

Article

# Energy Management Model for HVAC Control Supported by Reinforcement Learning

Pedro Macieira, Luis Gomes  and Zita Vale \* 

GECAD—Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, Polytechnic of Porto (P.PORTO), P-4200-072 Porto, Portugal; picma@isep.ipp.pt (P.M.); lfg@isep.ipp.pt (L.G.)

\* Correspondence: zav@isep.ipp.pt

**Abstract:** Heating, ventilating, and air conditioning (HVAC) units account for a significant consumption share in buildings, namely office buildings. Therefore, this paper addresses the possibility of having an intelligent and more cost-effective solution for the management of HVAC units in office buildings. The method applied in this paper divides the addressed problem into three steps: (i) the continuous acquisition of data provided by an open-source building energy management systems, (ii) the proposed learning and predictive model able to predict if users will be working in a given location, and (iii) the proposed decision model to manage the HVAC units according to the prediction of users, current environmental context, and current energy prices. The results show that the proposed predictive model was able to achieve a 93.8% accuracy and that the proposed decision tree enabled the maintenance of users' comfort. The results demonstrate that the proposed solution is able to run in real-time in a real office building, making it a possible solution for smart buildings.

**Keywords:** building energy management systems; HVAC control; Internet of Things; occupancy prediction; reinforcement learning



**Citation:** Macieira, P.; Gomes, L.; Vale, Z. Energy Management Model for HVAC Control Supported by Reinforcement Learning. *Energies* **2021**, *14*, 8210. <https://doi.org/10.3390/en14248210>

Academic Editors: Ana-Belén Gil-González and Fabrizio Ascione

Received: 26 October 2021  
Accepted: 3 December 2021  
Published: 7 December 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Smart grids enable the active participation of end-users and allow them to actively manage their demand using demand-side management [1,2]. By managing home available energy resources and loads, the end-users are able to reduce energy costs [3], maximize the use of renewable generation [4], participate in the smart grid [5], and transact energy with other end-users [6].

The ability to manage their own resources enables end-users to take control over their facilities and promote the dissemination of smart buildings [7]. Actions for demand-side management and for the active participation of end-users can also be achieved using internet of things (IoT) devices [8]. The possibilities of energy management are vast and they represent a good opportunity for end-users to enable the reduction of energy costs and optimization of energy usage [9].

One of the biggest contributions to the energy consumption in buildings comes from heating ventilating air conditioning (HVAC) systems [10]. Therefore, the control of HVAC systems is frequently found in the literature as a way to reduce energy costs and greenhouse gas emissions [11,12]. The control of HVAC can also be used for the active participation of end-users in smart grids [13]. Because of their characteristic preconditioning time, i.e., the preheating or cooling time of HVAC, the existing models for HVAC control usually results in planned action ahead in time, demanding the use of prediction models to forecast contexts. The use of prediction models, in smart grids, is common, in particular, to forecast the consumption of end-user and individual loads, the generation of renewable resources, and the flexibility of end-users [14].

The optimization of HVAC systems, as any other electrical load placed in the end-user facility, needs to consider the users' preferences and needs [15]. It is then necessary to

balance the users' comfort and the energy cost reduction. This is where prioritization techniques come in, allowing energy consumption to be reduced and avoiding unnecessary electricity expenses [16].

This paper proposes a novel model for HVAC control using a prediction reinforcement learning algorithm and a decision tree for the consideration of building context, energy costs, and preconditioning time. The prediction model is used to predict occupancy in the building ahead in time. The occupancy prediction is then used in a decision tree to control of the HVAC units, considering the energy prices and the current temperature of the building. The proposed solution can intelligently control the HVAC units and promote the reduction of energy costs.

The proposed solution, comprised of the prediction model and the decision tree, was deployed in a real office building using a building energy management system (BEMS) that was implemented based on the open-source platform Home Assistant (<https://www.home-assistant.io/>, accessed on 6 December 2021). This BEMS enabled the integration of multiple IoT devices and the implementation of the proposed models in a real building, allowing a continuous operation. The proposed solution addresses some of the limitations that were identified in the literature review, such as continuous learning, the consideration of several contexts in the building, and the users' comfort.

This paper is structured as follows. After this first introductory section, related works regarding the management of HVAC systems, namely occupancy-based solutions, are presented in Section 2. In Section 3, several solutions to develop a BEMS are presented and the proposed BEMS, based on Home Assistant, is described. Section 4 presents the proposed methodology including the prediction model and the decision tree. The results of the case study, using a real building, are described in Section 5. Section 6 presents the discussion of results, while the main conclusions are presented in Section 6.

## 2. Related Works

The occupancy of the building and/or of different building zones can be used by energy management systems to promote the contextual optimization of resources [15]. The information regarding occupancy allows systems to better understand the building context and provide better resource optimization [17]. The exact occupation of the building and the location of people is hard to achieve, but estimates can be obtained using several techniques, using equipment already installed in the building or that requires new equipment to be installed [18]. In [19], a machine learning classification model is proposed for week-ahead occupancy prediction using real-time smart meter data. Applied to HVAC systems, [20] proposed a control model based on students' location inside auditoriums. This model uses Wi-Fi data to determine the number of connected devices and enables a noninvasive approach that does not require the installation of new equipment. In [21], a model for HVAC control inside buildings is proposed using a rule-based approach supported by a deep learning algorithm that predicts the preconditioning time. These solutions address the prediction and identification of occupancy in buildings. However, they lack the ability to consider multiple contexts. The contextual reinforcement learning predictive model proposed in this paper addresses the dynamics of the building usage considering several contexts.

A reinforcement learning model for occupant behavior is proposed in [22] using a Q-learning model to allow automated control of the building's thermostat. This work compared the prediction of the reinforcement learning model with an artificial neural network (ANN) prediction model. The results showed that the ANN had better results. In [23] the use of reinforcement learning models to directly control the HVAC units demonstrated an increase in energy costs, even when dealing with on/off control. However, the best results were achieved when the model was used to control five flow levels.

Other contextual data, beyond occupancy, can be used to control HVAC systems [24]. A model for HVAC control considering not only the building occupancy but also the characterization of the building, namely the mean radiant temperature of building rooms

is proposed in [25]. In [26], it is proposed the use of IoT devices to obtain contextual information regarding outdoor temperature to improve the efficiency of HVAC control and achieve lower payback times, when compared to occupancy-based models. However, these solutions lack the ability to learn from historical data, and more importantly, the ability to be able to continuously learn. The proposed solution addresses this limitation by conceiving a reinforcement learning model.

