Energy & Buildings 224 (2020) 110225

Contents lists available at ScienceDirect

Energy & Buildings

journal homepage: www.elsevier.com/locate/enb

Deep reinforcement learning to optimise indoor temperature control and heating energy consumption in buildings

Silvio Brandi [a], Marco Savino Piscitelli [a], Marco Martellacci [b], Alfonso Capozzoli [a,*]

[a] Department of Energy "Galileo Ferraris", Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy
[b] Enerbrain s.r.l, Strada Villa d'Agliè 26, 10132 Torino, Italy

ARTICLE INFO

Article history:
Received 19 March 2020
Revised 28 May 2020
Accepted 6 June 2020
Available online 9 June 2020

Keywords:
Deep reinforcement learning
Building adaptive control
Energy efficiency
Temperature control
HVAC

ABSTRACT

In this work, Deep Reinforcement Learning (DRL) is implemented to control the supply water temperature setpoint to terminal units of a heating system. The experiment was carried out for an office building in an integrated simulation environment. A sensitivity analysis is carried out on relevant hyperparameters to identify their optimal configuration. Moreover, two sets of input variables were considered for assessing their impact on the adaptability capabilities of the DRL controller. In this context a static and dynamic deployment of the DRL controller is performed. The trained control agent is tested for four different scenarios to determine its adaptability to the variation of forcing variables such as weather conditions, occupant presence patterns and different indoor temperature setpoint requirements. The performance of the agent is evaluated against a reference controller that implements a combination of rule-based and climatic-based logics. As a result, when the set of variables are adequately selected a heating energy saving ranging between 5 and 12% is obtained with an enhanced indoor temperature control with both static and dynamic deployment. Eventually the study proves that if the set of input variables are not carefully selected a dynamic deployment is strictly required for obtaining good performance.

1. Introduction

The last few years have seen a deep transformation in the energy system of many countries worldwide. The progressive introduction of renewable energy sources in buildings and the consequent effort for decarbonisation have changed the way to use and manage energy [1]. Important opportunities to address this task are provided by the implementation of strategies aimed at improving the building energy management and operation. In this context the increasing implementation of Internet of things (IoT) and Information and Communication Technologies (ICT) in buildings have supported an easier availability of a huge amount of building-related data [2,3] making it possible a bi-directional communication between infrastructures and operators [4,5].

Energy Management and Information Systems (EMIS) enable building owners to operate their buildings more efficiently and with improved occupant comfort. According to [6] EMIS can be categorized in three main families of data analytics-based tools including Energy Information System (EIS), Fault Detection and Diagnosis systems (FDD) and Automated System Optimisation (ASO) tools. ASO tools offer the opportunity to continuously analyse and modify control settings for optimising the building system energy usage. Advanced control strategies are mainly enabled by the progressive introduction of Advanced Metering Infrastructure (AMI) which allow the collection, storage, and analysis of a vast amount of building-related data. For this reason, the information gathered, if it is properly processed through data-driven procedures, may provide crucial knowledge on the actual and future building operational status including exogenous and endogenous variables influencing control performance [7,8].

In this context, more and more researchers worldwide are focusing on the development of advanced ASO for the optimal energy management of buildings leveraging on the great opportunities provided by the current advances in applied Artificial Intelligence (AI).

The optimal management of Heating Ventilation and Air Conditioning (HVAC) systems is one of the most promising application to investigate, considering that such systems along with lighting system account for more than a half of the energy demand in any type of building [9]. The main aim of controlling such systems is to guarantee the indoor comfort level while reducing the energy consumption during operation. Such control problem needs then to handle contrasting objectives and its formulation often represents a complex task to be accomplished. Moreover, the behaviour of building's occupants, which is extremely stochastic, and the

* Corresponding author.
E-mail address: alfonso.capozzoli@polito.it (A. Capozzoli).

https://doi.org/10.1016/j.enbuild.2020.110225
0378-7788/

interaction with the grid, furtherly contribute to increase the complexity of the control and optimisation process of building performance during operation.

In this context, buildings energy flexibility has been recognised as a key resource to be exploited in Demand Response (DR) scenarios [10]. According to Clauß et al. [11] the flexibility is the property of a building that defines the margin in which it can be operated according to its functional requirements. A further definition introduces the flexibility has the ability to manage a building according to grid requirements, climate conditions and user needs [10]. Actually, buildings can leverage their properties such as thermal inertia, electrical and thermal storages and renewable production to provide energy flexibility by adjusting HVAC systems operations. However, HVAC systems commonly implement classical control strategies such as on–off or Proportional-Integrative-Derivative (PID) control instead of more advanced solutions. This is mainly due to the current lack of guidelines and framework in literature for a robust implementation of advanced control for building industry [12]. Classical control are based on rule-based or reactive strategies which do not take into account prediction about external disturbances influencing energy consumption and thermal comfort in buildings. Moreover they do not perform any optimisation process and are not able to handle multiple and contrasting objective functions [13,14]. PID controllers provide great stability but fails when the operating conditions vary from the tuning conditions [15]; in this case manual tuning of PID is necessary but it is an extremely time consuming activity [10].

Advanced control techniques that are not widespread in the building industry include non-linear, robust and optimal control. Non-linear control is effective in catching non-linear dynamics of HVAC systems, but it requires rather complex mathematical modelling. Optimal and robust control are able to deal with time-varying disturbances but their applicability is limited due to the dynamic operation conditions of HVAC systems [15,16].

Among hard control methods [14], Model Predictive Control (MPC) aims at facing the main challenges of HVAC system control such as non-linear and time-varying dynamics and disturbances through an optimisation process performed over a receding time horizon [17,18]. The current scientific literature includes several works in which MPC was successfully applied to complex HVAC systems [19–21]. However, its application requires the definition of accurate models of the controlled environment [22,23] limiting MPC widespread adoption in the building industry. To overcome these limitations hybrid control strategies, such as adaptive control, have been successfully applied to HVAC systems [24–26]. Adaptive controllers do not require a priori identification of system parameters, as unknown parameters are estimated in real time through a parameter estimator which provides to the controller enough flexibility to adapt to time-varying disturbances and to account for uncertainty.

An alternative is provided by model-free control approaches such as Reinforcement Learning (RL) which can be employed, with no need of a-priori formalization of the controlled environment or process. In the RL paradigm, a control agent directly learns an optimal policy from its interactions with the environment through a delayed reward mechanism [27]. RL and in particular Deep Reinforcement Learning (DRL) was recently successfully applied to control problems previously unsolvable [28,29]. However, the exploration of this novel control approach is still in its infancy and effectiveness and limitations in energy and buildings applications need to be further explored. In the next section an overview on the existing literature related to the application of Deep Reinforcement Learning to address HVAC systems control is reported with the aim of introducing the current knowledge gaps and the consequent contribution of the present paper.

1.1. Related works to the application of Deep Reinforcement learning control in HVAC systems

Deep Reinforcement Learning (DRL) has recently gained popularity among RL algorithms due to its ability to adapt to very complex control problems characterized by a high dimensionality and contrasting objectives. DRL employs deep neural networks in the control agent due to their high capacity in describing complex and non-linear relationship of the controlled environment.

The first application of RL to HVAC systems dates back to 1998 [16], from this year up to 2012 the number of scientific publications about RL application to energy systems was limited to few works per year. From this period the interest of the scientific community about RL control framework has increased also due to recent advancements in deep learning. Recent studies exploited DRL for the regulation of supply water temperature setpoint [30], supply air flow-rate [31], supply air temperature [32], indoor temperature or humidity setpoint [33–35], fan speed or damper position [32,36] and tank temperature setpoint [37,38].

Zhang et al. [30,39] applied Asynchronous Advantage Actor-Critic (A3C) reinforcement learning control to a novel radiant heating system in an office building. The agent controlled the supply water temperature value achieving a reduction of 16.7% in energy demand while slightly increasing the Percentage of Person Dissatisfied (PPD). The authors highlighted the importance of introducing guidelines for practitioners for the design process of DRL applied to the built environment. Vàsquez-Canteli et al. [37,38] applied Deep Q-Learning to control a heat pump coupled with chilled water tank to minimize the energy consumption of the system. The control agent showed better performance compared to a Rule-Based Control (RBC) achieving 10% of energy saving.

Two exhaustive review work were recently published focusing on the application of RL control in buildings for demand response [40] and occupant comfort [41] respectively. In [40] were identified four major categories of energy systems where RL and DRL are applied: HVAC and Domestic Hot Water (DHW) systems, Appliances, Electric Vehicles (EV) and distributed generation coupled with storage systems. The authors identified a significant lack in real-world studies of RL controllers that may cause scepticism of building owners and managers about this technology. Moreover, the integration of RL with actual human feedback and the development of Multi-Agent Reinforcement Learning Controllers (MARL) were recognized as promising trends for future research in the energy and building sector.

The second review work [41] mainly focused on the comfort aspects identifying a lack of studies dealing with comfort factors different from indoor temperature such as Indoor Air Quality (IAQ) and visual comfort parameters. The integration of occupancy schedules and human feed-back into the control loop were identified as open research challenges to be addressed for developing effective occupant-centric building control.

In ideal conditions, a DRL agent should be directly implemented online in a real-world HVAC system learning, and its control policy should be refined by continuously interacting with the controlled environment through a trial and error process. However, in the initial stage of the learning process, the online implementation may lead to poor control performance since the agent could explore extreme states of the environment (e.g. poor thermal comfort conditions) in order to fully map the relation between the space of the state actions and the corresponding rewards obtained. In addition, DRL agent may take a considerable amount of time (between 20 up to 50 days) to converge to an acceptable control policy [35,37,42]. Therefore, to overcome this problem, the majority of researchers developed simulation environments combining various building energy simulation tools (EnergyPlus, CitySim) with deep learning libraries (Tensorflow, Pytorch) [30,32,33,37] to pre-train and test

DRL algorithms in off-line conditions. However, the development of accurate simulation models adds requires a considerable effort. In some cases, fully engineering models are not always capable to simulate the complexity of HVAC systems operation and the effect of occupant behaviour. An alternative is provided by black-box models built on historical data collected from Building Automation System (BAS) [36]. Despite such models proved to be able to accurately capture HVAC dynamics from collected monitored data, they could lack in generalizability, given that although they are able to easily reproduce patterns observed in the historical data set, suffer from extrapolation issues. Indeed, DRL represents a novel and promising approach research to HVAC control. However, it is still in its precocial state and further investigation are required in order to assess its performance compared to other solutions.

2. Knowledge gaps and contribution of the paper

Despite the advantages provided by the implementation of DRL as a control method for HVAC systems, some major drawbacks in the design and the training process of the DRL agent need to be further explored.

A DRL agent is characterized by a number of hyperparameters that need to be carefully tuned depending on the specific case study and objective functions [39]. As a consequence, despite its model-free nature, DRL requires a sort of modelling effort in its initial state to find the set of hyperparameters which may lead to the learning of a control policy close to the optimum in less time as possible and with an acceptable uncertainty [32]. In the existing literature an analysis on the effect of the hyperparameters settings on the performance of the control strategy was poorly investigated. Moreover, two opposite approaches can be followed when deploying a DRL agent previously trained offline: static deployment and dynamic deployment [30]. In the static deployment approach, the agent is implemented in the control loop as a static entity, meaning that the control policy is no longer updated, and any learning goes on. The advantages of such approach are the limited computational cost and the relative stability provided by a static control policy. The disadvantage is that the agent is unable to automatically adapt in the case key-features of the controlled system change (e.g. revamping intervention) and may need to be retrained. Conversely, in the dynamic deployment approach, the agent continuously learns from experience constantly updating its control policy. Following this approach a DRL agent can adapt to a changing system at the expense of higher computational cost and with the risk of stability issues for the control policy [30].

Moreover, in the design of the DRL a proper selection of the variable set which describe the environment is particularly important, considering it represents the environment as it is observed by the control agent. The effect of variable section on the adaptability capability of the DRL controller need to be further explored respect to the exiting literature.

The present paper focuses on the development of a DRL agent to control the setpoint of supply water temperature to heating terminal units system serving a thermal zone of an office building. The whole process was developed in an integrated simulation environment combining EnergyPlus [43] and Python. The developed simulation environment makes it possible to overcome some limitations of EnergyPlus in simulating advanced contol logics. The main scope of the work is to extensively test the operation of a robust agent by exploring its adaptability to the variation of forcing variables such as weather conditions, occupant presence patterns and different indoor temperature setpoint requirements. The analyses were conducted considering both a static and dynamic deployment with the aim of underlining limitations and opportunities. Moreover, two different sets of input variables (with an adaptive and non-adaptive approach respectively) were analysed for assessing the impact of variable selection process on the adaptability capabilities of the RL controller. To the best of authors knowledge such a comprehensive study has not been reported earlier in the literature.

On the basis of the literature review on DRL control in HVAC systems presented in Section 1.1 the main innovative contributions that this paper intends to provide can be summarised as follows:

- The control performance of a DRL agent was analysed both in terms of indoor temperature control and energy consumption against a baseline controller implementing a climatic-based logic of supply water temperature setpoint and a rule-based control of heating system operation.
- The design of a DRL agent was conducted performing a sensitivity analysis on the hyperparameters which may strongly affect the control performance of the agent.
- A proper variable selection was proposed to prevent the agent from learning an overfitted control policy. When a DRL agent is developed, in most of the cases the input variables describing the controlled environment are not defined to provide information to the agent in an adaptable manner with respect to control objectives. To this purpose, the variable selection process was performed both with adaptive and non-adaptive approach in order to produce an effective comparison.
- The two approaches of DRL deployment, static and dynamic, were compared in four different deployment scenarios to assess the adaptability of the agent to the variation of forcing variables such as weather conditions, occupancy patterns and different indoor temperature setpoint requirements.

