters utilized for developing the GA........ 123

Table 7.5. Hyperparameters of the ML models employed as building models. 124

Table 7.6. The characteristics of the virtual building and the conditions under which the simulation is performed. 126

Table 7.7. The hyperparameters of the DDQN model........ 128

Table 8.1. The state space, action space, and reward function defined for training the RL agent. 139
Table 8.2. The structural parameters of the DDQN algorithm. 140

Chapter 1: Introduction

1.1. Motivation

The upward trend of energy consumption in buildings has become a significant concern worldwide. Globally, buildings account for more than one-third of the total energy consumption, and it is even expected to increase in the future [1]. Energy-efficient heating, ventilation, and air-conditioning (HVAC) control has drawn considerable attention to increase building energy performance [2]. In addition to energy saving, control strategies give particular attention to occupants' comfort, as people spend the majority of their time indoors [3]. However, energy saving objectives are often conflicting with the occupants' comfort and indoor air quality requirement, causing a challenging optimization problem [4].

In order to make a trade-off between these objectives, information about occupancy schedules plays an essential role in control systems [5]. In other words, information about occupancy states (i.e. presence/absence states) can be employed for regulating appropriate setback temperature to save energy during unoccupied periods while providing an acceptable level of thermal comfort for occupants upon their arrival [6]. The use of occupancy information helps the controller to avoid conditioning vacant indoor environment and energy waste [7].

Despite the key role that occupancy information plays in improving the performance of buildings, dynamic occupancy information has been mostly neglected by researchers in this field. Not taking this type of occupant behavior into account can cause unnecessarily conditioning during vacancy periods, and over-ventilation due to maximum occupancy assumptions in ventilation control [7,8]. Dynamic occupancy changes are mostly ignored in commercial buildings by implementing static occupancy profiles as a function of building types and typical working hours [9]. In contrast to commercial buildings where facility managers often define occupancy schedules [10], in residential buildings, HVAC systems can be usually adjusted to occupancy states through direct interventions of occupants. However, users often mis-program their thermostats or entirely neglect the programming features, making such thermostats useless or even causing more energy consumption in many cases [11,12]. In addition, manually defined schedules by occupants often differ from the actual schedules, which can cause thermal discomfort, especially upon their arrival, and additional energy use [11,13]. For example, due to unexpected events that might occasionally occur, occupants might return home before the preheating process is fulfilled or long after the building is fully preconditioned based on the predefined temperature schedules. Such events can cause thermal discomfort and energy waste in many cases.

Reactive control has been proposed to eliminate the need for occupant interventions in the control system to minimize the mentioned limitations. Such systems adjust indoor temperature to the current occupancy state. They can also adjust the ventilation rate to the number of people to avoid over-ventilating the zone. However, the occupant count and ventilation control are out of scope of this study. This study focuses on optimally scheduling indoor air temperature with the consideration of dynamic changes in occupancy states.

1.2. Problem statement

Despite the benefits that reactive controllers provide, they suffer from a major limitation, which is due to the thermal mass of buildings and lag time of HVAC systems. More specifically, as soon as a zone becomes occupied, the reactive control initiates the HVAC operation to reach the desired temperature. However, the time required to provide an acceptable temperature depends on the weather conditions, building material, and HVAC system. This time lag can cause thermal discomfort for occupants during such transition periods.

Researchers have made attempts to address the mentioned limitation of reactive control by proposing predictive control as advanced algorithms. Such control systems take advantage of occupancy models to dynamically forecast the future occupancy schedules. Theoretically, they predict the future occupancy states and prepare indoor environment in advance to minimize thermal discomfort and maximize energy saving. Model predictive control (MPC) has been one of the promising control approaches for occupancy-based HVAC operation in the literature. Using a receding time horizon as well as optimization algorithms, MPC demonstrated the ability to well address non-linearities and disturbances, such as weather and occupancy changes, in the control system. However, implementing the predictive power and optimization algorithms adds great deal of complexity in the control process. These complications often require expensive occupancy monitoring networks, IoT devices, and data storage. Therefore, it is of vital importance for such predictive systems to justify the added complexities by providing superior performance than the conventional ones. There is a need for a comprehensive evaluation of the predictive control performance from the perspectives of financial profitability, energy saving, and peak-demand management, when compared with reactive control.

One of the complications of the predictive control is related to data collection, data storage, and data management for developing occupancy models. In the literature, a variety of different features have been proposed that might impact the performance of occupancy prediction. The features include time of day, weekends/weekdays, day of the week, CO2 concentration, previous occupancy states, lighting, and appliance electricity consumption. The problem is that implementing too many or too few features can, respectively, result in overfitting or underfitting problems [14]. Considering many different candidate features, a wise selection of attributes is essential as it can improve the prediction accuracy, provide a better understanding of the underlying process and enhance the computational speed [15]. Additionally, as mentioned in [16], finding the most relevant features in developing an optimal model can minimize the cost of purchasing sensors and the storage expenses. However, despite its importance in developing data-driven models, feature selection was generally neglected, and it is still unclear what number and what types of attributes should be selected for future occupancy prediction.

The occupancy prediction models can be considered as the heart of the predictive control. Consequently, the accuracy of the models has considerable impact on the performance of the control system. It was reported that a 10% reduction in prediction errors would result in up to almost 9% and 16% improvement in energy saving and thermal comfort, approximately [17]. Although many occupancy models have been developed in earlier research to improve the prediction performance, there is a need to find the best performing models.

In addition to the mentioned complexities in implementing advanced predictive control, there are other factors preventing them from being practical in the building sector. These barriers include the need for accurate models of buildings [18], substantial knowledge and expertise for developing such models, and their poor potential of being generalized for different cases [19]. To deal with such issues, model-free data-driven control systems, such as reinforcement learning (RL) algorithms, have been proposed as an alternative control method. In this method, the controller interacts with the environment and tries to learn the optimal decisions through experience [20]. As a result, control decisions can be made without a need for developing and calibrating models of buildings and the occupancy [21]. Despite their successful operation in different applications [22–26], there is a lack of information on their performance when integrated with dynamic occupancy states in buildings.

From a practical point of view, the described model-free control is prone to causing a poor performance during the initial learning phase because of their trial-and-error learning process. This nature of the algorithms can put thermal comfort of occupancy in a high level of risk during the training period. As demonstrated in Ref. [27], the algorithms might take up to multiple years to be trained before finding the optimal policy. Because of the importance of indoor environment thermal comfort, such wrong decisions are not tolerated in practice and can limit their applications.

Based on the mentioned research gaps and limitations related to occupancy-based HVAC control systems, the following research questions arise:

1. What are the benefits and drawbacks of including dynamic occupancy information in HVAC control in terms of financial profitability, energy saving, and peak-demand management?
2. What predictor variables are necessary to develop the occupancy prediction models?
3. Which types of occupancy modelling methods provide superior thermal comfort and energy saving when integrated with predictive control in buildings?
4. What are the strengths and limitations of the model-free data-driven control, compared with the conventional MPC?
5. How to minimize the training duration of the model-free control systems to avoid thermal discomfort and energy waste?

1.3. Research objective

The primary objective of this research work is to develop a self-learning data-driven control methodology that learns occupancy patterns and building characteristics without a need for development of online occupancy prediction models and building models. The proposed framework leverages the experience of pre-trained controllers to minimize the training duration of the new controllers that are applied to unseen buildings. The self-learning nature of the system is intended to enhance the generalization potential of optimal occupancy-based control systems for use in different buildings. The objectives of this study can be summarized as follows:

1. To develop a control framework to leverage the experience of previously trained control systems for reuse in new unseen buildings to minimize the training period,
2. To develop a self-learning control system to schedule optimal indoor temperature by consideration of dynamic occupancy patterns, weather conditions, and building characteristics,

3. To develop an optimal feature selection method to determine the most valuable predictor variables in occupancy prediction by minimizing redundancy and irrelevancy in the feature set,
4. To evaluate the performance of occupancy-based control in terms of energy, financial profitability, and peak-demand metrics, and
5. To determine the most effective data-driven occupancy models for predicting future occupancy patterns.

1.4. Organization

The rest of the dissertation is structured as follows: Chapter 2 is devoted to a comprehensive literature review covering the previous studies on occupancy-based HVAC control. It includes a description of the control systems and explore their limitations. Additionally, statistics are also provided to demonstrate the limitations and propose potential research directions in this field. Chapter 3 discusses the potential profitability of the occupancy-based HVAC operation through a comprehensive financial analysis and shed light on their limitations from demand side management point of view. In addition, the influences of using online estimations of preconditioning time on the control system are quantified. In Chapter 4 and 5, the impact of selecting the occupancy prediction approach and selecting different machine learning models on the performance of the control system is quantified. Additionally, the correlations between conventional performance assessment merits with energy efficiency and thermal comfort are measured. Chapter 6 is devoted to exploring the optimal feature subsets using advanced feature selection methods. In Chapter 7, the self-learning control system is developed and its performance is evaluated and compared with an MPC algorithm. In Chapter 8, a transfer learning framework is proposed based on unsupervised learning to accelerate the learning process in new unseen buildings. Ultimately, summary, conclusions, and the limitations of the current work are discussed in Chapter 9 to provide future research directions.

1.5. Type of the current dissertation

The current dissertation is written in a manuscript-based structure, in which most of the following chapters are the manuscripts of scientific papers. The manuscripts have been published or under preparation for scientific journals and conferences:

Chapter 2:

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

Chapter 3:

M. Esrafilian-Najafabadi, F. Haghighat, Occupancy-based HVAC control using deep learning algorithms for estimating online preconditioning time in residential buildings, Energy and Building 252 (2021) 111377. https://doi.org/10.1016/j.enbuild.2021.111377.

Chapter 4:

M. Esrafilian-Najafabadi, M. Babahaji, F. Haghighat, Deep learning models for future occupancy prediction in residential buildings, in: 5th International Conference on Building, Energy and Environment 25th-29th July, 2022.

Chapter 5:

M. Esrafilian-Najafabadi, F. Haghighat, Impact of occupancy prediction models on building HVAC control system performance: Application of machine learning techniques, Energy and Building 257 (2022) 111808. https://doi.org/https://doi.org/10.1016/j.enbuild.2021.111808.

Chapter 6:

M. Esrafilian-Najafabadi, F. Haghighat, Impact of predictor variables on the performance of future occupancy prediction: Feature selection using genetic algorithms and machine learning, Building and Environment (2022) 109152. https://doi.org/10.1016/J.buildenv.2022.109152.

Chapter 7:

M. Esrafilian-Najafabadi, F. Haghighat, Towards self-learning control of HVAC systems with the consideration of dynamic occupancy patterns: Application of model-free deep reinforcement learning, (2022) (*Submitted*)

Chapter 8:

M. Esrafilian-Najafabadi, F. Haghighat, Transfer learning for model-free HVAC control based on unsupervised learning of occupancy profiles and deep reinforcement learning, (2022). (*Under preparation)*

Chapter 2: Occupancy-based HVAC control systems in buildings: A state-of-the-art review[1]

2.1. Overview

Intelligent buildings have drawn considerable attention due to rapid progress in communication and information technologies. These buildings can utilize current and historical data, collected from occupancy detection and monitoring networks, to predict occupancy profiles and adjust heating, ventilating, and air conditioning (HVAC) operations accordingly. This adjustment aims to minimize the energy consumption of HVAC systems while maintaining an acceptable level of thermal comfort and indoor air quality. To provide a trade-off between these conflicting objectives, a variety of occupancy-based control strategies have been proposed in the literature. The present chapter aims to review the research works concerning occupancy-based control systems, classify them based on the integration of occupancy information with control systems and identify their strengths and limitations. Finally, research gaps in this field are discussed from different aspects, including performance evaluation metrics, control methods, occupancy models, and building types. Future research directions are also proposed to fill the identified gaps.

2.2. Introduction

2.2.1. Motivation for occupancy-based HVAC control

The upward trend of energy consumption in buildings has become a significant concern worldwide. Globally, buildings account for more than one-third of the total energy consumption, and it is even expected to increase in the future [1]. To deal with this issue, advanced controllers have drawn great attention as promising solutions to increase building energy performance [2]. In addition to energy saving, control strategies give particular attention to occupants' comfort, as people spend the majority of their time indoors [3]. However, energy saving objectives are often conflicting with the occupants' comfort and indoor air quality requirement, causing a challenging optimization problem [4]. To address this problem, researchers often apply strict constraints to the decision variables according to thermal comfort criteria or employ multi-objective optimization algorithms that can consider both energy and comfort objectives simultaneously. In order to make a trade-off between these objectives, information about occupant behavior plays an essential role in control systems [5]. Information about occupancy states (i.e. presence/absence states) can be employed for regulating appropriate setback temperature to save energy during unoccupied hours while providing an acceptable level of thermal comfort for occupants upon their arrival [6]. Indeed, conditioning vacant building indoor environment can bring about unnecessary run time of HVAC operation and consequently, cause energy waste [7]. Masoso et al. [28] emphasized the importance of unoccupied hours in overall energy consumption in offices and reported that more than half of the overall energy was consumed during unoccupied hours in their cases.

---

[1] This chapter is based on the following publication: M. Esrafilian-Najafabadi, F. Haghighat, Occupancy-based HVAC control systems in buildings: A state-of-the-art review, Building and Environment 197 (2021) 107810. https://doi.org/10.1016/j.buildenv.2021.107810.

Despite the key role that occupancy information plays in improving the performance of buildings, analyzing studies on HVAC control systems reveals that dynamic occupancy patterns, as a kind of occupant behavior in buildings, have been mostly neglected by researchers in this field. Not taking this type of occupant behavior into account can cause unnecessarily conditioning during vacant hours, and over-ventilation due to maximum occupancy assumptions used in ventilation control [7,8]. Dynamic occupancy patterns are mostly ignored in commercial buildings by implementing static occupancy profiles as a function of building types and typical working hours [9]. In contrast to commercial buildings where facility managers often define occupancy schedules [10], in residential buildings, HVAC systems can be often adjusted to occupancy through direct interventions by occupants. However, users often mis-program their thermostats or entirely neglect the programming features, making such thermostats useless or even causing more energy consumption in many cases [11,12]. In addition, manually defined schedules by occupants often differ from the actual schedules, which can cause thermal discomfort, especially upon their arrival, and additional energy use [11,13]. For example, due to unexpected events that might occasionally occur, occupants might return home before the preheating process is fulfilled or long after the building is fully preconditioned based on the predefined temperature schedules, which can respectively cause thermal discomfort or energy waste.

In order to overcome the mentioned shortcomings, control systems can automatically infer and predict occupancy patterns. Thanks to the Internet of Things (IoT) devices in smart buildings, capturing these relationships have become more possible by employing data mining and machine learning approaches [29]. One of the current trends of HVAC research is to make systems aware of current and future occupancy information through learning behaviors.

2.2.2. Previous literature reviews

Table 1 summarizes the review articles concerning occupant behavior-based HVAC control systems. Although these articles are high-quality reviews, there is still a gap for a review paper focusing on the state-of-the-art occupancy-based HVAC control systems. On one hand, some of these articles concentrated only on a few aspects of these systems. For example, articles [6,30–32] only focused on occupancy detection and modeling in buildings, neglecting the integration of occupancy information with control systems. These papers provided no insights into the challenges and limitations related to the control of occupancy-based HVAC systems. On the other hand, some other articles provided comprehensive reviews, covering various topics related to building control based on occupancy patterns, including lighting control, HVAC control, comfort-aware HVAC systems, occupancy detection methods, and occupancy models. It should be noted that these topics have been active research areas in recent years, and there have been a large number of papers published in this field. As a consequence, these review papers provided a general overview of the previously published papers in this field and identifying the strengths and limitations of the methodologies were mostly neglected.

Limited review work has focused on occupancy-based HVAC control systems. The closest review to this work was done by Mirakhorli and Dong [33], where they reviewed the research works on occupancy-based model predictive control (MPC). However, they mostly ignored other types of occupancy-based control strategies, such as reactive and rule-based control. Additionally, this paper was published more than 4 years ago. As demonstrated in Fig. 2.1, since then, a relatively large number of research papers have been published in this field. This growing trend highlights the necessity of an updated review work to explore the recent state-of-the-art methods.

Table 2.1.
Previously published review papers on the topics related to occupant behavior-based control in buildings.

Reference	Year	Research focus
Nguyen and Aiello [34]	2013	Intelligent buildings with occupancy models used for optimal control of lighting, HVAC, and appliances.
Mirakhorli and Dong [33]	2016	Control strategies in occupancy-centric buildings, focusing on model-based predictive control (MPC).
Shen et al. [6]	2017	State-of-the-art methodologies for occupancy detection and monitoring in office buildings.
Naylor et al. [30]	2018	Strengths and limitations of using different types of occupancy detection methods in buildings.
Salimi and Hammad [35]	2019	Occupancy detection techniques, occupancy models, and control of HVAC and lighting in office buildings.
Jung and Jazizadeh [36]	2019	Comfort-aware HVAC operation, occupancy detection, occupancy models, and occupancy-based control strategies.
Park et al. [37]	2019	Field evaluation studies on occupancy-centric control strategies.
Sun et al. [31]	2020	Occupancy measurement approaches with a focus on vision-based techniques.
Azar et al. [38]	2020	Design procedure of occupant behavior-based control strategies.
Dai et al. [32]	2020	Prediction of occupancy and window-opening behavior using machine learning algorithms.

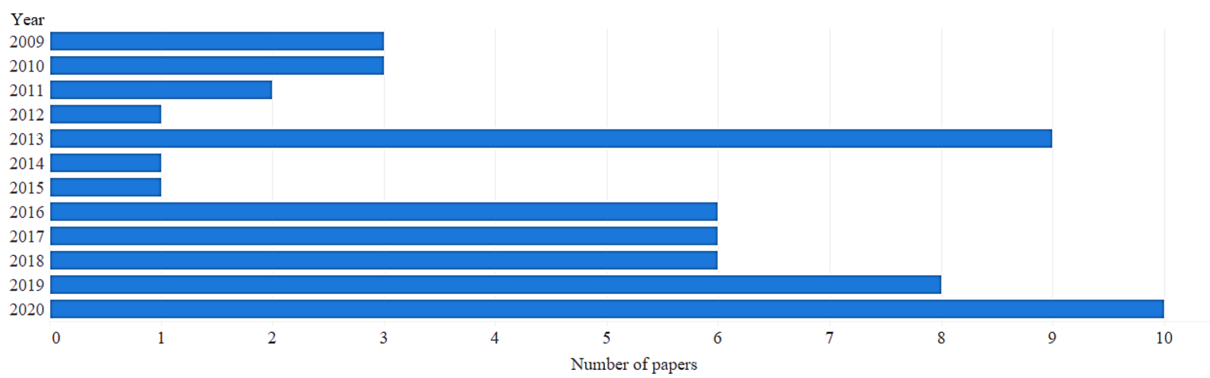


Fig. 2.1. Distribution of reviewed papers based on publication year.

2.2.3. Contributions and structure

This chapter aims to address the limitations of earlier review papers by providing a focused review on the state-of-the-art occupancy-based HVAC control strategies. The ultimate target is to utilize the results of this thorough literature review to inspect the current research gaps in this field and accordingly, propose research directions for future work. For this purpose, the reviewed papers are first grouped into different categories according to the incorporation of occupancy information in control systems. The papers in each category are reviewed in detail from different perspectives, including performance evaluation methods, feature use for occupancy models, occupancy models, types of testbeds, and occupancy detection and monitoring techniques. Then, the reviewed papers are summarized and discussed from each perspective to reveal the research limitations from different dimensions.

The rest of this chapter is structured as follows. In Section 2.3, the reviewed papers are classified into different categories for a detailed review. The research items are rigorously reviewed in Sections 2.4-2.6 and their limitations and strengths are discussed. Next, the main research gaps are identified and discussed in Section 2.7. In Section 0, conclusions are made by summarizing the findings and contributions.

2.3. Research classification

As shown in Fig. 2.2, occupancy-based HVAC control strategies are classified into two main categories based on the integration of occupancy information in control systems: control strategies based on user-defined schedules, and occupancy detection and monitoring. In the former category, occupancy information is manually included by users, and HVAC systems act according to user-defined schedules. This category is further classified into three subcategories based on their flexibility to receive occupancy schedules: manual, programmable, and scheduled thermostats. These control strategies will be discussed in Section 2.4. The control systems in the latter category are integrated with the occupancy detection and monitoring networks to automatically capture occupancy information without the need for occupants' interventions. These controllers are further classified into reactive and predictive control. Reactive approaches act based on real-time occupancy information in the monitored building. The performance of these strategies is mainly dependent on occupancy detection and types of sensor networks implemented to infer occupancy. In addition to using real-time information, predictive approaches also take advantage of occupancy models to provide insight into future occupancy states. Such strategies are also called proactive, as they can be prepared in advance before future events happen. As well as occupancy detection systems, their performance is also dependent on occupancy prediction models. These strategies are divided into rule-based, and optimal control categories, which will be discussed in Section 2.6.

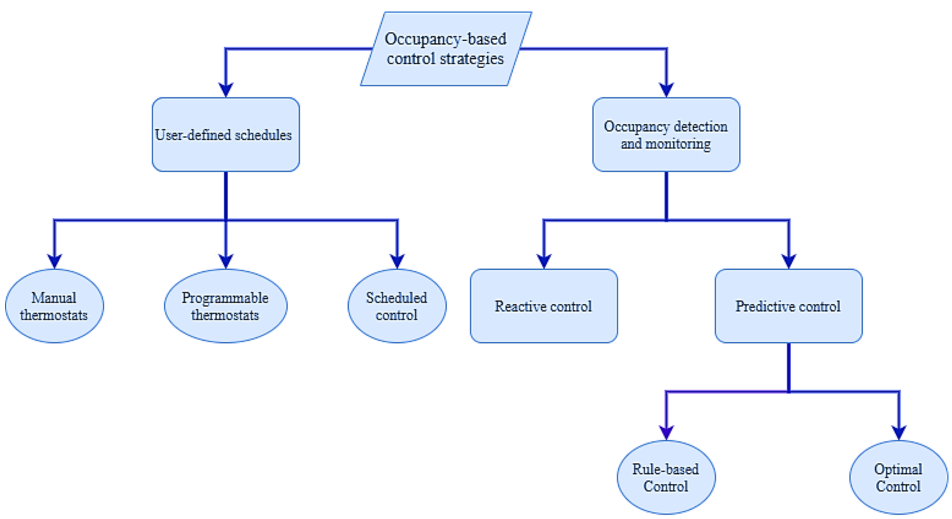


Fig. 2.2. Classification of occupancy-based HVAC control systems based on types of control strategies and the integration of occupancy information in systems.

2.4. HVAC operation based on user-defined occupancy schedules

The integration of occupancy information with control systems differs depending on building types. In commercial buildings, occupants usually lack access to thermostat control for changing the setting [39]. It is usually the responsibility of facility managers to define setback or setpoint schedules in commercial buildings [10]. Fixed occupancy profiles are often defined by facility managers based on the type of buildings and its functionality. In this study, this type of control that is based on fixed pre-defined occupancy schedules is called scheduled control. The shortcoming of this control approach is that the static schedules can be different from the actual ones in many cases. The discrepancy between these schedules can cause unnecessary conditioning during unoccupied hours and over ventilate during partial occupancy.

In contrast, occupants in residential buildings can control their indoor environment by directly interacting with thermostats, opening or closing windows, and changing their clothing. Using manual and programmable thermostats, occupants can define a setback temperature according to their occupancy patterns and need. However, due to the interactive nature of these kinds of thermostats, they do not necessarily lead to energy saving. As demonstrated in [12], using manual and programmable thermostats can even negatively impact the energy performance as compared with always-on thermostats. According to several field evaluations covering more than 700 homes in [40], employing programmable thermostats results in no significant energy saving or even can increase energy usage depending on occupants' behavior. In some other studies, a slight improvement in energy saving from programmable thermostats was reported. For example, an average of 6% and 3.6% were reported in [41,42]. In addition to the poor energy saving ability of such thermostats, the mismatch between pre-defined schedules and the real daily occupancy patterns can also cause thermal discomfort especially upon arrival [13].

Fig. 2.3. shows the popularity of thermostat types among US households in 2015 [43]. It can be observed that programmable thermostats were the least popular and accounted for 5% and 20% of households with individual and central air conditioning units, respectively. In households with individual units, manual thermostats were the most popular type at 64% of households, while the most preferred approach for people with central units was to keep the thermostat unchanged.

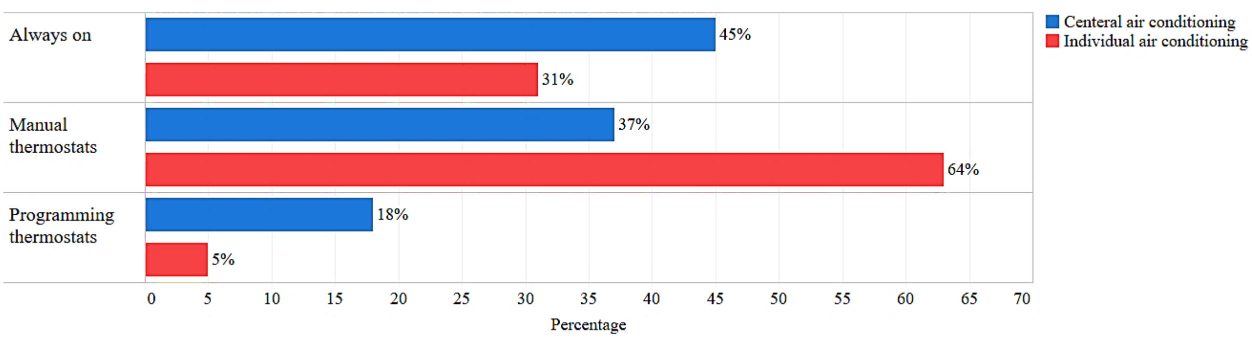


Fig. 2.3. The proportion of U.S. households using different types of thermostats with central or individual air conditioning system [43].

With the advent of IoT devices in smart buildings, the remote-control of HVAC systems have gained a great popularity. Koehler et al. [44] conducted a field evaluation and a survey to evaluate the performance of remote controllers. In this study, they developed an eco-feedback system on occupants' smartphones. The system was employed by 10 participants over a three-month period. The participants were provided with a control interface through Android applications so that they could control the thermostat from any location. To provide occupants with a guideline to help them with saving more energy, this application sent them frequent real-time feedback about their energy consumption patterns and possible energy saving strategies. The application compared daily energy consumption patterns and demonstrated the effects of their behavior on energy consumption. They defined two performance criteria: "UnnecessaryHeating" and "LostComfort".

These two indicators were utilized in the literature using different terminologies. In this study, all the indicators with the same concept are called "WasteTime" and "MissTime" based on the definitions in [45,46]. Throughout this chapter, WasteTime and MissTime are respectively defined as the average of unoccupied hours when the indoor temperature deviates from the setback, and the average of occupied hours when indoor temperature deviates from the desired setpoint. According to the results, an average of 72.5% of the overall testing period, setpoint and setback temperature were correctly set by occupants. It was indicated that this remote-control system on average ended up with almost 5 hours of WasteTime, and 73 minutes of MissTime for occupants. Forgetting to set the temperature to the recommended setback accounted for 96% of the overall WasteTime.

2.5. Reactive control

The main shortcoming of the control strategies using user-defined occupancy schedules is linked with manual interventions required from occupants. This issue made programmable thermostats useless in many cases, and as a consequence, Energy Star suspended its certification on programmable thermostats in 2009 [47]. The following findings were the main reasons preventing them from being widely recommended:

- Complexity for some users to effectively program it.
- Uncertainties in occupants' plans that make a discrepancy between actual and programmed schedules.
- Unwillingness of occupants to put effort into interacting with the thermostats.

In order to solve the main problem of user-defined schedules, i.e., manual interactions required by users, reactive control systems started to increasingly gain attention. These systems minimize manual interventions while improving performance [48]. As the name suggests, reactive control automatically reacts to real-time occupancy information received from occupancy detection and monitoring systems to dynamically change the temperature setpoint. In addition, they can make use of occupancy levels (i.e. counts) to adjust ventilation rates. Researchers have implemented a variety of different occupancy detection instruments such as passive infrared (PIR) sensors, CO_2 sensors, electricity meters, door reed switches, chair movement sensors, Wi-Fi, cameras, and Bluetooth devices depending on the type of building and privacy issues [7,49–53]. However, providing details about these methods is out of the scope of this review study. Interested readers in occupancy detection technologies are referred to the related review articles summarized in Table 2.1.

Table 2.2.
Summarization of occupancy-based control strategies proposed in the literature.

Ref.	Building type	Occupancy detection	Control				Performance evaluation		
			Type	V[1]	H[2]	C[3]	Method	Performance	Baseline
[54]	Residential	PIR	Reactive control	-	✓	✓	Field evaluation	Up to 9% energy saving on regular days and up to 30% during long vacancy periods.	Manual
[50]	Office	Wi-Fi network	Reactive control	✓	✓	✓	Field evaluation	17.8% Energy saving No thermal discomfort	Scheduled
[52]	Office	Door reed switches PIR	Reactive control	-	-	✓	Simulation EnergyPlus	15% energy saving	Scheduled
[55]	Residential	-	Reactive control	-	✓	✓	Field evaluation	17% decrease in the runtime of HVAC	Programmable
[56]	Residential	PIR	Reactive control	-	✓	-	Field evaluation	14.4% energy saving	Whole building control

[57]	Office	-	Reactive control	-	-	✓	Simulation EnergyPlus	Up to 28.3% energy saving	Scheduled
[58]	Residential	Survey	Reactive control	-	✓	✓	Simulation EnergyPlus	20% energy saving Payback period of one year	Always-on Programmable
[59]	Residential	-	Reactive control	-	✓	✓	Simulation DOE2	Up to 38.7% decrease in energy consumption 34.7% reduction in peak demand	Always-on
[60]	Office	-	RBC[4] using occupancy probability	-	✓	✓	Simulation EnergyPlus	Up to 50% electricity saving Up to 66 million fossil fuel saving 0.9-3.7 million metric tons CO2 reduction 168-658 million dollars saving in energy cost	Always-on Programmable
[13]	Residential	GPS Cellular network Wi-Fi	RBC using occupancy probability	-	✓	✓	Simulation EnergyPlus	28% energy saving 48 min MissTime	Programmable Reactive Perfect prediction
[61]	Office	Motion sensors	RBC using occupancy probability	-	-	✓	Field evaluation	7-52% energy saving.	Scheduled
[45]	Residential	RFID tags Motion sensors	RBC using conditioning rate	-	✓	-	Field evaluation	No energy saving compared to baseline energy consumption. Up to 92% MissTime reduction	Programmable
[62]	Residential	GPS	RBC using preconditioning time				Field evaluation	7% energy saving.	Manual Programmable
[44]	Residential	GPS	RBC using preconditioning time	-	✓	-	Field evaluation	MissTime of 0.95 hr WasteTime of 0.94 hr	Manual GPS-thermostat PreHeat
[63]	Office	Cellular network Wi-Fi	RBC Preconditioning time	-	✓	-	Field evaluation	26% energy saving compared	Always on Reactive
[64]	Office	Thermal sensors PIR	RBC using preconditioning time	✓	✓	✓	Simulation EnergyPlus	Up to 25% energy saving Maximum RMSE of 0.415° C in temperature violation	Scheduled Reactive Setpoint control without ventilation
[65]	Conference hall	Acoustics Lighting PIR CO2 Temperature Humidity	RBC using preconditioning time	-	-	✓	Simulation EnergyPlus	Up to 30% saving	Scheduled Meeting schedules Reactive
[66]	Office	Cameras	RBC using preconditioning time	✓	✓	✓	Simulation EnergyPlus	42% less energy consumption	Scheduled Reactive Setpoint control without ventilation
[67]	Office	Cameras	RBC using preconditioning time	✓	✓	✓	Field evaluation Simulation	26% energy saving for field evaluation and up to 30% based on yearly energy simulation	Scheduled Reactive Setpoint control without ventilation Perfect prediction
[17]	Office	PIR Ultrasonic sensors	RBC using preconditioning time	-	✓	✓	Simulation EnergyPlus	Up to 28% energy saving Up to 40% improvement in thermal comfort	Reactive
[46]	Residential	Energy meters	RBC using preconditioning time	-	✓	✓	-	0.42 kWh energy saving per day 44.28 min decrease in mismatch time	Programmable
[68]	Residential	Applications on smartphones	RBC using preconditioning time	-	✓	-	Simulation RC	Up to 26.2% energy saving 28.2 hr and 132.1 hr for rule-based and reactive thermostats respectively.	Always on Reactive
[69]	Office	Bluetooth tags	Optimal control	-	✓	✓	Simulation EnergyPlus	2% energy saving 50% decrease in thermal discomfort	Scheduled
[70]	Office	-	Optimal control (MPC)	-	✓	✓	Simulation RC	Negligible energy saving for including occupancy prediction in MPC	MPC without occupancy information
[71]	Office	-	Optimal control (MPC)	-	✓	✓	Simulation RC	Negligible energy saving for including occupancy prediction in MPC	MPC without occupancy prediction

[72]	Residential	Light sensors	Optimal control (MPC)	-	✓	-	Simulation RC	32.80% decrease in temperature violation	MPC without occupancy prediction
[73]	Office	-	Optimal control (MPC)	-	✓	✓	Simulation ThermalSim	Robustness to errors	MPC without PEC
[74]	Residential	-	Optimal control (MPC)	-	-	✓	Simulation RC	8% energy saving	Conventional MPC
[75]	Residential	Applications on smartphones Surveys	Optimal control (MPC)	-	✓	✓	Simulation EnergyPlus	Up to 13.3% energy saving.	Always on Rule-based Reactive
[76]	Residential	-	Optimal control	-	-	-	-	MissTime: Up to 40% reduction Conditioned time: 15% decrease	Programmable

[1]Ventilation, [2]Cooling, [3]Heating, [4] Rule-based control

A summary of the reactive control algorithms reviewed in this chapter can be found in Table 2.1. Among these studies, some researchers employed reactive control in buildings while keeping the investment cost low by avoiding large-scale retrofits such as new rewiring systems. To this aim, Agarwal et al. [52] proposed a reactive control strategy based on low-cost magnetic door reed switches and PIR sensors in office buildings. This reactive control responded to occupancy states achieved from a combination of these two types of sensors. In their study, setpoint and setback temperatures were selected as 22.9 °C and 26.1 °C, respectively. Using EnergyPlus [77] to simulate office buildings, they reported that this reactive control reduced energy consumption by 15%, compared with scheduled control as the baseline. Although the occupancy detection system required a low-cost infrastructure, it was prone to providing lower accuracy than more complex systems. As occupancy detection performance is the most essential factor in reactive controllers, decreasing accuracy can cause thermal discomfort and higher energy consumption [52]. However, in this study, the performance of occupancy detection was not evaluated to demonstrate the effectiveness of the system. Balaji et al. [50] used the existing WiFi networks to identify occupants and their locations for reactive control. This network provided an accuracy of 86%, without a need for any additional investment in infrastructure. They showed that integrating the occupancy detection algorithm with HVAC control systems could result in 17.8% electricity saving. It was also reported that no considerable thermal discomfort was caused using this proposed system. It was because of using a highly conservative temperature setback during unoccupied hours to guarantee thermal comfort upon arrival. However, using highly conservative setbacks limits the system energy saving ability.

Stopps and Touchie [55] conducted a field experiment to monitor HVAC runtime, controlled by programmable and smart thermostats. The experiment was conducted using field data gathered from 56 thermostats in two different apartment buildings. To achieve a fair comparison between the thermostats, each day during the monitoring period was randomly associated with controllers. In addition, a random forest algorithm was trained using weather data to provide an estimation of baseline run time in those days with smart thermostats running. The results showed that using smart thermostats reduced HVAC runtime by an average of 17%. However, the direct relationship considered between HVAC run time and energy consumption can be mentioned as a limitation of this study. According to the field evaluation performed by Pritoni et al. [54], implementing HVAC run time as a performance criterion can cause overvaluation of the performance. They monitored 2500 rooms in three university residence halls before and after retrofitting them with smart thermostats. Before the retrofit, all rooms were equipped with manual thermostats, which were selected as the baseline. Smart thermostats, installed after the retrofit, utilized real-time occupancy information from PIR sensors to learn appropriate setback temperatures that can be recovered in an acceptable period of time. It was reported that due to heat flows from corridors and neighboring rooms, room temperature rarely reached the defined setback despite having long unoccupied hours with HVAC systems turned off. In other words, although the run time was significantly reduced in one vacant zone, an extra energy load was induced to the HVAC system because of the interzonal heat flows between occupied and unoccupied zones. As a result, consideration of HVAC run time as an energy indicator can cause overestimation of the performance.

A limitation of previously reviewed studies is that they mostly focused on whole-building control strategies. Indeed, occupancy patterns often vary across different rooms in buildings and the operation of HVAC systems can be adjusted to these zonal patterns. As highlighted by Kim and Oldham [78], using zoned temperature and humidity control in different hotel rooms can improve occupants' thermal comfort. Yang and Becerik-Gerber [57] demonstrated that considering a unique optimal schedule for each individual zone led to 5% more energy saving, compared with employing a shared optimal profile in the whole building. Sookoor et al. [56] developed a zoned reactive HVAC control strategy, called RoomZoner, and compared its performance with that of a whole-building control approach as the benchmark. RoomZoner was designed based on the concept of micro-zoning that was utilized by Rose et al. [79]. In this algorithm, the airflow from HVAC systems to each zone was dynamically changed based on the current temperature and occupancy states in each room. In order to keep the initial cost as low as possible, they developed a low-cost occupancy inferring system using PIR motion sensors installed in each zone. They assessed RoomZoner performance based on a 42-day field evaluation. It was demonstrated that using this zonal control resulted in a 14.4% reduction in energy consumption.

In the reviewed previous work, the proposed approaches were evaluated in terms of energy saving and thermal comfort, neglecting other performance indicators such as economic and peak-shifting criteria. To fill this gap, Krarti [59] evaluated the peak-shifting ability of zonal reactive thermostats in residential buildings. They reported that these thermostats decreased peak demand and energy consumption by up to 34.7% and 38.7%, respectively compared with an always-on thermostat. Wang et al. [58] conducted a financial analysis to evaluate the economic merits of reactive control. Always-on and programmable thermostats were considered as baselines. They used American Time Use Survey [80] to construct a probability function to involve occupancy patterns in the development of the control strategy. To construct occupancy profiles, they used random numbers between 0 and 1, which were then compared with the probability of occupancy in each time-interval. They showed that the reactive strategy resulted in an energy saving of 20% and a payback period of less than one year.

2.6. Predictive control

As discussed in the previous section, reactive control strategies are able to overcome the main shortcoming of the control strategies using user-defined occupancy schedules. However, reactive control can cause thermal discomfort during transitions from setback to setpoint temperature upon occupants' arrival. The lag time associated with the HVAC systems to return to the desired setpoint temperature is the main reason for this limitation [81].

To deal with this issue, the control system can take advantage of occupancy prediction models to create proactive control rather than being reactive. Proactive controllers forecast future occupancy patterns and accordingly, precondition buildings before occupancy. However, the performance of these control strategies is highly dependent on the model prediction performance. Indeed, predicting future occupancy in a building is a challenging task as occupant behavior is highly stochastic in nature [61]. Furthermore, it is also challenging to effectively integrate the occupancy models in HVAC control in order to make a trade-off between energy saving and thermal comfort. The predictive control methods and occupancy models developed in the literature are respectively summarized in Table 2.2 and Table 2.3.

2.6.1. Occupancy prediction models

Because of the complexities and importance of occupancy prediction, some researchers focused on developing occupancy prediction models without investigating the integration of the models with control systems. Occupancy models are established based on databases gathered during occupancy monitoring periods. Depending on available data, occupancy models can predict binomial states of future occupancy, estimate occupancy levels, or predict occupancy patterns for each individual occupant.

Table 2.3.
Summarization of occupancy models used in the reviewed papers.

Ref.	Building	Model	Features	Occupancy Level	Performance
[66]	Office	Markov model	Time of day	✓	Duration JSD Occupancy flow
[17]	Office	Hypothetical model	-	-	Accuracy False negative/positive rates
[82]	Office	SVM Random Forest Decision tree kNN	Holiday Time of day Day of week Weekends Season	-	Accuracy F-score
[83]	Office	kNN SVM Linear regression M5-Rules REPTree Gaussian processees Baggigng	Day of week Month Occupancy duration Arrival time Departure time Time of day Temperature Wind speed Cloudiness Precipitation Snowfall	✓	MAE
[84]	Office	KNN-DTW Random forest SVM PreHeat [45]	Time of day Day of week Weekends Season	-	F-score
[85]	Residential	HMM kNN SVM	Zonal occupancy states Time of day Occupancy duration	✓	F-score
[46]	Residential	Proportional model	Time of day	-	Accuracy Matthews Correlation Coefficient
[76]	Residential	Proportional model	Time of day Departure time Arrival time	-	-
[64,67]	Office	Blended Markov model	Time of day	✓	Accuracy
[86]	Residential	PreHeat [45] Proportional model Hybrid model proposed in [87]	Time of day Location	-	Accuracy
[88]	Office	Markov model Semi-Markov model	Season Day of week Time of day State transition	✓	NRMSE

[65]	Conference hall	Semi Markov Model	Acoustic level Lighting level CO2 Concentration Temperature Relative Humidity	-	-
[89]	Office	Genetic programming	Day of week Time of day Occupancy duration	✓	Accuracy
[90]	Office	Markov model	Arrival time Departure time Occupancy duration State transition	✓	NRMSE K-L
[91]	Office	Markov model	Identity Occupancy duration Activity type	✓	Accuracy
[92]	Exhibition hall	LSTM ARIMA RNN Holt-winter	Time of day	✓	RMSE MAE MAPE
[93]	Commercial	GRU	Location Activity type	✓	RMSE MAE MSE
[94]	Research laboratory	LSTM	CO2 concentration	✓	Accuracy RMSE
[75]	Residential	Proportional model Perfect prediction	Time of day Day of week	-	Accuracy
[68]	Residential	Perfect prediction	-	-	-
[69]	Office	Proportional model	Time of day Day of week	-	-
[87]	Residential	Proportional model Drive time	Time of day Day of week		
[95]	Residential	Logistic regression Random forest kNN Markov model HMM LSTM	Time of day Weekends	-	Accuracy Computational time
[96]	Office	Decision tree HMM	Time of day CO2 concentration Appliance energy consumption Lighting level	✓	Accuracy
[60]	Office	Proportional model	Time of day	-	-
[13]	Residential	Proportional model	Time of day	-	Accuracy
[62]	Residential	Drive time	Location	-	-
[44]	Residential	Hybrid of destination prediction and proportional model.	Location Time of day	-	Accuracy
[63]	Office	Hybrid of route classification and a time-aided order-k Markov predictor	Location Time of day	-	Accuracy
[70,71]	Office	Perfect prediction	-	-	-
[72]	Residential	k-means clustering	Time of day Day of week	-	-
[73]	Office	Proportional model	-	-	Accuracy
[74]	Residential	Revised Logistic regression	Time of day Weekends	-	MAE
[61]	Office	kNN k-means clustering	Arrival time Departure time Vacancy duration	-	Accuracy
[45]	Residential	PreHeat	Time of day Weekends	-	Accuracy

2.6.1.2. Occupancy state/level prediction

Occupancy states are usually presented by 0 and 1, respectively denoting vacant and occupied hours, and used to regulate the setback and setpoint temperature. In order to develop models to predict occupancy states, relatively simple monitoring sensors, such as motion detectors, would suffice to provide an acceptable level of accuracy, while for occupancy levels, more advanced infrastructure such as camera networks is often required. Occupancy levels are helpful for controlling ventilation rates, as ventilation rates are adjusted based on the number of occupants in the monitored space.

Chen et al. [90] proposed an inhomogeneous Markov chain model to predict occupancy levels in office buildings based on their preliminary results [97]. Five attributes were considered in developing this model: mean of occupancy level, first arrival time, last departure time, occupied duration, and occupancy transitions. To assess the performance of the developed model, normalized root mean square deviation and Kullback-Leibler (K-L) were employed. They compared the performance of the proposed model with that of an agent-based algorithm proposed by Liao et al. [98,99] and showed the superiority of their proposed Markov models. Adamopoulou et al. [88] developed a model using spatial-temporal features to capture relationships between occupancy patterns in different rooms. They grouped highly correlated rooms into different zones and deployed a unique Markov model for each defined zone. Transition matrices were constructed to provide the probability of interzone occupants' transitions. These matrices were then used in occupancy models to capture spatial relationships. They employed these matrices as well as season, time of day, and day of week as features to develop the model. Depth-image cameras with 95% accuracy were implemented to collect occupancy data in an office and a kitchen while in a rest area, PIR and acoustic sensors were used. A limitation of these reviewed studies is linked with the fundamental assumption considered in Markov chain models. To be more specific, in the Markov models, it is assumed that future states are only dependent on the current state. Using this assumption might neglect some important information existing in the past time intervals for predicting occupancy states.

In order to deal with this issue, Kim et al. [92] established a long short-term memory (LSTM) network, which was able to memorize important information from past events. They used this model to predict occupancy levels in a large exhibition hall and compared its performance with that of an auto regressive integrated moving average (ARIMA), holt-winter, and recurrent neural networks (RNN). They collected data from the exhibition hall in 15-min time intervals using image sensors. Root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) were utilized as evaluation criteria to assess the performance of the deep learning algorithms. According to the results, LSTM model provided the best performance among the implemented machine learning models. However, there is a shortcoming associated with using cameras in data collection, as implementing cameras in buildings could cause privacy issues for occupants.

To protect occupant privacy, some researchers have implemented other alternatives, such as environmental features, in developing occupancy models. Ryu et al. [96] developed a decision tree for occupancy detection using indoor and outdoor CO2 concentration, lighting, and appliance energy consumption. They applied the results of occupancy detection to develop a hidden Markov model (HMM) algorithm to predict future occupancy states and levels in office buildings. According to the results, occupancy prediction using environmental features had an accuracy of more than 90%. Elkhoukhi et al. [94] deployed an LSTM algorithm to predict short-term occupancy levels using historical CO2 concentration data. First, they evaluated the correlation between CO2 concentration and the occupant number. In the next step, they trained an LSTM model to predict future CO2 concentrations. Based on the steady-state equation proposed in [100], they estimated the number of occupants as a function of CO2 concentration in a research laboratory. The results showed that this LSTM algorithm provided accuracy and RMSE of 70% and 2.93, respectively. Although these research studies addressed privacy issues with a high level of accuracy for occupancy prediction, they did not evaluate the impact of using these alternatives on the model performance. There is a need for comparing these proposed models with common approaches to demonstrate whether using environmental features can have any impact on the model prediction performance.

There have been a variety of occupancy models developed by researchers, which highlights the necessity of evaluating and comparing the models using the same dataset to reveal their strengths and limitations. For this purpose, Kleiminger et al. [86] compared the performance of PreHeat, a hybrid model, and a proportional model which were respectively proposed in [13,45,87]. They applied these models to a dataset collected using GPS, Wi-Fi connections, and cellular networks through smartphones, described in [101]. The results demonstrated that the proportional method outperformed other algorithms with a median accuracy of 85%. Using the upper bounds of human mobility prediction performance defined in [102], they also estimated an upper bound accuracy of almost 90% for using more complex methods. Huchuk et al. [95] compared the performance of logistic regression, kNN, LSTM, random forest, Markov models, and HMM in terms of computational time and accuracy. Time of day, weekends/weekdays, and previous occupancy states were utilized as features to train the models. The results showed that the random forest algorithm provided the highest accuracy for occupancy prediction among the selected machine learning techniques, while logistic regression provided the best computational cost. Caleb et al. [82] applied SVM, random forest, decision trees, and k-NN, to an occupancy dataset with a resolution of 10 minutes to develop occupancy models. They included seasons, holidays, time of day, weekends, and day of week features in developing the models. The results showed that for schedules with a high occupancy rate (i.e., the frequency of being occupied), SVM outperformed other algorithms, while for lower occupancy rates, all models provided a similar performance. Sangogboye et al. [84] assumed that no single occupancy model can perform the best for all cases. Hence, they established the PROMPT algorithm that consisted of four occupancy algorithms, namely PreHeat, SVM, kNN-DTW, and random forest, developed based on previous work [45,103]. F-score performance of each model was compared in every prediction case, and the best algorithm was selected for the prediction task. It was shown that PROMPT increased F-score by up to 2.3% when compared with each individual model. In these studies, the impact of the data collection period, dataset resolution, the window size, and types of features used to develop the models were neglected. As the performance of models could depend on these factors, analyzing their sensitivity to these parameters can be important while comparing the models.

The impact of training set duration and dataset resolution on the performance of occupancy prediction models were evaluated by Salimi and Hammad [104] through a sensitivity analysis. They employed an inhomogeneous Markov chain model which was proposed in their previous work [91] and selected the coefficient of determination and normalized root mean square error (NRMSE) as the performance evaluation metrics. They showed that a sudden improvement in the accuracy was achieved by changing the period of the training dataset from one to two months, while for longer periods, the impact on accuracy was relatively smaller. In addition, the best time resolutions for the two occupancy zones investigated in this study were 5- and 10-minute intervals. However, the dataset used was limited to monitoring occupancy levels in a single office building. More research is needed using wider datasets to achieve more reliable results.

2.6.2. Prediction of individuals' occupancy patterns

Some researchers developed models to predict individuals' occupancy patterns rather than predicting occupancy levels in an entire zone. These predicted patterns can be, then, aggregated to construct occupancy level profiles. Occupancy models developed based on individuals' behavior might result in more accurate predictions, because they employ occupant identities as an extra feature [35]. However, using identities can cause privacy issues for occupants.

Salimi and Hammad [91] aimed to develop an adaptive occupancy prediction model in office buildings, utilizing identities, occupancy duration, and activity types (e.g. dining or working states) to train a Markov model. Occupancy information was gathered using Bluetooth tags, associated with each occupant. In this study, different prediction horizons were utilized based on two applications of occupancy prediction: HVAC and lighting control. In terms of HVAC control, due to the lag time associated with preheating, a 30-min prediction horizon was considered, while for lighting, a short-term prediction, i.e. 5-min, was utilized. The results showed that prediction accuracy significantly decreased from 86% to 68% when the prediction horizon increased from 5 to 30 minutes. A limitation of this study is linked with the occupancy detection system; the system did not collect any occupancy information regarding visitors to the office. Therefore, if one person occupied the office temporarily, the system failed to detect the occupancy presence and could not meet thermal comfort conditions. Additionally, they limited the prediction horizon to only one lag time, i.e. 30 minutes. As discussed in Section 2.6.2.1, the lag time varied from 30 minutes to 3 hours, and considering a wider range of preconditioning horizons would be more valuable when evaluating an occupancy model.

Shao et al. [85] developed an episode-generating HMM for occupancy prediction based on temporal relationships in occupancy patterns. An episode was defined as an ordered sequence of different zones visited by an occupant, such as kitchens, bedrooms, and corridors. The episode prediction algorithm was similar to life pattern mining proposed in [105]. They predicted future events according to the most frequent episodes, and the time of start and end of the last event. The results showed that the proposed model improved the F-score performance of occupancy prediction, compared with that of kNN and SVM models. However, they limited the proposed model to sequential visited spaces inside the building, such as different rooms. In other words, they neglected the relationship between these indoor spaces and outdoor locations visited by each occupant. Hence, there is a need for evaluating the possible improvements that can be achieved by defining an episode as an ordered sequence of visited indoor zones and outdoor locations in occupancy models.

Instead of predicting occupancy patterns, Das and Kjærgaard [93] focused on predicting the next specific locations of each occupant using a gated recurrent unit (GRU) algorithm. The locations were determined using video cameras by recording occupants' coordinates. To make the prediction more accurate, they added *sitting* and *moving* labels to each record in the dataset. MSE, MAE, and RMSE were selected as the performance evaluation criteria. They demonstrated that the GRU model predicted future locations with an RMSE of 4.79 centimeters. Although predicting the next locations of each individual can be helpful for predicting future occupancy states, the performance of this method for future pattern prediction was not evaluated.

Yu [89] proposed a novel model using a genetic programming algorithm for predicting occupancy patterns of individuals in single-person offices. The model was trained based on the following attributes: day of week (except for weekends), time of day, duration of occupancy in the last state, and the period from the last arrival time to the office. According to the results, the algorithm provided up to 83% accuracy for predicting occupancy states. However, no comparison between the proposed method and commonly used machine learning and statistical models in the literature were made. As the prediction performance significantly depends on occupancy profiles in buildings, without providing a direct comparison between occupancy models applied to the same dataset, the effectiveness of the proposed model cannot be well evaluated.

In contrast to the previously reviewed studies that used a few features, Gjoresk et al. [83] developed an occupancy prediction algorithm based on a relatively wide range of features including day of week, month, difference between total and expected working hours per month, historical arrival time, historical departure time, morning temperature, wind speed, cloudiness, daily precipitation quantity, and snowfall. A database collected from historical occupancy patterns of seven employees and weather data in 2 years were utilized to establish prediction algorithms. Naïve approach as a baseline model as well as kNN, SVM, Linear regression, M5-Rules, REPTree, Gaussian processes, and bagging algorithms were implemented for this regression problem. The proposed algorithm decreased MAE value by up to 50% and 32% for arrival and departure time predictions, respectively, compared with the baseline. However, although a relatively large number of features were utilized in the development of this algorithm, the importance of these features in predicting individual patterns was not clarified. To be more specific, a feature selection method was lacking in this study to determine the most effective features in developing occupancy models.

2.6.2.1. Rule-based control

In rule-based (RB) control, a pre-defined set of rules are applied to predicted future occupancy to provide energy saving and thermal comfort. In the literature, these rules were mainly defined as a function of preconditioning time, conditioning rate, or occupancy probability. Preconditioning time, also called the lag time of HVAC systems or pre-heating/cooling time, is defined as the time it takes to precondition a building to reach the setpoint temperature from a setback. Conditioning rate is defined as the rate of temperature change per hour during preconditioning periods. The previous research items are categorized based on the types of rules and are discussed in the following subsections.

2.6.2.2. RB control using preconditioning time and conservative setback

The majority of earlier work defined the rules in RB control as a function of future occupancy states and estimated preconditioning time. In these strategies, first, an approximate preconditioning time is estimated based on previous operations of HVAC systems. Then, it is used as the prediction horizon of the occupancy model. If the building is predicted to become occupied during this period, the controller turns the HVAC system on to bring back the setpoint before occupants' arrival.

Researchers considered various ranges for setback during unoccupied hours. As the depth of setback can considerably impact energy saving and occupants' comfort, the reviewed articles on RB control are classified into conservative and deep setback categories. The conservative setback is defined as a temperature deviation from the setpoint by 3 °C. Larger deviations are classified as a deep setback.

Lee et al. [63] deployed an RB control strategy using a hybrid occupancy prediction model based on occupants' locations received from cellular towers and Wi-Fi connections. This hybrid model used two different methods depending on transition time (i.e. the time it took for occupants to reach the target place from their current location). In cases when the transition time was larger than preconditioning time, a route classification algorithm was deployed. For short transitions, a time-aided order-k Markov predictor was applied to historical occupancy data to predict future destinations and associated arrival time. Using historical data, preconditioning time was estimated as 21.82 minutes with 1.98 minutes error for an average temperature difference of 2.5 °C. In order to evaluate the performance, they measured energy consumption and air temperature in target zones. The results showed that the proposed strategy decreased energy consumption by 26%, compared with always-on thermostats and was able to predict 70% of transitions with an error of less than 10 minutes. However, the method failed to learn unregular occupancy patterns with an overall accuracy of less than 60%.

Dong and Andrews [65] applied RB HVAC control to a conference hall. First, they developed an occupancy pattern recognition method based on gathered data from 6 types of sensors: acoustics, lighting, motion detectors, CO2 concentration, temperature, and relative humidity. The most frequent patterns and their subsets were selected as candidates to determine patterns with the strongest correlation with occupant behavior. In all control strategies, night and day setback temperatures were defined at 30 °C and 27 °C, respectively with a setpoint of 24 °C. The results showed that RB control resulted in up to 30% energy saving. As well as setpoint temperature control used in this study, OBSERVE control algorithm proposed by Erickson et al. [66] took both ventilation and temperature control into account using an RB control. The number of occupants from 8 different zones including offices, laboratories, and conference rooms was gathered using a network of 7 cameras installed in each corridor for a period of five days. In this study, a Markov chain model was used to predict occupancy. To evaluate the occupancy model performance, duration of occupancy states and the rate of people entering and exiting different zones were compared with the ground truth data. Jensen Shannon divergence (JSD) indicator was also used to compare the distributions of predicted results and ground truth during the testing period. The ventilation rates were adjusted based on the occupancy levels in each zone using the relationships suggested in ASHRAE Standard 62.1 [106]. The allowed temperature was in the range of 21.1-23.9 °C and 25.5-27.8 °C for heating and cooling seasons, respectively. Based on three virtual test environments simulated in EnergyPlus, it was demonstrated that this control system can save 42% annual energy, compared with scheduled control.

However, most of the mentioned works limited their study to evaluating energy saving performance and provided no analysis regarding thermal comfort of occupants. As the main aim of predictive control is to ensure a thermally comfortable indoor environment for occupants, it is essential to consider this criterion when evaluating predictive control systems. Some earlier work considered both energy saving and thermal comfort when evaluating the performance of control systems. Beltran et al. [64] proposed a control strategy for ventilation and temperature regulation, named ThermoSense. ThermoSense employed a network of thermal and PIR sensors and implemented a blended Markov chain model [66], to predict future occupancy states with a prediction horizon of one hour. The results demonstrated that ThermoSense decreased the annual energy consumption by 24.8%. However, it was shown that implementing a reactive control instead of ThermoSense provided higher energy saving of 29.6%. The amount of energy consumed for preconditioning the building before arrival time was the main reason for the lower energy performance of ThermoSence. However, due to this preconditioning period, ThermoSence provided lower temperature deviation from the setpoint with a maximum RMSE of 0.415 °C, while reactive thermostat resulted in a maximum RMSE of 1.23 °C. A similar algorithm, named POEM, was proposed by Erickson et al. [67]. The main distinction of POEM algorithm with ThermoSense was using a network of cameras to provide more accurate occupancy detection. The camera network provided an estimation of occupancy levels to regulate temperature schedules and minimize ventilation rates while meeting the ASHRAE standards [106]. The results showed that POEM reduced energy consumption by 26%. Moreover, it was demonstrated that there was no significant difference between the performance of POEM and reactive control algorithms in terms of energy saving; however, the reactive controller caused thermal discomfort upon occupants' arrival.

2.6.2.3. RB control using preconditioning time and deep setback

A limitation of the research items reviewed in the previous section is linked with the conservative temperature range considered in the control systems. This conservative range of setback temperature, although is essential in reactive control to minimize the discomfort upon arrival, can be widened in predictive control to save more energy in long unoccupied periods . It is because of the ability of the system to predict future occupancy and precondition the zone before the arrival time. Therefore, some researchers have studied a wider range of temperature in the control strategies.