The control and management of HVAC systems are usually used in the energy domain as a way to reduce energy costs [27], due to the high energy consumption of HVAC equipment. A day-ahead HVAC control model for industrial buildings, to decrease energy costs considering on-peak and off-peak tariffs and weather conditions, is proposed in [28]. A multiperiod optimization model for supervisory control and data acquisition (SCADA) systems is proposed in [29] to minimize energy costs while complying with users constraints. In [30], HVAC systems are controlled using market-based transactive controls in commercial buildings to promote grid balance and stability. Although energy costs are an important aspect of the building, the user comfort needs to be considered, as well as the building's context. The proposed solution addresses the users' comfort considering ahead HVAC control signals to prepare the building for the users' arrival while trying to minimize energy costs. The context of the building is considered in the proposed solution, for instance, the HVAC units' control considers the status of the windows.

The control of HVAC units can also be done to improve air quality. In [31], a computer vision-based system is used to get the occupancy of a room and a neural network is used to classify the users' activities. This solution enables the prediction of CO<sub>2</sub> levels and controls the HVAC unit to prevent potentially dangerous situations. In [32], a solution for commercial buildings is proposed where a scalable model, based on multiagent deep reinforcement learning, is used to minimize energy costs while considering user comfort and air quality.

More complex solutions can also be implemented to enable the combined management of multi-HVAC systems. In [33], a data-driven mathematical model using contextual sensor data is used to optimize multi-HVAC systems that serve the same space. In [34], several optimization models are compared for multi-HVAC systems management considering the outdoor temperature, building ambient temperature, and occupancy. The compared models were a multiobjective genetic algorithm, two nondominated sorting genetic algorithms, an optimized multiobjective particle swarm optimization, a speed-constrained multi-objective particle swarm optimization, and a random search. The authors were not able to identify a unique best solution but presented the benefits of each model.

### 3. Building Energy Management System Based on Home Assistant

A BEMS can be installed in a variety of ways, normally this is done using dedicated, commercial platforms such as DEXMA Energy Intelligence (<https://www.dexma.com/>, accessed on 6 December 2021) or Entronix (<https://entronix.io/>, accessed on 6 December 2021). However, for the purposes of this paper, a home automation system was used, enabling the combination of Internet of Things (IoT) devices with the ability to gather data and affect the environment around them with rules created by the user to manage the smart home.

There are a variety of home automation system software solutions, both open and closed source. The biggest open-source solutions available are:

- openHAB (<https://www.openhab.org/>, accessed on 6 December 2021), created by Kai Kreuzer, which is built using Java and was initially released in 2010;
- Home Assistant, created by Paulus Schoutsen, based on Python and configurable with the use of YAML files, which was first released in 2013;
- Domoticz (<https://www.domoticz.com/>, accessed on 6 December 2021), created by Gizmocuz, is written in C++ and configurable with Blockly or Lua, and with the first release date at the end of 2012.

There are also a wide variety of commercial closed source solutions, with the biggest ones being owned by large players:

- Google Assistant (<https://assistant.google.com/>, accessed on 6 December 2021), Google’s solution for home assistant software, allows the use of Google Assistant software together with Google Home devices to control a house with voice commands and using a graphical user interface (GUI). It also enables the customization of automation rules;
- Alexa (<https://www.amazon.com/b?ie=UTF8&node=21576558011>, accessed on 6 December 2021), Amazon’s solution for home automation, works similarly to Google Assistant, by using an Alexa device to allow the user to control their smart home with voice commands and with GUI;
- HomeKit (<https://www.apple.com/ios/home/>, accessed on 6 December 2021), developed by Apple, allows users to integrate their houses with the Apple Ecosystem, allowing the use of voice commands through Siri but also providing a GUI for more powerful smart home management.

For the purposes of this paper, Home Assistant was chosen due to its many advantages such as being open-source, easily customizable with simple YAML files, and the powerful but still simple set-up GUI.

First, some IoT devices were installed and configured in a docker containerized Home Assistant installation, with their uses described in Table 1. The IoT devices integrated can be seen as actuators and sensors, depending on the type of actions they allow. Additionally to market-available IoT devices, the Home Assistant configuration also integrates a SCADA system, developed by the researchers of the authors’ research center, that was already present in the building. This SCADA system is represented in Table 1 as GECAD API. This API enables the reading of other IoT devices that are already integrated into the SCADA system by using REST-based HTTP requests. The last device identified in Table 1 is a simulated sensor that was integrated into Home Assistant as a file sensor, enabling the reading of a file and the publishing of that information. The file sensor is a dataset of real electricity prices that were obtained from MIBEL, the Iberian electricity wholesale market.

**Table 1.** Devices configured for the Home Assistant installation.

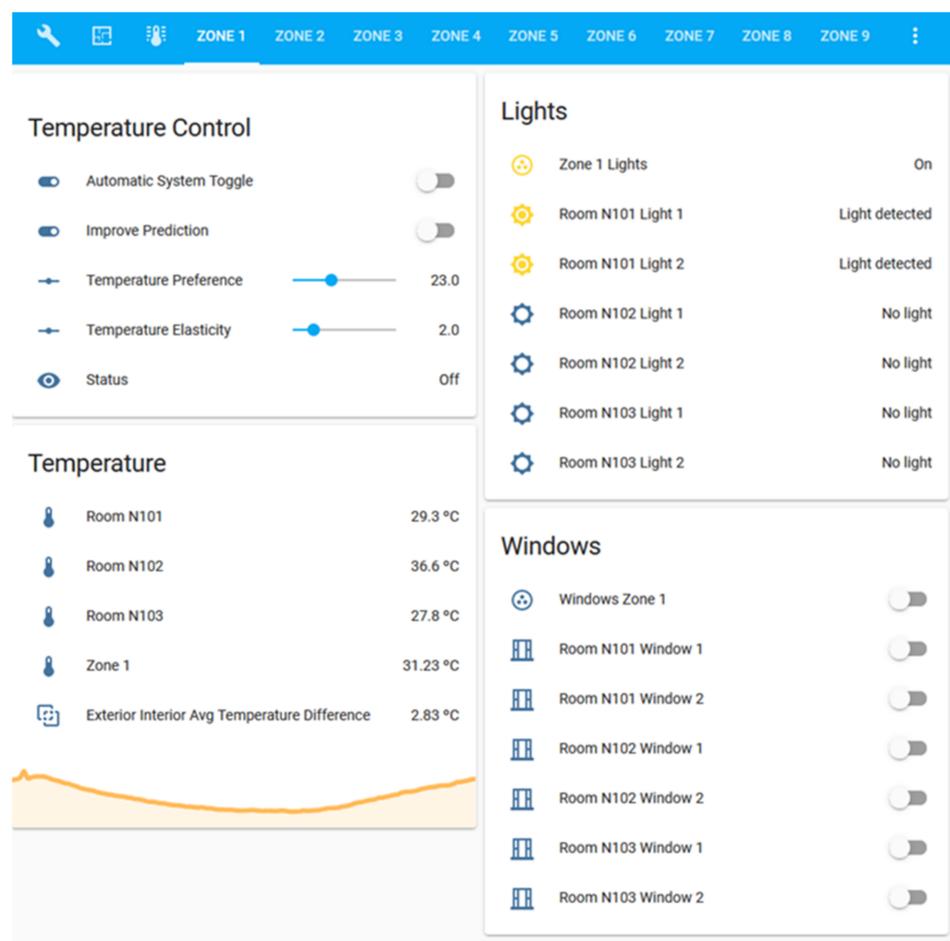
Device Name	Monitoring Data	Actions
RM Pro +	Any IR or RF signal	Emission of IR or RF signal
Sonoff Pow R2	Current (A), power (W), voltage (V), load status (on/off)	Turn on/off
D-Link DSP-W115	Load status (on/off)	Turn on/off
D-Link DSP-W215	Power (W), total power consumption (kWh), temperature (°C)	Turn on/off
Sonoff RF Bridge	N/A	N/A
Sonoff Sensor DW1	Door/window open	triggered/A
GECAD API (REST Sensors)	Exterior temperature (°C)	N/A
	Interior temperature of individual rooms (°C)	
	Light switch states (%)	
File Sensor (JSON Source)	Electricity price on a specific month, time, day, and hour (EUR/kWh)	N/A

