

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

Original

Deep reinforcement learning to optimise indoor temperature control and heating energy consumption in buildings / Brandi, S.; Piscitelli, M. S.; Martellacci, M.; Capozzoli, A.. - In: ENERGY AND BUILDINGS. - ISSN 0378-7788. - 224:(2020), p. 110225. [10.1016/j.enbuild.2020.110225]

Availability:

This version is available at: 11583/2837827 since: 2020-07-01T12:03:55Z

Publisher:

Elsevier Ltd

Published

DOI:10.1016/j.enbuild.2020.110225

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

Elsevier postprint/Author's Accepted Manuscript

© 2020. This manuscript version is made available under the CC-BY-NC-ND 4.0 license
<http://creativecommons.org/licenses/by-nc-nd/4.0/>. The final authenticated version is available online at:
<http://dx.doi.org/10.1016/j.enbuild.2020.110225>

(Article begins on next page)

Deep Reinforcement Learning to optimise indoor temperature control and heating energy consumption in buildings

Silvio Brandi ^a, Marco Savino Piscitelli^a, Marco Martellacci^b and 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

* Corresponding author: Tel: +39-011-090-4413, fax: +39-011-090-4499, e-mail: alfonso.capozzoli@polito.it

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.

Keywords: deep reinforcement learning, building adaptive control, energy efficiency, temperature control, HVAC.

28 **1. Introduction**

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

38 Energy Management and Information Systems (EMIS) enable building owners to
39 operate their buildings more efficiently and with improved occupant comfort. According to
40 [6] EMIS can be categorized in three main families of data analytics-based tools including
41 Energy Information System (EIS), Fault Detection and Diagnosis systems (FDD) and
42 Automated System Optimisation (ASO) tools. ASO tools offer the opportunity to
43 continuously analyse and modify control settings for optimising the building system energy
44 usage. Advanced control strategies are mainly enabled by the progressive introduction of
45 Advanced Metering Infrastructure (AMI) which allow the collection, storage, and analysis
46 of a vast amount of building-related data. For this reason, the information gathered, if it is
47 properly processed through data-driven procedures, may provide crucial knowledge on the
48 actual and future building operational status including exogenous and endogenous variables
49 influencing control performance [7,8].

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

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

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

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

84 Among hard control methods [14], Model Predictive Control (MPC) aims at facing the
85 main challenges of HVAC system control such as non-linear and time-varying dynamics and
86 disturbances through an optimisation process performed over a receding time horizon
87 [17,18]. The current scientific literature includes several works in which MPC was
88 successfully applied to complex HVAC systems [19–21]. However, its application requires
89 the definition of accurate models of the controlled environment [22,23] limiting MPC
90 widespread adoption in the building industry. To overcome these limitations hybrid control
91 strategies, such as adaptive control, have been successfully applied to HVAC systems [23–
92 25] . Adaptive controllers do not require a priori identification of system parameters, as
93 unknown parameters are estimated in real time through a parameter estimator which provides
94 to the controller enough flexibility to adapt to time-varying disturbances and to account for
95 uncertainty.

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

105 Reinforcement Learning to address HVAC systems control is reported with the aim of
106 introducing the current knowledge gaps and the consequent contribution of the present paper.

107 **1.1. Related works to the application of Deep Reinforcement Learning control in** 108 **HVAC systems**

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

114 The first application of RL to HVAC systems dates back to 1998[16], from this year up
115 to 2012 the number of scientific publications about RL application to energy systems was
116 limited to few works per year. From this period the interest of the scientific community about
117 RL control framework has increased also due to recent advancements in deep learning.
118 Recent studies exploited DRL for the regulation of supply water temperature setpoint [30],
119 supply air flow-rate [31], supply air temperature[32], indoor temperature or humidity setpoint
120 [33–35], fan speed or damper position [32,36] and tank temperature setpoint [37,38].

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

130 Two exhaustive review work were recently published focusing on the application of RL
131 control in buildings for demand response [40] and occupant comfort [41] respectively. In
132 [40] were identified four major categories of energy systems where RL and DRL are applied:
133 HVAC and Domestic Hot Water (DHW) systems, Appliances, Electric Vehicles (EV) and
134 distributed generation coupled with storage systems. The authors identified a significant lack
135 in real-world studies of RL controllers that may cause scepticism of building owners and
136 managers about this technology. Moreover, the integration of RL with actual human feedback
137 and the development of Multi-Agent Reinforcement Learning Controllers (MARL) were
138 recognized as promising trends for future research in the energy and building sector.

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

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

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

166 **2. Knowledge gaps and contribution of the paper**

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

170 A DRL agent is characterized by a number of hyperparameters that need to be carefully
171 tuned depending on the specific case study and objective functions [39]. As a consequence,
172 despite its model-free nature, DRL requires a sort of modelling effort in its initial state to find
173 the set of hyperparameters which may lead to the learning of a control policy close to the
174 optimum in less time as possible and with an acceptable uncertainty [32]. In the existing
175 literature an analysis on the effect of the hyperparameters settings on the performance of the
176 control strategy was poorly investigated. Moreover, two opposite approaches can be followed
177 when deploying a DRL agent previously trained offline: static deployment and dynamic
178 deployment [30]. In the static deployment approach, the agent is implemented in the control
179 loop as a static entity, meaning that the control policy is no longer updated, and any learning
180 goes on. The advantages of such approach are the limited computational cost and the relative

181 stability provided by a static control policy. The disadvantage is that the agent is unable to
182 automatically adapt in the case key-features of the controlled system change (e.g. revamping
183 intervention) and may need to be retrained. Conversely, in the dynamic deployment
184 approach, the agent continuously learns from experience constantly updating its control
185 policy. Following this approach a DRL agent can adapt to a changing system at the expense
186 of higher computational cost and with the risk of stability issues for the control policy [30].

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

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

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

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

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

230 **3. Reinforcement Learning: concept and formulation**

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

$$241 \quad v_{\pi}(s) = E[r_{t+1} + \gamma v_{\pi}(s') | S_t = s, S_{t+1} = s']$$

242 Equation 1

243 Where γ [0,1] is the discount factor for future rewards [27]. An agent employing a
 244 discount factor equal to 1 will give greater importance to rewards that can be obtained in the
 245 future. Whereas, an agent implementing a discount factor of 0 will assign higher values to
 246 states that lead to high immediate rewards. Similarly, the *action-value function* represents the
 247 expected return of the agent when selecting action a starting from state s and following policy
 248 π :

$$249 \quad q_{\pi}(s, a) = E[r_{t+1} + \gamma q_{\pi}(s', a') | S_t = s, A_t = a]$$

250 Equation 2

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

$$255 \quad Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

256 Equation 3

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

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

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

Equation 4

Where τ is the Boltzmann temperature constant. Typically, the learning process is initialized with high values of ε (e.g. $\varepsilon = 1$ 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

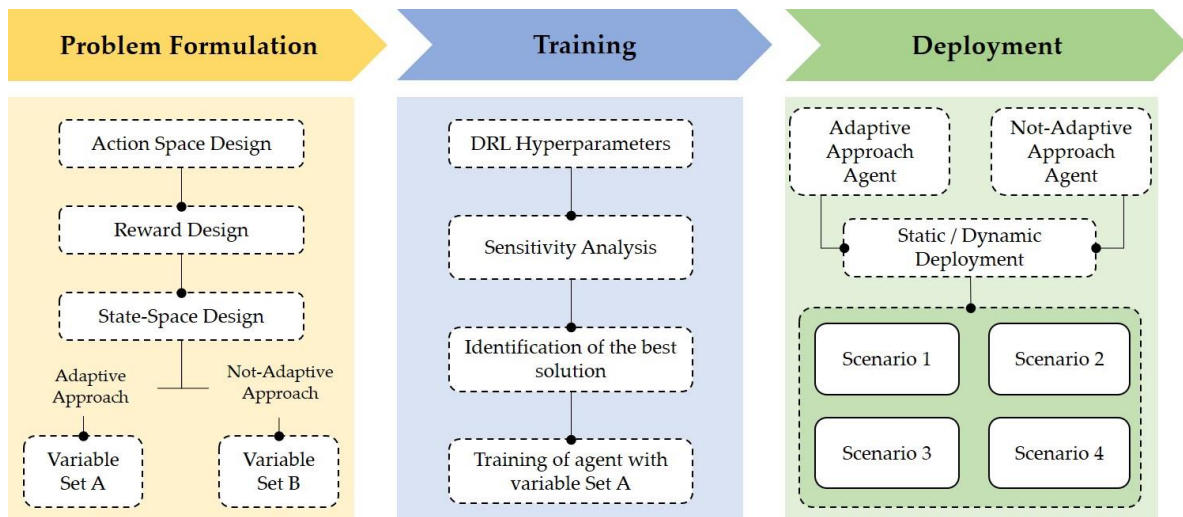
In its classical formulation, Q-learning algorithm employs lookup tables to store and retrieve state-action values where each entry represents a state-action tuple (s, a) . However, adopting a tabular representation may be unfeasible in practical problems where the state and action spaces are very large. 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

301 controlled environment. At each control step, the Q-values are updated according to Equation
302 3 and used as targets to retrain the deep neural network.

303 Some improvements were introduced in literature in order to improve the DQN
304 formulation. The first one is the introduction of the replay memory to store previous
305 experience obtained by the agent. In the optimisation process of the network weights a
306 random mini batch is extracted from the replay memory and used to fit DNN-regression using
307 as targets Q-values updated according to Equation 3. This enables the re-utilization of
308 previous experience collected by the agent and overcome the problem of correlated
309 observations while performing the optimisation process. The second improvement involves
310 the employment of two neural networks[45]. The first one, called *online network*, is
311 constantly updated and directly used in the interaction with the environment; the second one,
312 called *target network*, is updated after N iterations and used to predict target values. The
313 target network is an exact copy of the online network and during the update the weights of
314 the online network are simply copied into the target network. In the present work Double
315 Deep Q-Learning with Memory Replay implementing the Max-Boltzmann exploration rule
316 was applied to develop a DRL control to optimize both heating energy consumption and
317 indoor thermal conditions. The whole process was trained and tested in a simulation
318 environment developed by the authors.