Koehler et al. [44] developed an RB control algorithm, called TherML. This algorithm employed a hybrid occupancy prediction model based on contextual information from individual occupants. More specifically, the occupancy prediction was dependent on whether the occupant was driving or not. If the monitored occupant was driving, a methodology, similar to that implemented in [107,108], was used, in which a sequence of previously visited locations by occupants as well as time of day were implemented to predict future destinations. Otherwise, the system made predictions using a proportional model by calculating the frequency of historical occupancy states in 5-min time intervals. The average preconditioning time to change temperature from 15.5 °C to personal desired temperature (an average of 20.6 °C over all participants) was estimated as 59 minutes based on a data collection over 5 weeks. They compared the performance of TherML with that of GPS-thermostat [62], PreHeat algorithm [45], and a manual remote controller described in Section 2.4. It was demonstrated that TherML outperformed other methods in terms of accuracy and energy saving, providing an average accuracy of 92.1%, which was 1.5% higher than that of PreHeat algorithm. The average MissTime and WasteTime for TherML were respectively estimated as 0.95 and 0.94 hours, which were slightly lower than those related to PreHeat algorithm. A sensitivity analysis was also performed to show the impact of traveling distance on the accuracy of TherML. For long distances, 14,000 meters, TherML could provide up to 7% higher accuracy than PreHeat, while for short distances such as 200 meters, only 1% improvement was achieved using TherML. Implementing the simple proportional occupancy model can be mentioned as one of the limitations of this hybrid model. More advanced techniques such as machine learning models could be implemented to find more hidden patterns in occupancy profiles to boost the accuracy of occupancy prediction.

Gluck et al. [17] evaluated the impact of occupancy prediction errors and depth of setback temperature on the performance of RB control. To this end, they proposed a hypothetical occupancy model with 25%, 15%, and 5% errors in accuracy. Their hypothetical model was based on real data collected from 235 rooms in an office building with PIR and ultrasonic sensors. Three different ranges for setback temperature were considered in this study; 1) low-bound setback with a maximum 2 °C deviation from setpoint, 2) medium-bound with a maximum 6 °C deviation, and

3) large-bound with a maximum 10 °C deviation as the deepest bound. A reactive thermostat based on the low-bound setback was utilized as the baseline. According to the results, using a medium-bound and large-bound setback saved up to 26% and 40% energy, respectively compared to reactive thermostats. It was also reported that a 10% reduction in prediction errors resulted in up to almost 9% and 16% improvement in energy saving and MissTime, respectively. Although they considered a deep setback in the high-bound and medium-bound temperature ranges, only a fixed approximate preconditioning time of one hour was considered in this study.

Nägele et al. [68] compared the performance of manual, programmable, reactive, and RB thermostats in terms of energy saving and thermal comfort. They also assessed the impact of including weather prediction, obtained from meteorological weather stations, on the performance of reactive and predictive control. They collected occupancy data through smartphone applications over a 14-month period. The preconditioning time was estimated as one hour. They estimated energy consumption of the building through a Resistance-Capacitance (RC) model and demonstrated that the highest energy saving was achieved using reactive control at an average of 26.2%, which was followed by RB control at 23.7%. On the other hand, predictive controllers provided the lowest MissTime of 28.2 hours, which was 103.9 hours lower than that of reactive control. A limitation of this study is linked with the utilized occupancy prediction model. More specifically, they implemented perfect occupancy prediction and neglected the impact of errors on the control system performance. Therefore, the results for the RB controller is overestimated, providing an ideal upper bound for the RB control. As well as utilizing perfect prediction, occupancy models can be also implemented to give an insight into predictive control performance in real-world applications.

Iyengar et al. [46] focused on improving currently installed programmable thermostats without a need of additional infrastructure. They called these revised thermostats iProgram. iProgram predicted future occupancy based on energy consumption patterns in residential buildings without the need of a training dataset, as it is not available in many homes with programmable thermostats. They evaluated the performance of the proposed occupancy detection algorithm using accuracy and Matthews correlation coefficient. As for controller performance, MissTime, WasteTime, mismatch time (sum of WasteTime and MissTime) were assessed by simulating 100 homes as virtual test cases. It was demonstrated that iProgram resulted in 0.42 kWh average energy saving per day. Furthermore, the mismatch time decreased by an average of 42 minutes having a median deviation of almost 30 minutes. However, the proposed occupancy detection method using energy consumption patterns was limited to predicting occupancy during the day and failed to predict occupancy during the nighttime when people are sleeping.

2.6.2.4. RB control using conditioning rate and occupancy probability

In the earlier work concerning RB control, researchers considered an average preconditioning time as the prediction horizon to make control decisions for initiating the preconditioning process. However, this static time cannot consider the actual difference between dynamic room temperature and the desired setpoint. As a consequence, when room temperature reaches a setback after a long vacant period, it clearly takes more time than the average to precondition the building before occupants' arrival, which increases the MissTime. Similarly, after a short vacancy, it can cause energy waste due to over preconditioning the zone. Hence, some researchers have defined the preconditioning time by multiplying the conditioning rate and the difference between actual room temperature and setpoint. This definition is more helpful when a wider range for temperature is allowed in control algorithms. Besides, some researchers have employed occupancy probability in control systems rather than using deterministic occupancy models. In these cases, RB control systems determine the depth of setback as a function of occupancy prediction confidence. For example, when the probability of occupancy remains negligible in the following hours, a deeper setback is defined to save more energy. In contrast, when the probability of presence is relatively high, a conservative setback is set to ensure occupants' comfort.

Nikdel et al. [60] studied the benefits of using RB control in office buildings from a national point of view. They evaluated the influence of such systems on the amount of fossil fuel consumption, greenhouse gas emission, and energy cost, compared with always-on and programmable thermostats. They defined setpoint temperature as a linear function of the presence probability. Based on this relationship, the temperature was allowed to change between 15.6 and 21.1 °C in the winter and 23.9 and 29.4 °C in the summer. The results showed that up to 50% and 87% reduction in electricity and natural gas consumption was respectively obtained using this occupancy-based HVAC operation, compared with always-on control. With all small offices in the US having this type of control, the saving of fossil fuel, CO2 emission, and cost at the national level were estimated as 15-66 million GJ, up to 3.7 million metric tons, and 658 million dollars, respectively. However, the impact of RB control on the thermal comfort conditions was neglected in this study. As improving cost or energy saving are often obtained at the expense of occupants' thermal comfort, the control system should be also evaluated in term of providing thermal comfort for occupants.

Using a two-level probabilistic occupancy model, Peng et al. [61] defined an RB HVAC control strategy for use in office buildings. At the first level, a k-means clustering algorithm was applied to a real occupancy dataset, collected using motion sensors, to cluster similar data records. In the second level, a k-NN algorithm was utilized to predict occupants' first arrival time and the duration of occupied hours in office buildings. Arrival time, last departure time, and maximum unoccupied duration were utilized as features for developing occupancy models. They defined the setback temperature as a function of the probability of presence obtained from the occupancy model. The setback temperature could increase from the comfort temperature to 35 °C in cooling seasons depending on the occupancy probability. In this study, a field evaluation was performed in an office building with a meeting room, and single- and multi-person offices. Energy meters were utilized to evaluate the amount of energy consumed by chillers. The results achieved from this experiment demonstrated 7%-52% energy saving in comparison with scheduled cooling systems. However, in this study, although a novel two-level occupancy prediction algorithm was proposed, no comparison between this algorithm and conventional one-level predictions was provided to show the strengths of the proposed model.

Lu et al. [13] proposed a smart thermostat to automatically adjust zone temperature to occupancy states in buildings. The proposed smart thermostat employed three strategies to save energy while keeping an acceptable thermal comfort level. They included turning off the HVAC system after residents left home or slept, preheating the home before their arrival, and employing a deep setback temperature during confidently vacant hours. They used a hidden Markov model (HMM) to infer three different possible occupancy modes at home; 1) active, 2) away, and 3) sleep modes. Future arrival time was estimated using the historical arrival times recorded by the occupancy detection system. Three features were used in the occupancy model; 1) time of day, 2) total number of true signals received from occupancy sensors, and 3) signals received from door sensors. The results showed that this approach led to almost 28% energy saving, compared to conventional thermostats.

Scott et al. [45] developed an RB control, called PreHeat, for heating five residential homes in the UK and US. They collected occupancy information using radio-frequency identification (RFID) tags associated with residences' keys and motion detectors installed in homes. Based on this data, they created a partial occupancy vector that consisted of occupancy information from midnight until the current time step. They used this vector as a primary attribute for developing occupancy prediction in 15-min intervals. They reported that occupant behavior in the considered homes was highly dependent on weekends or weekdays, and as a result, they used this parameter as the second feature. They used a kNN algorithm to calculate the probability of presence, based on the mean of 5 nearest historical occupancy states. PreHeat employed occupancy models and an empirical preconditioning rate based on the historical data. They utilized a room-based control for 2 homes in the UK and whole-home control for 3 homes in the US. They measured gas consumption for each residential unit to evaluate the energy performance of controllers. The results demonstrated that PreHeat decreased MissTime by up to 92% without the need for human interventions. However, PreHeat did not provide improvements in energy saving compared with programmable thermostats.

As well as using historical occupancy patterns to predict future occupancy in a target zone, some researchers have employed information about occupants' routines even when they are not in the monitored building. Gupta et al. [62] proposed a thermostat, called GPS-Therm based on real traveling data collected from four households in the US. Occupants' locations were utilized to estimate the time it took for the nearest occupant to reach home. The estimated traveling time was then employed to schedule a setback profile to save energy. The setback was defined as a function of traveling time so that the HVAC system has enough time to preheat the building before the arrival. Implementing this heating strategy resulted in 7% energy saving in the considered households. However, as the setback temperature was defined as a function of traveling time, in cases with short traveling time, a too shallow setback is defined using this method. Therefore, in these cases, the GPS-Therm leads to no substantial energy saving compared with always-on thermostats. Krumm and Brush [87] developed a proportional occupancy model and compared its performance with that of GPS-Therm in terms of occupancy prediction accuracy. They developed their model based on real data collected from 11 households using time of day and day of week as input features. It was demonstrated that the proportional model performed much better than the drive-time algorithm. The reason was that, occupants spent most of their time near their homes, and as a result, the traveling time was often short. They reported a slight improvement in the performance of occupancy prediction when they combined both methods.

2.6.3. Optimal control

In contrast to rule-based control, in which a set of rules are implemented, in optimal control, optimization algorithms are employed to make optimal sequences of decisions. In most studies conducted in this field, optimal controllers were employed to make an optimal trade-off between minimizing energy consumption and maximizing thermal comfort. Model predictive control (MPC) has been the most utilized occupancy-based optimal control strategy in the literature. Interested readers can find more information about MPC operation in HVAC control applications in [33].

Despite many papers published on MPC-related topics, there are a limited number of papers that implemented occupancy prediction in MPC. Oldewurtel et al. [70] evaluated the impact of using occupancy prediction in MPC in terms of energy saving. To this end, they compared the performance of an MPC using instant information from occupancy sensors, as the baseline, with that of an MPC that took advantage of perfect occupancy prediction for HVAC and lighting control. In this study, the temperature setpoint was constrained in the range of 5 to 40 °C in the optimization problem during vacant hours. This wide temperature range was assumed to give an upper bound of energy saving that can be achieved from this control system. Based on the assumptions, they concluded that the amount of energy saving achieved by using complex occupancy predictions led to a relatively low improvement in the system performance and might not compensate for its complexity. In a similar work by Goyal et al. [71], the performance of a reactive control and that of an MPC using 24-hr perfect occupancy prediction were compared in terms of energy saving. In this study, the range of setback temperature in all cases during unoccupied periods was considered from 21.1 °C to 24.4 °C, which was close to the range of 21.9 °C to 23.6 °C, considered during occupied hours. They showed that all three proposed methods resulted in a considerable amount of energy saving (almost 50%). However, despite the high complexity of MPC-based algorithms, these methods did not lead to significantly higher performance. Goyal et al. [109] also verified the conclusions made in [71] by conducting an experimental study on a single zone office as a testbed. However, it should be noted that in none of these studies the impact of MPC on occupants' comfort was evaluated. As one of the key purposes of using predictive control is to improve thermal comfort, considering energy-efficiency as the sole indicator can lead to underestimation of the system performance.

As well as minimizing energy consumption of the system, Salimi and Hammad [69] utilized a multi-objective genetic algorithm to minimize MissTime. In their study, occupancy profiles were predicted by applying a Markov model to historical occupancy data gathered from an individual office. They utilized EnergyPlus to estimate the performance of the proposed algorithm and demonstrated that this optimal approach enhanced thermal comfort by 50% and decreased the energy consumption by 2%, compared to a scheduled control. However, MissTime fails to consider the difference between the actual and desired room temperature in the optimization problem. To deal with this limitation, Shi et al. [74] proposed a cost function, similar to that implemented by Killian and Kozek [72], to be minimized in the optimization algorithm of MPC. The cost function consisted of thermal discomfort and energy consumption terms. The former term was defined by multiplying the presence probability, achieved from occupancy models, and a thermal discomfort factor, defined as the difference between the actual room and the desired room temperature. In this optimization problem, setpoint and setback temperatures were respectively constrained in the ranges of 18-28 °C and 20-24 °C. The results indicated that using occupancy prediction in MPC improved energy saving by up to 8%, compared with the traditional MPC, and maintained an acceptable level of thermal comfort. Turley et al. [75] evaluated the performance of occupancy-based MPC and compared it with that of reactive control. The ground truth occupancy data was gathered using applications installed on the users' phones and using regular surveys filled by occupants. Prediction and control horizons for MPC were selected as 24 hr and 1 hr, respectively. They employed a particle swarm optimization algorithm to find the temperature that optimized predicted mean vote (PMV) and energy use in a weighted optimization algorithm. The results showed that MPC reduced energy consumption by up to 13.3%. They also reported that MPC with perfect occupancy prediction saved almost 3% more energy, compared to MPC with occupancy prediction models.

Killian et al. [72] compared the performance of MPC with and without integration to occupancy prediction models in terms of temperature violation from setpoint in residential buildings. They applied a k-means clustering algorithm to an occupancy dataset, collected using lighting sensors, to develop an occupancy prediction model. They showed that an MPC with occupancy prediction reduced the temperature violation by up to 32.80%. Gao and Whitehouse [76] developed an optimal control based on modifying programmable thermostats. They called the system "self-programmed thermostats", that was able to construct a temperature schedule based on historical occupancy data. Future occupancy patterns were constructed based on maximum departure time and minimum arrival time in the last one-month occupancy patterns. They constructed an optimization algorithm to minimize HVAC run time as an indicator of energy saving. Their preliminary results showed that using self-programmable thermostats rather than standard programmable ones decreased HVAC run time by 15%. It also decreased MissTime by 12%-40% depending on occupancy patterns.

2.7. Discussion of previous research limitations

Based on this comprehensive literature review, the research gaps in the field of occupancy-based HVAC control are discussed in this section. To this aim, the reviewed papers are summarized and explored from the following different viewpoints: performance assessment methodologies for occupancy models and control strategies, the features used in developing occupancy models, types of occupancy models and control systems, and testbeds used for model evaluation.

2.7.1. Performance indicators for evaluating control systems

Fig. 2.4 demonstrates the frequency of different performance criteria used to assess occupancy-based control systems. It can be seen that energy saving was the most popular performance indicator in previous studies. However, as discussed in Section 2.6.3, using energy saving as the only indicator can result in undervaluation of predictive control performance, especially when compared with the performance of reactive control. This is because reactive control often decreases energy consumption at the expense of thermal comfort, while predictive control improves comfort by preconditioning the building. This preconditioning can cause more energy consumption, and if only energy saving is concerned, reactive control can outperform predictive control by neglecting their benefits in providing thermal comfort. Therefore, it is strongly recommended not to ignore thermal comfort conditions while evaluating the performance of predictive control strategies.

It can be also observed that HVAC runtime has been rarely utilized by researchers to assess the merits of the control strategies. This is consistent with the results achieved by Pritoni et al. [54], reviewed in Section 2.5. Briefly, they demonstrated that there is not always a relationship between HVAC run time and energy saving, and employing HVAC run time as performance merit can lead to overestimation of system performance. Therefore, using energy saving rather than HVAC run time is more recommended to provide more reliable results.

As demonstrated in Fig. 2.4, the financial merit of control systems has been one of the least-used criteria in evaluating control systems in the earlier studies. However, this is an essential factor for encouraging building owners to replace the old control systems in buildings with more intelligent ones. To be more specific, the main barrier for many owners in retrofitting the buildings with intelligent control is the initial costs; therefore, evaluating the economic benefits, estimated through financial analysis, can be a great incentive for them. Similarly, the majority of earlier work have neglected peak-shifting performance, despite its importance in demand response management. In some areas with cold climates, utility companies encounter serious challenges to provide electricity for customers during on-peak periods in winter. The focus in such areas is to find a solution to shift on-peak demand to the off-peak periods via encouraging costumers to manage peak demands or using energy storage [110]. However, the occupancy-based control systems proposed in the literature aimed to keep the temperature at setback during vacant hours and return the thermal condition as soon as occupancy states were changed. It naturally can bring about a sudden change in energy demand profiles and, consequently, are prone to cause peak energy use. However, as researchers have rarely investigated the control systems in terms of peak demand, these aspects have not been clarified yet. In addition to these factors, the environmental impacts of occupancy-based HVAC control have been mostly neglected by researchers and there is not enough information to show their performance from environmental dimensions.

Overall, the performance of occupancy-based HVAC control systems from economic, energy, and environmental points of view have not been well investigated in previous studies. Because of the importance of these aspects, it is recommended to take these factors into account when proposing and evaluating control algorithms in this field.

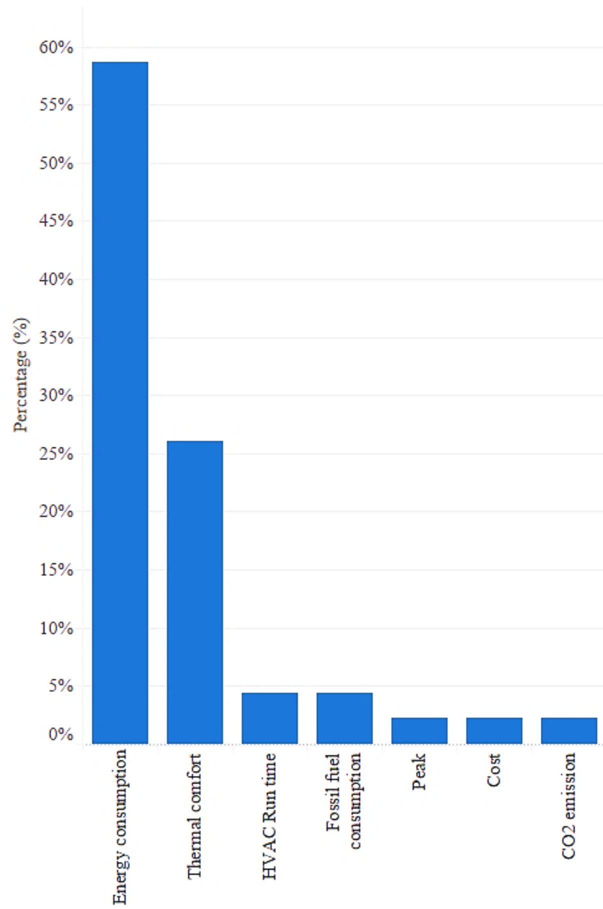


Fig. 2.4. The frequency of different performance criteria utilized to assess the merits of occupancy-based control strategies in the reviewed studies.

2.7.2. Features used to develop occupancy models

The frequency of features implemented to develop occupancy models is represented in Fig. 2.5. Time of day was the most frequent feature at 25%, that is followed by day of week, that accounts for almost 12% of all the utilized features. It is clear that features such as time of day, day of week, weekends and seasons are most likely to affect occupancy prediction performance, as we can find obvious correlations between them and occupancy patterns. For example, people mostly work during the day and sleep at night or schedule their routines based on day of week. However, it is still a question that which features are the most effective ones and how many features will provide the best results. It is shown that almost 29 different features have been employed in earlier works to develop occupancy models. Considering this variety of features, a wise selection of attributes can be of great importance as it can improve the accuracy of the prediction, provide a better understanding of the underlying process and enhance the computational speed [15] [111]. Too many or too few features can, respectively, result in overfitting or underfitting problems [14]. Additionally, as mentioned in [16], finding the most relevant features in developing an optimal model can minimize the cost of purchasing too many sensors. However, despite the importance of feature engineering in developing data-driven models, there is still a question that what number and what types of attributes should be selected for future occupancy prediction. In other words, feature engineering studies in the field of occupancy prediction lacks in the literature. Hence, investigating the optimal features is recommended as future work to provide a guideline for researchers to select the most appropriate features.

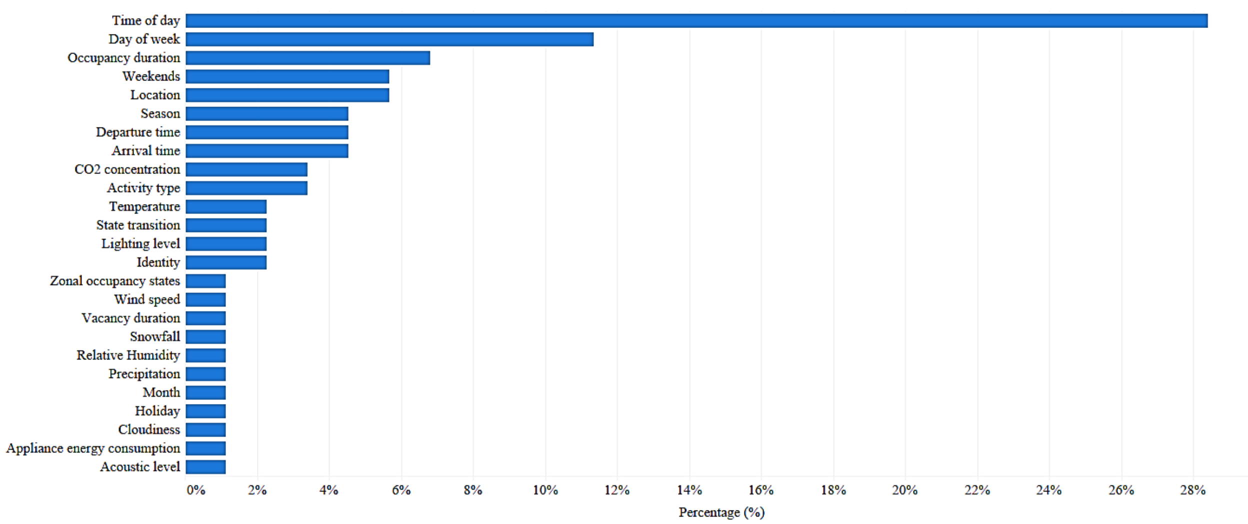


Fig. 2.5. Frequency of different features used for developing occupancy models in the reviewed papers.

2.7.3. Types of control strategies and evaluation methods

Fig. 2.6 demonstrates the distribution of earlier works based on types of control strategies and evaluation methods (i.e. simulation or field evaluation). As can be seen, rule-based control was the most used control strategy in earlier research work, accounting for almost half of all the studies. It was followed by reactive and optimal control at 26%. It is observed that all the studies on optimal control were simulation-based studies, and no researchers have applied them to actual testbeds. In contrast, half of the studies on reactive control were evaluated using real case studies. One of the reasons can be linked with the fact that while optimal control strategies are still under development, reactive and rule-based controls are commercially available for use in buildings. In fact, many commercial buildings already utilize reactive strategies to control HVAC and lighting systems, which makes field evaluation more possible. In addition, optimal control often requires higher computational power than the currently available power in building management systems. These factors have made the experimental evaluation of optimal strategies more challenging. Hence, there is a need to deal with this challenging task and investigate such systems when applied to real-world applications.

As mentioned in Section 2.6.3, MPC was the most-utilized approach for occupancy-based optimal control in the literature. Despite its promising performance, there are some barriers preventing them from being widely utilized in the building sector, such as their poor potential of generalization for use in different buildings and the intensive expertise required to develop MPC models [19]. In the last decade, reinforcement learning (RL) has attracted a great deal of attention as a powerful alternative to MPC in HVAC control systems. RL algorithms are data-driven approaches that can be utilized in HVAC control without a need for the development and calibration of building models [21]. RL has been utilized in many different building-related applications such as the control of lighting systems [22], cogeneration systems [23], domestic water heaters [24], energy storage [25], and HVAC systems [26]. However, there is still a need for investigating the effectiveness of RL algorithms for occupancy-based HVAC control. It is recommended to evaluate and compare the performance of RL-based HVAC control with that of MPC-based algorithms to reveal the strengths and limitations of each approach in this field.

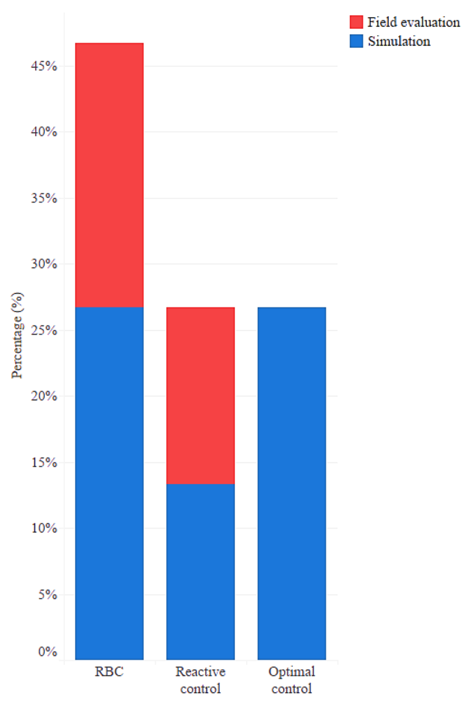


Fig. 2.6. Distribution of the papers based on control type and evaluation methods.

2.7.4. Building types

Fig. 2.7 depicts the proportion of different building types used as testbeds in the reviewed papers. It can be seen that office buildings accounted for more than half of the previous research studies. These offices were mostly occupied by researchers in educational institutions. The reason is that such buildings are more accessible for researchers to conduct experiments and collect the required data. In contrast, gathering data in other types of commercial buildings can cause more privacy issues and have been less studied by researchers. It can be observed that other commercial buildings such as exhibition halls, conference halls, and hotels account for 9% of the previous studies, and there is less information about the effectiveness of occupancy-centric HVAC control in these applications. Therefore, investigating different testbeds rather than research offices and residential buildings are recommended to provide an insight into using these control strategies in various case studies.

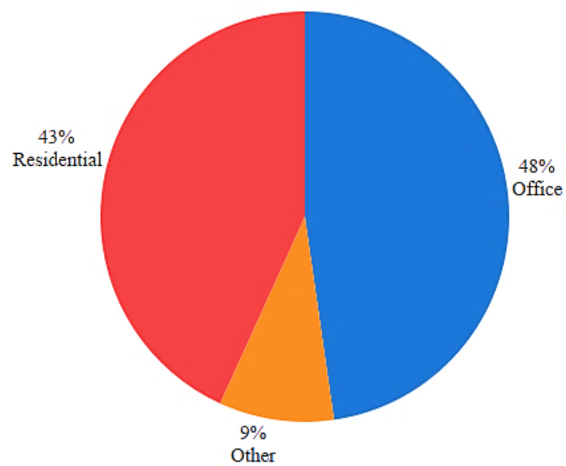


Fig. 2.7. The distribution of testbeds used to evaluate occupancy-based control systems in the reviewed papers.

2.7.5. Occupancy prediction models

The frequency of using different occupancy models is demonstrated in Fig. 2.8. The most used occupancy models were Markov and proportional models which accounted for 28% of all the occupancy models. These models are not complex to be implemented and provided an acceptable level of accuracy in most of the cases. These models are followed by kNN that accounted for 6% of the models. However, there has been relatively less effort put into developing more complex models such as deep learning algorithms. More specifically, deep learning algorithms, namely, LSTM, GRU, and RNN account for less than 8% of the developed models. Hence, it is recommended to make attempts into applying more complex models to occupancy prediction in order to capture further hidden patterns in occupancy datasets and improving the performance of the models.

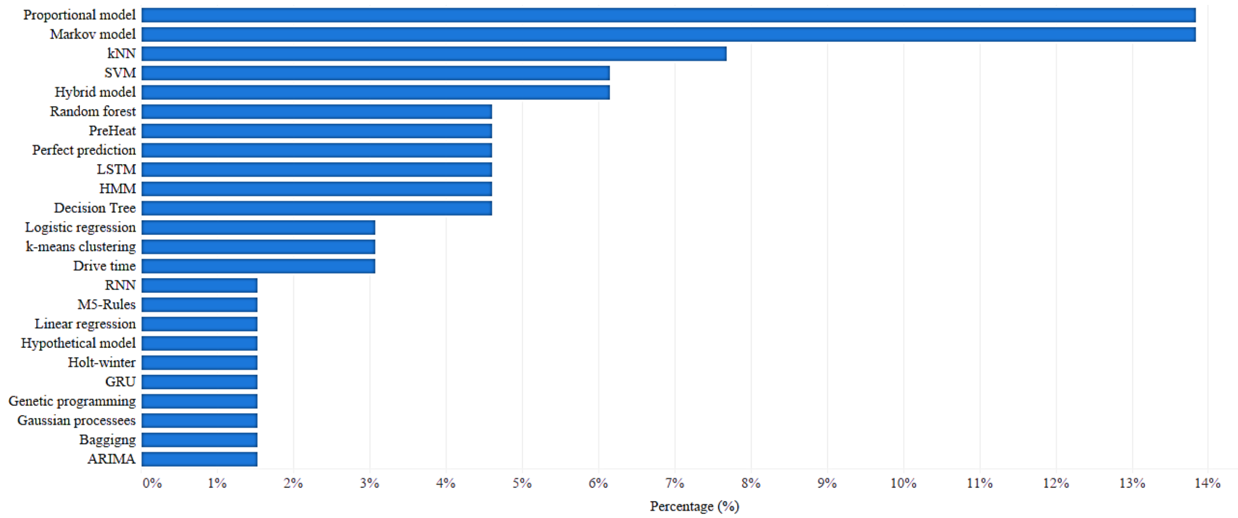


Fig. 2.8. Frequency of various occupancy models used in the literature.

2.7.6. Performance indicators for occupancy models

Different performance metrics utilized for evaluating occupancy models in the reviewed papers are shown in Fig. 2.9. Accuracy was by far the most utilized indicator, at around 45% of all metrics. However, accuracy might be misleading when used as the primary indicator to show the effectiveness of an occupancy model [46]. In some cases, when occupancy/vacancy rates are high (i.e. people stay home or are outdoors most of the time), occupancy models might select the most frequent state in the dataset as the future occupancy prediction results for all future time steps. In this way, the model can provide an acceptable level of accuracy, while the results would not be practically useful for HVAC control. Besides, as mentioned in [73], the impact of occupancy prediction errors on HVAC control depends on its nature and timing. However, the commonly used performance indicators address none of these factors. Hence, there is a need for research studies to compare and evaluate different performance indicators to show their limitations when applied to occupancy models in HVAC control.

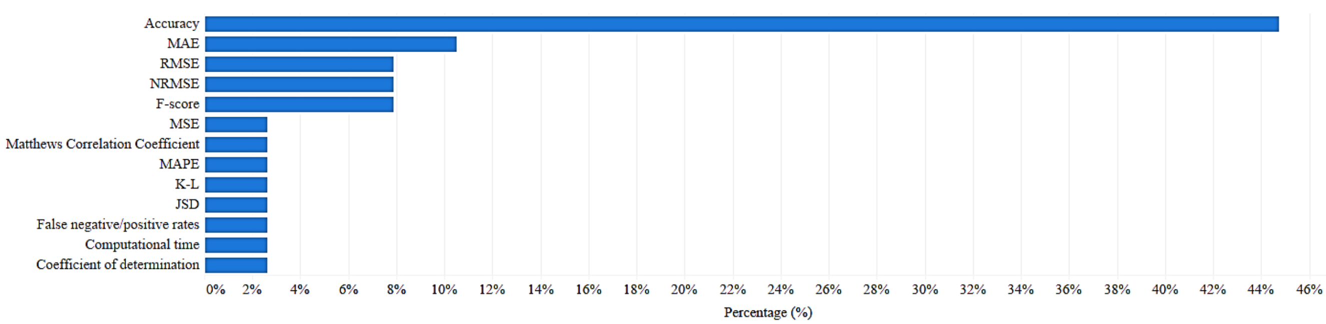


Fig. 2.9. Performance criteria used for evaluation of occupancy models.

2.8. Conclusion

This chapter reviews the state-of-the-art occupancy-based HVAC control systems proposed in the literature for residential and commercial buildings. To this aim, previously published papers are classified into two main categories according to the integration of occupancy information with control systems. Then, the papers are further classified based on the control strategies applied to occupancy information. The research items in each category are explored in detail, and the methodologies are investigated from different viewpoints. It is pointed out that the HVAC control systems based on user-defined occupancy schedules provide poor performance in terms of energy saving and thermal comfort in many cases. Therefore, there has been an increasing interest in using occupancy detection and monitoring systems to minimize occupants' interventions. Reactive control is considered as a successful approach to minimize the interactions; however, employing this type of control can cause thermal comfort issues for occupants upon arrival owing to the lag time of HVAC systems. Predictive control strategies were proposed in the literature as a solution to this issue by enhancing thermal comfort of occupants. However, it was reported that in many cases, using predictive control results in lower energy performance than reactive control due to preconditioning buildings before occupant arrival.

This thorough literature review shed light on the limitations of different methodologies. These limitations are summarized according to different dimensions, including feature utilization in developing occupancy models, types of occupancy models, types of buildings used as testbeds, performance metrics used to evaluate control systems and occupancy models, and types of control systems integrated with occupancy models. It is indicated that the majority of occupancy-based control systems were studied with respect to energy efficiency and thermal comfort, and consequently, their effectiveness from other viewpoints such as financial merits, demand-response management, and greenhouse gas emissions was not well evaluated in the literature. Additionally, despite many studies on developing occupancy models, a limited number of research items implemented more advanced algorithms, such as deep learning methods, to capture more hidden patterns in occupancy profiles. Furthermore, it is shown that 91% of the previous studies focused on occupancy-based control systems in office or residential buildings. Consequently, limited information regarding their performance in other types of buildings, such as conference halls, schools, and shopping centers, can be found in the literature.

Chapter 3: Occupancy-based HVAC control using deep learning algorithms for estimating online preconditioning time in residential buildings[2]

3.1. Overview

This chapter presents a rule-based (RB) heating, ventilation, and air-conditioning (HVAC) control system using a multi-layer perceptron network, as a deep learning algorithm, for estimating dynamic preconditioning time in residential buildings. The proposed system takes advantage of occupancy, indoor temperature, and weather data to make control decisions in buildings. The system performance is evaluated in terms of financial, demand-side management, energy-efficiency, and thermal comfort viewpoints. The proposed approach considers the perfect occupancy prediction assumption to remove the impact of the uncertainty associated with occupancy prediction and to estimate an upper bound limit for the system performance. The system performance is compared with that of conventional RB control approaches to show its effectiveness. To select the optimal control system, the TOPSIS method, as a multi-criteria decision-making approach, is employed. The sensitivity of the proposed system to the temperature setback is also assessed by considering conservative, medium, and deep setback bounds. It is demonstrated that the proposed system outperforms other alternatives when the deep and medium bounds are utilized. This study reveals two limitations of occupancy-based control systems by investigating their performance from financial and peak-demand points of view. First, these systems can cause peak-demand issues and increase the on-peak energy consumption by up to 10%. Secondly, employing a conservative setback can significantly decrease the financial merits of the system, leading to a discounted payback period of 10.87 years for implementing smart thermostats.

3.2. Introduction

Space heating and cooling demands account for more than half of the building energy consumption worldwide [112,113]. As discussed in Ref. [60], almost 39% of this amount could be wasted due to conditioning vacant zones, over-conditioning, air leakages, and the use of appliances with low efficiency. This huge amount of energy waste highlights the necessity of implementing more efficient control systems to optimize building energy consumption and cost. In 1906, the first types of programmable thermostats were produced as a solution to energy waste, caused by conditioning unoccupied spaces [11]. It was previously estimated that a considerable amount of energy saving could be achieved through properly programmed thermostats [58]. However, many studies reported that such thermostats practically suffer from being inappropriately programmed by users and do not lead to significant energy saving in real cases [40]. To deal with this issue, reactive control was proposed and investigated [50,52,54–59]. This control system automatically infers occupancy states using sensor networks and utilizes a setback temperature in vacant spaces, which

[2] This chapter is based on the following publication: M. Esrafilian-Najafabadi, F. Haghighat, Occupancy-based HVAC control using deep learning algorithms for estimating online preconditioning time in residential buildings, Energy and Building 252 (2021) 111377. https://doi.org/10.1016/j.enbuild.2021.111377.

can result in higher energy efficiency and cost reduction. However, it is essential to implement a conservative setback temperature to minimize the occupants' thermal discomfort upon arrival [17].

Earlier research studies investigated predictive control, as an alternative to reactive control, by integrating future occupancy predictions with control systems. Knowing future occupancy patterns in advance provides HVAC systems with enough time to precondition the building before occupants' arrival. Theoretically, using such proactive control systems allows the use of deeper setback to save more energy during unoccupied periods while maintaining thermal comfort when occupants are present.

Rule-based (RB) control has been one of the most utilized occupancy-based predictive control strategies in the literature [114]. In RB control, a set of rules is implemented to define setpoint/setback schedules as a function of future occupancy states, forecasted by occupancy models. Most earlier research studies developed RB control by approximating a static preconditioning time, also named preheating time or HVAC lag time, as the prediction horizon of occupancy models (i.e., how far the models predict occupancy profiles in advance). However, the actual preconditioning time often differs from this static average value, which can cause inaccuracies in the control system and result in energy waste or thermal discomfort. To be more specific, on the one hand, after a long unoccupied period when the temperature can fully drift to the setback, it naturally takes more time than the average value to precondition a building before occupants' arrival. On the other hand, after a short unoccupied period, using the average preconditioning time can cause a waste of energy owing to unnecessarily conditioning unoccupied spaces. Additionally, the preconditioning time can also depend on weather conditions that need to be considered for more accurate modeling. It is obvious that it takes much longer to preheat a building on a cold day than on a day with moderate temperature, and as a consequence, neglecting this dependency might also cause imprecisions in the control system. However, to the best of the authors' knowledge, no earlier studies investigated the integration of online preconditioning-time estimations with RB control systems.

Optimal control algorithms, such as model predictive control (MPC), have been proposed as alternatives to RB control. Such control systems take advantage of accurate building models, weather data, and robust online optimization algorithms, as well as occupancy models to make optimal sequences of control decisions to maximize energy saving and thermal comfort [115]. On account of using building models in optimal control, the optimization algorithm can dynamically estimate the preconditioning time during the decision-making process. However, there are some limitations on the use of such algorithms that prevented them from being widely applied in the building sector [19]. The limitations include its relatively low potential for being generalized in different cases, the need for high skills in the development stage, and the high computational power requirement. Additionally, as discussed in Ref. [109,116], the amount of energy saving achieved by using MPC might not compensate for the complications added to the control problem.

Another limitation of the earlier research works is that they mainly focused on energy efficiency and thermal comfort when evaluating the performance of RB control systems, neglecting other factors such as financial benefits and demand-response management. However, these criteria are of great importance when evaluating a control strategy in buildings. Financial indicators are the most effective incentives for many building owners and project managers to replace the old thermostats with new ones. Furthermore, in many regions, power plants are struggling with covering the energy demand of users during the peak periods [117,118]. In such cases, the timing of energy consumption can be as important as the amount of total consumption, and the main focus is given to demand-response management programs. Hence, there is a need for considering all these factors when studying occupancy-based control systems.

This chapter proposes an RB control system, aiming to address the limitations of previous studies by predicting online preconditioning time. The proposed control system is expected to outperform conventional RB control systems while avoiding the complexities of optimal control. To this end, a multi-layer perceptron (MLP) network, as a deep learning algorithm, is developed using the historical data for outdoor weather, indoor temperature, and the lag time of HVAC systems. The performance of the proposed control system is evaluated and compared to that of conventional RB systems with static preconditioning time. As well as energy-efficiency and thermal comfort merits, the performance is evaluated in terms of economic and demand-response management. A comprehensive financial analysis is performed to quantify the economic benefits that can be obtained by using occupancy-based HVAC control systems. Different financial indicators, such as the internal rate of return (IRR), discounted payback period (DPB), net present value (NPV), and annual total cost saving ratio (ATCSR) with the consideration of tax and inflation rates, are estimated in this analysis. The TOPSIS (Techniques for Order Preferences by Similarity to the Ideal Solution) method, as a multi-criteria decision-making approach, is employed to provide the trade-off between the conflicting criteria for selecting the best control method among the proposed and the conventional control systems. In addition, the sensitivity of the control systems to setback temperature is evaluated using conservative, medium, and deep setback bounds. The main limitation of earlier work and the contributions of this study can be summarized as follows:

- The dynamic nature of the preconditioning time was ignored in conventional occupancy-based RB control systems implemented in previous research works. This study proposes a novel RB control system using online estimation of the preconditioning time, as a function of indoor operative temperature and weather conditions, via deploying deep neural networks.
- Previous research works mainly focused on the energy and thermal comfort merits of occupancy-based HVAC control systems while neglecting the financial profitability and demand-response management potential of the systems. In this chapter, a comprehensive financial analysis is carried out to assess the economic gains achieved from replacing old thermostats with smart ones. Furthermore, the impact of using occupancy-based HVAC control systems on the peak-demand profiles of buildings is also investigated.

The rest of the chapter is structured as follows: Section 3.2.1 provides a literature review on the related research works and discusses the methodologies implemented for occupancy-based control. In Section 3.3, the proposed RB control system is described in detail. The building simulation tool used for evaluating the proposed algorithm is discussed in Section 3.4. Section 3.5 describes the performance evaluation criteria utilized in this study. The results are presented in Section 3.6, and finally, the conclusions and main findings are discussed in Section 0.

3.2.1. Background

Among different types of occupancy-based HVAC control methods, RB control has been the most utilized system in the literature [114]. RB control systems are cost-effective and require relatively low computational power. Lee et al. [63] conducted an experimental study to evaluate the performance of an RB control system in terms of energy saving using occupancy information collected from smartphones. They approximated a static preconditioning time of almost 22 minutes for increasing the temperature by an average of 2.5 °C in different case studies. According to the results, this system decreased the energy consumption by 26% in comparison with an always-on control system as the baseline. Erickson et al. [66] developed an RB control system, called OBSERVE, which used the information from a network of cameras to detect occupancy and accordingly regulated setback temperature and ventilation rates with a preconditioning time of almost one hour. Using a simulation approach, they demonstrated that this system was able to save 42% of energy annually. Dong and Andrews [65] also reported up to 30% energy saving when they applied occupancy-based RB control to a conference hall.

As well as energy-efficiency indicators, some earlier works studied the performance of RB control from a thermal comfort viewpoint. Beltran et al. [64] applied RB control to regulate temperature and ventilation rates in an office building, using a fixed preconditioning time of one hour. They reported that this system resulted in 24.8% annual energy saving; however, it caused temperature deviation from the setpoint during occupancy hours with a mean square error (MSE) of almost 0.4 °C. Koehler et al. [44] developed an RB control, called TherML, based on a preconditioning time of 59 minutes using 5-week historical data for the lag time of HVAC systems to increase the temperature from 15.5 to 20.6 °C. They demonstrated that TherML could lead to almost 22 hours with correct temperature setting (i.e. maintaining setpoint temperature during occupied periods and setback temperature during vacant periods). Gluck et al. [17] developed an RB control using a fixed one-hour prediction horizon and three different setback bounds with 10, 5, and 2 °C deviations allowed from the setpoint during the vacancy. They reported that using the same temperature bounds, reactive control always outperformed the RB controller in terms of energy saving. However, they assumed a more relaxed setback range for the predictive control and showed that the RB control provided almost 21% more energy saving than the reactive one. In addition, RB control demonstrated the ability to reduce MissTime (i.e. how long occupants encounter a space in which the actual temperature deviates from the desired setpoint temperature) of reactive control by up to 180 min per day. Nägele et al. [68] studied and compared the operation of reactive and RB control from energy-efficiency and thermal-comfort viewpoints using occupancy data from smartphones and building simulation. They reported that the reactive control led to the highest energy saving by almost 26%, whereas RB control accounted for the lowest MissTime of approximately 28 hours. Iyengar et al. [46] proposed iProgram as an opportunity to upgrade current programmable thermostats to become occupancy-aware without the need for additional infrastructure. Through simulating 100 homes as virtual test cases, they reported 0.42 kWh daily energy saving and an average MissTime of around 15 minutes.

As discussed in Ref. [114], MPC has been the most popular optimal control approach utilized for occupancy-based HVAC control. By considering future disturbances such as weather and occupancy changes, as well as using optimization algorithms and building models, MPC is able to dynamically estimate preconditioning time. MPC often showed superior performance than RB control systems [119]. Refs. [72,74] implemented an MPC algorithm to minimize the thermal discomfort of occupants and energy consumption. The optimization algorithm in the control system utilized a setback temperature when no one was present while preheating the building before the arrival time to improve thermal comfort. It was shown that the system led to 8% energy saving and kept the thermal comfort within the acceptable range. Turley et al. [75] demonstrated that utilizing MPC algorithms could decrease building energy consumption by almost 13%, compared with reactive control with an assumption of perfect occupancy prediction. Killian et al.

[72] investigated the performance of occupancy-based MPC in terms of the ability to reduce the temperature deviation from the desired setpoint. The results showed that using MPC decreased temperature violation by around 33%. A number of researchers demonstrated that despite the superior performance of occupancy based MPC, it causes too much complexity in developing the control system. Goyal et al. [71,109] demonstrated that including occupancy information in the control system can lead to almost 50% energy saving. However, they argued that the energy saving achieved by MPC algorithms might not compensate for the complexity induced in the control system, compared with the conventional control algorithms.

3.3. Methodology

3.3.1. System description

The schematic diagram of the RB control systems utilized in this study is demonstrated in Fig. 3.1. In this framework, occupancy information is collected using occupancy detection and monitoring networks, such as PIR sensors [54,56], Wi-Fi networks [50], cameras [66], or environmental sensors [94]. The occupancy data are continuously stored using database management systems (DBMS). The real-time and historical occupancy data are, then, utilized to train an occupancy model. As the control system makes online decisions, the occupancy model should be regularly updated using the most recent data. Next, the predicted occupancy patterns are utilized as an input to the RB controller. In contrast to conventional RB systems, which are independent of weather conditions, the proposed system receives weather data as input. The weather parameters as well as indoor operative temperature, as a feedback signal received from temperature sensors, are employed to make control decisions. The control decisions are utilized in the HVAC system to regulate setback/setpoint temperature in the monitored spaces. This process is repeated in each time step and the control decisions are updated continuously.

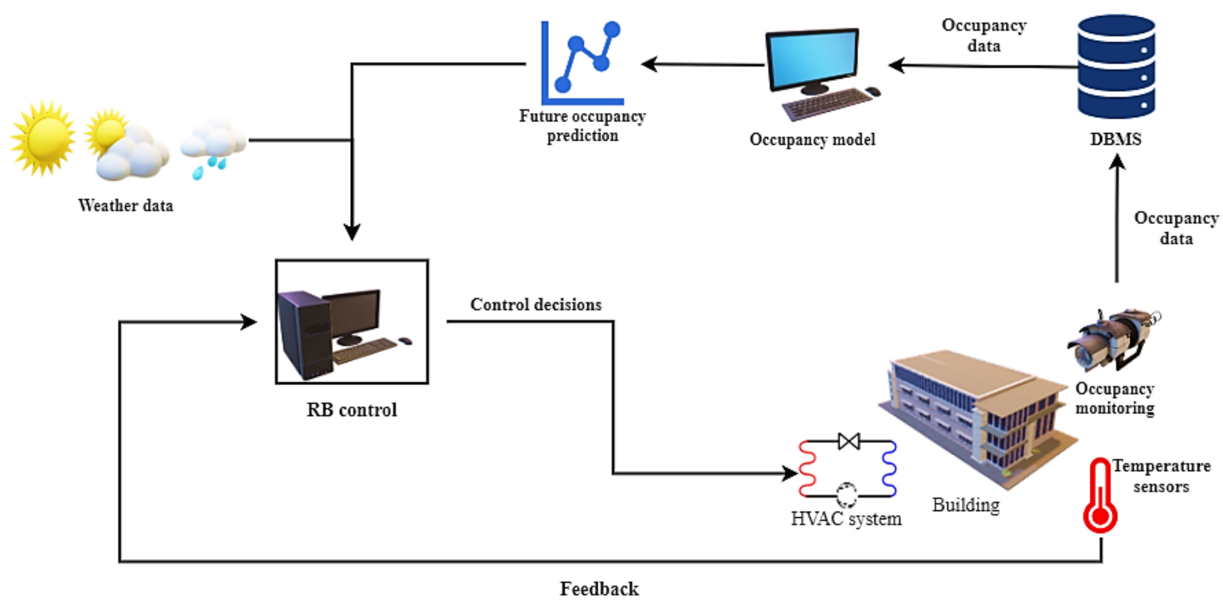


Fig. 3.1. Schematic diagram of the proposed RB control framework.

3.3.2. Occupancy database and model description

A real occupancy database, gathered from five residential units, described in Refs. [29,120], is utilized in this study to develop and evaluate the control systems. It is worth noting that because of privacy issues that might be caused for the households due to the monitoring process, there are a few occupancy databases available for use in this study, among which the selected database provides the most reliable and validated information [29,120]. More specifically, the occupancy monitoring system consisted of a relatively large number of PIR sensors, which is expected to improve the accuracy of occupancy detection in buildings [95]. 13 PIR sensors were implemented in each unit: two sensors were used in the living rooms, five sensors in corridors, and one sensor in each kitchen, bathroom, and bedroom.

The data were originally gathered at one-minute time intervals; however, the resolution is transformed to 30-min time intervals in the preprocessing step. One of the reasons for changing the resolution of occupancy data is linked with the data collection method using motion detectors. More specifically, the accuracy of PIR sensors for occupancy detection is dependent on the number and positioning of the sensors as well as on the mobility of the occupants. For example, the sensors might report no occupancy during some events when the mobility of the occupants decreases while watching TVs or taking a nap. As mentioned in Ref. [95], changing the resolution to 30-min intervals can help with addressing such errors associated with PIR sensors. It is worth noting that using more complex occupancy monitoring systems such as cameras, fine time intervals can also be adopted. However, such systems are rarely used in residential buildings due to privacy and cost issues. Besides, using 30-min time interval rather than the original one can enhance the computational efficiency of the control system. Occupancy states at each 30-min time interval can accept 0 or 1, respectively demonstrating the vacancy and occupancy states. A time interval shows an occupied state if at least one motion is detected by one of the sensors during this period. The raw data include some missing values for motion detectors in different records. During such periods when no signals are recorded from the sensors, the building is assumed to be occupied to ensure that the thermal comfort of occupants is not compromised. The data are stored in MySQL databases and are pre-processed using MySQL workbench [121].

As shown in Ref. [95], the accuracy of occupancy prediction models mainly varies in the range of 60-90% depending on the model and case study. Due to the rapid improvements in artificial intelligence and data analysis techniques, many different occupancy models have been proposed in the literature and their performance has been improved during the last decade. Hence, the performance of the control system might depend on the applied occupancy model. Furthermore, the performance can also depend on the size of the occupancy database; long-term monitoring might result in improved prediction performance of the occupancy model. To remove the dependency of the control system performance on the selection and development of the occupancy model, a perfect occupancy prediction assumption is made in this study. Perfect occupancy prediction was also utilized in earlier studies [70,122] to achieve an ideal upper bound performance of the control system.

3.3.3. Rule-based control

Fig. 3.2 demonstrates the flow diagram of the proposed rule-based (PRB) control framework. As can be observed, first, current indoor temperature and weather data are utilized as the input of the preconditioning-time model, described in Section 3.3.4. This model provides an estimation of the time required to reach the desired setpoint temperature as a function of input variables. Then, the estimated preconditioning time is utilized as the prediction horizon of the occupancy model. It means that the occupancy model forecasts the future occupancy states in a period that is equal to the preconditioning time. If a building is going to remain vacant during this period, the HVAC system is turned off, allowing the current temperature to approach the setback. Otherwise, the control algorithm initiates the HVAC operation to bring back the setpoint temperature before the occupant arrival. In this study, a minimum prediction horizon of one hour is considered to ensure that the thermal comfort conditions are met in most cases. This process is repeated in each time step to update the control decision.

In this study, to evaluate the effectiveness of the proposed control strategy, its performance is compared with that of two types of conventional control systems. The first one is an always-on thermostat, which is selected as the baseline. It maintains the setpoint temperature regardless of the occupancy patterns or any other factors. Conventional RB (CRB) control systems utilized in the literature, operating based on static preconditioning time, are also investigated in this study. Three CRB systems with different fixed preconditioning times of 60, 90, and 120 minutes are considered to investigate the impact of preconditioning time on the system performance.

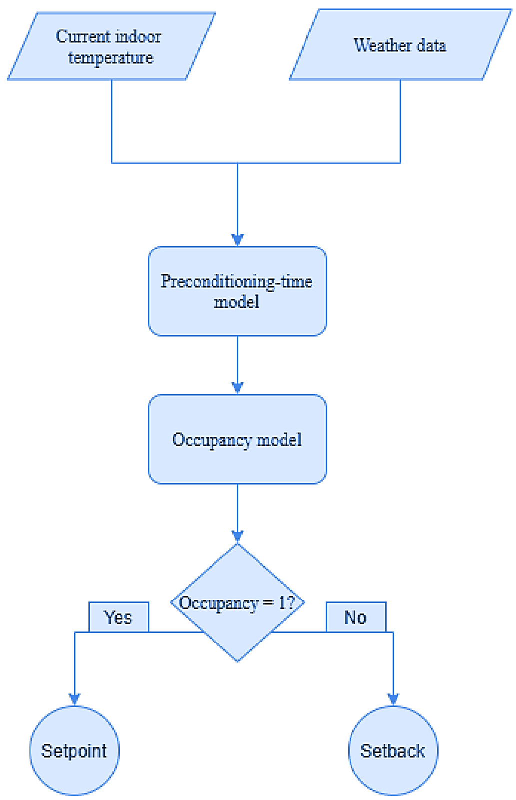


Fig. 3.2. The flow diagram of the RB control system proposed in this study.

3.3.4. Preconditioning-time model

As mentioned earlier and demonstrated in Fig. 3.2 , a model in the PRB control system is developed to provide an online estimation of the HVAC lag time. This model should be computationally cost-effective to be feasible for online control systems. However, using building simulation software is not the optimal approach due to its relatively low computational speed, required building physics knowledge, model development expertise, and relatively poor generalization potential. Data-driven models have attracted great attention because of their promising ability to address these issues and as a result, are implemented to predict preconditioning time. In this study, a multi-layer perceptron (MLP) network as a deep learning algorithm is employed. As shown in Table 3.1, seven candidate features, including indoor temperature as well as outdoor environmental attributes such as solar radiation and temperature, which can affect the prediction performance are selected for developing the model. In order to remove irrelevant features, a Pearson correlation coefficient as a filter feature selection method is adopted. The correlation coefficient among each variable and the output is demonstrated in Table 3.1. The correlation coefficients in the range of -0.3-0.3 are considered weak and eliminated from the feature set [123]. Fig. 3.3 shows that four features among the potential feature set are indoor temperature, outdoor temperature, horizontal solar radiation, and diffuse solar radiation. As the changes in the weather condition are mostly small in the relatively short-term periods of preconditioning time, the impact of the weather changes on the prediction performance is neglected.

Table 3.1.
Pearson correlation coefficients between the input variables and the output.

Variable	Pearson correlation coefficient
Indoor operative temperature	-0.86
Outdoor dry bulb temperature	-0.52
Horizontal solar radiation	-0.43
Diffuse solar radiation	-0.35
Direct solar radiation	-0.27
Precipitation	-0.08
Wind speed	-0.02

To construct the dataset required for training the MLP network, the energy simulation tool and the test case described in Section 3.4, are utilized. A complete simulation is performed for a five-month period from November 1st to March 31st to cover the entire heating period. In this simulation, temperature schedules are defined using a programmable thermostat, in which the HVAC system initiates the preconditioning process at 15:00 every day. A maximum duration of 6 hours (i.e. from 15:00 to 21:00) is associated with the preheating process. Over the rest of the day, the HVAC system remains off to provide enough time for the building to reach a deep setback. However, a minimum temperature of 10 °C is assumed to preserve the building from possible issues such as pipe freezing. A dataset with 217,440 elements is constructed using the mentioned simulation. It is noted that more simulations with different setback/setpoint schedules can be utilized for making a larger dataset to obtain a more accurate model [124]. However, in real-world applications, it takes several years to provide such a comprehensive dataset; thus, in this study, only one simulation is considered to provide an estimation of the system performance close to that of real ones.

The entire database is randomly divided into three parts: training, validation, and testing datasets, with a ratio of 80:10:10. The training and validation parts are utilized for the hyperparameter tuning process using a random-search method. The number of hidden layers, number of neurons in each layer, batch size, and the learning rate are determined in this process. After finding the optimal hyperparameters, the model is trained using both training and validation datasets and is applied to the testing set to achieve unbiased results for the model performance.

For training the model, the number of epochs is considered 20,000. In order to prevent the model from overfitting, an early stopping methodology is employed to stop the training procedure after 500 epochs without any improvement in the performance. Rectified linear activation function (ReLU) is implemented for each neuron, and an Adam optimizer is selected to minimize MSE for finding the optimal weights of the network. The model is developed using Keras library in Python [125].

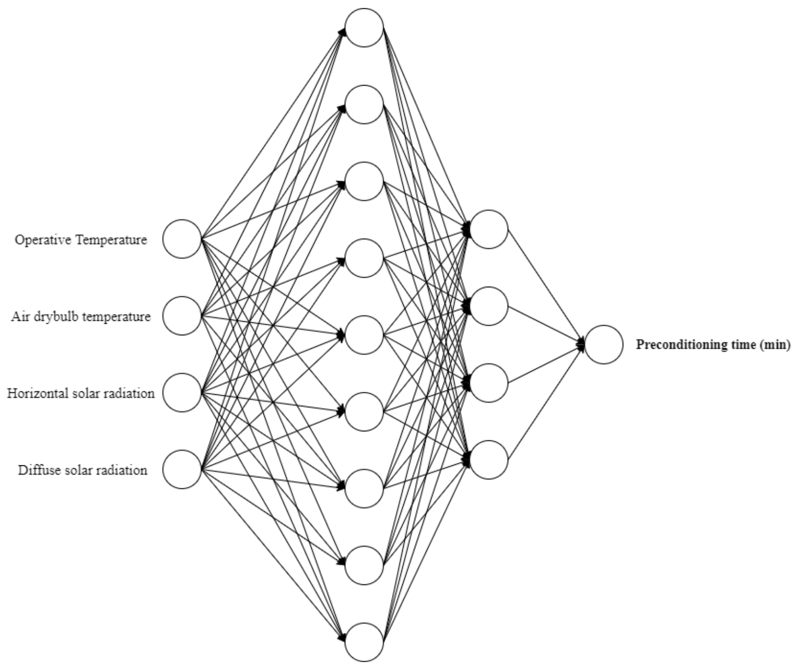


Fig. 3.3. The diagram of a sample multi-layer perceptron network implemented for preconditioning time estimation.