To manage the energy of a building in an intelligent way using Home Assistant, complex logic needs to be integrated. However, Home Assistant has some constraints, namely the fact that it is configured by YAML files, which can turn simple scripts into a large amount of complex virtual sensors that are tightly interconnected and have little use outside of single automation. This difficult the setting-up of the installation and makes it very hard to share the logic with other Home Assistant installations.

To solve this issue, Pyscript (<https://github.com/custom-components/pyscript>, accessed on 6 December 2021), an add-on to Home Assistant, has been used for the work presented in this paper. This add-on can import standard Python libraries and interact with Home Assistant triggers and events when IoT devices or Home Assistant itself create

them. In this way, the user can create applications that can be configured with YAML when there is a need to replicate them multiple times, and, most importantly, it allows the use of Python to write the scripts that can interact in real-time with Home Assistant. This enables the Home Assistant to be integrated with all kinds of external services that can manage its resources.

Home Assistant was configured with ten views, which resulted in tabs inside the Home Assistant dashboard, one for each building zone, and a general view with data that is not specific to any zone. The building is divided into nine zones, where each has from two to three rooms that can be offices, meetings rooms, server rooms, or laboratories. Figure 1 presents the aspect of the interface for zone 1 of the considered building.



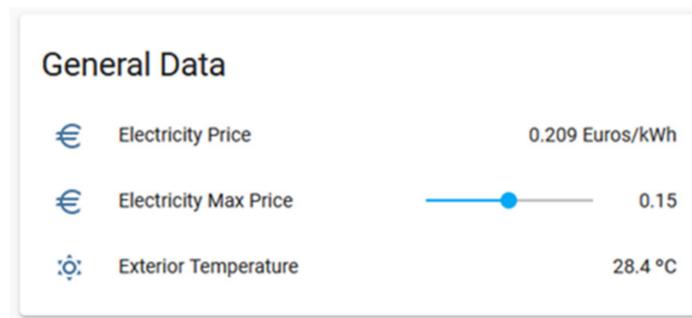
**Figure 1.** The Home Assistant interface for Zone 1.

The view for each zone contains multiple panels for IoT device monitoring and control. It has a ‘temperature control’ panel where the users can activate and control the proposed HVAC control model, allowing:

- The activation of the proposed HVAC control model;
- The activation of continuous learning, to enable the reinforcement learning prediction model to learn;
- The setting of user temperature preferences;
- The setting of user temperature elasticity;
- The current status of the system (e.g., “On, heating up” or “Off” or “Off, the window is open”).

The general view of the Home Assistant interface can be seen in Figure 2 where electricity market-related data are presented. In the general view, it is also possible to

control the maximum price for the system to run autonomously, to save money, and avoid turning on the energy-expensive HVAC systems when energy costs are too high.



**Figure 2.** The Home Assistant interface for general data.

#### 4. Method and Proposed Solution

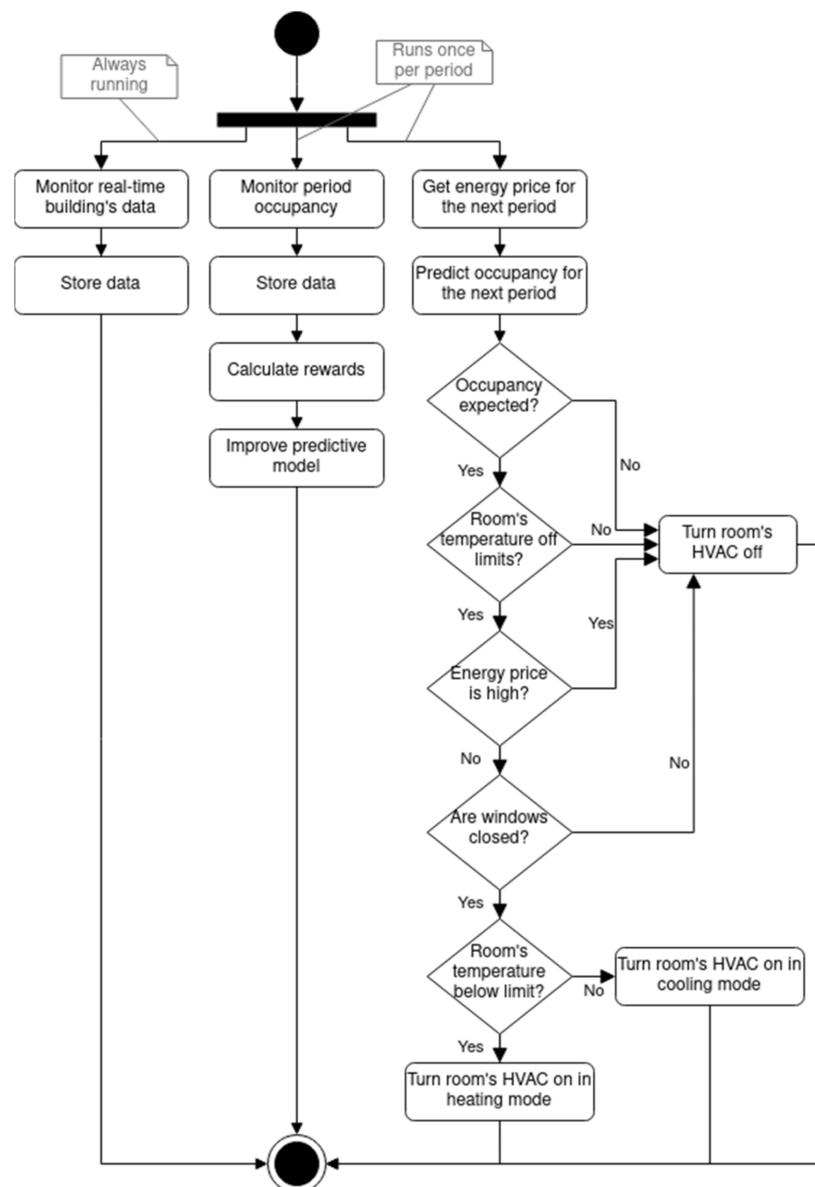
Currently, most HVAC systems are manually controlled by users who are on-site and find themselves at an uncomfortable temperature level. The user must manually configure the HVAC system and wait for it to effectively adjust the temperature up or down to a more comfortable level. Alternatively, there is a central system that can be configured with sensors to detect temperature changes and automatically control the HVAC system. However, central systems also raise some issues, such as the waiting time until the HVAC system successfully regulates itself back to a comfortable temperature. An even bigger issue is the fact that it may be turning the HVAC system on when there are no users in the room/building, thus resulting in the waste of energy.