319 **4. Framework of the analysis**

320 In this section the methodological framework is presented with the aim of introducing
321 each stage of the DRL control agent development. The present framework unfolds over three
322 different stages as shown in Figure 1.



323

324

Figure 1 - Framework of the application of DRL control.

325

326

327

328

329

330

331

332

333

334

335

336

337

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.

338

339

340

341

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

342 to analyse the variations in the results obtained. The training process was implemented in an
343 offline fashion using a training episode (i.e. a time period representative of the specific
344 control problem) multiple times to constantly refine agent's control policy. The sensitivity
345 analysis was performed for an agent implementing the variable set A. The best configuration
346 of hyperparameters resulting from the analysis was successively employed to train the agent
347 with variable set B. Details on the DRL training stage are provided in section 6.5.

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

356 5. Description of the simulation environment

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

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

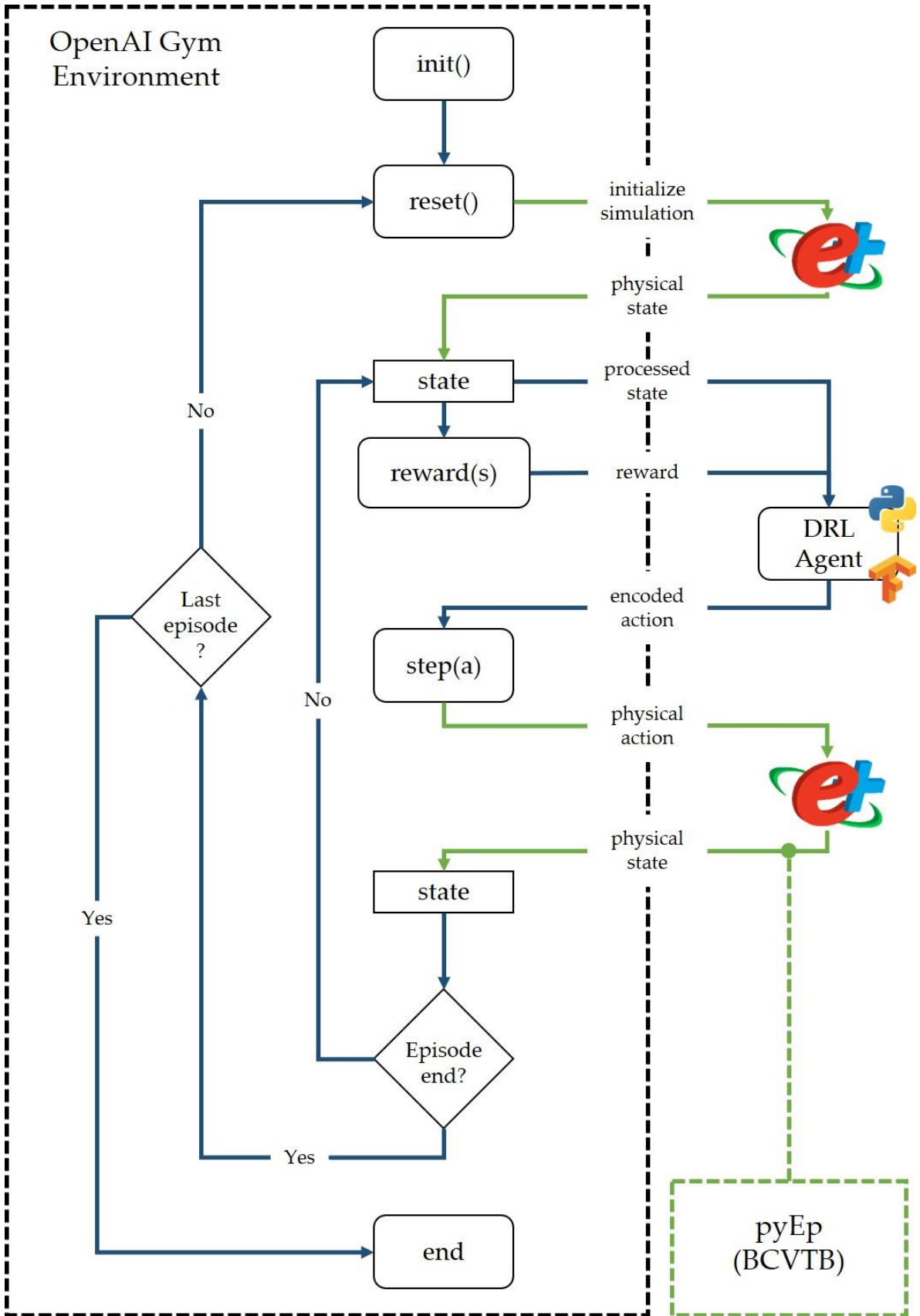
- 369 • *Control time step*: it represents the time step during which the action is taken by the
370 agent. In this application the control time step was set equal to 15 minutes.
- 371 • *Simulation time step*: it is defined in the EnergyPlus environment and it is not directly
372 linked to control time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

383 Figure 2 shows the information flow that occurs during a simulation of DRL control
384 interacting with the EnergyPlus simulation model. The dynamic simulation starts with the
385 initialization (*init()* function) of the OpenAI Gym environment which is formalized as a
386 Python class. The *reset()* function is called at the beginning of each episode. This function
387 re-initializes the EnergyPlus simulation process performing the simulation warm-up and
388 returning the first state of the environment (e.g. the initial state of the building at the
389 beginning of the simulation process). The state returned by EnergyPlus is defined as physical
390 quantities and must be processed before they are provided to the DNN of the DRL agent.

391 Details about the selection of the variables included in the state for the specific case study
392 are presented in section 6.4.3.

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

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



403

404

405

Figure 2 – Architecture of simulation environment for RL control in HVAC systems

406 **6. Case study**

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

411 **6.1. Building description**

412 The simulated building is representative of a huge portion of the Italian building stock
413 in terms of both heating system configuration and building construction features. It is a six-
414 level mixed-use building with a net heated surface of 9300 m² located in Turin, Italy. The
415 indoor environment is heated through water terminal units (i.e., radiators). The building is
416 composed of three thermal zones served by different hot-water circuits and was built between
417 1930 and 1960. The average transmittance values of the opaque and transparent envelope
418 components are respectively 1.084 and 2.921 W/m²K. The ratio between heat transfer surface
419 and gross volume (i.e, aspect ratio) is equal to 0.25 m⁻¹. The implementation of the DRL
420 controller is tested for one thermal zone which includes only office rooms. This zone is
421 composed of four-levels with a net heated surface of 7000 m² and a net heated volume of
422 33000 m³. The remaining zones are occupied by the local police department and the warden
423 of the whole building. Figure 3 shows a picture of the real building and highlights the thermal
424 zone modelled in this work.



425

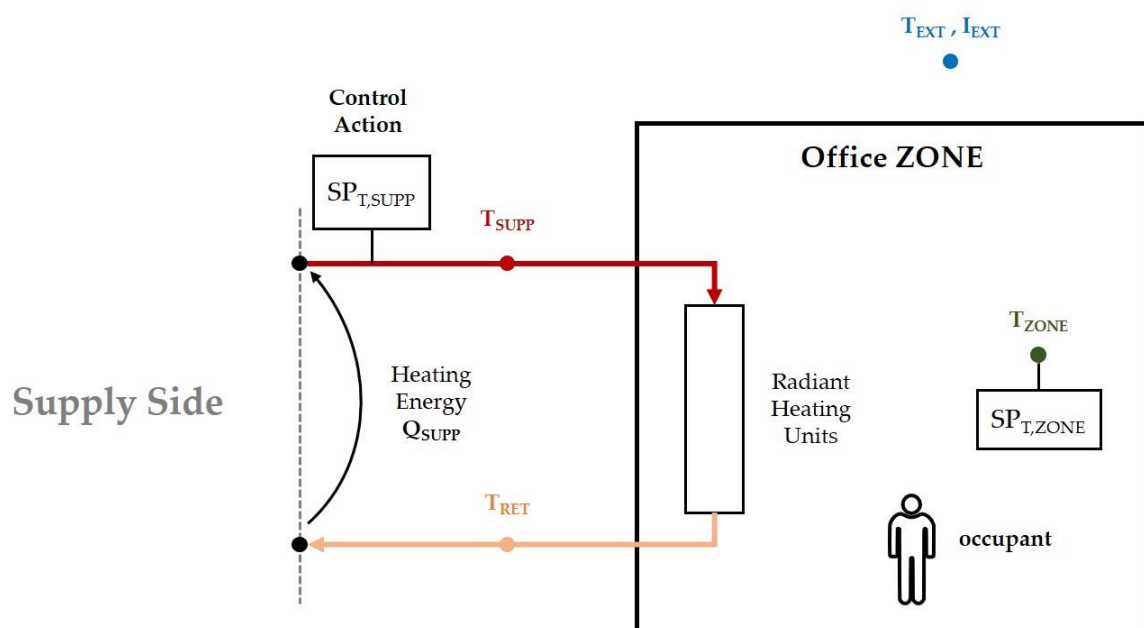
426 Figure 3 - Office case study located in Torino, Italy. Detail of the office zone modelled in this work.

427 **6.2. Heating system and control objectives**

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

436 In the present case study, the control problem focuses on the regulation the supply water
437 temperature (T_{SUPP}) to heating terminal units of a single thermal zone. The heating system
438 was modelled in EnergyPlus with a single hot water loop. The supply side includes a single
439 gas fired boiler (*Boiler:HotWater*) and a constant speed pump (*Pump:ConstantSpeed*). The
440 supply water temperature setpoint (SP_{TSUPP}) was managed through a
441 *SetPointManager:Scheduled* which directly receives inputs from Python through the
442 *ExternalInterface*. The demand side includes one thermal zone and its relative bypass branch.