The rest of the present paper is organised as follows. Section 3 provides an introduction to reinforcement learning theoretical formulation. Section 4 presents the methodological framework adopted for testing the DRL controller. Section 5 briefly describes the integrated simulation environment developed for this work. Section 6 introduces the case study and defines the control problem. Section 7 presents the results obtained for the analysed case study. The last two sections discuss the results and include concluding remarks and future directions of the research.

3. Reinforcement learning: concept and formulation

In the standard reinforcement learning formulation applied to HVAC control an *agent* (e.g. a control module linked to building management system running in the cloud) performs an *action* (e.g. turning on the heating system) when the *environment* (e.g. a building thermal zone) is in a *state* (e.g. the building is occupied and the indoor temperature is below the desired setpoint) and receives a *reward* which represents how much the agent is performing well by taking that action in that state with respect to control objectives. The goal of the agent is to learn an optimal control *policy* (π) that formally is a mapping between states and the probability of each action of being selected [27]. The *state-value function*, represents the expected return (i.e. the cumulative sum of future rewards) of the agent when starting from state s and following policy π:

$$v_\pi(s) = E\left[r_{t+1} + \gamma v_\pi(s') | S_t = s, S_{t+1} = s'\right] \quad (1)$$

where γ [0,1] is the discount factor for future rewards [27]. An agent employing a discount factor equal to 1 will give greater importance to rewards that can be obtained in the future. Whereas, an agent implementing a discount factor of 0 will assign higher values to states that lead to high immediate rewards. Similarly, the

action-value function represents the expected return of the agent when selecting action a starting from state s and following policy π:

$$q_\pi(s,a) = E\left[r_{t+1} + \gamma q_\pi(s',a') | S_t = s, A_t = a\right] \quad (2)$$

The values of v_π and q_π can be directly learned from experience. In this paper the most widely applied model-free reinforcement learning approach, namely Q-learning, was employed. Q-learning aims at estimating *state-action values* or *Q-values* from experience. These values are updated according to the following formula:

$$Q(s_t,a_t),\ Q(s_t,a_t) + \alpha\left[r_t + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t)\right] \quad (3)$$

where α [0,1] is the learning rate which determines with which extension new knowledge overrides old knowledge. When α is equal to 1 new knowledge completely substitutes old knowledge, instead, when, α is set equal 0 no learning happens and new knowledge is not employed to update the control policy. The higher the estimation of the Q-value for a specific state-action tuple (*s,a*) the higher is the expected reward of the agent for taking that specific action *a* in the state *s*.

One of the peculiarities that characterize reinforcement learning is the trade-off between exploration and exploitation. In order to maximize the rewards stream, an agent must select actions previously tried that have been found to be effective in obtaining high rewards (*exploitation*). However, to identify such actions it must select actions never tried before (*exploration*). Two of the most frequently used methods to select actions balancing exploration and exploitation are the ε-greedy and the soft-max methods. ε-greedy assigns equal probabilities to all non-optimal actions leading to poor results in some circumstances, while soft-max approach has shown problems on selecting the best-performing action [44]. The Max-Boltzmann exploration rules combines the two previously mentioned approaches. Following this approach, the agent acts almost deterministically when the estimations of the Q-values are not ambiguous (i.e. the Q-value associated with the best performing action significantly differs from the others), while it allows wider exploration in the region of the state-action space where the Q-values estimations are more ambiguous [44]. According to Max-Boltzmann rule the agent with probability ε selects actions with probabilities related to their Q-values:

$$\Pr(a|s) = \frac{e^{\frac{Q(s,a)}{\tau}}}{\sum e^{\frac{Q(s,a)}{\tau}}} \quad (4)$$

where τ is the Boltzmann temperature constant. Typically, the learning process is initialized with high values of ε (e.g. ε = 1 that means that the agent selects actions always based on soft-max distribution of the Q-values) and gradually reduce this value in order to exploit obtained knowledge.

3.1. Deep-Q-learning

In its classical formulation, Q-learning algorithm employs lookup tables to store and retrieve state-action values where each entry represents a state-action tuple (*s,a*). However, adopting a tabular representation may be unfeasible in practical problems where the state and action spaces are very large. A solution to this problem is to represent Q-values through a function approximator that allows state-action values to be represented by employing only a fixed amount of memory which depends only by the function used to approximate the problem. In particular, Deep Neural Networks (DNNs) have gained popularity due to their capacity to build an effective representation of the problem through their hidden layer structure. The first work implementing Q-learning and DNNs was developed by Minh et al. [28]. In Deep Q Networks (DQN) the Q-value function is parametrized by ϑ, where ϑ are the weights of the network. The number of neurons in the input layer of the network is equal to the number of variables from which a *state* is composed, while, the output layer has many neurons as the number of actions that the agent may take at each control interaction with the *environment*. Through this structure, the network is used to learn the relation between states and the Q-value for each action. However, in the RL paradigm, the true Q-value for each state-action pair is not known a-priori but it is learnt over successive interaction with the controlled environment. At each control step, the Q-values are updated according to Eq. (3) and used as targets to retrain the deep neural network.

Some improvements were introduced in literature in order to improve the DQN formulation. The first one is the introduction of the replay memory to store previous experience obtained by the agent. In the optimisation process of the network weights a random mini batch is extracted from the replay memory and used to fit DNN-regression using as targets Q-values updated according to Eq. (3). This enables the re-utilization of previous experience collected by the agent and overcome the problem of correlated observations while performing the optimisation process. The second improvement involves the employment of two neural networks [45]. The first one, called *online network*, is constantly updated and directly used in the interaction with the environment; the second one, called *target network*, is updated after N iterations and used to predict target values. The target network is an exact copy of the online network and during the update the weights of the online network are simply copied into the target network.

In the present work Double Deep Q-Learning with Memory Replay implementing the Max-Boltzmann exploration rule was applied to develop a DRL control to optimize both heating energy consumption and indoor thermal conditions. Fig. 1 depicts the control and learning loops in the Double Deep Q-Learning structure.

4. Framework of the analysis

In this section the methodological framework is presented with the aim of introducing each stage of the DRL control agent development. The present framework unfolds over three different stages as shown in Fig. 2.

Problem formulation: the first stage of the framework was aimed at defining the main components of the reinforcement learning control problem. The action-space includes all the possible control actions that can be taken by the control agent. Considering that a Deep-Q-learning was implemented, the action space is discrete. The reward is a function which describes the performance of the control agent with respect to the control objectives. Finally, the state-space is a set of variables related to the controlled environment which are fed to the agent in order to learn the optimal control policy which maximizes the reward function. The state-space was formalized following two approaches. In the first approach (Adaptive), the variables were selected in order to make them flexible to possible changes in the controlled environment (Variable Set A). In the second approach (Non-Adaptive), the selected variables are equally representative of the state of the environment but do not follow an adaptability paradigm (Variable Set B). A detailed description of the DRL problem formulation stage for the specific HVAC control case is provided in Section 6.4.

Training: in the second stage of the process the DRL agent was trained. As introduced in Section 3 reinforcement learning agents are characterised by many hyperparameters which require appropriate tuning. In this stage, a sensitivity analysis was carried out on some of the most important hyperparameters by training the agent with different configurations, in order to analyse the variations in the results obtained. The training process was implemented in an offline fashion using a training episode (i.e. a time period

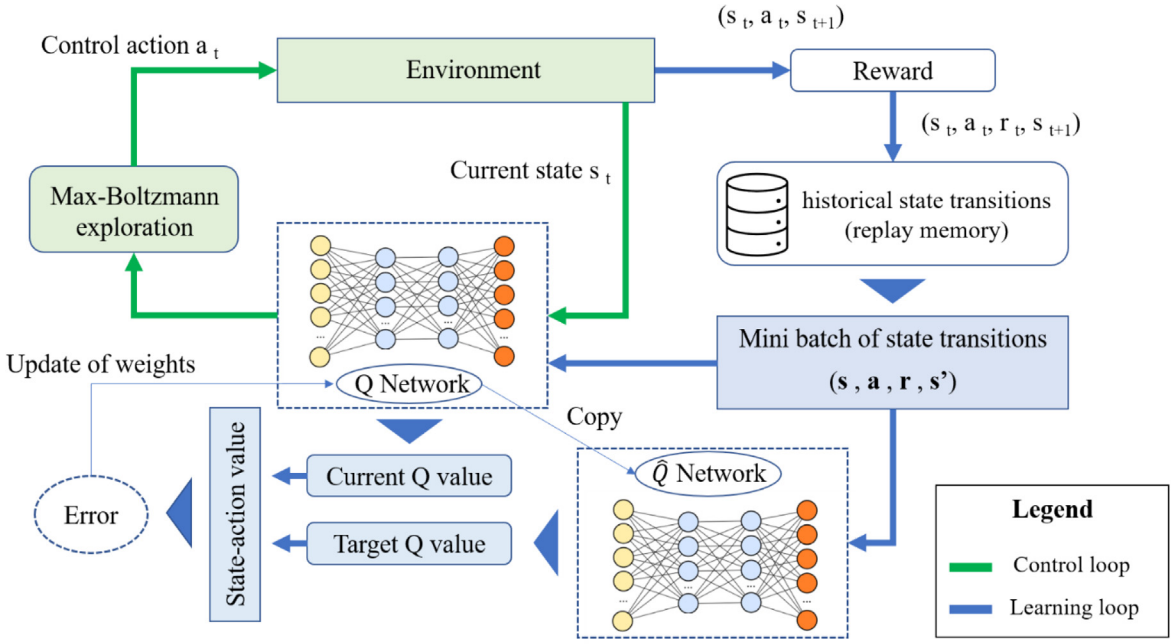

Fig. 1. Double Deep Q-Learning structure.

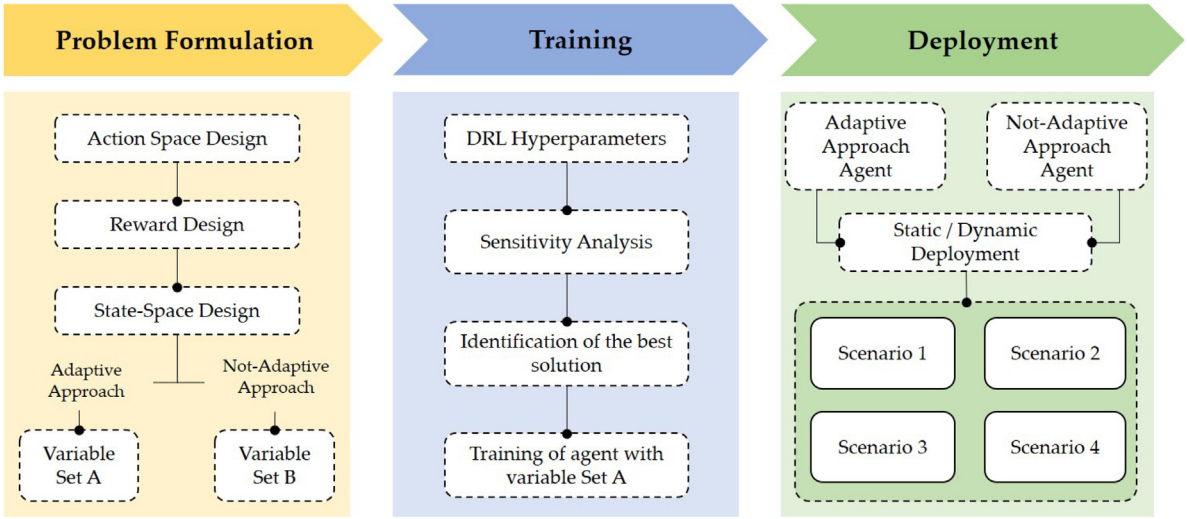

Fig. 2. Framework of the application of DRL control.

representative of the specific control problem) multiple times to constantly refine agent's control policy. The sensitivity analysis was performed for an agent implementing the variable set A. The best configuration of hyperparameters resulting from the analysis was successively employed to train the agent with variable set B. Details on the DRL training stage are provided in Section 6.5.

Deployment: the resulting agents, one trained on adaptive approach (using variable set A) and the other one trained with non-adaptive approach (using variable set B), were tested in the last stage. Both agents were tested through a static and dynamic deployment in one episode which includes a different period (i.e. weather conditions) from the training episode. Moreover, the deployment was performed in four different scenarios including different occupant presence patterns and indoor temperature requirements from the training stage. Eventually, a comparison of the performance obtained with the different approaches was proposed. Details on the deployment phase are provided in section 6.6.

5. Description of the simulation environment

As discussed in Section 3 the DRL agent aims at learning an optimal control policy by interacting with the controlled environment. In this work, the interaction between the control agent and the building was simulated within a surrogate environment which integrates EnergyPlus and Python. In particular, a EnergyPlus model of the building was wrapped in Python interface based on OpenAI Gym [46]. Through this approach a DRL agent, developed in python using existing libraries such as Tensorflow [47] and Keras [48], is able to virtually interact with a simulated building in order to learn the optimal control policy. The whole environment relies on Building Control Virtual Test Bed (BCVTB) and the *ExternalInterface* function of EnergyPlus.

