Expert Systems With Applications 201 (2022) 116995

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Power output optimization of electric vehicles smart charging hubs using deep reinforcement learning

Andrea Bertolini, Miguel S.E. Martins *, Susana M. Vieira, João M.C. Sousa

IDMEC, Instituto Superior Técnico, Universidade de Lisboa, 1 Av. Rovisco Pais, 1049-001 Lisbon, Portugal

ARTICLE INFO

Keywords:
Reinforcement learning
Electric vehicles
Real-time charging scheduling
Neural network
Clustering algorithm

ABSTRACT

Since most branches of the distribution grid may already be close to their maximum capacity, smart management when charging electric vehicles (EVs) is becoming more and more crucial. In fact, office buildings might not be able to handle several transactions at the same time, especially considering the next generation of fast chargers which are very power expensive. Thus, an efficient charging policy needs to be found. This paper proposes the scheduling of real-time EVs charging through deep reinforcement learning (DRL) techniques. DRL has been chosen because it can adaptively learn from interacting with the surrounding environment. The focus of the optimization is to ensure the completion of the charging transactions in a timely manner, while shifting the load from the times of peak demand. The novelty of the proposed approach lies in its innovative framework: pools of electric vehicles with different characteristics are categorized using a clustering algorithm, a tree-based classifier has been developed to sort new instances of EVs, and a multilayer perceptron artificial deep neural network has been trained to predict the expected duration of each charging session. These features are used as inputs to the DRL agent, and are mapped into actions that adjust the maximum power associated to each charging station. The model has been compared to a traditional charging algorithm and increasingly challenging scenarios have been considered. Results have shown that the developed algorithm fails less than the baseline, with a reduction of the load due to EVs charging of 80% during peak times.

1. Introduction

1.1. Motivation

The transportation sector is one of the major responsible for energy consumption in the world. This environmental issue, accompanied by governments' incentives, have pushed for a strong adoption of electric vehicles (EVs) in many countries, which have received considerable attention in recent years due to their low operating cost and potential for energy sustainability (Quddus, Yavuz, Usher, & Marufuzzaman, 2019). In fact, the electrification of the transportation sector, together with the increasing electricity generation from renewable energy sources (RES) can lower the reliance on fossil fuels and lead to emission reduction (Nour, Chaves-Ávila, Magdy, & Sánchez-Miralles, 2020). Nevertheless, it has to be mentioned that EVs scale adoption will bring a massive high-power load into the electric grid, that might represent a threat to the health of the current energy system. Therefore, if multiple EVs were charged simultaneously in an uncontrolled way, they could increase the peak demand on the grid, contributing to overloading and to the need for upgrades at the distribution level (IRENA, 2019). This is the case of *traditional charging*, where charging stations possess no means of intercommunicating with other IT devices and power is immediately delivered at maximum speed rate. Instead, the technology behind smart charging stations is able to adjust the power output, optimizing the charging process while ensuring grid stability. *Smart charging* or intelligent charging formally refers to a system where an electric vehicle shares a data connection with a charging device, and the charging device shares a data connection with a charging operator (Virta, 2020). In other words, it is a way of optimizing the charging process according to distribution grid constraints and local renewable energy availability while respecting customers' needs for vehicle availability (IRENA, 2019).

1.2. State of the art

With the rising share of EVs used in the service industry, the optimization of their specific constraints is gaining importance. Lowering energy consumption, time of charging and the strain on the electric grid are just some of the issues that must be tackled, to

* Corresponding author.
E-mail addresses: andrea.bertolini@tecnico.uslisboa.pt (A. Bertolini), miguelsemartins@tecnico.ulisboa.pt (M.S.E. Martins), susana.vieira@tecnico.ulisboa.pt (S.M. Vieira), jmsousa@tecnico.ulisboa.pt (J.M.C. Sousa).

https://doi.org/10.1016/j.eswa.2022.116995
Received 29 April 2021; Received in revised form 23 March 2022; Accepted 25 March 2022
Available online 4 April 2022

ensure a cleaner and more efficient industry (Karakatič, 2021). In the past years, several techniques have been explored to tackle this problem. Amjad, Ahmad, Rehmani, and Umer (2018) give a complete overview of what has been done already in the field, differentiating by charging approaches, optimization objectives and optimization approaches adopted. Sortomme and El-Sharkawi (2012) applied convex programming to schedule energy and ancillary services with the objective of maximizing the profit of an aggregator and providing low charging costs to the final customers. However, in many cases, such techniques are used to schedule day-ahead the charging of the EVs, therefore requiring a second algorithm to effectively deliver power to the vehicles. Wu, Zeng, Lu, and Boulet (2017) addressed the challenge of energy scheduling in office buildings integrated with photovoltaic systems and workplace EV charging. Electric vehicles can also be optimally charged using dynamic programming, where the problem is divided into multiple sub-problems, which solutions are stored in the memory after being solved. In their paper, Škugor and Deur (2014) separately optimize first the charging of each individual EV of the fleet, thus providing globally optimal solution on the vehicle level. Moreover, Jin, Xu, and Yang (2020) studied the joint scheduling of battery energy storage system operation in the presence of random renewable generation, EV arrivals, and electricity prices by formulating a cost-minimizing scheduling problem as a dynamic program. Another family of techniques potentially suitable for this problem is meta-heuristics. To this end, Elmehdi and Abdelilah (2019) scheduled the charge of an EVs fleet with genetic algorithms, while Zheng and Yang (2020) introduced a global intelligent method to find optimal cooperation charging/discharging strategies for EVs to minimize the operation cost with particle swarm optimization (PSO). Although the aforementioned methods achieved some success in day-ahead charging/discharging scheduling, they may be unsuitable for real-time scenarios where the variations in EV charging demand and the electricity prices are much more complex. Thus, a different approach is given by *Deep Reinforcement Learning*. While it is usually needed to precisely formulate the optimization problem (in its objective function and constraints, for instance), in DRL the intelligent agent can act *model-free*. In this paper, DRL has been used to dynamically optimize the charging scheduling of a fleet of electric vehicles. The main contributions of this work reside in the problem design by combining multiple algorithms in one single framework, using the following techniques: clustering EVs charging patterns, predicting EVs' charging time and simulation-based optimization, as well as proposing a novel reward mechanism. Recent experiments have been found in literature, showing a growing interest in the subject. However, these differentiate from the proposed work in terms of scope, data availability or utilized methods. Sallam, Chakrabortty, and Ryan (2021) and Silva, Souza, Souza, and Bazzan (2019) used reinforcement learning frameworks to address scheduling problems in areas different from the energy sector. Mocanu et al. (2019) explored for the first time in the smart grid context the benefits of using DRL to perform on-line optimization of schedules for building energy management systems. Similarly, Wei, Wang, and Zhu (2017) proposed a system DRL-based algorithm for building HVAC (Heating, Ventilation and Air-Conditioning) control. Wan, Li, He, and Prokhorov (2019) deal with the optimal EV charging problem with DRL, with the objective to find cost-efficient charging/discharging schedules to take full advantage of the real-time electricity price while fulfilling user's driving demand. Sadeghianpourhamami, Deleu, and Develder (2018) used a model-free approach with RL to achieve coordinated and scalable EV charging with demand-response. Zhang, Liu, Wu, Tang, and Fan (2021) formalize the scheduling problem of EV charging as a Markov Decision Process (MDPs) and propose DRL algorithms to address it, with the objective to minimize the total charging time of EVs. Similarly, Li, Wan, and He (2020) found a constrained charging/discharging scheduling strategy to minimize the charging cost as well as to guarantee that the EV can be fully charged. The authors formulate the problem as a constrained MDP, without requiring any domain knowledge about the randomness and without manually design a penalty term or tune a penalty coefficient. Wang, Bi, and Zhang (2018) propose a profit-maximizing joint charging scheduling and pricing scheme for a public charging station with a RL approach. Lee, Lee, and Kim (2020) analyzed the effectiveness of a RL-based, real-time EV charging and discharging algorithm from the perspectives of charging cost and load shifting effect. Ding et al. (2020) propose an optimal EV charging strategy in a distribution network to maximize the profit of the grid operators while satisfying all the physical constraints, utilizing RL to analyze the impact of uncertainties on the charging strategy. Zhang, Yang, and An (2021) formulate the EV charging control as a MDP and propose a novel RL approach to learn the optimal charging control strategy for satisfying the user's requirement of battery energy while minimizing the user's charging expense. Dang, Wu, and Boulet (2019) applied a Q-Learning based charging scheduling scheme for EVs, considering bidirectional interaction between the vehicle and the grid, including the grid-to-vehicle charging and the *vehicle-to-grid* (V2G) electricity returning. Shi and Wong (2011) also modeled a V2G control algorithm, with the extra action of providing frequency regulation services to the grid. Lee and Choi (2020) presented a method for the scheduling of energy consumption of smart home appliances and distributed energy resources, including the charging of an electric vehicle. Jin and Xu (2021) propose a model-free soft-actor–critic (SAC) based method to address the scheduling of large-scale EV charging in a power distribution network under random renewable generation and electricity prices. Zishan, Haji, and Ardakanian (2020) used multi-agent reinforcement learning to determine the optimal control of EVs charging to avoid grid congestion and transformer overloading. Lastly, Fang et al. (2019) also developed a multi-agent RL approach for optimally scheduling energy in a residential microgrid environment, integrated with EVs and renewable generation.