443 The goal of the control policy is to reduce the amount of thermal energy provided to the
 444 supply water while maintaining indoor air temperature within an acceptability range during
 445 occupied periods. This application, even being developed in a simulation environment in
 446 which every thermal comfort-based parameter can be easily evaluated, considers only the
 447 zone air temperature (T_{ZONE}). In fact, other comfort related-variables are not monitored in the
 448 real building. Moreover, the water terminal units can control only the sensible part of the
 449 thermal load. If the zone air temperature value falls between upper and lower threshold of a
 450 pre-defined acceptability range, then indoor temperature requirements are satisfied. In this
 451 application the acceptability range was defined in the interval $[-1,1]$ °C from the desired
 452 indoor temperature setpoint ($SP_{T,\text{ZONE}}$). The work focuses on the energy supplied for heating
 453 the carrier fluid (Q_{SUPP}) regardless the type of the generation system serving the building.
 454 Technically, in real life implementations, the regulation of supply water temperature can be
 455 achieved through different solutions such as three-way mixing valves or by modulating boiler
 456 or heat pumps. The control policy developed through the presented approach could be then
 457 employed independently by the actual generation system installed. Figure 4 provides a
 458 simplified scheme of the heating system and of the control problem formulation.



459

460

Figure 4 - Schematic of the heating system analysed.

461 **6.3. Baseline control logic**

462 The performance of the DRL control was evaluated against a baseline control logic
463 implementing a combination of rule-based and climatic-based logics for the control of the
464 supply water temperature. The starting time of the heating system was determined according
465 to the value of indoor temperature and the amount of time before the occupant's arrival. The
466 controller is enabled to turn on the heating system up to four hours before the arrival of the
467 occupants if the difference between the actual indoor temperature and the low threshold of
468 the acceptability range is higher than 3°C, or up to three hours before if that difference is
469 higher than 2°C. In any other case the controller turns on the heating two hours before
470 occupant's arrival if the zone temperature is lower than the low threshold of the acceptability
471 range. When the zone reaches the upper threshold of the acceptability range the heating
472 system is turned off. If the zone temperature falls below the lower threshold the heating
473 system is turned on again. This control strategy is operated until two hours before occupants
474 leave the building, when the heating system is turned off to exploit thermal inertia until the
475 next day. The supply temperature value is linearly interpolated between a maximum value of
476 70 °C when the outdoor air temperature falls below -5 °C and a minimum value of 40 °C
477 when the outdoor air temperature is over 12 °C. These values were selected according to the
478 control logic of the supply temperature actually implemented in the Energy Management
479 System of the real building.

480 **6.4. Design of DRL control problem**

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

485 **6.4.1. Design of the action-space**

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

$$490 \quad A_t = [20, 40, 50, 60, 70]$$

491

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

503 **6.4.2. Design of the reward function**

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

510 agent in taking energy-intensive actions when the outdoor temperature is very low and vice-
511 versa.

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

$$518 \quad R = -\beta * \frac{Q_{supp}}{SP_{T,ZONE} - T_{EXT}} - \rho * |(SP_{T,ZONE} - T_{zone})^2|_{OCC=1}$$

519 Equation 5

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

522 **6.4.3. Design of the state-space**

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

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

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

536 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

537 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

538

539 *External Air Temperature* and *Direct Solar Radiation* were both included in variable set
 540 B, as they are the most influencing ambient variables affecting building heating energy
 541 consumption and indoor temperature. On the contrary, in the feature set A, *External Air*
 542 *Temperature* was substituted by the variable *ΔT Indoor Setpoint – External Air* since it is
 543 directly related to the formulated reward function (Equation 5). This formulation was found

544 to be effective in removing the dependency of the learnt control policy from a fixed value of
545 indoor temperature setpoint which could limit agent adaptability.

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

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

568 Eventually, the agent needs information about the zone air temperature which is directly
569 connected with the temperature-related term of the reward function. This information was

570 straightforwardly added to the *variable set B* along with its 3 lagged values in the past (15,
 571 30 and 45 minutes lag respectively) and the *Indoor Setpoint*. Contrarily, in *variable set A*,
 572 this information was provided indirectly introducing as variable the difference between the
 573 *Zone Air Temperature* and *Indoor Setpoint* along with its 3 lagged values in the past (15, 30
 574 and 45 minutes lag respectively).

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

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

580 6.5. Setting of training phase

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

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

590 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

591

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

601

602 Table 4 – Different hyperparameter configurations implemented in the training phase.

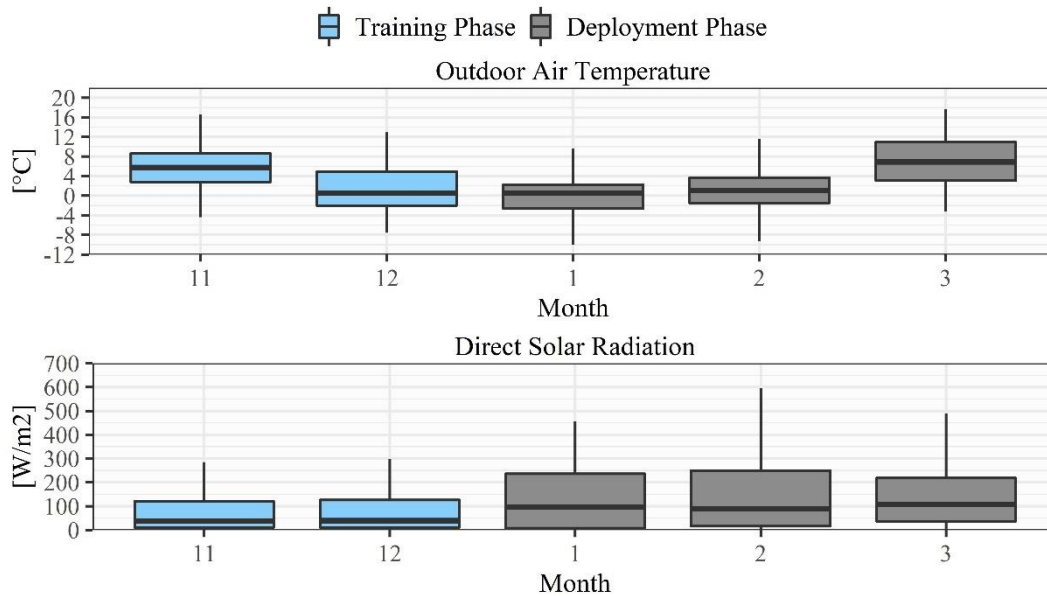
run	Discount Factor γ	Weight Factor ρ
1,2,3	0.9	10
4,5,6	0.95	10
7,8,9	0.99	10
10,11,12	0.9	20
13,14,15	0.95	20
16,17,18	0.99	20
19,20,21	0.9	1
22,23,24	0.95	1
25,26,27	0.99	1

603 The performance of Deep Reinforcement Learning is affected by the stochastic
604 behaviour that is intrinsic in both deep neural networks and controlled environments. In order
605 to account for this aspect, each configuration has been ran three times employing multiple
606 random seeds in order to ensure consistency according to [51]. Successively, the

607 hyperparameters of the run leading to the best performance in terms of both energy savings
608 and temperature control were selected to train also the agent implementing *variable set B*.

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

619 Figure 5 shows the patterns of outdoor air temperature and direct solar radiation in the
620 two periods (i.e. training and deployment period). For the sake of legibility, the solar radiation
621 values include only daylight period. The training period was selected for its wide range of
622 temperature values spanning between -8 °C and 17 °C while the direct solar radiation is
623 higher during the deployment period. However, this latter aspect allows to test the
624 adaptability of DRL agent different climatic patterns from those used for the training. In the
625 training phase occupancy was simulated between 07:00 and 19:00 from Monday to Saturday.
626 The required indoor setpoint was set equal to 21 °C and the temperature acceptability range
627 between 20 °C and 22 °C.



628

629

Figure 5 – Outdoor Air Temperature patterns during training and deployment periods.

630

6.6. Deployment phase

631

632

633

634

635

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:

636

637

638

639

- 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.

640

641

642

643

644

- 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 Figure 6 maintaining unchanged the zone temperature setpoint respect to the training

645 conditions. The lighting and electric appliances schedules were also changed according
 646 to the new occupancy schedule.



647
 648 Figure 6 – Occupancy schedules and indoor setpoint in different design conditions.