The proposed solution considers three components that will enable the individual intelligent management of each HVAC unit present in the building. The first component is the deployment of the open-source BEMS that is presented in Section 3. The BEMS enables the continuous monitoring and control of the building's resources and the storing of the data. The historical data of the BEMS is fed to a predictive model supported by machine learning. The predictive model is proposed to enable the prediction of office space usage. The last component of the proposed solution is the proposed decision tree that will act on the HVAC units considering the prediction, current environmental context, the limits and preferences of the users, and the current energy prices.

Figure 3 shows the complete flowchart of the proposed solution. The flowchart represents three threads that are executed. The first one is executed in every instant to allow Home Assistant to have continuous monitoring of the building. The second one is used to improve the reinforcement learning algorithm used as a predictive model, and it is executed once every period. This flow monitors the room's occupancy, stores the data, and recalculates the rewards. In the last flow, the control over the HVAC units is performed. The last thread is executed once every period and is able to collect the energy price, predict the room's occupancy, apply the decision tree, and control the HVAC units accordingly.

This paper proposes the use of a contextual reinforcement learning algorithm to predict the occupancy of the rooms of a building. The use of reinforcement learning enables the proposed model to have a continuous learning process, suitable to adjust the prediction to new routines or the addition of team members working in the same office space. Moreover, the proposed model is a contextual reinforcement learning algorithm that is able to have predictions considering the context of the office.

The reinforcement learning algorithm's configuration and tuning are described and a test between linear and neural models is done to obtain the best learning accuracy. The paper also proposes a novel decision tree to enable the ahead control of HVAC units taking into consideration energy prices and user preferences, namely the desired temperature. This proposed methodology was integrated into the proposed building energy management system, presented in Section 3.



**Figure 3.** Solution flowchart.

#### 4.1. Reinforcement Learning Model for Occupancy Forecast

The proposed HVAC control model is based on the reinforcement learning algorithm published in [35], enabling the use of contextual reinforcement learning, used in this paper to predict the building's occupancy. In this paper, the algorithm was configured and trained to enable queries, using a specific date and time, to return an occupancy prediction as a response. The combination of this occupancy model with the previously described home automation system will allow for the setup of an intelligent HVAC system management system capable of correctly setting the temperature while still avoiding the situation of energy waste.

The biggest advantage of this model is that when the model wrongfully predicts the occupancy of the building, it will also be capable of learning from its mistakes and adapting itself over time, eventually figuring out the new pattern of occupation. This enables the system to self-adjust itself to new realities of the building, such as the changing of usage patterns created by new collaborators.

The model is executed every 30 min, every day of the week in a continuous fashion. Every time it is queried it will attempt to predict if a group of rooms, henceforth called a

zone, will be occupied or not on the next 30 min period, thus giving ample time for the HVAC to adjust the room temperature, if needed.

#### 4.2. Hyperparameter Tuning

After the initial training of the model, it was further refined with a process of hyperparameter tuning. In this way, the model can achieve the highest possible accuracy score and the correct amount of learning/exploring so it can keep high accuracy scores in the future. For this, a specific set of parameters were tuned:

- The number of hidden layers of the neural model;
- The size of each hidden layer;
- The learning rate;
- The decay of the learning rate;
- $\alpha$  and  $\beta$  which are used to calculate the inverse  $\gamma$  and Gaussian inference used by the model to perform exploration;
- The  $\lambda$  prior.

#### 4.3. Decision Tree

After the prediction of occupancy, a decision tree is used to identify the need for HVAC control. The decision tree is shown in Figure 3, being executed after the predictive model. There are three possible actions resulting from the decision tree: turn off the HVAC unit of the room, turn it on in cooling mode, and turn it on in heating mode.

The decision tree is used to control the HVAC system considering several contextual variables. The decision tree is applied 15 min ahead and during the targeted period, enabling a continuous monitoring and feedback loop between the context and the HVAC control. The decision tree was constructed to consider more parameters than just whether the zone will be occupied or not, taking into consideration:

- Current room temperature to check if there is even a need for the HVAC to be turned on in the first place;
- If there are open windows which would represent a significant waste of energy due to the outside–inside temperature differential;
- If the current electricity price is too high;
- Whether it will turn the heating on or the cooling on.