1.3. Problem formulation

The objective of this work is to develop an innovative decision-based optimization algorithm able to send power limit adjustments to the charging points where EVs can connect. The aim is to alter in real-time the charging loads in order to satisfy power grid and vehicle constraints. The energy system taken in consideration is a generic office building, equipped with a photovoltaic (PV) plant for solar power production and with a garage where the charging stations are installed. The building considered is connected to the grid with 250 kW of power supply, it has up to 74 kWp of solar power installed and it is provided with 10 charging stations where EVs connect and disconnect during the whole day (Fig. 1).

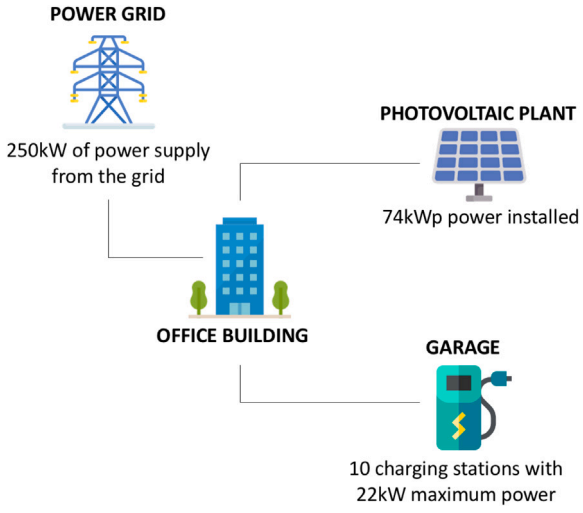

Fig. 1. Global overview of the energy system considered.

The final goal of the optimization is to ensure that every EV successfully performs their charging transactions in a timely manner, by transferring as much energy as possible to each EV without the associated load threatening the grid stability. This last condition has been encoded in the form of a threshold curve, which represents the total power available and whose values the charging system should never exceed. This limit is obtained by subtracting, at each instant of time, the consumption of the building from the grid power supply, and adding up the solar production. Combining all the elements, the problem can be formulated as follows:

$$\begin{aligned} \underset{P_t^i}{\text{maximize}} \quad & \textstyle\sum_{i=1}^{N}(E_T^i - E_0^i) \\ \text{subject to} \quad & \textstyle\sum_{i=1}^{N} P_{t=0}^i \le P_{nom} - P_{cons,t=0} + P_{PV,t=0} \\ & \textstyle\sum_{i=1}^{N} P_{t=1}^i \le P_{nom} - P_{cons,t=1} + P_{PV,t=1} \\ & \vdots \\ & \textstyle\sum_{i=1}^{N} P_{t=T}^i \le P_{nom} - P_{cons,t=T} + P_{PV,t=T} \end{aligned} \tag{1}$$

where:

- $E_T^i - E_0^i$ is the total energy transferred to the vehicle i, where T and 0 correspond respectively to the final and initial times of the charging transaction;
- P_t^i is the charging power rate of the vehicle i at time t;
- P_{nom} is power grid supply from the grid;
- $P_{cons,t}$ is the building consumption at time step t;
- $P_{PV,t}$ is the photovoltaic solar power production at time step t.

Within the scope of this paper, a smart charging algorithm has been proposed using reinforcement learning with a specific focus on *peak shaving*, namely flattening the peak demand by incentivizing charging during times with large penetration of renewables (IRENA, 2019). The objective is to increase self-consumption of locally produced renewable electricity as well as lowering dependence on the electricity grid. It is important to mention that even though this project focuses on building energy dynamics, it can be equally applied to other facilities such as parking lots, curb-side charging and gas station charging hubs. The rest of the paper is organized as follows. *Section* 2 contains a theoretical introduction of the elements of reinforcement learning as well as an overview of the innovative framework proposed. *Section* 3 discusses the simulation dynamics, delving into the techniques used to cluster and classify pools of electric vehicles, as well as to predict the expected duration of their transactions. *Section* 4 discusses the optimization problem, outlining how the intelligent agent and the environment have been designed and how they interact between each other. *Section* 5 shows the results obtained along with the approach that has been followed, comparing the performances of the algorithm in the different scenarios considered and against the baseline. *Section* 6 draws the conclusions.

2. Reinforcement learning

2.1. Theoretical foundations

Reinforcement learning (RL) is a sub-field of machine learning that addresses the problem of the automatic learning of optimal decisions over time (Lapan, 2020). RL algorithms do not learn from pre-processed data, but features and target are generated by the continuous interaction between two main modules, that iteratively exchange information to each other (Fig. 2): the *agent*, which is the learner and decision-maker in the system, and the *environment*, where the agent acts upon. At each time step, the agent receives the current representation of the environment, defined as its internal *state (s)*. Based on the information acquired, it performs an *action (a)*, that might possibly alter the state, converting it to the *next state (s')*. As a consequence of the action taken, the agent receives a *reward (r)*, moves to the *next state* and the scheme is repeated. The mapping from the perceived states to the actions to be taken in those states, namely the way the agent is behaving, is called policy, usually denoted by π.

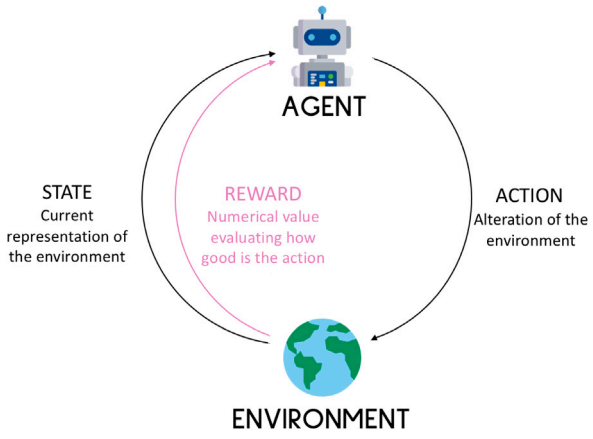

Fig. 2. Agent-environment interaction.

The reward is a numerical signal whose calculation is designed to make it high if the agent takes good actions and low if the agent misbehaves. It is common for reinforcement learning algorithms to estimate *value functions (v(s))*, a metric that assesses *how good* it is for the agent to be in a given state (or to perform a given action in a given state) (Sutton & Barto, 2014-2015). Value functions are interesting and widely used in reinforcement learning tasks because they satisfy particular recursive relationships. The most relevant to the scope of this paper expresses the relation between the value of a state and the values of its successor states (Sutton & Barto, 2014-2015):

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)\left[r + \gamma v_\pi(s')\right] \tag{2}$$

Eq. (2) is known as the *Bellman equation*. The policy (or set of policies) that behaves better than all the other ones is called *optimal policy* and denoted by π_*. The relationship can be now written in a special form:

$$v_*(s) = \max_{a\in A(s)} \sum_{s',r} p(s', r|s, a)\left[r + \gamma v_*(s')\right] \tag{3}$$

which is known as *Bellman optimality equation* for $v_*(s)$. The environment of the reinforcement learning problem described in this work is considered to be a *Markov decision process* (*MDP*), where the future system dynamics from any state have to depend on the current state only (Lapan, 2020). For finite MDPs, the Bellman optimality equation has a unique solution independent of the policy.