3.4. Building simulation

The developed control system is applied to a case study to evaluate its performance and effectiveness. As no HVAC data are available for the original residential building discussed in Section 3.3.2, a hypothetical building is considered in this study as a virtual testbed based on the floor plan of the original one. In order to construct a reliable energy model, EnergyPlus is used. EnergyPlus is an energy simulation tool, which has been widely validated against many testbeds such as those demonstrated in ASHRAE standard 140 [126]. It has been made as bug-free as possible [127] and can provide reasonable accuracy by considering detailed heat transfer models. However, the appropriate selection of input variables is essential to develop a reliable energy model. To this end, the input variables, including construction materials of the building, are selected based on ASHRAE standards [128] available as OpenStudio libraries and EnergyPlus weather data [129].

The 3D model as well as the 2D plan of the building, designed in the SketchUp software [130], is demonstrated in Fig. 3.4. The 3D model is imported into EnergyPlus via OpenStudio plug-in [131]. As discussed in Section 3.3.2, five occupancy datasets collected from different residential units are implemented in five separate building simulations to consider the influence of occupancy behavior on the system's performance. Electric baseboards are considered as the heating equipment and as a result, electricity is the main source of providing heating demand in this testbed. The size of the heating equipment is determined using the auto-sizing feature in EnergyPlus for the always-on control. Then, the determined size is also utilized for other control systems.

It is noted that in each time step of the simulation, indoor temperature and weather data are required for estimating the preconditioning time, which is, then, utilized to make control decisions for regulating the setpoint temperature in the following step. In other words, after completing the simulation in each time-step, the setpoint for the following time-step needs to be determined and manipulated in EnergyPlus. Additionally, the results need to be retrieved and stored before starting the simulation for the next step. This process cannot be performed as a one-time simulation that is conventionally utilized in the literature. To deal with this task, EnergyPlus/Python application programming interface (API) [132] is employed, which provides flexibility for the users to manipulate EnergyPlus variables from the Python environment. This API also makes it possible to retrieve and store EnergyPlus results in Python after completing each simulation. The interactions between software and different libraries used in this study are demonstrated in Fig. 3.5.

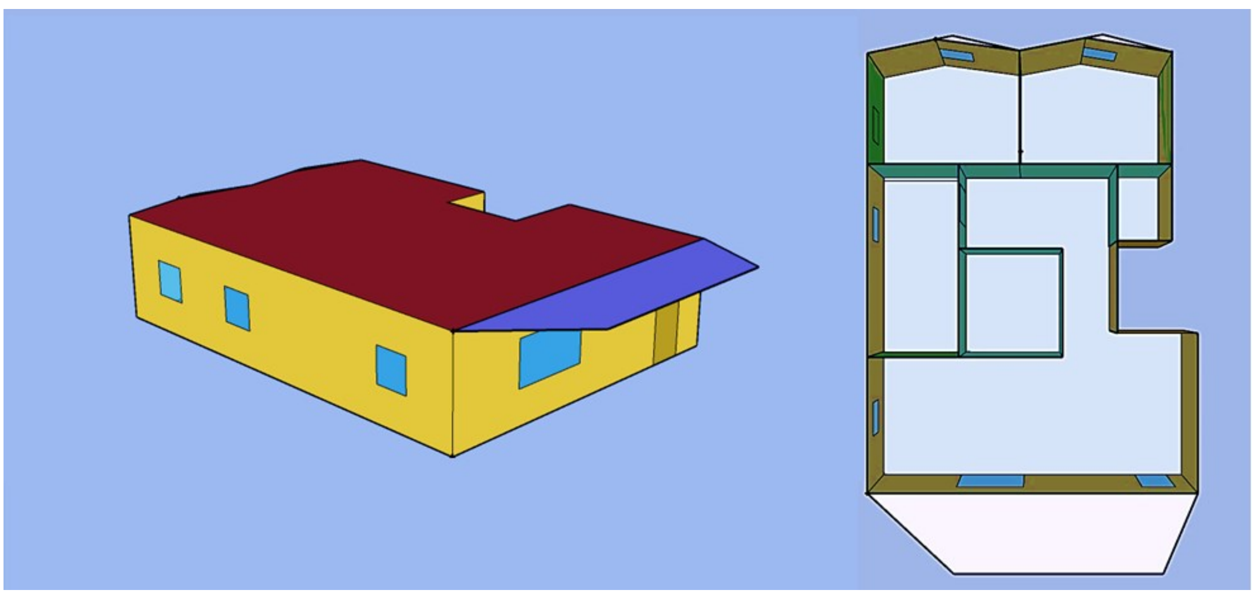


Fig. 3.4. The designed model of the simulated building as the case study.

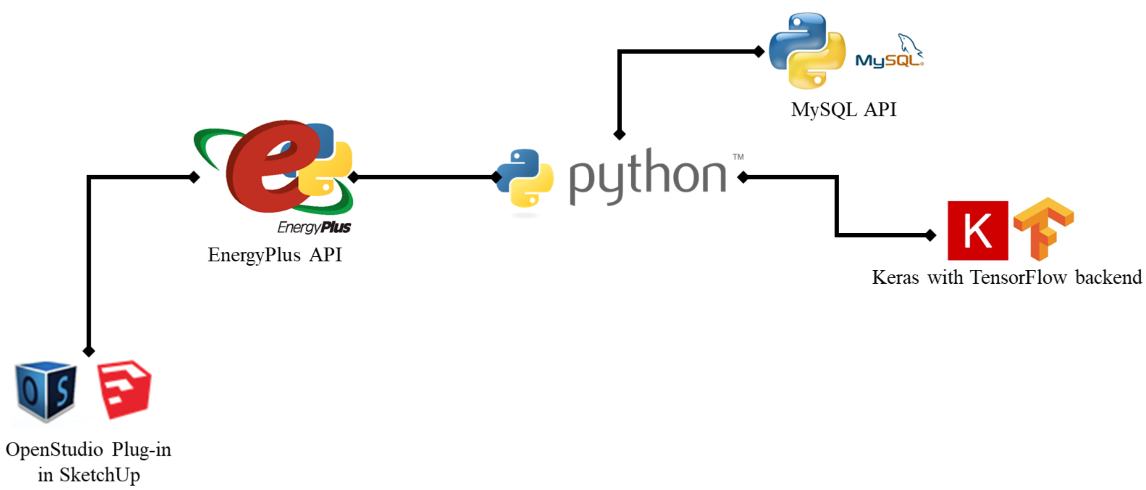


Fig. 3.5. Interactions between different applications and libraries used in this study.

3.5. Performance evaluation criteria

3.5.1. Financial indicators

The economic merits of the proposed control system could be the most important decision-making criteria for many owners. It is because one of the main barriers to retrofitting the buildings with smart thermostats is often their relatively high cost. Knowing about the long-term economic gains achieved through this investment can be a great incentive in such cases. In this section, a comprehensive financial analysis of the system is performed by estimating the main economic indicators, including IRR, ATCSR, DPB, and NPV associated with replacing old thermostats with smart ones.

First, it is needed to estimate the annual operating cost of each system. In this study, it is assumed that the operating cost equals the utility cost, which is estimated according to the local electricity tariffs for residential buildings as follows [133]:

$$C_{op} = C_{tier1} + C_{tier2} \tag{1}$$

where C_{tier1} and C_{tier2} indicate the electricity cost in the first and second tier, respectively. In this two-tiered pricing, the electricity price per kWh varies based on the total amount of electricity consumption. More specifically, if the total electricity consumption by customers during a billing period, E, is less than the first-tier limit, E_{tier1}, they are billed based on the electricity tariffs in the first tier, c_{tier1}. If their consumption exceeds this limit, the rest of their consumption, $E - E_{tier1}$, is billed based on the electricity tariffs in the second tier, c_{tier2}, which is higher than that in the first tier. Accordingly, the electricity costs in each tier can be calculated using the following relationships:

$$C_{tier1} = c_{tier1}.\max(E, E_{tier1})(1+t)$$
$$C_{tier2} = c_{tier2}.\max(0, E - E_{tier1})(1+t) \qquad (2)$$

where c is the cost of electricity per kWh in each tier, E denotes the amount of electricity consumption, E_{tier1} represents the energy consumption limit associated with the first tier, and t denotes the tax rate.

Smart thermostats are expected to reduce the annual operating costs by saving energy, compared with conventional thermostats. In this regard, the annual benefits, B_{ann}, is defined using the following relationship to quantify the amount of annual cost saving:

$$B_{ann} = C_{op} - C_{op,baseline} \qquad (3)$$

where $C_{op,baseline}$ denotes the operating cost associated with the always-on control. Using the annual benefits, NPV can be calculated. NPV demonstrates the difference between the initial investment cost, C_{cap}, of smart thermostats and the equivalent present value of future benefits. A positive NPV shows that the project is profitable and negative values show net loss associated with the current project. NPV can be defined using the following equation [134]:

$$NPV = \frac{B_{ann}}{CRF} - C_{cap} \qquad (4)$$

The first term represents the present value of the future benefits of the system, in which CRF denotes the capital-recovery factor and is defined as follows:

$$CRF = \frac{d(1+d)^n}{(1+d)^n - 1} \qquad (5)$$

where d is the discount rate (also called interest rate), and n is the life span of the system. In order to take into account the annual inflation rates, that can lead to an increase in the future electricity prices, the discount rate is defined based on the market interest rate, i, and inflation rate, f, as follows [135]:

$$d = \frac{i-f}{1+f} \qquad (6)$$

Based on Eq. (4) for NPV, IRR can be defined as the discount rate that can make NPV equal to zero. IRR is utilized as another indicator to quantify profitability by estimating the expected yearly rate of return. It can be calculated by solving the following equation through trial and error [136]:

$$\frac{(1+IRR)^n - 1}{IRR(1+IRR)^n} - \frac{C_{cap}}{B_{ann}} = 0 \qquad (7)$$

Having the annual benefits and the capital costs for each system, the DPB can also be estimated. DPB estimates the time it takes for a project to break even. In other words, it shows the point in time when the future benefits fully cover the capital cost while respecting the time value of money. DPB can be estimated using the following equation [137]:

$$DPB = \frac{\ln\left(\frac{B_{ann}}{B_{ann} - C_{cap} d}\right)}{\ln(1+d)} \tag{8}$$

ATCSR is utilized as an indicator to estimate how much total cost can be saved annually by replacing conventional systems with smart thermostats. ATCSR can be estimated using the following equation [138]:

$$ATCSR = \frac{ATC - ATC_{baseline}}{ATC_{baseline}} \tag{9}$$

where ATC is the system total annual cost. It is noted that ATC considers the investment costs as well as the annual operating cost. For this purpose, the capital recovery factor is utilized to calculate an equivalent annual cost for the initial investment as follows [139]:

$$ATC = \mathrm{C}_{capital}.\mathrm{CRF} + \mathrm{C}_{op} \tag{10}$$

The parameters required for the financial analysis are summarized in Table 3.2.

Table 3.2.
The economic parameters utilized for the financial analysis in this study.

Parameter	Unit	Value	Ref.
Market interest rate, i	%	5.0	
Life span, n	Year	20	[140]
Capital cost, C_{cap}	CAD	250[1]	
Electricity price in the 1st tier, c_{tier1}	CAD/kWh	0.0608	[133]
Electricity price in the 2nd tier, c_{tier2}	CAD/kWh	0.0938	[133]
1st tier limit for electricity consumption, E_{tier1}	kWh/day	40	[133]
Tax rate, t	%	14.975	[141]
Inflation rate, f	%	2.0	[142]

[1] The average price of smart thermostats available in the market including tax.

3.5.2. Energy and peak-demand management indicators

Energy-efficiency has been the most popular indicator for evaluating the performance of occupancy-based HVAC operations in the literature [114]. In this study, the annual energy saving ratio (AESR) is utilized to quantify the energy-efficiency performance of the systems. This indicator demonstrates the annual amount of energy that can be saved using smart thermostats:

$$AESR = \frac{AE - AE_{baseline}}{AE_{baseline}} \tag{11}$$

where AE denotes the annual energy consumed for covering the heating demand in kWh associated with each system using different setback temperatures. One of the limitations of this definition is that it does not consider the timing of energy consumption when evaluating the performance and consequently, cannot reflect the potential of demand-response management. To address this issue, the annual energy saving is also calculated for mid- and on-peak periods, as defined in Fig. 3.6.

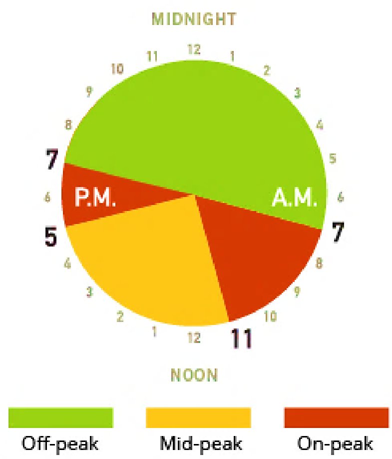


Fig. 3.6. Off-, mid-, and on-peak periods considered for evaluating the peak-demand performance of the systems [143].

3.5.3. Thermal comfort metrics

As providing the thermal comfort requirement is the main purpose of using HVAC system, it is of essential importance to include comfort criteria when evaluating HVAC control system. For this purpose, MissTime which is defined as the period when occupants encounter a temperature that deviates from the desired setpoint is employed in this study [17]. In other words, any period when occupants are present, and the temperature is less than the setpoint is considered as thermal discomfort. In this study, an operative temperature of 21.5 °C is considered as the setpoint (i.e. desired temperature) for occupants.

3.6. Results

3.6.1. Performance of the preconditioning-time model

To determine the optimal structure of the MLP model, a random search-based method for tuning the hyperparameters is implemented. In this process, the number of hidden layers, number of neurons in each layer, batch size, and the learning rate in the optimization algorithm are explored as hyperparameters. This method is an iterative technique that randomly tries different values of these hyperparameters from predefined ranges, represented in Table 3.3. An MLP model is trained based on the selected hyperparameter in each iteration and its performance in terms of MSE is evaluated and recorded. This process is repeated at every iteration and stops when the maximum number of iterations, which is 100 iterations in this study, is reached. The hyperparameters that result in the best model, providing the lowest MSE, are chosen to train the final preconditioning-time model. The optimal values for the parameters obtained in this process are shown in Table 3.3.

Table 3.3.
The range of hyperparameters used in the tuning process as well as the optimal values obtained.

Parameter	Range	Selected value
Number of layers	[1-4]	2
Number of neurons	[2, 6, 16, 32, 64, 128, 256]	First layer: 65 Second layer: 256
Learning rate	[0.005, 0.01, 0.05, 0.1]	0.05
Batch size	[5,000, 10,000, 15,000]	5,000

Using the optimal hyperparameters, the final MLP model is trained using both training and validation sets. Then, the model is applied to the test set to obtain unbiased results for evaluating the performance. Fig. 3.7 represents MAE after each epoch for training and testing datasets. It is observed that the training process is completed before completing 15,000epochs of training, due to early stopping conditions explained in Section 3.3.4. Ultimately, the model provides a test MAE of 2.68 min. As the control decisions are made at 30-min intervals, the performance of the MLP algorithm is considered reasonable in this control framework.

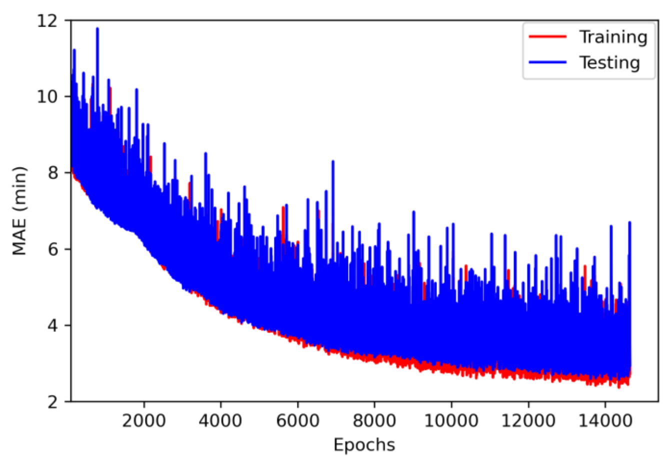


Fig. 3.7. The performance of the final MLP model after each epoch of the training process.

3.6.2. Multi-criteria decision making

This section is devoted to selecting the best-performing RB control among the conventional and proposed systems. As the energy-efficiency and financial merits are consistent, using NPV as an indicator among these factors suffices for the decision-making process. NPV associated with each RB system in different setback bounds is demonstrated in Fig. 3.8. In this figure, CRB-1, CRB-2, and CRB-3 respectively denote the conventional RB control systems with 60-minute, 90-minute, and 120-minute preconditioning time. It is observed that CRB-1 resulted in the highest economic benefits in all temperature bounds with a median NPV of up to \$1,643. Naturally, when the preheating time increases in CRB systems, the economic merits decrease as well, as more energy is consumed for preheating the space. Using a longer preheating time can even lead to a negative NPV in the conservative setback range with a median NPV of as low as \$13. It shows that replacing old thermostats might not be profitable in some cases. In all the temperature bounds, CRB-1 is followed by the PRB system, with a median NPV of up to $1545 for the deepest setback and a median NPV of $289 for using a conservative temperature. It is observed that employing the PRB system always leads to a positive NPV in all cases.

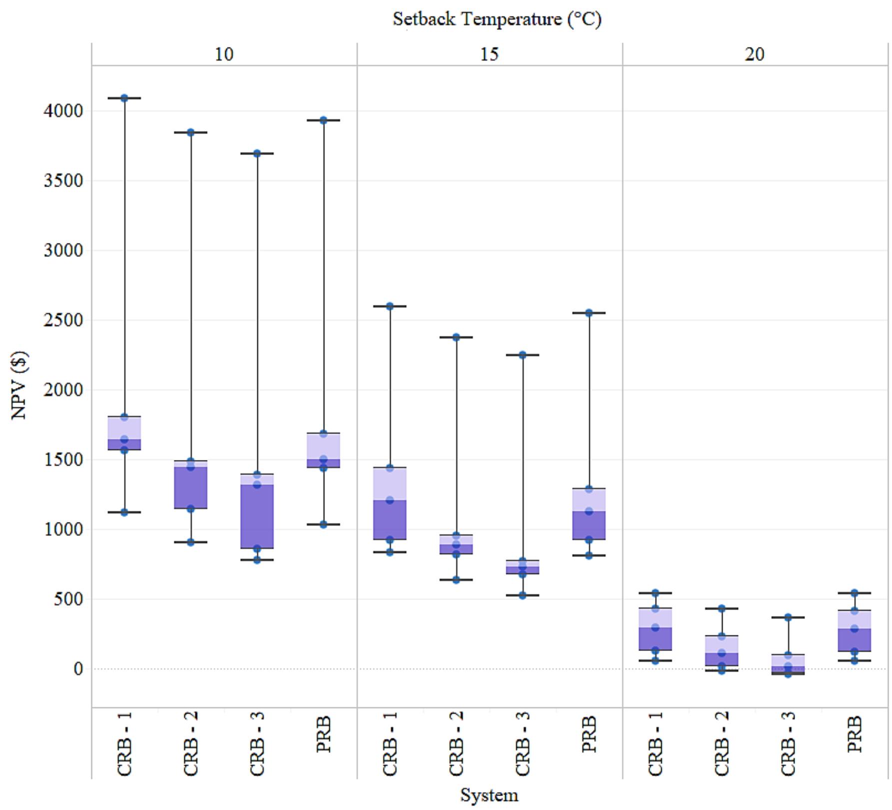


Fig. 3.8. The distribution of NPV for the RB control systems in different setback bounds.

Fig. 3.9 illustrates the distribution of MissTime for the control systems in different temperature setback bounds. In contrast to the economic merits, CRB-1 provides the lowest MissTime at a median of up to almost 30 hours. Increasing the preconditioning time for the conventional control strategies clearly reduces the MissTime due to using longer preconditioning time that helps the HVAC system to reach the setpoint prior to occupants' arrival. The PRB control provides the shortest MissTime in the deep setback range at a median time of 5 hours. As for the setback temperature of 15 °C, CRB-3 and PRB systems result in the same performance with no thermal discomfort in most cases. For a conservative setback, using all control systems causes no thermal discomfort for all cases.

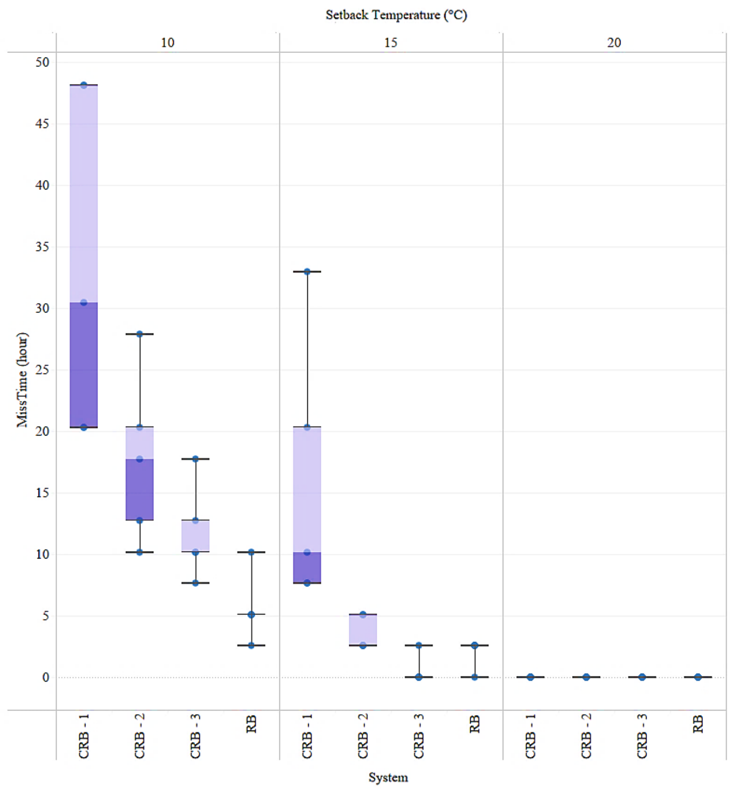


Fig. 3.9. The distribution of MissTime for RB control systems in different setback temperature bounds.

As demonstrated above, economic and thermal comfort objectives are mostly conflicting for the RB control systems. In other words, none of the systems outperform other alternatives regarding both objectives, except for using a conservative temperature bound, in which CRB-1 provides the best MissTime and NPV. Therefore, there is not a unique optimal control system for deep and medium setback bounds. To systematically choose the best system among these alternatives, TOPSIS (Techniques for Order Preferences by Similarity to the Ideal Solution) method, as a multi-criteria decision-making approach, is employed. The first step in the decision-making process is constructing an m×p matrix, called a decision matrix [144], in which m indicates the number of alternatives (i.e. the candidate solutions in the decision-making process) and n is the number of criteria. As represented in Table 3.4, there are four different alternatives and two decision-making criteria in this problem for each temperature bound, and as a result, the decision matrix has a shape of 4×2. Next, the matrix is nondimensionalized using the following relationships [144]:

$$N_{ij} = \begin{cases} \dfrac{x_{ij} - \min\limits_i x_{ij}}{\max\limits_i x_{ij} - \min\limits_i x_{ij}} & NPV\ criterion,\ j = 1 \\ \dfrac{\max\limits_i x_{ij} - x_{ij}}{\max\limits_i x_{ij} - \min\limits_i x_{ij}} & MissTime\ criterion,\ j = 2 \end{cases} ,\ i = 1, \ldots, m \tag{12}$$

where N and x respectively indicate the elements of the normalized and original matrices. The optimal alternative has the smallest distance from the Positive Ideal Solution (PIS) and the largest distance from the Negative Ideal Solution (NIS) [145]. In this study, PIS is a point that has the highest NPV while having the least MissTime, and NIS is the opposite point to PIS. Having PIS and NIS, the distances of all the alternatives from these two points are measured using the traditional Euclidean metric. In order to find the best point in terms of their distances from PIS and NIS, a relative closeness indicator, R_i, for each alternative is defined as follows:

$$R_i = \frac{d_i^{NIS}}{d_i^{PIS} + d_i^{NIS}}, \quad i = 1, \ldots, m \tag{13}$$

where d_i^{NIS} and d_i^{PIS} are the distances of the i^th alternative from the NIS and PIS, respectively. The alternative with an R_i closest to 1 is selected as the solution. The values of the relative closeness for the alternatives are represented in Table 3.4. As can be observed, the PRB control provides the highest R at 0.84 and 0.85, respectively for deep and medium bound setback and is selected as the optimal RB systems.

Table 3.4.

The mean NPV and MissTime for RB control systems used in the decision-making process as well as the relative closeness index achieved from the TOPSIS method.

System	Deep setback			Medium setback			Conservative Setback	
	MissTime (hour)	NPV ($)	R	MissTime (hour)	NPV ($)	R	MissTime (hour)	NPV ($)
CRB-1	33.44	2,045	0.5	15.71	1,400	0.5	0	289
CRB-2	17.73	1,765	0.46	3.55	1,135	0.56	0	153
CRB-3	11.65	1,607	0.43	0.51	990	0.5	0	79
PRB	5.57	1,937	0.84	2.03	1,325	0.85	0	283

3.6.3. Impact of setback temperature

Fig. 3.10 demonstrates the impact of setback temperature on the MissTime and NPV of the selected RB control systems. It is observed that increasing the setback from 10 °C to 15 °C can lead to a 31.61% fall in the NPV while improving the MissTime by 63.64%. Similarly, using a conservative setback can cause a 78.66% further reduction in the financial profitability of the system and a 100% decrease in the MissTime. Table 3.5 summarizes the average values of performance indicators for the PRB control systems based on different setback bounds. It can be observed that implementing an RB control using conservative setback temperature does not provide compelling economic performance. It leads to a payback period of 10.92 years, which is approximately three times longer than that obtained for using a medium setback. It also provides an ATCSR of 1.81%, which is one-fifth of that for using a medium setback bound. In terms of energy-efficiency, this system results in almost 3.64% energy saving without compromising thermal comfort. By relaxing the constraints on setback temperature and using a setback of 15 °C, the economic performance considerably increases; it leads to a payback period of 3.58 years, reduces annual total cost by 8.9%, and decreases annual energy consumption by 9%. By decreasing the setback to 10 °C, the performance of the system, except for the MissTime, can be further improved. It results in a payback period of 2.74 years, which is almost a 23% improvement compared with that of the medium setback. A closer look at this table reveals that the improvements achieved by the transition from the conservative to medium setback temperature are much higher than those obtained from the medium to deep setback. The reason is linked with the thermal mass of the building, which can store energy and resist temperature reduction. Due to this passive energy storage, long vacancy periods are required to reach a setback temperature of less than 15 °C. However, as occupants often occupy the buildings for most of the time, the temperature can rarely decrease to a deep setback range.

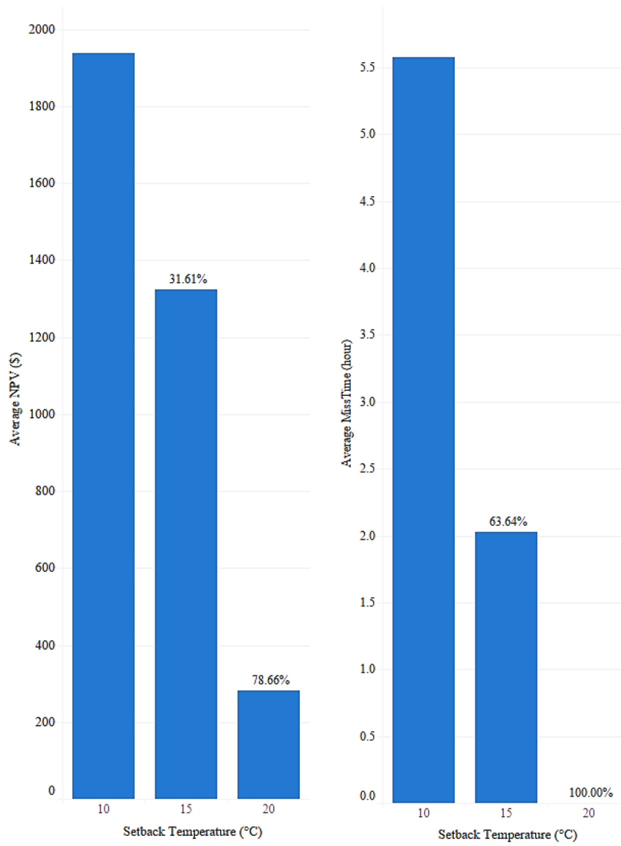


Fig. 3.10. Average MissTime and NPV for the selected RB control systems based on the allowed setback temperature.

Table 3.5.
The average performance indices for the selected RB control systems in different setback bounds.

Performance indicator	Unit	Setback temperature		
		10 °C	15 °C	20 °C
AESR	%	12.24	8.92	3.29
ATCSR	%	12.91	8.81	1.83
Electricity consumption	kWh/day	64.24	66.67	70.80
DPB	Year	2.72	3.63	10.87
IRR	%	45.12	32.75	10.25
NPV	CAD	1,937	1,325	283
MissTime	Hour	5.57	2.03	0

3.6.4. Peak-demand performance

To study the impact of using occupancy information in the control systems on the peak-demand performance, the average hourly energy consumption of the baseline and PRB control systems for a sample case is depicted in Fig. 3.11. It shows that during the off-peak period, from midnight to 7:00, both the baseline and the PRB control system consume almost the same amount of energy. It is because of the fact that during this period, the occupants are often sleeping and occupying the buildings. However, the PRB control starts to save energy during the morning on-peak period, 7:00 – 11:00. As summarized in Table 3.6, the PRB control can save energy by 17% during this period using a medium setback range. The highest amount of energy saving happens during the mid-peak when people usually go outside, and the building is likely to be vacant; the PRB control can save energy by 18% during this period with the medium setback. However, in the on-peak period, from 17:00 to 19:00, the baseline system considerably outperforms the PRB control. More specifically, using occupancy-based control can cause an 8% increase in the peak energy demand using a medium setback. It is because of the fact that when people are outside of their homes during the day, the indoor temperature gradually decreases towards the setback. Hence, before the arrival time, there would be a sudden change in the HVAC operation (energy consumption) to preheat the building and provide the thermal comfort condition as soon as possible. Such sudden changes can significantly increase the on-peak demands during this period. The energy consumption of PRB control still remains slightly higher than that of the baseline during the period between 19:00 and 23:00. It is because although occupants have mostly returned home during this period, the temperature of the building envelopes often remains low. Therefore, an extra amount of energy is typically consumed during this period to recharge the thermal mass of the building.

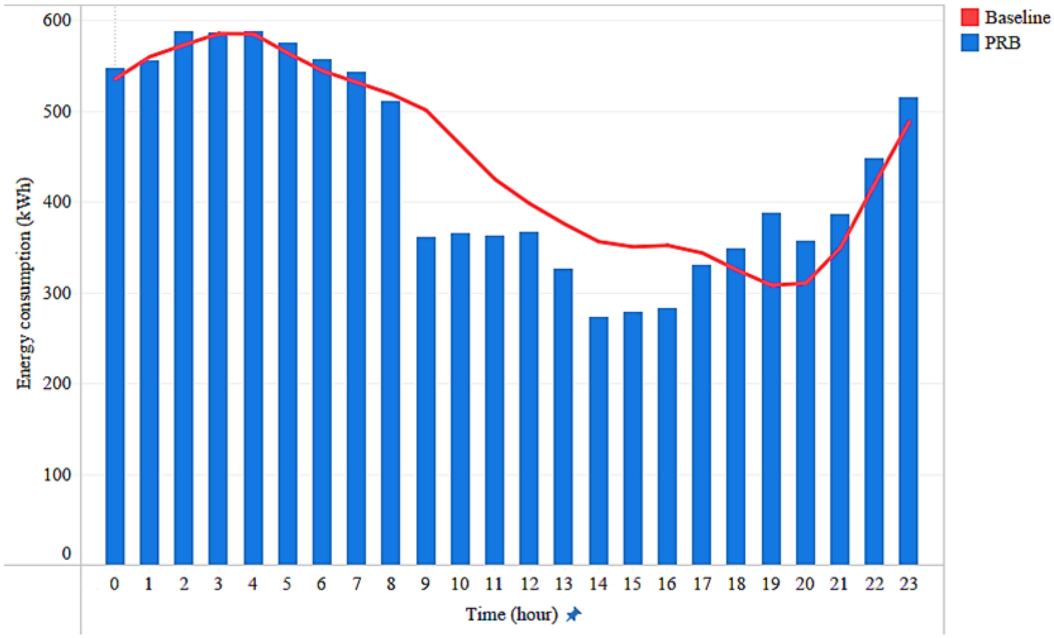


Fig. 3.11. Average hourly energy consumption of the PRB system using the medium setback.

Table 3.6.
On-peak and mid-peak energy consumption using RB control with different setback temperatures.

Setback	7:00-11:00 peak period		11:00-17:00 mid-peak period		17:00-19:00 peak period	
	Energy consumption (kWh)	Energy saving	Energy consumption (kWh)	Energy saving	Energy consumption (kWh)	Energy saving
10	1652	18%	1850	19%	734	-10%
15	1682	17%	1877	18%	720	-8%
20	1871	8%	2132	7%	703	-5%
Always on	2029	-	2295	-	669	-

3.7. Conclusion

In this chapter, an RB control system, using dynamic estimations of preconditioning time and future occupancy patterns, is proposed, and its performance is assessed in terms of economic, energy, peak-demand management, and thermal comfort. In contrast to conventional RB control systems that use static preconditioning time for regulating indoor temperature, this proposed system takes advantage of a deep learning algorithm to provide online estimations as a function of indoor temperature and outdoor environmental features. To evaluate the impact of setback temperature on the system performance, deep, medium, and conservative setback ranges are considered. The main findings can be summarized as follows:

1. The MLP model is able to provide reasonable performance for estimating the preconditioning time with an MAE of 2.68 min,

2. Using the TOPSIS method, it is shown that employing dynamic estimations of preconditioning time in RB control can improve the overall performance of the system in deep and medium setback bounds. However, no improvement is observed when a conservative setback is concerned,
3. The financial analysis demonstrates that replacing old thermostats with smart ones can result in up to 12.91% saving in annual total cost, 2.72 years of payback period, and $1,937 net present value,
4. Implementing conservative setback temperature in the control systems can significantly decrease the profitability of the thermostats, leading to a median DPB of 10.87 years. In some cases, negative NPV is obtained, showing that using such thermostats can cause net loss,
5. It is revealed that occupancy-based control systems can inversely impact the on-peak energy consumption of the control systems. It is demonstrated that it can increase up to 10% of the on-peak energy consumption due to the sudden rises in the HVAC energy consumption prior to occupants' arrival. This negative impact highlights the necessity of taking peak-demand management into account while developing occupancy-based control systems.

It is noted that although using a building block or a district as a case study can provide more generalizable results in terms of financial, energy saving, and peak demand performance, this study was limited to consideration of a single house as the case study to evaluate the performance of the proposed system. Hence, more comprehensive case studies can be considered for investigating the RB control systems in future work. Additionally, the occupancy-based control systems are examined with an assumption of perfect occupancy prediction in this study, and as a result, this study provides an upper bound level of the system's performance. Implementing actual occupancy models in the control systems is proposed as future work to evaluate the impact of occupancy prediction errors on system behavior.

Chapter 4: Deep learning models for future occupancy prediction in residential buildings[3]

4.1. Overview

This chapter contributes to the occupancy prediction problem by developing state-of-the-art deep learning models. To this end, the occupancy prediction problem is addressed from two different viewpoints: multi-label classification and a sequence-to-sequence time-series analysis using encoder-decoder architectures. The following deep learning algorithms are employed in this study to construct occupancy models: multi-layer perceptron (MLP), recurrent neural networks, long-short term memory (LSTM), gated recurrent units (GRU), and bidirectional LSTMs. The performance of these models is evaluated and compared in terms of accuracy and computational speed. The results demonstrate that addressing this problem using MLP models provides the best performance for short-term predictions, while for predictions more than 90 minutes ahead, GRU results in the highest accuracy. It is also demonstrated that the accuracy of the deep learning models can be approximated as a function of the occupancy index with an MAE of 0.014.

4.2. Introduction

In 2019, residential and commercial sectors were responsible for 28% of the overall energy consumption in the U.S [146]. This huge amount of energy demand had also a significant impact on the peak load [147,148]. Intelligent control strategies, with the ability to learn occupancy patterns in buildings, have shown promising abilities to address this issue by increasing building performance. Gao and Whitehouse [76] demonstrated that using control systems with occupancy prediction can save energy by 15% and improve thermal comfort by up to 40% in comparison with conventional thermostats. Because of the important role that occupancy prediction models play in intelligent control systems, researchers have applied a wide variety of machine learning models to occupancy prediction in the past decade. k-means clustering, decision trees [149], support vector machines (SVM), random forests, k-nearest neighbors (kNN) [82,84], linear regression, and logistic regression [95] were previously applied for occupancy prediction. However, Huchuk et al. [95] demonstrated that the accuracy of the previously proposed occupancy models is mostly limited in the range of 65% to 85% depending on actual occupancy profiles. Gluck et Al. [17] estimated that reducing the errors in occupancy models by 10% can improve the occupants' thermal comfort by 16% and enhance energy saving by 9%. It highlights the necessity of putting more effort to develop occupancy models that are able to capture hidden patterns in occupancy profiles to provide higher performance.

Despite the importance of occupancy prediction models, researchers have rarely focused on developing deep learning models to predict occupancy profiles. This study aims to fill this research

[3] This chapter is based on the following publication: M. Esrafilian-Najafabadi, M. Babahaji, F. Haghighat, Deep learning models for future occupancy prediction in residential buildings, in: 5th International Conference on Building, Energy and Environment 25th-29th July, 2022.

gap by developing different deep learning algorithms to forecast future occupancy patterns based on real-world occupancy data collected from five apartments. In this study, the occupancy prediction problem is formulated as multi-label classification and time-series sequence-to-sequence (S2S) forecasting. For this purpose, multi-layer perceptron (MLP) as well as recurrent neural networks (RNN), long short-term memory (LSTM), gated recurrent unit (GRU), and bidirectional LSTM (BLSTM) using auto decoder-encoder architectures are used. The performance of these models is assessed in terms of computational cost and prediction accuracy. The optimal structures of the models are determined using a random search-based hyperparameter tuning process. Additionally, in order to evaluate the impact of the occupancy profiles on the prediction accuracy, the occupancy index is proposed in this study. The correlation between the proposed index and the accuracy is assessed using Pearson correlation. Next, this index is utilized to approximate the accuracy of deep learning models as a function of the nature of occupancy profiles in buildings. This estimation provides an insight into the performance of occupancy models based on deep learning algorithms.

4.3. Method

4.3.1. Database description and data pre-processing

In this study, a dataset collected from 5 three-bedroom apartments is used to develop the deep learning models. The data was collected over a one-year period in 2016 using passive infrared (PIR) motion sensors. The dataset is imported into the MySQL database management system for the preprocessing step. The data was originally collected in 1-min intervals, which is too short to train deep learning models; hence, the resolution of the dataset is processed from 1-min intervals to 30-min intervals.

The time of day as the most utilized feature in the literature is selected to train occupancy prediction models. To include the time of day as a feature, every time step is represented by an integer number from 1 to 47. 1 in this scale shows the time step from midnight to 00:30, and 47 denotes the period between 23:30 to midnight. However, this feature should be transformed into dummy variables because it is not a numeric variable in nature. In this way, the time-of-day feature transformed into 47 different features. Each new feature takes a value of **1** if the data element is recorded in the associated time of day and otherwise takes **0**. For example, the first feature which is associated with 00:30 takes **1** only for the elements recorded during this period and all other dummy variables take **0** in this case. Additionally, as discussed in [86], whether the prediction is performed on weekdays or weekends can have a positive impact on the prediction performance. Therefore, the *Weekend* feature is also used to train the models; this feature can accept 1 and 0, respectively showing that the prediction is performed for the weekends and weekdays. As well as the mentioned features, the occupancy states from previous 24 timesteps (i.e., 12 hours) are also added as candidate features.

4.3.2. Deep learning models for occupancy prediction

In this study, an MLP algorithm is implemented to address occupancy prediction as a multi-label classification problem, in which each data record is associated with eight labels (i.e. occupancy states at the future eight time intervals, which is equivalent to a prediction horizon of 4 hours). In addition, an RNN model is also employed to address the occupancy prediction as a time-series prediction problem. In contrast to the MLP architecture, in which all the features are fed into the first hidden layer regardless of the timely order of the features, in RNN algorithms, the features are separately fed into the network in each time step. Then, the output of the previous layers of RNN is used in the proceeding layers until the current time step. These connections between time steps enable the model to discover the temporal correlations between the occupancy states at each time step. In addition, using encoder-decoder architectures for this S2S problem enables the model to learn the correlations between future time steps. This architecture consists of two sub models of encoders and decoders.

Theoretically, RNN is very powerful to learn the temporal relationships between each time step. However, from the practical point of view, they can be hard to train properly. The *Vanishing gradient* is mentioned as one of the main reasons for this issue [150]. LSTM networks are widely employed to overcome the vanishing gradient problem. LSTM is similar to RNN with the distinction that it employs memory blocks in each hidden layer with three gates, namely input, output and forget gates. This architecture enables LSTM to remember important correlations from temporally distant events to the current time step [151]. GRU was also proposed as a solution to the vanishing gradient problem of RNN. GRU is a gated approach similar to LSTM, while it uses two gates, namely, update and forget gates in the model. As stated in [152], in some cases, GRU could provide higher computation speed and higher accuracy compared with the performance of RNN.

All the mentioned time series algorithms can be also utilized in bidirectional networks. These structures utilize two separate hidden layers; one of them processes the input vector in the forward direction while the other one processes the input in the opposite direction. These two hidden layers are connected to the same output layer in the network. This structure enables the model to use the information alongside both backward and forward directions of the input sequence [153]. Although it has been shown that bidirectional networks can perform better in a variety of fields such as phoneme classification [154], no previous studies have reported the results of implementing this approach for occupancy prediction. To fill this gap, a BLSTM algorithm, is also developed as an occupancy model in this study.

4.3.3. Model evaluation

To develop and evaluate the performance of occupancy models in terms of accuracy and computational time, each database is split into three sets in this study: a training set, a three-month period from February 1st to June 1st, a validation set, the entire month of July, and a test set, the entire month of August. First, the training and validation sets are employed to find the optimal values of hyperparameters through a random search-based hyperparameter tuning process with 150 iterations. The number of neurons and hidden layers as well as batch size and learning rate are considered the hyperparameters. In this process, a number of candidates hyperparameters are randomly selected and utilized to construct the deep learning models. Then, the performance of the models associated with each set of hyperparameters is evaluated. The values that lead to the highest performance are selected as the ultimate parameters of the network. The number of epochs is selected as 300 and an early stopping method with the *patience* of 50 iterations is employed to avoid the overfitting problem. Binary cross-entropy is chosen as the loss function, which is optimized using Adam optimizer. In the next step, the test dataset is utilized, as unseen data, to estimate the performance of the final occupancy models. In this step, as well as the training set, the validation set is also implemented for training the model, as this part of the data carries the most recent information related to the testing set.

The data preprocessing step is performed using MySQL workbench, and MySQL/Python APIs are utilized to retrieve queries from MySQL in the Python environment to develop the occupancy models using Keras 2.4 with TensorFlow backend. The computations are performed on the Google Collaboratory [155] (or Colab for short). Deep learning models are executed using GPUs on this platform with 12 GB of RAM available.

4.4. Results

The accuracy of each model on the test dataset for the apartments is represented using box plots in Fig. 4.1. The results are classified based on the prediction horizons: T denotes predicting occupancy states one timestep ahead (30 minutes) as the shortest period, T+3 shows a prediction horizon of three timesteps, and T+7, which is the longest horizon, represents the prediction for 7 timesteps in advance. This classification aims to investigate the impact of the prediction horizon on the models' performance. As depicted in the figure, depending on the model, horizon, and case study, the accuracy can vary between 75% to more than 90%. With a closer look at the figure, it can be observed that for short-term predictions, the MLP outperformed other methods with the highest median accuracy at 87.45%. It was followed by the LSTM, RNN, and GRU models respectively at 86.81%, 86.18%, and 83.97% accuracy. However, for longer durations of prediction, the GRU model provides the best performance among the deep learning models. It results in a median accuracy of 82.81% and 80.59% for T+3 and T+7 horizons, respectively, while MLP leads to 81.77% and 79.85% accuracy, respectively for the same time intervals.

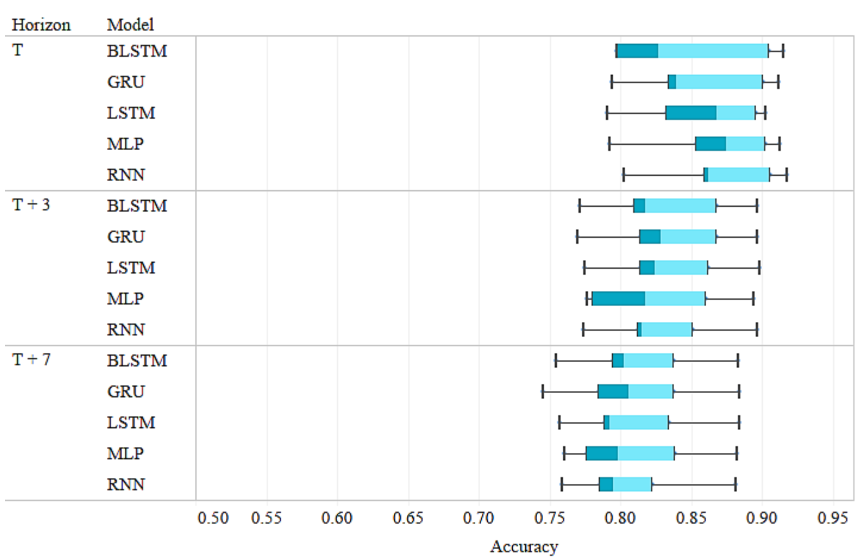


Fig. 4.1. The accuracy of the deep learning models based on prediction horizon.

It can be also observed that the accuracy of the models is highly dependent on the case study. In other words, using a specific model for a certain time horizon, the accuracy can change by more than 10% depending only on the apartment. As suggested by Sangogboye et al. [82], the reason is linked to the differences in the vacancy frequency (i.e. how frequently the apartments become unoccupied) in occupancy profiles in different apartments. To quantify the relationship between accuracy of the deep learning models and the actual occupancy profiles, an occupancy index, O_{index}, is defined in this study as follows:

$$O_{index} = \begin{cases} O_{rate} & O_{rate} \le 0.5 \\ 1 - O_{rate} & O_{rate} > 0.5 \end{cases} \tag{14}$$

where O_{rate} denotes the occupancy rate, defined as the sum of all occupied time slots divided by the number of total time intervals. The occupancy index can change in the range of 0-0.5. The value of **0** shows that the occupancy pattern of the household tends to be uniform. For example, the building might be in the occupied or vacant states all the time. The value of **0.5** shows that the occupancy states and vacancy states are equally exists at the occupancy pattern, that can cause more randomness and harder prediction. It is found that there is a direct relationship between the occupancy index and the accuracy of deep learning-based occupancy models with a Pearson correlation of 0.89. By taking advantage of this direct correlation, the accuracy of occupancy models developed in this study are approximated using an exponential relationship as a function of the occupancy index with a mean absolute error of 0.0141:

$$Accuracy = 0.7399 \exp(0.7767 O_{index}) \tag{15}$$

Fig. 4.2 represents the performance of the models in terms of computational time needed for training the models. It can be seen that the MLP model is the fastest model with a median training time of 11.76 seconds. It is followed by GRU which needs 4.8 seconds more time to be trained. In contrast, the slowest training is associated with the BLSTM at almost 40 seconds.

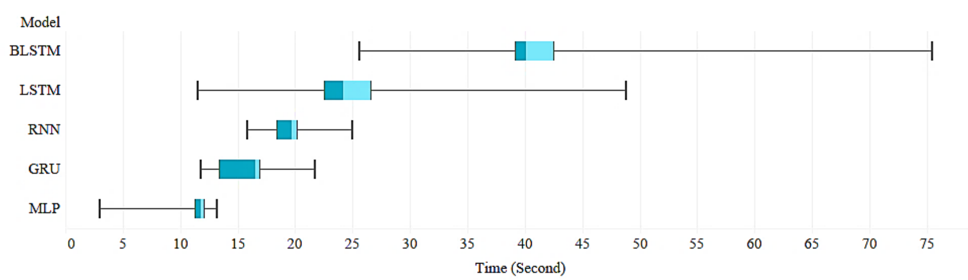


Fig. 4.2. Computational time required for training the deep learning models.

4.5. Discussion

It is found that the performance of the deep learning models depends on the prediction horizon. For short-term predictions (i.e., occupancy prediction for less than 90 minutes in advance), the MLP model is the preferred method as it provides the best median accuracy. In contrast, for longer-term prediction, using the GRU model as a more complex approach can improve the accuracy. The reason behind selecting different models for long-term and short-term predictions can be linked with the level of complexity in the predictions. For example, the use of MLP for short-term predictions might indicate that the temporal relationships between the data elements are not important in short-term predictions. In contrast, the selection of GRU might indicate the importance of the temporal patterns in the long-term occupancy prediction. It is revealed that there is also a strong correlation between the models' performance and the occupancy patterns in the apartments. To quantify the relationship, the Pearson correlation coefficient is estimated between the accuracy and the occupancy index proposed in this study. Then, the prediction accuracy is estimated using a regression model based on the occupancy index that can result in an MAE of less than 0.02. This equation can be helpful for the control system designers to get an estimation of the occupancy prediction for their specific applications. However, it should be noted that this study is performed based on five residential cases; more cases are required to generalize the results for a wide range of applications, which is recommended as future work.

4.6. Conclusions

In this work, several state-of-the-art deep learning algorithms are employed to develop occupancy prediction models. The results demonstrate that both time series and multi-label classification approaches can be utilized to develop occupancy prediction models with an accuracy of more than 80% in most cases. However, the performance considerably depends on the prediction horizon. For short-term predictions, MLP outperforms other algorithms while for longer predictions, GRU leads to the best accuracy. These two methods also showed the lowest computational time required for training the models. It is also revealed that the accuracy of the deep learning models is strongly correlated with an occupancy index proposed in this study, having a Pearson correlation of almost 90%. Utilizing this correlation, the accuracy of the models is estimated as a function of the occupancy index through a regression model with an MAE of 0.0141. It is demonstrated that the accuracy of the models is still limited to less than 85% in most cases. More research is needed to further improve the accuracy with a special focus on the input variables, which have been often neglected in previous studies.

Chapter 5: Impact of occupancy prediction models on building HVAC control system performance: Application of machine learning techniques[4]

5.1. Overview

This chapter analyzes the impact of using different types of occupancy models on the performance of occupancy-based heating, ventilation, and air conditioning (HVAC) control systems in residential buildings. The future occupancy prediction problem is approached from two different viewpoints: 1) as a regression task for arrival-time prediction and 2) as a classification task for occupancy-state prediction. Four machine learning techniques, namely decision trees (DT), k-nearest neighbors (kNN), multi-layer perceptron (MLP), and gated recurrent units (GRU), are implemented to predict future occupancy from each perspective. The performance of the occupancy models is evaluated in terms of mean absolute error (MAE), accuracy, and the ability to provide occupants' thermal comfort and to save energy. An overall performance score is proposed by making a trade-off between the energy and thermal comfort objectives using the technique for order of preference by similarity to the ideal solution method. The results demonstrate that selecting the optimal viewpoint of occupancy prediction has a higher impact on the control performance than selecting the machine learning techniques, with occupancy-state prediction models providing superior performance in most cases. It is also shown that the machine learning evaluation metrics (i.e., MAE and accuracy) provide a weak to moderate correlation with the overall performance score. Consequently, relying solely on the MAE and accuracy might fail to provide a reliable evaluation of the occupancy model performance for use in HVAC control systems.

5.2. Introduction

During the past decades, smart thermostats have been proposed and investigated as a promising solution for increasing building energy performance and enhancing occupants' thermal comfort and productivity. Smart thermostats have the ability to save energy by modifying the indoor air temperature based on the current occupancy states in buildings [13]. More specifically, the thermostat can automatically employ a setback temperature while the residents are not present in the building, which could help to save energy and reduce negative environmental impact. Given the fact that the residential sector is responsible for an average of 30% of the overall energy consumption worldwide [135], implementing smart thermostats can highly contribute to mitigating carbon dioxide (CO_2) emissions and decreasing fossil fuel consumption. Smart thermostats can go beyond merely inferring current occupancy states by implementing occupancy prediction models. In other words, they can give insight into the future changes in occupancy states and provide adequate time for the heating, ventilation, and air conditioning (HVAC) system to precondition indoor environment before the building becomes occupied [86]. However, starting

[4] This chapter is based on the following publication: M. Esrafilian-Najafabadi, F. Haghighat, Impact of occupancy prediction models on building HVAC control system performance: Application of machine learning techniques, Energy and Building 257 (2022) 111808. https://doi.org/https://doi.org/10.1016/j.enbuild.2021.111808.

the preconditioning process too soon or too late can respectively cause energy waste or occupants' thermal discomfort.

Thereby, many researchers focused on developing more accurate and feasible occupancy models for possible integrations with predictive control systems. To this end, data-driven methodologies, especially machine learning methods, have attracted great attention among many researchers [35,36]. Previous studies formulated the occupancy prediction problem using two different approaches. The majority of earlier works addressed the occupancy prediction problem as a classification task. In these approaches, occupancy datasets are often divided into equal time intervals, with each interval taking values of **0** or **1**, respectively showing vacancy or occupancy states for the corresponding period. The occupancy models developed based on this approach forecast occupancy states in a determined number of future time steps, called prediction horizon. The prediction horizon is often determined based on the lag time of HVAC systems, which can vary depending on the weather conditions and the characteristics of the buildings and HVAC systems [156]. The occupancy prediction problem was also formulated as a function of arrival and departure time. In this approach, instead of predicting occupancy states in separate time intervals, the occupancy model provides predictions on the future occupants' arrival or departure time. This type of prediction can be addressed as a regression task, in which future arrival or departure time is forecasted.

The variety of occupancy prediction models proposed in the literature indicates the importance of making attempts to find the most effective approaches. There is still a question of which occupancy prediction viewpoint and technique can provide the most reliable results among the many available alternatives. Furthermore, when comparing the performance of different models, most of the earlier works solely relied on machine learning and statistical evaluation metrics, such as accuracy, overlooking the ability of occupancy models to enhance energy saving and occupants' thermal comfort. There is a question of what evaluation metrics are the best to quantify the effectiveness of the occupancy prediction models. The following section provides an overview of the occupancy prediction model developments and the applications in HVAC control systems. Next, the current research gaps and the contributions of this research work are elaborated.

5.2.1. Related research works

The majority of earlier works addressed the occupancy prediction problem as a classification task. Sangogboye et al. [84] proposed an occupancy prediction model, named PROMPT, in which four different machine learning methods were implemented to predict future occupancy states. They assumed that there might not be a unique optimal machine learning method that can provide the best performance in all prediction cases. Therefore, the PROMPT algorithm tried to find the model that could provide the best performance in each specific case. They applied each machine learning algorithm to every case study, evaluated its performance, and selected the model with the highest prediction performance for each case. They showed that this integrated model increased the F-score of occupancy prediction by 2.3% compared with using a single model. Salimi et al. [91] developed an occupancy model using the inhomogeneous Markov model based on the identity, time of day, and zonal locations of the occupants in an office room. The model used two different time horizons of 30 minutes and 5 minutes for HVAC and lighting control systems, respectively. Based on the length of the prediction horizon, they reported an accuracy in the range of 68% to 86% for predicting future occupancy states.

Koehler et al. [44] proposed an occupancy prediction model based on the occupants' locations and status. If an occupant was in the driving status, the model predicted the following possible destinations using previously visited locations. Otherwise, the most frequent occupancy states in the historical data were considered in a proportional model to predict future occupancy states. This hybrid model resulted in 92.1% accuracy for occupancy prediction. Some researchers took advantage of environmental features to develop occupancy prediction models. Ryu et al. [96] applied a DT model to CO_2 concentration and lighting level data to estimate the occupancy levels (i.e., the number of occupants) in an office building. The estimated occupancy information was used to train a hidden Markov model (HMM) to predict occupancy states in future time steps. The model yielded an accuracy of 90%. Elkhoukhi et al. [94] proposed an occupancy model by predicting future CO_2 concentration in an office building. In the first step, the algorithm was trained using historical CO_2 data to provide short-term predictions for CO_2 levels. In the next step, the predicted level of CO_2 was translated into occupant counts based on a simple steady-state room CO_2 mass balance [100]. The accuracy of the model was estimated as 70%.

Some researchers implemented the occupancy-state prediction in HVAC control systems and measured the building performance. It was shown that using Markov models for occupancy prediction in a control system could save up to 42% annual energy, compared with using an always-on control system [64,66,157]. Iyengar et al. [46] investigated the possibility of making traditional programmable thermostats occupancy aware in residential buildings without any need for providing more infrastructure or capital costs. To this end, they proposed a model using household energy consumption patterns that helped to detect and predict future occupancy. This proposed thermostat showed a potential saving of 0.42 kWh energy per day on average, taking standard programmable thermostats as the baseline. Gluck et al. [17] studied the effects of the occupancy prediction errors on the control system performance. First, they demonstrated that a predictive occupancy-based HVAC control could result in up to 40% energy saving compared with reactive thermostats. They also reported that a 10% reduction in the prediction error could enhance the energy saving by 9% and the thermal comfort by 16%. Killian and Kozek [72] used a k-means clustering algorithm, as an unsupervised machine learning method, for occupancy prediction in a model predictive control (MPC) to regulate the indoor air temperature. This system showed a potential to improve the occupants' thermal comfort by decreasing the temperature deviation from the setpoint by almost 32%, as compared with an MPC without occupancy prediction models. Peng et al. [61] employed two machine learning techniques, namely k-means clustering and kNN models, to predict future occupancy patterns. They implemented the clustering model to categorize similar data elements, as a preprocessing step. Then, a KNN model was trained on this preprocessed database to predict future occupancy states in an office building. They reported that up to 52% energy saving was achieved by using the proposed algorithm in the control system, when compared with conventional scheduled thermostats. Scott et al. [45] established an occupancy-based control system called PreHeat and experimentally evaluated its performance by applying it to five different homes. This system predicted future occupancy states using a kNN algorithm. It was reported that PreHeat was able to improve occupants' thermal comfort by more than 90% while consuming almost the same amount of energy, taking the standard programmable thermostats as the baseline,

There are also approaches that consider occupancy prediction as a regression task, in which the arrival and departure time of occupants rather than occupancy states are predicted. Gjoresk et al. [83] developed occupancy prediction models based on arrival and departure data collected from seven employees with flexible working hours in an office building. They demonstrated that the mean absolute error (MAE) of arrival-time prediction models varied between 10 min to 60 min depending on the employees. However, occupancy models provided superior performance for departure-time prediction with an MAE of 40 min to 80 min. Lee et al. [63] developed a model to predict occupants' arrival time in an office building utilizing the current and historical locations of occupants via the occupants' cellphones. They reported that this control system forecasted future arrival time with approximately a 10-min error in 70% of the cases. The implementation of the models in a control system led to a 26% decrease in energy consumption when compared with the performance of scheduled thermostats. Gao and Whitehouse [76] developed a control system, named self-programming thermostats, based on one-month daily arrival and departure time. They reported that the proposed thermostat was able to improve occupants' thermal comfort by up to 40% and decrease the energy demand by 15%, in comparison with the performance of conventional programmable thermostat.