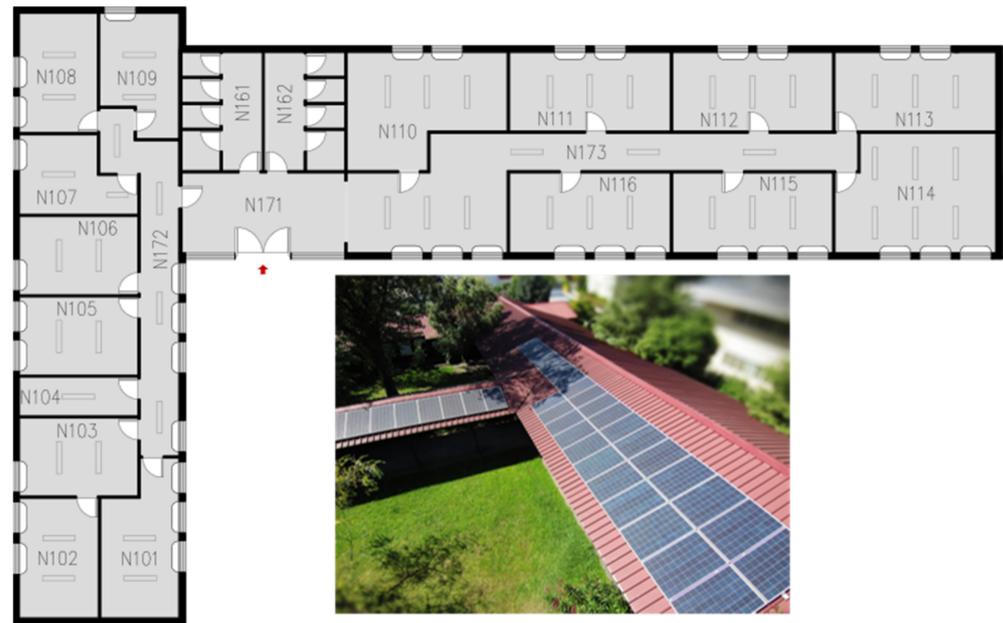
The variables used in the decision tree are a combination of sensor data and user specifications. The users of each office space are responsible to agree, converge, and define the temperature comfortable limits. The building's owner needs also to define the energy price threshold used for the decision tree.

### 5. Case Study

The proposed solution was deployed in a multiple office building with IoT sensors that were configured with the used home automation software (i.e., Home Assistant). The building is organized into nine zones. The proposed HVAC model was tested on Zone 1, which contains three office rooms named N101, N102, and N103. An overview of the floorplan of the building and an aerial photo (Figure 4) demonstrates how the building is laid out.

#### 5.1. Training Dataset

The dataset used to train the learning model consists of one-year data collected in 2019. The decision to use the year 2019 data is due to the SARS-CoV-2 restrictions that have been in use since March 2020. In this way, during 2020 the building had usage patterns radically different from the ones occurring during a normal year. The structure of the dataset can be seen in Table 2.



**Figure 4.** Aerial view of the building and its floorplan.

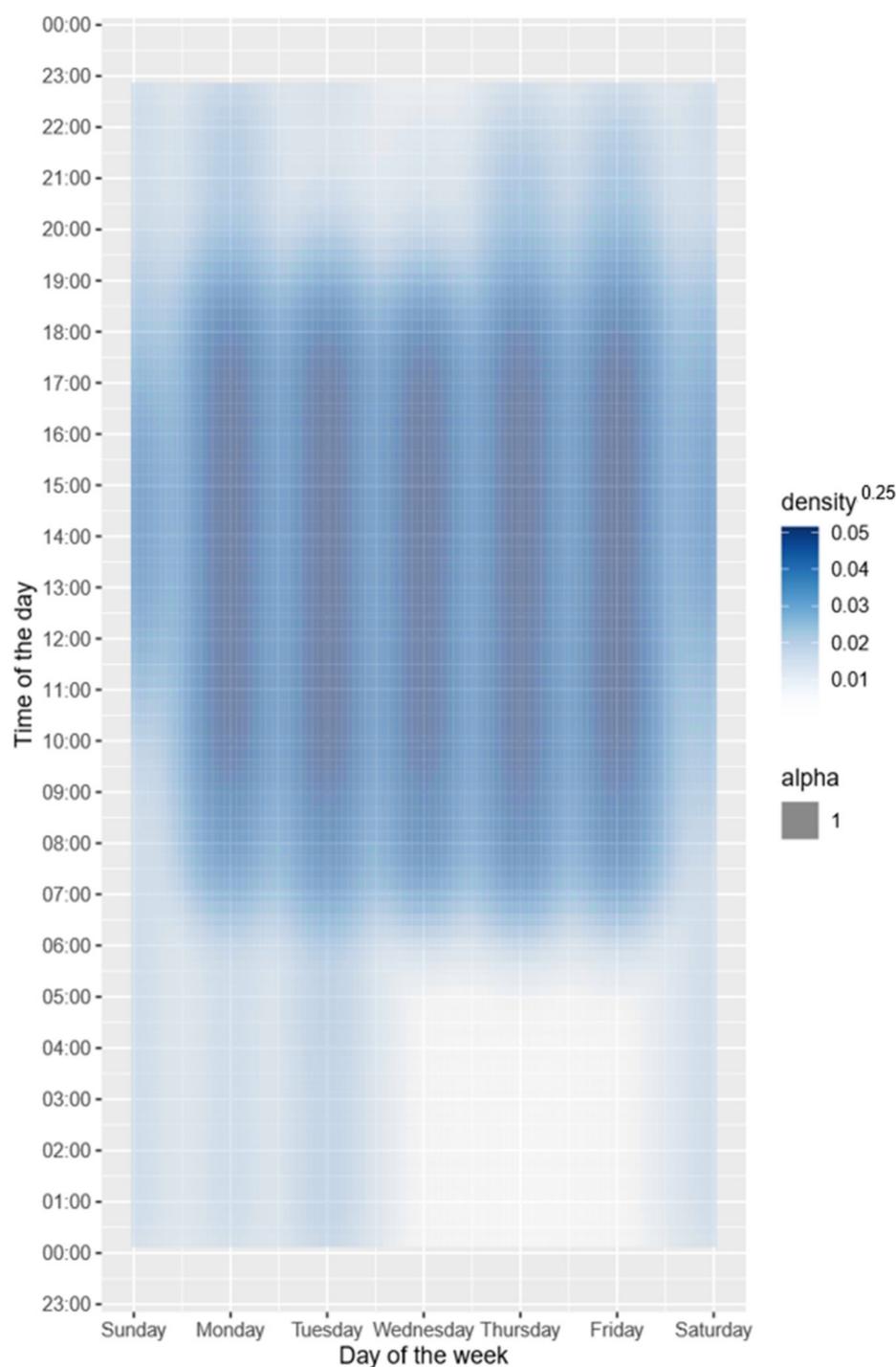
**Table 2.** Breakdown of the preprocessed dataset.