2.2. Q-learning algorithm

Q-learning is an *off-policy* method, namely it evaluates a different policy from the one used to generate the data (Sutton & Barto, 2014-2015). In Q-learning, the action-value function associated to each state–action pair is updated at every step, complying with the following rule:

$$\begin{aligned} Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) \\ - Q(S_t, A_t)] \end{aligned} \tag{4}$$

where S_t is the state at time t, A_t is the action performed at time t, $Q(S_t, A_t)$ is action-value function at time t, γ is the discount factor and α is the learning rate. The *learning rate* α determines to what extent newly acquired information overrides old information. In reinforcement learning problems, often happens that states encountered by the agent have never been experienced before. It is then necessary to generalize from previously experienced states to ones that have never been seen (Sutton & Barto, 2014-2015).

Algorithm 1 Q-Learning

Initialize $Q(s;a)\ \forall\ s \in S$; $a \in A(s)$ arbitrarily and $Q(\text{terminal state}) = 0$
for $episode = 1, \ldots, k$ **do**
  Initialize s
  for $step = 1, \ldots, n$ **do**
    Choose a from s using policy derived from Q
    Take action a, observe r, s'
    $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\right]$
  end for
end for

The kind of generalization required is called *function approximation*, a technique for estimating an unknown underlying function using historical or available observations from the domain (Goodfellow, Bengio, & Courville, 2016). The combination of the popular Q-learning algorithm with neural networks (*Deep Q-Learning*, or *DQN*) is known to overestimate action values under certain conditions. A specific adaptation to the DQN algorithm, called *Double Deep Q-Learning*, or *Double DQN*, has been adopted since it has been shown that the resulting algorithm leads to much better performance (Hasselt, Guez, & Silver, 2015). The novelty is that the target is calculated differently, due to a change in the Bellman equation:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma Q'(S_{t+1}, a^*) - Q(S_t, A_t)] \quad (5)$$

Where:

$$a^* = \max_a Q(S_{t+1}, a)$$

The main difference with respect to traditional Q-learning is that the update was originally computed looking at the highest q-value predicted in the next state by the current action (Eq. (4)). On the other hand, in Double DQN, the best action from the next state is selected and used in the prediction performed by the target network.

2.3. Proposed framework

A Double DQN algorithm has been proposed in this paper to address the complex problem of dynamically scheduling electric vehicles' charging. Fig. 3 shows the overall architecture of the artificial intelligence framework that has been designed.

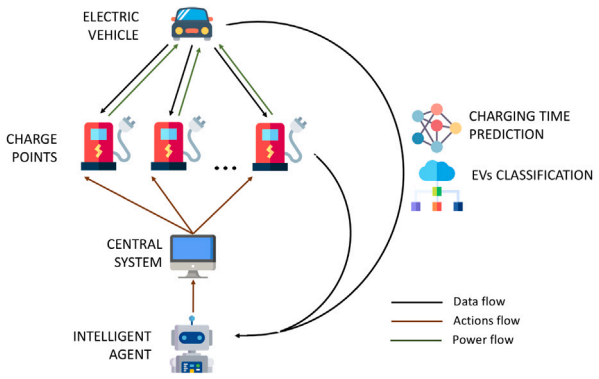

Fig. 3. Architecture of the proposed framework: input data are gained from the EVs (black arrows), the agent sends actions to the charge points (red arrows) that are translated in power adjustments back to the EVs (green arrows).

The key element is the *intelligent agent*, namely a neural network (NN), whose predictions can be thought as power rate variations during the EVs' charging transactions. These outputs are sent to a central system and, in turn, forwarded to the charge points in terms of power limit adjustments. The basic idea is that the agent, at each moment, interacts with the charge points to acquire useful information on the current state of the charging processes. Based on this newly gained knowledge, the agent shapes the charging scheduling of the EVs connected, impacting the information that it will receive in the next step. However, just a part of those data is partially observable. To mitigate this limited information, three subsystems have been devised to complement the existing data. First, a clustering algorithm has been used to identify pools of electric vehicles; then, a tree-based model has been developed to classify new instances of EVs and lastly an additional neural network has been trained to predict the expected duration of the charging session for each new vehicle arriving, see Fig. 4. The terms *clusters* and *classes* will be used throughout the paper to indicate the same group of samples but referring to different tasks. Clusters are the data structures recognized from the initial unlabeled data set, which in turn have been used as classes to sort new instances. Furthermore, a simulated environment has been created in order to iteratively respond to the agent's requests and to evaluate its performance. The interaction between the central system and the charge points has been emulated, as well as the EVs typical arrival and departure scheduling profiles. The work has been structured in a way that the intelligent agent had to fulfill tasks of increasing difficulty. In particular, two different scenarios have been evaluated. Initially, the power threshold is a constant line, taking into account the only grid supply. From this initial step, a more challenging problem has been constructed, considering also building consumption and solar production in the equation, with a power limit variable in time based on real output curves.

3. Charging event generator

3.1. Simulation architecture

The simulation consists of three main objects: the *DQN agent*, in which the neural network is designed and the learning process takes place, the *Central System*, responsible for emulating the behavior of the charging stations and that contains all the environment's characteristics and the *Electric Vehicle*, a separate object in charge of simulating the randomness associated with the EVs arrival/departure and their charging specifications. Those objects are Python classes, which interaction is represented in Fig. 5.

3.2. Arrival and departure scheduling

Data of 120 charging transactions have been considered in order to reproduce the frequency of EVs arriving and leaving from the building's garage. As expected, the majority of the people reach the office in the morning, with a lower and lower number of EVs arriving as the day passes. Naturally, the departure function has the opposite trend. A typical day has been subsequently divided in 12 time bands, in order to calculate the *arrival and departure rates*, namely the probabilities with which a generic electric vehicle arrives or leaves during the day. The rates have been used to build daily occupancy profiles of the building's parking lot, as represented in Fig. 6.

3.3. Electric vehicles clustering

The shape of a charging pattern is an important feature that distinguishes electric vehicles. Although having a similar structure, these trends present differences that might reveal essential information about the type of car under charge at that time. For this reason, a *clustering method* has been used in order to detect pools of cars that have similar charging characteristics. To this end, one of the most popular and well-known algorithms for unsupervised learning tasks has been chosen: *K-Means*. Because of the featureless nature of the available data, the key was to understand which features needed to be designed in order

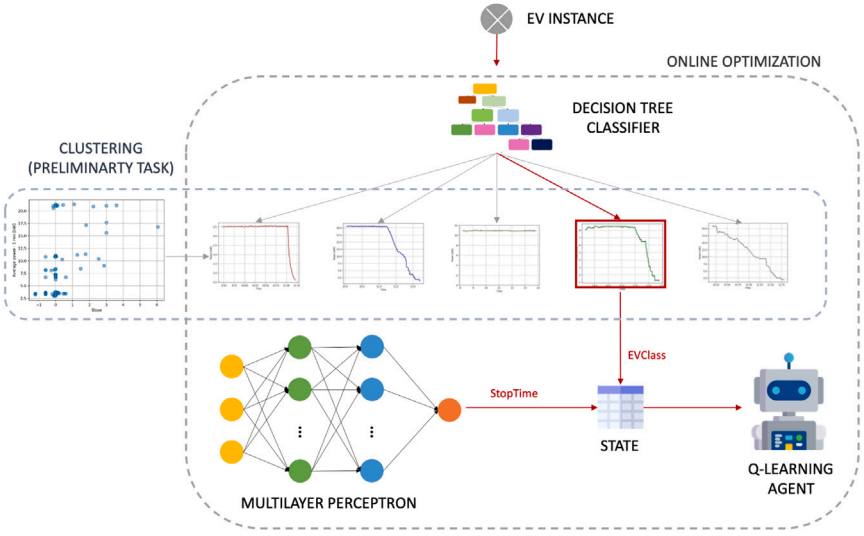

Fig. 4. Flowchart showing the interaction among the different subsystems of the proposed framework.