The variety of the proposed occupancy prediction models highlights the necessity of investigating their performance and finding the limitations and strengths of each model. To this end, occupancy models need to be applied to the same database to provide a reliable comparison between different alternatives. Huchuk et al. [95] applied machine learning models to an occupancy database collected from 100 homes to find the benefits and limitations of different occupancy models. They implemented kNN, recurrent neural network (RNN), random forest (RF), Markov models, HMM, and logistic regression models for predicting future occupancy states. Among the alternatives, RF was the best-performing model in terms of accuracy. Kleiminger et al. [86] compared the performance of different occupancy models, including the PreHeat occupancy prediction model [45] and a probabilistic proportional model [87], in terms of accuracy. They reported that the proportional model provided the highest accuracy at 85%. Caleb et al. [82] applied support vector machine (SVM), RF, DT, and kNN to an occupancy database. They employed various calendar features, including seasons and holidays, in the occupancy model development and demonstrated that the selection of the best models was linked with the frequency of vacancy/occupancy state changes. The SVM provided the highest performance for the higher frequencies, while all the models yielded a similar performance for less frequent changes between different states.

5.2.2. Research gaps and contributions

Earlier studies have formulated the occupancy prediction problem from two different perspectives; some researchers implemented classification methods to forecast occupancy states at future time intervals, while others addressed this problem as a departure or arrival time prediction. However, the benefits and drawbacks of different viewpoints towards occupancy prediction modeling have not been thoroughly investigated. There is a need to clarify which approach is more appropriate for application in occupancy-based HVAC control systems. Analyzing the strengths and limitations of each approach can be helpful in selecting the most effective occupancy prediction viewpoint. Although there are a few studies providing a comparison framework between occupancy prediction models, they were limited for evaluating the techniques from the same viewpoint. Additionally, these studies evaluated the performance of occupancy models based on machine learning evaluation metrics such as accuracy and MAE, neglecting more practical indicators such as occupants' thermal comfort and energy saving. It is worth noting that, as reported in [73], the negative impact of the prediction errors on the control performance might depend on the timing of the errors. For instance, a mistake in the occupancy prediction near arrival time of occupants after a relatively long vacancy period can have detrimental effects on occupants' thermal comfort. In contrast, the impact of some inaccuracies during short vacancy periods can be negligible. However, the accuracy, as the most implemented indicator [114], cannot consider the timing nature of the errors. Hence, assessing an occupancy model while relying solely on accuracy or MAE might be deceptive in some cases [46].

The primary objective of this chapter is to evaluate different viewpoints for occupancy prediction modeling and to reveal their benefits and limitations when applied to occupancy-based HVAC control. This study also aims to quantify the correlation between the commonly utilized statistical performance metrics (accuracy and MAE) and the building performance criteria (thermal comfort and energy efficiency) to find out whether they can interchangeably be utilized in evaluating and comparing occupancy models. In order to achieve the research objectives, this chapter analyzes the performance of different types of occupancy models when applied to a predictive rule-based (RB) control framework. Occupancy modeling is addressed from both viewpoints of occupancy-state prediction as a classification task and arrival-time prediction as a regression task. Additionally, the performance of different machine learning techniques is assessed by considering DT, kNN, MLP, and GRU techniques. As well as using accuracy and MAE, the performance of the occupancy models is measured and compared in terms of their ability to improve occupants' thermal comfort and energy-efficiency of the HVAC operation. As maximizing thermal comfort and energy saving are often conflicting objectives, an overall performance score is proposed to make a trade-off between them based on the technique for order of preference by similarity to ideal solution (TOPSIS) method. Using the Pearson correlation coefficients, the correlation between the machine learning evaluation metrics and the performance of the control system is quantified.

The remaining parts of the chapter are structured as follows: Section 5.3 elaborates the methodology utilized in this study to construct the occupancy models and to evaluate their performance. This section describes the machine learning models, the database preprocessing step, the control system, the performance criteria, and the case study utilized to evaluate the models. The results are described in Section 0, and finally, the conclusions are made and discussed.

5.3. Methodology

5.3.1. Development of the occupancy prediction models

This section investigates the occupancy prediction problem from two different perspectives of occupancy-state prediction and arrival-time prediction. While considering different perspectives, the impact of the machine learning techniques on the performance of the control system is also assessed by employing different models. kNN and DT techniques, as straightforward and widely utilized algorithms, are implemented in this study. These algorithms were among the most used machine learning techniques for occupancy prediction [114]. In addition, two deep learning algorithms are also employed to model occupancy behavior. Firstly, the MLP model is deployed to explore hidden patterns in the occupancy database. However, MLP models do not consider the temporal relationships between sequential data elements. As occupancy prediction is a time-series prediction problem, consideration of temporal relationships might improve the prediction performance. Hence, the GRU model is also developed as a deep learning algorithm, which is able to capture temporal relationships between different elements of data. However, whether these models can practically result in a better prediction performance than conventional machine learning models has not been widely investigated for occupancy prediction algorithms.

5.3.1.1. k-nearest neighbors model

The kNN algorithm is capable of tackling both classification and regression problems and is considered one of the simplest machine learning algorithms [158]. kNN is part of lazy learning algorithms, in which the training process is not initiated unless there is a need to respond to a query [159]. The algorithm keeps the training set in memory until the query has been answered [160]. By finding the similarities between an unseen input data element and the samples in the training set, the algorithm can fulfill the prediction task. The similarities are often measured using distance metrics. Euclidean distance, as one of the most popular distance metrics, is employed for this purpose in this study. A predefined number of the nearest neighbors (k) to the new input data is determined and utilized to estimate the label of the new element. In this study, k is considered as a hyperparameter and is determined using a grid search method. Based on the optimal value of k, the algorithm assigns the mode and mean of the closest neighbors' labels to the input sample for classification and regression problems, respectively. In the classification task, the aim is to predict the occupancy states in a desired number of the following consecutive time steps (N). To this end, N independent kNN models are developed for predicting each state at the future time steps. On the other hand, only one model is developed for the regression problem with one label that is the arrival time of the occupants.

5.3.1.2. Decision Tree models

Similar to kNN algorithms, DT can also be employed for both classification and regression problems. The benefits of DT algorithms are their simplicity, high interpretability, and the ability to be applied to mixed-type data [161–164]. DT categorizes the data into different branches, creating a tree-like structure with a root and internal and leaf nodes [163]. The creation of DT is an iterative process. In each iteration, the algorithm aims to find the best feature that can split the entire dataset into multiple groups. For this purpose, different features are tried, and the one that can minimize a cost function, called attribute selection criterion (AST), is chosen [165]. In this study, mean squared error (MSE) and the entropy function are considered as the AST for the regression and classification tasks, respectively. The first selected feature splits the data and creates the first node in the tree, called the root of the tree. A similar iterative splitting procedure is performed to divide the rest of the dataset into different branches and additional internal nodes.

The splitting procedure should be stopped in a way that maximizes the performance criteria while minimizing the number of splits to avoid overfitting the training set. To properly define the stopping criteria, maximum depth of the tree, minimum samples required for creating a leaf and a split, and the maximum number of features are considered as hyperparameters to be tuned using a grid search method. Similar to the kNN, N independent trees are trained for occupancy prediction at each future time step, and one tree for the arrival-time prediction. The kNN and DT models are developed using the Scikit-learn package [166] in Python.

5.3.1.3. Multi-layer perceptron

MLP is a type of feed-forward artificial neural network (ANN), which was originally developed based on the operation of the human brain [167]. The network involves at least three layers of neurons, namely input, hidden, and output layers. Each neuron, except for those in the input layer, employs an activation function, allowing the network to learn non-linear relationships between the predictor variables and the labels. In this study, rectified linear unit (ReLU) is implemented as the activation function for neurons in the hidden layers as a recommended choice for developing neural networks [168]. Furthermore, linear and sigmoid functions are respectively utilized for the output layers in the regression and classification models. Backpropagation, as a supervised learning technique, is used to determine the optimal weights and bias values for each neuron during the learning process. An Adam optimizer is selected to minimize MSE and binary cross-entropy, for arrival time and occupancy-state prediction models, respectively. Table 5.1 provides a summary of the structural parameters of the MLP model, including the batch size, learning rate, number of neurons, and number of hidden layers, which are determined as hyperparameters.

Sample MLP for both regression and classification tasks are depicted in Fig. 5.1. These networks are similar with a distinction in the output layer. In the occupancy-state prediction, the problem is addressed as a multi-label classification approach, in which *N* future occupancy states are associated with each data element as the data labels. Therefore, the output layer of MLP has N neurons to provide predictions for each future time step. In contrast, only one output, i.e. arrival time, is predicted when approaching the problem as a regression task, and therefore, only one output neuron is utilized. The input layer includes $M+3$ attributes, in which M denotes the number of lagged occupancy states. The input features involve the occupancy states in the previous M time steps (from O_{t-M} to O_{t-1}), the current occupancy state (O_t), time of day (hour), and day of week.

Table 5.1.

The summary of the parameters utilized for developing the MLP and GRU models.

Parameter	Description
Number of neurons	Defined as a hyperparameter
Number of layers	Defined as a hyperparameter
Batch size	Defined as a hyperparameter
Learning rate	Defined as a hyperparameter
Activation function for the neurons in hidden layers	ReLU
Activation function in the output layer	Linear function for arrival time prediction and Sigmoid function for occupancy state prediction
Optimizer	Adam
Optimization objectives	To minimize the binary cross entropy for the classification models and to minimize the MSE for the regression models
Regularization	Weight decay and early stopping
Lambda	Defined as a hyperparameter

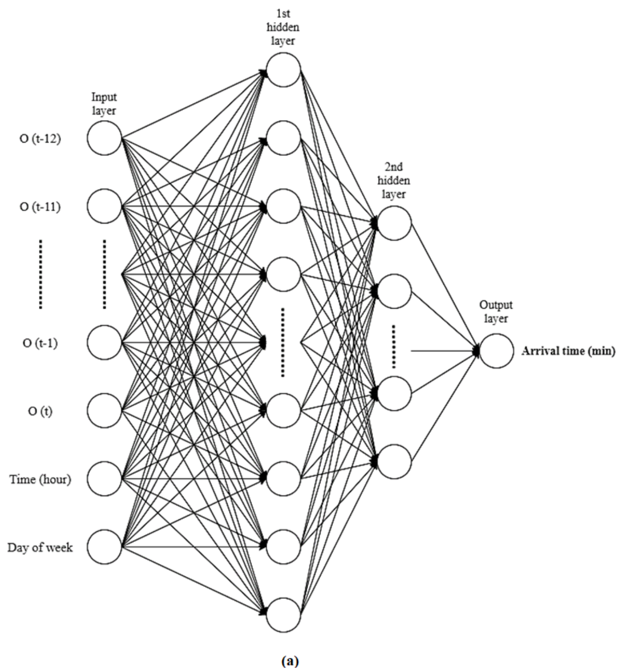


Fig. 5.1. Sample MLP model used for a) arrival time and b) occupancy-state prediction with two hidden layers (N=12 and M=12).

One of the main challenges in implementing deep learning algorithms is to develop a model with a high generalization capability when being applied to unseen data. While too simple models might not be able to properly capture hidden patterns in the training set, too complex models are prone to the overfitting problem and can provide worse performance on the test set due to their poor generalization ability [169]. In order to prevent the overfitting issue while training the models, the early stopping and weight decay methods are implemented. Early stopping is one of the most utilized regularization approaches, recognized as a straightforward and successful technique [168]. In this method, the performance of the machine learning algorithms is monitored during the training process for both train and test datasets. The training process is halted before the test performance starts to decrease due to the overfitting issue. The *patience* value (i.e., the number of epochs with no improvements in the test accuracy) of the early stopping method is considered 50 epochs. In the weight decay technique, the algorithm encourages the weights to reach zero unless reinforced by the data [170]. In other words, this method avoids using large weights in the neural networks, which is one of the reasons for the overfitting, by penalizing the sum of the squared values of weights in the optimization problem. The amount of lambda (i.e. regularization rate) is selected as a hyperparameter and determined in a random search process.

5.3.1.4. Gated recurrent units

GRU was originally proposed to address the difficulties that might occur in training RNN models. RNN is theoretically a powerful tool for modeling sequential data; However, in practice, they might encounter a severe issue named *vanishing gradient* [171]. *Vanishing gradient* is referred to as the tendency of the loss-function gradients to reach zero during the backpropagation. This issue makes it too hard for a network to carry information alongside a long sequence in the network. As a consequence, some vital information might be lost during the training process. The GRU algorithm showed the ability to address this issue by using some mechanisms, called *update* and *reset* gates, which enable the layers to protect and carry important data alongside the network [172].

In this study, two different architectures of GRU, namely many-to-one and many-to-many architectures, are respectively implemented for the arrival time and occupancy-state predictions. The former architecture is demonstrated in Fig. 5.2. In contrast to the MLP, which does not differentiate between data sequences, the input features, namely time of day (T), day of week (D), and occupancy states (O), enter the GRU model in M+3 separate time sequences. Then, the information is consecutively carried forward from the first time step to the current one and finally yields a prediction for the arrival time. On the other hand, the occupancy-state model requires a more complex network due to having more than one output (i.e., a sequence of outputs) [173]. This problem is addressed using a different architecture, which is called an encoder-decoder network, shown in Fig. 5.3. As the name suggests, this network involves two different blocks, namely encoder and decoder parts of the model. In the former part, the input data is processed in the GRU units and, finally, is encoded in a vector named encoder vector. This vector is passed into the first GRU unit in the decoder part as the input. The decoder block employs N cells to provide separate predictions for the future occupancy states. The predicted probabilities are then compared with the actual values, and accordingly, the loss function is calculated and minimized using the backpropagation. The structural parameters and the regularization methods related to the GRU network are similar to those of the MLP and are summarized in Table 5.1. The deep learning algorithms are developed using Keras library [125] in the Python environment.

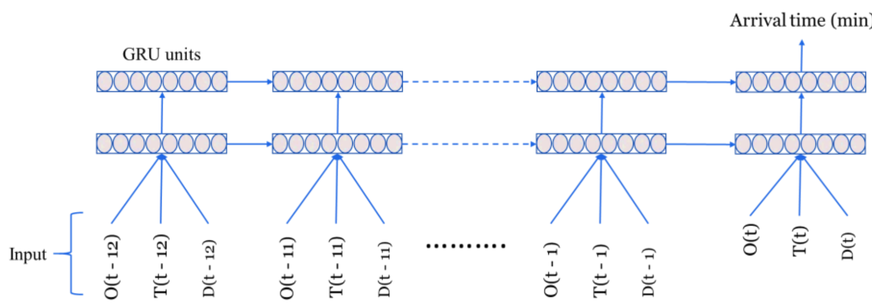


Fig. 5.2. Many-to-one architecture of the GRU model for arival time prediction (M=12).

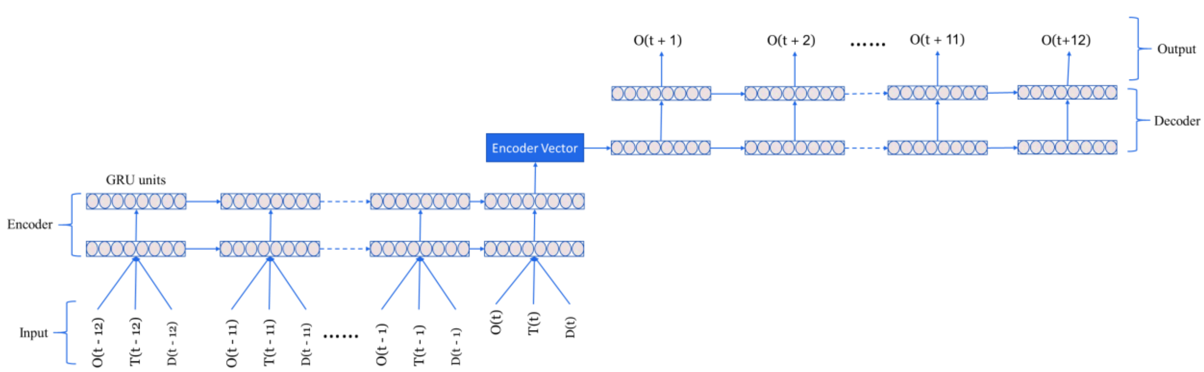


Fig. 5.3. Many-to-many architecture of the GRU method for occupancy-state prediction (N=12 and M=12)

5.3.2. Occupancy database description and preprocessing

The models are trained using an occupancy database collected from five apartments [29,120,156,174,175]. The database was collected for the year 2016 with a resolution of 1 minute using 13 passive infrared (PIR) motion detectors installed in different zones of each apartment. The whole database was provided from a single data source, and as a result, there is no need to aggregate separated datasets in this study. As part of the data preprocessing step, first, the database is resampled from 1-min to 30-min intervals. In other words, each day is divided into 48 elements with equal durations. Each element is assigned with **0** or **1**, denoting that no motion or at least one motion was detected at the corresponding time interval, respectively. As for the classification task, the occupancy models predict the occupancy states in 12 successive future time steps (N=12). Therefore, each row in the database is associated with 12 different labels, which show the future occupancy states in the next 6 hours. Thereby, this task is addressed as a multi-label classification problem, in which multiple mutually non-exclusive labels are associated with each instance [176]. In contrast, in the regression task, each element in the database has only one label, which is the occupants' arrival time. The arrival time is calculated as the duration between the current time step (i.e., at the end of each 30-min time interval) and the closest point of time when the first motion is recorded from any sensors based on the original resolution of the database (i.e., 1-min interval). The arrival time is associated with each data instance at each 30-min interval as the label.

As demonstrated in [114], time of day and day of week were the most utilized features to develop occupancy models in the literature and, as a result, are utilized in the development of the occupancy models. The time-of-day attribute can accept integers from 0 to 23, representing each hour of the day. This categorical variable can be used in training the DT models without any further changes, as the DT is able to directly deal with categorical variables. In contrast, to train other models, this variable is transformed into 24 separate attributes using the one-hot encoding method. The new attributes take **1** if the data element was recorded at the corresponding hour of day and otherwise take **0**. The day-of-week feature is also preprocessed similarly with a distinction that it initially takes integers from **0** to **6**, representing days of a week starting from Monday. It is worth noting that as the occupancy prediction problem is a time series analysis, in which there are meaningful ordered time sequences, the occupancy states in the previous time steps can also be helpful in predicting future states. Hence, 12 consecutive occupancy states in the previous 6 hours, as well as the current occupancy state, are also used as attributes in developing the occupancy models.

Each dataset is divided into two subsets of validation and testing. The validation set is limited between August 1st and the end of October, which is utilized for tuning the hyperparameters of the models. It is worth noting that the temporal order of the data elements should be respected while constructing the datasets since the prediction problem is a time series analysis. After finding the best hyperparameters, the ultimate occupancy models are trained on the whole validation set, and their performance is assessed on the unseen test set. The test set consists of data starting from November 1st to the end of December. This period is also utilized to evaluate the control performance. The entire data preprocessing stages are performed using MySQL [121] and Pandas library [177] in Python.

5.3.3. Control system and performance evaluation criteria

5.3.3.1. Case study description

In order to evaluate the performance of each occupancy model for use in the HVAC control, a residential building is simulated as a virtual testbed. A single-story house, also implemented in [156], is taken as the case study with the floor plan shown in Fig. 5.4. The building has a net conditioned area of 135 m^2, including three bedrooms, a kitchen, a living room, two bathrooms, and corridors. The geometry of the building is modeled in the SketchUp software [130] and then imported into EnergyPlus [178] using Openstudio [131] for energy simulation. The weather data available in [129] for ASHRAE climate zone 6A, which is a cold humid climate [179–181], is employed as the input to the energy model. The building construction material is assumed based on the ASHRAE recommendations for the same climate zone, available as a library in Openstudio. The simulation is run during the same period as the occupancy test set (i.e., from November 1st to December 31st), during the winter with only the heating demand. It is assumed that the heating demand is fulfilled using electric baseboards, and consequently, electricity is the source of the heating energy provision.

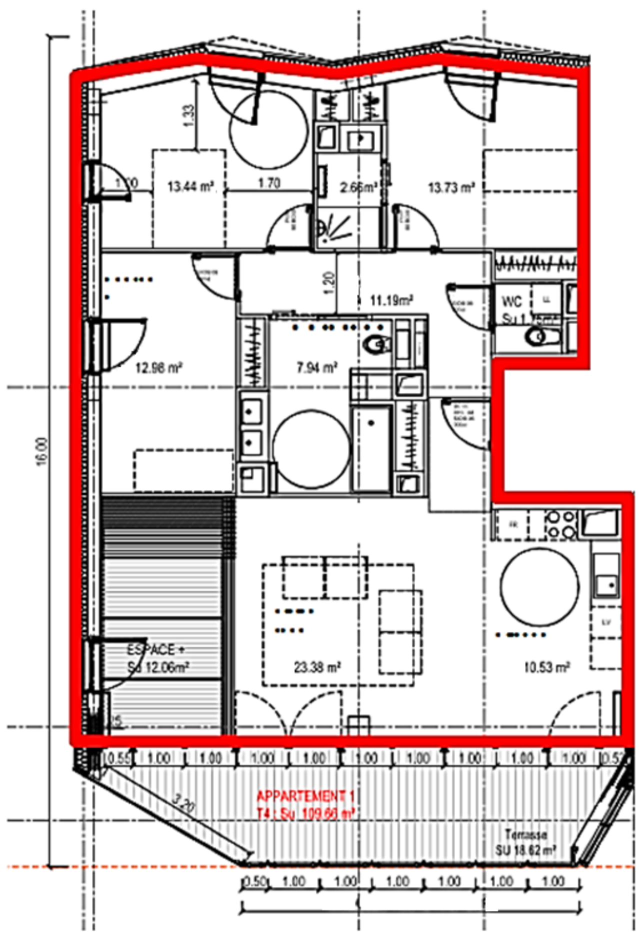


Fig. 5.4. The floor plan of the virtual test case.

5.3.3.2. Control system description

One of the main applications of developing the occupancy models is to be implemented in predictive control systems for adjusting the setpoint temperature according to the occupancy states. As a result, it is essential to evaluate the effectiveness of the occupancy models in a predictive control context. To this end, the occupancy model performance is evaluated in the RB control framework proposed in [156] because of its superior performance over conventional controllers. This control system takes advantage of an MLP model to dynamically estimate the preheating time. The database required for training the MLP model is created through building energy simulation using the EnergyPlus software. The created database provides information about how long it takes to bring back the setpoint temperature as a function of solar radiation, current indoor temperature, and outdoor temperature. In the simulation procedure, the indoor setpoint temperature is defined according to a programmable thermostat with static setpoint/setback schedules. It is noted that the building material and local weather conditions need to be reflected in the database since they can significantly impact the lag time of the HVAC systems. As these parameters can substantially vary among different buildings, the database is created based on the energy simulation of the case study building, described in Section 5.3.3.1, to enable the MLP model to learn the characteristics of the building. More details about creating the lag time database can be found in [156].

The functional diagram of the control system is demonstrated in Fig. 5.5. In this framework, the lag time of the HVAC system is estimated by training the MLP model using the database. Next, the occupancy-state prediction models take the approximated preheating time as their prediction horizon. For instance, given a preheating time of 70 minutes, the occupancy model yields **1** if the building is predicted to be occupied at one of the next three-time steps (given that the length of each time step equals 30 minutes), and otherwise, the model gives **0** as the output. Accordingly, the control system defines a setpoint or a setback temperature if the occupancy model returns **1** or **0**, respectively. In terms of using the arrival time prediction models, the preheating time is directly compared with the estimated arrival time of occupants. A longer preheating time than the arrival time indicates that occupants will probably arrive before the building has been fully preheated, and therefore, the preheating process is initiated immediately. Otherwise, there would be enough time for keeping a setback temperature to save more energy while not causing thermal discomfort.

The setpoint operative temperature is defined as 21.5 °C for all the systems, which is within the recommended thermal comfort range [182]. In practical cases, a conservative setback temperature is often selected to ensure that the thermal comfort is maintained even when there are occupancy prediction errors. Nevertheless, as one of the goals of this study is to investigate the impact of such occupancy prediction errors on the control system performance, the temperature constraints are further relaxed, and a setback of 15 °C is assumed. This relatively low setback temperature is selected to ensure that the benefits and drawbacks of different occupancy models are clearly observed.

It is noted that there might be some periods with immobile occupants, which can be inferred as vacant intervals due to detection errors of conventional sensors. Consequently, such events can bring about wrong temperature settings and, as a consequence, thermal discomfort. In order to minimize the negative impact of such events, the control system always maintains setpoint temperature during the first unoccupied time interval after an occupancy period regardless of the prediction results.

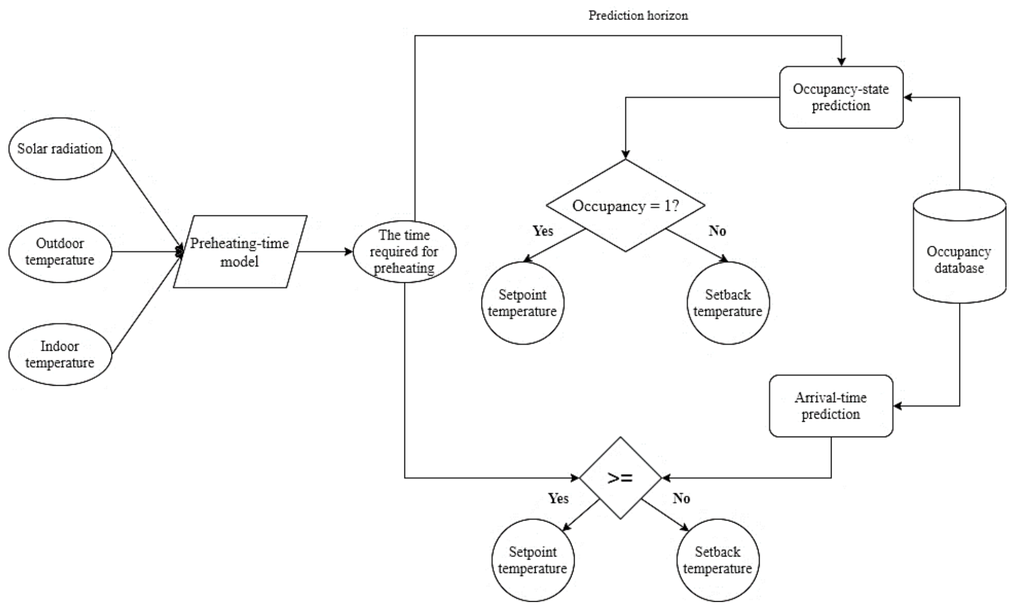


Fig. 5.5. The RB control system using occupancy models for regulating setpoint/setback temperature.

5.3.3.3. Evaluation criteria

The performance of the occupancy models can be measured using the accuracy and MAE, as conventional statistical metrics, for occupancy-state and arrival time prediction models, respectively. Furthermore, the performance of the models is evaluated in the control context in terms of the energy efficiency and thermal comfort criteria, as the most popular indicators utilized for occupancy-based HVAC control systems in the literature [114]. Energy efficiency is calculated as the amount of energy saving achieved by using the occupancy-based control, compared with using an always-on control system (i.e., the control system that always maintains the indoor temperature at 21.5 °C regardless of occupancy states) as the baseline. *MissTime* is selected as a criterion to quantify the occupants' thermal discomfort. This factor is defined as the amount of time when occupants are present, but the temperature deviates from the desired setpoint [45].

5.3.3.4. Overall performance score

Due to the conflicting nature of the thermal comfort and energy saving criteria, it is not straightforward to select the best-performing model among different candidates. In other words, some models, although provide the highest energy-efficiency performance, might not lead to satisfying occupants' thermal comfort. Multi-criteria decision-making (MCDM) is a powerful tool for choosing the best system amongst different alternatives in such complex problems [183].

Among possible MCDM methods, TOPSIS is selected because of its simplicity, understandable process, speed, and the ability to find the ideal and worst solutions [184,185]. In this method, two ideal solutions called positive ideal solution (PIS) and negative ideal solution (NIS) are determined as the ideal best and worst performance points, respectively. PIS takes the lowest *MissTime* and the largest energy saving among all the alternatives, and the NIS is the opposite point with the highest *MissTime* and the lowest energy saving. The system with the shortest Euclidean distance from the PIS and the longest distance from the NIS is selected as the optimum. In order to rank the alternatives, a score, called relative closeness (R), is associated with each system that can vary in the range of **0-1**. This parameter is defined as the distance between the alternative and the NIS point divided by the sum of the distances from the alternative to the PIS and to the NIS points. The closer the indicator is to **1**, the higher it ranks among the alternatives [144]. Based on the definition, the relative closeness includes both *MissTime* and occupants' thermal comfort while evaluating the performance of the occupancy models, and as a result, it is considered as an overall performance score for each occupancy model.

It is worth noting that in the TOPSIS method, the decision-maker can put higher priority to one criterion by assigning a larger weight to it. In some cases, the thermal comfort of occupants needs to be strictly maintained at a high level, while in other cases, occupants might be willing to save energy and cost with higher flexibility in defining the thermal comfort criterion. Therefore, depending on occupants' preferences and applications, different weights can be utilized. The criterion with the higher weight can contribute more to the relative closeness. In order to show the effectiveness of the TOPSIS method in this decision-making process, a case is assumed in which the energy and thermal comfort criteria are equally important. However, it is acknowledged that different weights can also be selected, which could provide different decision-making results. There are also mathematical approaches, such as the Entropy method, that can be utilized to determine the corresponding weights. Such methods aim to minimize the possible user-dependent biases in determining the weights in the TOPSIS method. More details about such techniques can be found in [186,187].

5.3.4. Correlation between the performance indicators

As mentioned previously, the statistical indicators were widely utilized in the literature to evaluate and compare the performance of the occupancy models regardless of the control system operation. A direct relationship between these indicators and the control system performance was assumed in earlier studies [95]. In order to evaluate the soundness of this assumption, the linear correlation between the metrics is quantified by employing the Pearson correlation coefficient, *r(x,y)*. Since the overall performance score includes both thermal comfort and energy saving, it is utilized as the control performance indicator, and its correlations with the accuracy, MAE, and their normalized values are calculated. The Pearson correlation coefficient is defined [188]:

$$r(x,y) = \frac{(n\sum x_i y_i - \sum x_i \sum y_i)}{\left[\sqrt{n\sum x_i^2 - (\sum x_i)^2}\right]\left[\sqrt{n\sum y_i^2 - (\sum y_i)^2}\right]} \quad \begin{cases} x = accuracy, MAE, or\ the\ normalized\ values \\ y = Overall\ performance\ score \end{cases} \tag{16}$$

where x_i and y_i denote the elements of the variables, and n represents the number of pairs of elements. The normalized MAE and accuracy are calculated based on the following equation:

$$\text{ormalized values} = \begin{cases} \dfrac{x - x_{minimum}}{x_{maximum} - x_{minimum}}, & x = accuracy \\ \dfrac{x_{maximum} - x}{x_{maximum} - x_{minimum}}, & x = MAE \end{cases} \tag{17}$$

5.4. Results and discussion

5.4.1. Development of the occupancy models

5.4.1.1. Optimal hyperparameters

Before the final occupancy models can be applied to the test dataset, the optimal hyperparameters need to be determined for each model. For the DT and kNN models, which are relatively computationally fast, a grid search technique is used. This method exhaustively searches among all the possible combinations of the hyperparameters in a defined range. In contrast, a random search method is utilized for the MLP and GRU techniques as they require a relatively long time for training. Instead of trying all the possible values, this method randomly goes through a limited number of iterations.

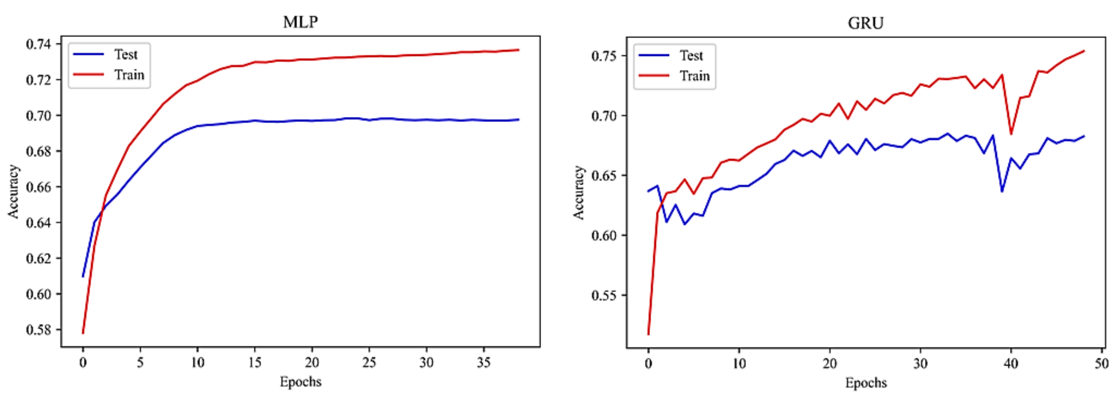
The search ranges and the averages and standard deviations of the optimal hyperparameters for each model are summarized in Table 5.2. Regarding the DT algorithm, the values for the maximum depth in both classification and regression tasks are close, with an average of six. Furthermore, the maximum number of features are similar, with average values of 10 and 11 respectively for the regression and classification. On the other hand, the minimum samples required for creating splits and leaves are much larger in the classification problem, with an average of almost 96. In terms of the kNN algorithm, the optimal number of neighbors for training a classification model is nearly three times higher than that of the regression models. Deep learning techniques require a relatively large number of hyperparameters to be tuned. The optimal MLP structures were developed using three and two hidden layers on average for the arrival-time and occupancy-state prediction, respectively. In the GRU model, an average of two hidden layers are selected for the encoder and decoder blocks. In both MLP and GRU, the arrival-time prediction is performed with a higher learning rate in comparison with the occupancy-state prediction models.

Table 5.2.
The mean, standard deviation, and ranges of hyperparameters utilized for developing the occupancy models.

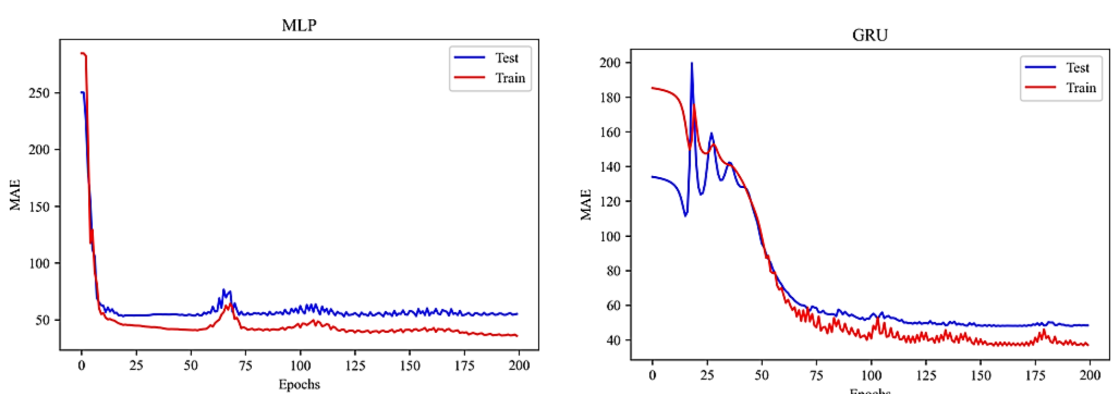
Model	Hyperparameter	Value		
		Regression	Classification	Range
DT	Maximum depth	6 ± 3	6 ± 2	[2 – 10]
	Minimum samples for split	50 ± 52	96 ± 52	[10 – 200]
	Minimum samples for leaves	16 ± 6	71 ± 59	[10 – 200]
	Maximum features	10 ± 1	11 ± 2	15
kNN	Number of neighbors	4 ± 3	15 ± 4	[2 – 20]
MLP	Number of hidden layers	3 ± 1	2 ± 1	[1, 2, 3, 4]
	Neurons in the 1st layer	60 ± 99	34 ± 26	[2, 6, 16, 32, 64, 128, 256]
	Neurons in the 2nd layer	132 ± 110	112 ± 91	[2, 6, 16, 32, 64, 128, 256]
	Neurons in the 3rd layer	49 ± 52	144 ± 112	[2, 6, 16, 32, 64, 128, 256]
	Neurons in the 4th layer	144 ± 112	-	[2, 6, 16, 32, 64, 128, 256]
	Batch size	920 ± 412	1750 ± 1274	[200, 400, 600, …, training set size]
	Learning rate	0.1 ± 0	0.008 ± 0.002	[0.005, 0.01, 0.05, 0.1]
	Lambda	4.004e-04 ± 4.9e-03	2.01e-05 ± 3.21e-04	[10^{-1} – 10^{-7}]
GRU	Number of encoder layers	-	2 ± 1	[1, 2, 3]
	Number of decoder layers	-	2 ± 0	[1, 2, 3]
	Number of layers	2 ± 1	-	[1, 2, 3, 4]
	Neurons in the 1st layer	25 ± 22	75 ± 93	[2, 6, 16, 32, 64, 128, 256]
	Neurons in the 2nd layer	19 ± 13	100 ± 102	[2, 6, 16, 32, 64, 128, 256]

Neurons in the 3rd layer	19 ± 13	32 ± 48	[2, 6, 16, 32, 64, 128, 256]
Neurons in the 4th layer	64	72 ± 95	[2, 6, 16, 32, 64, 128, 256]
Neurons in the 5th layer	-	65 ± 51	[2, 6, 16, 32, 64, 128, 256]
Neurons in the 6th layer	-	-	[2, 6, 16, 32, 64, 128, 256]
Batch size	760 ± 463	1800 ± 1307	[200, 400, 600, …, training set size]
Learning rate	0.025 ± 0.02	0.0082 ± 0.0036	[0.005, 0.01, 0.05, 0.1]
Lambda	2.22e-05 ± 3.91e-05	1.3e-05 ± 4.2e-05	[10^{-1} – 10^{-7}]

By employing the determined hyperparameters, the final occupancy prediction models can be trained, and their performance is evaluated on the test dataset. Fig. 5.6 demonstrates the performance of sample deep learning models during the training process. Monitoring the performance can provide an opportunity to avoid the overfitting problem. As discussed in [169], during the training process, the models' performance are improved until an optimal point and, then, starts moving away, providing a poor performance as the training process continues due to the overfitting issue. It can be observed that at the end of the training process for the classification models, the performance on the test set remains almost unchanged while the training performance still improves. As it can be a sign of the onset of overfitting, the training process is halted by the early stopping algorithm described in Section 5.3.1.3.



(a)



(b)

Fig. 5.6. The performance of deep learning models during the training process for (a) classificatoin and (b) regression tasks.

5.4.1.2. Performance of the machine learning techniques

The accuracy and MAE distributions of the occupancy prediction models are demonstrated using box plots in Fig. 5.7 and Fig. 5.8. The kNN model provides the highest median accuracy at 75.92%, which is closely followed by the DT and GRU models, leading to a median accuracy of 75.25% and 75.08%, respectively. The use of the MLP for occupancy-state prediction results in the poorest performance with a median accuracy of 73.33%. In contrast, the MLP yields the best median MAE at 58.11 minutes when the arrival-time prediction models are concerned. It is followed by the GRU and kNN models at a median MAE of 59.30 minutes and 61.51 minutes, respectively. The highest median MAE minutes is obtained at 67.15 by implementing the DT algorithm.

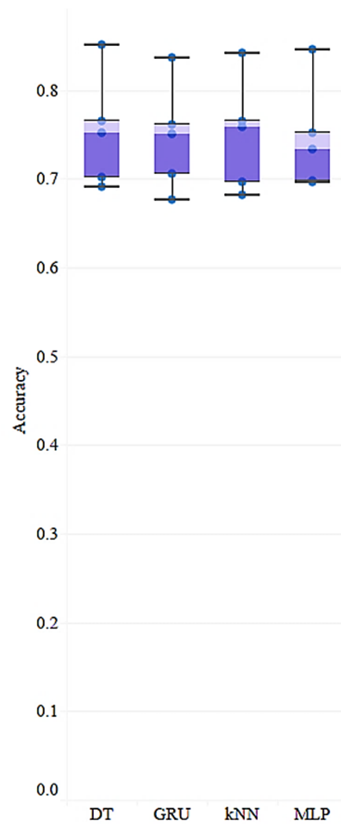


Fig. 5.7. The distribution of the accuracy calculated for the occupancy-state prediction models.

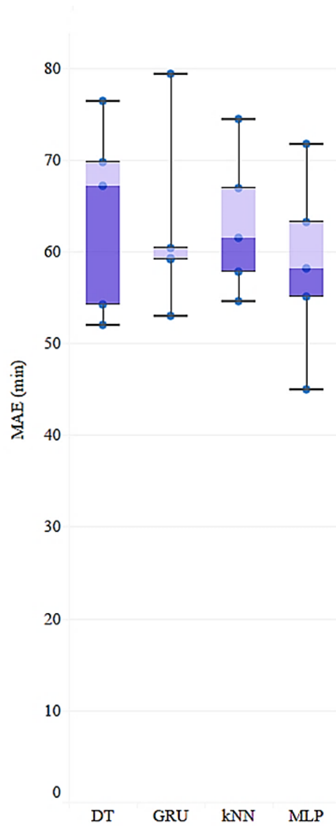


Fig. 5.8. The distribution of the MAE calculated for the arrival time prediction models.

5.4.2. Control system performance

The distribution of *MissTime* estimated for the RB control system based on different occupancy models for all the residential units is shown in Fig. 5.9. Using the occupancy-state prediction in the control system mainly results in lower *MissTime*, compared with the arrival-time models. kNN leads to the lowest *MissTime* with a median of 23 min/day. It is followed by the GRU model for the occupancy-state and arrival time prediction with a median of 33 and 35 min/day, respectively. Using the DT model for arrival-time prediction causes a poor thermal comfort condition at a median *MissTime* of 70 min/day. In addition to the developed occupancy models, the performance of the control system is also evaluated based on perfect occupancy prediction (i.e., the control system knows the actual future occupancy states in advance). Using perfect prediction in the control system leads to a *MissTime* of less than 5 minutes in all cases, which gives by far the best thermal comfort performance.

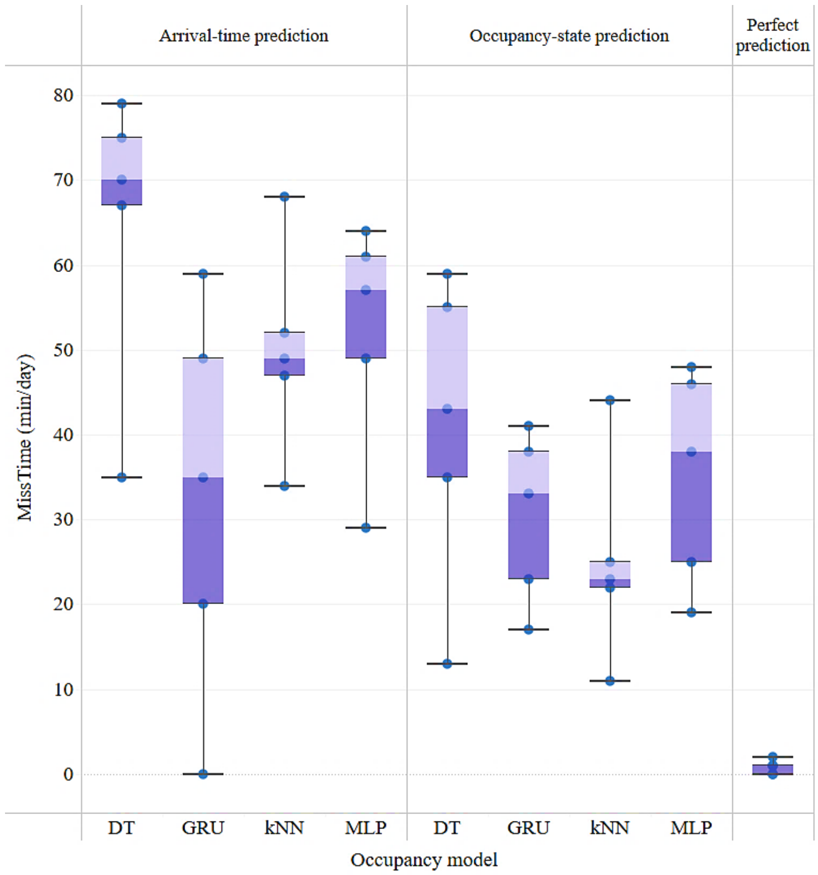


Fig. 5.9. The ditribution of *MissTime* performance for using different occupancy models in the RB control system.

Fig. 5.10 indicates the distribution of the energy saving obtained by using different occupancy models in the control system for all apartments, taking the always-on control as the baseline. Implementing the actual occupancy models leads to a higher energy saving than using the perfect occupancy prediction model in most cases. The energy saving is obtained due to the prediction errors in the occupancy models, which reduce the preheating periods at the expense of occupants' thermal comfort. It is not surprising that using arrival-time prediction models often provides a higher amount of energy saving than the occupancy-state prediction models since they usually sacrifice more thermal comfort, as demonstrated in Fig. 5.9. Utilizing the DT model for the arrival time prediction yields the best energy efficiency, saving up to 10.52% energy with a median of 5.77%, which is 1.47% higher than that of the perfect prediction model. The DT is succeeded by the MLP, which results in a median energy saving of 5.28%. Among all the models, the kNN model used for occupancy-state prediction showed the lowest potential of energy saving at a median of 4.12%, 0.18% lower than that of the perfect occupancy model.

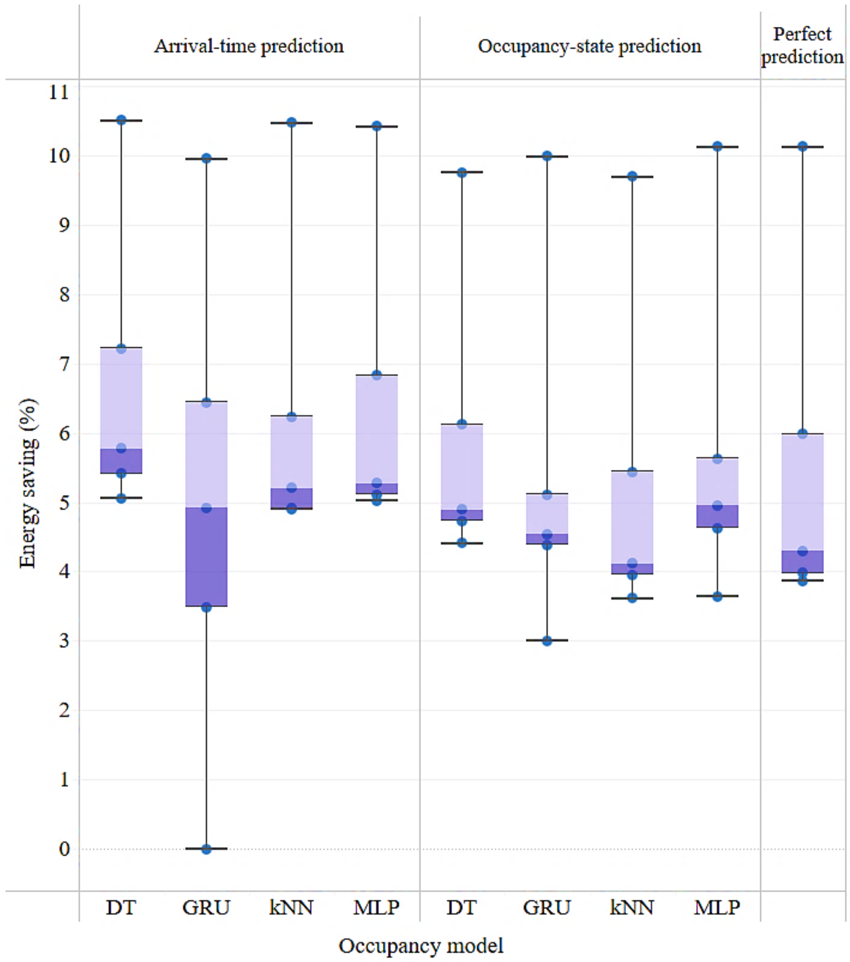


Fig. 5.10. The distribution of energy-efficiency performance of using different occupancy models in the RB control system.

5.4.3. Overall performance of the occupancy models

The mean relative closeness (i.e., overall performance score) calculated for each occupancy model used in the control system is demonstrated in Fig. 5.11. It is observed that using the occupancy-state prediction models in the control system provides higher performance than using the arrival-time prediction models. Regarding the occupancy-state prediction, the highest score is held by the kNN model at an average of 0.71. It is followed by the MLP, DT, and GRU, leading to similar R values at an average of around 0.6. Regarding the arrival-time prediction models, the control system based on the GRU model outperforms other algorithms, providing an average score of 0.51. The MLP and kNN result in a similar relative closeness at around 0.43. Employing the DT for the arrival time prediction significantly reduces the performance of the control system by providing an average score of as low as 0.32.

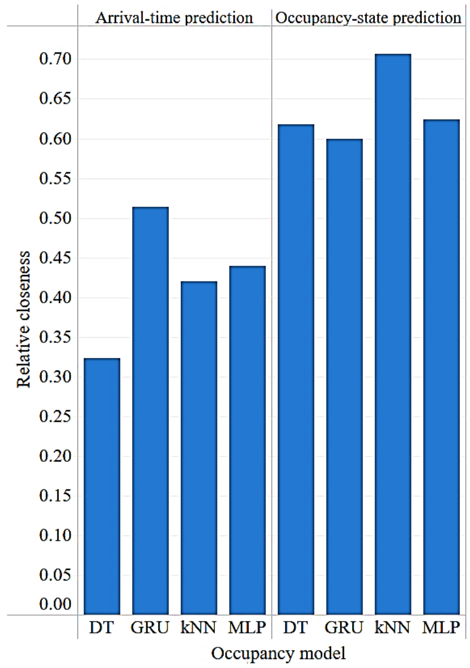


Fig. 5.11. The average relative closeness for the occupancy-state and arrival-time prediction models utilized in the RB control system.

5.4.4. Correlation between the performance criteria

In order to explore the relationship between the machine learning performance indicators (i.e., MAE and accuracy) and the performance of the control system, the averages of the indicators and the relative closeness are summarized in Table 5.3. The DT technique results in the highest average accuracy at 75.24%, and the MLP technique provides the lowest MAE of 58.62 minutes. Nevertheless, regarding the occupancy-state prediction models, the DT model offers a low relative closeness at 0.62, while the best score was provided by the kNN at 0.71. Furthermore, the GRU model outperforms the MLP technique by providing a relative closeness of 0.51 for the arrival-time prediction.

Table 5.3.
The average of the MAE, accuracy, and the overall performance score (relative closeness) for each occupancy model.

Model	Occupancy-state prediction		Arrival-time prediction	
	Accuracy (%)	Relative closeness (R)	MAE (min)	Relative closeness (R)
DT	75.24	0.62	63.91	0.32
GRU	74.65	0.60	62.23	0.51
kNN	74.92	0.71	63.04	0.42
MLP	74.51	0.62	58.62	0.44

To visually explore the relationship of the accuracy and MAE with the relative closeness, the normalized indicators and R values associated with each system are represented using a dot chart in Fig. 5.12. When normalized MAE increases from 0 to 0.32, the values of R also increase. However, when the MAE rises to 1, the score significantly falls to 0.44. It shows that the relationship between the MAE and R might not be strong. In terms of accuracy, there are two inverse correlations when normalized accuracy increases from 0 to 0.19 and from 0.56 to 0.62 as well as a direct relationship when accuracy rises from 0.19 to 0.56. This inconsistency between the variables can be a sign of week relationship between the variables.

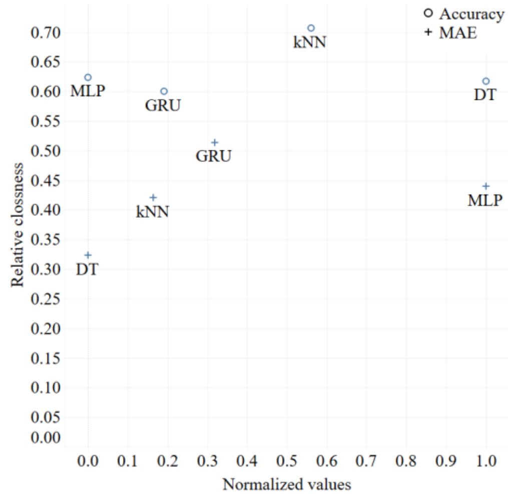


Fig. 5.12. Average relative closeness and the corresponding normalized accuracy and MAE associated with each model.

The Pearson correlation coefficients calculated between the MAE, accuracy, and normalized values and the relative closeness are depicted in Fig. 5.13. The MAE provides the highest correlation coefficient at 0.42, which is in the range of 0.3 - 0.7 and is considered as a moderate correlation [123]. However, for the accuracy, the coefficient is highly reduced to a range of 0 – 0.3, showing a weak correlation. Considering both MAE and accuracy indicators in a normalized form, the correlation is achieved as 0.23, which is also considered weak.

The weak to moderate correlations between the indicators point out that using the accuracy and MAE might be misleading criteria for evaluating and comparing the merits of occupancy models. One of the reasons could be linked with the fact that these indicators do not reflect the timing of the prediction errors. For example, the MAE and accuracy do not differentiate between the prediction errors that might occur after a long or a short vacancy period. In the former case, inaccuracies in the occupancy prediction are more substantial, as they can cause a significant amount of thermal discomfort for occupants. However, in the latter case, no considerable loss might occur because the temperature has not much receded from the setpoint yet. Furthermore, the MAE and accuracy are defined regardless of the control rules employed to operate HVAC systems. Depending on the control rules and constraints, occupancy prediction results in some points of time might have no effects on the HVAC operation. For instance, as discussed in Section 5.3.3.2, the implemented control system maintains a setpoint temperature one timestep after the departure of the occupants to ensure that the thermal comfort is not damaged due to the probable detection errors. Thereby, during such events when the temperature is always kept at the setpoint, the occupancy prediction results have no effects on the HVAC operation, and consequently, any errors in such periods do not have a negative impact on the overall systems' performance.

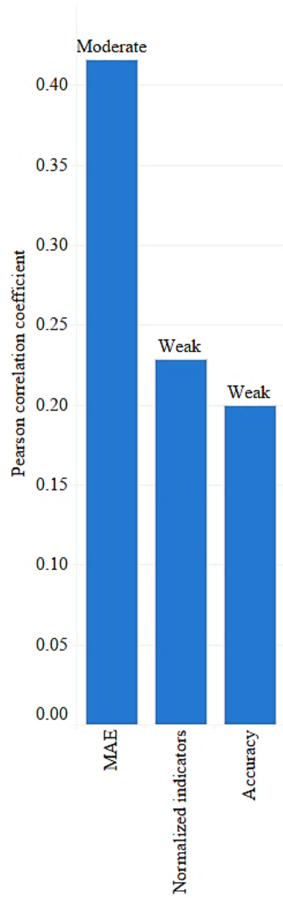


Fig. 5.13. Pearson correlation coefficient for evaluating the relationship between the machine learning performance metrics and relative closeness.

5.5. Conclusion

This study investigates the impact of selecting occupancy prediction models on the performance of occupancy-based HVAC control systems in residential buildings. Two different viewpoints of occupancy prediction modeling, namely arrival-time prediction and occupancy-state prediction, are studied. Additionally, the effectiveness of using different machine learning techniques, DT, kNN, MLP, and GRU, to predict future occupancy states in a predictive control system is evaluated. This study assesses the performance of the occupancy models when applied to a control system in terms of energy efficiency, thermal comfort, accuracy, and MAE. The TOPSIS method is implemented to provide an overall performance score by making a trade-off between the energy efficiency and thermal comfort.

The results reveal that selecting the occupancy prediction viewpoint (i.e., arrival time prediction or occupancy states prediction) has a higher impact on the HVAC performance than selecting the machine learning techniques. In cases when the thermal comfort of occupants is considered as the superior criterion, the occupancy-state prediction models generally provide higher performance in most cases with the kNN model leading to the lowest *MissTime*. In contrast, the arrival-time prediction models mostly provide higher energy saving with the DT model leading to the best median energy saving. However, in terms of the average overall performance score with the consideration of equally weighted thermal comfort and energy saving, the occupancy-state prediction models outperform other alternatives. Hence, in cases when the importance of the thermal comfort is greater than or equal to that of the energy saving, occupancy-state prediction using the kNN model often provides the best performance and is the recommended method. On the other hand, when the energy saving is given a substantially higher priority, using the DT model for the arrival-time prediction is recommended.

It is also noted that there is a weak to moderate correlation between the machine learning indicators and the overall performance score. The correlation between the accuracy and the overall score is calculated as low as 0.2, which indicates a weak correlation. It can be concluded that using statistical criteria to evaluate and compare the performance of the occupancy models might be misleading in some cases. Instead, it could be more reliable to evaluate the occupancy model performance when applied to a control framework. It is worth noting that the current study is performed based on a limited number of residential cases. In order to generalize the results, more comprehensive cases and different control systems need to be employed.

Chapter 6: Impact of predictor variables on the performance of future occupancy prediction: Feature selection using genetic algorithms and machine learning[5]

6.1. Overview

This study analyzes the impact of employing different features on the performance of future occupancy prediction models. The aim is to identify the most effective predictor variables for occupancy models in order to enhance the prediction performance. To this end, a multi-objective genetic algorithm (MOGA) is proposed and employed to maximize accuracy while minimizing the number of features utilized in the model. A trade-off between the mentioned objectives is provided by using the technique for order of preference by similarity to ideal solution (TOPSIS). The performance of the proposed MOGA is compared with that of a single-objective genetic algorithm, forward sequential selection, and backward sequential selection methodologies to assess the effectiveness of the proposed method. The results reveal that the MOGA provides superior performance and is able to improve the median accuracy by up to 4.81% from 70.94% for long-term occupancy prediction based on backward sequential feature selection while utilizing fewer features. Based on the TOPSIS method, no more than six features are required for developing the occupancy models with recent occupancy states and day of week selected as the essential features. In some cases, using recent CO2 levels, occupancy duration, lighting states, and previous occupancy states also shows a potential improvement in the prediction performance.