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

655 7. Results

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

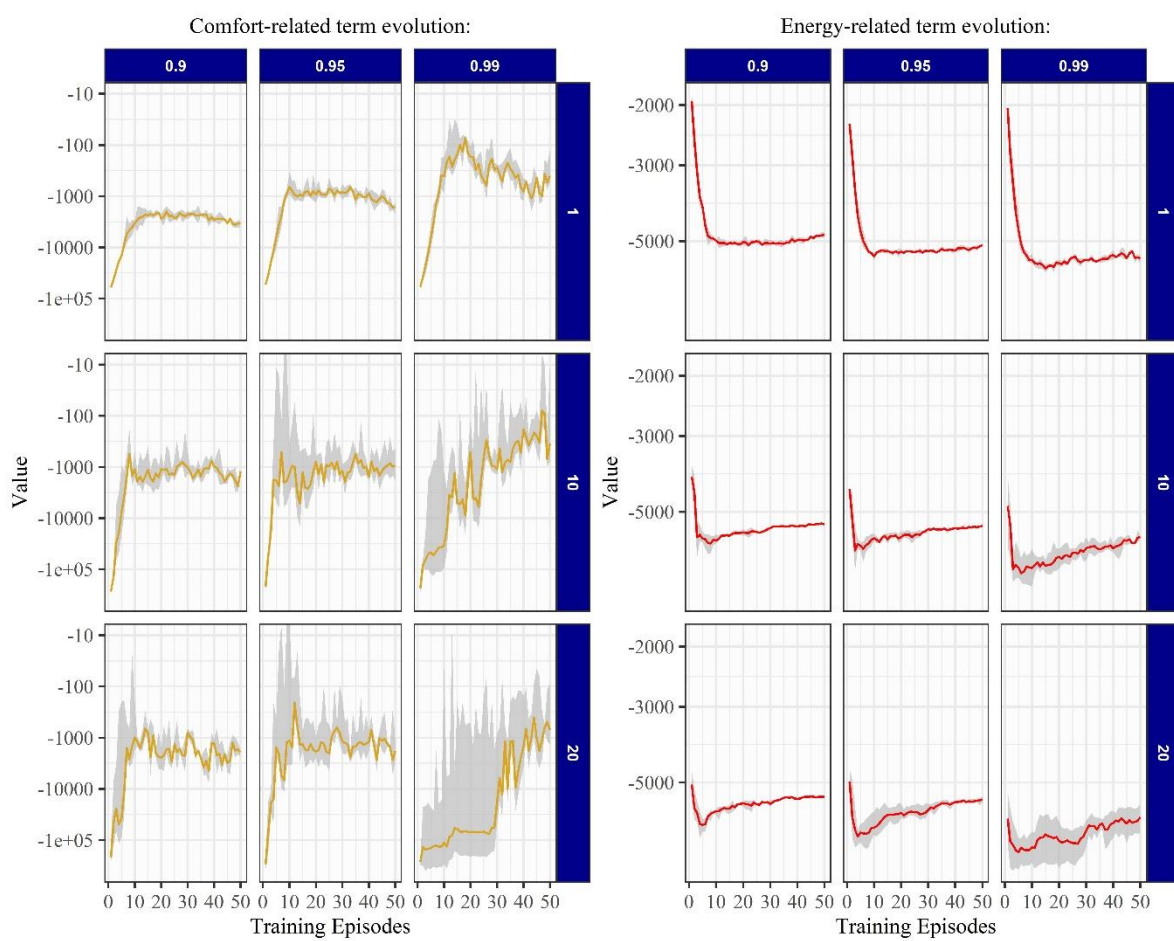
661 7.1. Results of the training process

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

664 of the control algorithm. The variable set based on adaptive approach introduced in section
665 6.4.3 was implemented for this sensitivity analysis.

666 A useful indicator to assess the goodness of the learning process of a DRL agent is
667 represented by the evolution of the cumulative reward per episode. The reward, which has
668 not a direct physical meaning, takes into consideration both the energy consumption and
669 indoor temperature values and combines them in a single value. Higher values of the reward
670 correspond to a better performance obtained by the control agent. It is important to supervise
671 if the reward converges to a stable value. A non-convergent trend in the reward may be caused
672 by an agent that failed in achieving an optimal control policy. To this purpose, the
673 convergence of the different configurations of the agent were analysed in the episode-reward
674 plot showed in Figure 7. The figure is split into two main panels representing the evolution
675 of the energy-related term and temperature-related term respectively. Each main panel is
676 furtherly organized in a grid in which each sub-panel represents a specific configuration of
677 the hyperparameters. Each sub-panel shows the evolution of the relative term of the reward
678 function during the training episode. The solid line shows the average value per episode of
679 the three different runs performed for each configuration, while the grey area was drawn
680 between maximum and minimum value per episode. In all the configurations the agent starts
681 exploring high values of the energy-related term and extremely low values of the
682 temperature-related term. Across the different runs, the agent firstly learns how to correctly
683 maintain indoor temperature during the first 20 episodes; this fact can be observed by
684 analysing the increase of the temperature-related term values and the relative decrease of the
685 energy-related term. From this stage (i.e. 20th episode) the agent begins to learn how to
686 reduce energy consumption while keeping indoor temperature in the range it previously
687 learned. In fact, the values of the temperature-related term are quite stable while the values
688 of the energy-related term increase. Agents that were initialized with a discount factor γ equal
689 to 0.99 represent an exception, showing highest variance in terms of temperature control

690 performance. The training runs performed with this specific configuration ($\gamma=0.99$) seek to
 691 obtain higher rewards in a longer time horizon compared to other agents generating an
 692 instability in the objective function. This aspect is particularly clear observing the evolution
 693 of the temperature-related term of the agent implementing a discount factor of 0.99 and a
 694 weight of the temperature-related term equal to 20. On the other hand, agents applying a
 695 discount factor equal to 0.9 shows the higher stability among all the training configurations
 696 due to the shorter time horizon considered.



697

698
699

Figure 7 - Evolution of energy-related and temperature-related term of the reward function during training phase.

700

701

702

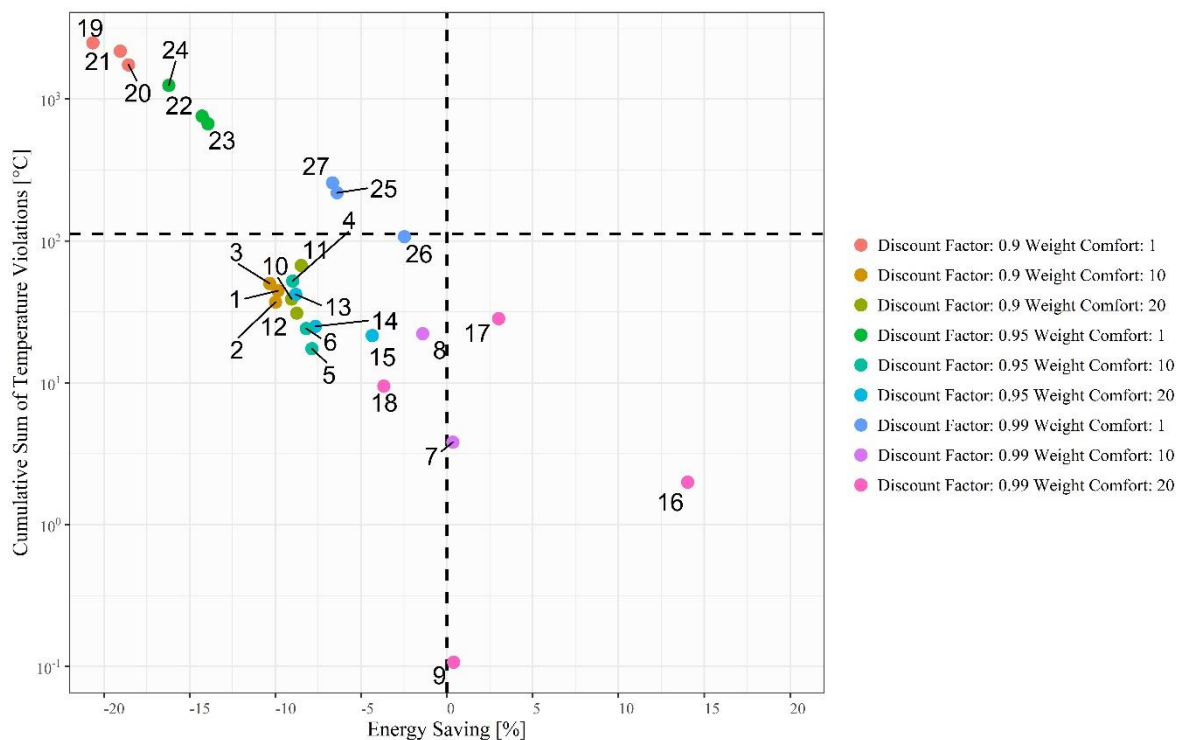
703

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.

704 While the energy performance can be straightforwardly evaluated comparing the amount
705 of heating energy supplied to the water, the temperature control performance requires the
706 definition of an appropriate metric. In the present work, the indoor temperature control
707 performance was evaluated by calculating the cumulative sum of temperature violations
708 during occupancy hours. A temperature violation occurs when the building is occupied, and
709 the indoor temperature falls outside the acceptability range. The magnitude of the
710 temperature violation is then calculated as the absolute difference between actual indoor
711 temperature and desired set point value at each simulation step. The cumulative value of this
712 quantity over an entire episode returns the performance of the control algorithm expressed in
713 °C.

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

730 interpreted as an indicator of the consistency of the control policy learned by such agents. As
 731 can be expected, agents implementing a weight factor of the temperature-related term equal
 732 to 1 achieved greater energy savings at the cost of worse temperature control. Following these
 733 considerations, the agent number 2, with a discount factor of 0.9 and a weight factor ρ of 10,
 734 was selected as best solution among configurations explored in the sensitivity analysis
 735 process.



736

737

738

Figure 8 – DRL control performance in the last episode of the training phase. Each point refers to a different training runs as reported in Table 4.

739

740

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

741

742

743

744

745

746

In Figure 9 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 ($\rho = 10$) 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

747 panels of the figure. However, in the solution obtained considering a discount factor equal to
 748 0.9, the agent learnt to better maintain the indoor temperature across lower threshold of the
 749 acceptability range. As can be observed from the left figure, the run performed with a
 750 discount factor of 0.99 considerably anticipated the start-up phase resulting in higher energy
 751 consumption compared to other solutions. Given the higher discount factor, this agent learnt
 752 how to optimise the rewards stream in a longer horizon causing higher instability. The agent
 753 implementing a discount factor of 0.9 selected higher values of the supply water temperature
 754 during the first hours of the morning. As a result, the zone air temperature reached exactly
 755 the lower threshold of the acceptability range (20°C) at the beginning of the occupied period
 756 (07:00). This agent led to a heating energy saving of about 100 kWh in comparison with the
 757 agent implementing a discount factor of 0.95 that shows a similar pattern of indoor air
 758 temperature.