The interaction between the two software is dynamic, and during a simulation a continuous exchange of data take place. The data flow is characterised by the following temporal features:

- *Control time step*: it represents the time step during which the action is taken by the agent. In this application the control time step was set equal to 15 min.
- *Simulation time step*: it is defined in the EnergyPlus environment and it is not directly linked to control time step. In this work the simulation time step was set equal to 5 min, as a result, a control action occurs every 3 simulation time steps.
- *Episode*: it is a simulation time period performed by EnergyPlus. One episode (or one simulation) is repeated multiple times during the training phase of the agent in order to allow the exploration of different trajectories. Conversely, an episode in the deployment phase is performed once in order to simulate the deployment of a trained control agent in a real building. Training and deployment episodes may differ, for example an agent can be trained on a heating season relative to one year and deployed in the heating season of the successive year. In this application a training episode lasts 2 months and a deployment episode lasts 3 months. Details about training and deployment episodes are provided in Section 6.

Fig. 3 shows the information flow that occurs during a simulation of DRL control interacting with the EnergyPlus simulation model. The dynamic simulation starts with the initialization (*init ()* function) of the OpenAI Gym environment which is formalized as a Python class. The *reset()* function is called at the beginning of each episode. This function re-initializes the EnergyPlus simulation process performing the simulation warm-up and returning the first state of the environment (e.g. the initial state of the building at the beginning of the simulation process). The state returned by EnergyPlus is defined as physical quantities and must be processed before they are provided to the DNN of the DRL agent. Details about the selection of the variables included in the state for the specific case study are presented in Section 6.4.3.

On the basis of the processed state the DRL agent selects one of the possible actions and passes it to the *step(a)* function which translate the encoded value into a physical control action. This latter value is passed to EnergyPlus as a schedule value through *ExternalInterface* function in order to simulate the next control step. From the second interaction with the environment the DRL agent receives also the reward which is used as a feedback signal to constantly improve its control policy as illustrated in section 3. This process continues until the end of an episode is reached. It is worth remembering that the length of an episode can be arbitrarily chosen, and it is defined within EnergyPlus model.

The green lines in the figure highlight the flow of data exchanged between Python and EnergyPlus that is handled through *BCVTB*.

6. Case study

The DRL algorithm described in the previous section was implemented to control the water supply temperature of a heating system for a simulated office building. In the following sub-sections, a description of the case study together with the formulation of the control problem are provided.

6.1. Building description

The simulated building is representative of a huge portion of the Italian building stock in terms of both heating system configuration and building construction features. It is a six-level mixed-use building with a net heated surface of 9300 m^2 located in Turin, Italy. The indoor environment is heated through water terminal units (i.e., radiators). The building is composed of three thermal zones served by different hot-water circuits and was built between 1930 and 1960. The average transmittance values of the opaque and transparent envelope components are respectively 1.084 and 2.921 W/m^2K. The ratio between heat transfer surface and gross volume (i.e, aspect ratio) is equal to 0.25 m^{-1}. The implementation of the DRL controller is tested for one thermal zone which includes only office rooms. This zone is composed of four-levels with a net heated surface of 7000 m^2 and a net heated volume of 33000 m^3. The remaining zones are occupied by the local police department and the warden of the whole building. Fig. 4 shows a picture of the real building and highlights the thermal zone modelled in this work.

6.2. Heating system and control objectives

The heating system installed in the real building is quite complex. It is composed by two hot water loops connected by a heat exchanger. The primary loop includes four gas-fired boilers with a total nominal capacity of 1300 kW. The secondary loop includes three zone-loops served by different pumping systems. The three zone-loops withdraw hot water from the same water collector. The control of the supply water temperature is achieved through three-way mixing valves. However, EnergyPlus does not reach this level of complexity in the definition of the HVAC system and some simplifications were introduced to model the building.

In the present case study, the control problem focuses on the regulation the supply water temperature (T_{SUPP}) to heating terminal units of a single thermal zone. The heating system was modelled in EnergyPlus with a single hot water loop. The supply side includes a single gas fired boiler (*Boiler:HotWater*) and a constant speed pump (*Pump:ConstantSpeed*). The supply water temperature setpoint (SP_{TSUPP}) was managed through a *SetPointManager:Scheduled* which directly receives inputs from Python through the *ExternalInterface*. The demand side includes one thermal zone and its relative bypass branch. The goal of the control policy is to reduce the amount of thermal energy provided to the supply water while maintaining indoor air temperature within an acceptability range during occupied periods. This application, even being developed in a simulation environment in which every thermal comfort-based parameter can be easily evaluated, considers only the zone air temperature (T_{ZONE}). In fact, other comfort related-variables are not monitored in the real building. Moreover, the water terminal units can control only the sensible part of the thermal load. If the zone air temperature value falls between upper and lower threshold of a pre-defined acceptability range, then indoor temperature requirements are satisfied. In this application the acceptability range was defined in the interval [−1,1] °C from the desired indoor temperature setpoint ($SP_{T,ZONE}$). The work focuses on the energy supplied for heating the carrier fluid (Q_{SUPP}) regardless the type of the generation system serving the building. Technically, in real life implementations, the regulation of supply water temperature can be achieved through different solutions such as three-way mixing valves or by modulating boiler or heat pumps. The control policy developed through the presented approach could be then employed independently by the actual generation system installed. Fig. 5 provides a simplified scheme of the heating system and of the control problem formulation.

6.3. Baseline control logic

The performance of the DRL control was evaluated against a baseline control logic implementing a combination of rule-based and climatic-based logics for the control of the supply water

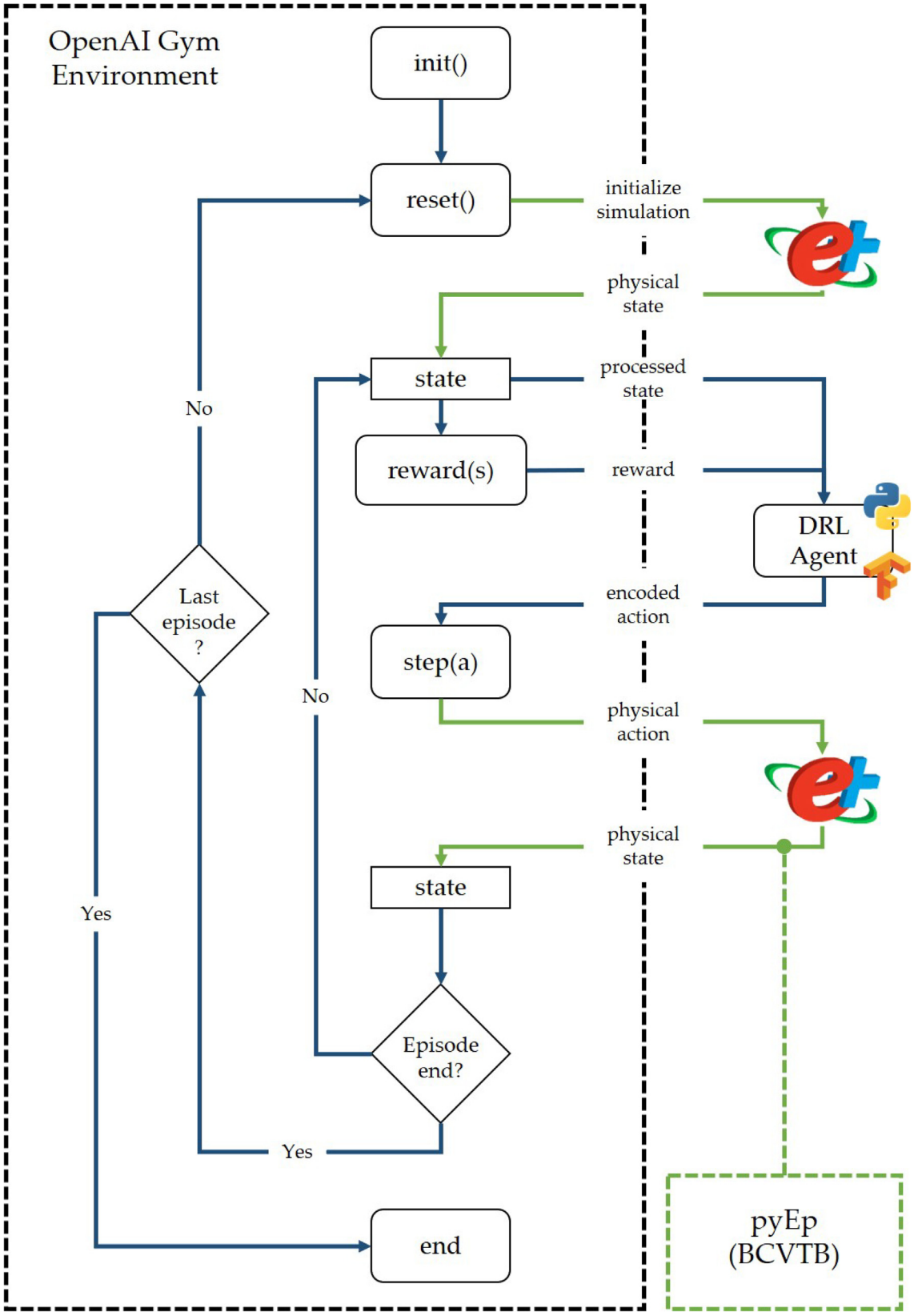

Fig. 3. Architecture of simulation environment for RL control in HVAC systems.

temperature. The starting time of the heating system was determined according to the value of indoor temperature and the amount of time before the occupant's arrival. The controller is enabled to turn on the heating system up to four hours before the arrival of the occupants if the difference between the actual indoor temperature and the low threshold of the acceptability

Fig. 4. Office case study located in Torino, Italy. Detail of the office zone modelled in this work.

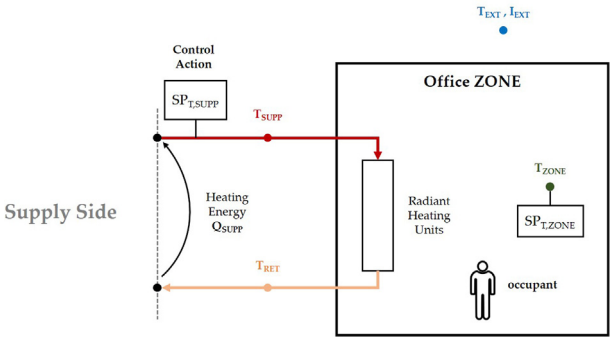

Fig. 5. Schematic of the heating system analysed.

range is higher than 3 °C, or up to three hours before if that difference is higher than 2 °C. In any other case the controller turns on the heating two hours before occupant's arrival if the zone temperature is lower that the low threshold of the acceptability range. When the zone reaches the upper threshold of the acceptability range the heating system is turned off. If the zone temperature falls below the lower threshold the heating system is turned on again. This control strategy is operated until two hours before occupants leave the building, when the heating system is turned off to exploit thermal inertia until the next day. The supply temperature value is linearly interpolated between a maximum value of 70 °C when the outdoor air temperature falls below −5 °C and a minimum value of 40 °C when the outdoor air temperature is over 12 °C. These values were selected according to the control logic of the supply temperature actually implemented in the Energy Management System of the real building.

6.4. Design of DRL control problem

The Deep Reinforcement Learning control algorithm described in Section 3 was trained and tested in a developed simulation environment. In the next sub-sections, the design of the action space and of the reward function are discussed along with the configuration of the training and deployment phases.

6.4.1. Design of the action-space

At each control time step the agent selects a value of supply temperature setpoint ($SP_{T,SUPP}$). Considering that the DQN was chosen as control agent the action-space is expressed in a discrete space. The space includes the following actions related to the supply water temperature in °C:

$$A_t = [20, 40, 50, 60, 70]$$

These values where selected in order to provide to the DRL agent the same range of supply water temperature setpoint as the baseline controller. At the same time, the values were selected to limit the actions to only five values in order to not overcomplicate the control problem formulation. Given the inertia of the water-based heating system intermediate values of supply water temperature can be reached by the agent switching between available control actions during system operation. The introduction of intermediate values of setpoint supply water temperature in the present action-space (e.g. 45 °C, 55 °C, 65 °C) would have only increased the complexity of the calculations performed by the neural network model [49] without effectively producing an improvement on the learned control policy. The simulation environment was set in order to shut down circulation pump when the supply water temperature value falls below 20 °C.

6.4.2. Design of the reward function

The reward that the agent receives after taking an action at each control time step depends by two competing terms: the energy and temperature-related terms. The energy-related term is proportional to the energy provided to supply water to reach the desired setpoint. Unlike other applications where the energy-related term is purely intensive [36,39], in this study this term was normalized with respect to the temperature difference between zone temperature setpoint and outdoor air temperature. This formulation was introduced for not penalizing the agent in taking energy-intensive actions when the outdoor temperature is very low and vice-versa.