6.2. Introduction

The efficient operation of heating, ventilation, and air-conditioning (HVAC) systems has been an active research area, as building operation is responsible for a large amount of energy consumption worldwide and people spend approximately 90% of their life indoors [159,189,190]. The primary objectives of numerous research works have been to decrease the amount of HVAC energy consumption while ensuring the quality of people's work and life by providing acceptable thermal comfort. However, HVAC operation is still prone to needlessly conditioning vacant spaces and as a consequence, wasting energy [7,36]. This limitation is a consequence of ignoring the dynamic nature of occupancy patterns in building HVAC control. Furthermore, although programmable thermostats consider occupancy schedules in temperature control, it has been shown in several studies that they often failed to successfully save energy in residential buildings due to factors such as complex user interface design and the residents' ignorance [114]. However, it was reported that the proper use of the occupancy information in HVAC control can save energy by more than 28% [13,58,71].

To address the shortcoming of conventional control systems, reactive occupancy-based control was proposed and evaluated in earlier studies [50,52,54–59]. Such systems adjust the indoor

[5] This chapter is based on the following publication: M. Esrafilian-Najafabadi, F. Haghighat, Impact of predictor variables on the performance of future occupancy prediction: Feature selection using genetic algorithms and machine learning, Building and Environment (2022) 109152. https://doi.org/10.1016/J.buildenv.2022.109152.

temperature to the current occupancy state. Each occupancy state takes binary values of 0 and 1, indicating the vacancy and occupancy states, respectively. Control systems can also adjust ventilation rates to the number of people by considering the information about occupant counts, which is out of the scope of this study. Reactive control employs a setback temperature during vacancy hours to save energy while bringing back the desired setpoint as soon as occupants arrived. However, it can cause thermal discomfort upon occupants' arrival as it takes a while to bring back the desired temperature from a setback due to HVAC lag time [114]. To minimize the thermal discomfort, the setback temperature needs to be selected conservatively, which can adversely impact the amount of energy and cost savings in the reactive control. Furthermore, the sudden transition from the setback to setpoint temperature might also negatively affect the peak energy demand of buildings. It was demonstrated that using occupancy-based HVAC control systems could cause a 10% increase in the peak energy demand, compared with conventional always-on control [61]. Predictive control was proposed and evaluated as a promising alternative to overcome the mentioned issues [13,44,45,61,62,68–71,191]. The predictive control goes beyond relying only on the current occupancy information by looking into future probabilities of occupancy. The control system can take advantage of future occupancy prediction to pre-condition the building in advance by considering demand profiles and HVAC lag time. Such predictive systems have also demonstrated the ability to define and plan the setback temperature based on the probability of future occupancy presence to save more energy [60].

Future occupancy prediction models can be considered the heart of the predictive control systems, and consequently, their accuracy can highly impact the overall system performance [192]. Gluck et al. [17] reported that a 10% reduction in false positive errors and false negative errors could improve the energy saving and thermal comfort by up to 9% and 16%, respectively. However, due to the highly scholastic nature of occupant behavior, future forecasting becomes highly challenging. In order to improve the prediction performance, many research works were devoted to developing different machine learning and statistical models. However, despite the variety of occupancy models proposed in the literature, the impact of their input variables on the prediction performance has been mostly neglected. The following section provides a literature review on the types of utilized features and the developed occupancy models. Next, the limitations of earlier research work and the contributions of this study are elaborated.

6.2.1. Related research works

Different attributes were utilized to develop the occupancy prediction models in the literature. Among different alternatives, time of day has been employed for occupancy prediction in most of the previous works [114]. It makes perfect sense that there can be a strong correlation between the time of day and occupancy patterns. Many people develop different routines, such as sleeping at night and waking up in the morning based on a specific time of day, which can be extremely helpful in predicting future occupancy states in residential buildings. The time of day was considered the main feature to predict future occupancy patterns in several studies [13,60,157,193]. Erickson et al. [66] developed a Markov chain model to predict future occupancy patterns based on historical occupancy data. The time of day was considered in an inhomogeneous Markov model by implementing different transition matrixes in different timesteps.

Most researchers utilized multiple features in developing the prediction models to boost the performance. The time of day was followed by the day of week in terms of the frequency of use [114]. Killian et al. [191] reported that the uncertainty in occupancy prediction increased on certain days of the week, while the patterns were more predictable on the other days. The dependence of occupancy patterns on different days of the week can highlight the importance of this attribute in occupancy prediction. Krumm and Brush [87] developed occupancy prediction models as a function of time of day and day of week. Based on the historical occupancy patterns collected using GPS locations, they developed occupancy prediction models for different households and achieved an accuracy of higher than 60% for most cases.

The results achieved by Turley et al. [122] showed that using the day of week might not be the best choice for developing occupancy models. They developed several occupancy models and included either day of week or the weekends (taking **0** and **1** for the predictions on the weekdays and the weekends, respectively) as the attributes. It was demonstrated that in five out of six houses considered as case studies, the occupancy models using the weekends feature outperformed those using the day of week. Huchuk et al. [95] developed various machine learning models, including logistic regression (LR), k-nearest neighbors (kNN), random forests (RF), and long short-term memory (LSTM), for occupancy prediction using the time-of-day and weekends attributes. They showed that depending on the case study, the prediction technique, and the prediction horizon, the accuracy can vary in the range of 60%-90% in most cases. Shi et al. [74] assumed that the occupancy behavior generally differs during the weekends and weekdays and, therefore, split the entire occupancy database based on the weekends attribute. They utilized each split dataset and made use of the time of day to predict future occupancy states. Similarly, Scott et al. [45] split their database based the weekends attribute and developed the occupancy models using each created dataset. They predicted future occupancy states based on the mean of several historical days that were the closest points to the target label. Sangogboye et al. [82,84] included the time of day, day of week, and weekends features in the occupancy modeling procedure. They claimed that holidays and seasons could also impact the prediction performance because some events, such as school closure during the summer and the dependency of business hours on the holidays, can affect occupancy patterns in residential buildings.

Some studies proposed additional features to be implemented in the occupancy models. Yu [89] reported that the previous occupancy and vacancy durations might impact the prediction performance. They employed these variables for occupancy prediction using genetic programming. Occupancy duration was also utilized in [83,88,90] for developing the occupancy models. Gjoreski et al. [83] predicted the future arrival and departure time of an employee in an office building using the month of the year and environmental features, such as outdoor temperature. Dong and Andrews [65] developed occupancy models based on the data received from acoustic, lighting, carbon dioxide (CO_2), temperature, and relative humidity sensors. They assumed that the changes in the sensor data can provide valuable information for occupancy prediction and defined such meaningful changes as different events in the database. The defined events were utilized in an episode discovery method to find the important patterns and to predict future occupancy. The discovered patterns were employed in an HVAC control system to adjust the ventilation and indoor temperature to occupancy for saving energy. Salimi et al. [91] employed occupants' identities, their activity types, their zonal locations, and time-of-day features in developing a Markov model to predict future occupancy. They considered different types of activities, such as working states, lunch breaks, short breaks, and meetings and reported up to 92% accuracy for predicting occupancy states in an office. Ryu and Moon [96] focused on using different attributes to deal with the privacy issues that might be caused due to using a network of cameras or motion detectors in residential buildings. They implemented CO_2 concentration, appliance energy consumption, lighting levels, and the time of day and developed a decision tree (DT) and a hidden Markov model to detect and predict occupancy states.

Some researchers developed occupancy models as a function of occupants' current locations. Gupta et al. [62] proposed a thermostat control that communicated with occupants' cellphones and retrieved their current locations. Accordingly, the control system estimated the time it took for the closest occupant to arrive at home and signaled the HVAC system to start preheating the building in advance. Using this control framework, no thermal discomfort was reported while saving 7% energy, compared with conventional thermostats. Krumm and Brush [87] integrated such spatial data obtained from occupants' cell phones with a probabilistic occupancy prediction model and showed that the integrated model could provide superior performance when compared to each standalone model.

6.2.2. Research gaps and contributions

As discussed in the previous section, a variety of different attributes were proposed and employed in earlier studies for developing the occupancy models. These predictor variables are summarized in Table 6.1. Many studies developed the prediction models based on the most frequent features, such as the time of day, day of week, and weekends. On the other hand, there are also studies that proposed additional features, such as indoor CO_2 concentration. Given the variety of candidate features, a wise selection of the features among the alternatives becomes essential for developing the occupancy models since employing too many or too few attributes might respectively bring about overfitting or underfitting issues [14].

Based on the mentioned features, different machine learning models, such as kNN and LR, have been developed in earlier studies for occupancy prediction. It has been reported that implementing feature selection (FS) methods as a preprocessing step before developing such models can improve the prediction performance in different applications, such as electricity price forecasting [194–196]. However, concerning the occupancy prediction problem, this step was mostly neglected in earlier studies.

Some research works implemented tree-based models for occupancy prediction. Due to their built-in FS algorithms, they have shown the ability to avoid the use of irrelevant features by ranking the attributes based on their importance [197,198]. However, an FS process before developing these models is also valuable as it can enhance the computational speed, further improve the accuracy, and result in a better understanding of the prediction process [15,199]. Furthermore, finding the most relevant features can minimize the cost and efforts of installing the sensors in buildings [16] and reduce the costs associated with data storage [200].

Peng et al. [61] employed a sequential FS method for developing future occupancy prediction models. However, they considered few candidate features in this process, neglecting most of the possible predictor variables. Consequently, it is still unclear what types and number of features can provide the best performance in occupancy prediction models. There is a need for a study to consider an extensive number of candidate features and to find the optimal types and size of feature sets by avoiding irrelevancy and redundancy.

This study aims to determine the most effective features and the optimal size of the feature sets for use in data-driven occupancy prediction models in residential buildings. For this purpose, an optimal FS method based on a multi-objective genetic algorithm (MOGA) is proposed and implemented. The objective of the optimization problem is to maximize the occupancy prediction accuracy while minimizing the number of utilized features. Two approaches, namely maximum accuracy and technique for order preference by similarity to ideal solution (TOPSIS), are proposed to determine an optimal solution within the Pareto front solution set. In order to show the superiority of the proposed FS process, the performance of the MOGA is compared with that of a single objective genetic algorithm (SOGA), forward sequential selection (FSS), and backward sequential selection (BSS) as conventional FS methods. The process is performed based on an occupancy database collected from 32 apartments.

The rest of the chapter is structured as: Section 6.3 provides details about the implemented database, its preprocessing step, and the FS methods. Section 6.4 discusses the results, and finally, the conclusions are presented in Section 6.5.

Table 6.1.
The attributes utilized for developing the occupancy prediction models in the literature.

Feature	Reference
Calendar features	
Time of day	[13,60,82,84,114,157,193]
Day of week	[82,84,87,122,191]
Weekends	[45,74,82,84,95,122]
Holiday	[82,84]
Month	[83]
Season	[82,84,88]
Indoor features	
Acoustic data	[65]
CO_2 concentration	[65,96]
Energy consumption	[96]
Indoor temperature	[65]
Lighting	[65,96]
Relative humidity	[65]
Other features	
Identity	[91]
Activity type	[91]
Occupancy/Vacancy duration	[83,88–90]
Location	[62,87]
Weather condition	[83]
Zonal location	[91]

6.3. Methodology

6.3.1. Database description and preprocessing

An extensive occupancy database, gathered from 32 apartments in Lyon, France [29,120,156,174,175], is utilized for investigating the impact of features on the occupancy prediction performance. In the original database, the occupancy data was recorded at 1-minute intervals based on the signals received from passive infrared (PIR) sensors. It is worth noting that the use of PIR sensors can cause inaccuracy in occupancy detection. More specifically, during periods when occupants are mostly stationary at home, such as when they are sleeping or resting, the PIR sensors are prone to reporting false vacancy states. In order to minimize the mentioned errors, two actions were taken in both data collection and preprocessing steps. Firstly, in the data collection phase, attempts were made to construct a sensor network that can capture as many movements as possible. For this purpose, each part of the apartments was equipped with at least one PIR sensor; one sensor was installed in each bedroom, bathroom, and kitchen. Additionally, two sensors in the living room and several sensors in the corridors. The sensor network is intended to catch small motions even while the occupants are resting with little mobility. However, it is natural that there are still some moments when the occupants might be completely stationary with no motions recorded by the sensors. In such cases, using too short time intervals (e.g., 1-min periods) can increase the chance of false vacancy states. For example, it can be common that occupants remain stationary with no motions at 1-min intervals; however, having no motions in longer periods such as half an hour is much less frequent. Hence, in the data preprocessing step, the data resolution is transformed to 30-minute intervals as recommended in [95]. In the transformed database, if any motions are recorded by any sensors in an apartment over a 30-min period, the corresponding time interval takes **1** to show the occupancy state; otherwise, **0** is stored as the label to indicate the vacancy state at the considered timestep.

Although the occupancy detection accuracy has been improved using the above-mentioned steps, the PIR sensors are still prone to reporting wrong vacancy states. The ideal approach would be to employ more accurate detection systems, such as networks of cameras. Nevertheless, using such networks in residential cases can cause privacy issues for the occupants and cause considerable infrastructure costs. It should be noted that there are different factors that need to be considered when selecting an occupancy monitoring network, including detection accuracy, infrastructure costs, privacy, and added complexities. To make a trade-off between such factors, the mentioned PIR network is selected and utilized for the purpose of occupancy detection, as also employed in earlier works [29,95,120,156,174,175].

A brief description of the apartments, including the types of the apartments, corresponding occupancy ratios, and the floor plan is provided in Table 6.2 and Fig. 6.1. The occupancy ratio is defined as the number of occupied states divided by the number of total timesteps, which quantifies the proportion of time when the apartment is occupied. It should be noted that due to the above-mentioned PIR sensor errors, the reported occupancy ratios are under-estimated. The database involves eight one-bedroom, eight two-bedroom, and 16 three-bedroom apartments. A direct correlation is observed between the occupancy ratio and the number of bedrooms. Three-bedroom apartments have the largest ratio at 55.95%, while 1-bedroom apartments result in an average ratio of as low as 41.59%. The reason could be that the apartments with more bedrooms in the same building are probably occupied by more crowded households and, as a result, can remain occupied for longer periods.

Table 6.2.
Information about the type of apartments and occupancy ratio based on 30-min intervals.

Type	Symbol	Num. of apartments	Num. of sensors per apartment	Area (m^2)	Occupancy ratio (mean ± SD)
One-bedroom apartment	T2	8	10	52	41.59 ± 14.08
Two-bedroom apartment	T3	8	11	74	55.59 ± 13.1
Three-bedroom apartment	T4	16	13	110 and 90	55.95 ± 19.4

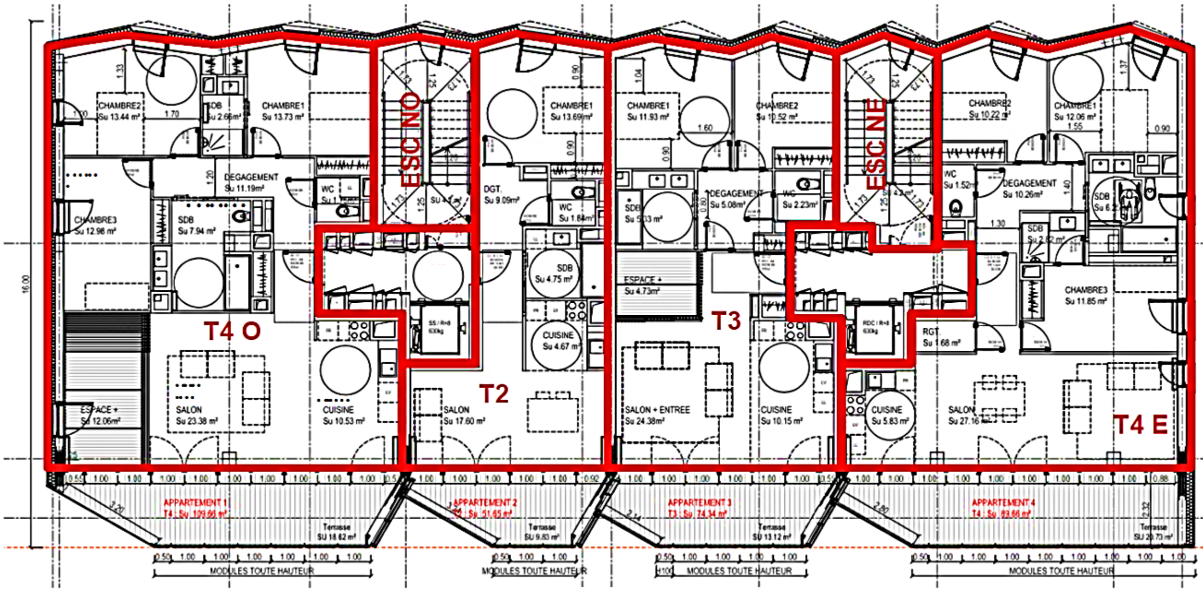


Fig. 6.1. The floor plan of the apartments utilized to collect the occupancy database.

In order to visually explore possible occupancy patterns in the database, the average occupancy ratios in each hour of day, and day of week are demonstrated for three sample apartments in Fig. 6.2. Although the whole feature selection process is performed based on 30-min intervals, the visualization of the occupancy ratios is provided based on 1-hour periods in this figure. It is because presenting the data at half hours (i.e., at 30-min intervals) might overcomplicate the visualization process without adding further values.

An occupancy pattern can be observed for the sample T4 apartment. The large occupancy ratios from 8:00 to 12:00 indicate the higher occupants' tendency to stay at home in the morning. However, the average occupancy ratios significantly fall between 12:00 and 16:00, showing that the probability of vacancy hours is higher during this period. Furthermore, the occupancy ratios significantly drop on Sundays, indicating the possible dependence of the occupancy patterns on the day of week. Similarly, T3 apartment also demonstrates a quite regular occupancy pattern. The household tends to leave home during working hours as the occupancy ratios are substantially smaller from 8:00 to 20:00. However, they often stay at home during the weekends leading to larger occupancy ratios. In contrast, no clear occupancy patterns can be detected by visualizing the occupancy ratios for the T2 apartment possibly because the household might not follow a regular daily occupancy routine.

6.3.1.1. Candidate features

Table 6.3 summarizes the features that are available in the utilized database and are employed as candidate features in this study. These features are classified into four categories: calendar, previous occupancy patterns, indoor, and environmental features. The features in the first category are intended to help the model to learn seasonality in the occupancy patterns. The seasonality in time-series data is defined as the patterns that are repeated at the same frequency [201]. Most of the previous studies highlighted the possible impact of time of day, day of week, and weekends on the occupancy patterns. Although a few studies suggested the use of longer-term features such as months and even seasons, they are excluded from the current study. It is because in order to capture monthly and quarterly patterns, the occupancy data over a long period is required, which is not available in this study.

The time-of-day feature is created by dividing the entire day into 48 equal intervals with each interval taking values from 0 to 47, denoting 00:00 to 23:30, respectively. The day-of-week attribute takes values from 0 to 6 to show the days of the week in a sequential order starting from Monday. As people tend to go for vacations on the weekends and holidays, these attributes might also contribute to providing more accurate occupancy predictions. Hence, weekends and holiday attributes are also included as candidates, taking **1** and **0** to indicate days off and working days, respectively.

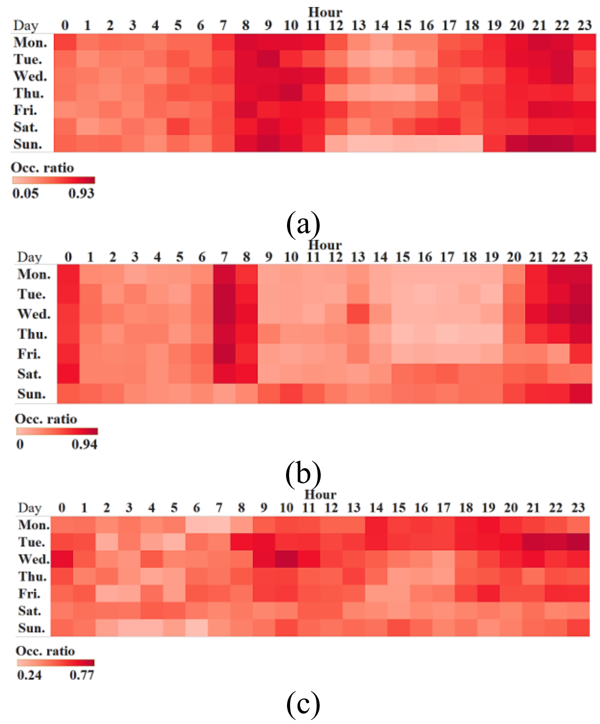


Fig. 6.2. The average occupancy ratios in different hours of day and days of week for three sample a) T4, b) T3, and c) T2 apartments.

Table 6.3.
The candidate features utilized in the FS process.

Feature	Unit
Calendar features	
Time of day	-
Day of week	-
Weekends	-
Holiday	-
Previous occupancy patterns	
Current and lagged occupancy states	-
Occupancy duration	-
Vacancy duration	-
Indoor features	
Current and lagged CO_2 concentration	ppm
Current and lagged lighting states	-
Current and lagged appliance energy consumption	kWh

Current and lagged window states	-
Weather condition	
Current and lagged outdoor temperature	°C

As the occupancy prediction problem is addressed as a time series analysis, the previous occupancy states might also be helpful in predicting future occupancy patterns. In this study, the occupancy state at timestep t (i.e., the current time step) and 12 occupancy states at previous time steps (i.e., at t-1, t-2, …, t-12), called lagged variables, are utilized to create a binary occupancy vector as shown in Fig. 6.3. In order to create the lagged variables, the current occupancy state at each time step needs to be regularly stored in a database so that the required information from the previous timesteps can be always available for the model. Each element of this vector can take **0** and **1** representing vacancy and occupancy states, respectively [45]. The occupancy information can also be used as the occupancy or vacancy duration. These variables demonstrate how long the building has remained in the last occupancy or vacancy states. For example, if the building has been in the vacancy state for *p* sequential timesteps, the vacancy duration takes *p* as its current value. In this case, the occupancy duration takes the number of consecutive timesteps when the building was occupied prior to the last state (i.e., the vacancy period in this case).

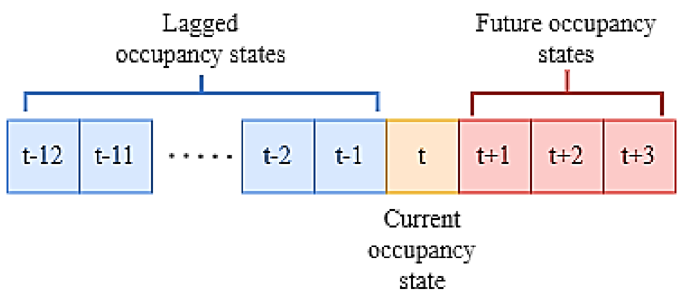


Fig. 6.3. The lagged and current occupancy states utilized to forecast future occupancy.

There might also be a relationship between the occupancy patterns and CO_2 concentration, lighting states, and energy consumption [65,96]. The CO_2 concentration is correlated with the number of people occupying a place [94,100]. This correlation can be utilized to correct the possible PIR sensor errors when people are stationary with minimum movements. During such events, although no motions are recorded by the motion detectors, the occupancy model can estimate whether the building is still occupied or not based on the CO_2 data. Additionally, the information about the occupant counts can help the model to better predict future occupancy states since there might be a relationship between the current number of people occupying a place and the possible future occupancy behavior. The changes in the CO_2 concentration can be a sign of changes in the occupancy patterns, and as a result, similar to the occupancy-state attribute, 13 CO_2 features (i.e., the CO_2 concentration at the current timestep and at 12 previous consecutive timesteps) are considered as candidate features in the selection process. The lighting state is also considered a candidate feature. It may also help the model to correct wrong occupancy sensor recordings. For instance, if no motions are recorded from the sensors, the probability of a false vacancy state is higher when the lighting states have remained unchanged. Besides, the lighting patterns can be correlated with the occupancy patterns and can help to improve the prediction accuracy [65]. The lighting states take **0** and **1**, denoting the on and off states, respectively. The plug loads in the households are also considered candidate features as there might be a relationship between occupants' habits, such as watching TVs and using coffee machines, with their occupancy behavior. The energy features recorded in the database have rare large values that might be outliers due to sensor errors [202]. Hence, the database needs to be preprocessed to detect and handle the potential outliers. To this end, the standard deviation (SD) method as a statistical approach is employed. According to this method, all the data elements which are not placed in the following range are labeled as outliers [203,204]:

$$SDMethod: \mu \pm 3\sigma \tag{18}$$

in which μ is the mean and σ is the standard deviation of the data. In this study, if the plug load value of a data instance, which is labeled as an outlier, is placed above (below) this range, they are replaced by the upper bound (lower bound) values. Fig. 6.4 illustrates the plug loads recorded for a sample apartment before and after the outlier handling step. There are a limited number of timesteps with an unusually large energy consumption of as high as 300 kWh before preprocessing the data. These rare events can be considered as noise and might cause overfitting to the train set when learning the occupancy behavior. After eliminating the potential noise, the data is placed in a more narrowed range of 10-110 kWh. Hence, the concerned rare events with unusually high energy consumption are smoothed out in the preprocessed database, and therefore, the SD method shows the ability to deal with the possible outliers. It is acknowledged that the energy consumption data might not follow the Gaussian distribution assumption in some cases. However, in this study, the SD method is utilized because of its ability to address too large elements of energy consumption data and its simplicity.

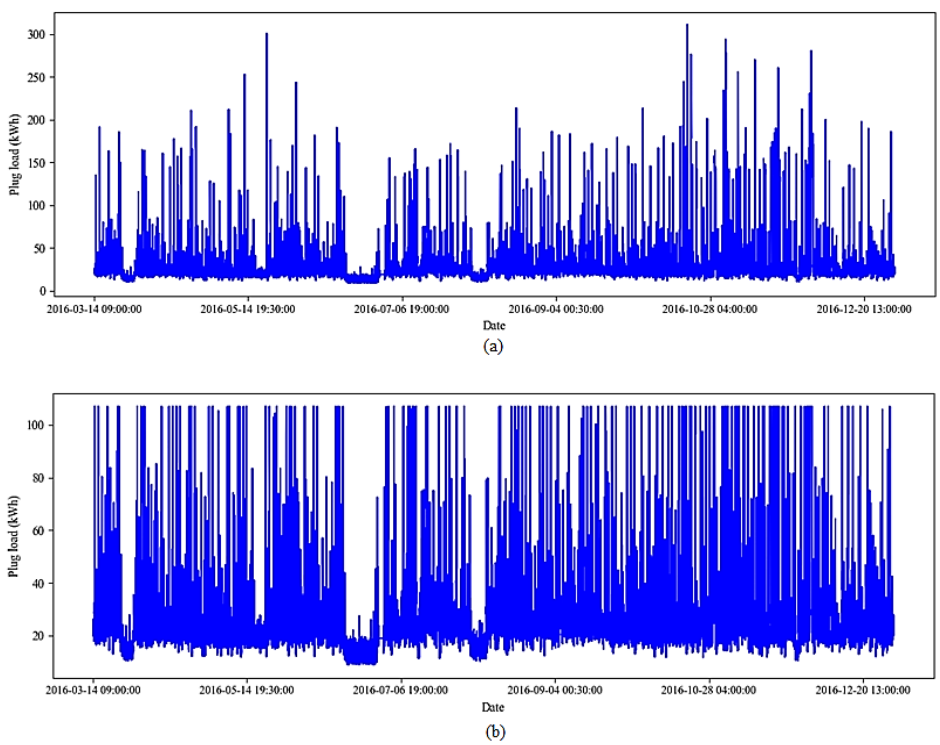


Fig. 6.4. The plug load patterns from a sample apartment a) before and b) after addressing the outliers.

The window states, taking **0** and **1** respectively for closed and opened windows, are selected as candidate features because of the direct relationship between the CO_2 concentration and the building ventilation rate. When occupants open a window, they can change the CO_2 level in the building by allowing more natural ventilation. In such cases, the correlation between the occupancy level and CO_2 level might change. Thereby, utilizing the window state and its lagged variables might be helpful in future occupancy prediction, especially when utilized along with the CO_2 concentration.

The last category consists of the outdoor temperature. This attribute can impact vacations and recreational activities and, thus, may be helpful in developing the occupancy models. For example, people might tend to stay at home in extremely cold weather and as a result, occupy the building more often during such events.

The database utilized in this study consists of the sensor data from March 14, 2016, to the end of the year. It is split into two datasets of training and test. The training dataset consists of the data from March 14, 2016, to November 14, 2016. The FS process is performed on the training dataset to find the effective features. In this process, the k-fold time-series cross-validation (CV) method is utilized to evaluate the performance of each occupancy model using the Scikit-learn library in Python [103]. Generally, the value of k is selected to make a trade-off between bias and variance of the prediction model [201]. However, as the FS approaches used in this study are rather computationally expensive, determining the k value should also consider the computational speed. Although the k values of 5 and 10 have been selected most frequently in the literature because of their acceptable bias-variance trade-off provided in most cases [205], this study selects a smaller value of k at 3, as also suggested in [95], to make the FS process computationally affordable. In the next step, the final performance of occupancy models based on each FS method is unbiasedly evaluated based on the unseen test dataset.

6.3.2. Optimal feature selection

The MOGA is proposed and developed to find the optimal features that can lead to the highest occupancy prediction performance. More specifically, the MOGA tries to find a set of solutions that maximize the accuracy of the occupancy prediction while minimizing the number of utilized features. Accuracy is defined as the number of correct predictions (i.e., the sum of true positive and true negative predictions) divided by the total number of predictions for each occupancy model. According to its definition, accuracy gives equal weights to all different errors regardless of their timing and importance. Hence, it was suggested in [192] that the performance of the occupancy prediction models should be evaluated in the context of HVAC control in terms of practical performance criteria such as thermal comfort and energy saving. However, implementing such detailed metrics requires thorough models of buildings and control systems, which can add to the problem complexity and the required computational time. Since the FS methods utilized in this study are iterative and relatively time-consuming, using such performance metrics is too computationally expensive. Hence, the accuracy, as the most selected performance score for occupancy model evaluation in the literature [114], is selected in this study to enhance the evaluation speed.

The reason behind considering the second objective (i.e., minimizing the number of features) is to avoid selecting redundant or irrelevant features since they might bring about the overfitting issue and, consequently, a decrease in the prediction accuracy when applied to new unseen datasets [206]. In order to assess the effectiveness of the proposed MOGA, its performance is compared with that of the SOGA, which only focuses on maximizing the prediction accuracy.

6.3.2.1. Genetic algorithm for feature selection

The genetic algorithm (GA) was initially inspired by the evolution theory [207]. It is an iterative search method that has been implemented for a variety of problems, aiming to find the global optimal solution [135]. As demonstrated in Fig. 6.5, the GA begins the optimization process by randomly initializing the first population of chromosomes (also called individuals). Each chromosome is a potential solution to the optimization problem (i.e., a candidate feature subset). The GA repeatedly tries to improve the population and obtain the optimal feature set using some operations, including crossover and mutation [208]. The number of chromosomes is called the population size, which is denoted by N in this study. In the FS problem, each chromosome can be represented by a bit string taking M binary values, in which M is the number of candidate features; **0** and **1** values respectively indicate whether the corresponding feature is utilized or neglected in developing every machine learning model. In the next step, N machine learning models are trained, and the corresponding accuracies are measured. In the MOGA, the fitness of each individual is evaluated as a function of the accuracy and the number of features, while only accuracy is considered in the SOGA. As the individuals with higher merits (i.e., higher fitness) are more likely to result in the optimal solution, they are ranked based on their fitness. The best N/2 of the individuals are selected to be carried forward to the next population using the tournament selection method [209], and the rest is eliminated from the population. In order to reconstruct the entire population, the removed individuals are replaced with the offspring of the selected ones. More specifically, the genes of the selected individuals (also called parents) are combined, and new children are generated. After creating the new generation, a mutation rate is utilized to add more diversity amongst the children by randomly changing their genes. The entire process is repeated, and the population is continuously improved until the stopping criterion is met (i.e., the maximum number of iterations). The parameters utilized to develop the GA in this study are summarized in Table 6.4.

Table 6.4.
The parameters utilized in the MOGA and SOGA.

Parameter	Value
Population size, N	200
Number of maximum iterations	200
Mutation rate	0.1
Selection method	Tournament
Maximum number of features	20

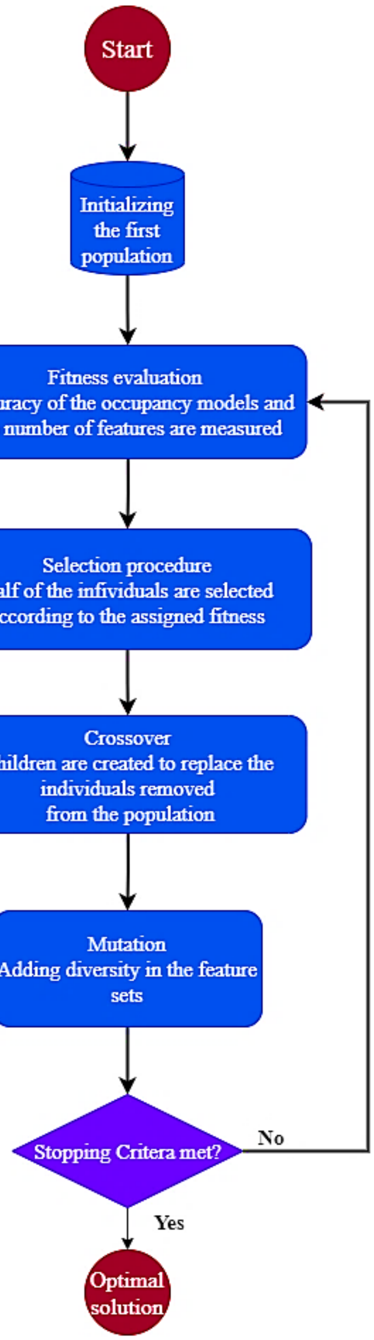


Fig. 6.5. The schematic diagram of the MOGA.

The FS methods utilized in this study are considered wrapper methods, in which the FS process is performed by repeatedly measuring and enhancing the performance of a developed prediction model [111]**.** Consequently, an occupancy prediction model needs to be selected and implemented in the FS process. Various algorithms, including conventional machine learning techniques, deep learning algorithms, and sequential models can be utilized for this purpose [192]. However, one of the main limitations of the GA algorithm is the rather long computational time [138], and consequently, it is of essential importance to choose a fast occupancy model. This limitation prevents the use of more complex algorithms, such as deep learning models, as they are often associated with relatively longer training time. In this study, the occupancy models are developed based on DT algorithms that can provide a relatively higher computational efficiency [210]. As demonstrated in [192], DT models can provide an acceptable level of accuracy when compared to kNN, multi-layer perceptron (MLP), and gated recurrent unit (GRU) algorithms. The DT algorithm is developed in Python using the Scikit-learn library [103] based on the entropy metric as the attribute selection criteria. The SOGA and MOGA techniques are developed using the Pymoo library [211], as a powerful multi-objective optimization tool, in the same environment. The MOGA and SOGA are employed separately for each apartment and for different prediction horizons of one, four, and eight timesteps (i.e., 30, 120, and 240-minute predictions in advance), which are respectively named short-, mid-, and long-term predictions throughout this chapter.

6.3.2.2. Multi-criteria decision making

As discussed earlier, the MOGA tries to maximize the accuracy and minimize the number of features utilized in the occupancy prediction. Since the MOGA does not often converge to a unique optimal solution due to the conflict between these objectives, it gives a set of the Pareto optimal solution (also called non-dominated solutions) for each apartment and prediction horizon. The Pareto optimal set is defined as a solution set that cannot be dominated by other solutions in terms of both criteria [212]. In other words, none of these solutions can be improved without sacrificing at least one of the criteria (i.e., either the accuracy or the number of selected features) [213].

In such multi-criteria decision-making (MCDM) processes, a solution among the alternatives can be selected according to the decision makers' preferences [214]. Two decision making (DM) approaches are implemented in this study: first, it is assumed that as long as the accuracy increases, the decision-maker is willing to provide the model with the required attributes, and as a result, there is no limit on the number of features utilized in the model. As the only focus of this method is to increase the accuracy, this approach is named the maximum accuracy (MA) method. In the second approach, it is considered that the decision-maker is willing to make a trade-off between the number of features and the accuracy. It is especially helpful when the decision-maker needs to minimize the number of sensors to be installed in a building and to keep the occupancy modeling problem as simple as possible. To make the mentioned trade-off, the TOPSIS methodology is utilized.

In TOPSIS, each alternative within the Pareto set is compared with two ideal points, namely the positive ideal solution (PIS) and the negative ideal solution (NIS) [145]. The PIS is an ideal solution with the best ideal performance, i.e., with the highest accuracy and fewest number of features among all the alternatives. In contrast, the NIS indicates the worst case with the largest feature subset and the lowest accuracy. The alternative with the shortest Euclidian distance from the PIS and longest distance from the NIS is considered the optimal point. In order to mathematically select the best solution, the relative closeness for the alternative j, R_j, can be calculated for each point as follows [144]:

$$R_j = \frac{d_j^{NIS}}{d_j^{PIS} + d_j^{NIS}} \tag{19}$$

in which d_j^{NIS} and d_j^{PIS} indicate the distances between the alternative j and the NIS and PIS points, respectively. The point with larger relative closeness is selected as the ultimate solution for each apartment and prediction horizon.

6.3.3. Sequential feature selection

It is acknowledged that the MOGA adds a great deal of complexity to the FS process and, as a result, it is essential to find out whether the added complications can be compensated by providing superior performance. In other words, using the MOGA need to result in a higher prediction accuracy or fewer selected features, compared with the more straightforward FS methods. To this end, two conventional sequential FS methods, namely FSS and BSS, are used as baselines. Although many studies have evaluated and compared the performance of the BSS and FSS techniques in different problems, it is still not clear which method provides superior performance for a given database [215]. Therefore, both methods are employed and compared in this study.

Fig. 6.6 shows the working diagram of the FSS algorithm. The process begins with an empty feature subset. In the first step, **M** different feature subsets are created with a size of **i**, in which **i** denotes the number of iterations. Each created subset is employed to develop an occupancy prediction model and its performance is evaluated using the CV. The feature set that leads to the highest occupancy prediction performance is permanently added to the feature subset and proceeds to the next iteration. This process continues until the size of the feature set equals a predefined feature number, which is assumed as 20 for both FSS and BSS.

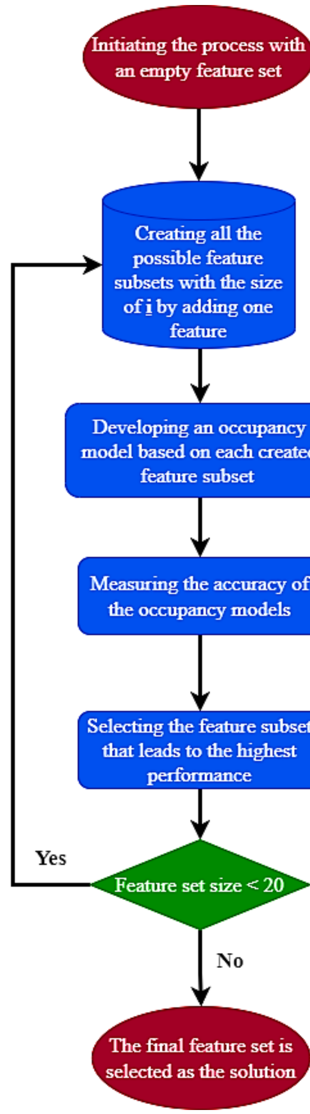


Fig. 6.6. The schematic diagram of the FSS process.

In contrast, the BSS starts the process with a subset involving all the candidate features. As shown in Fig. 6.7, in the first step, all the possible feature sets with the size of **M-i** are created by removing a single feature. Based on each subset, a DT model is developed, and its performance is assessed using the CV. The feature that its removal leads to the highest performance is permanently removed from the candidate features for proceeding to the successive iterations [216]. Similar to the FSS, the process continues until the predefined number of features is met.

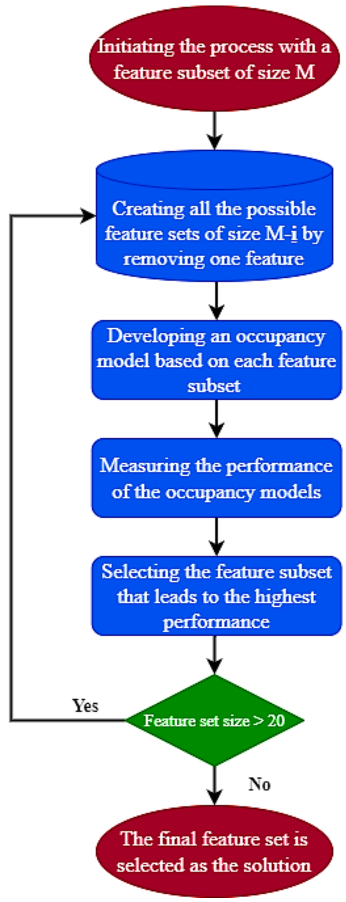


Fig. 6.7. The schematic diagram of the BSS process.

6.4. Results and discussion

6.4.1. Multi-objective genetic algorithm

This section discusses the Pareto solution set provided by the MOGA and compares the performance of the MA and TOPSIS approaches. The normalized Pareto frontier solution set, PIS, NIS, and the selected solutions based on the TOPSIS and MA methods for a sample T4 apartment and for the short-term prediction are demonstrated in Fig. 6.8. The contradiction between the two objectives can be observed; as the number of features decreases, the accuracy of the model falls. Consequently, as discussed in Section 6.3.2.2, no features can provide a superior performance when both criteria are considered. Using TOPSIS, an optimal solution is chosen among the alternatives with the shortest and longest distances from the PIS and NIS, respectively, while the MA approach selects the point with the highest accuracy regardless of the number of features.

In order to compare the final solutions selected based on the TOPSIS and MA methods, the mean and standard deviations of the test accuracy for both approaches are summarized in Table 6.5. The largest distinction between the average performance occurs for mid-term occupancy prediction; the accuracy decreases from 79.09% in the MA method to 78.38% in the TOPSIS method, which is a 0.71% difference in the mean test accuracy. In contrast, the selection approach has a minor impact on the short-term occupancy prediction with a 0.31% difference between the mean accuracy of both methods. As for the mid-term occupancy prediction, the average accuracy drops by 0.56% from 77.85% for the MA to 77.29% for TOPSIS.

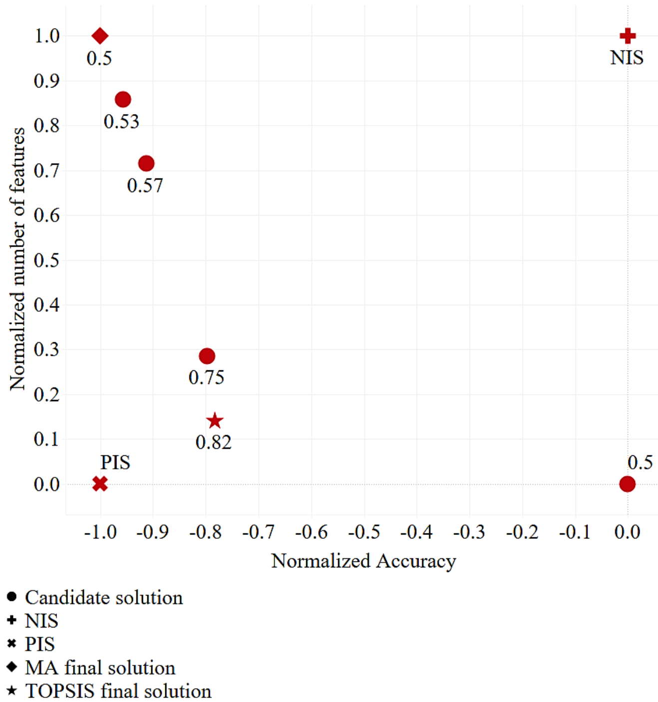


Fig. 6.8. A sample Pareto frontier solution including the PIS, NIS, and the selected solutions based on the TOPSIS and MA methods.

Table 6.5.
Mean and standard deviation of the occupancy prediction test accuracy for the MA and TOPSIS methods.

DM approaches	Prediction horizon		
	Short term	Mid term	Long term
MA	83.89 ± 5.45	79.09 ± 7.55	77.85 ± 7.68
TOPSIS	83.58 ± 5.56	78.38 ± 7.70	77.29 ± 7.74

It is worth noting that the existence of a set of solutions rather than a unique optimum makes it rather complicated to check whether the algorithm converges to the global optimum point, or it can still search for a higher quality solution. To address this issue, hypervolumes are calculated to track the convergence of the algorithm. The hypervolume indicator has been one of the most utilized set-quality indicators in multi-objective optimization problems [217]. It is defined as the volume between the Pareto solutions and a user-defined reference point [214]. Fig. 6.9 demonstrates the changes in the hypervolume as the algorithm proceeds with the reference point assumed at (1, 1) for a sample apartment. After almost 12,000 evaluations, the algorithm almost converges to a stable solution; the following negligible changes in the hypervolume increase the probability that the optimization algorithm has reached the global optimum rather than the local ones.

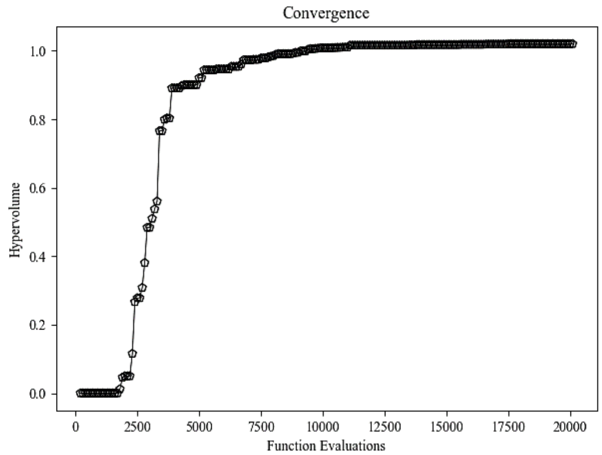


Fig. 6.9. The convergence of the hypervolume quality indicator based on the number of function evaluations.

6.4.2. Performance of feature selection methods

In order to compare the performance of the FS methods, the distribution of the test accuracy of the occupancy prediction models developed based on the MOGA, SOGA, FSS, and BSS are demonstrated using box plots in Fig. 6.10. To analyze the impact of the prediction horizon, the results are categorized according to the short-, mid-, and long-term predictions. As this comparison is based on testing accuracy, the MA approach is implemented for decision-making in the MOGA. It can be observed that the MOGA outperforms other methods by providing the largest maximum, minimum, and median test accuracy in all the categories. It is observed that the prediction accuracy falls by increasing the prediction horizon regardless of the FS method. It is because predicting longer-term events often causes higher uncertainty and, consequently, lower prediction accuracy. The MOGA yields a median accuracy from 75.75% for the long-term to 81.95% for the short-term predictions. It is followed by the FSS for all the prediction horizons in most cases. It is worth mentioning that despite the higher complications in developing the SOGA, it is mostly outperformed by the FSS. In most cases, the BSS gives the poorest performance, leading to median test accuracy in the range of 70.94%-79.05% as a function of the prediction horizon. The difference between the median accuracy of the MOGA and the BSS is as high as 4.81% (for long-term predictions), highlighting the importance of choosing the right FS method in occupancy prediction.

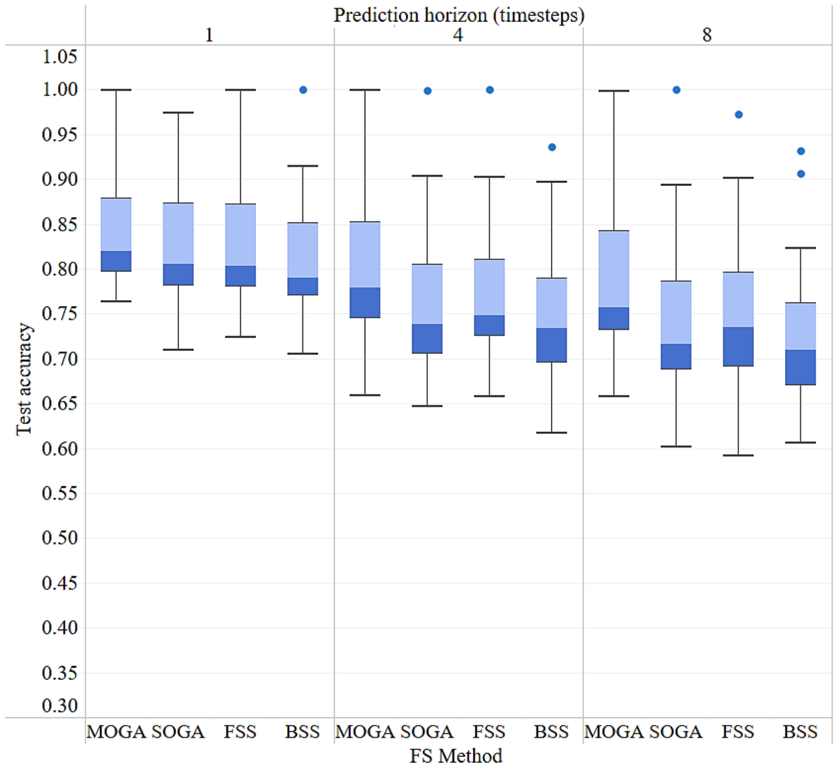


Fig. 6.10. Distribution of the test accuracy based on different FS methods.

The distribution of the mean CV accuracy of different methods is demonstrated in Fig. 6.11. Although the MOGA led to the best median test accuracy among the models, it results in the lowest median train accuracy from 77.34% for the long-term prediction to 84.06% for the short-term prediction. The FSS method gives the second-lowest median accuracy in the range of 79.46%-84.86%. It is followed by the SOGA and BSS methods, respectively. Hence, the methods that lead to the lowest training performance give the best predictions on the test set. It can be concluded that the lower testing performance, given by the SOGA, FSS, and BSS methods, can be partially due to the overfitting issue. Therefore, implementing the second objective (i.e., minimizing the number of features) in the MOGA can be considered a successful approach to enhance the final accuracy of the model by avoiding this issue.

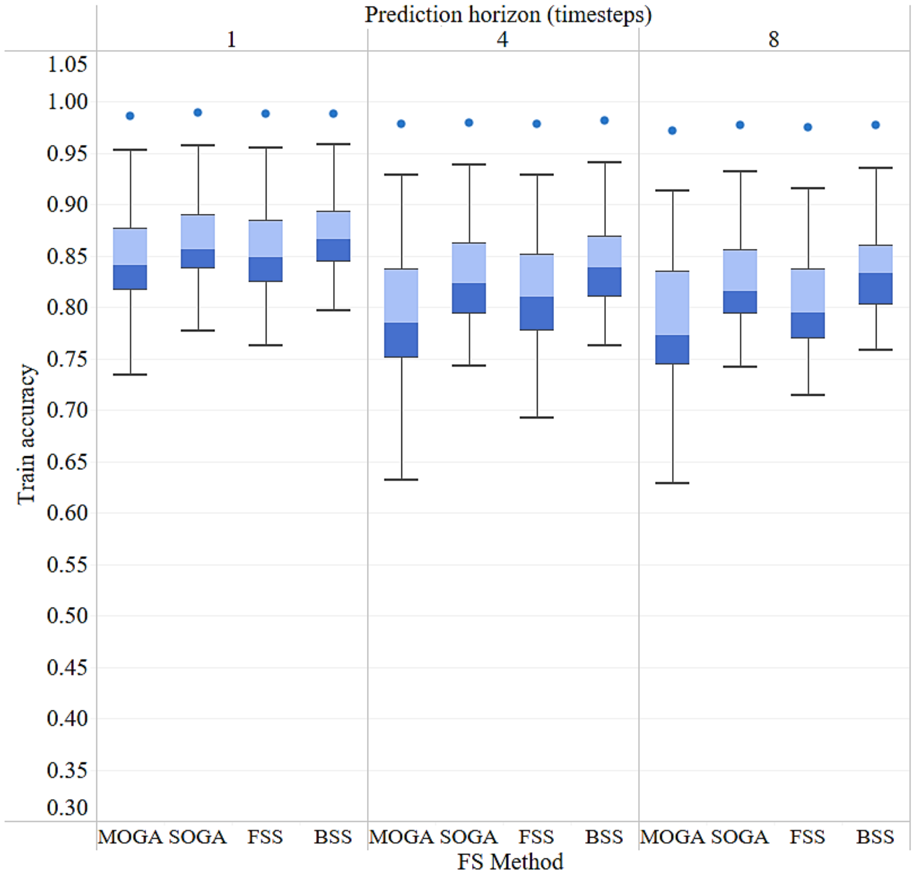


Fig. 6.11. Distribution of the mean CV accuracy based on different FS methods.

Fig. 6.12 compares the distribution of the number of features selected using the MOGA and the SOGA for all the prediction horizons and apartments. It is noted that the number of features utilized in the FSS and the BSS is not demonstrated because these methods work based on the predefined number of features assumed as 20. As can be seen, there is a significant distinction between the number of features selected using both methods; the MOGA leads to a median feature number of as low as three, while the SOGA requires a median of 18 features, which is close to the maximum limit.

Overall, regarding both the number of features and test accuracy, the MOGA provides superior performance in comparison with other methods. Hence, the rest of the section is devoted to elaborating the results obtained by the MOGA to find out which features result in the best accuracy and are essential in developing future occupancy prediction models.

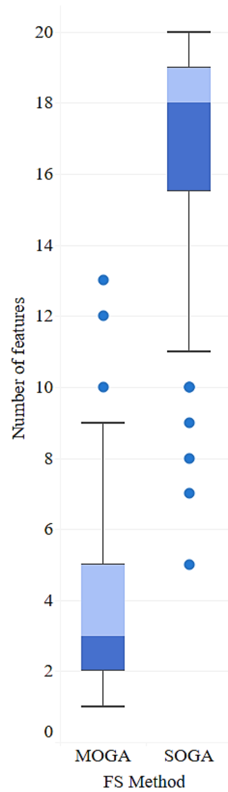


Fig. 6.12. Distribution of the number of features selected using the MOGA and SOGA.

6.4.3. Optimal feature types

Fig. 6.13 and Fig. 6.14 respectively demonstrate how frequently each feature is selected based on the MA and TOPSIS approaches. In these figures, [t-3, t] indicates that the corresponding variables give information at a time step in the range of t-3 to t, in which t is the current time step. Similarly, [t-11, t-8] and [t-7, t-4] demonstrate the variables recorded at the determined time ranges. For instance, Occupancy [t-3, t] refers to an occupancy state at either the current time step t or earlier timesteps of t-1, t-2, or t-3. As each timestep presents information over half an hour, each range demonstrates a 2-hour time period for each variable.

In all the prediction horizons and approaches, the attribute of recent occupancy states (i.e., the occupancy states at the previous four timesteps), denoted by Occupancy [t-3, t], is selected the most for predicting future occupancy, accounting for up to 38.5% of the utilized features. This attribute is followed by the hour of day with a selection frequency in the range of 14%-25%.

The selection of the rest of the features depends on the prediction horizon and the DM approach. In terms of the short-term prediction, recent CO_2 concentration, denoted by CO_2 [t-3, t], follows the time-of-day attribute and accounts for 8.09% and 11.84% of the selected features in the MA and TOPSIS methods, respectively. Based on TOPSIS, the mentioned three features constitute 81.57% of the selected attributes while the others provide a contribution of less than 3% each. On the other hand, in the MA approach, recent lighting and window states can also contribute to the occupancy prediction with 6.62% and 5.88% selection frequency, respectively. Each of the other features is utilized less than 5% for the short-term prediction according to the MA approach.

Regarding the mid- and long-term predictions, vacancy duration was the third most-used variable, with a selection frequency in the range of 7.8%-17.74%. Following the vacancy duration, the information about the older occupancy states, namely occupancy [t-7, t-4], occupancy [t-11, t-8], and occupancy duration, provided the highest contribution. In the mid-term prediction, the mentioned features, day of week, hour of day constitute 74.55% and 91.94% of the total features in the MA and TOPSIS approaches, respectively. In the long-term prediction, the holiday and recent lighting states might also improve the prediction performance with a selection frequency of more than 5%. Other features for mid- and long-term predictions were rarely utilized in developing the occupancy models and removing them from the feature sets might not significantly impact the prediction accuracy in most cases.

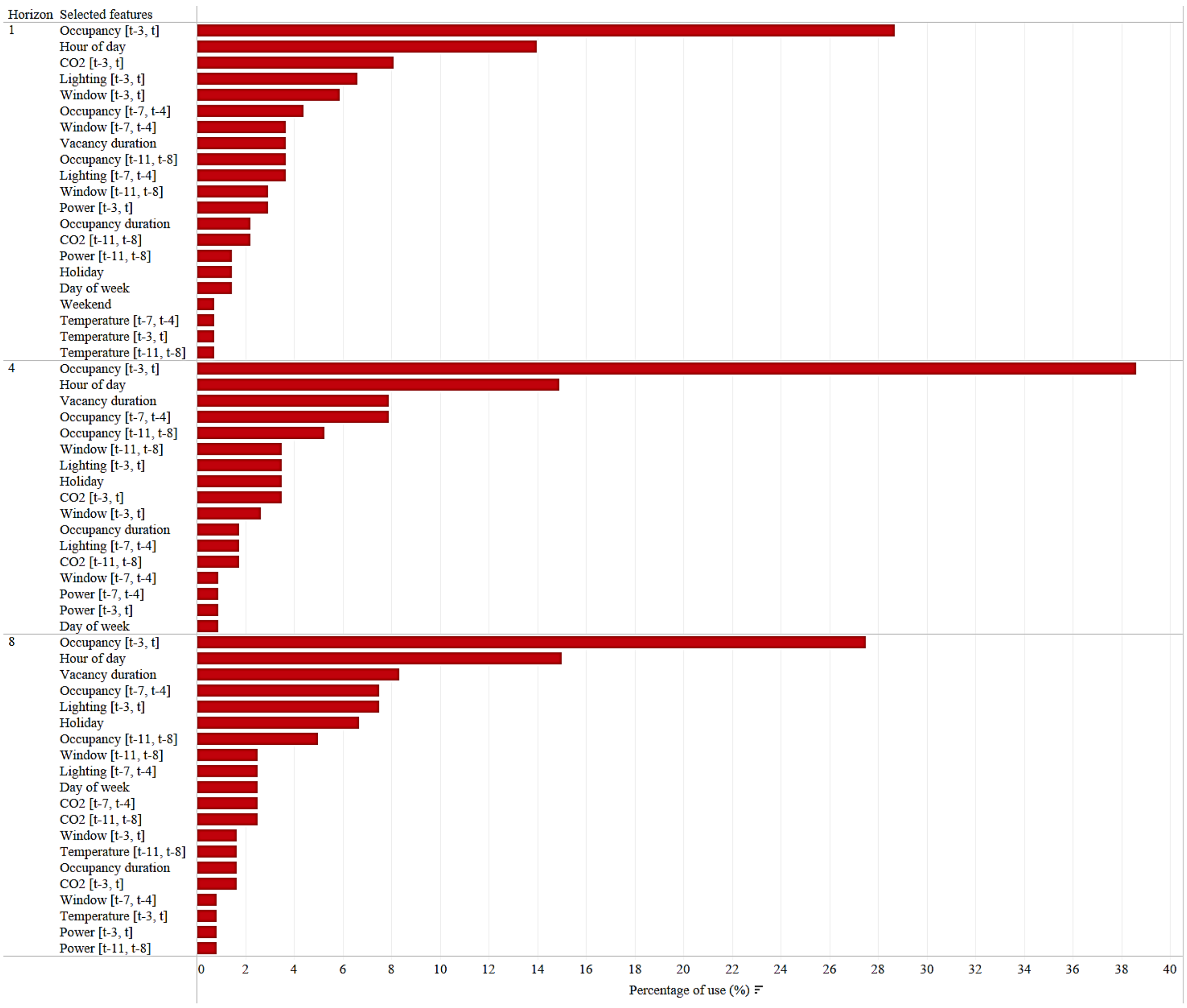


Fig. 6.13. The frequency of the selected features based on the MA approach.