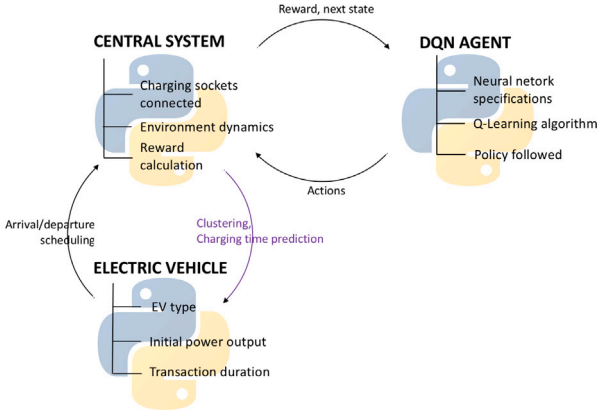

Fig. 5. Interaction among the three main objects of the simulation and their associated attributes.

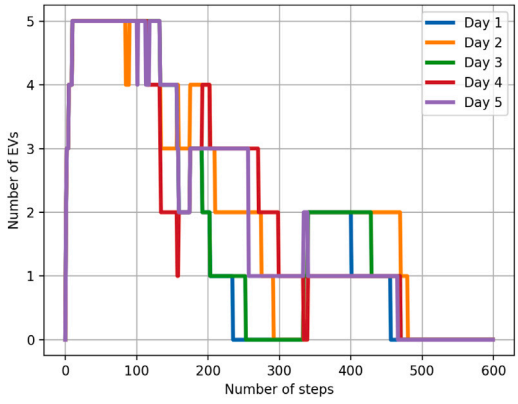

Fig. 6. Parking lot occupancy profiles for five independent episodes/days, each consisting of 600 steps.

for the algorithm to successfully identify clusters among the data. In particular, due to data volatility, the average rate during the first 5 min of charging has been considered:

$$P_{avg} = \frac{\sum_{i=1}^{5} P_i}{5} \tag{6}$$

Where P_i [kW] is the speed charging rate at each time step i. Regarding the configuration of the curve, the slope σ has been created, in the form:

$$\sigma = 3 \times (P_1 - P_{15}) \tag{7}$$

Where P_1 and P_{15} are, respectively, the power rates at time step 1 and 15. A factor of 3, obtained empirically, has also been added to help spreading the data points in space, which contributes to a better performing of K-Means. The results are shown in Fig. 7, where the five clusters can be distinguished. As a new EV connects to a charge point, the car has to be re-conducted to one of the clusters previously introduced. This can be translated in a classification problem when the classes are the clusters themselves. The classes distribution is also *imbalanced*, since some clusters contain significantly more transactions than others. For the purpose of classifying clusters of electric vehicles, a *decision tree* has been trained having available the same data set containing 135 charging transactions that has been used for the clustering task. The features considered for this classification problem are the ones described in Eq. (6) and Eq. (7). The algorithm achieved a value of accuracy on the validation set equal to 94%. However, this result has been improved to 100% accuracy by training many decision trees and computing the aggregated answer of all of them, instead of using the prediction from the individual classifier. The method used in this thesis work is *Random Forests* (Ho, 1995), an ensemble of decision trees trained with the *bagging* method (Géron, 2019).

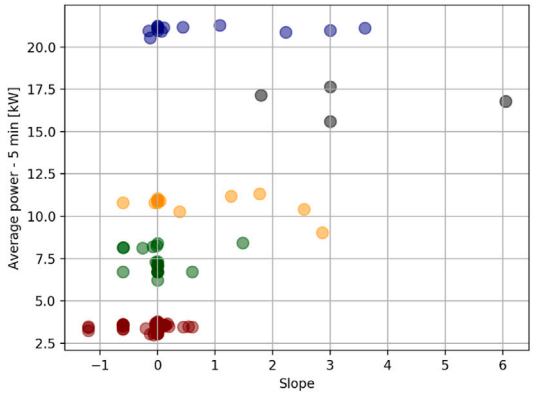

Fig. 7. Detection of the five different clusters.

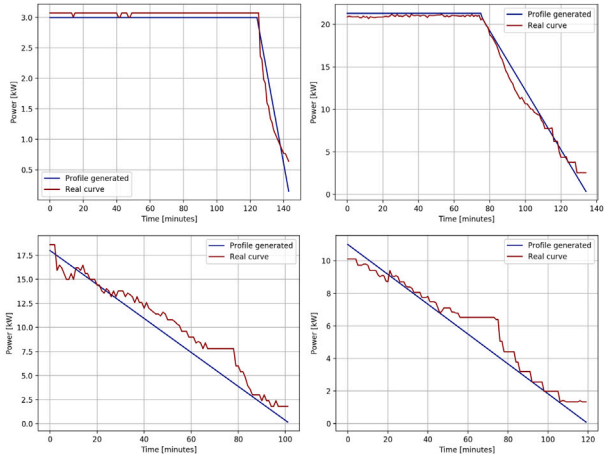

Fig. 8. Profile generated in comparison with the real curves: cluster 1 (upper-left), cluster 3 (upper-right), cluster 4 (lower-left) and cluster 5 (lower-right).

Table 1
Major hyperparameters of the neural network tuned for predicting the charging session time.

Parameter	Value
Number of hidden layers	2
Number of neurons per hidden layer	32
Activation function (hidden layers)	ReLU
Activation function (output layer)	Linear
Loss function	MSE
Optimizer	Adam
Learning rate	0.001
Batch size	16

3.4. Charging session time prediction

The charging pattern of an electric vehicle can be well approximated knowing the type of EV (namely, its cluster), its maximum charging speed admissible and the time needed to complete the transaction. A tailored function has then been created in order to faithfully reproduce the charging pattern of the different EVs (Fig. 8). The purpose of this tool has a key role in the simulation, since it allows to emulate with accuracy the behavior of a high number of EVs, which data would not be available otherwise. A feed-forward neural network has been chosen for predicting the time at which the EVs are expected to be done charging. The data set includes data accounting for 135 charging transactions. The features considered are the *average power rate* during the first 5 min of the transaction (Eq. (6)), the *slope* (Eq. (7)) and the *starting time* of the transaction. The neural network in consideration has been able to reach a mean squared error in the test set equal to 0.02, which translates to 10 min of difference between the target and the value predicted. Table 1 lists the values tuned of the network's most relevant hyperparameters.

4. Real-time electric vehicles charging

4.1. State

The state is the container of information the agent is allowed to visit and from which it extracts knowledge. More formally, the state is represented by a vector that contains all the characteristics of the current environment at a specific instant of time t:

$$s_t = [u_t^1, \dots, u_t^n,\ c_t^1, \dots, c_t^n,\ t_t^1, \dots, t_t^n,\ p_t^1, \dots, p_t^n,$$
$$l_{t+1}, \dots, l_{t+180}]$$

In this work, the state encapsulates five types of information:

- u_t^i is a boolean value indicating whether the charging socket is connected with the EV i or not, informing about the occupancy of the parking lot;
- c_t^i specifies the predicted cluster to which the EV i belongs to;
- t_t^i represents the predicted time needed by the EV i for completing the charging transaction;
- p_t^i is the real-time charging speed rate that the EV i is receiving at that instant of time;
- $l_t \dots, l_{t+180}$ are 3 h ahead forecasted power threshold data, which define the limit of power consumption due to EVs charging.