Column	Description
Weekday	The day of the week as an integer (i.e., from 0 to 6)
Time	The time that marks the start of the interval in seconds counting from midnight (i.e., from 0 to 86,400)
Occupied	A Boolean representing whether the zone was occupied during the interval or not (i.e., true or false)

The dataset consists of 17,520 entries, each entry corresponding to a single 30-min interval of the year 2019. The occupation value was not directly provided by any IoT integrated into the proposed solution but has been inferred from the ceiling lamp consumption. If during the 30-min interval the power draw of any lamp in the zone was higher than 0 watts, the zone was assumed to be occupied, which is a sensible approach in face of the building usage patterns.

The initial dataset was then split into two. A subdataset of 16,176 entries (92.32%) was used for the training of the reinforcement learning model. The remaining 1344 entries (7.68%), representing 4 weeks, were used for evaluation. The split of the dataset was done by weekly periods, meaning that the evaluation subdataset has four weeks of random months, but where each counts with seven sequential days.

As the building is not usually open outside business hours, the majority of the entries have the building marked as unoccupied, meaning that the used dataset was unbalanced. A density graph of the occupation of the building elaborated from this dataset can be viewed in Figure 5.



**Figure 5.** Occupancy graph of the building during 2019.

### 5.2. Forecast Errors Evaluation

Every time the model is executed, it takes in two input parameters, the current day of the week represented as an integer, ranging from 0 representing Monday to 6 representing Saturday, and the number of seconds that have passed since midnight of the current day up until the desired 30-min prediction interval.

Two different models were evaluated based on [36]: a linear model that simply attempted to regress the inputs and predict an output, and a more complex neural linear model that was able to learn a representation of the inputs and make a prediction based on that representation.

The training dataset was used on both reinforcement learning models to compare and evaluate their results. This evaluation step allowed the identification of the model that best performs under our case study.

The linear model was faster to train but it was limited by its ability to represent the problem and accurately predict whether the zone would be occupied or not, mostly predicting that it would be empty. Because an unbalanced dataset was used, the linear model managed to achieve an accuracy of around 68% as can be seen in its confusion matrix in Table 3. The neural linear model, using its default configuration of hyperparameters, was able to learn from the training dataset and achieve an accuracy of 92% (Table 4). From this step forward, the linear model was discarded due to its low accuracy score and the focus was set on improving the result of the neural linear model.

**Table 3.** Confusion matrix for the linear prediction model.

True Label	Predicted Label	
	Empty Zone	Occupied Zone
Empty zone	855	64
Occupied zone	373	80
Accuracy	68.15%	

**Table 4.** Confusion matrix for the neural linear prediction model.

True Label	Predicted Label	
	Empty Zone	Occupied Zone
Empty zone	873	46
Occupied zone	60	393
Accuracy	92.27%	

### 5.3. Hyperparameter Tuning Results

To tune the hyperparameters, Python's Optuna library was used due to its flexibility and learning curve, along with the ability to provide useful data and information at the end of a hyperparameter tuning session.

First, a selection of hyperparameters to tune was made, and then a range of appropriate values was chosen for each of the hyperparameters to be tuned. Then, using a loop, a random combination of hyperparameters was chosen and the model was trained and evaluated. To avoid wasting time in nonviable trials, the algorithm continuously checks the current accuracy and if it were lower than the average accuracy of the previous trials the entire trial would be pruned and the next trial starts.

In this case study, a hyperparameter tuning session with 500 trials was executed. From the obtained results, more sensible ranges were defined to achieve more accurate models. After another 400 trials, the most accurate model to come out of the hyperparameter tuning sessions was able to achieve an accuracy of 93.8%.

The configuration with the best result uses a network with three hidden layers with 26 nodes in the first and in the second layers, and with 12 nodes for the last layer. The chosen network had a learning rate of 0.03473 and a lambda of 0.20787.

### 5.4. HVAC Control Test

The control of HVAC units was done using Broadlink RM pro devices that were integrated into the Home Assistant solution. The Broadlink RM pro devices were located in each room of the building in Figure 4. The control of HVAC units was done, by the proposed solution, considering the decision tree shown in Figure 3. The decision tree was continuously executed in the proposed BEMS. However, the prediction value was only updated every 15 min and it considered the predicted presence of users for the next 15-min

period. Therefore, the control was done ahead of time, i.e., to prepare the room for the next period by considering the time that HVAC units take to reach the desired temperature.

For this case study, the business day of Friday, 27 August 2021, was considered. Figure 6 shows the results for 24 h. The orange area represents the hours where users are predicted to be inside the office N101, part of Zone 1. The case study data considered two temperature limits of 25 and 23 °C, with a range of +3 °C, meaning that at the beginning of the day the temperature limit was from 25 °C to 28 °C, and it was changed to a limit between 23 °C and 26 °C. The energy threshold was set to 0.24 EUR/kWh.

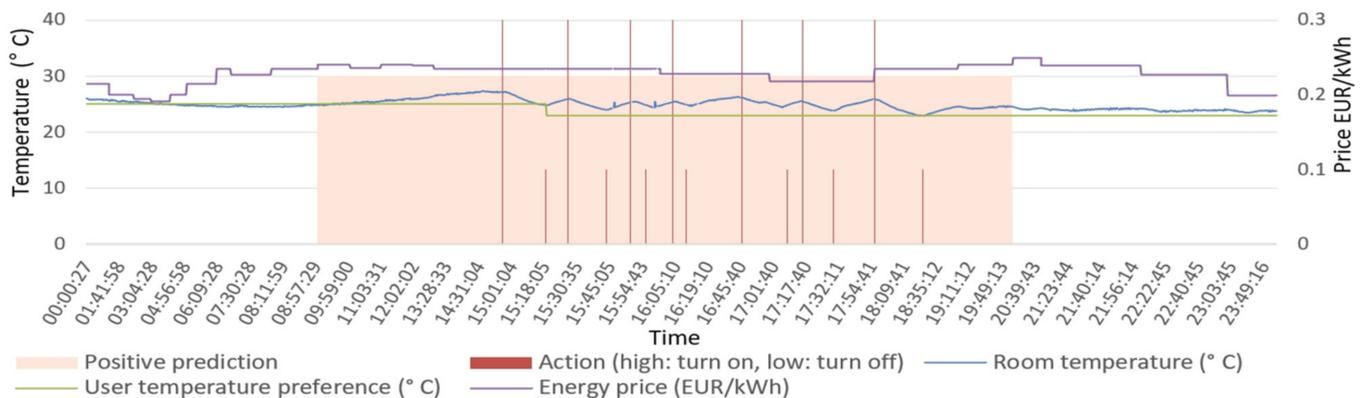


Figure 6. HVAC control signals for N101 during 27 August 2021.