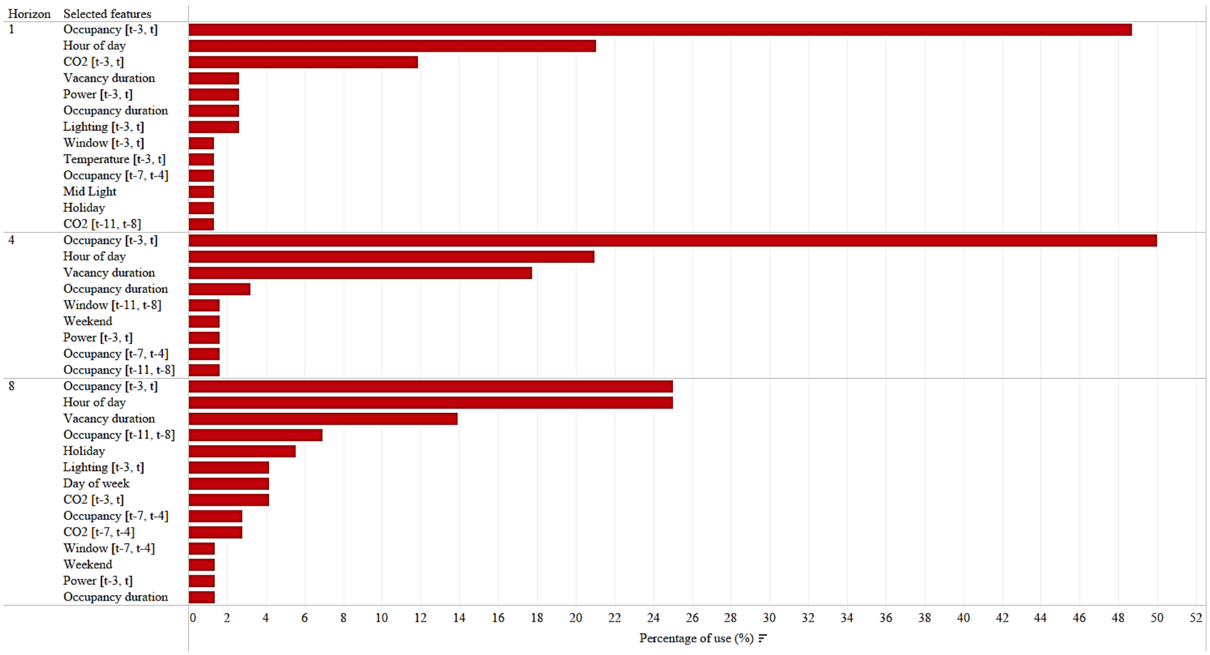


Fig. 6.14. The frequency of the selected features based on the TOPSIS method.

6.4.4. Optimal number of features

The average number of features selected for the occupancy prediction models based on the MA and TOPSIS approaches are respectively demonstrated in Fig. 6.15 and Fig. 6.16. According to the MA approach, almost half of the occupancy prediction models employed less than four features. The second largest group is associated with four to six features, accounting for 34% of the total occupancy prediction models. Only 14% of the models require more than six features for occupancy prediction with the 10-13 feature group accounting for only 3%. The models developed based on the TOPSIS method employed a significantly smaller number of features with no models using more than six features. Almost half of the models implemented only two attributes. The three-feature category, as the second-largest group, makes up 22% of the occupancy models, which is followed by single-feature models at 19%. Only 6% of the models are developed using more than three features.

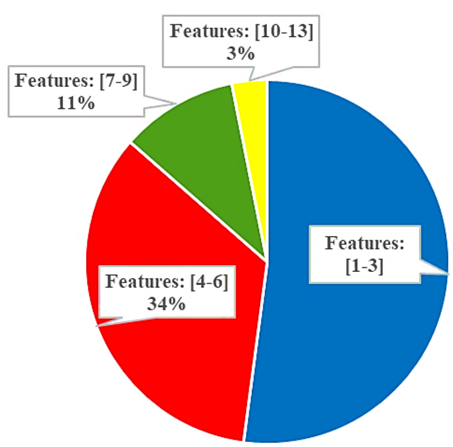


Fig. 6.15. The average number of features selected based on the MA approach.

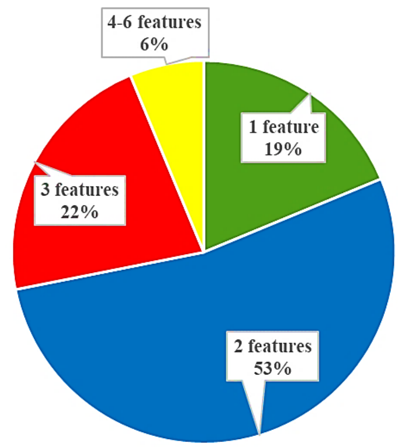


Fig. 6.16. The average number of features selected according to the TOPSIS method.

6.5. Conclusion

In this study, an optimal FS method based on the MOGA is proposed and applied to a database collected from 32 apartments. The MOGA minimizes the number of features utilized in occupancy prediction models and maximizes the prediction accuracy. The aim is to enhance the occupancy prediction performance by optimizing the features employed in the occupancy models. More specifically, this study investigates which features are essential in developing the occupancy prediction models, which features are likely to contribute to the occupancy prediction, how many features lead to the best accuracy in most cases, and which FS methodologies can yield superior performance. The optimal solutions are determined among the Pareto solution set using the MA and TOPSIS approaches. In order to investigate the effectiveness of the proposed MOGA, its performance is compared with that of the SOGA, FSS, and BSS in terms of test accuracy and the number of utilized features. The FS process is performed separately for short-term, mid-term, and long-term predictions to consider the impact of the prediction horizon on the selected features.

The results show that the MOGA outperforms other algorithms, leading to the best accuracy while requiring a substantially smaller number of features in developing the occupancy models. The MOGA yields different results depending on the DM approaches. Employing the MA approach results in up to 0.71% rise in the average accuracy, compared with the TOPSIS method; however, TOPSIS leads to a fewer number of features used in the occupancy models. According to the TOPSIS method, no more than six features are required in developing the occupancy models, while based on the MA approach, up to 13 features can be implemented in some cases. In all the prediction horizons and DM approaches, recent occupancy states and the time of day are shown to be essential in developing the occupancy prediction models. However, the selection of the rest of the optimal features depends on the prediction conditions and the DM methods. The recent CO_2 concentration is the recommended feature for short-term predictions, while for longer horizons, vacancy duration, previous occupancy states, holiday, and lighting states demonstrate potential contributions to the occupancy prediction performance.

It should be noted that the performance of the FS algorithms presented in this study depends on the performance of the machine learning model (i.e., the DT model). Although DT provides benefits such as high computational speed and the ability to capture non-linear relationships between the variables, it is prone to overfitting the database, which can negatively impact the testing performance. Hence, it is suggested to investigate the impact of different machine learning models when analyzing the features in future research. Furthermore, this study considers a limited number of FS methods, namely FSS, BSS, SOGA, and MOGA. There are also other feature selection methodologies such as recursive feature elimination, premutation importance, filter, and embedded FS methods, which can also be considered. It is suggested to investigate more FS methods to compare and find the most effective features in future works. Additionally, there are limitations in terms of the utilized database. Although the database provided extensive information and features, it lacks some variables, such as occupants' activity types and their locations when they are not present at home, which might be helpful for occupancy prediction. It is recommended to collect and utilize a wider range of variables, as candidate features, in the FS process in future works.

Chapter 7: Towards self-learning control of HVAC systems with the consideration of dynamic occupancy patterns: Application of model-free deep reinforcement learning[6]

7.1. Overview

This study proposes a self-learning control system with the ability to learn dynamic changes in occupancy patterns and HVAC lag time. This control system takes advantage of a double deep Q-networks (DDQN) model, as a model-free reinforcement learning algorithm. The aim is to learn an optimal decision-making policy by interacting with the environment without a need for developing building and occupancy models. The control system's performance is evaluated and compared with that of a model-based predictive control (MPC) algorithm, which is assisted by supervised learning models. The results reveal that the proposed method results in superior thermal comfort for occupants by taking prediction uncertainty into account for making control decisions. This ability improves the average temperature deviation and deviation period respectively by 0.24 °C and 7.87% with MPC as the benchmark. However, analyzing the performance of the algorithm shows significant thermal comfort violations during the initial training periods. It causes a 2.8% more deviation period and 0.32 °C higher temperature deviation in the first episode of training compared with the performance of the fully-trained agent.

7.2. Methodology

7.2.1. Reinforcement learning (RL)

7.2.1.1. Markov decision process (MDP)

To develop the RL algorithm to make the sequential control decisions, the problem is first formulated as Markov decision processes (MDP). The MDP framework consists of three key components of *states*, *actions*, and *rewards*. Each state is a representation of the *environment*, which is observed by the *agent* (i.e., the element of the system that makes the control decisions) [218]. In every state, the agent interacts with its environment by making control decisions, also called actions. Taking each action can lead to a transition from the current state to a different state and an instant reward. The agent always tries to improve the actions to maximize the amount of the cumulative reward [20]. Hence, the reward function is utilized to define the ultimate goal of the RL algorithm. In the following sections, the state space, action space, and the reward signal are defined in more detail.

7.2.1.1.1. State-space formulation

In this study, five attributes are employed in the MDP framework to represent the states to the control agent. As summarized in Table 7.1, these features are classified into two categories based on the type of information they provide for the agent: *HVAC lag time* and *Occupancy schedules*. The first category is primarily utilized to help the agent to learn the HVAC system lag time. In

[6] This chapter is based on the following publication: M. Esrafilian-Najafabadi, F. Haghighat, Towards self-learning control of HVAC systems with the consideration of dynamic occupancy patterns: Application of model-free deep reinforcement learning, (2022) (Submitted)

other words, the agent is expected to estimate how long it takes for the HVAC system to warm up or cool down the indoor environment. The first feature in this category is the outdoor temperature, which is selected because of the dependency of HVAC performance on outdoor weather conditions. It is worth noting that there are also other factors such as relative humidity, solar radiation, wind speed, and rainfall, which can be utilized for this purpose. Nevertheless, as mentioned in [14], among these variables, the outdoor temperature has the highest impact on the building energy prediction, and for the sake of not over-complicating the control problem, other features are neglected in this study. The HVAC lag time also depends on the indoor temperature, which is also utilized to represent the agents. For example, it naturally takes more time to warm up a building when the temperature difference between the indoor and desired temperature is larger.

As will be discussed in Section 7.2.1.2, the agent always strives to maximize the expected cumulative rewards. To this end, the algorithm tries to estimate the future rewards associated with different actions. As the goal of the control system is to maximize energy saving and thermal comfort, occupancy information might enable the agent to learn the occupancy patterns and adjust the current temperature to the future probabilities of occupancy. In this way, the agent can pre-heat or pre-cool to prepare the zone before occupants' arrival time to save energy while avoiding thermal discomfort. The variables placed in the second category are expected to help the agent to learn and forecast the occupancy patterns. Various attributes, such as day of week, time of day, type of day (i.e., working days or weekends), and carbon dioxide (CO_2) concentration, might be useful for this purpose. However, two essential features, namely occupancy states in the last two hours and time of day, can provide an acceptable level of accuracy in most cases [219]. Therefore, these variables are selected to provide the agent with occupancy information in this study.

Due to the complex nature of occupant behavior, which includes occupancy patterns, a great deal of uncertainty exists in future occupancy patterns [220]. Such uncertainty and prediction errors can negatively impact the control system performance [17]. To investigate such effects, a scenario with perfect occupancy prediction (i.e., occupancy prediction with no errors) is also considered. In this scenario, the control system accesses the *actual future occupancy states* as a state feature. In other words, the agent receives the next following 2 hours of the occupancy data in advance at the current timestep of interest. Consequently, the agent does not receive other occupancy-related features (i.e., recent occupancy states and time of day) in this case.

Table 7.1.
The features utilized to represent the state-space for the agent.

Feature	Unit
HVAC lag time	
Outdoor temperature	°C
Indoor operative temperature	°C
Occupancy schedules	
Recent occupancy states	-
Time of day	Hour
Actual future occupancy states	-

7.2.1.1.2. Action-space formulation

The task of the control system is to determine a setpoint temperature at every control timestep. Although temperature can be defined as a continuous variable, the action space, A, is discretized for the sake of avoiding complexities in the control framework. Three indoor temperature settings of deep setback at 15 °C, conservative setback at 19 °C, and setpoint temperature at 22 °C are considered as the action set. It should be noted that the thermal inertia of the building makes it possible for the agent to reach the intermediate temperature values via switching between these settings. Thus, introducing more actions does not necessarily lead to performance improvement [18]. The action space is defined based on the described setpoint/setback temperatures as follows:

$$A_t = [15, 19, 22]\ °\mathrm{C} \tag{20}$$

7.2.1.1.3. Reward function

The control system's function is to accomplish two main tasks: to minimize energy consumption and thermal comfort. These objectives need to be reflected in the reward function so the agent can find the optimal actions accordingly. It is a common practice to represent the former objective as a function of the amount of heating energy supplied to the building, denoted by Q_{heat} [221]. Regarding the latter, the thermal discomfort can be defined as the difference between the indoor operative temperature, T_{op}, and the desired setpoint, T_{sp}, when occupants are present. The operative temperature can be defined as the average of *mean radiant temperature* and the air temperature [222]. This variable quantifies the impacts of both radiant and convective heat transfers from the occupants' bodies to the environment, which are two main components for representing the thermal comfort of occupants [182]. The following reward function can be defined based on both energy and thermal comfort terms [18,223]:

$$R_t = -\beta(Q_{heat}) - (1-\beta).Occ.\left(T_{op} - T_{sp}\right)^n \tag{21}$$

where β weights each objective while n weights the magnitude of temperature deviation. The weights can be determined based on the designer's experience and trial and error [224]. *Occ* indicates the binary occupancy state (i.e., 0 and 1 respectively showing the absence and presence of occupants). As suggested in [225], normalizing the reward function to put the energy and thermal comfort on the same scale might help to learn the optimal *policy* (i.e., the mapping between the states and the selection probability of the actions [18]). Hence, the following normalized reward function is also considered in this study:

$$R_t = -\beta\left(\frac{Q_{heat}}{Q_{heat,max}}\right) - (1-\beta).Occ.\left(\frac{T_{op} - T_{sp}}{\Delta T_{max}}\right)^n \tag{22}$$

in which $Q_{heat,max}$ and ΔT_{max} represent the maximum values of heating energy and temperature difference, respectively. The smart design of the reward function is a vital task in RL [226]. Thus, the parameters of the reward function, namely n and β, and the type of the reward function (i.e., normalized or original form of the reward function) are carefully selected through trial and error.

7.2.1.2. Q-learning algorithm

Q-learning has been a popular RL algorithm proposed by Watkins [227], which can be utilized to find the optimal solutions to the MDPs [22,228]. It is a model-free algorithm, in which the agent learns directly from the experience through trial and error. The agent tries to find the actions that lead to the highest *Q-value* in each state. The *Q*-value, also called the *state-action value* is defined as the expected reward for taking a specific action *a* given a state *s*, which can be formulated as follows [20]:

$$Q(s,a) = \mathrm{E}(r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots \mid s_t = s, a_t = a) \tag{23}$$

where $Q(s,a)$ is the *Q*-value, *E* is the expected return, *r* is the instant reward at each timestep, and γ is the *discount factor*. The *Q*-values are randomly initiated and are updated at each timestep based on the new experience gained by the agent using the following equation [229]:

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a)) \tag{24}$$

in which α is the learning rate, s' is the next state, and a' is the next action. The learning rate can be selected in the range of 0 – 1; a learning rate that equals to **1** leads the model to replace the old *Q*-values with the new ones, while a learning rate that equals to **0** results in an algorithm that practically does not update the old values. The value of the learning rate is considered a hyperparameter of the RL algorithm and is tuned through trial and error.

7.2.1.3. Deep Q-network

Q-learning techniques traditionally make use of so-called Q-tables, in which the values of the Q-function for each action and state are recorded [230]. However, this methodology becomes infeasible for cases in which a large number of actions or states are implemented [225]. As a promising solution to this issue, DQN was proposed [231]. The DQN replaces the use of conventional Q-tables with a deep neural network (DNN) as a well-practiced function approximator. In this approach, the number of neurons in the input layer equals the number of features that represent the states, defined in Section 7.2.1.1.1. The number of neurons in the output layer equals the number of actions, defined in Section 7.2.1.1.2. The DNN structure tries to learn the relationships between the Q-values and each action and state. The weights of this Q-network, θ, are updated via optimizing the following loss function, *L*, as follows [231]:

$$L(\theta) = \left[Y^{DQN} - Q(s,a;\theta) \right]^2 \tag{25}$$

where Y^{DQN} is the target value that can be defined as:

$$Y^{DQN} = r + \gamma \max_{a'} Q(s',a';\theta^-) \tag{26}$$

in which θ^- represents the weights of the *target network*. The target network is a separate DNN that is utilized along with the Q-network to prevent the learning process from possible divergences [225]. The target network utilizes the same structure as the Q-network. It was initially proposed that the weights of the target network are updated cyclically as a copy of the weights of the Q-network, which is known as the *hard update* method [231,232]. However, another method called *soft update* was later introduced and integrated with the developed libraries [233,234]. This method updates the weights via interpolation with a fixed ratio in each step. The update method is selected in the hyperparameter tuning process, which will be discussed in more detail in Section 7.2.1.6. Table 7.2 summarizes the structural parameter of the implemented networks.

Table 7.2.
The structural parameters of the function approximator.

Parameter	Value
Number of hidden layers	3
Number of neurons	(64, 32, 64)
Optimizer	Adam [235]
Activation function in the hidden layers	Rectified Linear Unit (ReLU) [168]

7.2.1.4. Double deep Q-networks (DQN)

In practice, DQN is prone to overestimating the Q-values, which might negatively affect the performance of the RL algorithm [229]. Hasalt et al. [236] stated that utilizing the same Q-network for both selection and evaluation of an action is the main reason for the overestimation problem in conventional DQN. They proposed double deep Q-networks (DDQN) to deal with this issue and demonstrated its ability to improve the optimal policy found by the RL algorithms. DDQN decouples the processes of action evaluation and selection by redefining the target value as follows:

$$Y^{DDQN} = r + \gamma Q(s', \arg\max_{a'} Q(s', a'; \theta), \theta^{-}) \tag{27}$$

Based on this relationship, first, the DDQN algorithm selects the optimal action based on the Q-network. Then, instead of using the same network to evaluate the Q-value, the target network is employed.

7.2.1.5. Policy

One of the key points in developing RL algorithms is to define an appropriate policy to provide a trade-off between the tasks of *exploration* and *exploitation*. More specifically, if a *greedy* policy (i.e., a policy that always exploits to maximize the expected rewards) is utilized in all states, the agent might never try all the possible actions. Consequently, it increases the likelihood of converging to a suboptimal solution. To solve this issue, the agent should be able to try many different possibilities while learning the Q-values. The balance between exploration and exploitation can be obtained using the *epsilon-greedy* policy [27]. According to this method, the agent selects a random action with a probability of epsilon (i.e., the rate of exploration) while exploiting with the probability of (**1** - epsilon).

It should be noted that the epsilon-greedy method assigns equal probabilities to each action regardless of their corresponding Q-values. In other words, the agent does not differentiate between different actions when it explores, which can cause the risk of having too large thermal discomfort for occupants. For example, when the agent is searching among possible actions and interacts with the environment, it might randomly select a deep setback temperature when occupants are present even though it might be associated with a relatively small Q-value. To address this limitation, this study also considers the Boltzmann exploration policy as an alternative approach. In this method, each action is assigned with a probability of being selected based on their Q-values using the following equation [237]:

$$P(a \mid s) = \frac{\exp(\frac{Q(s,a)}{T})}{\sum_i \exp(\frac{Q(s,i)}{T})} \tag{28}$$

where T is the *temperature parameter* that is used to change the exploration rate. The higher values of the temperature parameter leads the model to search more. Based on this equation, the higher the Q-values of a certain action, the more probable that the action is selected by the agent. As a result, this policy tends to become greedy when there are large differences between probabilities, while it does more exploration when the probability distribution is almost uniform. This strategy is valuable for decreasing the risks associated with exploring new actions by giving more chances to those with higher Q-values.

It should be noted that in the initial steps of the learning process, the estimated Q-values can be unreliable, and consequently, more explorations are required for both epsilon-greedy and Boltzmann methods. However, as the learning process proceeds and more trustable Q-values are obtained, the agent can exploit more to minimize the risk of thermal comfort violation and energy waste. Hence, the temperature and epsilon parameters are initiated with higher values of **1** (i.e., higher explorations), and while the learning process proceeds, they linearly decrease to a minimum value. This minimum value is important to ensure that the agent always tries to improve its actions and adapts to any changes to the occupancy patterns. In this study, the policy type (i.e., epsilon-greedy and Boltzmann), the decay duration (i.e., the number of episodes in which the search coefficients are linearly decreasing), and the minimum values of epsilon and T are considered as hyperparameters that need to be tuned.

7.2.1.6. Hyperparameter tuning

The hyperparameters of the DDQN and the assigned search ranges are summarized in Table 7.3. *Buffer size* indicates the maximum number of the recent data elements that are stored in memory to be utilized for training the agent. *Target model update* shows how frequently and based on what update rule the target model is updated. The values higher than **1** indicates the number of steps after which the weights of the target network are replaced by a copy of the Q-network weights based on a hard update. On the other hand, the values less than **1** refer to the *soft update* method, in which the parameter values demonstrate the interpolation coefficient [232]. *Learning rate* controls how fast the gradient descent adjusts the weights of the Q-network to the target values. While too large values can cause divergence in the learning process, small values can increase the training time [225]. *Discount factor* determines the weights that the agent gives to the distant future rewards, compared with immediate rewards. A discount factor equal to **0** results in an agent that only consider the immediate rewards, ignoring the effects of the current actions on the following steps. In contrast, a discount factor equal to **1** leads the agent to equally weight all the future rewards for making its current control action [238]. As discussed in Section 7.2.1.5, the policy, decay duration, and the minimum search coefficient are also considered hyperparameters and are determined in the tuning process. Those hyperparameters leading to the best accumulative rewards based on trial and error are considered the final parameters for the network.

Table 7.3.
Hyperparameters of the DDQN model with the associated search ranges for the tuning process.

Hyperparameters	Ranges
Buffer size	[2,000-20,000]
Target model update	[10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 0, 1, 10, 100]
Learning rate	[10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}]
Discount factor	[0.1-0.9]
Policy	[E-greedy, Boltzmann]
Decay duration (episodes)	[5-10]
Minimum search coefficient	[0.05, 0.1, 0.15, 0.2]

7.2.1.7. Employment of DDQN

The working diagram of the DDQN algorithm is demonstrated in Fig. 7.1. The learning process begins with the environment sending the initial state to the control agent. In other words, in step 1, the current state at time step *t*, *S(t)*, which consists of occupancy states over the last four timesteps, *Occ [t-4, t]*, hour of day, *H(t)*, outdoor temperature, $T_{out}(t)$, and operative temperature, $T_{op}(t)$, are given to the control agent. This information is directly utilized in the Q-network to estimate the Q-values associated with each action. In step 2, according to the policy and the estimated Q-values, the action at time step *t*, A_t, (i.e., the setpoint temperature for the following time step, $T_{sp}(t+1)$, is selected and is applied to the environment. Along with $T_{sp}(t+1)$ *and* $T_{op}(t)$, the values of solar radiation, *SR(t+1)*, and outdoor temperature in the next timestep, $T_{out}(t+1)$, are passed into the building model to obtain the associated energy consumption, *EC(t+1)* and indoor temperature, $T_{op}(t+1)$, at time step *t+1*. Now, step 3 calculates the reward, *R(t+1)*, that is associated with the taken action using the reward function. Then, the feedback of the environment involving the state, action, and reward values can be given to the agent (step 4). The control algorithm, first, stores the feedback in the reply buffer, and then, passes a batch of recent experiences to the networks. The aim is to calculate the loss function and update the weights of the Q-networks via the *gradient descent* method. Depending on the update approach (i.e., soft or hard update methods) the weights of the target network are regularly updated. The learning and control processes are continuously repeated until the end of the training episode.

A heating season starting from the first of November to the end of March is utilized for training the DDQN algorithm. The agent is trained repeatedly on the database with each iteration defined as a training episode. After 20 episodes of training, the agent is applied to unseen occupancy data over a heating season of the same length to evaluate its performance using a greedy policy. More details about the databases are provided in Section 7.3.

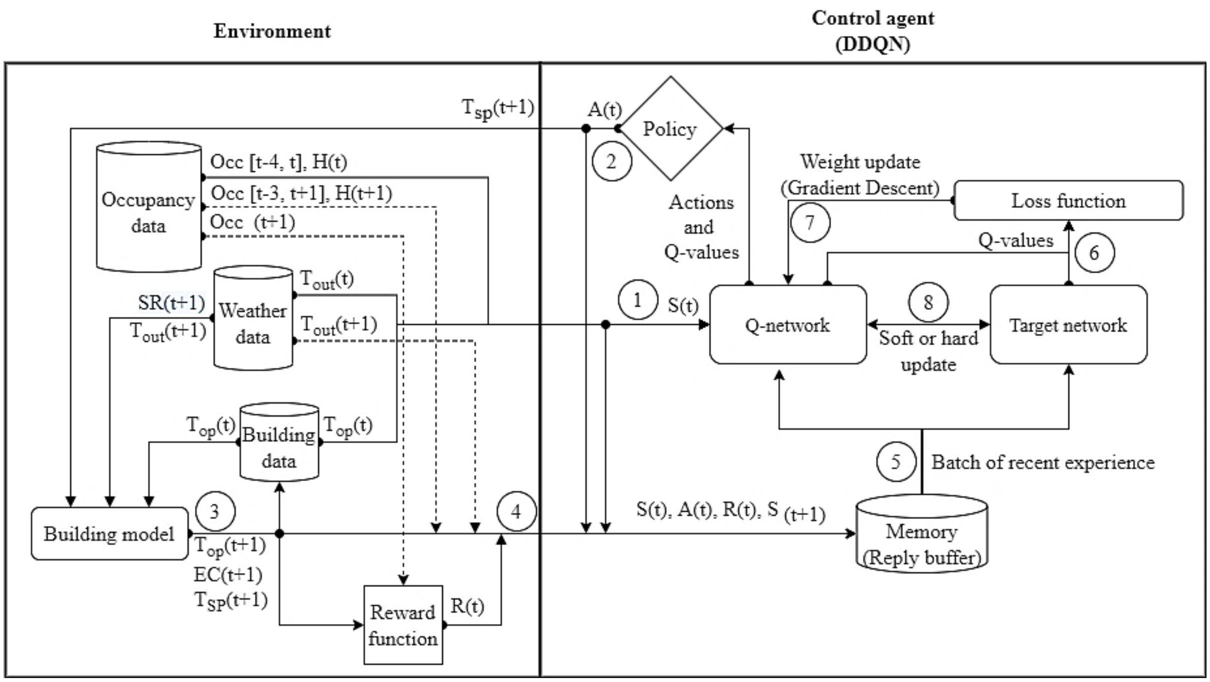


Fig. 7.1. The working diagram of the implemented DDQN algorithm.

Fig. 7.2 demonstrates a general overview of the interactions between the software packages and Python libraries utilized to develop the DDQN agent. In the first step, a virtual testbed is constructed by developing a white-box model using the EnergyPlus [129] engine as described in Section 7.3.2. The virtual building provides an interactive environment for the control system, which is later utilized to evaluate the system performance. As will be discussed in Section 0, the simulation process is accelerated by developing a black-box building model that replicates the behavior of the virtual building. The building models are developed using Keras [125] and scikit-learn [103] libraries in Python. Moreover, the Keras library is also utilized to develop the function approximator of the DDQN.

The interactions between the building environment and the control agent, which is developed using Keras-RL [239], are enabled through the use of OpenAi Gym toolkit in Python [240]. The OpenAi gym includes a Python class, in which three main functions, namely *Initialization*, *Step*, and *Reset*, are defined and utilized to enable the agent to directly interact with the environment. In the *Initialization* function, the action space, observation space, the initial state of the building, and the length of each episode (i.e., the five-month heating period) are defined. These variables are always initialized in the first step of the learning process. The interactions between the agent and environment are performed through the *Step* function. More specifically, in each control step, the agent sends the selected action to OpenAi Gym using this function, and then, the next state and instant rewards are sent back to the control system. The rewards are calculated as a function of the occupancy and weather data, stored in a MySQL database [121], and the outputs of the building model in each step. Next, the agent makes the next decision and updates the Q-values accordingly. This process continues and the Q-values are updated until the first episode ends. At this point, the *Reset* function is utilized to reset the episode and begin the learning process from the initial state. After the last episode, the learning process is completed, and the agent behaves based on the optimal policy.

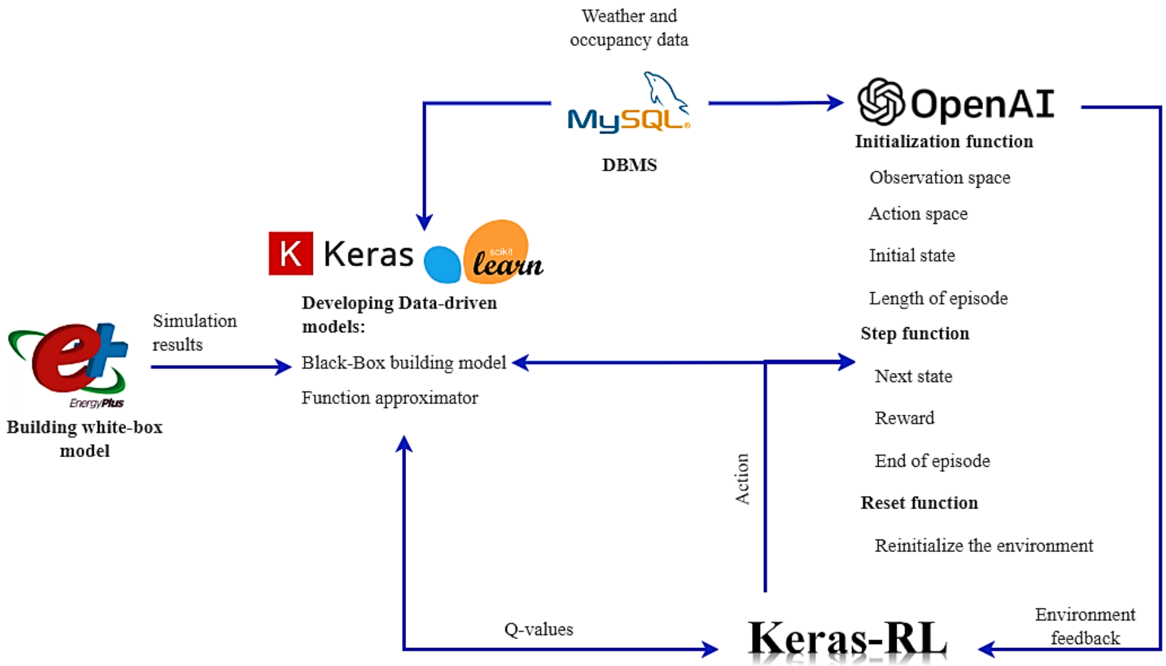


Fig. 7.2. The general overview of the interactions between libraries and software packages utilized to develop the DDQN controller.

7.2.2. Model-based predictive control

The general working diagram of the MPC is demonstrated in Fig. 7.3. Generally, the control framework consists of two main parts: 1. The prediction models (i.e., the occupancy and building models), and 2. The optimization algorithm. The prediction models are trained based on the collected historical data, which will be discussed in detail in Section 7.2.2.2. Regarding the second part of the system, the optimization algorithm takes advantage of the prediction models to estimate the future disturbances (i.e., occupancy patterns) for the control decisions. Using the weather data as another input, the optimization algorithm searches for the optimal control actions that can be employed to minimize a cost function, which will be discussed in Section 7.2.2.3. When the control decisions are made in each timestep, it is applied to the building. The feedback is measured continuously and is stored in the related databases that can be utilized later to improve the quality of the models of the building and occupancy. Moreover, the information regarding the indoor operative temperature is always employed to adjust the HVAC operations in the following timesteps.

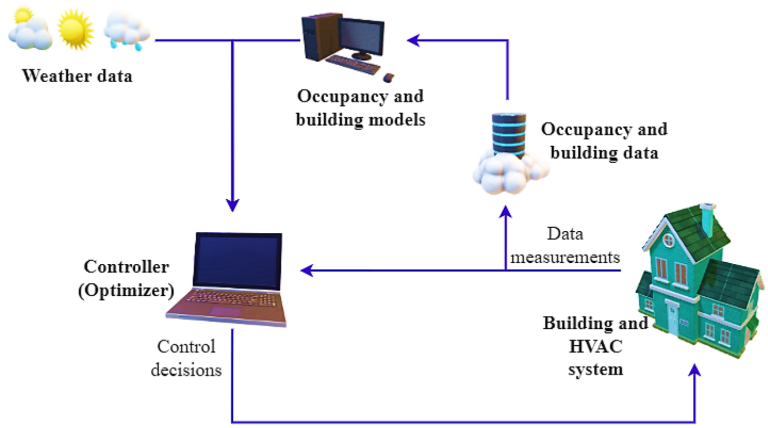


Fig. 7.3. The overview of the workflow of the MPC.

7.2.2.2. Building and occupancy prediction models

The MPC algorithm developed in this study is leveraged with an occupancy model so that it can adjust the indoor operative environment to the future occupancy patterns. More specifically, the system can determine the depth of setback as a function of the estimated duration of vacancy periods. For example, the control system might opt for a deep setback when the zone is expected to remain vacant for a long time. The controller might turn the HVAC system on before occupants' arrival to provide thermal comfort for occupants upon arrival. Many studies have made attempts to capture the complexities in the occupancy patterns and predict the future occupancy states in buildings; from simple Naïve prediction models to complex deep learning algorithms have been developed for this purpose [76,82–84,86,94,95]. As reported in an earlier comparative study [192], a kNN model can outperform other alternatives in most cases by providing a higher average of thermal comfort and energy saving. Thus, a kNN model with a k value of 15, as suggested in [192], is developed to be integrated with the MPC algorithm.

The occupancy model can receive a variety of different attributes, such as time of day, day of week, holidays, weekends, and CO_2 concentrations, as input variables [114]. However, it was demonstrated in [219] that among different predictor variables, previous occupancy states in the last two hours and the time of day are essential features, which are adequate for occupancy prediction in many cases. Hence, these variables are selected as the input features of the kNN model. It is assumed that the future occupancy states during a two-hour period are adequate for decision-making in MPC because of the generally short-term HVAC lag time. It is worth noting that a separate kNN model is developed to predict the occupancy states in each future control timestep. For example, given a 30-min control timestep, four kNN models need to be developed to provide the MPC with future occupancy states in two hours.

In addition to the occupancy models, the MPC algorithm needs an accurate building model to evaluate the impacts of the control decisions on the system performance over its prediction horizon [241]. As developing building models are often accompanied by prediction errors (i.e., the model is not 100% accurate), MPC might encounter uncertainty about future decision making. Such potential inaccuracy might negatively impact the MPC performance. This study eliminates such effects by utilizing the same building model for both making control decisions and evaluating the performance. In other words, a perfect building model is assumed in the MPC algorithm. Hence, the performance of the MPC is overestimated in this study. The building modeling procedure and the description of the case study building are provided in Section 0 and Section 7.3, respectively.

7.2.2.3. Optimization algorithm

MPC makes the control decisions by solving an optimization problem at every time step [242]. It seeks the same goal as the DDQN agent, i.e., to maximize thermal comfort while minimizing energy consumption. Therefore, the same reward function, described in Section 7.2.1.1.3, can also be utilized to define the objective function of the MPC algorithm. In this way, the MPC aims to find the optimal setpoint temperature that can maximize the cumulative rewards over the prediction horizon, H. The objective function of the MPC can be formulated as follows:

$$\max \sum_{h=1}^{H} R_{t+h} \tag{29}$$

where R_{t+h} represents the instant reward at each future time step, estimated using Eq. 17 or Eq. 18. The MPC algorithm searches for the optimal operative temperature (i.e., the *manipulated variable*) over the prediction horizon, that can maximize the objective function. Although the manipulated variable is optimized over the prediction horizon, only the first optimal variable (i.e., the setpoint temperature at the next timestep) is applied to the building. Then, the whole optimization process is repeated in the next sampled timestep to regulate the following indoor temperature.

7.2.2.4. MPC employment

The working diagram of the MPC is demonstrated in Fig. 7.4. At each timestep *t*, the process begins with an optimization algorithm. For this purpose, a genetic algorithm (GA) is employed in this study because of its potential ability to reach the global optimum and its promising performance in different applications [124,138]. The hyperparameters of the GA are summarized in Table 7.4. The process begins by initializing the first *population* with a size of *N* in the GA. A population consists of a set of *chromosomes* (also called individuals), where each chromosome represents a candidate solution to the optimization problem (i.e., the indoor operative temperatures over the prediction horizon). Every chromosome contains H *genes* that represent the setpoint at each future timestep. The prediction horizon of the MPC is considered as four timesteps because of the short-term nature of the occupancy-based control systems.

In the next step, the fitness of the individuals needs to be evaluated based on the objective function. To this end, each solution ($T_{op}(t+1)$, …, $T_{op}(t+H)$) along with $SR(t)$, $T_{out}(t)$, and $T_{op}(t)$ are utilized in the building model to provide an estimation of the energy consumption and indoor temperature over the prediction horizon. For the sake of not overcomplicating the control process, it is assumed that the changes in the weather condition are negligible over the course of the prediction horizon, and therefore, no future weather forecast is required in the control framework. It is also worth noting that as well as the current value of the operative temperature recorded by the sensors in the building, its future values (i.e., $T_{op}(t+1)$, …, $T_{op}(t+H-1)$) are also required as the inputs of the building model. As the future values are not available at timestep *t*, the estimated output of the building model is also utilized as the model input. For example, $T_{op}(t)$ is first utilized to estimate the future operative temperature, $T_{op}(t+1)$. Next, this estimated value is then employed in the building model for calculating the following value, $T_{op}(t+2)$.

Table 7.4.
The parameters utilized for developing the GA.

Parameters	MPC optimizer
Population size	30
Mutation rate	0.05
Selection method	Tournament
Stopping criteria	Not improvement in the solution

In order to calculate the fitness of the individuals, it is also essential to estimate future occupancy patterns to estimate the future values of *Occ* as an input for the reward function. The occupancy model receives the recent occupancy states (i.e., *Occ(t-4)*, …, *Occ(t)*) and time of day as input and yields predictions of future occupancy states over the prediction horizon. The same dataset utilized to train the DDQN network is also employed to develop the kNN model for occupancy prediction. In step 4, the reward function is calculated, as defined in Eq. 25, and the fitness of each individual is achieved.

After calculating their fitness, the individuals are ranked and the best half of them are selected to proceed to the next step. A *crossover* operation is performed on the selected individuals to repopulate the whole set via producing the *children*. The children are randomly undergone *mutation* to add more diversity to the solution set to prevent getting stuck in the local optima. This process is iteratively repeated to improve the population until the stopping criterion is met. In the last iteration at timestep *t*, $T_{sp}(t+1)$ is selected and applied to the building. The process repeats at the next timestep to make the control decisions.

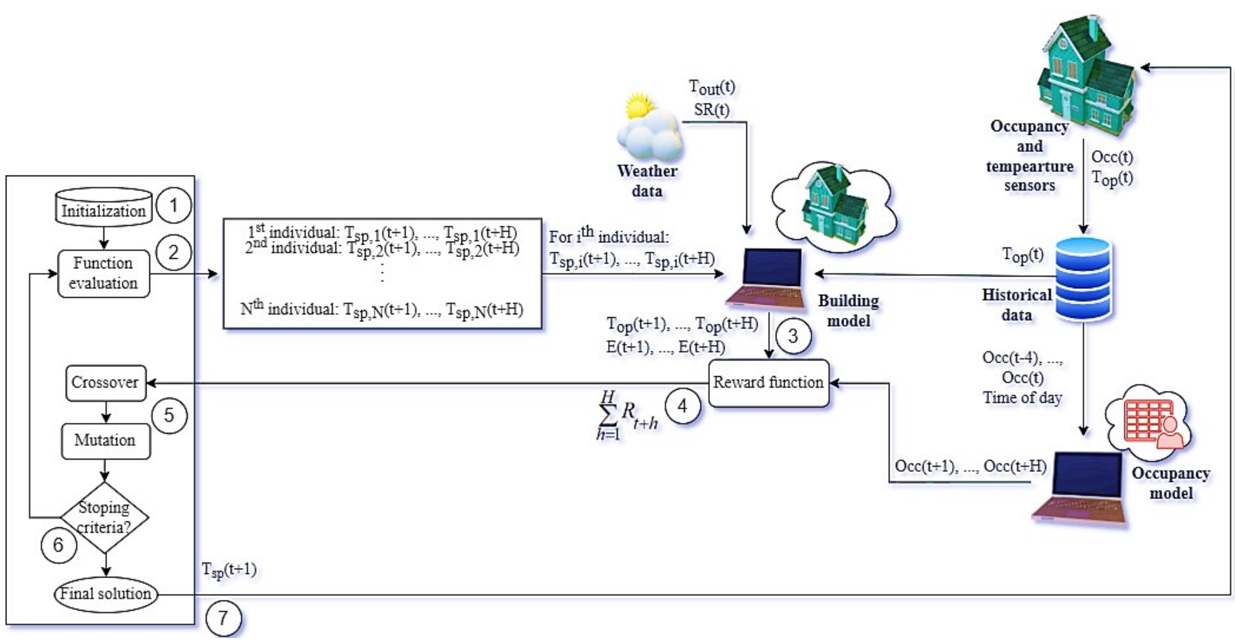


Fig. 7.4. The schematic diagram of the MPC implemented in this study.

7.2.3. Building model

The building model is a key component for the RL and MPC control systems. To briefly recall, the DDQN control requires a building model to enable the interactions between the agent and the environment. The MPC algorithm needs the building model to estimate the energy consumption and indoor temperature associated with the control decisions. Although white-box models often yield high accuracy by solving detailed mathematical equations, they are computationally inefficient for use in such control applications [124]. More specifically, the MPC and DDQN are iterative techniques, which are computationally expensive, and thus, it is essential to select a model that can be fast and reliable. To this end, this study utilizes black-box models to replicate the behavior of the virtual building, described in Section 7.3.2, by estimating future energy consumption and indoor operative temperature. Various machine learning models can be employed for this purpose [111]. Among the possible models, some of the most popular ones, namely random forest (RF), linear regression (LR), kNN, and artificial neural networks (ANN) are developed, and their performance is compared in terms of execution time and mean absolute error (MAE).

The hyperparameters of different algorithms are determined through trial and error and are summarized in Table 7.5. RF, LR, and kNN models are developed using Scikit-learn [166] library and the ANN model is developed using Keras [125] with TensorFlow backend [243] in the Python environment. In order to generate the required database to develop the black-box model, 10 separate simulations are performed using the white-box model described in Section 7.3.2. Each simulation is run using a different setpoint schedule, generated by assigning a random operative temperature in the range of 15-23 °C to each hour of day. The simulations are run based on 10-min intervals, with each simulation providing 21,745 data records for the heating season. the granularity of the created database is transformed into 30-min intervals to be consistent with the length of control timesteps. The entire database is split into two subsets of train and the test with a ratio of 80:20 based on random sampling. The models are trained on the train set and their performance is measured on the test dataset in terms of MAE and the execution time. The execution time is calculated as the time the model takes to predict future occupancy states for the entire test dataset.

Table 7.5.
Hyperparameters of the ML models employed as building models.

Hyperparameters	Values
RF	
Number of trees	10
kNN	
Number of neighbors	10
ANN	
Number of hidden layers	2
Number of neurons in each hidden layer	1st layer: 64 2nd layer: 128
Learning rate	0.005
Optimizer	Adam
Activation function	Hidden layers: ReLU Output layer: Linear function
Regularization method	Early stopping with a *patience* size of 2,000
Number of epochs	20,000
Batch size	5,000

7.2.4. Performance evaluation metrics

The thermal comfort of occupants is measured from two different viewpoints: 1) average temperature deviation, $\bar{T}_{dev}$, and 2) deviation period, P_{dev}. The former criterion is defined as the average deviation of the indoor operative temperature from the desired setpoint when occupants are present at home and can be formulated as a function of temperature deviation, T_{dev}, as follows:

$$T_{dev}(t) = \begin{cases} \left| T_{op}(t) - T_{desired} \right| & Occ(t) = 1 \\ 0 & Otherwise \end{cases} \tag{30}$$

$$\bar{T}_{dev} = \frac{\sum_{t=1}^{L} T_{dev}(t)}{L} \tag{31}$$

where $Occ(t)$ shows the occupancy state at timestep *t* and *L* shows the number of total timesteps in each episode. The deviation period is defined as the proportion of time when occupants encounter a temperature deviation of more than 0.5 ºC. The period of temperature deviation, P_{dev}, can be calculated using the following equation:

$$C_{dev}(t) = \begin{cases} 1 & Occ(t) = 1 \ \& \ T_{dev}(t) > 0.5\,^{\circ}C \\ 0 & Otherwise \end{cases} \tag{32}$$

$$P_{dev} = \frac{\sum_{t=1}^{L} C_{dev}(t)}{L} \times 100 \tag{33}$$

in which $C_{dev}(t)$ is the temperature deviation coefficient. The control performance is also assessed in terms of saving energy. For this purpose, the energy consumption of each control system is compared with that of an always-on controller as the baseline. The always-on controller maintains the temperature at the desired setpoint all the time regardless of the occupancy states. The energy saving, ES, of the RL and MPC is estimated using the following equation:

$$ES = \frac{EC_c - EC_{baseline}}{EC_{baseline}} \times 100, \quad c = RL \text{ and } MPC \tag{34}$$

where $EC_{baseline}$ and EC_c are respectively the energy consumption of the baseline and the controllers.

7.3. Case study

7.3.1. Occupancy database preprocessing

This study employs an occupancy database gathered from 20 residential units in an apartment block during the years 2015 to 2017. A detailed description of the database can be found in [192,219]. In the preprocessing step, the occupancy data is aggregated from 1-min to 30-min intervals to be consistent with the duration of the control timesteps, which can accelerate the simulation process and reduce the impact of occupancy detection errors [156]. Each interval takes **1** if at least one motion is detected over the 30-min interval and otherwise takes **0** to show the vacancy state. There are also missing values in the occupancy data possibly due to sensor or network failure. The missing values are replaced with **1** to ensure that thermal comfort is not sacrificed.

As discussed in Sections 7.2.1.1.1 and 7.2.2.2, two types of predictor variables are implemented to predict future occupancy patterns. First, the hour-of-day attribute is employed to reflect the time of day for the occupancy prediction. This variable can take integer values in the range of 0-23 to indicate each hour of day. In order to utilize this feature in the ML model, it is converted to 23 independent dummy features. In addition to the time of day, the occupancy states from four previous timesteps (i.e., two hours) are also utilized for the occupancy prediction.

7.3.2. Building testbed

To assess the performance of the control algorithms, a virtual detached residential building, also implemented in earlier studies [156,192], is employed as a testbed. The characteristics of the building and the conditions used for the simulation are summarized in Table 7.6. The simulation consists of a five-month period covering the whole heating season from the first of October to the end of March. Electric baseboards are selected as the heating equipment to meet the thermal comfort requirements. Given the efficiency of electric baseboards as 100%, the amount of electricity consumption equals the amount of heating energy associated with each controller. The building envelope and the construction material are defined in SketchUp software [130] via Openstudio plug-in [131]. Then, the developed model is imported into EnergyPlus to complete the simulation.

Table 7.6.
The characteristics of the virtual building and the conditions under which the simulation is performed.

Parameter	Value
Net conditioned area	135 m^2
Number of floors	1
Type	Detached house
Number of bedrooms	3
Season	Winter (Heating season)
Weather data	Very cold climate [129]
Heating equipment	Electric baseboards
Efficiency of heating device	100%
Building material	ASHRAE standard [128]
Design software	SketchUp and Openstudio [130,131]
Simulation engine	EnergyPlus [178]
Simulation timestep	10 minutes

7.4. Results and discussion

7.4.1. Building model selection

To evaluate the control performance, the best building model needs to be determined among the candidate ML algorithms. Fig. 7.5 demonstrates the performance of the ML algorithms in terms of MAE and computational time (i.e., the time it takes to perform the prediction task on the test dataset). The ANN model provides the most accurate prediction with an MAE of 0.18 kWh and 0.21 °C respectively for energy and temperature estimations. However, the high accuracy is obtained at the expense of computational time, taking 0.23 seconds to perform the prediction task. The RF model closely follows the performance of the ANN model by providing an MAE of 0.19 kWh and 0.22 °C respectively for the energy and temperature estimation while providing a substantially faster computation with 0.05 seconds execution time.

It should be noted that as the MPC and RL are iterative control algorithms, the execution time, although small, can accumulate and might bring about too large computational time, especially in the hyperparameter tuning process. Hence, a tradeoff between prediction performance and computational efficiency becomes necessary in such applications. In this study, the RF model is selected as the preferred building model because it is almost four times faster than the ANN model while providing acceptable performance. It is noted that although the LR model provides much faster execution at 0.002 seconds, it is not selected because of its poor prediction performance with MAE of 0.59 kWh and 0.69 °C respectively for energy and temperature predictions.

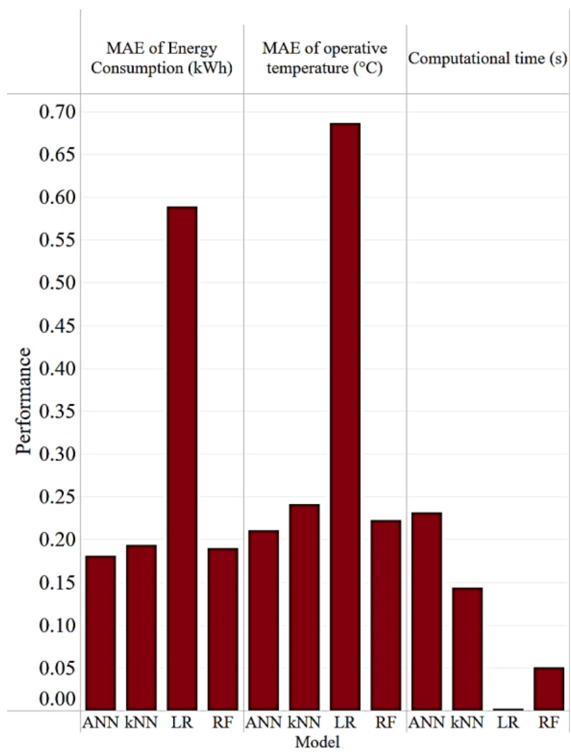


Fig. 7.5. The performance of the building models in terms of prediction performance and computational time.

7.4.2. DDQN control system

7.4.2.1. Hyperparameter tuning

Table 7.7 summarizes the parameters of the reward function and the hyperparameters of the control system determined based on trial and error. In the first step, the type and parameters of the reward function need to be selected. For this purpose, various values are tried, and the performance of the agent is closely monitored. Those values that can enable the agent to learn the following behavior are selected: 1) not sacrificing thermal comfort of occupants to save energy and 2) preheating the building before the occupants arrive.

After determining the parameters of the reward function, the hyperparameters of the DDQN are determined through trial and error. The parameters leading to the highest accumulative reward are selected to develop the ultimate models. It is observed that a median buffer size of 12,875 is selected in this process. This value demonstrates that the agent uses the experience from more than one episode in most cases as an episode consists of around 7,200 steps. The median coefficient of the target model update is calculated as 0.1, showing that a soft update method is the preferred update strategy in most cases. Additionally, the Boltzmann policy is mainly selected with a median minimum search coefficient of 0.10.

Table 7.7.
The hyperparameters of the DDQN model.

Hyperparameters	Ranges
Reward function	
type	Original form of the function
n	2
beta	0.7
DDQN algorithm	
Median buffer size	12,875
Median target model update	0.1
Median f the learning rate	0.001
Median discount factor	0.8
Median decay duration	7 episodes
Mode Policy	Boltzmann
Median minimum search coefficient	0.10

7.4.2.2. Training process and convergence

Fig. 7.6 shows the rewards obtained by the DDQN algorithm over 40 episodes of training for a sample apartment. During the initial episodes, the algorithm continuously improves its performance and increases the rewards. The episodic rewards improve from less than -8,500 to more than -5,000 after 10 episodes of training, which is approximately 41% improvement in the performance. However, after 10 episodes of training, the agent cannot find a superior policy to enhance the performance.

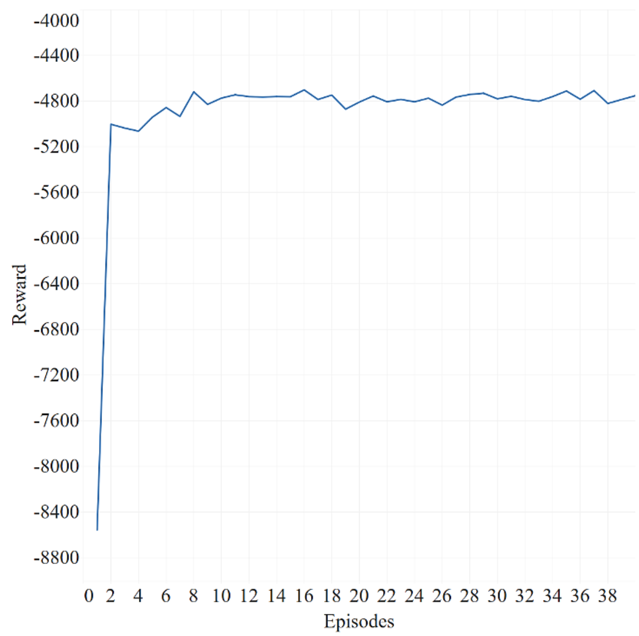


Fig. 7.6. The episodic rewards achieved by the DDQN agent during the training process.

As discussed above, the agent yields a relatively poor performance over the initial periods, which might cause important thermal discomfort or energy waste. Fig. 7.7 depicts the average values of $\overline{T}_{dev}$ caused by the DDQN algorithm during the first 10 episodes of training. The impact of the training process can be observed in the first two episodes, during which $\overline{T}_{dev}$ decreases from 0.44 °C to 0.28 °C, an almost 36% improvement in the thermal comfort. After 10 episodes, the training performance reaches an average of 0.10 °C, which is more than 75% enhancement in the training performance. Fig. 7.8 demonstrates the average values of P_{dev} over the same duration. During the first episode, occupants encounter a thermally uncomfortable environment almost 4% of the time. However, this value decreases by 2.8% to 1.2% after the training process is completed.

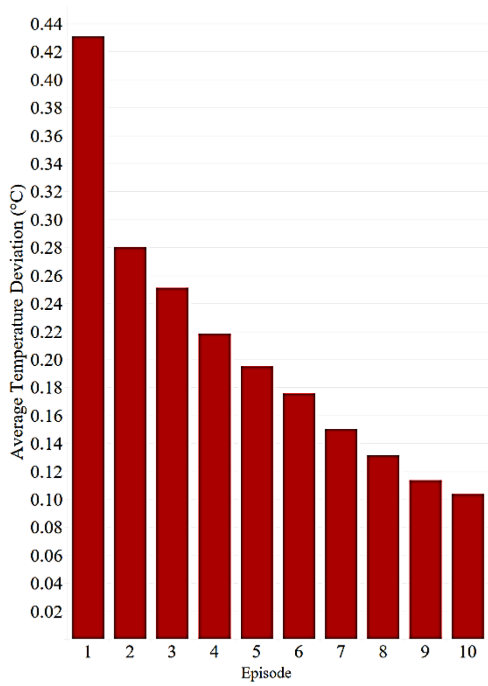


Fig. 7.7. Average temperature deviation during the training period of the DDQN controller.

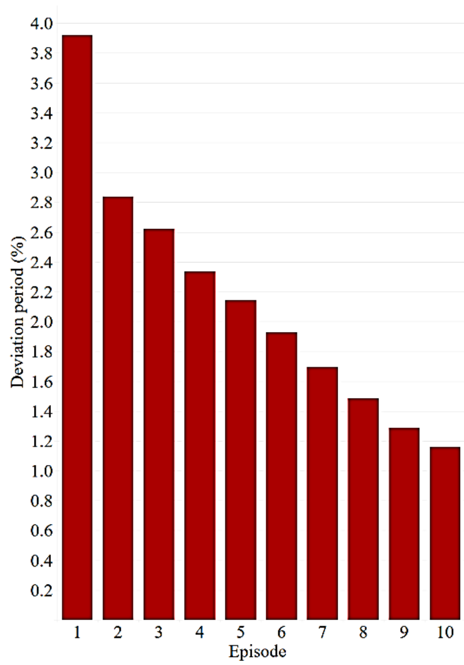


Fig. 7.8. Deviation period during the training period of the DDQN controller.

The average energy consumption per control timestep is demonstrated in Fig. 7.9. Despite the improvements in thermal comfort, the average energy consumption almost remains the same in the initial and the last episodes. More specifically, although the energy consumption increases from 2.04 kWh in episode 1 to 2.06 kWh in episode 2, it gradually decreases to 2.03 kWh in episode 10.

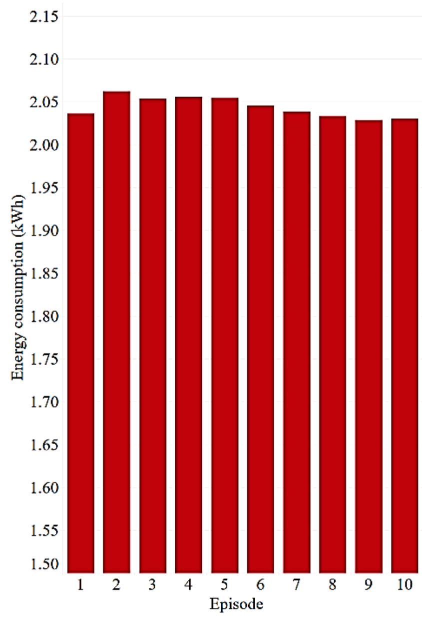


Fig. 7.9. The energy consumption per timestep during the training period of the DDQN controller.