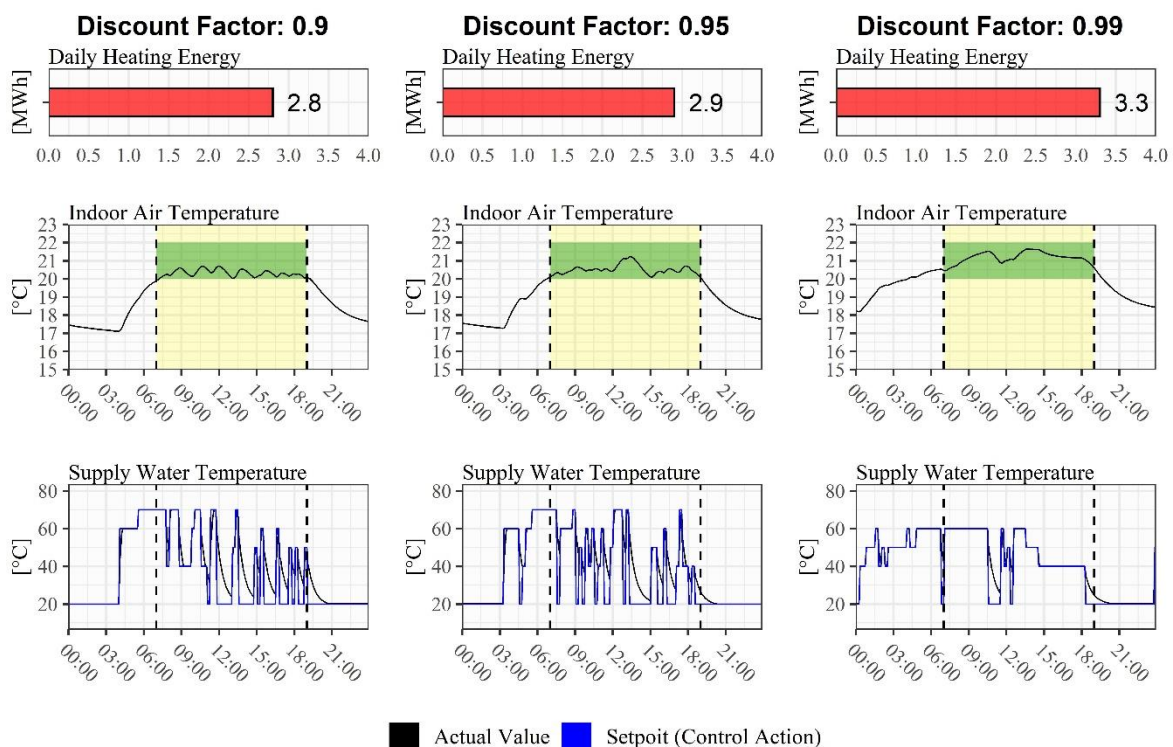


Figure 9 – Comparison between agents implementing different discount factors during a training day.

761 Figure 10 reports the performance of the trained agents considering different values of
 762 the weight factor ρ and a constant discount factor ($\gamma = 0.9$). It is possible to notice the relative
 763 importance given to temperature violations obtained in the three different solutions.

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

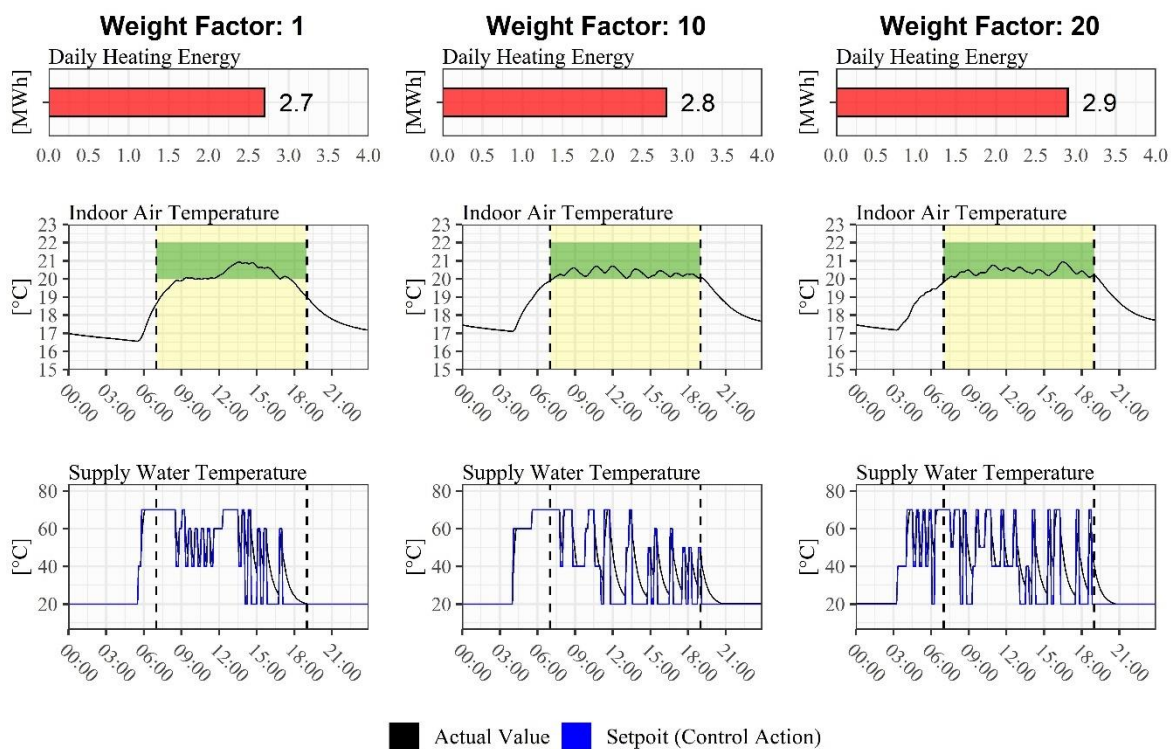


Figure 10 – Comparison between agents implementing different weight factors of the temperature-related term during a training day.

At the end of the training phase, the same hyperparameter configurations of the best solution resulting from sensitivity analysis (i.e., discount factor $\gamma = 0.9$ and weight factor $\rho = 10$) 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

776 two agents relative to the last (50th) training episode which lasts for 2 months between the 1st
 777 of November and 31st December.

778 Table 5 – Performance comparison at the end of the training phase between agents implementing
 779 adaptive and non-adaptive variable set in the definition of the state-space ($\gamma=0.9$, $\rho=10$).

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

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

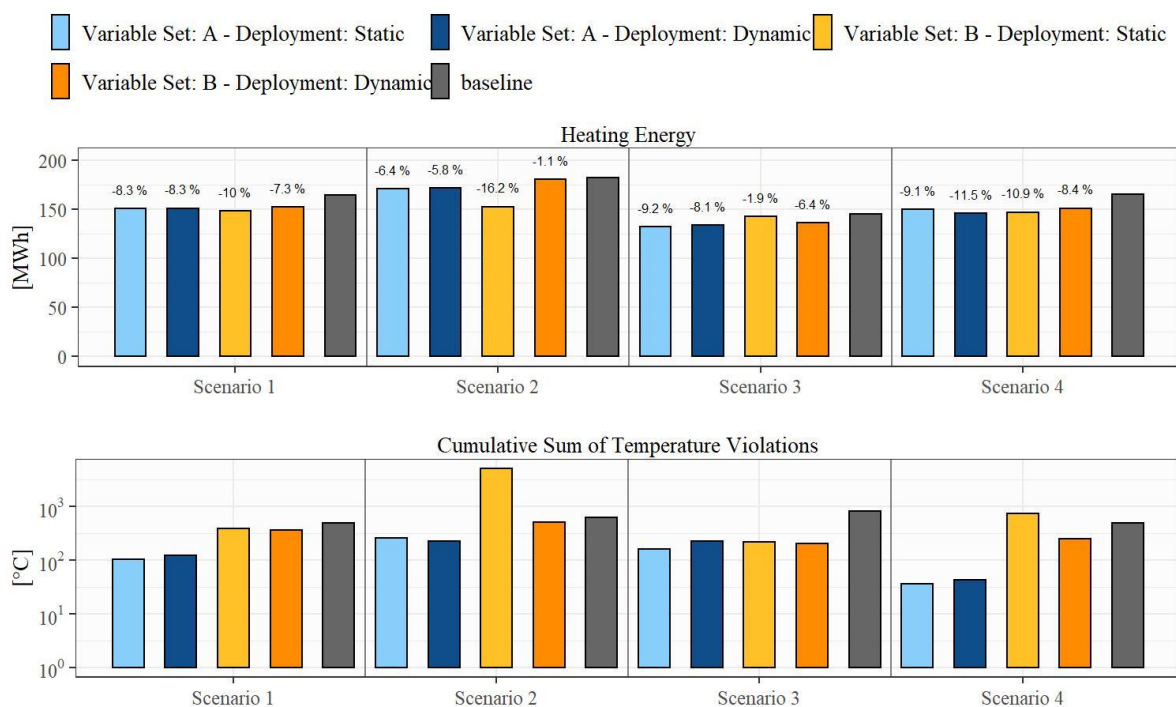
790 7.2. Results of the deployment phase

791 In this last section are analysed the results of the deployment of the two agents (trained
 792 with *variable set A and B* and considering $\rho=10$ and $\gamma=0.9$) in the four different scenarios
 793 introduced in section 6.6. The deployment of each agent was simulated both in a static and
 794 dynamic way for one episode. As previously introduced, the deployment episode is 3 months
 795 long, including January, February and March, and the climatic data employed in the
 796 simulation are gathered from the reference weather file referred to Torino (*ITA_TORINO-*
 797 *CASELLE_IGDG.epw*). Figure 11 summarises the performance of the agents in terms of

798 supplied heating energy and cumulative sum of temperature violations. The performance of
799 the agent trained with the *variable set A* did not produce always with dynamic deployment
800 configuration an improvement with respect to static deployment across the four scenarios
801 (azure and blue bars in the figure). In particular, in scenarios S2 and S3 the dynamically
802 deployed agent achieved a lower energy saving compared to its statically deployed
803 counterpart. In scenario S2 this led to a slight improvement of temperature control
804 performance while in scenario S3 the temperature control was performed with less accuracy
805 compared to statically deployed agent. Even without updating its control policy the agent
806 trained with the *variable set A* is capable to adapt to the different requirements in the different
807 scenarios achieving better performance than the baseline controller. The agent based on
808 *variable set B*, instead, shows opposite behaviour and the effect of dynamic deployment over
809 static deployment is particularly significant (yellow and orange bars in the figure). For
810 example, in the scenario S2, which considers an increased temperature setpoint compared to
811 training condition, the statically deployed agent obtained the lowest consumption (yellow bar
812 in the first panel of the bottom figure) but an extremely high value of the cumulative sum of
813 temperature violations (yellow bar in the second panel of the bottom figure) meaning that the
814 control policy was not able to adapt to the new indoor temperature requirements. On the
815 contrary, the dynamically deployed agent in the same scenario achieved an overall
816 performance comparable with agent implementing the *variable set A* conceived with an
817 adaptive approach.