The temperature-related term is quadratically proportional to the distance between zone air temperature setpoint and its actual value. This formulation was found to be effective in speeding up the learning process, making the agent able to easily avoid the exploration of states characterised by unacceptable conditions of the indoor environment from the very beginning of the training phase. The formulation of the reward function is expressed by the following equation:

$$R = -\beta * \frac{Q_{supp}}{SP_{T,ZONE} - T_{EXT}} - \rho * \left| (SP_{T,ZONE} - T_{zone})^2 \right|_{OCC=1} \quad (5)$$

The coefficients ρ and β were introduced to weight the importance of the two terms of the reward function.

6.4.3. Design of the state-space

The state represents the environment as it is observed by the control agent. The agent, at each control time step, choses among the available actions according to the values assumed by the state. In this work, two different state-space were designed as introduced in section 4. The first one includes a set of input variables (*variable set A*) selected in order to guarantee the maximum adaptability of the learned control policy. The second state-space, instead, is composed by a set of input variables (*variable set B*) which do not follow an adaptive approach. In both cases the variables were selected according to the following criteria:

- The variables must provide to the agent all the necessary information to predict immediate future rewards.
- The variables must be feasible to be collected in a real-world implementation.

The two variable sets are reported in Tables 1 and 2 respectively. Overall, the adaptive set (*variable set A*) includes 11 variables while the not-adaptive set includes 13 variables (*variable set B*).

External Air Temperature and *Direct Solar Radiation* were both included in variable set B, as they are the most influencing ambient variables affecting building heating energy consumption and indoor temperature. On the contrary, in the feature set A, *External Air Temperature* was substituted by the variable *ΔT Indoor Setpoint*

– *External Air* since it is directly related to the formulated reward function (Eq. (5)). This formulation was found to be effective in removing the dependency of the learnt control policy from a fixed value of indoor temperature setpoint which could limit agent adaptability.

The *supplied Heating Energy* was selected considering that it is proportional to the energy-related term in the reward function and it represents a key information that has to be provided to the agent. Moreover, the heat supplied to the water depends by the *Supply Water Temperature* and by the *Return Water Temperature*. These variables, which represent the main operational parameters of heating system, were included in both the variable sets.

Information about the presence of occupants in the zone, from which depends the temperature-related term in the reward function, is provided through three different variables. The *Occupants' Presence Status*, added in the set built following not-adaptive approach, indicates if, in a certain control time step, the zone is occupied or not (it depends only by the occupancy schedule) and it is expressed in the range [0,1]. However, this information alone is not comprehensive. It would be desirable for the agent to learn when it is convenient to pre-heat the zone so as to ensure an adequate indoor air temperature during occupancy period. A common approach to this problem in the literature, implemented in the non-adaptive set, is to select as variables *time-of-the-day* and *day-of-the-week*. However, following this procedure, the agent may learn to fit only to a specific occupancy-schedule provided during the training process. To overcome this issue, the variables *time to occupancy start* and *time to occupancy end* were introduced in the *variable set A* to define the time left for the subsequent change in the occupancy pattern. When the building is not occupied, *time to occupancy start* represent the number of hours left before occupants' arrival time, during occupancy periods this variable is equal to 0. Conversely, when the building is occupied, *time to occupancy end* represent the number of hours to occupants' leaving time, during off-occupancy periods this variable is equal to 0.

Eventually, the agent needs information about the zone air temperature which is directly connected with the temperature-related term of the reward function. This information was straightforwardly added to the *variable set B* along with its 3 lagged values in the past (15, 30 and 45 min lag respectively) and the *Indoor* Setpoint. Contrarily, in *variable set A*, this information was provided indirectly introducing as variable the difference between the *Zone Air Temperature* and *Indoor* Setpoint along with its 3 lagged values in the past (15, 30 and 45 min lag respectively).

The *Relative Humidity* was not included in the two set of variables considering that the heating system based on water radiators is capable to control only the sensible part of the heating load.

In order to feed the variables to the neural network, they were scaled in the (0, 1) range according to a min–max normalization.

Table 1
Variables included in the *variable set A* conceived with an adaptive approach.

Variable	Min Value	Max Value	Unit
ΔT Indoor Setpoint – External Air	6	31	°C
Direct Solar Radiation	0	720	W/m^2
Supplied Heating Energy	0	125	kWh
Supply Water Temperature	10	80	°C
Return Water Temperature	10	80	°C
Time to Occupancy Start	0	36	h
Time to Occupancy End	0	12	h
ΔT Indoor Setpoint – Indoor Air	−3	10	°C
ΔT Indoor Setpoint – Indoor Air, 15 min lag	−3	10	°C
ΔT Indoor Setpoint – Indoor Air, 30 min lag	−3	10	°C
ΔT Indoor Setpoint – Indoor Air, 45 min lag	−3	10	°C

Table 2
Variables included in the variable set B conceived with a non-adaptive approach.

Variable	Min Value	Max Value	Unit
Time of the day	0	24	h
Day of the week	1	7	–
External Air Temperature	−12	26	°C
Direct Solar Radiation	0	720	W/m^2
Supplied Heating Energy	0	125	kWh
Supply Water Temperature	10	80	°C
Return Water Temperature	10	80	°C
Occupants' Presence Status	0	1	–
Indoor Set Point	13	25	°C
Indoor Air Temperature	13	25	°C
Indoor Air Temperature, 15 min lag	13	25	°C
Indoor Air Temperature, 30 min lag	13	25	°C
Indoor Air Temperature, 45 min lag	13	25	°C

6.5. Setting of training phase

The Reinforcement Learning framework is characterised by a number of hyperparameters that strongly affect the behaviour of the control agent. In order to analyse their impact on the performance of the control agent, different configurations of the most interesting hyperparameters were tested and compared in this study (Table 4). The configurations implemented for the training of the DRL agent are described in the following tables.

This sensitivity analysis was performed only with the agent implementing the state space built following adaptive approach (*variable set A*). In Table 3 are listed the values of the hyperparameters kept unchanged during the training.

Table 4 reports the details of each hyperparameter configuration implemented for the sensitivity analysis. The two hyperparameters involved in the sensitivity analysis are the discount factor and the weight factor of the temperature-related term (ρ). As explained in section 3, the discount factor determines the importance of future rewards over immediate rewards and directly affects the magnitude of Q-values. The weight factor of the temperature-related term of the reward function (ρ) defines the relative importance of indoor temperature requirements with respect to energy consumption. Lower values may result in a control policy which guarantees higher energy saving at the expense of higher temperature violations and vice-versa.

The performance of Deep Reinforcement Learning is affected by the stochastic behaviour that is intrinsic in both deep neural networks and controlled environments. In order to account for this aspect, each configuration has been ran three times employing multiple random seeds in order to ensure consistency according to [51]. Successively, the hyperparameters of the run leading to the best performance in terms of both energy savings and temperature control were selected to train also the agent implementing *variable set B*.

Table 3
Fixed Hyperparameters of the DRL Agent training.

	Variable	Value
1	DNN architecture	4 Layers
2	Neurons per hidden layer	512
3	DNN Optimizer	RMSprop [50]
4	Optimizer Learning Rate	0.0001
5	DQN batch size	32 Control Steps
6	Episode Length	5856 Control Steps (61 days)
7	Sequential Memory Size	5 Episodes
8	Target Model Update	672 Control Steps (7 days)
9	Training Episodes	50
10	τ Boltzmann Temperature	1
11	ε Start	1
12	ε End	0.1
13	Energy-related term weight factor (β)	1

Table 4
Different hyperparameter configurations implemented in the training phase.

run	Discount Factor γ	Weight Factor ρ
1,2,3	0.9	10
4,5,6	0.95	10
7,8,9	0.99	10
10,11,12	0.9	20
13,14,15	0.95	20
16,17,18	0.99	20
19,20,21	0.9	1
22,23,24	0.95	1
25,26,27	0.99	1

As stated in section 4, a training episode includes 2 months, from 1st of November to 31st of December (5856 control steps, one every 15 min). The weather file used in this work is the reference weather file (*ITA_TORINO-CASELLE_IGDG.epw*) available in EnergyPlus for Torino, Italy. The same weather file from the 1st of January to 31st of March was used for the deployment phase. As reported in Table 3 each training episode was repeated 50 times for each hyperparameter configuration in order to let the agent explore several control strategies. On average one episode took 3 min to be simulated on a machine with an Intel® Core™ i7-8550 CPU @ 1.80 GHz processor and 16,0 GB RAM. An entire training period (including 50 episodes) for each hyperparameter configuration took on average 150 min to be simulated.

Fig. 6 shows the patterns of outdoor air temperature and direct solar radiation in the two periods (i.e. training and deployment period). For the sake of legibility, the solar radiation values include only daylight period. The training period was selected for its wide range of temperature values spanning between −8 °C and 17 °C while the direct solar radiation is higher during the deployment period. However, this latter aspect allows to test the adaptability of DRL agent different climatic patterns from those used for the training. In the training phase occupancy was simulated between 07:00 and 19:00 from Monday to Saturday. The required indoor setpoint was set equal to 21 °C and the temperature acceptability range between 20 °C and 22 °C.

6.6. Deployment phase

In the last phase of the process the two agents were deployed in four different scenarios in order to assess the adaptability capabilities of the learned control policy to different configurations related to the controlled environment. Each agent was deployed for one episode including the period between 1st January and 31st March. The four different scenarios are:

- Scenario S1: this is the base case where no changes in the controlled environment were implemented. The goal is to test the adaptability of the RL controller only to patterns of outdoor conditions (i.e. air temperature and solar radiation) never observed during the training phase.
- Scenario S2 & S3: in these scenarios the zone temperature setpoint was increased to 22 °C and decreased to 20 °C respectively in order to assess the performance of the agent in satisfying temperature requirements that differ from the ones assumed in the training.
- Scenario S4: in this case the zone occupancy schedule was modified as shown in Fig. 7 maintaining unchanged the zone temperature setpoint respect to the training conditions. The lighting and electric appliances schedules were also changed according to the new occupancy schedule.

The trained control agents were deployed in each testing scenario in both static and dynamic configuration. In the static configuration the control policy was not updated during the deployment of the agent. Contrarily, dynamically deployed agents constantly leverage new experience obtained interacting with the environment to adjust their control policy. The second solution, despite providing greater adaptability, requires additional computational cost and may cause instabilities in the learned control policy [39].

7. Results

The framework presented in Section 3 was implemented in the integrated simulation environment. The results are presented in this section in order to compare the performance of different DRL control agents (trained with different input variable sets and deployed following different approaches) and the baseline control of supply water temperature to terminal units of a heating system.

7.1. Results of the training process

As introduced in Section 3, in the first step of the training process a sensitivity analysis was carried out on two DRL hyperparameters to highlight their influence on the performance of the control

Fig. 6. Outdoor Air Temperature patterns during training and deployment periods.

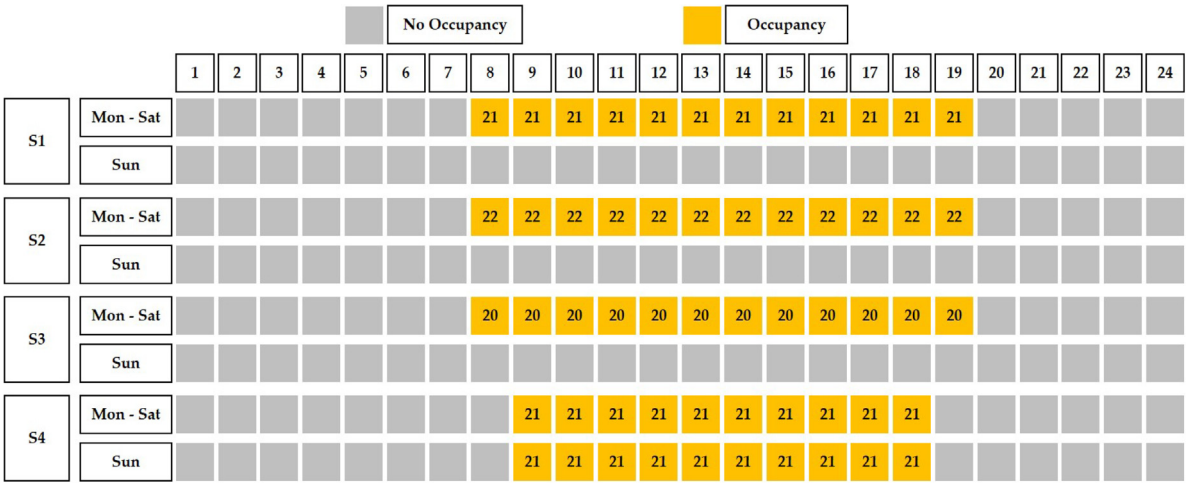

Fig. 7. Occupancy schedules and indoor setpoint in different design conditions.

algorithm. The variable set based on adaptive approach introduced in Section 6.4.3 was implemented for this sensitivity analysis.