The algorithm handles the generation of the state and its transition to the next state with two powerful functions: *reset* and *step* functions. Fig. 9 shows schematically how these two functions interact with each other. At the beginning of the episode the reset function is called, and the state vector is created for the first time. The initial situation is an empty parking lot, so the state vector is empty, at least information-wise. This function is just used at the beginning of every episode for this purpose. Later, the agent processes an action to take based on the information, although initially poor, that he gained from the environment. For the first time, the step function is called into question: the state is altered with the new changes caused by the new action performed. In particular, this last procedure is divided in two different *stages*. In the first one, the arrivals and departures of the EVs are controlled and auxiliary algorithms are called into action to determine the missing features of the state. In the second one, the environment receives the agent's action and applies it along with its consequences. As it can be seen from the picture, the step procedure is iterated every time step of the episode, while the reset function will be called again at the beginning of the next episode.

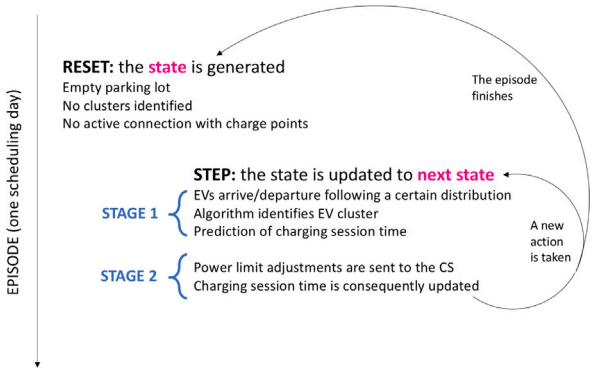

Fig. 9. Transition from a state to the next state through the reset-step functions workflow.

4.2. Actions

The predictions of the neural network translate into the *actions* that the agent will perform in the next time step. The actions have been thought to be power limit adjustments that would intelligently modify the charging scheduling of the electric vehicles. In this regard, an *on/off approach* has been explored, where the agent only has to choose between two actions: *charge* or *do not charge* (Fig. 10). This format is a combination of effectiveness and simplicity, since the update of the environment does not require large computational time and the agent has a variety of just two options to select from. The algorithm needs to send power adjustments to several EVs connected simultaneously, controlling many charge points at the same time. This means that the output of the neural network, namely the action taken by the agent, will be actually translated into as many individual actions as many charging sockets are considered in the problem.

In particular, the total number of possibilities is given by n^k, where n is the number of individual actions and k is the number of charging sockets considered.

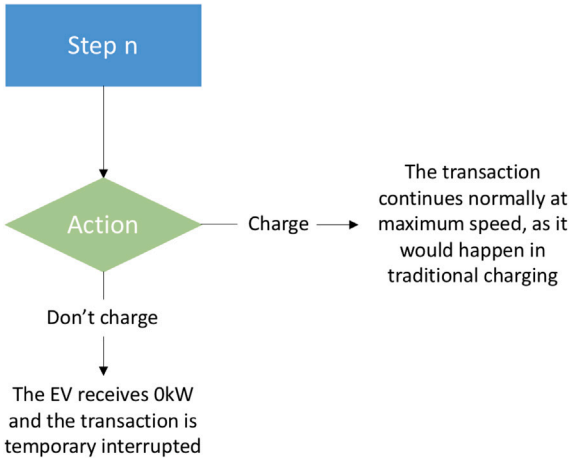

Fig. 10. Decision flowchart for the agent's actions scheme.

4.3. Reward system

The reward is a numerical value that expresses how good was the action taken by the agent. The final reward system has been formulated at the end of a step-by-step process. In fact, the problem has been initially considered in its easiest form, increasing gradually the complexity once good results were obtained. Fig. 11 illustrates the methodology flow that has been followed.

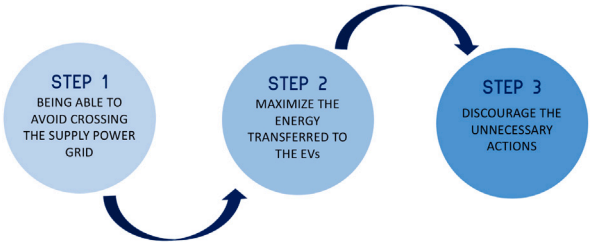

Fig. 11. Reward system shaping methodology.

The goal was to start working with a basic reward implementation whose policy could have quickly been learned by the agent, and then moving to the most detailed and advanced model. The transition between a station and the next one would happen just when the previous model would give robust results. The very first implementation of the model had the easiest task. The only concern here is to make sure that the total charging load, given by the sum of all the EVs simultaneous power consumption, would never exceed the power threshold. As it has been already explained in Section 1.3, the threshold is considered constant and coincides with the power grid in the first scenario, while it depends on external factors such building consumption and photovoltaic solar production in the second scenario, making it variable throughout the day. The formulation of the reward for this first step at each instant of time is given by the following expression:

$$r_t = \begin{cases} -15 & if \quad \sum_{n=1}^{N} P_t^n \geq PT_t, \\ \frac{PT_t - \sum_{n=1}^{N} P_t^n}{100} & otherwise. \end{cases} \tag{8}$$

Where N is the total number of EVs charging, P_t^n is the charging rate of EV n at time t and PT_t is the power threshold at each time step. The strategy that has been pursued was to provide a small positive reward for each time step that the agent successfully performs the task and a large penalty when the agent fails. The *failure* causes also the immediate end of the episode, conveying to the agent the message that those are terminal states and need to be avoided. With the second step, the core of the optimization problem is introduced. On one side, the total charging load has to lie below the power threshold but on the other side, the vehicles need to be as much charged as possible by the time they leave. The agent gets rewarded based on the percentage of energy it allows to transfer during each transaction. The amount of energy transferred corresponds to the area underneath the charging curve, and can be computed with a integral operation. This calculation is performed each time an EV leaves, checking the state of charge at that point. However, often happens that an EV does not stay connected enough to the charge point to even allow the agent to charge it until maximum capacity. For this reason, the reward update falls into two cases:

- *Case 1*: the EV leaves after the time needed to complete the transaction (9);
- *Case 2*: the EV leaves before the time needed to complete the transaction (10).

$$r^n = \begin{cases} -0.5 & if \quad \frac{E^n}{E^n_{max}} < 0.8, \\ \frac{E^n}{E^n_{max}} \times \frac{1}{100} & otherwise. \end{cases} \tag{9}$$

$$r^n = \begin{cases} -0.5 & if \quad \frac{E^n}{E^n_{act,max}} < 0.8, \\ \frac{E^n}{E^n_{act,max}} \times \frac{1}{100} & otherwise. \end{cases} \tag{10}$$

Where E^n is the actual energy transferred to the EV at the time it disconnects from the charging socket, E^n_{max} is the maximum energy transferable to the EV if it would stay there enough time to be completely charged, and $E^n_{act,max}$ is the maximum energy transferable to the EV if it would not stay there enough time to be completely charged. It can be noticed that the update of the reward is at any time step t and for any EV connected n. Moreover, the weights of the two conditions can be compared and it can be seen how the new penalty introduced is 30 times lighter than the punishment for exceeding the power threshold. This is due to the fact that, on average, 30 are the transactions simulated by the environment in one episode during training. The message for the agent is that surpassing the power limit is as serious as transferring to each EV less than 80% of the energy possible. Lastly, the ideal algorithm should interfere as little as possible into the charging dynamics. The continuous receiving of power adjustments might indeed cause problems to the battery management system of the electric vehicle. For this reason, an additional condition has been added to the system in order to encourage the agent to intervene as little as possible. Formally, it can be formulated in the following way:

$$r_t^n = \begin{cases} 0.001 & if \quad a_t^n = a_{t-1}^n, \\ -0.001 & otherwise. \end{cases} \tag{11}$$

Where r_t^n is the reward value given at time t due to the event occurred to the EV n, a_t^n is the action performed by the agent at time step t to the EV n, and a_{t-1}^n is the action performed by the agent at time step $t-1$ to the EV n. The weight associated to this reward condition has been chosen consistently with the importance that has been attributed to it. The diagram in Fig. 12 summarizes the three conditions above explained.