818 A similar condition occurred also for the fourth scenario, which considers the presence
819 of the occupants during Sunday (contrarily the training period) where the dynamic
820 deployment drastically improved the indoor temperature control performances of the agent
821 trained with *variable set B*. The same agent (trained with *variable set B*) shows a different
822 pattern in the third scenario. In this case, in which the desired indoor setpoint was reduced
823 from 21 °C to 20°C, the statically deployed solution was capable to achieve satisfying

824 temperature control performance (yellow bar in the third panel of the bottom figure), but it
 825 obtained lower energy saving. On the contrary, the dynamically deployed solution achieved
 826 almost the same temperature control performance (orange bar in the third panel of the bottom
 827 figure) but increased the energy savings obtained from 1.9% to 6.4%. Also in this case the
 828 dynamic deployment was found to be effective in improving performance of the agent by
 829 means of continuous refinement of the control policy during the deployment episode.
 830 However, as the Figure 11 clearly shows, even in the dynamic deployment configuration the
 831 agent trained with *variable set B* was not able to achieve the performance of the agent trained
 832 with *variable set A* across all the four scenarios.

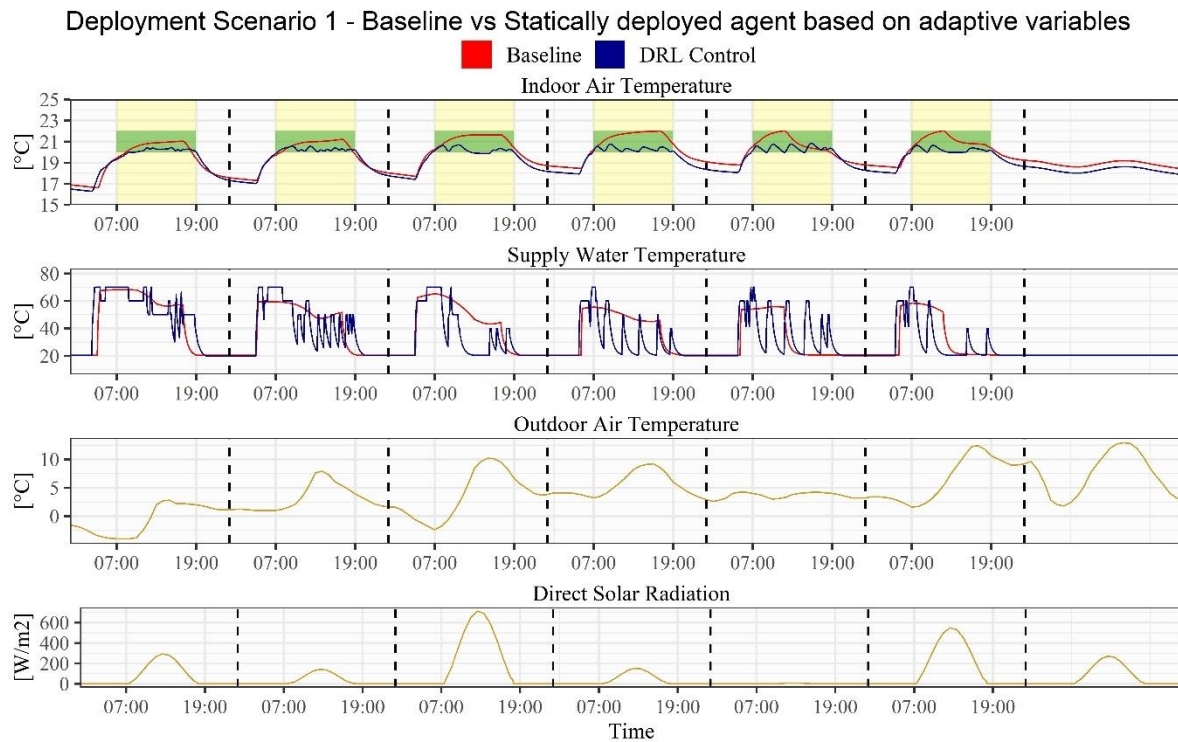


833

834 Figure 11 – Heating energy supplied and cumulative sum of temperature violations for agents trained
 835 with both variable sets in four different scenarios under static and dynamic deployment
 836 configuration. In the upper part of the figure are reported on the bars the heating energy saving
 837 respect to the baseline.

838 Figure 12 shows a comparison between statically deployed agent trained with *variable*
 839 *set A* and the baseline controller during a week of the deployment period. The plot shows the
 840 indoor air temperature patterns generated by the two controllers along with supply water

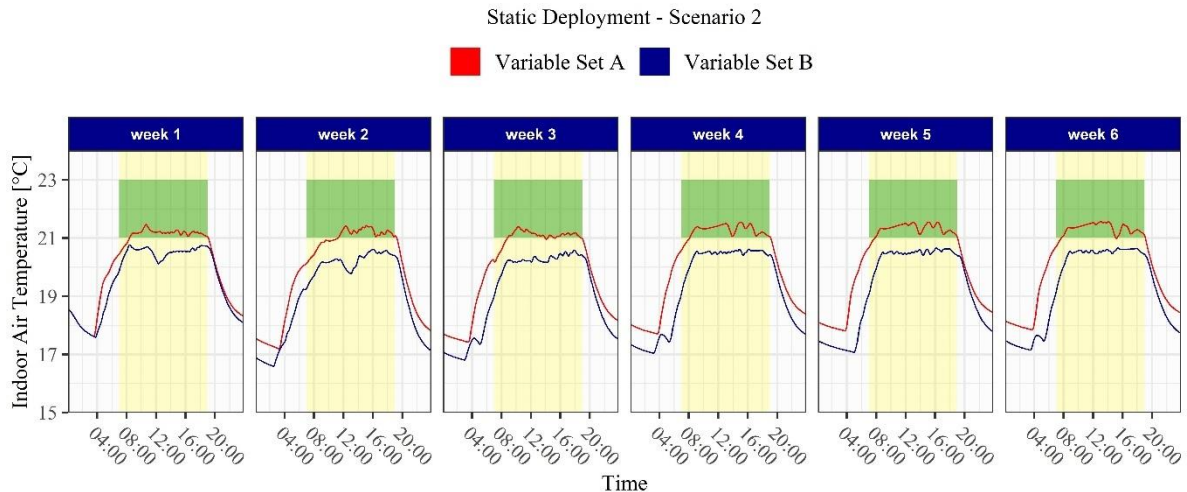
841 temperature, outdoor air temperature and direct solar radiation profiles. The DRL agent was
 842 able to exploit solar heat gains reducing supply water temperature and, consequently, save
 843 energy. This aspect is particularly relevant during the third and sixth day when solar radiation
 844 is higher.



845

846 Figure 12 - Comparison between statically deployed agent trained with *variable set A* and baseline
 847 controller during a week of the deployment period.

848 Figure 13 highlights a comparison between agent trained with *variable set A* (red lines)
 849 and agent trained *variable set B* (blue lines). The plot shows for different weeks and the same
 850 working day (Tuesday), the daily indoor temperature profiles in the scenario S2, which
 851 implements an increased indoor setpoint (22 °C) compared to the training phase (21 °C). As
 852 can be observed the agent based on adaptive variables (*variable set A*) was promptly able to
 853 adapt to the change of indoor temperature requirements maintaining satisfying conditions
 854 within the zone despite any learning goes on during static deployment. On the other hand,
 855 the agent trained with non-adaptive variables (*variable set B*) was not capable to adapt
 856 without relying on dynamic deployment.

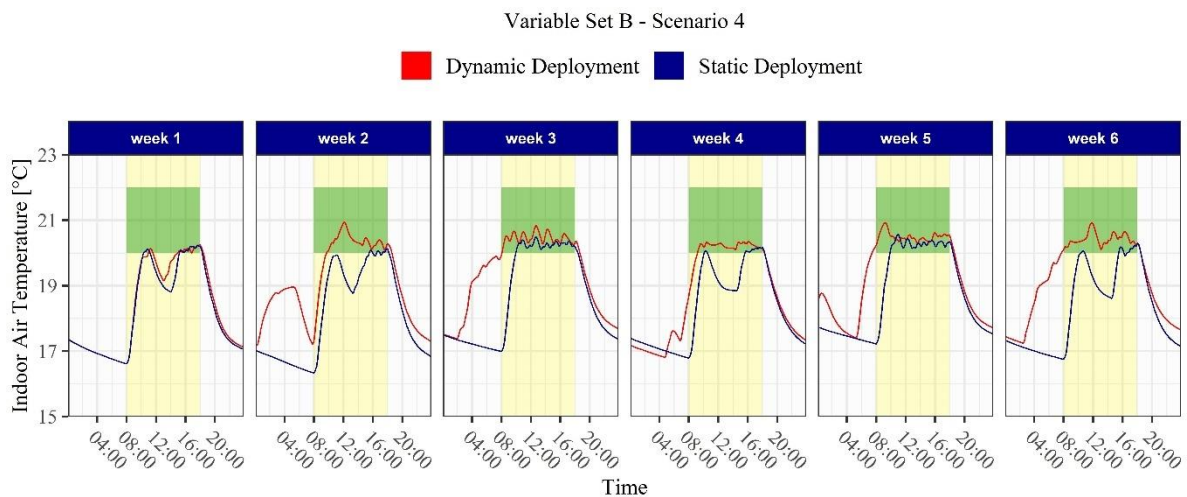


857

858 Figure 13 - Comparison between statically deployed agents trained with *variable set A* and *variable set*
 859 *B* in terms of daily indoor temperature profiles during Tuesdays in the scenario S2.