A useful indicator to assess the goodness of the learning process of a DRL agent is represented by the evolution of the cumulative reward per episode. The reward, which has not a direct physical meaning, takes into consideration both the energy consumption and indoor temperature values and combines them in a single value. Higher values of the reward correspond to a better performance obtained by the control agent. It is important to supervise if the reward converges to a stable value. A non-convergent trend in the reward may be caused by an agent that failed in achieving an optimal control policy. To this purpose, the convergence of the different configurations of the agent were analysed in the episode-reward plot showed in Fig. 8. The figure is split into two main panels representing the evolution of the energy-related term and temperature-related term respectively. Each main panel is furtherly organized in a grid in which each sub-panel represents a specific configuration of the hyperparameters. Each sub-panel shows the evolution of the relative term of the reward function during the training episode. The solid line shows the average value per episode of the three different runs performed for each configuration, while the grey area was drawn between maximum and minimum value per episode. In all the configurations the agent starts exploring high values of the energy-related term and extremely low values of the temperature-related term. Across the different runs, the agent firstly learns how to correctly maintain indoor temperature during the first 20 episodes; this fact can be observed by analysing the increase of the temperature-related term values and the relative decrease of the energy-related term. From this stage (i.e. 20th episode) the agent begins to learn how to reduce energy consumption while keeping indoor temperature in the range it previously learned. In fact, the values of the temperature-related term are quite stable while the values of the energy-related term increase. Agents that were initialized with a discount factor γ equal to 0.99 represent an exception, showing highest variance in terms of temperature control performance. The training runs performed with this specific configuration (γ = 0.99) seek to obtain higher rewards in a longer time horizon compared to other agents generating an instability in the objective function. This aspect is particularly clear observing the evolution of the temperature-related term of the agent implementing a discount factor of 0.99 and a weight of the temperature-related term equal to 20. On the other hand, agents applying a discount factor equal to 0.9 shows the higher stability among all the training configurations due to the shorter time horizon considered.

In this application the reward function is the weighted sum of supplied heating energy to water and temperature control performance (see equation..). Therefore, the reward value alone cannot directly provide a straightforward metric to evaluate the overall performance of DRL control.

While the energy performance can be straightforwardly evaluated comparing the amount of heating energy supplied to the water, the temperature control performance requires the definition of an appropriate metric. In the present work, the indoor temperature control performance was evaluated by calculating the cumulative sum of temperature violations during occupancy hours. A temperature violation occurs when the building is occupied, and the indoor temperature falls outside the acceptability range. The magnitude of the temperature violation is then calculated as the absolute difference between actual indoor temperature and desired set point value at each simulation step. The cumulative value of this quantity over an entire episode returns the performance of the control algorithm expressed in °C.

Fig. 9 shows, in a four-quadrant visualization, the cumulative sum of temperature violations during occupancy periods, as a function of the heating energy saving with respect to climatic-based control baseline for the different hyperparameter configurations reported in Table 4. The figure reports the results obtained in the last episode (50th) of the training process. For the sake of legibility of the plot the y-axis was defined on a logarithmic scale. The black-dashed lines indicate the performance achieved by the baseline controller. The left-bottom quadrant includes all the solutions that have performed better than the baseline both in terms of indoor temperature control and energy consumption. Worst solutions, corresponding to higher energy consumption and temperature violations than the baseline, should be displaced in the right-top quadrant. None of the training runs produced results that fall within this latter region. In particular, solutions with a discount factor (γ) of 0.99 and a weight of temperature-related term (ρ) of 10 (runs 7, 8 and 9) and 20 (runs 16, 17 and 18) show the highest variability. Agents trained with discount factors (γ) of 0.9 and 0.95 and a weight (ρ) of 10 or 20 lead to the best trade-off solution achieving, at the same time, energy saving and temperature control improvement. In particular, the setting of the discount factor equal to 0.9 (run 1, 2 and 3) produced the less scattered solutions. This aspect can be interpreted as an indicator of the consistency of the control policy learned by such agents. As can be expected, agents implementing a weight factor of the temperature-related term equal to 1 achieved greater energy savings at the cost of worse temperature control. Following these considerations, the

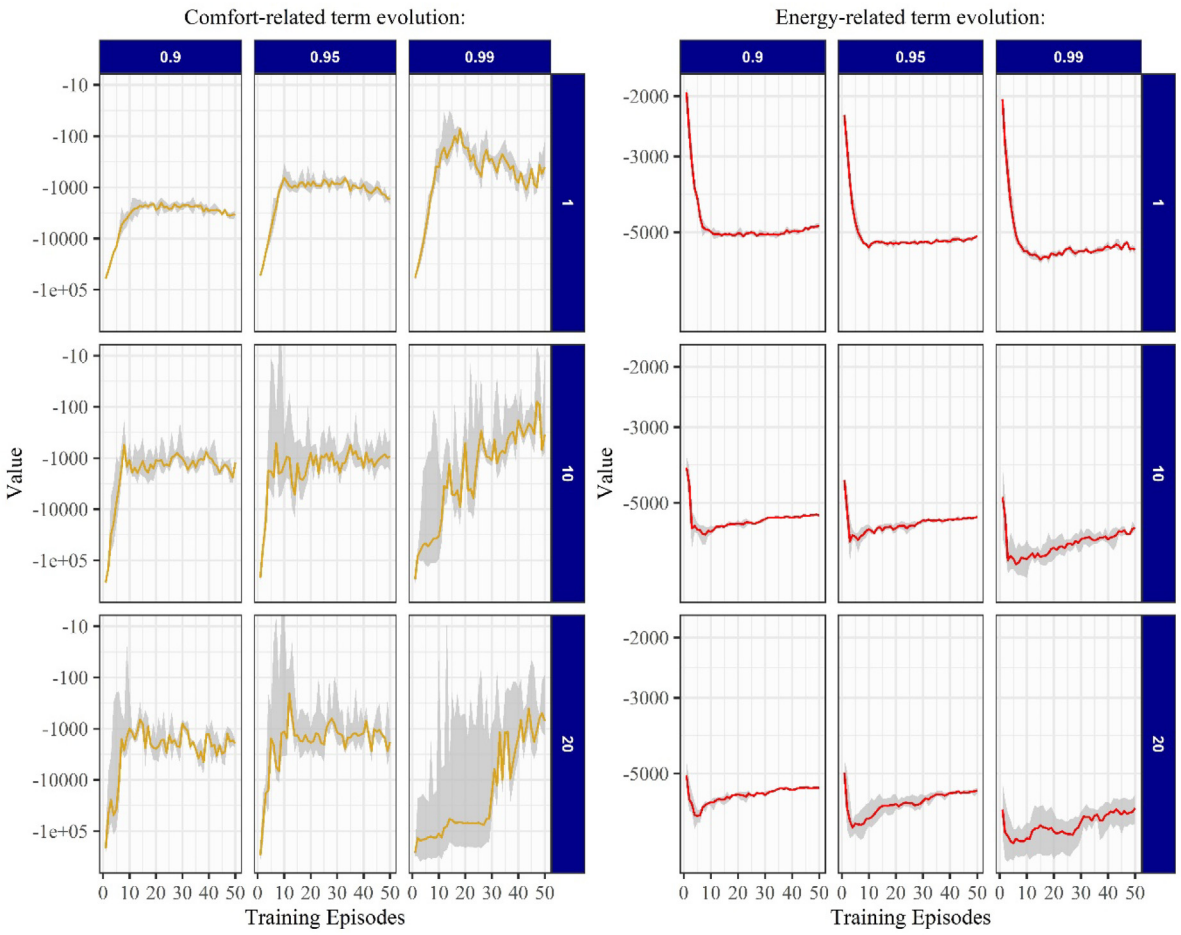

Fig. 8. Evolution of energy-related and temperature-related term of the reward function during training phase.

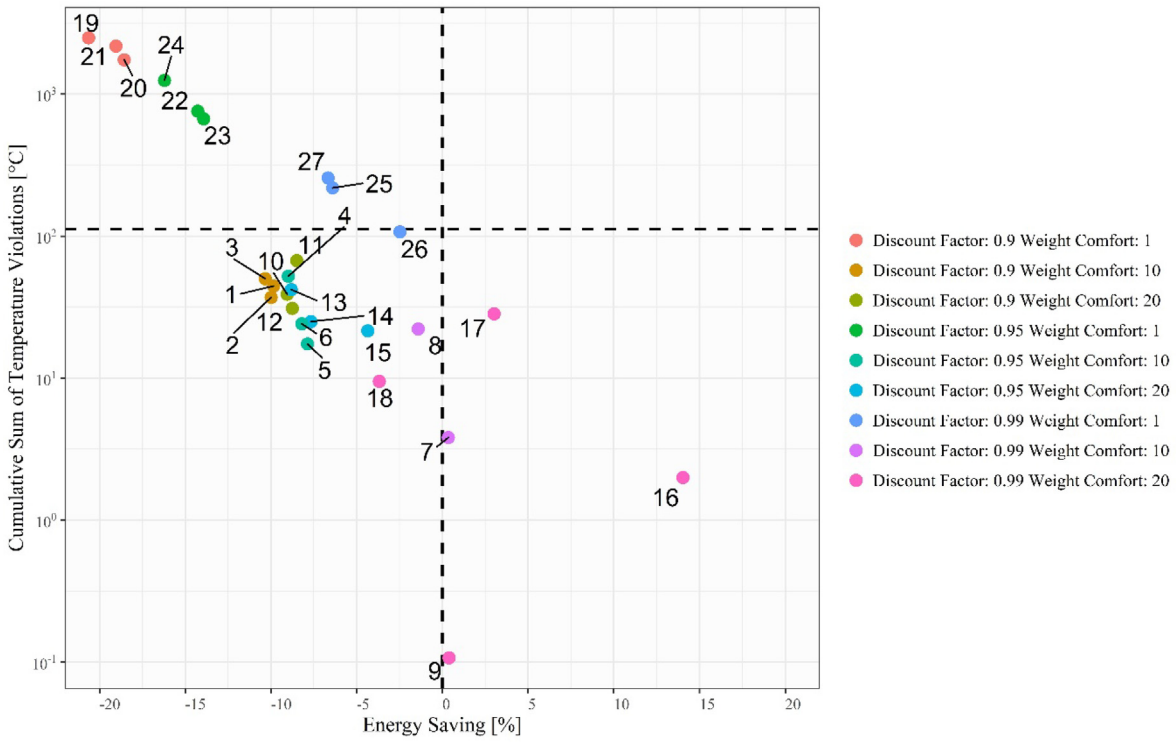

Fig. 9. DRL control performance in the last episode of the training phase. Each point refers to a different training runs as reported in Table 4.

agent number 2, with a discount factor of 0.9 and a weight factor ρ of 10, was selected as best solution among configurations explored in the sensitivity analysis process.

In order to furtherly characterise the results of the training phase, the performance of the different solutions was analysed on daily scale.

In Fig. 10 are compared three agents implementing different values of the discount factor γ. The comparison is proposed for the same working day of the training episode. The figure shows the behaviour of the agent when the discount factor changes while the weight factor is kept constant (ρ = 10) for the same day of the training period. Overall, in the three training runs, the agent has learnt to maintain the indoor temperature between lower and upper thresholds of the temperature acceptability range as can be observed from the central panels of the figure. However, in the solution obtained considering a discount factor equal to 0.9, the agent learnt to better maintain the indoor temperature across lower threshold of the acceptability range. As can be observed from the left figure, the run performed with a discount factor of 0.99 considerably anticipated the start-up phase resulting in higher energy consumption compared to other solutions. Given the higher discount factor, this agent learnt how to optimise the rewards stream in a longer horizon causing higher instability. The agent implementing a discount factor of 0.9 selected higher values of the supply water temperature during the first hours of the morning. As a result, the zone air temperature reached exactly the lower threshold of the acceptability range (20 °C) at the beginning of the occupied period (07:00). This agent led to a heating energy saving of about 100 kWh in comparison with the agent implementing a discount factor of 0.95 that shows a similar pattern of indoor air temperature.

Fig. 11 reports the performance of the trained agents considering different values of the weight factor ρ and a constant discount factor (γ = 0.9). It is possible to notice the relative importance given to temperature violations obtained in the three different solutions.

In detail, the agent trained with a weight factor equal to 1 sacrificed indoor temperature control at the beginning and ending of the occupancy period. However, this agent obtained a further daily energy saving of about 100 kWh, respect to the previously discussed solution (ρ = 10, γ = 0.9), at the cost of keeping indoor air temperature 1 °C below the lower threshold of the acceptability range at 07:00 and 19:00.

At the end of the training phase, the same hyperparameter configurations of the best solution resulting from sensitivity analysis (i.e., discount factor γ = 0.9 and weight factor ρ = 10) were employed to train a second agent with the variables of the state-space selected following the non-adaptive approach (*variable set B*). Table 5 report the performances of the two agents relative to the last (50th) training episode which lasts for 2 months between the 1st of November and 31st December.

As can be observed the two agents show similar performance in terms of energy saving obtained compared to baseline. The temperature violations during occupancy were expressed both in terms of cumulative value of violations (°C) and occurrence rate (%). As a reference, a temperature violation with an occurrence rate of 5% means that the indoor temperature is out of range for the 5% of the total simulation steps included in the occupied periods of the building. Despite both agents improved the indoor temperature control and reduced heating energy consumption respect to the baseline, the agent trained with *variable set A* performed slightly better especially in terms of indoor temperature control. This aspect suggest that this agent was capable to better exploit internal and external heat gains, improving temperature control and, at the same time, increasing energy saving.