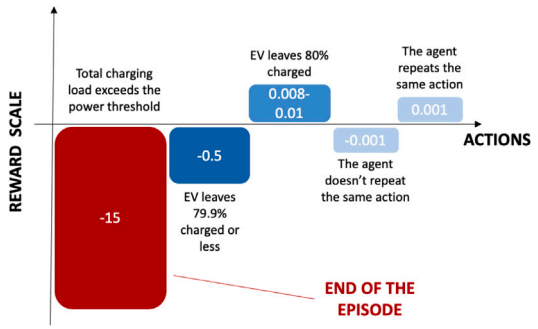

Fig. 12. Representation of the overall reward system with the associated weights for each condition implemented.

4.4. Deep Q-Network

Due to the little time required and its good performance, the agent has been represented by a deep neural network with 4 hidden layers and 32 neurons in each layer. The activation functions associated to the layers are *ReLU* function for the hidden layers and *linear* function for the output layer, due to the regression nature of the problem. The *exploration rate* ϵ is initially set to 1 and it reaches a minimum of 0.01 with a decay rate of 0.995, in order to always have a bit of exploration even when the agent should have learned a very good policy. Experimental results showed that a higher final exploration rate would divert the agent from learning a good policy and maximize the reward. The discount factor γ has been set as 0.99, and the target network is updated every 200 episodes.

5. Experimental results

5.1. Dual approach

The approach followed has been structured gradually, solving the easiest tasks initially and sequentially tackling harder problems (Fig. 13).

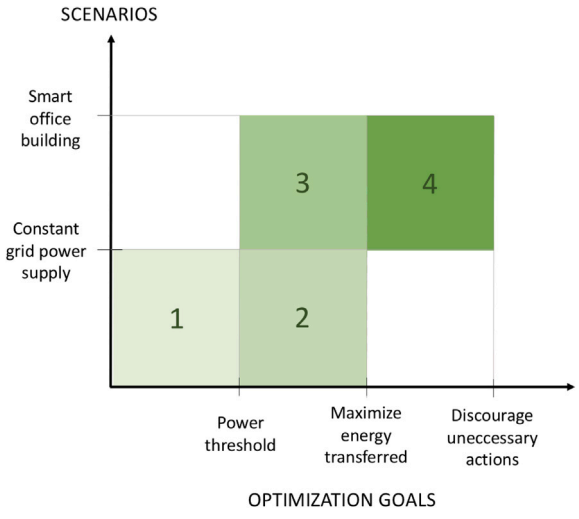

Fig. 13. Scheme of the approach pursued, given by different combinations of scenarios and optimization goals considered.

On one hand, two different *scenarios* have been considered: the purpose was to train the agent in an easier environment before facing more challenging situations. On the other hand, demanding optimization features were progressively added to the problem in order to make it more and more complex. During the course of this section, both scenarios have been analyzed in terms of *charging scheduling* and *policy adopted* and subsequently compared to each other, evaluating the performance of the agent according to specific metrics. Both agents have been compared to a traditional charging algorithm. It has been then possible to see which benefits a smart charging algorithm developed with reinforcement learning can bring and, on the other hand, what are its limitations. The input data have been adjusted in order to deal with a more challenging problem: the grid power supply has been set to 30 kW, while the parking lot has 5 charging sockets to connect EVs. The algorithm has been entirely developed in *Python*. The deep learning part has been supported by the libraries *Tensorflow* and *Keras*, while the agent and environment have been developed from scratch, without using any specific external library as reference. The metrics have been monitored using the tensorflow-based platform *Tensorboard*.

5.2. Scenario 1 — Constant grid power supply

As a very first goal of the optimization, building consumption and solar production are not taken into account. The simulation has been run for a total of 2000 episodes, after which it has not be recognized any valuable improvement and the computational time would rapidly increase. As it can be seen from Fig. 14, in the first scenario the agent find a solid way to improve his reward constantly after each episode.

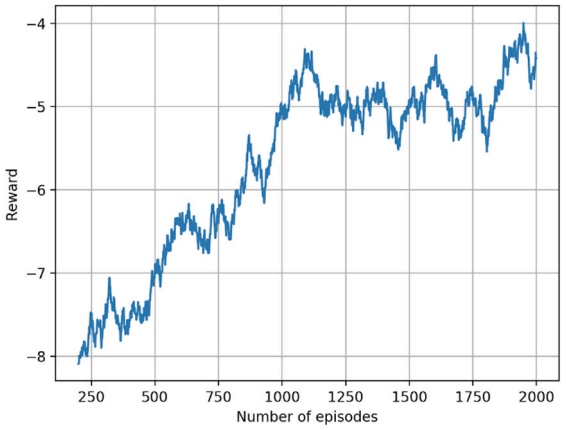

Fig. 14. Reward trend for scenario 1.

Fig. 15 shows the results obtained after testing the agent in a typical day, comparing the scheduling generated by the intelligent agent as opposed to the traditional charging mechanism.

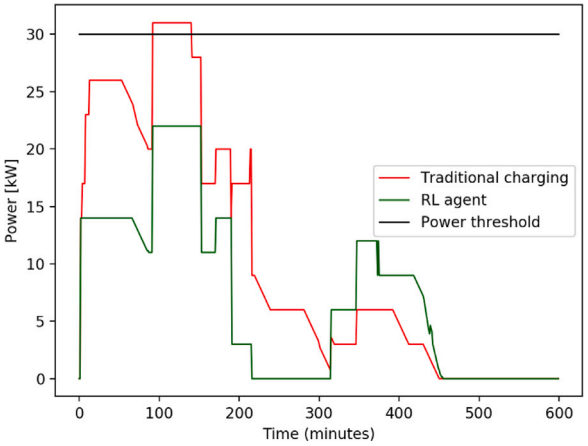

Fig. 15. Charging scheduling found by the RL agent (green curve) compared against the baseline currently implemented (red curve) for scenario 1.

First, it can be noticed that the total cumulative load due to EVs charging controlled by the intelligent agent, denoted by the green

curve, never exceeds the power line. It can be seen how the traditional charging algorithm (red curve) fails in the morning, when the occupancy undergoes a sudden peak due to all the employees arriving at the office. The agent demonstrates to be aware of the danger and it does not charge a lot during the morning peak, avoiding in this way the failure. It is also important to notice that a part of the load has been shifted later in the day. As it has been explained in Section 4.2, the agent can pick an action out of the many available, precisely 32 for this configuration. Fig. 16 shows the policy adopted by the agent in the first scenario.

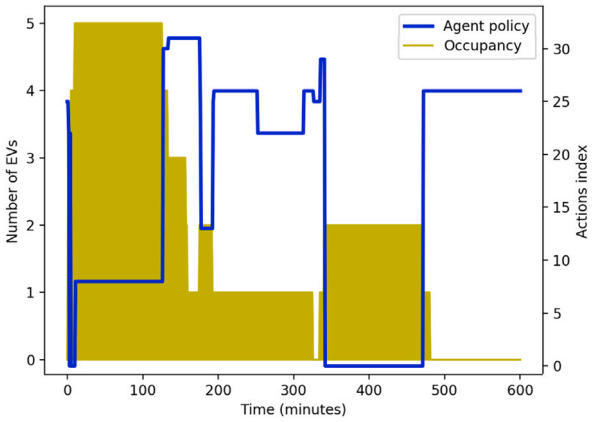

Fig. 16. Policy adopted by the agent compared against the occupancy in scenario 1.

The blue line represents the actions selected by the agent at every time step, while the green area in the background stands for the occupancy of that specific day considered. It is interesting to notice that the agent's policy is strongly affected by the level of occupancy. In fact, it is more conservative in the morning and early afternoon, where the occupancy has its peaks and more audacious when the parking lot is nearly empty. The agent reacts very well to the arrival/departure scheduling of the EVs without getting any direct information about the time of the day from the environment, and just knowing about the presence or absence of the cars.

5.3. Scenario 2 — Smart office building

The big challenge given by the second scenario is that the power threshold is not constant anymore, but instead variable with time. In the reward trend of Fig. 17, the network seems to converge quite early and still fluctuating a lot. Fig. 18 shows the results of the simulation during a scheduling day.