## 6. Discussion

The results shown in Figure 6 indicate that the HVAC unit of room N101 was turned on seven times and turned off seven times over the 24 h. As can be seen, these actions happen during the hours where the predictive model forecasts the presence of users inside the office space of room N101, i.e., from 9:00 a.m. to 8:00 p.m. The turning on of the HVAC units also matched the increase of temperature above the upper user limit, while the turning off matched the below limit.

At 3:18 p.m., the user updated the temperature limit to a range between 23 °C and 26 °C, changing the system performance. This action tested the system's ability to adjust to the users' needs. However, as future improvement, the authors suggest the use of building thermal modeling to increase the intelligence of the management model.

The windows of room N101 were open between 4:07 p.m. and 4:45 p.m., leading to an early turn-off of the HVAC unit at 4:08 p.m. when the temperature was 24.93 °C (i.e., 1.93 °C above the lower limit of 23 °C). When the windows were closed, the HVAC almost immediately started once again to decrease the room's temperature.

The system demonstrated its ability to continuously manage the HVAC unit considering the current context of the room while also considering the predictive data of the proposed model. The control of the temperature inside the room was not only made according to the current context. If the predictive model did not forecast the presence of users during the next period, no control would be made in the HVAC units. All control signals, to turn the unit on, were made because the predictive model forecasted the presence of users during the next period.

The BEMS, based on the open-source solution of Home Assistant, was able to provide real-time monitoring and control while providing historical data to train the reinforcement learning algorithm. The proposed predictive model and the decision tree were deployed in BEMS using Python language and were able to cooperate with the BEMS to provide a continuous intelligent operation without the need for manual actions.

## 7. Conclusions

This paper proposes a novel model for the control of HVAC units based on a prediction model and a decision tree. The control of the units is performed ahead of time to minimize the units' waiting times (i.e., the time they take to reach the desired temperature). The prediction model uses a reinforcement algorithm that can provide continuous learning according to several contexts. The high accuracy of the model, 93.8%, demonstrates the ability of such algorithms to be used in smart buildings to efficiently manage energy loads and resources across different contexts.

Using the prediction result, a decision tree is proposed to consider the current context and the predicted context, inside a building's zone, to control the HVAC units. The proposed model was tested and evaluated using real data from one year in an office building. The validation of the proposed solution for 24 h is presented to demonstrate the use of the proposed solution as a whole.

Moreover, the work described in the paper demonstrated the ability to have open-source-based automation solutions to deploy complex models of artificial intelligence and energy management to operate in real-time and also ahead of time. This combination of technologies provides a complete tool to test and validate energy management models in real buildings.

**Author Contributions:** Conceptualization, P.M., L.G. and Z.V.; methodology, P.M., L.G. and Z.V.; software, P.M.; validation, P.M. and L.G.; formal analysis, P.M. and L.G.; investigation, P.M.; resources, Z.V.; data curation, P.M.; writing—original draft preparation, P.M. and L.G.; writing—review and editing, L.G. and Z.V.; visualization, P.M. and L.G.; supervision, L.G. and Z.V.; project administration, Z.V.; funding acquisition, Z.V. All authors have read and agreed to the published version of the manuscript.

**Funding:** The present work has received funding from European Regional Development Fund through COMPETE 2020—Operational Programme for Competitiveness and Internationalisation through the P2020 Project TloCPS (ANI | P2020 POCI-01-0247-FEDER-046182), and has been developed under the EUREKA—ITEA3 Project TloCPS (ITEA-18008).

**Institutional Review Board Statement:** Not Applicable.

**Informed Consent Statement:** Not Applicable.

**Data Availability Statement:** Not Applicable.

**Acknowledgments:** The authors acknowledge the work facilities and equipment provided by GECAD research center (UIDB/00760/2020, UIDP/00760/2020) to the project team.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Lezama, F.; Soares, J.; Canizes, B.; Vale, Z. Flexibility management model of home appliances to support DSO requests in smart grids. *Sustain. Cities Soc.* **2020**, *55*, 102048. [[CrossRef](#)]
2. Gazafroudi, A.S.; Soares, J.; Ghazvini, M.A.F.; Pinto, T.; Vale, Z.; Corchado, J.M. Stochastic interval-based optimal offering model for residential energy management systems by household owners. *Int. J. Electr. Power Energy Syst.* **2019**, *105*, 201–219. [[CrossRef](#)]
3. Yu, L.; Xie, W.; Xie, D.; Zou, Y.; Zhang, D.; Sun, Z.; Zhang, L.; Zhang, Y.; Jiang, T. Deep Reinforcement Learning for Smart Home Energy Management. *IEEE Internet Things J.* **2020**, *7*, 2751–2762. [[CrossRef](#)]
4. Liere-Netheler, I.; Schuldt, F.; von Maydell, K.; Agert, C. Simulation of Incidental Distributed Generation Curtailment to Maximize the Integration of Renewable Energy Generation in Power Systems. *Energies* **2020**, *13*, 4173. [[CrossRef](#)]
5. Pinto, T.; Morais, H.; Sousa, T.M.; Sousa, T.; Vale, Z.; Praça, I.; Faia, R.; Pires, E.J.S. Adaptive Portfolio Optimization for Multiple Electricity Markets Participation. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 1720–1733. [[CrossRef](#)]
6. Gomes, L.; Vale, Z.A.; Corchado, J.M. Multi-Agent Microgrid Management System for Single-Board Computers: A Case Study on Peer-to-Peer Energy Trading. *IEEE Access* **2020**, *8*, 64169–64183. [[CrossRef](#)]
7. Yu, L.; Qin, S.; Zhang, M.; Shen, C.; Jiang, T.; Guan, X. Deep Reinforcement Learning for Smart Building Energy Management: A Survey. *arXiv* **2020**, arXiv:2008.05074.
8. Gomes, L.; Ramos, C.; Jozi, A.; Serra, B.; Paiva, L.; Vale, Z. IoH: A Platform for the Intelligence of Home with a Context Awareness and Ambient Intelligence Approach. *Futur. Internet* **2019**, *11*, 58. [[CrossRef](#)]