7.2. Results of the deployment phase

In this last section are analysed the results of the deployment of the two agents (trained with *variable set A and B* and considering ρ = 10 and γ = 0.9) in the four different scenarios introduced in

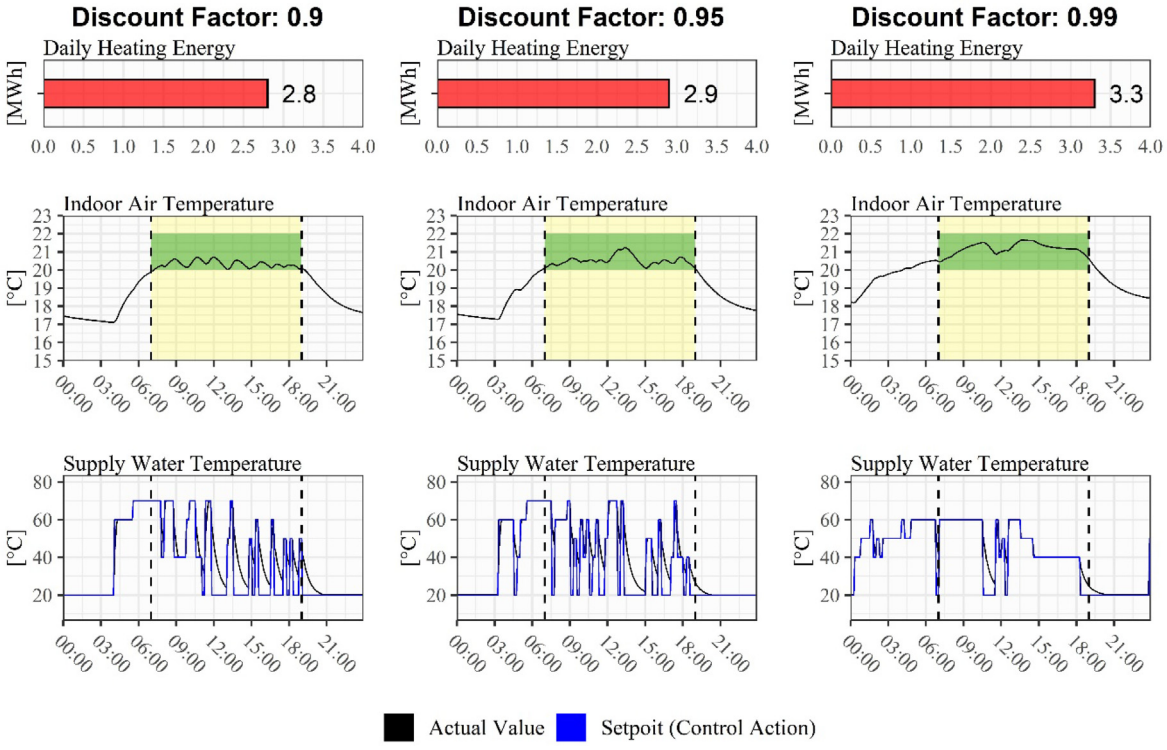

Fig. 10. Comparison between agents implementing different discount factors during a training day.

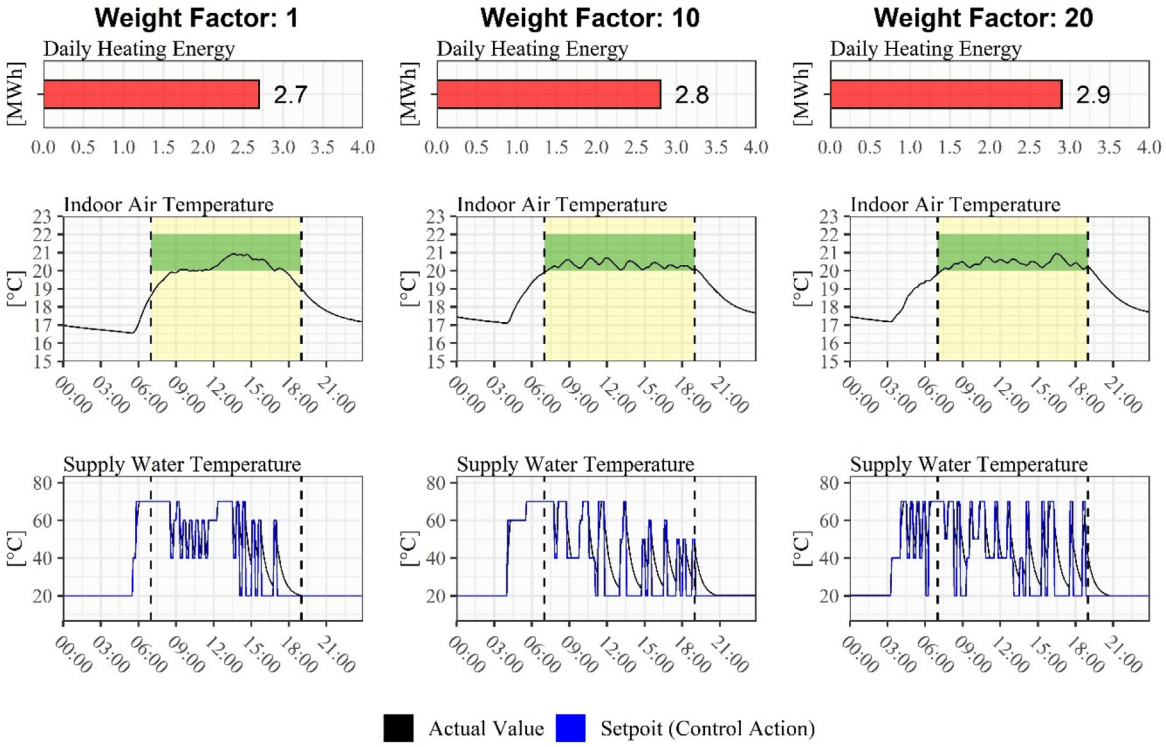

Fig. 11. Comparison between agents implementing different weight factors of the temperature-related term during a training day.

Table 5
Performance comparison at the end of the training phase between agents implementing adaptive and non-adaptive variable set in the definition of the state-space (γ = 0.9, ρ = 10).

Variable Set	DRL Control			Climatic-Based Control			Energy Saving [%]
	Consumption [MWh]	Temperature Violations		Consumption [MWh]	Temperature Violations		
		Cumulative [°C]	Occurrence-rate [%]		Cumulative [°C]	Occurrence-rate [%]	
A	101	37	2.8	113	112	3.3	**−10.0**
B	102	96	5.7				**−9.92**

section 6.6. The deployment of each agent was simulated both in a static and dynamic way for one episode. As previously introduced, the deployment episode is 3 months long, including January, February and March, and the climatic data employed in the simulation are gathered from the reference weather file referred to Torino (*ITA_TORINO-CASELLE_IGDG.epw*). Fig. 12 summarises the performance obtained in terms of supplied heating energy and cumulative sum of temperature violations for all the possible configurations resulting from the combination of the four scenarios, two variable sets, and two deployment processes (16 configurations) including also the baseline configuration. The performance of the agent trained with the *variable set A* did not produce always with dynamic deployment configuration an improvement with respect to static deployment across the four scenarios (azure and blue bars in the Fig. 12). In particular, in scenarios S2 and S3 the dynamically deployed agent achieved a lower energy saving compared to its statically deployed counterpart. In scenario S2 this led to a slight improvement of temperature control performance while in scenario S3 the temperature control was performed with less accuracy compared to statically deployed agent. Even without updating its control policy the agent trained with the *variable set A* is capable to adapt to the different requirements in the different scenarios achieving better performance than the baseline controller. The agent based on *variable set B*, instead, shows opposite behaviour and the effect of dynamic deployment over static deployment is particularly significant (yellow and orange bars in the figure). For example, in the scenario S2, which considers an increased temperature setpoint compared to training condition, the statically deployed agent obtained the lowest consumption (yellow bar in the first panel of the bottom figure) but an extremely high value of the cumulative sum of temperature violations (yellow bar in the second panel of the bottom figure) meaning that the control policy was not able to adapt to the new indoor temperature requirements. On the contrary, the dynamically deployed agent in the same scenario achieved an overall performance comparable with agent implementing the *variable set A* conceived with an adaptive approach.

A similar condition occurred also for the fourth scenario, which considers the presence of the occupants during Sunday (contrarily the training period) where the dynamic deployment drastically improved the indoor temperature control performances of the agent trained with *variable set B*. The same agent (trained with *variable set B*) shows a different pattern in the third scenario. In this case, in which the desired indoor setpoint was reduced from 21 °C to 20 °C, the statically deployed solution was capable to achieve satisfying temperature control performance (yellow bar in the third panel of the bottom figure), but it obtained lower energy saving. On the contrary, the dynamically deployed solution achieved almost the same temperature control performance (orange bar in the third panel of the bottom figure) but increased the energy

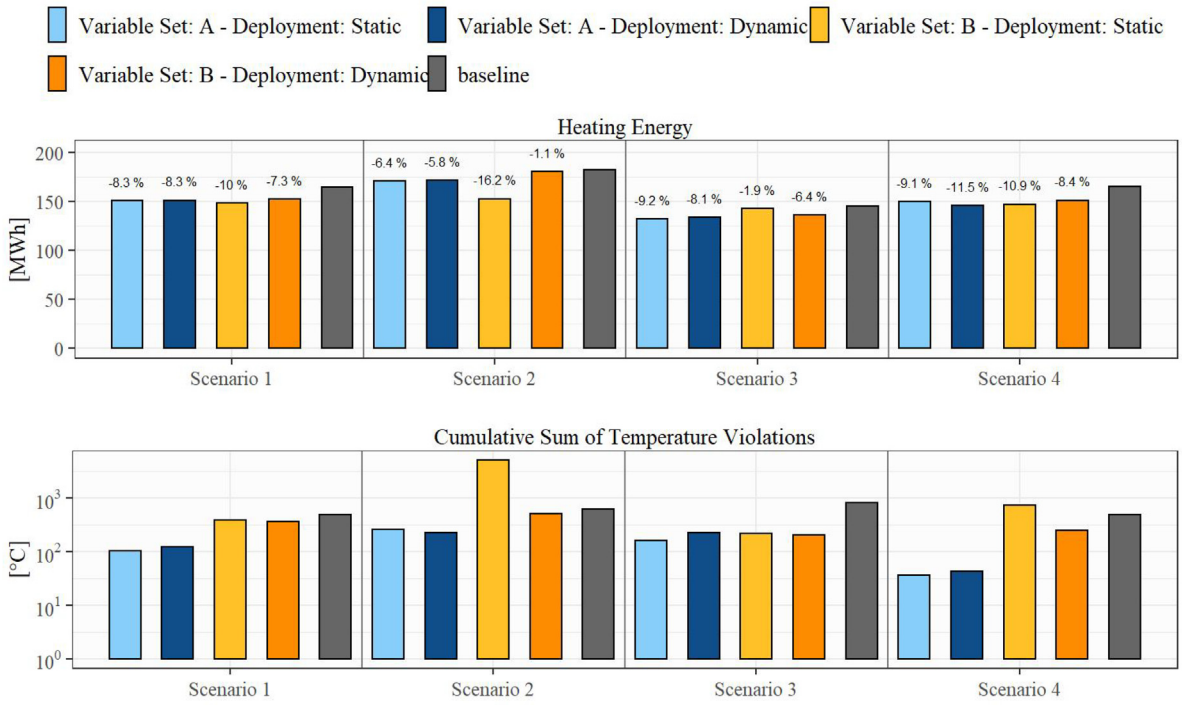

Fig. 12. Heating energy supplied and cumulative sum of temperature violations for agents trained with both variable sets in four different scenarios under static and dynamic deployment configuration. In the upper part of the figure are reported on the bars the heating energy saving respect to the baseline.

savings obtained from 1.9% to 6.4%. Also in this case the dynamic deployment was found to be effective in improving performance of the agent by means of continuous refinement of the control policy during the deployment episode. However, as the Fig. 12 clearly shows, even in the dynamic deployment configuration the agent trained with *variable set B* was not able to achieve the performance of the agent trained with *variable set A* across all the four scenarios.

The successive figures (from Figs. 13–15) provide details about some configurations that are of particular interest for supporting the discussion.

Fig. 13 shows a comparison between statically deployed agent trained with *variable set A*, and the baseline controller during a week of the deployment period. The plot shows the indoor air temperature patterns generated by the two controllers along with supply water temperature, outdoor air temperature and direct solar radiation profiles. The DRL agent was able to exploit solar heat gains reducing supply water temperature and, consequently, save energy. This aspect is particularly relevant during the third and sixth day when solar radiation is higher.

Fig. 14 highlights the differences between agent trained with *variable set A* (red lines) and agent trained *variable set B* (blue lines). The plot shows for different weeks and the same working day (Tuesday), the daily indoor temperature profiles in the scenario S2, which implements an increased indoor setpoint (22 °C) compared to the training phase (21 °C). As can be observed the agent based on adaptive variables (*variable set A*) was promptly able to adapt to the change of indoor temperature requirements maintaining satisfying conditions within the zone despite any learning goes on during static deployment. On the other hand, the agent trained with non-adaptive variables (*variable set B*) was not capable to adapt without relying on dynamic deployment.