860 A similar comparison is presented in Figure 14 between static and dynamic deployment
 861 for the agent trained with variables selected according to the non-adaptive approach (*variable*
 862 *set B*). The figure shows the results obtained during the first 6 Sundays in deployment
 863 scenario S4. This scenario is particularly interesting because, differently from the training
 864 conditions, implements the presence of occupants during Sundays. The plot shows, for the
 865 first 6 weeks, the daily indoor temperature profiles generated by the two agents. It is
 866 interesting to notice that the divergence between the profiles increases over time suggesting
 867 that the two agents have different adaptability capabilities. During the first week the two
 868 agents generated almost the same pattern which clearly do not satisfy the indoor temperature
 869 requirements. The larger temperature violation is localized during the first hours of the day
 870 since both the agents were not able to anticipate occupants' arrival. A second temperature
 871 violation region is localized in the middle part of the day, when, during training, the agent
 872 correctly learnt to exploit solar heat gains in order to reduce supply water temperature.
 873 However, the reduction of supply water temperature caused the occurrence of temperature
 874 violation condition since the agent did not performed a sufficient pre-heating of the zone in
 875 order to reach the acceptability range of the indoor temperature. This pattern was replicated
 876 by the statically deployed agent among the six weeks demonstrating its lack in adapting to

877 the modified occupancy schedule. On the contrary, the dynamically deployed agent was
 878 capable to learn from experience and it was able to achieve satisfying temperature conditions
 879 starting from the third week of deployment.



880

881 Figure 14 – Comparison between dynamically and statically deployed agent trained with *variable set*
 882 *B* in terms of daily indoor temperature profiles during Sundays in scenario S4.

883 8. Discussion

884 The present paper focuses on the development of a DRL controller of supply water
 885 temperature setpoint to terminal units of a heating system. The developed controller was
 886 trained and deployed in a simulation environment which combines EnergyPlus and Python.
 887 The controller aims at optimising both energy consumption and indoor temperature control
 888 trying to identify the best trade-off between the two contrasting functions. The control
 889 problem analysed in this work was relatively simple, not involving elements such as
 890 renewable energy sources or storages which may effectively require an optimised controller
 891 to be fully exploited. Although the only two features of the building that could be exploited
 892 in the considered optimisation process were the building thermal mass and the temperature
 893 acceptability range, the DRL controller led to good performance improvements in
 894 comparison to the baseline controller.

895 In DRL algorithms hyperparameters tuning and reward design play a key role in
 896 identifying the optimal configuration of DRL controller. In this work, a sensitivity analysis

897 was carried out on some of the main hyperparameters to highlight their influence on the final
898 performance of the developed controller. Given this strong dependence it seems necessary
899 for reinforcement learning applications in HVAC systems to rely on simulated environments,
900 at least in the initial stage of training. As a consequence, despite the model-free nature of
901 reinforcement learning control, a modelling effort needs to be accounted.

902 The effect of adaptive variables defining the state-space was analysed. A variable set
903 designed to enhance adaptability and flexibility of a DRL agent with respect to variable
904 requirements of the indoor environment (i.e. indoor temperature setpoint and occupancy
905 schedule) was introduced. A DRL agent based on adaptive variables was compared with an
906 agent trained with more classic non-adaptive variables. The comparison was performed by
907 simulating the deployment of the two agents in four different scenarios. Moreover, the
908 agents' deployment was simulated both in static and dynamic configuration. The agent
909 trained with adaptive variable set was capable to adapt to each scenario performing better
910 than the baseline controller even if statically deployed. The dynamic deployment of the same
911 agent did not produce significant improvements on the overall performance, showing slight
912 poorer performance compared to static deployment case.

913 On the contrary when the variables were selected with a non-adaptive approach the
914 dynamic deployment performed better than the static deployment in all the scenarios
915 analysed. These results proved that the proposed variable selection process was useful in
916 providing to the agent the capability to adapt itself to changes that may occur in the controlled
917 environment. This analysis suggests that a DRL controller with a carefully designed state-
918 space is capable to provide the necessary flexibility and adaptability to changing indoor
919 requirements even in a static deployment configuration. Through this approach is possible to
920 leverage the advantages provided by static deployment (i.e. lower computational costs and
921 higher stability) without sacrificing adaptability. However, the adoption of an adaptive
922 approach in the design of the state space may not be enough to guarantee a good control

923 performance in the case of retrofit on the HVAC system or other building components. In
924 such cases thermal dynamics of the controlled environment may change requiring DRL
925 controller to update its policy through a dynamic deployment.

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

945 **9. Conclusions and future works**

946 In the present paper, the application of DRL control in a water-based heating system was
947 developed and analysed in a simulation environment. The flexibility and adaptability of the

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

961 Future works will be focused on the following aspects:

- 962 • Exploring the capabilities of Multi-Agent Reinforcement Learning (MARL) framework.
963 The present work focuses on only one zone of an office building characterized by a
964 complex HVAC system. MARL could provide a solution to coordinate multiple
965 actuators that are present in an HVAC system in order to reach a global optimum
966 solution.
- 967 • Comparing the performance of DRL with model-based control solution such as MPC.
968 Due to its model-free formulation, Reinforcement Learning is diametrically opposed to
969 Model Predictive Control. A robust comparison between these two techniques in terms
970 of control performance, computational cost and modelling effort could provide useful
971 insights on the strength and weakness of DRL controllers.
- 972 • Applying DRL to novel HVAC systems. Given the ability of DRL to handle multi-
973 objective function, HVAC systems characterized by higher level of complexity (e.g.

974 including RES generation and storage) could provide an excellent testbed to prove the
975 effectiveness of DRL control over classical control methods.

976 • Introducing comfort parameters in the objective function. Even if the monitoring of
977 many comfort parameters (e.g. air velocity, mean radiant temperature) is a non-trivial
978 task in real world applications, in a simulative context the evaluation of thermal comfort
979 performance achieved by a DRL agent could be explored in future works.

980 • Implementing the developed controller in a real-world testbed. Moving from simulation
981 to real-world implementation is extremely complicated and present some major
982 challenges related to the required infrastructure to effectively deploy the controller.
983 Future works will be focused on investigating these aspects and on the evaluation of the
984 performance of DRL control agent once deployed in-field.

985 • Exploring furtherly the paradigm of dynamic deployment of DRL agents. Despite the
986 disadvantage of possible instabilities in the learned control policy, dynamic deployment
987 might be necessary to obtain a fully-flexible agent which is capable to adapt even when
988 the thermal dynamics of the controlled environment changes (e.g. retrofit intervention).
989 In the future works dynamic deployment will be analysed in order to enhance its
990 robustness and stability.

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

993

994 **Acknowledgements**

995 The work of Silvio Brandi is supported by Enerbrain s.r.l. in the context of a Ph.D.
996 scholarship at Politecnico di Torino.

997 **References**

- 998 [1] Capozzoli A, Mechri HE, Corrado V. Impacts of architectural design choices on building energy
999 performance applications of uncertainty and sensitivity techniques 2009;15217:1000–7.
- 1000 [2] Yu Z, Fung BCM, Haghghat F. Extracting knowledge from building-related data - A data mining
1001 framework. *Build Simul* 2013;6:207–22. <https://doi.org/10.1007/s12273-013-0117-8>.
- 1002 [3] Fan C, Xiao F, Yan C. A framework for knowledge discovery in massive building automation data and
1003 its application in building diagnostics. *Autom Constr* 2015;50:81–90.
1004 <https://doi.org/10.1016/j.autcon.2014.12.006>.
- 1005 [4] Capozzoli A, Piscitelli MS, Brandi S. Mining typical load profiles in buildings to support energy
1006 management in the smart city context. *Energy Procedia* 2017;134:865–74.
1007 <https://doi.org/10.1016/j.egypro.2017.09.545>.
- 1008 [5] Piscitelli MS, Brandi S, Capozzoli A. Recognition and classification of typical load profiles in buildings
1009 with non-intrusive learning approach. *Appl Energy* 2019;255.
1010 <https://doi.org/10.1016/j.apenergy.2019.113727>.
- 1011 [6] Kramer H, Lin G, Granderson J, Curtin C, Crowe E. Synthesis of Year One Outcomes in the Smart
1012 Energy Analytics Campaign Building Technology and Urban Systems Division 2017.
- 1013 [7] Molina-Solana M, Ros M, Ruiz MD, Gómez-Romero J, Martín-Bautista MJ. Data science for building
1014 energy management: A review. *Renew Sustain Energy Rev* 2017;70:598–609.
1015 <https://doi.org/10.1016/j.rser.2016.11.132>.
- 1016 [8] Capozzoli A, Piscitelli MS, Brandi S, Grassi D, Chicco G. Automated load pattern learning and anomaly
1017 detection for enhancing energy management in smart buildings. *Energy* 2018;157.
1018 <https://doi.org/10.1016/j.energy.2018.05.127>.
- 1019 [9] Martinopoulos G, Papakostas KT, Papadopoulos AM. A comparative review of heating systems in EU
1020 countries, based on efficiency and fuel cost. *Renew Sustain Energy Rev* 2018;90:687–99.
1021 <https://doi.org/10.1016/j.rser.2018.03.060>.
- 1022 [10] Finck C, Beagon P, Clauss J, Thibault P, Vogler-Finck PJC, Zhang K, et al. Review of applied and
1023 tested control possibilities for energy flexibility in buildings: a technical report from IEA EBC Annex
1024 67 Energy Flexible Buildings 2017:1–59. <https://doi.org/10.13140/RG.2.2.28740.73609>.
- 1025 [11] Clauß J, Finck C, Vogler-finck P, Beagon P. Control strategies for building energy systems to unlock
1026 demand side flexibility – A review Norwegian University of Science and Technology , Trondheim ,
1027 Norway Eindhoven University of Technology , Eindhoven , Netherlands Neogrid Technologies ApS /
1028 Aalborg. 15th Int Conf Int Build Perform 2017:611–20.
- 1029 [12] Afram A, Janabi-Sharifi F, Fung AS, Raahemifar K. Artificial neural network (ANN) based model
1030 predictive control (MPC) and optimization of HVAC systems: A state of the art review and case study
1031 of a residential HVAC system. *Energy Build* 2017;141:96–113.
1032 <https://doi.org/10.1016/j.enbuild.2017.02.012>.