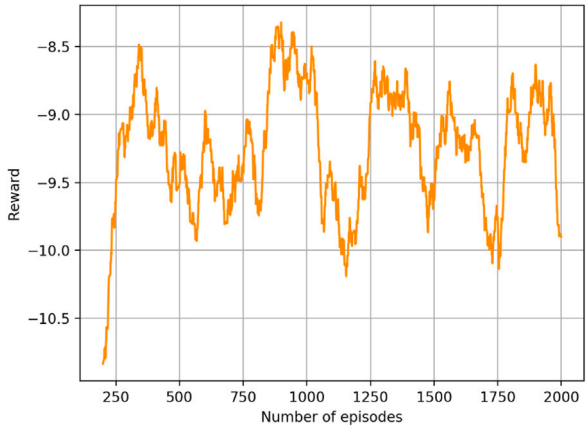

Fig. 17. Reward trend for scenario 2.

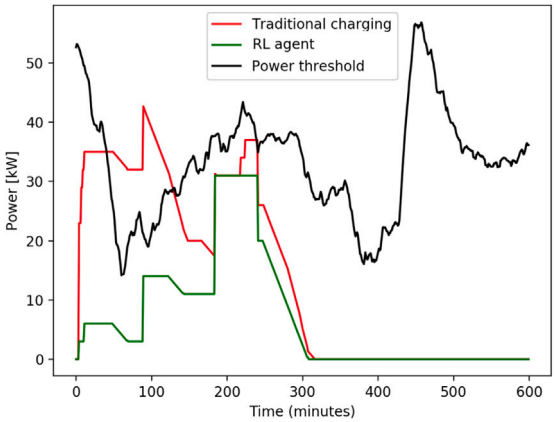

Fig. 18. Charging scheduling found by the RL agent (green curve) compared against the baseline currently implemented (red curve) for scenario 2.

The test is now more ambitious for the agent, because the power threshold varies during time and it reaches its minimum points during the occupancy's peaks. As it can be seen from the picture, the smart agent (green curve) successfully manages to avoid the failure and charge as much as it can the vehicles without exceeding the limit. Here it should be expected that the agent would take advantage of the uncharged cars left in the parking lot in order to complete the transactions. From how it can be seen in Fig. 19, the agents favors especially one action over the others, which corresponds to a specific combination of EVs charging and not charging. As a consequence, the energy is not homogeneously spread among the different charging sockets, which is with all probabilities the biggest limitation encountered by this approach. This behavior is possibly due to the fact that the reward condition related to the unnecessary actions has too much weight and discourage the agent of exploring other possibilities.

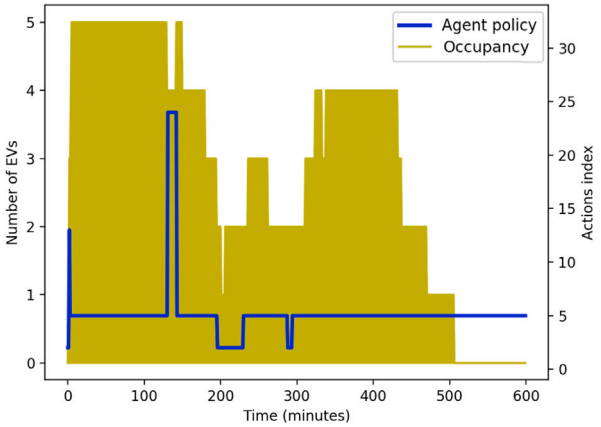

Fig. 19. Policy adopted by the agent compared against the occupancy in scenario 2.

5.4. Scenarios comparison

The two scenarios have been evaluated in the light of three main aspects: the *failures volume*, the amount of *energy transferred* and the number of *power adjustments*, namely the agent's frequency of taking different actions. A set of ten independent days has been simulated.

5.4.1. Failures volume

It has been thought that an interesting way to evaluate the gravity of failing was to measure the amount of energy that the agent would transfer where it would not be allowed to (namely, above the power

threshold line), also defined as *failures volume*. Fig. 20 shows that, in the first scenario, the agent seems to learn a policy that efficiently accomplishes this task, while in the second scenario failure occurs a few times reaching up to 20 kW. The failure volume has decreased up to around 80% on average in the 10 independent days considered compared to the baseline.

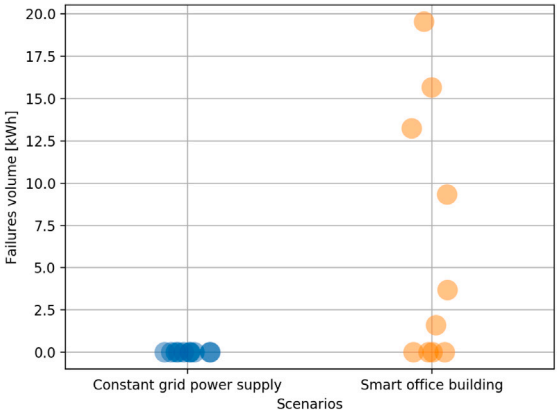

Fig. 20. Comparison between scenario 1 and scenario 2 in the failures volume.

5.4.2. Energy transferred

The other crucial metric of the problem is the percentage of energy transferred, on average, to every EV. Fig. 21 shows a scatter plot of the values of energy transferred in the ten runs considered for the two different agents. Once again, it can be seen the difference in the two scenarios: in the first one, the average is permanently above 70%, while in the second one it ranges between 40% and 60%.

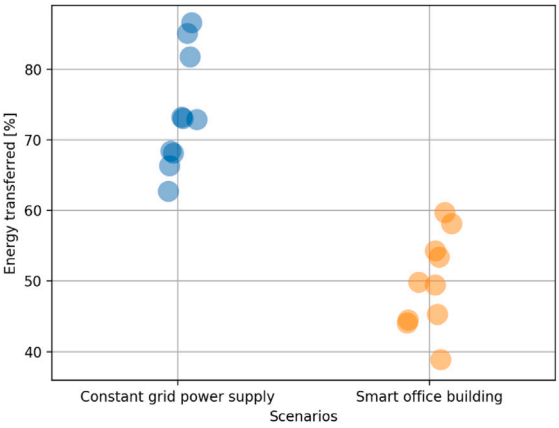

Fig. 21. Comparison between scenario 1 and scenario 2 in the energy transferred.

5.5. Power adjustments

Fig. 22 shows the number of power adjustments sent from the agent to the central system in order to change the scheduling of the EVs charging. The results in this case are highly affected by the policy learned by the agent during training. The charging scheme changes on average around 20 times, but if this number remains stable in scenario 1, it fluctuates in scenario 2.

6. Conclusions

In this paper, the problem of efficiently scheduling the charging of a fleet of electric vehicles has been addressed. The main objective was to avoid that the cumulative charging load would threaten the

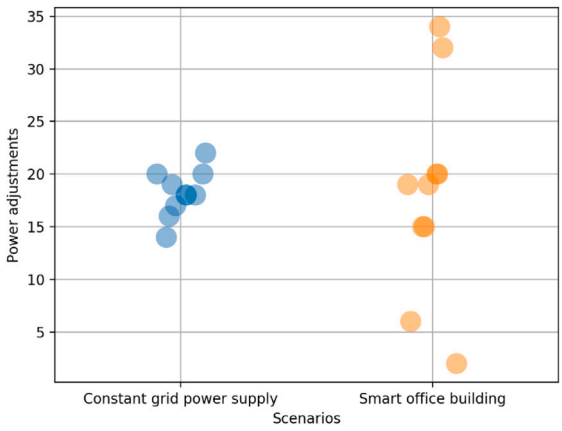

Fig. 22. Comparison between scenario 1 and scenario 2 in the power adjustments.

health of the energy system, both locally and from the perspective of the power grid. To this purpose, *deep reinforcement learning* concepts have been explored. An *environment* has been created in order to mimic real world complex dynamics, such as the basic operation of a charging station and the stochastic EVs arrival and departure functions. Multiple techniques have been combined to create a unique architecture, due to the fact that important information were not readily available but needed to be separately computed. The algorithm has been compared with a *traditional charging* simulated behavior, which revealed strengths and limitations of the proposed implementation. Moreover, the performance of the intelligent agent has been evaluated in two different scenarios, looking at tailored metrics such as *failures volume*, *energy transferred* and *power adjustments*. Results have showed that the developed algorithm achieved a reduction of the load due to EVs charging of 80% during the peak times of the day compared with the baseline. The amount of energy transferred varies with the scenario considered: 75% in the case of constant grid power supply and around 50% considering the smart office building as a whole. Further research could explore multi-agent reinforcement learning (Zhang, Yang, & Başar, 2019) for simultaneously handling the charging of multiple sockets. In addition, the implementation of a tailored application protocol (Alliance, 2015) for the communication between EV charging stations and the central system could also be investigated.