Fig. 15 compares the effect of a static and a dynamic deployment for the agent trained with variables selected according to the non-adaptive approach (*variable set B*). This detail is particularly interesting considering that, as can be observed in Fig. 12, the differences between the two deployment strategies are more emphasized for the agent trained with the *variable set B*. The figure shows the results obtained during the first 6 Sundays in deployment scenario S4. This scenario is particularly interesting because, differently from the training conditions, implements the presence of occupants during Sundays. The plot shows, for the first 6 weeks, the daily indoor temperature profiles generated by the two agents. It is interesting to notice that the divergence between the profiles increases over time suggesting that the two agents have different adaptability capabilities. During the first week the two agents generated almost the same pattern which clearly do not satisfy the indoor temperature requirements. The larger temperature violation is localized during the first hours of the day since both the agents were not able to anticipate occupants' arrival. A second temperature violation region is localized in the middle part of the day, when, during training, the agent correctly learnt to exploit solar heat gains in order to reduce supply water temperature. However, the reduction of supply water temperature caused the occurrence of temperature violation condition since the agent did not performed a sufficient pre-heating of the zone in order to reach the acceptability range of the indoor temperature. This pattern was replicated by the statically deployed agent among the six weeks demonstrating its lack in adapting to the modified occupancy schedule. On the contrary, the dynamically deployed agent was capable to learn from experience and it was able to achieve satisfying temperature conditions starting from the third week of deployment.

8. Discussion

The present paper focuses on the development of a DRL controller of supply water temperature setpoint to terminal units of a heating system. The developed controller was trained and deployed in a simulation environment which combines EnergyPlus and Python. The controller aims at optimising both energy consumption and indoor temperature control trying to identify the best trade-off between the two contrasting functions. The control

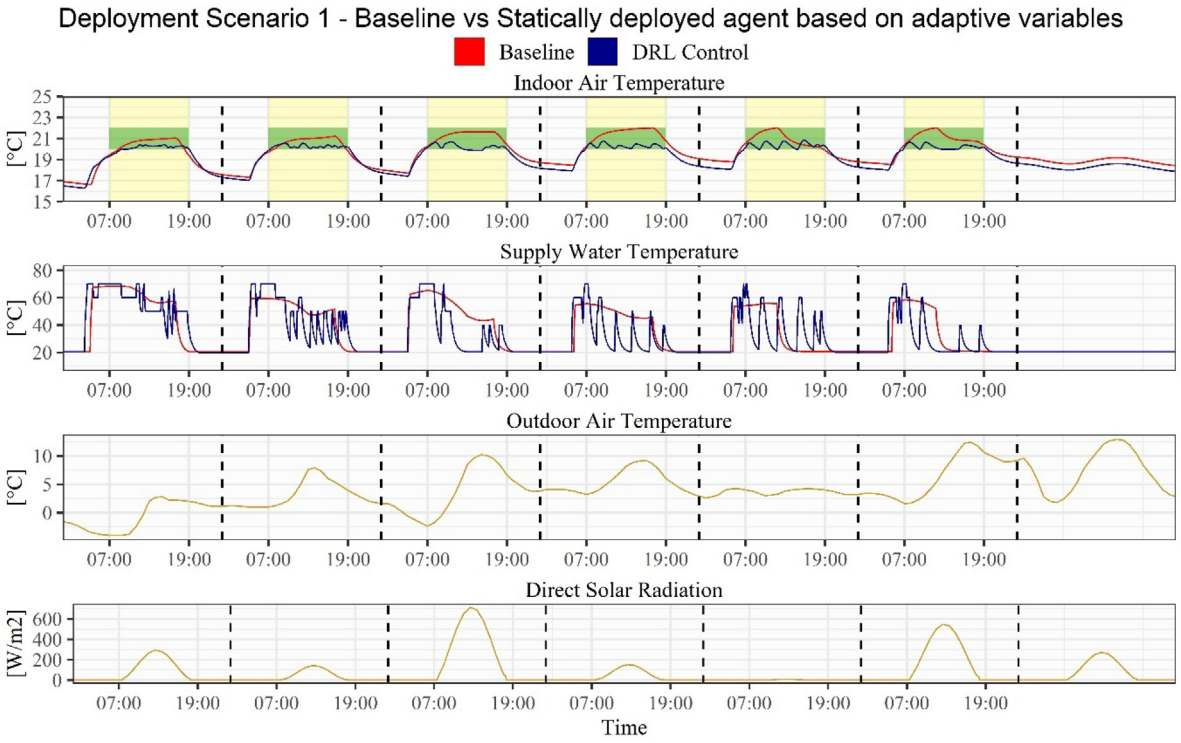

Fig. 13. Comparison between statically deployed agent trained with *variable set A* and baseline controller during a week of the deployment period.

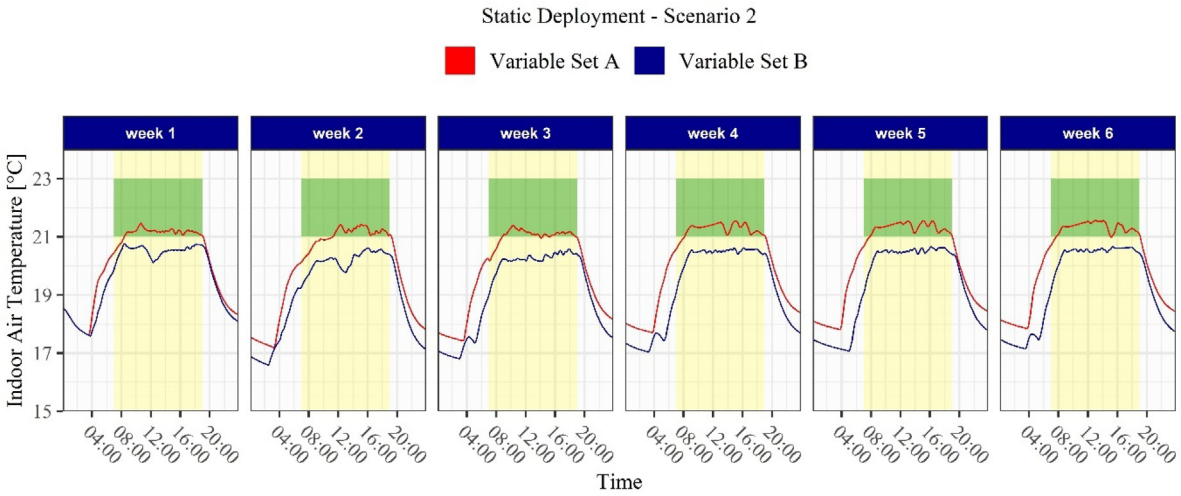

Fig. 14. Comparison between statically deployed agents trained with *variable set A* and *variable set B* in terms of daily indoor temperature profiles during Tuesdays in the scenario S2.

problem analysed in this work was relatively simple, not involving elements such as renewable energy sources or storages which may effectively require an optimised controller to be fully exploited. Although the only two features of the building that could be exploited in the considered optimisation process were the building thermal mass and the temperature acceptability range, the DRL controller led to good performance improvements in comparison to the baseline controller.

In DRL algorithms hyperparameters tuning and reward design play a key role in identifying the optimal configuration of DRL controller. In this work, a sensitivity analysis was carried out on some of the main hyperparameters to highlight their influence on the final performance of the developed controller. Given this strong dependence it seems necessary for reinforcement learning applications in HVAC systems to rely on simulated environments, at least in the initial stage of training. As a consequence, despite the model-free nature of reinforcement learning control, a modelling effort needs to be accounted.

The effect of adaptive variables defining the state-space was analysed. A variable set designed to enhance adaptability and flexibility of a DRL agent with respect to variable requirements of the indoor environment (i.e. indoor temperature setpoint and occupancy schedule) was introduced. A DRL agent based on adaptive variables was compared with an agent trained with more classic non-adaptive variables. The comparison was performed by simulating the deployment of the two agents in four different scenarios.

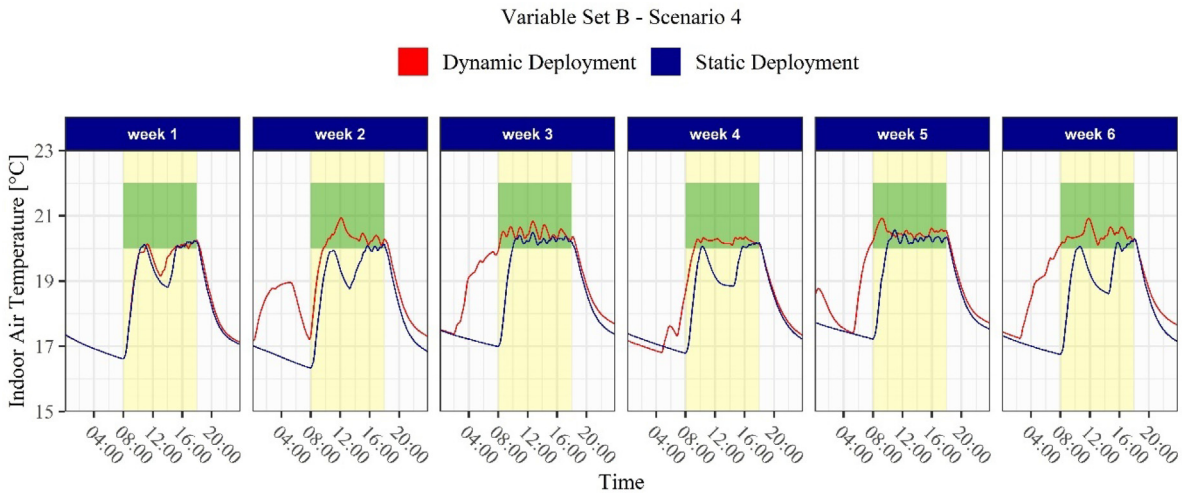

Fig. 15. Comparison between dynamically and statically deployed agent trained with *variable set B* in terms of daily indoor temperature profiles during Sundays in scenario S4.

Moreover, the agents' deployment was simulated both in static and dynamic configuration. The agent trained with adaptive variable set was capable to adapt to each scenario performing better than the baseline controller even if statically deployed. The dynamic deployment of the same agent did not produce significant improvements on the overall performance, showing slight poorer performance compared to static deployment case.

On the contrary when the variables were selected with a non-adaptive approach the dynamic deployment performed better that the static deployment in all the scenarios analysed. These results proved that the proposed variable selection process was useful in providing to the agent the capability to adapt itself to changes that may occur in the controlled environment. This analysis suggests that a DRL controller with a carefully designed state-space is capable to provide the necessary flexibility and adaptability to changing indoor requirements even in a static deployment configuration. Through this approach is possible to leverage the advantages provided by static deployment (i.e. lower computational costs and higher stability) without sacrificing adaptability. However, the adoption of an adaptive approach in the design of the state space may not be enough to guarantee a good control performance in the case of retrofit on the HVAC system or other building components. In such cases thermal dynamics of the controlled environment may change requiring DRL controller to update its policy through a dynamic deployment.

The implementation of the proposed controller in a real-world testbed requires the monitoring of a few variables that can be easily collected through low-cost solution already available in the market. An outdoor ambient sensor is required to monitor outdoor air temperature and solar radiation. Alternatively, those data can be easily obtained by an external weather data provider. Many of those services requires no fees for a limited number of data requests and already implement Application Program Interfaces (APIs) which enable the streaming of data. Low-cost solutions are available also for what concerns indoor Air temperature monitoring. Supply and return water temperature are usually collected by the Building Management System (BMS) and thermocouples must be installed in the relative pipes. The most challenging quantity to be monitored is the supplied heating energy. This variable can be indirectly calculated from supply and return water temperature if the water mass flow rate through the system is known and collected through an appropriate sensor or directly by installing a non-invasive heat meter. Since the considered case study is an office building the variables *time to occupancy start* and *time to occupancy end* included in the variable set based on adaptive approach can be easily obtained through working timetables. The most challenging aspect is to design an infrastructure capable to manage the stream of data from different sources in order to provide to the controller the required input information. The static or dynamic deployment can be achieved in situ if the BEMS allows the running python scripts otherwise all the operations can be performed in a cloud server.

9. Conclusions and future works

In the present paper, the application of DRL control in a water-based heating system was developed and analysed in a simulation environment. The flexibility and adaptability of the control agent to different occupancy schedules and indoor temperature requirements was tested in different scenarios showing the potentialities of the proposed solution. A proper selection of variables defining the state-space was proposed with the aim of developing a controller capable to adapt to dynamic changes of the environment. The importance of hyperparameters selection was highlighted by analysing the sensibility of the results for different configurations of their values. The DRL control agent with variables selected according to adaptive approach led to savings between 5% and 12% of heating energy depending by the analysed scenario. This agent was able to achieve these performances in a static deployment configuration suggesting that a careful design of the state space may be sufficient in providing to an agent the capability to adapt to changes in the controlled environment without scarifying its stability with a dynamic deployment configuration. At the same time, the controller achieved satisfying performance in controlling indoor air temperature.