7.4.3. Test performance evaluation

This section discusses the performance of the fully trained DDQN agent on the test dataset and compares it with that of the MPC. The amount of energy saving is calculated by taking the always-on control as the baseline for both the DDQN and MPC algorithms. The following sections discuss their intra-day and annual performance.

7.4.3.1. Intra-day performance

Fig. 7.10 illustrates intra-day indoor operative temperature and associated energy consumption for both DDQN agent and MPC for a sample apartment based on the perfect occupancy prediction. It is observed that both control systems successfully adjust the indoor temperature to the occupancy states, leading to similar energy consumption and indoor temperature patterns. They define a slight setback temperature during the nighttime to save energy when short vacancy intervals are reported while providing an acceptable level of thermal comfort as soon as the occupancy states are reported. Over the long vacancy period starting from 11:30 in the morning, both controllers select a deep setback of as low as 15 ºC to maximize the energy consumption. They start pre-heating the indoor environment one timestep before the arrival of occupants to avoid thermal discomfort. Hence, adopting a setback temperature saved a considerable amount of energy during the mentioned period while not violating thermal comfort conditions.

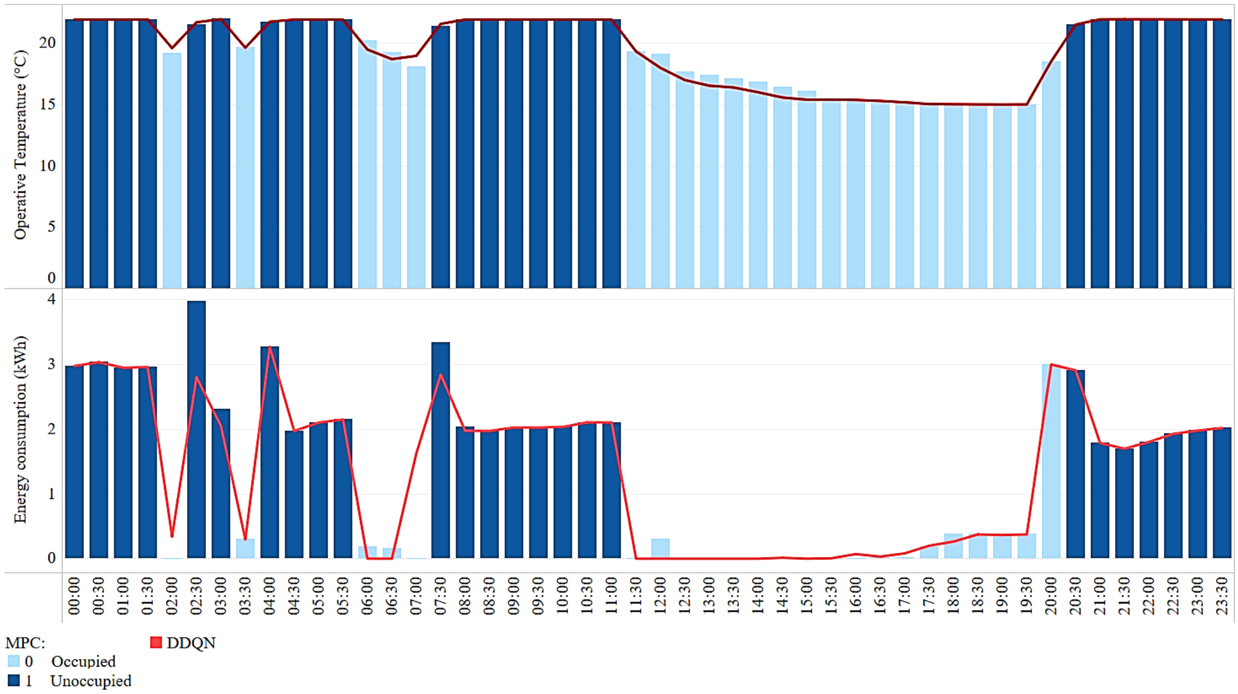


Fig. 7.10. The intra-day operative temperature and energy consumption based on perfect occupancy prediction for a sample apartment on February 1st.

Although both MPC and DDQN make similar control decisions when perfect occupancy prediction is concerned, they respond quite differently to the occupancy prediction uncertainty. Fig. Fig. 7.11 demonstrates the decisions of the control systems based on actual occupancy prediction for the same sample apartment. The DDQN agent follows a conservative strategy for setting the temperature while the MPC acts more aggressively to save as much energy as possible. It is worth mentioning that the use of motion detectors for occupancy detection is prone to reporting false vacancy states during stationary periods such as when people are sleeping at night. During such periods, the uncertainty and randomness is expected to increase. The DDQN agent shows the ability to recognize such uncertainty during the nighttime and, as a result, maintains the thermal comfort level regardless of the reported vacancy states.

The controllers also act differently in response to the long vacancy period during the day. The DDQN agent takes a conservative approach to minimize the possible thermal discomfort that might be caused due to any unexpected return of occupants. It also starts preheating the building before the arrival time of occupants. On the other hand, the MPC algorithm chooses a deep setback and allows the temperature to drop to as low as 17 °C right before the occupants' arrival. Hence, although higher energy saving is achieved by the MPC, a considerable amount of thermal comfort is sacrificed.

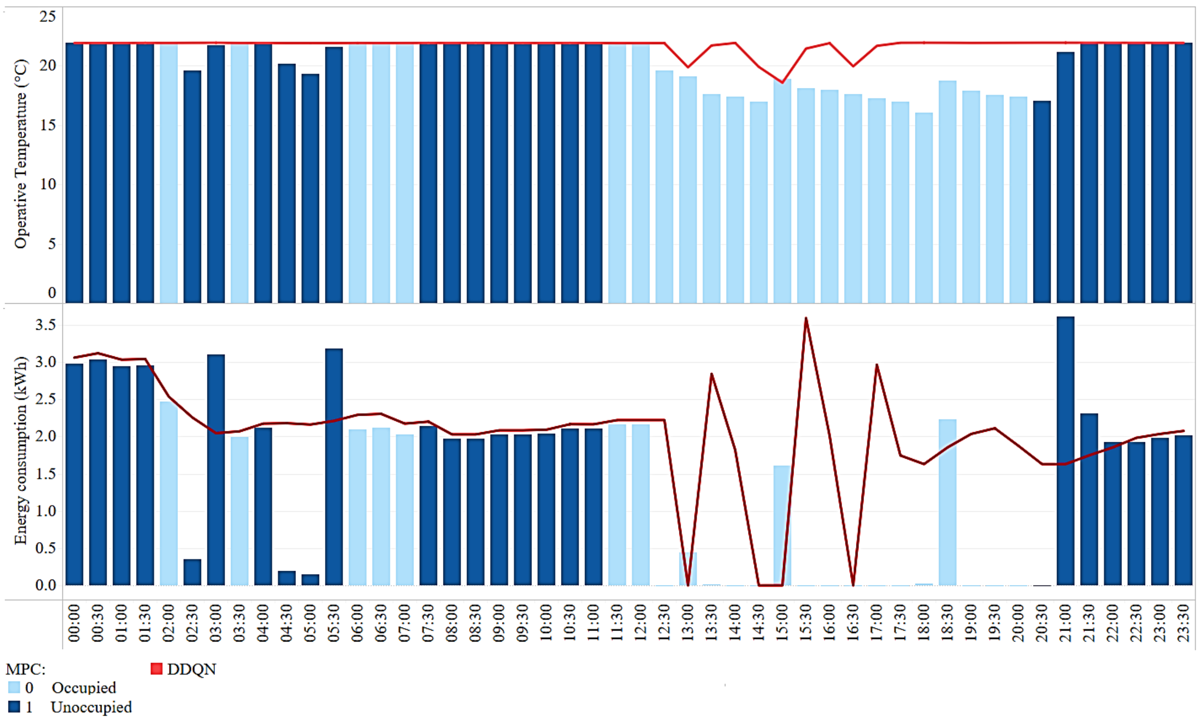


Fig. 7.11. The intra-day operative temperature and energy consumption based on actual occupancy prediction for a sample apartment on February 1st.

7.4.3.2. Overall performance

The distribution of the annual performance of the control systems based on perfect and actual occupancy predictions is investigated in terms of average temperature deviation, deviation period, and energy saving. Fig. 7.12 depicts the distribution of the average temperature deviation for perfect and actual occupancy predictions. Regarding the perfect occupancy prediction, both control systems result in a similar temperature deviation of less than 0.1 °C in most cases. The MPC slightly outperforms the DDQN agent by providing a median value of 0.066 °C, which is 0.008 °C lower than that of the DDQN agent. In contrast, when occupancy prediction uncertainty is added to the problem, the MPC causes poor thermal comfort with a median $\overline{T}_{dev}$ of as large as 0.32 °C. The DDQN agent still keeps $\overline{T}_{dev}$ below 0.1 °C in most cases despite the higher uncertainty. A similar pattern can also be observed for P_{dev} as shown in Fig. 7.13. The MPC outperforms the DDQN agent when perfect prediction is available; it results in a median P_{dev} of 2.23%, which is 1.22% lower than that of the DDQN. However, the performance of the DDQN considerably improves when there is uncertainty about the future patterns as it leads to a median P_{dev} of as low as 1.17%. In contrast, the MPC causes a great deal of thermal discomfort at a median P_{dev} of 9.04% in the same situation.

The higher thermal comfort provided by the DDQN for actual occupancy prediction can be justified by its conservative intra-day decisions described in Section 7.4.3.1. More specifically, the DDQN agent is aware of the uncertainty in occupancy prediction and often acts conservatively to avoid thermal discomfort in cases of unexpected occupancy arrivals. On the other hand, the MPC does not take the prediction errors into account while making the control decisions, which ends up causing large thermal discomfort for occupants.

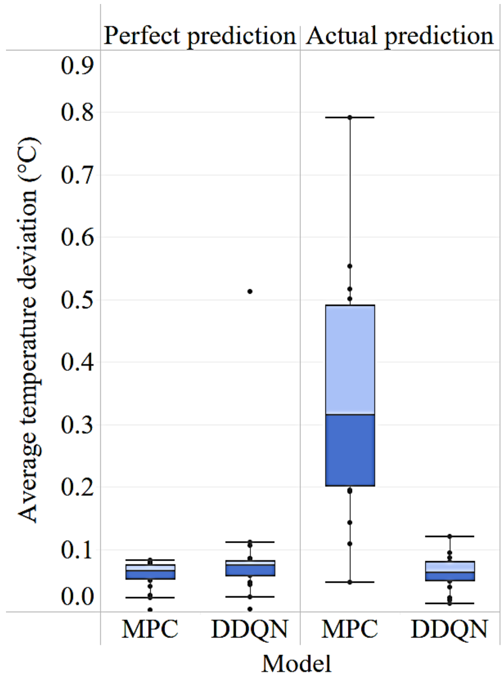


Fig. 7.12. The distribution of average temperature deviation associated with the control systems based on perfect and actual occupancy models.

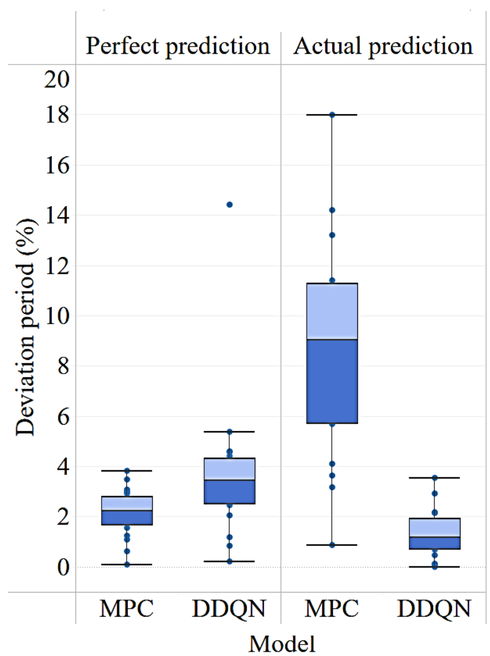


Fig. 7.13. The distribution of deviation period associated with the control systems based on perfect and actual occupancy models.

The distribution of the energy saving percentage is shown in Fig. 7.14 for the perfect and actual occupancy predictions. Along with its superior thermal comfort, the MPC algorithm yields slightly higher energy saving at a median of 27.97% in perfect prediction cases, which is almost 0.5% higher than that of the DDQN. Therefore, it can be concluded that the MPC outperforms the DDQN algorithm when perfect occupancy prediction is available. The reason behind the higher performance of the MPC can be linked to the building model that is used in the optimization algorithm. As discussed in Section 7.2.2.2, while the MPC algorithm takes advantage of a perfect building model in the optimization process, the DDQN agent needs to forecast the building energy demands and indoor temperature changes via trial-and-error. The use of such a perfect building model causes the over-estimation of the MPC performance. Concerning the actual occupancy prediction, the performance of the DDQN algorithm significantly drops to a median energy saving of less than 10%. This drop in the performance is consistent with its conservative temperature setting strategy and its higher thermal comfort level. In contrast, the MPC keeps the energy saving level at around that of the perfect prediction case with a median saving of 27.60%.

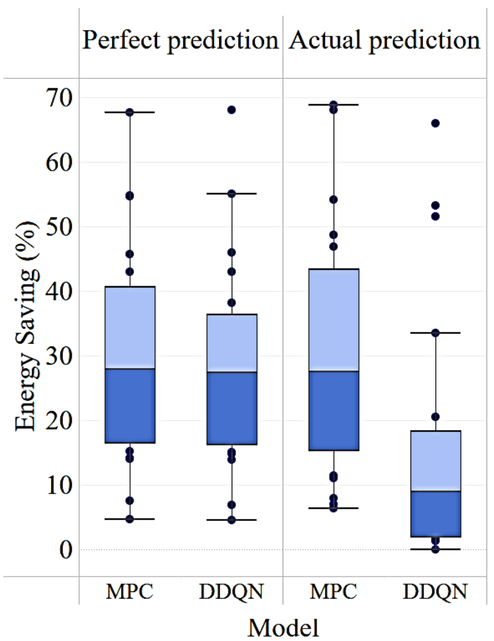


Fig. 7.14. The distribution of energy saving associated with the control systems based on perfect and actual occupancy models.

7.5. Conclusion

This study proposes a self-learning control based on a DDQN algorithm with the aim of learning the dynamic occupancy patterns and HVAC behavior to make optimal control decisions accordingly. The effectiveness of the proposed control system is evaluated by comparing its performance with that of an MPC algorithm, as a well-practiced predictive control approach, and an always-on control as the baseline. The impact of occupancy prediction uncertainty is also assessed by considering two scenarios of perfect and actual occupancy prediction in the control system.

The DDQN agent shows the ability to learn the patterns with no need for developing the occupancy models or building models. When the perfect prediction of occupancy is considered in the control framework, both the DDQN and MPC provide similar energy saving and thermal comfort performance. However, they follow a different strategy in response to the uncertainty. When compared with the MPC, one of the major strengths of the DDQN algorithm is its ability to learn the uncertainty in occupancy prediction. Such understanding leads the agent to adopt a conservative strategy for temperature setting, which significantly reduces the occupants' thermal discomfort. Hence, the DDQN leads to a deviation period of 1.17%, which is 7.87% lower than that of the MPC. However, the higher thermal comfort is achieved at the cost of energy saving as this strategy yields almost three times higher energy consumption than that of the MPC. The DDQN algorithm also shows the ability to reduce the negative impacts of occupancy detection errors due to the use of motion detectors. During the periods when the possibility of false vacancy states is high, such as nighttime when the occupants are mostly stationary, the controller maintains a setpoint temperature. In contrast, the MPC opts for a setback even during such short and random vacancy periods.

Despite its promising ultimate performance on the test dataset, the DDQN is prone to causing serious thermal comfort issues in the training period, especially during the first learning episode. More specifically, the DDQN interactions with the environment can cause an almost 0.43°C average temperature deviation and a 3.9% deviation period in most cases. However, as the learning process continues, these values decrease to as low as 0.10 °C and 1.2%, respectively. This limitation of the model-free DDQN algorithms can be a barrier to the wide adoption of such smart controllers in practical cases. Hence, more research studies are required to focus on the learning phase of the control system to avoid thermal discomfort.

Chapter 8: Transfer learning for model-free HVAC control based on unsupervised learning of occupancy profiles and deep reinforcement learning [7]

8.1. Overview

Reinforcement learning (RL) has shown promising potential for developing self-learning control systems in smart buildings with no need for building models or occupancy models. However, the trial-and-error nature that exists in the RL learning process makes the controller prone to causing serious thermal discomfort for occupants, especially during the initial learning periods. Because of the vital importance of thermal comfort, this limitation can be a key barrier that prevents such control systems from being practical in this sector. This study proposes a control framework that leverages a transfer learning (TL) technique to deal with this limitation by accelerating the learning process. In this control system, the transfer learning is performed based on a similarity analysis in terms of occupancy patterns using unsupervised learning of occupancy profiles. To this end, a k-means clustering algorithm using dynamic time warping is proposed and applied to the occupancy databases collected from 26 residential units. The results indicate that the proposed method can considerably enhance the performance of the system by improving the jumpstart performance and total rewards by almost 25% and 5%, respectively, compared with a conventional RL algorithm. The reward improvements reflect an increase in thermal comfort of occupants as it enhances the deviation period and mean temperature deviation by around 4% and 68%, respectively.

8.2. Methodology

8.2.1. General description of the control framework

The schematic diagram of the proposed control framework is demonstrated in Fig. 8.1. The control system primarily works based on the information that is available from a set of buildings called train cases. These cases can be virtual buildings created through building simulation software or can be experimental testbeds with actual control systems. In the first step, the data from the buildings need to be collected and preprocessed as described in Section 8.2.2. Then, the building data, including indoor temperature and occupancy states, as well as weather data are utilized to develop an RL control system for each building (Section 8.2.3). Additionally, an unsupervised learning technique is applied to the occupancy databases to find the dominant occupancy patterns associated with each household (Section 8.2.4). The extracted occupancy patterns and the pre-trained control agents are stored in a database so that they can be employed in the transfer learning process (Section 8.2.5). The system leverages such information to enhance the control performance in a new unseen building, called a test case. As soon as a test case connects to the network, the system starts collecting new occupancy data. Such information is implemented to provide an estimation of the new occupancy pattern to find its most similar train case (Section 8.2.5.1). The experience of the most similar pre-trained agent is transferred to the new building

[7] This chapter is based on the following publication: M. Esrafilian-Najafabadi, F. Haghighat, Transfer learning for model-free HVAC control based on unsupervised learning of occupancy profiles and deep reinforcement learning, (2022). (Under preparation)

to accelerate the learning process. Ultimately, the control system starts interacting with the environment to improve itself and to save energy while enhancing thermal comfort. The performance of the system is measured on the test case to evaluate the effectiveness of the proposed controller (Section 8.2.5.2).

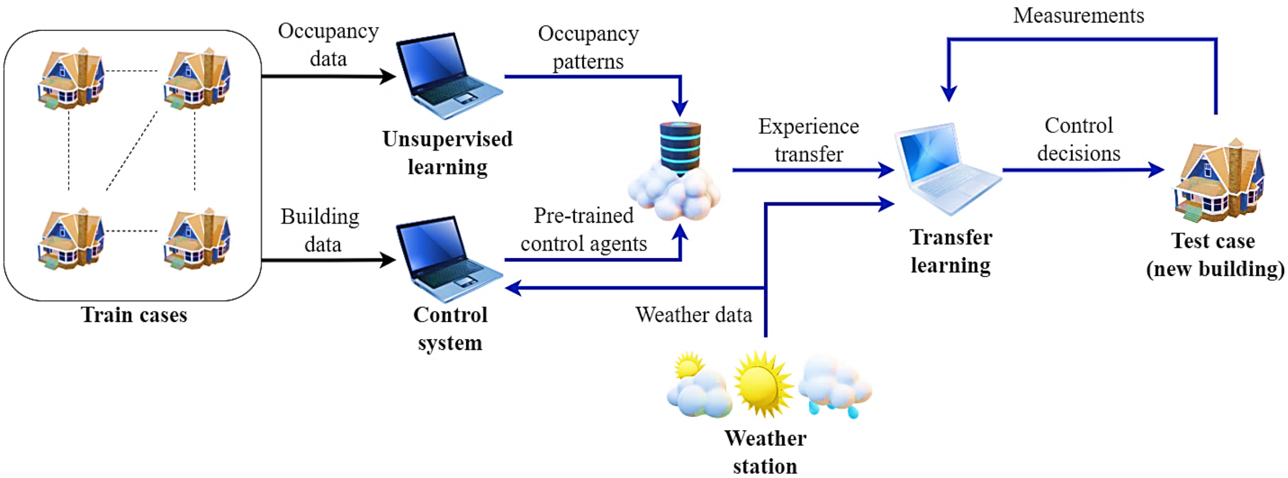


Fig. 8.1. The general framework of the proposed control system.

8.2.2. Database description and preprocessing

A general overview of the occupancy database and the preprocessing steps are illustrated in Fig. 8.2. The database is constructed based on the sensor signals received from passive infrared (PIR) motion detectors installed in 26 residential units in an apartment block. The building consists of one-, two-, and three-bedroom units with an average of 13 sensors installed in different zones of every apartment, including kitchen, bedrooms, and living rooms. A detailed description of the apartments can be found in [219].

The data is stored in an occupancy database at 1-min intervals with each data record taking values of **0** or **1**, showing the vacancy and occupancy states, respectively. The data is prepared for use in the control process through two preprocessing steps. First, the data granularity is reduced by aggregating the data from its original one-minute level to a 30-min level to accelerate the control process and to minimize the impact of the occupancy detection errors of the motion detectors [95]. Secondly, the missing values that exist in the database possibly due to sensor or network failures are replaced by the value of **1**, to ensure that the thermal comfort of occupants is not sacrificed.

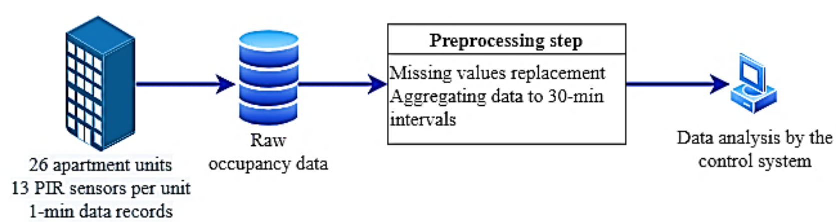


Fig. 8.2. The schematic diagram of the data collection and preprocessing process.

As described in the previous section, the control framework requires two sets of apartments, namely train and test cases. These two sets are created by randomly selecting an individual apartment as the test case and selecting the rest as the train cases. The performance of the control system is assessed based on the test case and stored in a database. Then, a different apartment is selected as the next test case, and the evaluation process repeats. This process results in 26 iterations to cover all the apartments. The evaluation is performed for an entire heating season starting from November 1st to March 31st, which is considered a training episode for the reinforcement learning algorithm, described in Section 8.2.3.

8.2.3. Control system

The DDQN algorithm, proposed in Chapter 7, is employed to learn the occupancy patterns and HVAC characteristics to make optimal control decisions. This technique is selected because of its potential ability to consider the uncertainty of occupancy prediction in making the control decisions and its superior thermal comfort performance when compared with model-based predictive control.

In the first step of developing the DDQN agent, the control problem is formulated as a Markov decision process (MDP). To this end, the reward function, action-space, and state-space need to be defined for the agent, as summarized in Table 8.1. With the aim of saving energy and improving thermal comfort, the reward is defined as a function of energy consumption and temperature deviation. The former term indicates the amount of heating energy, Q_{heat}, that is supplied to the building. The latter term reflects the thermal discomfort caused when the operative temperature, T_{op}, deviates from the setpoint, T_{sp}. The binary variable *Occ* indicates the occupancy state (i.e., taking **0** and **1** respectively for vacancy and occupancy states) because when the building is vacant the thermal comfort is not an issue. As the agent tries to maximize the reward function, these terms are multiplied by a minus sign to make it a minimization problem. γ and β are defined to weight each term of the reward function based on their importance. Regarding the action space, the agent can choose between three discrete actions of deep setback, conservative setback, and setpoint temperature, which are respectively assumed as 15 °C, 19 °C, and 22 °C. It is worth mentioning that the thermal mass of the building might enable the agent to reach intermediate temperatures by switching between the defined setbacks. Therefore, although defining a continuous action space can overcomplicate the control process, it does not necessarily improve the performance. Concerning the state-space, four types of features, namely previous occupancy states, the hour of day, outdoor temperature, and operative temperature, are utilized to help the agent to make the optimal decisions. The first two attributes are selected to enable the agent to learn the occupancy patterns. Accordingly, the controller can try to forecast the values of *Occ* and adjust the indoor temperature. The second two features are selected because of the dependency of the HVAC energy consumption and lag time on the indoor and outdoor temperature. In other words, the agent can utilize these features to learn the HVAC system behavior based on the historical patterns.

Table 8.1.
The state space, action space, and reward function defined for training the RL agent.

Parameter	Value
Reward function	$R = -\gamma(Q_{heat}) - \beta.Occ.\left(T_{op} - T_{sp}\right)^2$ $\beta = 0.7$ $\gamma = 1 - \beta$
Action-space	Deep setback at 15 °C, Conservative setback 19 °C, and Setpoint at 22 °C.
State-space	Occupancy states in the last two hours, Hour of day, Outdoor temperature, and Indoor operative temperature.

A general overlook of the control system, including the interactions between the agent and environment, is demonstrated in Fig. 8.3. In the first step, the state of the environment (i.e., occupancy states, time of day, indoor temperature, and outdoor weather conditions) is given to the agent so that it can make a control decision. Then, the control action (i.e., a setpoint or setback temperature) is passed into the HVAC system. The HVAC system adjusts its operation to provide the required indoor temperature. Next, the sensor network, including motion detectors, temperature sensors, and energy meters, sends the next state and rewards to the controller as feedback. The RL agent starts learning the consequences of the made decisions and corrects itself to maximize energy saving and thermal comfort. This process is iterated in each timestep to enhance the control decisions and adapt to the new conditions.

Fig. 8.3. The general description of the RL agent via MDP.

The characteristics and the hyperparameters of the control systems are summarized in Table 8.2. A Boltzmann policy is implemented with a temperature value of 0.1 to make a tradeoff between exploration and exploitation tasks. The control system takes advantage of an ANN as a function approximator to replace the traditional Q-tables with a neural network of three hidden layers. In this network, the number of neurons implemented in the input and output layers equals the number of states and actions, respectively. The interested readers can find more details regarding the selection of the hyperparameters in Chapter 7.

Table 8.2.
The structural parameters of the DDQN algorithm.

Parameter	Value
Policy	
Policy type	Boltzmann
Temperature value	0.1
Function approximator	
Number of hidden layers	3
Number of neurons	(64, 32, 64)
Optimizer	Adam [235]
Activation function	Rectified Linear Unit (ReLU) [168]
DDQN hyperparameters	
Buffer size	12,875
Target model update	0.1
Learning rate	0.001
Discount factor	0.8

8.2.4. Unsupervised learning

The target of the unsupervised learning process in the control framework is to determine the dominant occupancy patterns of each household. As demonstrated in Fig. 8.4, the process begins with an additional occupancy data preprocessing step, in which a new structure of the occupancy database is constructed with the daily occupancy profiles as its elements (Section 8.2.4.2). Next, the k-means clustering method is applied to this database to extract the dominant occupancy patterns associated with every household (Section 8.2.4.3). After finding the occupancy patterns of all the apartments, the process is stopped, and the results are stored.

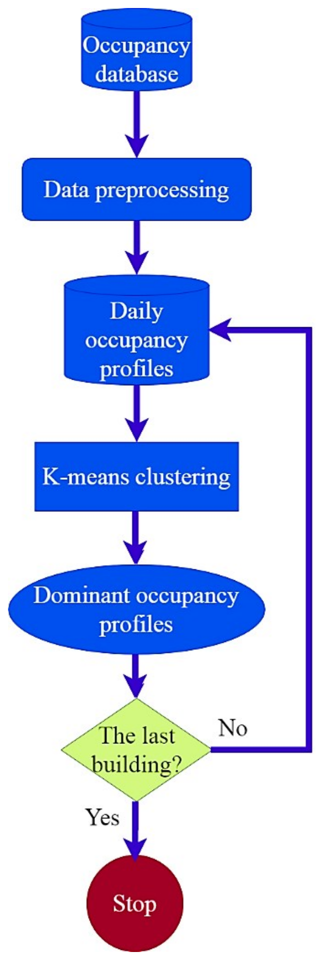


Fig. 8.4. The unsupervised learning process to find the dominant occupancy profiles.

8.2.4.2. Database preprocessing

Before employing the clustering method, the daily occupancy profiles need to be extracted from the original occupancy database. This process creates a new database, in which every element represents a daily occupancy pattern with 24 hour-of-day features. These attributes are binary variables, taking a value of **1** if occupants are present at the corresponding hour. A sample data element of this database is shown in Fig. 8.5. This diagram can be interpreted as a daily occupancy profile, showing that the occupants stayed at the apartment from almost 18:30 to 9:30 and mostly left the place during working hours.

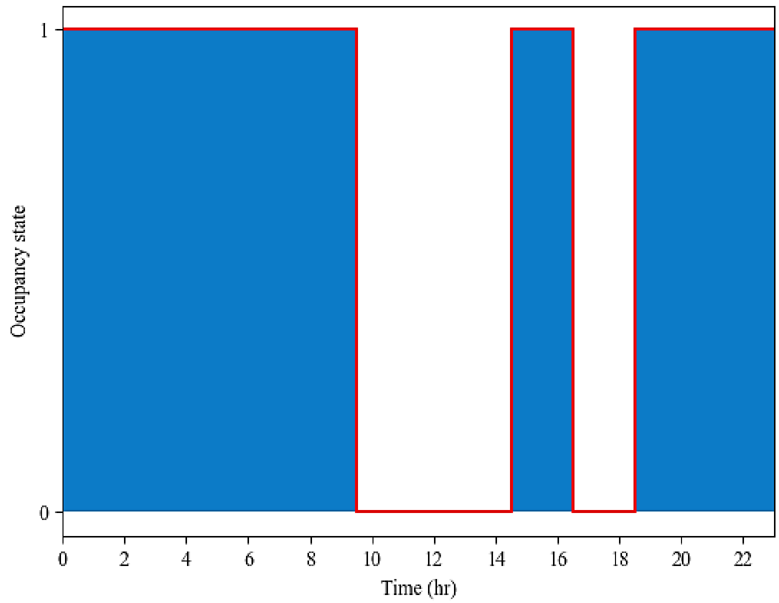


Fig. 8.5. Daily occupancy patterns for a random day in a sample apartment.

8.2.4.3. K-means clustering using dynamic time warping

The k-means clustering, as a widely utilized unsupervised learning technique [244], is implemented to extract the dominant occupancy patterns for each household. This algorithm organizes the data elements within different groups (i.e., clusters) based on their similarity [245]. It is a common practice to measure the similarity via the Euclidean metric. Using this metric, the distance between the occupancy profiles on two different days, denoted by d_1 and d_2, can be calculated as follows:

$$D_{Euclidean} = \sqrt{\sum_{t=0}^{23} (O_{d_1,t} - O_{d_2,t})^2} \quad (35)$$

in which $D_{Euclidean}$ is the Euclidian distance, t denotes hour of day, and O indicates the occupancy state at each time interval. The clustering algorithm begins with defining k centroids with random positions within the database, in which k is a predefined number of clusters. Then, the distance between the data records and the centroids is measured, and then, each record is assigned to its closest cluster. After organizing all the data into the determined groups, the centroids are recalculated and the average of the intra-cluster points is considered as the new position of the corresponding centroid [246]. The process stops when the centroids' locations converge to a final solution [247,248].

It should be noted that the use of the Euclidean distance might lead to ignorance of some similarities in the shape of occupancy profiles [174]. More specifically, although the overall shape of daily patterns can be very similar, the arrival or departure time might be shifted by one or two timesteps. To illustrate such cases, two sample daily occupancy patterns from the same apartment are demonstrated in Fig. 8.6. Both days follow quite similar occupancy patterns; the occupants leave at around 10:00 in the morning and come back after 18:00. Despite their exact arrival time, they leave almost one hour earlier in the morning in day (b). In such cases, using the Euclidean metric might result in rather large distances and low similarity between the apartments. A bi-directional alignment of the occupancy profiles can enable the algorithm to consider the similarity in the shapes. To this end, the dynamic time warping (DTW) algorithm is utilized in this study. The DTW aims to minimize the distance between two time-series profiles by implementing dynamic programming [249]. In this method, the distance of different pairs of points in two profiles is calculated and stored in a corresponding matrix. Using dynamic programming, the shortest paths between the points in the profiles are utilized to measure the closeness between different points. The described clustering methodology based on DTW metric is performed in Python using *tslearn* library [250].

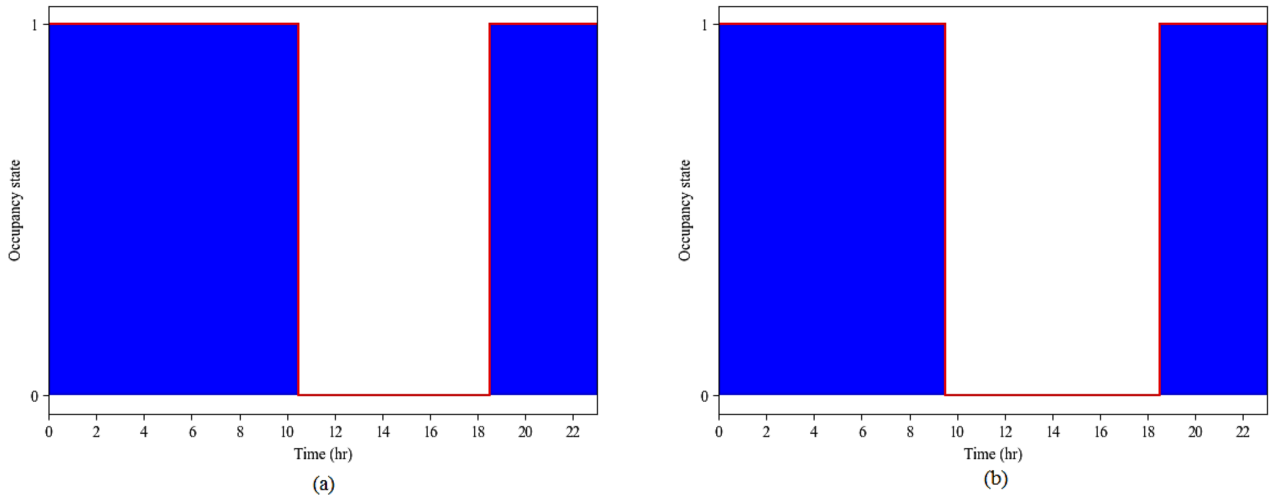


Fig. 8.6. The occupancy patterns of two sample days from the same apartment.

As mentioned earlier, the clustering process using k-means techniques depends on the defined number of clusters. In this study, its optimal number is determined based on a silhouette score [251]. This score evaluates the similarity of data records to their own clusters (intra-cluster similarities) and compares it to their similarity to the neighbor clusters (inter-cluster similarities). A larger score indicates that the samples are good fits for their assigned clusters. The number of clusters that lead to the highest score can be considered optimal [252]. However, each cluster must be representative of a certain occupancy pattern of the households. Hence, a cluster with a size of only a few days is too small to reflect a dominant occupancy schedule for people. To avoid creating such meaningless clusters, a minimum sample number of 14 days is selected as a constraint for cluster creation. The highest number that meets this requirement and results in the highest Silhouette score is selected as the optimal cluster number.

8.2.5. Transfer learning

This section discusses how to transfer the information from a pre-trained controller to the new agent employed in an unseen apartment with the aim of accelerating the training duration. The schematic diagram of this process is depicted in Fig. 8.7. In the first step, the control process begins with collecting and analyzing the occupancy information from the new test case. Although a long data collection can lead to more accurate analysis, it delays the start of the control process, which might cause a waste of energy and cost. In this study, it is assumed that 14 days of data collection from the new apartment can provide an acceptable trade-off between these factors and is selected as the initial data collection duration. In step 2, the occupancy pattern of the new apartment is compared with those of the train cases so that the similarity of the apartments can be evaluated (Section 8.2.5.1). Then, the RL model trained on the most similar apartment is reused to initialize the Q-network weights of the new control agent. In step 3, the control loop begins, and the agent starts interacting with the environment to make control decisions while improving its policy. During this process, the layers of the DNN, which are initialized based on the *source* task (i.e., train cases), are fine-tuned based on the interaction of the new agent and the *target* task (i.e., test cases).

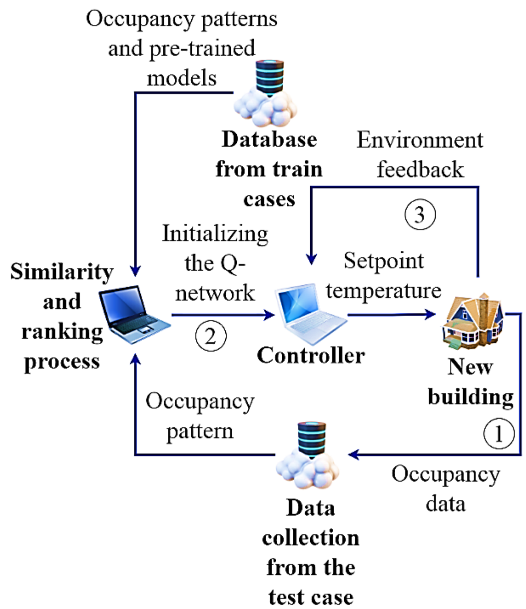


Fig. 8.7. The schematic diagram of the transfer learning process.

8.2.5.1. Similarity analysis

The similarity between the occupancy schedules is quantified by measuring the Euclidean distance, *d*, between the occupancy pattern of the new apartment and the centroids of the training cases. An overall score, *S*, is assigned to each centroid as a function of the measured distance using the following equation:

$$S_c = \frac{d_c - d_{\min}}{d_{\max} - d_{\min}} \tag{1}$$

in which d_c represents the distance between centroid *c* and the profile of the new apartment. The defined score can vary in the range of **0**-**1**; the higher values indicate the higher similarity between the cases. It is worth noting that there is a limitation to using this score coefficient for transferring the agent experience to the new apartment; the formula does not consider the number of the samples that create the occupancy patterns in the apartment. Take a train case apartment with two clusters as an example; one cluster contains only 14 days of elements, called the minor cluster, while the other one contains data from multiple months, called the primary cluster. Assume that a new apartment and the centroid of the minor cluster provide the highest similarity score and are selected for transfer learning. In this case, it is expected that the pre-trained agent is mostly influenced by the occupancy patterns of the major cluster, and consequently, the transfer learning might be negatively affected. To avoid this potential shortcoming, the similarity score is revised based on the number of samples in the train cases as follows:

$$WS_c = S_c \frac{\text{Number of data elements in cluster } c}{\text{Number of total data elements of the corresponding apartment}} \tag{2}$$

where WS_c represents the *weighted score* for cluster *c.* Using this score function, the clusters with the highest elements have priority to be selected for transfer learning.

8.2.5.2. Performance evaluation

It is acknowledged that the use of transfer learning adds considerably higher complexity to the control process. Hence, the proposed control framework essentially needs to provide superior performance to justify the added complications, compared with conventional algorithms. To this end, the proposed system needs is applied to a case study to evaluate its performance. a single-story detached building developed and proposed in [156] is considered in this study as a virtual building testbed. Three metrics are employed to quantify the TL performance as proposed in [253]:

Jumpstart performance: This criterion investigates the initial performance of the agent, defined as the sum of the rewards achieved over the first heating period (i.e., episode of training) for the RL algorithm with and without TL. As the main goal of the control system is to reduce the training time and decrease the thermal discomfort of occupants during the training period, and as a result, this metric is considered the primary target of this study.

Asymptotic performance: In contrast to the jumpstart performance, this metric focuses on the final performance of the agent. In other words, it quantifies how much the transfer learning can improve the ultimate policy learned by the agent. As the learning process is a continuing task (i.e., the learning and adjustment of the control system do not end in practice), the maximum episodic reward during the training process is defined as asymptotic performance.

Total reward: As the name suggests, this criterion evaluates the total performance of the agent by calculating the accumulative reward throughout the training period, which equals the area under the reward curve. An agent can gain more rewards by starting with higher performance in the initial step, learning faster, and converging to higher performance.

As well as the above-mentioned criteria which are based on the obtained rewards, the performance of the TL is also assessed in terms of the energy consumption and thermal comfort of occupants. As proposed in Chapter 7, the thermal comfort can be quantified using *mean temperature deviation* and *deviation period.* The former is defined as the average temperature difference between the actual indoor temperature and the setpoint when the occupants are present in the monitored building. The latter is defined as the proportion of time when the occupants are present but the indoor temperature deviates from the setpoint by more than 0.5 °C. Moreover, the amount of energy consumption associated with the proposed and conventional control systems is also evaluated and compared.

8.3. Results

8.3.1. Occupancy patterns extraction

This section discusses the dominant occupancy patterns of each apartment given by the unsupervised learning method. Fig. 8.8 illustrates the distribution of the number of clusters for use in the k-means clustering method. The number of clusters indicates the variety of the occupancy patterns in every household. In other words, an apartment with more clusters is expected to give more diversity in occupancy patterns. It is observed that more than 60% of the cases yield only two occupancy patterns, creating the largest category. It is followed by the four-cluster group, accounting for almost 27% of the apartments. Three clusters are assigned with the smallest number of apartments at 11.54%.

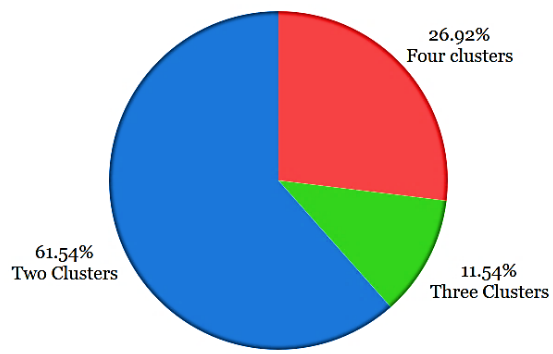


Fig. 8.8. The distribution of the number of clusters for use in k-means clustering algorithm.

The centroids of some sample apartments are investigated in more detail to illustrate the occupancy patterns in the households. Fig. 8.9 demonstrates the occupancy patterns of a sample household with two clusters. The first cluster can be interpreted as a regular working schedule, in which occupants are present during the night and leave the apartment during the day from 10:00 to 19:00. In contrast, cluster 2 indicates an almost uniform occupancy pattern throughout the day. This pattern might illustrate the tendency of the occupants to say home during the vacations.

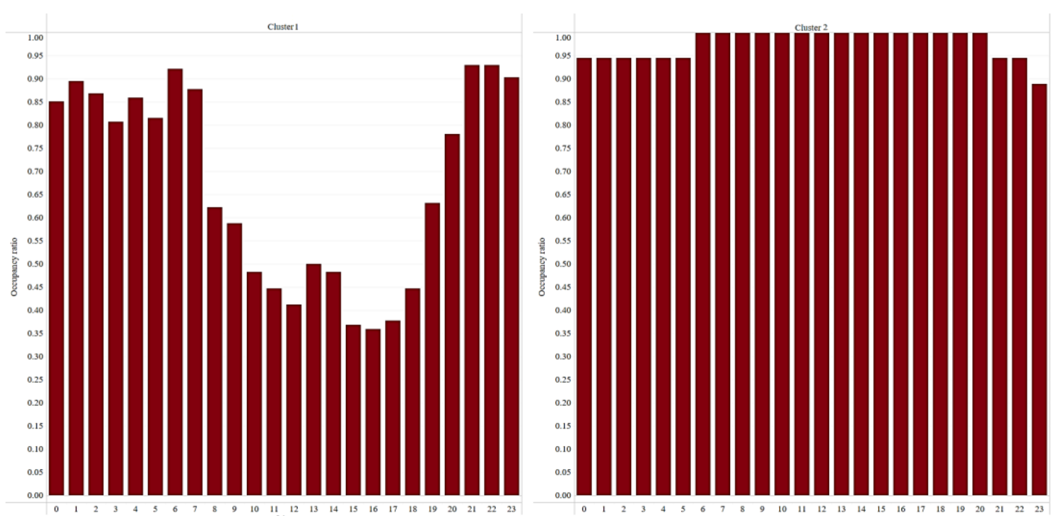


Fig. 8.9. The centroids of a sample household with two clusters.

Fig. 8.10 demonstrates the extracted occupancy patterns from a sample apartment with four clusters. Similar to the previous apartment, cluster 1 refers to a working schedule, in which people usually leave home at 9:00 and return at 19:00. In contrast to the previous apartment, in which people mostly occupy the apartment, this apartment remains vacant most of the day as indicated by the other clusters. Clusters 2 and 4 show that the residents often are not present at home for more than half of the day. Moreover, as can be seen in cluster 3, it is common that the home remains vacant for the entire day.

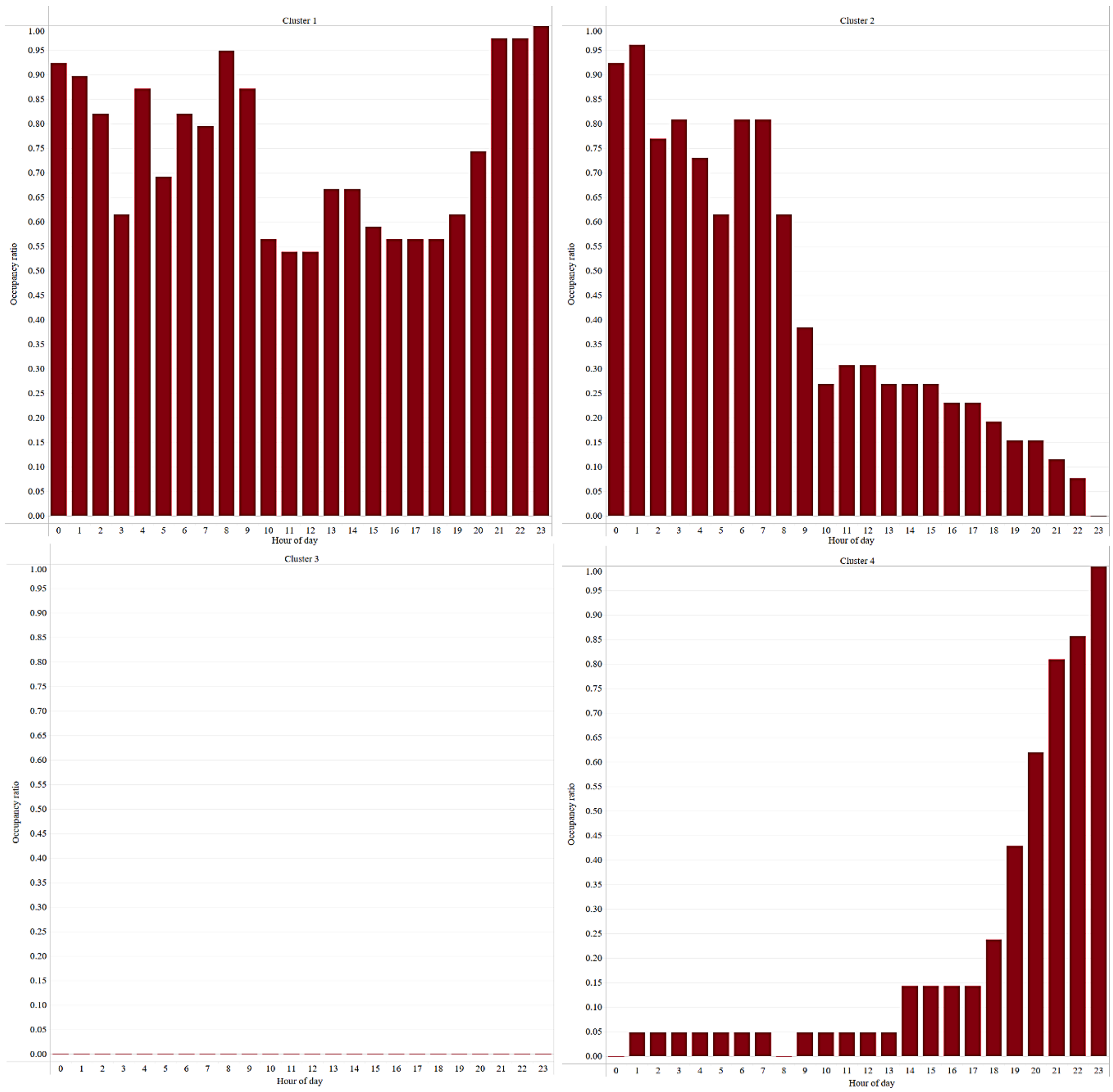


Fig. 8.10. The centroids of a sample apartment with four dominant cluster centroids.

8.3.2. Similarity between the occupancy patterns

After determining the occupancy patterns for each apartment, the weighted score between the test case and the train cases can be calculated to find the most similar patterns. Fig. 8.11 illustrates the initial occupancy pattern in a sample test case and its most similar profile from the training database with the highest weighted score. Both patterns follow a working day profile with the occupancy ratios mostly varying in the range of 0.4-0.8. In both diagrams, the ratios are higher than 0.4 most of the time. As can be seen, there are also dissimilarities between the profiles; the occupancy ratio in the train case can reach as high as 0.9 while it is always limited to 0.6 in the other one.

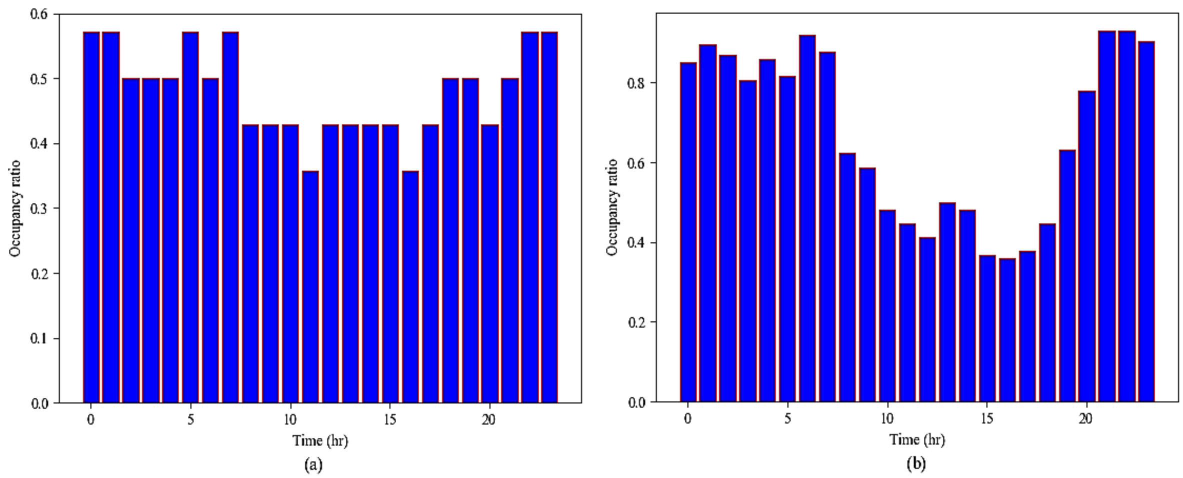


Fig. 8.11. The occupancy patterns of a) a sample test case and b) its most similar train apartment.

8.3.3. Transfer learning performance

Fig. 8.12 illustrates the median values of jumpstart performance, asymptotic performance, and total rewards. RL and TL respectively refer to the performance of the conventional RL and the proposed TL method. The highest impact of the TL on the system is demonstrated in the jumpstart performance. The TL method improves the median jumpstart performance from -6900 to -5076, an almost 25.43% increase in the performance. It also boosts the total rewards, increasing from -5206 for conventional RL control to -4947 for using TL, which is approximately a 5% improvement in the median accumulative rewards. However, the TL method has a negligible impact on the asymptotic performance of the control systems. Hence, it suggests that although TL can accelerate the learning process, it might not be helpful to find a superior ultimate policy.

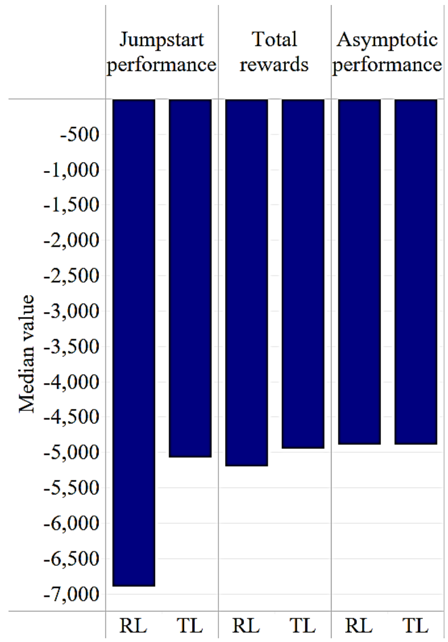


Fig. 8.12. The median jumpstart, asymptotic, and total performance of the conventional control (RL) and the proposed transfer learning method (TL).

Based on the performance of the system, the main strength of the transfer learning is enhancing the jumpstart performance. This improvement can be valuable because it can address the main limitation of the RL algorithm in the building control, which is the extreme thermal discomfort caused during the learning period. Fig. 8.13 depicts the distribution of the deviation period, average temperature deviation, and energy consumption. As expected, the conventional RL agent causes higher thermal discomfort for occupants in terms of the deviation period and average temperature deviation. When the conventional RL is employed, the occupants of almost half of the apartments encounter thermal discomfort more than 6% of the heating season. However, the TL method substantially improves this metric to less than 2% in most cases. Moreover, more than 75% of the households experience a mean temperature deviation of more than 0.13 ºC with a median value of 0.19 ºC when the traditional RL algorithm is implemented. However, almost all the apartments experience a temperature deviation of less than 0.1 ºC, when the TL method is applied. More specifically, the use of TL yields a median deviation of 0.06, almost 68% improvement.

A large improvement in thermal comfort is achieved at the expense of energy consumption. It is observed that the conventional RL slightly overcome the TL method in terms of energy saving. The conventional RL agent consumes a median of 2.47 kWh energy per each time interval (i.e., 30-minute periods) which is 0.08 kWh lower than that of the TL method. The difference is equivalent to an almost 3% rise in energy consumption of the households. However, regarding the improvements in thermal comfort, the added energy can be considered relatively small.

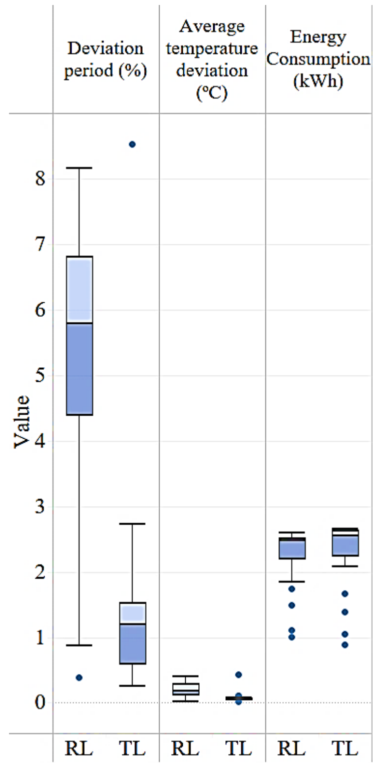


Fig. 8.13. The distribution of the average temperature deviation, deviation period, and energy consumption in the first episode of training for the conventional RL control and the proposed TL.

8.4. Conclusion

This study investigates the use of transfer learning to improve the performance of self-learning HVAC control systems based on model-free reinforcement learning. The transfer learning is performed based on an inter-apartment similarity analysis in terms of occupancy schedules, using a k-means clustering algorithm based on the DTW distance metric. The effectiveness of the proposed control system is evaluated by measuring its jumpstart, asymptotic, and total performance and comparing them with those of a traditional RL algorithm. Moreover, the initial performance of the control system is assessed from thermal comfort and energy viewpoints by evaluating the deviation period, mean temperature deviation, and energy consumption.

The unsupervised learning reveals that the households can be assigned to no more than four dominant occupancy patterns. More than half of the cases result in two occupancy patterns, which may be interpreted as working and non-working schedules. The TL-based control framework shows the ability to improve the performance of the RL agent in unseen test buildings. It improves the jumpstart performance and total rewards by 25% and 5%, respectively. However, no improvements are observed in terms of asymptotic performance. The higher jumpstart leads to considerably less thermal discomfort for occupants during the first episode of learning; the median of the deviation period and average temperature deviation are improved by 4% and 68%, respectively. It is worth noting that the thermal comfort enhancements come with a 3% rise in the median energy consumption during this period.

The promising performance of the proposed framework can provide opportunities to deal with one of the main limitations of the conventional model-free controllers, which is their poor thermal comfort performance during the initial period of training. However, this study is performed based on the occupancy data of a limited number of apartments, and more cases are still required to generalize the results. Additionally, only occupancy data is utilized as a criterion to investigate the similarity among the control agents. Undoubtedly, there are more factors such as weather conditions, building material, and HVAC characteristics that need to be considered for the similarity analysis and performance evaluation. There is a need for future research to investigate the impact of such factors on the performance of the control system in residential buildings.

Chapter 9: Summary, conclusion and future directions

9.1. Summary and conclusion

This research work begins with identifying the current research gaps and limitations regarding occupancy-based control methods based on a comprehensive literature review on the state-of-the-art systems. The literature review classifies the earlier studies based on the control strategy and discusses the strengths and limitations of different strategies. The limitations are discussed from different perspectives, such as feature utilization, model selection, case study buildings, and performance evaluation metrics.

This research work proposes a new control methodology that leverages artificial intelligence to enhance thermal comfort and energy saving in residential buildings. The control framework works based on a DDQN algorithm to interact with the building environment with the goal of learning occupancy schedules and HVAC lag time. Without a need for developing models of the occupancy and building, the control system seeks optimal control decisions to save energy while maintaining an acceptable level of thermal comfort for occupants. The merit of the proposed controller is evaluated by comparing its performance with that of an MPC algorithm, as a well-practiced method. This algorithm takes advantage of occupancy models to forecast future states. Because of the importance of occupancy information, the impact of selecting different occupancy models is investigated. Two different approaches, namely arrival time prediction and occupancy states predictions, are implemented and the performance of different machine learning models including DT, kNN, MLP, GRU, LSTM, and BLSTM are also evaluated. Based on a TOPSIS method, the model that leads to the best trade-off between thermal comfort and energy saving is selected as the occupancy forecasting model of the control system.