9. Djenouri, D.; Laidi, R.; Djenouri, Y.; Balasingham, I. Machine learning for smart building applications: Review and taxonomy. *ACM Comput. Surv.* **2019**, *52*, 1–36. [[CrossRef](#)]
10. Pérez-Lombard, L.; Ortiz, J.; Pout, C. A review on buildings energy consumption information. *Energy Build.* **2008**, *40*, 394–398. [[CrossRef](#)]
11. Esrafilian-Najafabadi, M.; Haghighat, F. Occupancy-based HVAC control systems in buildings: A state-of-the-art review. *Build. Environ.* **2021**, *197*, 107810. [[CrossRef](#)]
12. Gholamzadehmir, M.; Del Pero, C.; Buffa, S.; Fedrizzi, R.; Aste, N. Adaptive-predictive control strategy for HVAC systems in smart buildings—A review. *Sustain. Cities Soc.* **2020**, *63*, 102480. [[CrossRef](#)]
13. Adhikari, R.; Pipattanasomporn, M.; Rahman, S. An algorithm for optimal management of aggregated HVAC power demand using smart thermostats. *Appl. Energy* **2018**, *217*, 166–177. [[CrossRef](#)]
14. Khalid, R.; Javaid, N. A survey on hyperparameters optimization algorithms of forecasting models in smart grid. *Sustain. Cities Soc.* **2020**, *61*, 102275. [[CrossRef](#)]
15. Jung, W.; Jazizadeh, F. Human-in-the-loop HVAC operations: A quantitative review on occupancy, comfort, and energy-efficiency dimensions. *Appl. Energy* **2019**, *239*, 1471–1508. [[CrossRef](#)]
16. Gomes, L.; Spínola, J.; Vale, Z.; Corchado, J.M. Agent-based architecture for demand side management using real-time resources' priorities and a deterministic optimization algorithm. *J. Clean. Prod.* **2019**, *241*, 118154. [[CrossRef](#)]
17. Petrosanu, D.M.; Carutasu, G.; Carutasu, N.L.; Pirjan, A. A review of the recent developments in integrating machine learning models with sensor devices in the smart buildings sector with a view to attaining enhanced sensing, energy efficiency, and optimal building management. *Energies* **2019**, *12*, 4745. [[CrossRef](#)]
18. Ardakanian, O.; Bhattacharya, A.; Culler, D. Non-intrusive occupancy monitoring for energy conservation in commercial buildings. *Energy Build.* **2018**, *179*, 311–323. [[CrossRef](#)]
19. Razavi, R.; Gharipour, A.; Fleury, M.; Akpan, I.J. Occupancy detection of residential buildings using smart meter data: A large-scale study. *Energy Build.* **2019**, *183*, 195–208. [[CrossRef](#)]
20. Simma, K.C.J.; Mammoli, A.; Bogus, S.M. Real-Time Occupancy Estimation Using WiFi Network to Optimize HVAC Operation. *Procedia Comput. Sci.* **2019**, *155*, 495–502. [[CrossRef](#)]
21. Esrafilian-Najafabadi, M.; Haghighat, F. Occupancy-based HVAC control using deep learning algorithms for estimating online preconditioning time in residential buildings. *Energy Build.* **2021**, *252*, 111377. [[CrossRef](#)]
22. Deng, Z.; Chen, Q. Reinforcement learning of occupant behavior model for cross-building transfer learning to various HVAC control systems. *Energy Build.* **2021**, *238*, 110860. [[CrossRef](#)]
23. Wei, T.; Wang, Y.; Zhu, Q. Deep Reinforcement Learning for Building HVAC Control. In Proceedings of the 54th Annual Design Automation Conference, Austin, TX, USA, 18–22 June 2017. [[CrossRef](#)]
24. Escobar, L.M.; Aguilar, J.; Garcés-Jimenez, A.; De Mesa, J.A.G.; Gomez-Pulido, J.M. Advanced fuzzy-logic-based context-driven control for HVAC management systems in buildings. *IEEE Access* **2020**, *8*, 16111–16126. [[CrossRef](#)]
25. Lou, R.; Hallinan, K.P.; Huang, K.; Reissman, T. Smart Wifi Thermostat-Enabled Thermal Comfort Control in Residences. *Sustainability* **2020**, *12*, 1919. [[CrossRef](#)]
26. Wang, C.; Pattawi, K.; Lee, H. Energy saving impact of occupancy-driven thermostat for residential buildings. *Energy Build.* **2020**, *211*, 109791. [[CrossRef](#)]
27. Yang, Y.; Hu, G.; Spanos, C.J. HVAC Energy Cost Optimization for a Multizone Building via a Decentralized Approach. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 1950–1960. [[CrossRef](#)]
28. Khan, K.H.; Ryan, C.; Abebe, E. Day Ahead Scheduling to Optimize Industrial HVAC Energy Cost Based on Peak/OFF-Peak Tariff and Weather Forecasting. *IEEE Access* **2017**, *5*, 21684–21693. [[CrossRef](#)]
29. Khorram, M.; Faria, P.; Abrishambaf, O.; Vale, Z. Air conditioner consumption optimization in an office building considering user comfort. *Energy Rep.* **2020**, *6*, 120–126. [[CrossRef](#)]
30. Corbin, C.D.; Makhmalbaf, A.; Huang, S.; Mendon, V.V.; Zhao, M.; Somasundaram, S.; Liu, G.; Ngo, H.; Katipamula, S. *Transactive Control of Commercial Building HVAC Systems*; Pacific Northwest National Lab.(PNNL): Richland, WA, USA, 2016.
31. Mutis, I.; Ambekar, A.; Joshi, V. Real-time space occupancy sensing and human motion analysis using deep learning for indoor air quality control. *Autom. Constr.* **2020**, *116*, 103237. [[CrossRef](#)]
32. Yu, L.; Sun, Y.; Xu, Z.; Shen, C.; Yue, D.; Jiang, T.; Guan, X. Multi-Agent Deep Reinforcement Learning for HVAC Control in Commercial Buildings. *IEEE Trans. Smart Grid* **2021**, *12*, 407–419. [[CrossRef](#)]
33. Aguilar, J.; Garcés-Jiménez, A.; Gallego-Salvador, N.; de Mesa, J.A.G.; Gomez-Pulido, J.M.; Garcia-Tejedor, À.J. Autonomic management architecture for multi-HVAC systems in smart buildings. *IEEE Access* **2019**, *7*, 123402–123415. [[CrossRef](#)]
34. Garcés-Jimenez, A.; Gomez-Pulido, J.-M.; Gallego-Salvador, N.; Garcia-Tejedor, A.J. Genetic and Swarm Algorithms for Optimizing the Control of Building HVAC Systems Using Real Data: A Comparative Study. *Mathematics* **2021**, *9*, 2181. [[CrossRef](#)]
35. Collier, M.; Llorens, H.U. Deep Contextual Multi-armed Bandits. *arXiv* **2018**, arXiv:1807.09809.
36. Riquelme, C.; Tucker, G.; Snoek, J. Deep Bayesian bandits showdown: An empirical comparison of Bayesian deep networks for Thompson sampling. In Proceedings of the 6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings, Vancouver, BC, Canada, 30 April–3 May 2018.