Future works will be focused on the following aspects:

- Exploring the capabilities of Multi-Agent Reinforcement Learning (MARL) framework. The present work focuses on only one zone of an office building characterized by a complex HVAC system. MARL could provide a solution to coordinate multiple actuators that are present in an HVAC system in order to reach a global optimum solution.
- Comparing the performance of DRL with model-based control solution such as MPC. Due to its model-free formulation, Reinforcement Learning is diametrically opposed to Model Predic-

tive Control. A robust comparison between these two techniques in terms of control performance, computational cost and modelling effort could provide useful insights on the strength and weakness of DRL controllers.
- Applying DRL to novel HVAC systems. Given the ability of DRL to handle multi-objective function, HVAC systems characterized by higher level of complexity (e.g. including RES generation and storage) could provide an excellent testbed to prove the effectiveness of DRL control over classical control methods.
- Introducing comfort parameters in the objective function. Even if the monitoring of many comfort parameters (e.g. air velocity, mean radiant temperature) is a non-trivial task in real world applications, in a simulative context the evaluation of thermal comfort performance achieved by a DRL agent could be explored in future works.
- Including a more detailed description of occupancy within the DRL control problem. The agent can benefit from the information related to the number of occupants in a thermal zone or a building in order to better optimise the control policy. Although it is not trivial to measure this variable, in a real-world facility some non-intrusive monitoring techniques such as laser sensors or wi-fi signals could be employed.
- Implementing the developed controller in a real-world testbed. Moving from simulation to real-world implementation is extremely complicated and present some major challenges related to the required infrastructure to effectively deploy the controller. Future works will be focused on investigating these aspects and on the evaluation of the performance of DRL control agent once deployed in-field.
- Furtherly explore the paradigm of dynamic deployment of DRL agents. Despite the disadvantage of possible instabilities in the learned control policy, dynamic deployment might be necessary to obtain a fully-flexible agent which is capable to adapt even when the thermal dynamics of the controlled environment changes (e.g. retrofit intervention). In the future works dynamic deployment will be analysed in order to enhance its robustness and stability.

A major effort to build upon this research work will be then focused on fully addressing all the mentioned challenges that are behind the next generation of "intelligent" buildings.

CRediT authorship contribution statement

Silvio Brandi: Conceptualization, Methodology, Investigation, Visualization, Software, Writing - original draft. **Marco Savino Piscitelli:** Conceptualization, Methodology, Formal analysis. **Marco Martellacci:** Investigation, Visualization. **Alfonso Capozzoli:** Conceptualization, Supervision, Formal analysis, Writing - review & editing, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The work of Silvio Brandi was made in the context of a Ph.D. scholarship at Politecnico di Torino funded by Enerbrain s.r.l.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.enbuild.2020.110225.

References

[1] Capozzoli A, Mechri HE, Corrado V. Impacts of architectural design choices on building energy performance applications of uncertainty and sensitivity techniques 2009;15217:1000–7.
[2] Z. Yu, B.C.M. Fung, F. Haghighat, Extracting knowledge from building-related data - a data mining framework, Build Simul 6 (2013) 207–222, https://doi.org/10.1007/s12273-013-0117-8.
[3] C. Fan, F. Xiao, C. Yan, A framework for knowledge discovery in massive building automation data and its application in building diagnostics, Autom Constr 50 (2015) 81–90, https://doi.org/10.1016/j.autcon.2014.12.006.
[4] A. Capozzoli, M.S. Piscitelli, S. Brandi, Mining typical load profiles in buildings to support energy management in the smart city context, Energy Procedia 134 (2017) 865–874, https://doi.org/10.1016/j.egypro.2017.09.545.
[5] M.S. Piscitelli, S. Brandi, A. Capozzoli, Recognition and classification of typical load profiles in buildings with non-intrusive learning approach, Appl Energy (2019) 255, https://doi.org/10.1016/j.apenergy.2019.113727.
[6] Kramer H, Lin G, Granderson J, Curtin C, Crowe E. Synthesis of Year One Outcomes in the Smart Energy Analytics Campaign Building Technology and Urban Systems Division 2017.
[7] M. Molina-Solana, M. Ros, M.D. Ruiz, J. Gómez-Romero, M.J. Martin-Bautista, Data science for building energy management: a review, Renew Sustain Energy Rev 70 (2017) 598–609, https://doi.org/10.1016/j.rser.2016.11.132.
[8] Capozzoli A, Piscitelli MS, Brandi S, Grassi D, Chicco G. Automated load pattern learning and anomaly detection for enhancing energy management in smart buildings. Energy 2018;157. DOI:10.1016/j.energy.2018.05.127.
[9] G. Martinopoulos, K.T. Papakostas, A.M. Papadopoulos, A comparative review of heating systems in EU countries, based on efficiency and fuel cost, Renew Sustain Energy Rev 90 (2018) 687–699, https://doi.org/10.1016/j.rser.2018.03.060.
[10] C. Finck, P. Beagon, J. Clauss, P. Thibault, P.J.C. Vogler-Finck, K. Zhang, et al., Review of applied and tested control possibilities for energy flexibility in buildings: a technical report from IEA EBC Annex 67 Energy Flexible, Buildings (2017) 1–59.
[11] Clauß J, Finck C, Vogler-finck P, Beagon P. Control strategies for building energy systems to unlock demand side flexibility – A review Norwegian University of Science and Technology , Trondheim , Norway Eindhoven University of Technology , Eindhoven , Netherlands Neogrid Technologies ApS / Aalborg. 15th Int Conf Int Build Perform 2017:611–20.
[12] A. Afram, F. Janabi-Sharifi, A.S. Fung, K. Raahemifar, Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC systems: a state of the art review and case study of a residential HVAC system, Energy Build 141 (2017) 96–113, https://doi.org/10.1016/j.enbuild.2017.02.012.
[13] Salsbury TI. A survey of control technologies in the building automation industry. vol. 16. IFAC; 2005. DOI:10.3182/20050703-6-cz-1902.01397.
[14] Behrooz F, Mariun N, Marhaban MH, Radzi MAM, Ramli AR. Review of control techniques for HVAC systems-nonlinearity approaches based on fuzzy cognitive maps. Energies 2018;11. DOI:10.3390/en11030495.
[15] A. Afram, F. Janabi-Sharifi, Theory and applications of HVAC control systems - a review of model predictive control (MPC), Build Environ 72 (2014) 343–355, https://doi.org/10.1016/j.buildenv.2013.11.016.
[16] D. Subbaram Naidu, C.G. Rieger, Advanced control strategies for heating, ventilation, air-conditioning, and refrigeration systems - an overview: part I: hard control, HVAC R Res 17 (2011) 2–21, https://doi.org/10.1080/10789669.2011.540942.
[17] Serale G, Fiorentini M, Capozzoli A, Bernardini D, Bemporad A. Model Predictive Control (MPC) for enhancing building and HVAC system energy efficiency: Problem formulation, applications and opportunities. Energies 2018;11. DOI:10.3390/en11030631.
[18] D. Subbaram Naidu, C.G. Rieger, Advanced control strategies for HVAC&R systems - an overview: part II: soft and fusion control, HVAC R Res 17 (2011) 144–158, https://doi.org/10.1080/10789669.2011.555650.
[19] M. Fiorentini, P. Cooper, Z. Ma, D.A. Robinson, Hybrid model predictive control of a residential HVAC system with PVT energy generation and PCM thermal storage, Energy Procedia 83 (2015) 21–30, https://doi.org/10.1016/j.egypro.2015.12.192.
[20] R. Halvgaard, N.K. Poulsen, H. Madsen, J.B. Jørgensen, Economic model predictive control for building climate control in a smart grid, in: 2012 IEEE PES Innov Smart Grid Technol ISGT, 2012, pp. 1–6, https://doi.org/10.1109/ISGT.2012.6175631.
[21] A.E.D. Mady, G.M. Provan, C. Ryan, K.N. Brown, Stochastic model predictive controller for the integration of building use and temperature regulation, Proc Natl Conf Artif Intell 2 (2011) 1371–1376.
[22] S. Prívara, Z. Váňa, D. Gyalistras, J. Cigler, C. Sagerschnig, M. Morari, et al., Modeling and identification of a large multi-zone office building, Proc IEEE Int Conf Control Appl (2011) 55–60, https://doi.org/10.1109/CCA.2011.6044402.

[23] S. Prívara, J. Cigler, Z. Váňa, F. Oldewurtel, C. Sagerschnig, E. Žáčeková, Building modeling as a crucial part for building predictive control, Energy Build 56 (2013) 8–22, https://doi.org/10.1016/j.enbuild.2012.10.024.
[24] G. Lymperopoulos, P. Ioannou, Energy & Buildings Building temperature regulation in a multi-zone HVAC system using distributed adaptive control R, Energy Build 215 (2020), https://doi.org/10.1016/j.enbuild.2020.109825 109825.
[25] A. Buonomano, U. Montanaro, A. Palombo, S. Santini, Temperature and humidity adaptive control in multi-enclosed thermal zones under unexpected external disturbances, Energy Build 135 (2017) 263–285, https://doi.org/10.1016/j.enbuild.2016.11.015.
[26] S. Baldi, F. Zhang, T. Le, P. Endel, O. Holub, Passive versus active learning in operation and adaptive maintenance of Heating, Ventilation, and Air Conditioning, Appl Energy 252 (2019), https://doi.org/10.1016/j.apenergy.2019.113478 113478.
[27] A.G. Barto, R.S. Sutton, Reinforcement learning: an introduction, Kybernetes 27 (1998) 1093–1096.
[28] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, et al., Human-level control through deep reinforcement learning, Nature 518 (2015) 529–533, https://doi.org/10.1038/nature14236.
[29] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, et al., Mastering the game of Go with deep neural networks and tree search, Nature 529 (2016) 484–489, https://doi.org/10.1038/nature16961.
[30] 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.
[31] Wei T, Wang Y, Zhu Q. Deep Reinforcement Learning for Building HVAC Control. Proc - Des Autom Conf 2017;Part 12828. DOI:10.1145/3061639.3062224.
[32] W. Valladares, M. Galindo, J. Gutiérrez, W.C. Wu, K.K. Liao, J.C. Liao, et al., Energy optimization associated with thermal comfort and indoor air control via a deep reinforcement learning algorithm, Build Environ 155 (2019) 105–117, https://doi.org/10.1016/j.buildenv.2019.03.038.
[33] Wang Y, Velswamy K, Huang B. A long-short term memory recurrent neural network based reinforcement learning controller for office heating ventilation and air conditioning systems. Processes 2017;5. DOI:10.3390/pr5030046.
[34] Gao G, Li J, Wen Y. Energy-Efficient Thermal Comfort Control in Smart Buildings via Deep Reinforcement Learning 2019:1–11.
[35] P. Fazenda, K. Veeramachaneni, P. Lima, U.M. O'Reilly, Using reinforcement learning to optimize occupant comfort and energy usage in HVAC systems, J Ambient Intell Smart Environ 6 (2014) 675–690, https://doi.org/10.3233/AIS-140288.
[36] 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), https://doi.org/10.1016/j.buildenv.2019.106535 106535.
[37] J.R. Vázquez-Canteli, S. Ulyanin, J. Kämpf, Z. Nagy, Fusing TensorFlow with building energy simulation for intelligent energy management in smart cities, Sustain Cities Soc 45 (2019) 243–257, https://doi.org/10.1016/j.scs.2018.11.021.
[38] J. Vázquez-Canteli, J. Kämpf, Z. Nagy, Balancing comfort and energy consumption of a heat pump using batch reinforcement learning with fitted Q-iteration, Energy Procedia 122 (2017) 415–420, https://doi.org/10.1016/j.egypro.2017.07.429.
[39] Z. Zhang, A. Chong, Y. Pan, C. Zhang, S. Lu, K.P. Lan, A deep reinforcement learning approach to using whole building energy model for HVAC optimal control, Build Perform Model Conf 2018 (2018) 675–682.
[40] J.R. Vázquez-Canteli, Z. Nagy, Reinforcement learning for demand response: a review of algorithms and modeling techniques, Appl Energy 235 (2019) 1072–1089, https://doi.org/10.1016/j.apenergy.2018.11.002.
[41] M. Han, R. May, X. Zhang, X. Wang, S. Pan, D. Yan, et al., A review of reinforcement learning methodologies for controlling occupant comfort in buildings, Sustain Cities Soc 51 (2019), https://doi.org/10.1016/j.scs.2019.101748 101748.
[42] G.T. Costanzo, S. Iacovella, F. Ruelens, T. Leurs, B.J. Claessens, Sustainable Energy, Grids and Networks Experimental analysis of data-driven control for a building heating system, Sustain Energy, Grids Networks 6 (2016) 81–90, https://doi.org/10.1016/j.segan.2016.02.002.
[43] D.B. Crawley, C.O. Pedersen, L.K. Lawrie, F.C. Winkelmann, Energy plus: energy simulation program, ASHRAE J 42 (2000) 49–56.
[44] M.A. Wiering, Explorations in efficient reinforcement learning PhD Thesis, Univ Amsterdam, 1999.
[45] H. Van Hasselt, A. Guez, D. Silver, Deep reinforcement learning with double Q-Learning, in: 30th AAAI Conf Artif Intell AAAI, 2016, pp. 2094–2100.
[46] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, et al., OpenAI Gym (2016) 1–4.
[47] Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems 2016.
[48] F. Chollet et al., Keras (2015).
[49] Dulac-Arnold G, Evans R, van Hasselt H, Sunehag P, Lillicrap T, Hunt J, et al. Deep Reinforcement Learning in Large Discrete Action Spaces 2015.
[50] G. Hinton, Tieleman T. Rmsprop, Divide the gradient by a running average of its recent magnitude. coursera: neural networks for machine learning, Tech Rep, Tech Rep (2012) 31.
[51] Lonza A. Reinforcement Learning Algorithms with Python 2019.