The proposed self-learning control utilizes two methods to improve the initial performance: 1) a MOGA for feature selection and 2) a transfer learning methodology based on unsupervised learning of occupancy patterns. To ensure that the proposed MOGA is a reliable method to remove redundancy and irrelevancy, its performance is compared with the FSS, BSS, and SOGA as conventional methods. The MOGA aims to maximize the accuracy of the models while minimizing the size of the feature set to reduce the impact of overfitting. Concerning the latter technique, the transfer learning method is employed to reuse the valuable knowledge gained by previously trained controllers in new buildings. The control framework takes advantage of a K-means clustering method based on the DTW distance metric to quantify the similarity between the buildings in terms of occupancy schedules. Next, this method utilizes the most similar buildings for transferring the information. This framework aims to reduce the training duration of the RL agent and, consequently, reduce the risk of thermal discomfort due to the trial-and-error nature of the RL method.

The trained DDQN agent shows superior thermal comfort performance with the MPC as a baseline. The MPC algorithm is developed based on a kNN algorithm as the occupancy-state prediction model because of its higher performance among the investigated models. In contrast to the MPC, the DDQN agent shows the potential of learning uncertainty in occupancy schedules and opts for a more conservative temperature setting, which results in an almost 8% lower temperature deviation. It also shows the ability to deal with the potential errors of the occupancy detection network as it often ignores the changes in occupancy states during the nighttime when people are mostly stationary.

The proposed MOGA yields superior performance among the utilized FS methods by increasing the accuracy by up to 4.81% and requiring the smallest feature set for occupancy prediction. According to the TOPSIS method, a maximum value of six features is needed to develop the occupancy models. While recent occupancy states and time of day show essential importance in occupancy prediction modeling, recent CO2 concentration, vacancy duration, holiday, and lighting states might also improve the performance in some cases.

The TL method shows promising effects on the first episode of training via improving the jumpstart performance by 25%. It also improves the total rewards over the training period by 5%. Regarding the building performance, this control system provides better thermal comfort for occupants by reducing the deviation period and average temperature duration by 4% and 68%, respectively. The unsupervised learning method demonstrates that the investigated households give up to four different occupancy schedules with more than half of the cases having only two clusters. The schedules can be mostly classified as working and non-working patterns in buildings.

The comprehensive performance analysis also reports a promising improvement that can be achieved by taking dynamic occupancy changes into account for making control decisions. It shows that the use of occupancy data in the control system can result in up to 2.72 years of payback period, 12,91% energy saving, and almost $2,000 net present value. It is also revealed that there are negative impacts on the peak-energy demand of the residential buildings; the system can increase the peak energy demand by up to 10%.

9.2. Contributions

The contributions of this research work can be summarized as follows:

1. This study developed a self-learning control system based on dynamic occupancy patterns with no need for employing models of occupancy and building,
2. A transfer-learning framework was proposed to enhance the training performance of the data-driven control system,
3. An optimal feature selection method was developed to determine the key features for forecasting occupancy patterns,
4. The effectiveness of the occupancy prediction models was evaluated for use in the predictive control systems,
5. An assessment framework was developed for a comprehensive evaluation of the financial, energy, and peak-demand performance, and
6. The research gaps in the field of occupancy-based HVAC control were carefully identified by conducting a comprehensive literature review on the state-of-the-art methodologies.

9.3. Limitations and directions for future work

Despite the mentioned contributions of this study and the promising performance of the proposed control framework, there are some limitations that need more research for further improvements towards the practical application in the residential sector. The limitations can be summarized as follows:

- The investigations of occupancy modeling and control system evaluations are performed based on a limited number of occupancy datasets collected from an apartment block. In order to generalize the results, there is a need to collect more data from different buildings for further analysis.
- In the unsupervised learning method and transfer learning framework, only the occupancy schedules are considered as a metric to quantify the similarity between the apartments. However, there are more factors such as building characteristics, type of HVAC systems, and weather conditions that can influence the decisions of the controller. There is a need to investigate the impact of such factors on improving the performance of the transfer learning method.
- Regarding the FS process, a limited number of methods, namely BSS, FSS, and SOGA, are employed. Given that there are more well-known methods, such as recursive feature elimination, premutation importance, filter, and embedded FS methods, there is a need to utilize more methods for feature selection to ensure that the accuracy is not sacrificed by eliminating valuable features.
- There is a wide variety of features that could be utilized in the FS process. Although most of them are considered in this study, some are lacking in the utilized database, such as occupants' identity, occupant counts, and the exact locations, which might be helpful. A database containing such information is required to reveal the impacts of these features on the control performance.

Bibliography

[1] International Energy Agency (iea), Transition to Sustainable Buildings, 2013. https://webstore.iea.org/download/direct/745.

[2] P. Palensky, D. Dietrich, Demand side management: Demand response, intelligent energy systems, and smart loads, IEEE Trans. Ind. Informatics. 7 (2011) 381–388. https://doi.org/10.1109/TII.2011.2158841.

[3] Itron inc., California commercial end-use survey, 2006.

[4] V. Singhvi, A. Krause, C. Guestrin, J.H. Garrett, H. Scott Matthews, Intelligent light control using sensor networks, in: SenSys 2005 - Proc. 3rd Int. Conf. Embed. Networked Sens. Syst., Association for Computing Machinery, New York, New York, USA, 2005: pp. 218–229. https://doi.org/10.1145/1098918.1098942.

[5] M.K. Masood, C. Jiang, Y.C. Soh, A novel feature selection framework with Hybrid Feature-Scaled Extreme Learning Machine (HFS-ELM) for indoor occupancy estimation, Energy Build. 158 (2018) 1139–1151. https://doi.org/10.1016/j.enbuild.2017.08.087.

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

[7] V.L. Erickson, Y. Lin, A. Kamthe, R. Brahme, A. Surana, A.E. Cerpa, M.D. Sohn, S. Narayanan, Energy efficient building environment control strategies using real-time occupancy measurements, in: BUILDSYS 2009 - Proc. 1st ACM Work. Embed. Sens. Syst. Energy-Efficiency Build. Held Conjunction with ACM SenSys 2009, ACM Press, New York, New York, USA, 2009: pp. 19–24. https://doi.org/10.1145/1810279.1810284.

[8] V.L. Erickson, A.E. Cerpa, Occupancy based demand response HVAC control strategy, in: BuildSys'10 - Proc. 2nd ACM Work. Embed. Sens. Syst. Energy-Efficiency Build., ACM Press, New York, New York, USA, 2010: pp. 7–12. https://doi.org/10.1145/1878431.1878434.

[9] Y. Agarwal, B. Balaji, S. Dutta, R.K. Gupta, T. Weng, Duty-cycling buildings aggressively: The next frontier in HVAC control, in: Proc. 10th ACM/IEEE Int. Conf. Inf. Process. Sens. Networks, IPSN'11, IEEE, 2011: pp. 246–257.

[10] F. Jazizadeh, A. Ghahramani, B. Becerik-Gerber, T. Kichkaylo, M. Orosz, Human-building interaction framework for personalized thermal comfort-driven systems in office buildings, J. Comput. Civ. Eng. 28 (2014) 2–16. https://doi.org/10.1061/(asce)cp.1943-5487.0000300.

[11] T. Peffer, M. Pritoni, A. Meier, C. Aragon, D. Perry, How people use thermostats in homes: A review, Build. Environ. 46 (2011) 2529–2541. https://doi.org/10.1016/j.buildenv.2011.06.002.

[12] D. Urieli, P. Stone, A learning agent for heat-pump thermostat control, in: Proc. 2013 Int. Conf. Auton. Agents Multi-Agent Syst., 2013: pp. 1093–1100.

[13] J. Lu, T. Sookoor, V. Srinivasan, G. Gao, B. Holben, J. Stankovic, E. Field, K. Whitehouse,

The smart thermostat: Using occupancy sensors to save energy in homes, in: SenSys 2010 - Proc. 8th ACM Conf. Embed. Networked Sens. Syst., 2010: pp. 211–224. https://doi.org/10.1145/1869983.1870005.

[14] L. Zhang, J. Wen, A systematic feature selection procedure for short-term data-driven building energy forecasting model development, Energy Build. 183 (2019) 428–442. https://doi.org/10.1016/j.enbuild.2018.11.010.

[15] I. Guyon, A. Elisseeff, An Introduction to Variable and Feature Selection, J. Mach. Learn. Res. (2003) 1157–1182.

[16] N. Haidar, N. Tamani, F. Nienaber, M.T. Wesseling, A. Bouju, Y. Ghamri-Doudane, Data collection period and sensor selection method for smart building occupancy prediction, in: IEEE Veh. Technol. Conf., Institute of Electrical and Electronics Engineers Inc., 2019. https://doi.org/10.1109/VTCSpring.2019.8746447.

[17] J. Gluck, C. Koehler, J. Mankoff, A. Dey, Y. Agarwal, A Systematic Approach for Exploring Tradeoffs in Predictive HVAC Control Systems for Buildings, ArXiv Prepr. ArXiv1705.02058. (2017). https://arxiv.org/abs/1705.02058v1.

[18] S. Brandi, M.S. Piscitelli, M. Martellacci, A. Capozzoli, Deep reinforcement learning to optimise indoor temperature control and heating energy consumption in buildings, Energy Build. 224 (2020) 110225. https://doi.org/10.1016/j.enbuild.2020.110225.

[19] G. Kontes, G. Giannakis, V. Sánchez, P. de Agustin-Camacho, A. Romero-Amorrortu, N. Panagiotidou, D. Rovas, S. Steiger, C. Mutschler, G. Gruen, Simulation-Based Evaluation and Optimization of Control Strategies in Buildings, Energies. 11 (2018) 3376. https://doi.org/10.3390/en11123376.

[20] R.S. Sutton, A.G. Barto, Reinforcement learning: An introduction, MIT press, 2018.

[21] Z. Wang, T. Hong, Reinforcement learning for building controls: The opportunities and challenges, Appl. Energy. 269 (2020) 115036. https://doi.org/10.1016/j.apenergy.2020.115036.

[22] Z. Cheng, Q. Zhao, F. Wang, Y. Jiang, L. Xia, J. Ding, Satisfaction based Q-learning for integrated lighting and blind control, Energy Build. 127 (2016) 43–55. https://doi.org/10.1016/j.enbuild.2016.05.067.

[23] B. Jiang, Y. Fei, Smart home in smart microgrid: A cost-effective energy ecosystem with intelligent hierarchical agents, IEEE Trans. Smart Grid. 6 (2015) 3–13. https://doi.org/10.1109/TSG.2014.2347043.

[24] F. Ruelens, B.J. Claessens, S. Vandael, S. Iacovella, P. Vingerhoets, R. Belmans, Demand response of a heterogeneous cluster of electric water heaters using batch reinforcement learning, in: Proc. - 2014 Power Syst. Comput. Conf. PSCC 2014, Institute of Electrical and Electronics Engineers Inc., 2014. https://doi.org/10.1109/PSCC.2014.7038106.

[25] Q. Wei, D. Liu, G. Shi, A Novel Dual Iterative Q-Learning Method for Optimal Battery Management in Smart Residential Environments, IEEE Trans. Ind. Electron. 62 (2015) 2509–2518. https://doi.org/10.1109/TIE.2014.2361485.

[26] Z. Zhang, A. Chong, Y. Pan, C. Zhang, K.P. Lam, Whole building energy model for HVAC optimal control: A practical framework based on deep reinforcement learning, Energy Build. 199 (2019) 472–490. https://doi.org/10.1016/j.enbuild.2019.07.029.

[27] Z. Zou, X. Yu, S. Ergan, Towards optimal control of air handling units using deep reinforcement learning and recurrent neural network, Build. Environ. 168 (2020) 106535.

[28] O.T. Masoso, L.J. Grobler, The dark side of occupants' behaviour on building energy use, Energy Build. 42 (2010) 173–177. https://doi.org/10.1016/j.enbuild.2009.08.009.

[29] S.M.R. Khani, F. Haghighat, K. Panchabikesan, M. Ashouri, Extracting energy-related knowledge from mining occupants' behavioral data in residential buildings, J. Build. Eng. 39 (2021) 102319. https://doi.org/10.1016/j.jobe.2021.102319.

[30] S. Naylor, M. Gillott, T. Lau, A review of occupant-centric building control strategies to reduce building energy use, Renew. Sustain. Energy Rev. 96 (2018) 1–10. https://doi.org/10.1016/j.rser.2018.07.019.

[31] K. Sun, Q. Zhao, J. Zou, A review of building occupancy measurement systems, Energy Build. 216 (2020) 109965. https://doi.org/10.1016/j.enbuild.2020.109965.

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

[33] A. Mirakhorli, B. Dong, Occupancy behavior based model predictive control for building indoor climate—A critical review, Energy Build. 129 (2016) 499–513. https://doi.org/10.1016/j.enbuild.2016.07.036.

[34] T.A. Nguyen, M. Aiello, Energy intelligent buildings based on user activity: A survey, Energy Build. 56 (2013) 244–257. https://doi.org/10.1016/j.enbuild.2012.09.005.

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

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

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

[38] E. Azar, W. O'Brien, S. Carlucci, T. Hong, A. Sonta, J. Kim, M.S. Andargie, T. Abuimara, M. El Asmar, R.K. Jain, M.M. Ouf, F. Tahmasebi, J. Zhou, Simulation-aided occupant-centric building design: A critical review of tools, methods, and applications, Energy Build. 224 (2020) 110292. https://doi.org/10.1016/j.enbuild.2020.110292.

[39] S. Karjalainen, O. Koistinen, User problems with individual temperature control in offices, Build. Environ. 42 (2007) 2880–2887. https://doi.org/10.1016/j.buildenv.2006.10.031.

[40] D. Shiller, Programmable thermostat program proposal, 2006. https://www.energystar.gov/ia/partners/prod_development/revisions/downloads/thermostats/Proposal_011106.pdf.

[41] RLW Analytics, Validating the Impact of Programmable Thermostats, Middletown, CT, 2007.

[42] P. Baillargeon, L. Megdal, Billing Analysis & Environment that "Re-Sets" Savings for Programmable Thermostats in New Homes, (2010).

[43] US. Energy information administration (EIA), Residential energy consumption survey (recs), 2015. https://www.eia.gov/consumption/residential/data/2015/.

[44] C. Koehler, B.D. Ziebart, J. Mankoff, A.K. Dey, TherML: Occupancy prediction for thermostat control, in: UbiComp 2013 - Proc. 2013 ACM Int. Jt. Conf. Pervasive Ubiquitous Comput., ACM Press, New York, New York, USA, 2013: pp. 103–112. https://doi.org/10.1145/2493432.2493441.

[45] J. Scott, A.J.B. Brush, J. Krumm, B. Meyers, M. Hazas, S. Hodges, N. Villar, PreHeat: Controlling home heating using occupancy prediction, in: UbiComp'11 - Proc. 2011 ACM Conf. Ubiquitous Comput., ACM Press, New York, New York, USA, 2011: pp. 281–290. https://doi.org/10.1145/2030112.2030151.

[46] S. Iyengar, S. Kalra, A. Ghosh, D. Irwin, P. Shenoy, B. Marlin, Inferring smart schedules for dumb thermostats, ACM Trans. Cyber-Physical Syst. 3 (2018) 1–29. https://doi.org/10.1145/3226031.

[47] M. Pritoni, A.K. Meier, C. Aragon, D. Perry, T. Peffer, Energy efficiency and the misuse of programmable thermostats: The effectiveness of crowdsourcing for understanding household behavior, Energy Res. Soc. Sci. 8 (2015) 190–197. https://doi.org/10.1016/j.erss.2015.06.002.

[48] M. Pritoni, J. Woolley, T. Peffer, M. Modera, Why occupancy-responsive adaptive thermostats do not always save - and the limits for when they should, ACEEE Summer Study Energy Effic. Build. (2014).

[49] T. Labeodan, W. Zeiler, G. Boxem, Y. Zhao, Occupancy measurement in commercial office buildings for demand-driven control applications - A survey and detection system evaluation, Energy Build. 93 (2015) 303–314. https://doi.org/10.1016/j.enbuild.2015.02.028.

[50] B. Balaji, J. Xu, A. Nwokafor, R. Gupta, Y. Agarwal, Sentinel: Occupancy based HVAC actuation using existing wifi infrastructure within commercial buildings, in: Proc. 11th ACM Conf. Embed. Networked Sens. Syst., 2013. https://doi.org/10.1145/2517351.2517370.

[51] A. Corna, L. Fontana, A.A. Nacci, D. Sciuto, Occupancy detection via iBeacon on Android devices for smart building management, in: Proc. -Design, Autom. Test Eur. DATE, Institute of Electrical and Electronics Engineers Inc., 2015: pp. 629–632. https://doi.org/10.7873/date.2015.0753.

[52] Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei, T. Weng, Occupancy-driven energy

management for smart building automation, in: BuildSys'10 - Proc. 2nd ACM Work. Embed. Sens. Syst. Energy-Efficiency Build., ACM Press, New York, New York, USA, 2010: pp. 1–6. https://doi.org/10.1145/1878431.1878433.

[53] T. Labeodan, K. Aduda, W. Zeiler, F. Hoving, Experimental evaluation of the performance of chair sensors in an office space for occupancy detection and occupancy-driven control, Energy Build. 111 (2016) 195–206. https://doi.org/10.1016/j.enbuild.2015.11.054.

[54] M. Pritoni, J.M. Woolley, M.P. Modera, Do occupancy-responsive learning thermostats save energy? A field study in university residence halls, Energy Build. 127 (2016) 469–478. https://doi.org/10.1016/j.enbuild.2016.05.024.

[55] H. Stopps, M.F. Touchie, Reduction of HVAC system runtime due to occupancy-controlled smart thermostats in contemporary multi-unit residential building suites, in: IOP Conf. Ser. Mater. Sci. Eng., Institute of Physics Publishing, 2019: p. 062013. https://doi.org/10.1088/1757-899X/609/6/062013.

[56] T. Sookoor, K. Whitehouse, RoomZoner: Occupancy-based room-level zoning of a centralized HVAC system, in: Proc. ACM/IEEE 4th Int. Conf. Cyber-Physical Syst. ICCPS 2013, 2013: pp. 209–218. https://doi.org/10.1145/2502524.2502553.

[57] Z. Yang, B. Becerik-Gerber, Assessing the impacts of real-time occupancy state transitions on building heating/cooling loads, Energy Build. 135 (2017) 201–211. https://doi.org/10.1016/j.enbuild.2016.11.038.

[58] C. Wang, K. Pattawi, H. Lee, Energy saving impact of occupancy-driven thermostat for residential buildings, Energy Build. 211 (2020). https://doi.org/10.1016/j.enbuild.2020.109791.

[59] M. Krarti, Evaluation of occupancy-based temperature controls on energy performance of KSA residential buildings, Energy Build. 220 (2020) 110047. https://doi.org/10.1016/j.enbuild.2020.110047.

[60] L. Nikdel, K. Janoyan, S.D. Bird, S.E. Powers, Multiple perspectives of the value of occupancy-based HVAC control systems, Build. Environ. 129 (2018) 15–25. https://doi.org/10.1016/j.buildenv.2017.11.039.

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

[62] M. Gupta, S.S. Intille, K. Larson, Adding GPS-control to traditional thermostats: An exploration of potential energy savings and design challenges, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), Springer, Berlin, Heidelberg, 2009: pp. 95–114. https://doi.org/10.1007/978-3-642-01516-8_8.

[63] S. Lee, Y. Chon, Y. Kim, R. Ha, H. Cha, Occupancy prediction algorithms for thermostat control systems using mobile devices, IEEE Trans. Smart Grid. 4 (2013) 1332–1340. https://doi.org/10.1109/TSG.2013.2247072.

[64] A. Beltran, V.L. Erickson, A.E. Cerpa, ThermoSense: Occupancy Thermal Based Sensing for HVAC Control, in: Proc. 5th ACM Work. Embed. Syst. Energy-Efficient Build. -

BuildSys’13, Association for Computing Machinery (ACM), New York, New York, USA, 2013: pp. 1–8. https://doi.org/10.1145/2528282.2528301.

[65] B. Dong, B. Andrews, Sensor-based occupancy behavioral pattern recognition for energy and comfort management in intelligent buildings, in: Proc. Build. Simul., 2009: pp. 1444–1451.

[66] V.L. Erickson, M.Á. CarreiraPerpiñán, A. E. Cerpa, OBSERVE: Occupancy-based system for efficient reduction of HVAC energy, in: 10th ACM/IEEE Int. Conf. Inf. Process. Sens. Networks, IEEE, 2011: pp. 258–269. https://ieeexplore.ieee.org/abstract/document/5779043.

[67] V.L. Erickson, S. Achleitner, A.E. Cerpa, POEM: Power-efficient occupancy-based energy management system, in: IPSN 2013 - Proc. 12th Int. Conf. Inf. Process. Sens. Networks, Part CPSWeek 2013, ACM Press, New York, New York, USA, 2013: pp. 203–216. https://doi.org/10.1145/2461381.2461407.

[68] F. Nägele, T. Kasper, B. Girod, Turning up the heat on obsolete thermostats: A simulation-based comparison of intelligent control approaches for residential heating systems, Renew. Sustain. Energy Rev. 75 (2017) 1254–1268. https://doi.org/10.1016/j.rser.2016.11.112.

[69] S. Salimi, A. Hammad, Optimizing energy consumption and occupants comfort in open-plan offices using local control based on occupancy dynamic data, Build. Environ. 176 (2020) 106818. https://doi.org/10.1016/j.buildenv.2020.106818.

[70] F. Oldewurtel, D. Sturzenegger, M. Morari, Importance of occupancy information for building climate control, Appl. Energy. 101 (2013) 521–532. https://doi.org/10.1016/j.apenergy.2012.06.014.

[71] S. Goyal, H.A. Ingley, P. Barooah, Occupancy-based zone-climate control for energy-efficient buildings: Complexity vs. performance, Appl. Energy. (2013). https://doi.org/10.1016/j.apenergy.2013.01.039.

[72] M. Killian, M. Kozek, Short-term occupancy prediction and occupancy based constraints for MPC of smart homes, IFAC-PapersOnLine. (2019). https://doi.org/10.1016/j.ifacol.2019.08.239.

[73] M. Jain, R.K. Kalaimani, S. Keshav, C. Rosenberg, Using personal environmental comfort systems to mitigate the impact of occupancy prediction errors on HVAC performance, Energy Informatics. 1 (2018) 1–21. https://doi.org/10.1186/s42162-018-0064-9.

[74] J. Shi, N. Yu, W. Yao, Energy Efficient Building HVAC Control Algorithm with Real-time Occupancy Prediction, Energy Procedia. 111 (2017) 267–276. https://doi.org/10.1016/j.egypro.2017.03.028.

[75] C. Turley, M. Jacoby, G. Pavlak, G. Henze, Development and Evaluation of Occupancy-Aware HVAC Control for Residential Building Energy Efficiency and Occupant Comfort, Energies. 13 (2020) 5396. https://doi.org/10.3390/en13205396.

[76] G. Gao, K. Whitehouse, The self-programming thermostat: Optimizing setback schedules based on home occupancy patterns, in: BUILDSYS 2009 - Proc. 1st ACM Work. Embed. Sens. Syst. Energy-Efficiency Build. Held Conjunction with ACM SenSys 2009, ACM

Press, New York, New York, USA, 2009: pp. 67–72. https://doi.org/10.1145/1810279.1810294.

[77] D.B. Crawley, L.K. Lawrie, C.O. Pedersen, F.C. Winkelmann, Energy plus: energy simulation program, ASHRAE J. 42 (2000) 49–56.

[78] H. Kim, E. Oldham, Characterizing Variations in the Indoor Temperature and Humidity of Guest Rooms with an Occupancy-Based Climate Control Technology, Energies. 13 (2020) 1575. https://doi.org/10.3390/en13071575.

[79] R.J. Rose, J. Dozier, EPA program impacts office zoning, ASHRAE J. 39 (1997) 37. https://search.proquest.com/docview/220491089?pq-origsite=gscholar&fromopenview=true.

[80] Bureau of labor statistics, American Time Use Survey, (2019). https://www.bls.gov/tus/#data.

[81] M.M. Manning, M.C. Swinton, F. Szadkowski, J. Gusdorf, K. Ruest, The Effects of Thermostat Set-back and Set-up on Seasonal Energy Consumption, Surface Temperatures and Recovery Times at the CCHT Twin House Facility, ASHRAE Trans. 113 (2004) 630–642.

[82] F.C. Sangogboye, K. Imamovic, M.B. Kjærgaard, Improving occupancy presence prediction via multi-label classification, in: 2016 IEEE Int. Conf. Pervasive Comput. Commun. Work. PerCom Work. 2016, Institute of Electrical and Electronics Engineers Inc., 2016. https://doi.org/10.1109/PERCOMW.2016.7457147.

[83] M. Gjoreski, H. Gjoreski, R. Piltaver, M. Gams, Predicting the Arrival and the Departure Time of an Employee, Preko 5000 Let Slov. Inov. (2013) 3–6. https://dis.ijs.si/hristijan/publications/2013/IS_2013.pdf.

[84] F.C. Sangogboye, M.B. Kjærgaard, PROMT: predicting occupancy presence in multiple resolution with time-shift agnostic classification, in: Comput. Sci. - Res. Dev., Springer Verlag, 2018: pp. 105–115. https://doi.org/10.1007/s00450-017-0351-x.

[85] H. Shao, Y. Li, F. Li, E. Griffiths, K. Whitehouse, N. Ramakrishnan, Temporal Mining Mixture Model for Residential Occupancy Prediction, UrbComp. (2017).

[86] W. Kleiminger, F. Mattern, S. Santini, Predicting household occupancy for smart heating control: A comparative performance analysis of state-of-the-art approaches, Energy Build. 85 (2014) 493–505. https://doi.org/10.1016/j.enbuild.2014.09.046.

[87] J. Krumm, A.J.B. Brush, Learning time-based presence probabilities, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), Springer, Berlin, Heidelberg, 2011: pp. 79–96. https://doi.org/10.1007/978-3-642-21726-5_6.

[88] A.A. Adamopoulou, A.M. Tryferidis, D.K. Tzovaras, A context-aware method for building occupancy prediction, Energy Build. 110 (2016) 229–244. https://doi.org/10.1016/j.enbuild.2015.10.003.

[89] T. Yu, Modeling occupancy behavior for energy efficiency and occupants comfort

management in intelligent buildings, in: Proc. - 9th Int. Conf. Mach. Learn. Appl. ICMLA 2010, 2010: pp. 726–731. https://doi.org/10.1109/ICMLA.2010.111.

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

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

[92] S. Kim, S. Kang, K.R. Ryu, G. Song, Real-time occupancy prediction in a large exhibition hall using deep learning approach, Energy Build. 199 (2019) 216–222. https://doi.org/10.1016/j.enbuild.2019.06.043.

[93] A. Das, M.B. Kjærgaard, Precept: Occupancy presence prediction inside a commercial building, in: UbiComp/ISWC 2019- - Adjun. Proc. 2019 ACM Int. Jt. Conf. Pervasive Ubiquitous Comput. Proc. 2019 ACM Int. Symp. Wearable Comput., Association for Computing Machinery, Inc, New York, New York, USA, 2019: pp. 486–491. https://doi.org/10.1145/3341162.3345605.

[94] H. Elkhoukhi, M. Bakhouya, M. Hanifi, D. El Ouadghiri, On the use of Deep Learning Approaches for Occupancy prediction in Energy Efficient Buildings, in: Proc. 2019 7th Int. Renew. Sustain. Energy Conf. IRSEC 2019, Institute of Electrical and Electronics Engineers Inc., 2019. https://doi.org/10.1109/IRSEC48032.2019.9078164.

[95] B. Huchuk, S. Sanner, W. O'Brien, Comparison of machine learning models for occupancy prediction in residential buildings using connected thermostat data, Build. Environ. 160 (2019) 106177. https://doi.org/10.1016/j.buildenv.2019.106177.

[96] S.H. Ryu, H.J. Moon, Development of an occupancy prediction model using indoor environmental data based on machine learning techniques, Build. Environ. 107 (2016) 1–9. https://doi.org/10.1016/j.buildenv.2016.06.039.

[97] Z. Chen, Y.C. Soh, Modeling building occupancy using a novel inhomogeneous Markov chain approach, in: IEEE Int. Conf. Autom. Sci. Eng., IEEE Computer Society, 2014: pp. 1079–1084. https://doi.org/10.1109/CoASE.2014.6899459.

[98] C. Liao, Y. Lin, P. Barooah, Agent-based and graphical modelling of building occupancy, J. Build. Perform. Simul. 5 (2012) 5–25. https://doi.org/10.1080/19401493.2010.531143.

[99] C. Liao, P. Barooah, An integrated approach to occupancy modeling and estimation in commercial buildings, in: Proc. 2010 Am. Control Conf. ACC 2010, 2010: pp. 3130–3135. https://doi.org/10.1109/acc.2010.5531035.

[100] S.A. Mumma, Transient occupancy ventilation by monitoring CO2, ASHRAE IAQ Appl. 5 (2004) 21–23. http://www.doas-radiant.psu.edu/IAQ_winter_04.pdf.

[101] N. Kiukkonen, J. Blom, O. Dousse, D. Gatica-Perez, Towards rich mobile phone datasets: Lausanne data collection campaign, in: Proc. ICPS, Berlin, 2010.

[102] C. Song, Z. Qu, N. Blumm, A.L. Barabási, Limits of predictability in human mobility,

American Association for the Advancement of Science, 2010. https://doi.org/10.1126/science.1177170.

[103] P. Fabian, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine Learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830. https://doi.org/10.1145/2786984.2786995.

[104] S. Salimi, A. Hammad, Sensitivity analysis of probabilistic occupancy prediction model using big data, Build. Environ. 172 (2020) 106729. https://doi.org/10.1016/j.buildenv.2020.106729.

[105] Y. Ye, Y. Zheng, Y. Chen, J. Feng, X. Xie, Mining individual life pattern based on location history, in: Proc. - IEEE Int. Conf. Mob. Data Manag., 2009: pp. 1–10. https://doi.org/10.1109/MDM.2009.11.

[106] American Society of Heating Refrigerating and Air-Conditioning Engineers, ASHRAE Standard 62.1: Ventilation for acceptable indoor air quality, 2007.

[107] B.D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J.A. Bagnell, M. Hebert, A.K. Dey, S. Srinivasa, Planning-based prediction for pedestrians, in: 2009 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS 2009, 2009: pp. 3931–3936. https://doi.org/10.1109/IROS.2009.5354147.

[108] B.D. Ziebart, A.L. Maas, A.K. Dey, J.A. Bagnell, Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior, in: UbiComp 2008 - Proc. 10th Int. Conf. Ubiquitous Comput., ACM Press, New York, New York, USA, 2008: pp. 322–331. https://doi.org/10.1145/1409635.1409678.

[109] S. Goyal, P. Barooah, T. Middelkoop, Experimental study of occupancy-based control of HVAC zones, Appl. Energy. 140 (2015) 75–84. https://doi.org/10.1016/j.apenergy.2014.11.064.

[110] H. Thieblemont, Simplified Predictive Control for Load Management: A Self-Learning Approach Applied to Electrically Heated Floor, Concordia University, 2017.

[111] Y. Sun, F. Haghighat, B.C.M. Fung, A review of the-state-of-the-art in data-driven approaches for building energy prediction, Energy Build. 221 (2020) 110022. https://doi.org/10.1016/j.enbuild.2020.110022.

[112] D. Ürge-Vorsatz, L.F. Cabeza, S. Serrano, C. Barreneche, K. Petrichenko, Heating and cooling energy trends and drivers in buildings, Renew. Sustain. Energy Rev. 41 (2015) 85–98. https://doi.org/10.1016/j.rser.2014.08.039.

[113] International Energy Agency (iea), The Future of Cooling Opportunities for energy-efficient air conditioning, (2018). https://www.iea.org/reports/the-future-of-cooling.

[114] M. Esrafilian-Najafabadi, F. Haghighat, Occupancy-based HVAC control systems in buildings: A state-of-the-art review, Build. Environ. 197 (2021) 107810. https://doi.org/10.1016/j.buildenv.2021.107810.

[115] Z. Jiang, M.J. Risbeck, V. Ramamurti, S. Murugesan, J. Amores, C. Zhang, Y.M. Lee, K.H.

Drees, Building HVAC control with reinforcement learning for reduction of energy cost and demand charge, Energy Build. 239 (2021) 110833. https://doi.org/10.1016/j.enbuild.2021.110833.

[116] S. Goyal, H.A. Ingley, P. Barooah, Occupancy-based zone-climate control for energy-efficient buildings: Complexity vs. performance, Appl. Energy. (2013). https://doi.org/10.1016/j.apenergy.2013.01.039.

[117] H. Thieblemont, F. Haghighat, A. Moreau, G. Lacroix, Control of electrically heated floor for building load management: A simplified self-learning predictive control approach, Energy Build. 172 (2018) 442–458. https://doi.org/10.1016/j.enbuild.2018.04.042.

[118] D. Bian, M. Pipattanasomporn, S. Rahman, A human expert-based approach to electrical peak demand management, IEEE Trans. Power Deliv. 30 (2015) 1119–1127. https://doi.org/10.1109/TPWRD.2014.2348495.

[119] A. Kathirgamanathan, M. De Rosa, E. Mangina, D.P. Finn, Data-driven predictive control for unlocking building energy flexibility: A review, Renew. Sustain. Energy Rev. 135 (2021) 110120. https://doi.org/10.1016/j.rser.2020.110120.

[120] J. Li, K. Panchabikesan, Z. Yu, F. Haghighat, M. El Mankibi, D. Corgier, Systematic data mining-based framework to discover potential energy waste patterns in residential buildings, Energy Build. 199 (2019) 562–578. https://doi.org/10.1016/j.enbuild.2019.07.032.

[121] Oracle Corporation, MySQL, (n.d.). https://dev.mysql.com/doc/.

[122] C. Turley, M. Jacoby, G. Pavlak, G. Henze, Development and Evaluation of Occupancy-Aware HVAC Control for Residential Building Energy Efficiency and Occupant Comfort, Energies. 13 (2020) 5396. https://doi.org/10.3390/en13205396.

[123] B. Ratner, The correlation coefficient: Its values range between+ 1/− 1, or do they?, J. Targeting, Meas. Anal. Mark. 17 (2009) 139–142. https://doi.org/https://doi.org/10.1201/b11508-11.

[124] J. Reynolds, Y. Rezgui, A. Kwan, S. Piriou, A zone-level, building energy optimisation combining an artificial neural network, a genetic algorithm, and model predictive control, Energy. 151 (2018) 729–739. https://doi.org/10.1016/j.energy.2018.03.113.

[125] F. Chollet, Keras, (2015). https://github.com/fchollet/keras.

[126] ASHRAE, Standard Method of Test for the Evaluation of Building Energy Analysis Computer Programs, ASHRAE, 2014.

[127] EnergyPlus, Testing and Validation, (n.d.). https://energyplus.net/testing.

[128] ASHRAE, Standard for the Design of High-Performance Green Buildings, ASHRAE, 2009.

[129] EnergyPlus, Weather Data by Location, (n.d.). https://energyplus.net/weather-location/north_and_central_america_wmo_region_4/CAN/PQ/CAN_PQ_Montreal.Intl.AP.716270_CWEC.

[130] Trimble, SketchUp, (2020). https://www.sketchup.com/.

[131] National Renewable Energy Laboratory, OpenStudio, (n.d.). http://nrel.github.io/OpenStudio-user-documentation/.

[132] National Renewable Energy Laboratory, EnergyPlus Python API, (2020). https://nrel.github.io/EnergyPlus/api/python/.

[133] Hydro Quebec, Rate D, (2021). https://www.hydroquebec.com/residential/customer-space/rates/rate-d-billing.html.

[134] A. Ardalan, Economic and financial analysis for engineering and project management, CRC Press, 1999.

[135] S.A. Shams, R. Ahmadi, Dynamic optimization of solar-wind hybrid system connected to electrical battery or hydrogen as an energy storage system, Int. J. Energy Res. (2021) er.6549. https://doi.org/10.1002/er.6549.

[136] A.B. Badiru, O.A. Omitaomu, Computational economic analysis for engineering and industry, CRC Press, 2007.

[137] C. Wang, K. Pattawi, H. Lee, Energy saving impact of occupancy-driven thermostat for residential buildings, Energy Build. 211 (2020) 109791. https://doi.org/10.1016/j.enbuild.2020.109791.

[138] M. Esrafilian, R. Ahmadi, Energy, environmental and economic assessment of a polygeneration system of local desalination and CCHP, Desalination. 454 (2019) 20–37. https://doi.org/10.1016/j.desal.2018.12.004.

[139] D.G. Newman, T. Eschenbach, J.P. Lavelle, Engineering Economic Analysis, Oxford University Press, 2012.

[140] The atmospheric fund (TAF), smart thermostats: technology assessment and field test findings in multi-unit residential buildings, 2019. https://taf.ca/wp-content/uploads/2019/04/Smart-Thermostats-White-Paper-March-2019.pdf.

[141] Revenu Quebec, Basic rules for applying the GST/HST and QST, (2021). https://www.revenuquebec.ca/en/businesses/consumption-taxes/gsthst-and-qst/basic-rules-for-applying-the-gsthst-and-qst/.

[142] Retraite Québec, Inflation and indexation, (2019). https://www.rrq.gouv.qc.ca/en/planification/reflexions/Pages/inflation-et-indexation.aspx.

[143] Ontario Energy Board, Managing costs with time-of-use rates, (2021). https://www.oeb.ca/rates-and-your-bill/electricity-rates/managing-costs-time-use-rates.

[144] Ewa Roszkowska, multi-criteria decision making models by applying the topsis method to crisp and interval data, Mult. Criteria Decis. Mak. / Univ. Econ. Katowice. 6 (2011) 200–230. https://mcdm.ue.katowice.pl/files/papers/mcdm11(6)_11.pdf.

[145] Y.J. Lai, T.Y. Liu, C.L. Hwang, TOPSIS for MODM, Eur. J. Oper. Res. 76 (1994) 486–500. https://doi.org/10.1016/0377-2217(94)90282-8.

[146] U.S Energy Information Administration, How much energy is consumed in U.S. buildings?, (2020). https://www.eia.gov/tools/faqs/faq.php?id=86&t=1.

[147] J. (Tom) Luo, M.M. Joybari, K. Panchabikesan, Y. Sun, F. Haghighat, A. Moreau, M. Robichaud, Performance of a self-learning predictive controller for peak shifting in a building integrated with energy storage, Sustain. Cities Soc. 60 (2020) 102285. https://doi.org/10.1016/j.scs.2020.102285.

[148] J. (Tom) T. Luo, M.M. Joybari, K. Panchabikesan, F. Haghighat, A. Moreau, M. Robichaud, Parametric study to maximize the peak load shifting and thermal comfort in residential buildings located in cold climates, J. Energy Storage. 30 (2020) 101560. https://doi.org/10.1016/j.est.2020.101560.

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

[150] R. Pascanu, M. Tomas, B. Yoshua, On the difficulty of training recurrent neural networks, in: Int. Conf. Mach. Learn., 2013: pp. 1310–1318.

[151] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, J. Schmidhuber, A novel connectionist system for unconstrained handwriting recognition, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2009) 855–868. https://doi.org/10.1109/TPAMI.2008.137.

[152] L. Ren, X. Cheng, X. Wang, J. Cui, L. Zhang, Multi-scale Dense Gate Recurrent Unit Networks for bearing remaining useful life prediction, Futur. Gener. Comput. Syst. 94 (2019) 601–609. https://doi.org/10.1016/j.future.2018.12.009.

[153] Z. Cui, R. Ke, Z. Pu, Y. Wang, Deep Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction, ArXiv Prepr. ArXiv1801.02143. (2018). http://arxiv.org/abs/1801.02143 (accessed September 16, 2020).

[154] A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, in: Neural Networks, Pergamon, 2005: pp. 602–610. https://doi.org/10.1016/j.neunet.2005.06.042.

[155] Google, Colaboratory, (2020). https://research.google.com/colaboratory/faq.html.

[156] M. Esrafilian-Najafabadi, F. Haghighat, Occupancy-based HVAC control using deep learning algorithms for estimating online preconditioning time in residential buildings, Energy Build. 252 (2021) 111377. https://doi.org/10.1016/j.enbuild.2021.111377.

[157] V.L. Erickson, S. Achleitner, A.E. Cerpa, POEM: Power-efficient occupancy-based energy management system, in: IPSN 2013 - Proc. 12th Int. Conf. Inf. Process. Sens. Networks, Part CPSWeek 2013, ACM Press, New York, New York, USA, 2013: pp. 203–216. https://doi.org/10.1145/2461381.2461407.

[158] L. Peterson, K-nearest neighbor, Scholarpedia. 4 (2009) 1883. https://doi.org/10.4249/scholarpedia.1883.

[159] M.S. Mirnaghi, F. Haghighat, Fault detection and diagnosis of large-scale HVAC systems in buildings using data-driven methods: A comprehensive review, Energy Build. 229 (2020) 110492. https://doi.org/10.1016/j.enbuild.2020.110492.

[160] D.W. Aha, ed., Lazy learning, Springer Science & Business Media, 2013. https://doi.org/10.1007/978-0-387-30164-8_443.

[161] J. Su, H. Zhang, A fast decision tree learning algorithm, in: Aaai, 2006: pp. 500–505. https://dl.acm.org/doi/10.5555/1597538.1597619.

[162] S.R. Safavian, D. Landgrebe, A Survey of Decision Tree Classifier Methodology, IEEE Trans. Syst. Man Cybern. 21 (1991) 660–674. https://doi.org/10.1109/21.97458.

[163] Y.Y. Song, Y. Lu, Decision tree methods: applications for classification and prediction, Shanghai Arch. Psychiatry. 27 (2015) 130–135. https://doi.org/10.11919/j.issn.1002-0829.215044.

[164] A.J. Myles, R.N. Feudale, Y. Liu, N.A. Woody, S.D. Brown, An introduction to decision tree modeling, J. Chemom. 18 (2004) 275–285. https://doi.org/10.1002/cem.873.

[165] Z. Yu, F. Haghighat, B.C.M. Fung, H. Yoshino, A decision tree method for building energy demand modeling, Energy Build. 42 (2010) 1637–1646. https://doi.org/10.1016/j.enbuild.2010.04.006.

[166] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830. https://www.jmlr.org/papers/v12/pedregosa11a.html.

[167] L. Noriega, Multilayer perceptron tutorial, Sch. Comput. Staff. Univ. (2005). https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.608.2530&rep=rep1&type=pdf.

[168] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016. https://mitpress.mit.edu/books/deep-learning.

[169] R. Reed, R.J. MarksII, Neural smithing: supervised learning in feedforward artificial neural networks, Mit Press, 1999. https://mitpress.mit.edu/books/neural-smithing.

[170] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006. https://link.springer.com/book/9780387310732.

[171] S. Hochreiter, The vanishing gradient problem during learning recurrent neural nets and problem solutions, Int. J. Uncertainty, Fuzziness Knowledge-Based Syst. 6 (1998) 107–116. https://doi.org/https://doi.org/10.1142/s0218488598000094.

[172] R. Fu, Z. Zhang, L. Li, Using LSTM and GRU neural network methods for traffic flow prediction, in: Proc. - 2016 31st Youth Acad. Annu. Conf. Chinese Assoc. Autom. YAC 2016, Institute of Electrical and Electronics Engineers Inc., 2017: pp. 324–328. https://doi.org/10.1109/YAC.2016.7804912.

[173] I. Sutskever, O. Vinyals, Q. V Le, Sequence to sequence learning with neural networks, ArXiv Prepr. ArXiv1409.3215. (2014). https://arxiv.org/abs/1409.3215.

[174] S. Akbari, F. Haghighat, Occupancy and occupant activity drivers of energy consumption in residential buildings, Energy Build. 250 (2021) 111303. https://doi.org/10.1016/J.ENBUILD.2021.111303.

[175] B. Yang, F. Haghighat, B.C.M. Fung, K. Panchabikesan, Season-Based Occupancy Prediction in Residential Buildings Using Machine Learning Models, E-Prime. (2021) 100003. https://doi.org/10.1016/J.PRIME.2021.100003.

[176] M.L. Zhang, Z.H. Zhou, ML-KNN: A lazy learning approach to multi-label learning, Pattern Recognit. 40 (2007) 2038–2048. https://doi.org/10.1016/j.patcog.2006.12.019.

[177] W. McKinney, Data structures for statistical computing in python, in: Proc. 9th Python Sci. Conf., Austin, TX, 2010: pp. 51–56. https://doi.org/https://doi.org/10.25080/majora-92bf1922-00a.

[178] U.S. Department of Energy (DOE), National Renewable Energy Laboratory (NREL), EnergyPlus, (n.d.). https://energyplus.net/documentation.

[179] Government of British Columbia, Determining ASHRAE 90.1-2010 Climate Zones, 2014. https://www2.gov.bc.ca/assets/gov/farming-natural-resources-and-industry/construction-industry/building-codes-and-standards/bulletins/b14-01_determining_ashrae_901-2010_climate_zones.pdf.

[180] Canada Weather Stats, Montréal Historical Heating Degree Days, (2021). https://montreal.weatherstats.ca/metrics/hdd.html.

[181] NAIMA Canada, Building Codes and Standards, (n.d.). https://www.naimacanada.ca/for-professionals/building-insulation/codes-standards/.

[182] ASHRAE, ASHRAE Standard 55-2017: Thermal environmental conditions for human occupancy, 2017. https://www.ashrae.org/technical-resources/bookstore/standard-55-thermal-environmental-conditions-for-human-occupancy.

[183] M. Aruldoss, T.M. Lakshmi, V.P. Venkatesan, A survey on multi criteria decision making methods and its applications, Am. J. Inf. Syst. 1 (2013) 31–43. http://pubs.sciepub.com/ajis/1/1/5/.

[184] R.-C. Tsaur, Decision risk analysis for an interval TOPSIS method, Appl. Math. Comput. 218 (2011) 4295–4304. https://doi.org/https://doi.org/10.1016/j.amc.2011.10.001.

[185] B. Vahdani, M. Salimi, M. Charkhchian, A new FMEA method by integrating fuzzy belief structure and TOPSIS to improve risk evaluation process, Int. J. Adv. Manuf. Technol. 77 (2015) 357–368. https://doi.org/https://doi.org/10.1007/s00170-014-6466-3.

[186] D. Das, R.K. Sharma, P. Saikia, D. Rakshit, An integrated entropy-based multi-attribute decision-making model for phase change material selection and passive thermal management, Decis. Anal. J. (2021) 100011.

[187] Y. Zhang, TOPSIS method based on entropy weight for supplier evaluation of power grid enterprise, in: 2nd Int. Conf. Educ. Reform Mod. Manag., 2015: pp. 334–337.

[188] H. Xu, Y. Deng, Dependent Evidence Combination Based on Shearman Coefficient and Pearson Coefficient, IEEE Access. 6 (2017) 11634–11640. https://doi.org/10.1109/ACCESS.2017.2783320.

[189] United Nations Environment Programme, Global status report for buildings and

construction, 2020. https://wedocs.unep.org/bitstream/handle/20.500.11822/34572/GSR_ES.pdf.

[190] Environmental Protection Agency, Indoor air quality, n.d. https://www.epa.gov/report-environment/indoor-air-quality.

[191] M. Killian, M. Kozek, Short-term occupancy prediction and occupancy based constraints for MPC of smart homes, IFAC-PapersOnLine. 52 (2019) 377–382. https://doi.org/10.1016/j.ifacol.2019.08.239.

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

[193] A. Beltran, V.L. Erickson, A.E. Cerpa, ThermoSense: Occupancy Thermal Based Sensing for HVAC Control, in: Proc. 5th ACM Work. Embed. Syst. Energy-Efficient Build. - BuildSys'13, Association for Computing Machinery (ACM), New York, New York, USA, 2013: pp. 1–8. https://doi.org/10.1145/2528282.2528301.

[194] W. Li, D.M. Becker, Day-ahead electricity price prediction applying hybrid models of LSTM-based deep learning methods and feature selection algorithms under consideration of market coupling, Energy. 237 (2021) 121543. https://doi.org/10.1016/J.ENERGY.2021.121543.

[195] C. Gunavathi, K. Premalatha, Performance analysis of genetic algorithm with kNN and SVM for feature selection in tumor classification, Int J Comput Electr Autom Control Inf Eng. 8 (2014) 1490–1497.

[196] O.S. Qasim, Z.Y. Algamal, Feature selection using particle swarm optimization-based logistic regression model, Chemom. Intell. Lab. Syst. 182 (2018) 41–46. https://doi.org/10.1016/J.CHEMOLAB.2018.08.016.

[197] D. Aggarwal, V. Bali, S. Mittal, An insight into machine learning techniques for predictive analysis and feature selection, Int. J. Innov. Technol. Explor. Eng. 8 (2019) 342–349.

[198] J. Chong, P. Tjurin, M. Niemelä, T. Jämsä, V. Farrahi, Machine-learning models for activity class prediction: A comparative study of feature selection and classification algorithms, Gait Posture. 89 (2021) 45–53. https://doi.org/10.1016/J.GAITPOST.2021.06.017.

[199] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Comput. Electr. Eng. 40 (2014) 16–28. https://doi.org/10.1016/j.compeleceng.2013.11.024.

[200] J. Li, K. Cheng, S. Wang, F. Morstatter, R.P. Trevino, J. Tang, H. Liu, Feature selection: A data perspective, ACM Comput. Surv. 50 (2017) 1–45.

[201] J. Gareth, W. Daniela, H. Trevor, T. Robert, An introduction to statistical learning: with applications in R, Spinger, 2013.

[202] V. Jakkula, D. Cook, Outlier detection in smart environment structured power datasets, in: Proc. - 2010 6th Int. Conf. Intell. Environ. IE 2010, 2010: pp. 29–33. https://doi.org/10.1109/IE.2010.13.

[203] S. Seo, A Review and Comparison of Methods for Detecting Outliers in Univariate Data Sets, Kyunghee University, 2006.

[204] C. Leys, C. Ley, O. Klein, P. Bernard, L. Licata, Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median, J. Exp. Soc. Psychol. 49 (2013) 764–766. https://doi.org/10.1016/j.jesp.2013.03.013.

[205] M. Kuhn, K. Johnson, Applied predictive modeling, Springer, 2013.

[206] H.-W. Cho, S.B. Kim, M.K. Jeong, Y. Park, T.R. Ziegler, D.P. Jones, Genetic algorithm-based feature selection in high-resolution NMR spectra, Expert Syst. Appl. 35 (2008) 967–975.

[207] J.H. Holland, Genetic algorithms, Sci. Am. 267 (1992) 66–73.

[208] O.H. Babatunde, L. Armstrong, J. Leng, D. Diepeveen, A genetic algorithm-based feature selection, (2014).

[209] Y. Fang, J. Li, A review of tournament selection in genetic programming, in: Int. Symp. Intell. Comput. Appl., Springer, 2010: pp. 181–192.

[210] T.-S. Lim, W.-Y. Loh, Y.-S. Shih, An empirical comparison of decision trees and other classification methods, (1998).

[211] J. Blank, K. Deb, pymoo: Multi-objective optimization in python, IEEE Access. 8 (2020) 89497–89509.

[212] Q. Xin, Optimization techniques in diesel engine system design, Diesel Engine Syst. Des. (2013) 203–296. https://doi.org/10.1533/9780857090836.1.203.

[213] H.Y. Alhammadi, J.A. Romagnoli, Process design and operation: Incorporating environmental, profitability, heat integration and controllability considerations, Comput. Aided Chem. Eng. 17 (2004) 264–305. https://doi.org/10.1016/S1570-7946(04)80063-4.

[214] A.P. Guerreiro, C.M. Fonseca, L. Paquete, The hypervolume indicator: Problems and algorithms, ArXiv Prepr. ArXiv2005.00515. (2020).

[215] D.W. Aha, R.L. Bankert, A comparative evaluation of sequential feature selection algorithms, in: Learn. from Data, Springer, 1996: pp. 199–206.

[216] J. Bemister-Buffington, A.J. Wolf, S. Raschka, L.A. Kuhn, Machine Learning to Identify Flexibility Signatures of Class A GPCR Inhibition, Biomolecules. 10 (2020) 454.

[217] M. Li, X. Yao, Quality evaluation of solution sets in multiobjective optimisation: A survey, ACM Comput. Surv. 52 (2019) 1–38.

[218] M.A. Wiering, M. Van Otterlo, Reinforcement learning, Adapt. Learn. Optim. 12 (2012) 729.

[219] M. Esrafilian-Najafabadi, F. Haghighat, Impact of predictor variables on the performance of future occupancy prediction: Feature selection using genetic algorithms and machine learning, Build. Environ. (2022) 109152. https://doi.org/10.1016/J.BUILDENV.2022.109152.

[220] Y. Zhang, X. Bai, F.P. Mills, J.C.V. Pezzey, Rethinking the role of occupant behavior in building energy performance: A review, Energy Build. 172 (2018) 279–294. https://doi.org/10.1016/J.ENBUILD.2018.05.017.

[221] Z. Zhang, A. Chong, Y. Pan, C. Zhang, S. Lu, K.P. Lam, A deep reinforcement learning approach to using whole building energy model for hvac optimal control, in: 2018 Build. Perform. Anal. Conf. SimBuild, 2018: pp. 22–23.

[222] American Society of Heating Refrigerating and Air-Conditioning Engineers (ASHRAE), 2009 ASHRAE Handbook - Fundamentals (SI Edition), ASHRAE, 2009.

[223] J.R. Dobbs, B.M. Hencey, Model predictive HVAC control with online occupancy model, Energy Build. 82 (2014) 675–684. https://doi.org/10.1016/j.enbuild.2014.07.051.

[224] W. Valladares, M. Galindo, J. Gutiérrez, W.-C. Wu, K.-K. Liao, J.-C. Liao, K.-C. Lu, C.-C. Wang, Energy optimization associated with thermal comfort and indoor air control via a deep reinforcement learning algorithm, Build. Environ. 155 (2019) 105–117.

[225] A. Gupta, Y. Badr, A. Negahban, R.G. Qiu, Energy-efficient heating control for smart buildings with deep reinforcement learning, J. Build. Eng. 34 (2021) 101739.

[226] Y. Wang, Y. Jia, Y. Tian, J. Xiao, Deep reinforcement learning with the confusion-matrix-based dynamic reward function for customer credit scoring, Expert Syst. Appl. 200 (2022) 117013. https://doi.org/10.1016/J.ESWA.2022.117013.

[227] C.J.C.H. Watkins, Learning from delayed rewards, (1989).

[228] C.J.C.H. Watkins, P. Dayan, Q-learning, Mach. Learn. 8 (1992) 279–292.

[229] H. Hasselt, Double Q-learning, Adv. Neural Inf. Process. Syst. 23 (2010) 2613–2621.

[230] B. Huchuk, S. Sanner, W. O'Brien, Development and evaluation of data-driven controls for residential smart thermostats, Energy Build. 249 (2021) 111201.

[231] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, Human-level control through deep reinforcement learning, Nature. 518 (2015) 529–533.

[232] T. Kobayashi, W.E.L. Ilboudo, T-soft update of target network for deep reinforcement learning, Neural Networks. 136 (2021) 63–71.

[233] A. Stooke, P. Abbeel, rlpyt: A research code base for deep reinforcement learning in pytorch, ArXiv Prepr. ArXiv1909.01500. (2019).

[234] Y. Fujita, P. Nagarajan, T. Kataoka, T. Ishikawa, Chainerrl: A deep reinforcement learning library, ArXiv Prepr. ArXiv1912.03905. (2019).

[235] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, ArXiv Prepr. ArXiv1412.6980. (2014).

[236] H. Van Hasselt, A. Guez, D. Silver, Deep reinforcement learning with double q-learning, in: Proc. AAAI Conf. Artif. Intell., 2016.

[237] M.A. Wiering, Explorations in efficient reinforcement learning, (1999).

[238] Y. Chen, L.K. Norford, H.W. Samuelson, A. Malkawi, Optimal control of HVAC and window systems for natural ventilation through reinforcement learning, Energy Build. 169 (2018) 195–205. https://doi.org/10.1016/J.ENBUILD.2018.03.051.

[239] M. Plappert, Keras-rl, GitHub Repos. (2016). https://github.com/keras-rl/keras-rl.

[240] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, W. Zaremba, Openai gym, ArXiv Prepr. ArXiv1606.01540. (2016).

[241] J.A. Candanedo, V.R. Dehkordi, M. Stylianou, Model-based predictive control of an ice storage device in a building cooling system, Appl. Energy. 111 (2013) 1032–1045. https://doi.org/10.1016/j.apenergy.2013.05.081.

[242] C. Wang, B. Wang, M. Cui, F. Wei, Cooling seasonal performance of inverter air conditioner using model prediction control for demand response, Energy Build. 256 (2022) 111708. https://doi.org/10.1016/J.ENBUILD.2021.111708.

[243] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, TensorFlow: A System for Large-Scale Machine Learning, in: 12th USENIX Symp. Oper. Syst. Des. Implement., 2016: pp. 265–283.

[244] D. Hsu, Comparison of integrated clustering methods for accurate and stable prediction of building energy consumption data, Appl. Energy. 160 (2015) 153–163. https://doi.org/10.1016/J.APENERGY.2015.08.126.

[245] Y. Wei, X. Zhang, Y. Shi, L. Xia, S. Pan, J. Wu, M. Han, X. Zhao, A review of data-driven approaches for prediction and classification of building energy consumption, Renew. Sustain. Energy Rev. 82 (2018) 1027–1047. https://doi.org/10.1016/j.rser.2017.09.108.

[246] G. Liu, J. Yang, Y. Hao, Y. Zhang, Big data-informed energy efficiency assessment of China industry sectors based on K-means clustering, J. Clean. Prod. 183 (2018) 304–314. https://doi.org/10.1016/J.JCLEPRO.2018.02.129.

[247] Y. Ding, W. Chen, S. Wei, F. Yang, An occupancy prediction model for campus buildings based on the diversity of occupancy patterns, Sustain. Cities Soc. 64 (2021) 102533. https://doi.org/10.1016/J.SCS.2020.102533.

[248] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proc. Fifth Berkeley Symp. Math. Stat. Probab., Oakland, CA, USA, 1967: pp. 281–297.

[249] P. Westermann, C. Deb, A. Schlueter, R. Evins, Unsupervised learning of energy signatures to identify the heating system and building type using smart meter data, Appl. Energy. 264 (2020) 114715. https://doi.org/10.1016/J.APENERGY.2020.114715.

[250] R. Tavenard, J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak, M. Rußwurm, K. Kolar, Tslearn, A Machine Learning Toolkit for Time Series Data., J. Mach. Learn. Res. 21 (2020) 1–6.

[251] P.J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987) 53–65. https://doi.org/10.1016/0377-0427(87)90125-7.

[252] T. Le, M.T. Vo, T. Kieu, E. Hwang, S. Rho, S.W. Baik, Multiple electric energy consumption forecasting using a cluster-based strategy for transfer learning in smart building, Sensors. 20 (2020) 2668.

[253] M.E. Taylor, P. Stone, Transfer learning for reinforcement learning domains: A survey., J. Mach. Learn. Res. 10 (2009).