- 1033 [13] Salsbury TI. A survey of control technologies in the building automation industry. vol. 16. IFAC; 2005.
1034 <https://doi.org/10.3182/20050703-6-cz-1902.01397>.
- 1035 [14] Behrooz F, Mariun N, Marhaban MH, Radzi MAM, Ramli AR. Review of control techniques for HVAC
1036 systems-nonlinearity approaches based on fuzzy cognitive maps. *Energies* 2018;11.
1037 <https://doi.org/10.3390/en11030495>.
- 1038 [15] Afram A, Janabi-Sharifi F. Theory and applications of HVAC control systems - A review of model
1039 predictive control (MPC). *Build Environ* 2014;72:343–55.
1040 <https://doi.org/10.1016/j.buildenv.2013.11.016>.
- 1041 [16] Subbaram Naidu D, Rieger CG. Advanced control strategies for heating, ventilation, air-conditioning,
1042 and refrigeration systems - An overview: Part I: Hard control. *HVAC R Res* 2011;17:2–21.
1043 <https://doi.org/10.1080/10789669.2011.540942>.
- 1044 [17] Serale G, Fiorentini M, Capozzoli A, Bernardini D, Bemporad A. Model Predictive Control (MPC) for
1045 enhancing building and HVAC system energy efficiency: Problem formulation, applications and
1046 opportunities. *Energies* 2018;11. <https://doi.org/10.3390/en11030631>.
- 1047 [18] Subbaram Naidu D, Rieger CG. Advanced control strategies for HVAC&R systems - An overview:
1048 Part II: Soft and fusion control. *HVAC R Res* 2011;17:144–58.
1049 <https://doi.org/10.1080/10789669.2011.555650>.
- 1050 [19] Fiorentini M, Cooper P, Ma Z, Robinson DA. Hybrid model predictive control of a residential HVAC
1051 system with PVT energy generation and PCM thermal storage. *Energy Procedia* 2015;83:21–30.
1052 <https://doi.org/10.1016/j.egypro.2015.12.192>.
- 1053 [20] Halvgaard R, Poulsen NK, Madsen H, Jørgensen JB. Economic Model Predictive Control for building
1054 climate control in a Smart Grid. 2012 IEEE PES Innov Smart Grid Technol ISGT 2012 2012:1–6.
1055 <https://doi.org/10.1109/ISGT.2012.6175631>.
- 1056 [21] Mady AED, Provan GM, Ryan C, Brown KN. Stochastic model predictive controller for the integration
1057 of building use and temperature regulation. *Proc Natl Conf Artif Intell* 2011;2:1371–6.
- 1058 [22] Prívvara S, Váňa Z, Gyalistras D, Cigler J, Sagerschnig C, Morari M, et al. Modeling and identification
1059 of a large multi-zone office building. *Proc IEEE Int Conf Control Appl* 2011:55–60.
1060 <https://doi.org/10.1109/CCA.2011.6044402>.
- 1061 [23] Prívvara S, Cigler J, Váňa Z, Oldewurtel F, Sagerschnig C, Žáčková E. Building modeling as a crucial
1062 part for building predictive control. *Energy Build* 2013;56:8–22.
1063 <https://doi.org/10.1016/j.enbuild.2012.10.024>.
- 1064 [24] Lympelopoulous G, Ioannou P. Energy & Buildings Building temperature regulation in a multi-zone
1065 HVAC system using distributed adaptive control R. *Energy Build* 2020;215:109825.
1066 <https://doi.org/10.1016/j.enbuild.2020.109825>.
- 1067 [25] Buonomano A, Montanaro U, Palombo A, Santini S. Temperature and humidity adaptive control in

- 1068 multi-enclosed thermal zones under unexpected external disturbances. *Energy Build* 2017;135:263–85.
1069 <https://doi.org/10.1016/j.enbuild.2016.11.015>.
- 1070 [26] Baldi S, Zhang F, Le T, Endel P, Holub O. Passive versus active learning in operation and adaptive
1071 maintenance of Heating , Ventilation , and Air Conditioning. *Appl Energy* 2019;252:113478.
1072 <https://doi.org/10.1016/j.apenergy.2019.113478>.
- 1073 [27] Barto AG, Sutton RS. Reinforcement Learning: An Introduction. *Kybernetes* 1998;27:1093–6.
- 1074 [28] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, et al. Human-level control
1075 through deep reinforcement learning. *Nature* 2015;518:529–33. <https://doi.org/10.1038/nature14236>.
- 1076 [29] Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, et al. Mastering the game of
1077 Go with deep neural networks and tree search. *Nature* 2016;529:484–9.
1078 <https://doi.org/10.1038/nature16961>.
- 1079 [30] Zhang Z, Chong A, Pan Y, Zhang C, Lam KP. Whole building energy model for HVAC optimal control:
1080 A practical framework based on deep reinforcement learning. *Energy Build* 2019;199:472–90.
1081 <https://doi.org/10.1016/j.enbuild.2019.07.029>.
- 1082 [31] Wei T, Wang Y, Zhu Q. Deep Reinforcement Learning for Building HVAC Control. *Proc - Des Autom*
1083 *Conf* 2017;Part 12828. <https://doi.org/10.1145/3061639.3062224>.
- 1084 [32] Valladares W, Galindo M, Gutiérrez J, Wu WC, Liao KK, Liao JC, et al. Energy optimization associated
1085 with thermal comfort and indoor air control via a deep reinforcement learning algorithm. *Build Environ*
1086 2019;155:105–17. <https://doi.org/10.1016/j.buildenv.2019.03.038>.
- 1087 [33] Wang Y, Velswamy K, Huang B. A long-short term memory recurrent neural network based
1088 reinforcement learning controller for office heating ventilation and air conditioning systems. *Processes*
1089 2017;5. <https://doi.org/10.3390/pr5030046>.
- 1090 [34] Gao G, Li J, Wen Y. Energy-Efficient Thermal Comfort Control in Smart Buildings via Deep
1091 Reinforcement Learning 2019:1–11.
- 1092 [35] Fazenda P, Veeramachaneni K, Lima P, O'Reilly UM. Using reinforcement learning to optimize
1093 occupant comfort and energy usage in HVAC systems. *J Ambient Intell Smart Environ* 2014;6:675–90.
1094 <https://doi.org/10.3233/AIS-140288>.
- 1095 [36] Zou Z, Yu X, Ergan S. Towards optimal control of air handling units using deep reinforcement learning
1096 and recurrent neural network. *Build Environ* 2020;168:106535.
1097 <https://doi.org/10.1016/j.buildenv.2019.106535>.
- 1098 [37] Vázquez-Canteli JR, Ulyanin S, Kämpf J, Nagy Z. Fusing TensorFlow with building energy simulation
1099 for intelligent energy management in smart cities. *Sustain Cities Soc* 2019;45:243–57.
1100 <https://doi.org/10.1016/j.scs.2018.11.021>.
- 1101 [38] Vázquez-Canteli J, Kämpf J, Nagy Z. Balancing comfort and energy consumption of a heat pump using

- 1102 batch reinforcement learning with fitted Q-iteration. *Energy Procedia* 2017;122:415–20.
1103 <https://doi.org/10.1016/j.egypro.2017.07.429>.
- 1104 [39] Zhang Z, Chong A, Pan Y, Zhang C, Lu S, Lan KP. A Deep Reinforcement Learning Approach to
1105 Using Whole Building Energy Model for HVAC Optimal Control. *2018 Build Perform Model Conf*
1106 2018:675–82.
- 1107 [40] Vázquez-Canteli JR, Nagy Z. Reinforcement learning for demand response: A review of algorithms
1108 and modeling techniques. *Appl Energy* 2019;235:1072–89.
1109 <https://doi.org/10.1016/j.apenergy.2018.11.002>.
- 1110 [41] Han M, May R, Zhang X, Wang X, Pan S, Yan D, et al. A review of reinforcement learning
1111 methodologies for controlling occupant comfort in buildings. *Sustain Cities Soc* 2019;51:101748.
1112 <https://doi.org/10.1016/j.scs.2019.101748>.
- 1113 [42] Costanzo GT, Iacovella S, Ruelens F, Leurs T, Claessens BJ. Sustainable Energy , Grids and Networks
1114 Experimental analysis of data-driven control for a building heating system. *Sustain Energy, Grids*
1115 *Networks* 2016;6:81–90. <https://doi.org/10.1016/j.segan.2016.02.002>.
- 1116 [43] Crawley DB, Pedersen CO, Lawrie LK, Winkelmann FC. Energy plus: Energy simulation program.
1117 *ASHRAE J* 2000;42:49–56.
- 1118 [44] Wiering MA. Explorations in efficient reinforcement learning. PhD Thesis, Univ Amsterdam 1999.
- 1119 [45] Van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double Q-Learning. *30th AAAI*
1120 *Conf Artif Intell AAAI 2016* 2016:2094–100.
- 1121 [46] Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J, et al. *OpenAI Gym* 2016:1–4.
- 1122 [47] Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al. *TensorFlow: Large-Scale Machine*
1123 *Learning on Heterogeneous Distributed Systems* 2016.
- 1124 [48] Chollet F, others. *Keras* 2015.
- 1125 [49] Dulac-Arnold G, Evans R, van Hasselt H, Sunehag P, Lillicrap T, Hunt J, et al. *Deep Reinforcement*
1126 *Learning in Large Discrete Action Spaces* 2015.
- 1127 [50] Hinton G, Tieleman T. Rmsprop: Divide the gradient by a running average of its recent magnitude.
1128 coursera: *Neural networks for machine learning*. Tech Rep, Tech Rep 2012:31.
- 1129 [51] Lonza A. *Reinforcement Learning Algorithms with Python* 2019.
- 1130