CRediT authorship contribution statement

Andrea Bertolini: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Visualization. **Miguel S.E. Martins:** Validation, Visualization, Writing – review & editing, Supervision. **Susana M. Vieira:** Writing – review & editing, Supervision. **João M.C. Sousa:** Writing – review & editing, Supervision.

Declaration of competing interest

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

Acknowledgments

This work was supported by the Portuguese Foundation for Science & Technology (FCT), through IDMEC, under LAETA, project UIDB/50022/2020. The work of Miguel Martins was supported by the PhD Scholarship 2020.08776.BD from FCT.

References

Alliance, O. C. (2015). Open charge point protocol 1.6.

Amjad, M., Ahmad, A., Rehmani, M. H., & Umer, T. (2018). A review of EVs charging: From the perspective of energy optimization, optimization approaches, and charging techniques. *Transportation Research Part D: Transport and Environment*, *62*, 386–417.

Dang, Q., Wu, D., & Boulet, B. (2019). A Q-learning based charging scheduling scheme for electric vehicles. In *2019 IEEE transportation electrification conference and expo* (pp. 1–5).

Ding, T., Zeng, Z., Bai, J., Qin, B., Yang, Y., & Shahidehpour, M. (2020). Optimal electric vehicle charging strategy with Markov decision process and reinforcement learning technique. *IEEE Transactions on Industry Applications*, *56*(5), 5811–5823.

Elmehdi, M., & Abdelilah, M. (2019). Genetic algorithm for optimal charge scheduling of electric vehicle fleet. In *NISS19: proceedings of the 2nd international conference on networking, information systems & security* (pp. 1–7).

Fang, X., Wang, J., Song, G., Han, Y., Zhao, Q., & Cao, Z. (2019). Multi-agent reinforcement learning approach for residential microgrid energy scheduling. *Energies*, *13*(1), 123.

Géron, A. (2019). *Hands-on machine learning with scikit-learn, keras and tensorFlow*. O'Reilly.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press, http://www.deeplearningbook.org.

Hasselt, H., Guez, A., & Silver, D. (2015). Deep reinforcement learning with double Q-learning. arXiv:1509.06461.

Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition, vol. 1* (pp. 278–282).

IRENA (2019). *Innovation outlook: Smart charging for electric vehicles*. International Renewable Energy Agency.

Jin, J., & Xu, Y. (2021). Optimal policy characterization enhanced actor-critic approach for electric vehicle charging scheduling in a power distribution network. *IEEE Transactions on Smart Grid*, *12*(2), 1416–1428.

Jin, J., Xu, Y., & Yang, Z. (2020). Optimal deadline scheduling for electric vehicle charging with energy storage and random supply. *Automatica*, *119*, Article 109096.

Karakatič, S. (2021). Optimizing nonlinear charging times of electric vehicle routing with genetic algorithm. *Expert Systems with Applications*, *164*, Article 114039.

Lapan, M. (2020). *Deep reinforcement learning hands-on*. Packt.

Lee, S., & Choi, D.-H. (2020). Energy management of smart home with home appliances, energy storage system and electric vehicle: A hierarchical deep reinforcement learning approach. *Sensors*, *20*(7), 2157.

Lee, J., Lee, E., & Kim, J. (2020). Electric vehicle charging and discharging algorithm based on reinforcement learning with data-driven approach in dynamic pricing scheme. *Energies*, *13*(8), 1950.

Li, H., Wan, Z., & He, H. (2020). Constrained EV charging scheduling based on safe deep reinforcement learning. *IEEE Transactions on Smart Grid*, *11*(3), 2427–2439.

Mocanu, E., Mocanu, D. C., Nguyen, P. H., Liotta, A., Webber, M. E., Gibescu, M., et al. (2019). On-line building energy optimization using deep reinforcement learning. *IEEE Transactions on Smart Grid*, *10*(4), 3698–3708.

Nour, M., Chaves-Ávila, J., Magdy, G., & Sánchez-Miralles, A. (2020). Review of positive and negative impacts of electric vehicles charging on electric power systems. *Energies*, *13*(18), 4675.

Quddus, M. A., Yavuz, M., Usher, J. M., & Marufuzzaman, M. (2019). Managing load congestion in electric vehicle charging stations under power demand uncertainty. *Expert Systems with Applications*, *125*, 195–220.

Sadeghianpourhamami, N., Deleu, J., & Develder, C. (2018). Achieving scalable model-free demand response in charging an electric vehicle fleet with reinforcement learning. In *The ninth international conference* (pp. 411–413).

Sallam, K. M., Chakrabortty, R. K., & Ryan, M. J. (2021). A reinforcement learning based multi-method approach for stochastic resource constrained project scheduling problems. *Expert Systems with Applications*, *169*, Article 114479.

Shi, W., & Wong, V. W. S. (2011). Real-time vehicle-to-grid control algorithm under price uncertainty. In *2011 IEEE international conference on smart grid communications* (pp. 261–266).

Silva, M. A. L., Souza, S. R., Souza, M. J. F., & Bazzan, A. L. C. (2019). A reinforcement learning-based multi-agent framework applied for solving routing and scheduling problems. *Expert Systems with Applications*, *131*, 148–171.

Sortomme, E., & El-Sharkawi, M. A. (2012). Optimal scheduling of vehicle-to-grid energy and ancillary services. *IEEE Transactions on Smart Grid*, *3*(1), 351–359.

Sutton, R. S., & Barto, A. G. (2014-2015). *Reinforcement learning: An introduction*. MIT Press.

Virta (2020). Smart charging of electric vehicles. URL https://www.virta.global/smart-charging. (Accessed 13 March 2021).

Škugor, B., & Deur, J. (2014). Dynamic programming-based optimization of electric vehicle fleet charging. In *2014 IEEE international electric vehicle conference* (pp. 1–8).

Wan, Z., Li, H., He, H., & Prokhorov, D. (2019). Model-free real-time EV charging scheduling based on deep reinforcement learning. *IEEE Transactions on Smart Grid*, *10*(5), 5246–5257.

Wang, S., Bi, S., & Zhang, Y. A. (2018). A reinforcement learning approach for EV charging station dynamic pricing and scheduling control. In *2018 IEEE power energy society general meeting* (pp. 1–5).

Wei, T., Wang, Y., & Zhu, Q. (2017). Deep reinforcement learning for building HVAC control. In *2017 54th ACM/EDAC/IEEE design automation conference* (pp. 1–6).

Wu, D., Zeng, H., Lu, C., & Boulet, B. (2017). Two-stage energy management for office buildings with workplace ev charging and renewable energy. *IEEE Transactions on Transportation Electrification*, *3*(1), 225–237.

Zhang, C., Liu, Y., Wu, F., Tang, B., & Fan, W. (2021). Effective charging planning based on deep reinforcement learning for electric vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *22*(1), 542–554.

Zhang, F., Yang, Q., & An, D. (2021). CDDPG: A deep-reinforcement-learning-based approach for electric vehicle charging control. *IEEE Internet of Things Journal*, *8*(5), 3075–3087.

Zhang, K., Yang, Z., & Başar, T. (2019). Multi-agent reinforcement learning: A selective overview of theories and algorithms. arXiv:1911.10635.

Zheng, Z., & Yang, S. (2020). Particle swarm optimisation for scheduling electric vehicles with microgrids. In *2020 IEEE congress on evolutionary computation* (pp. 1–7).

Zishan, A., Haji, M., & Ardakanian, O. (2020). Adaptive control of plug-in electric vehicle charging with reinforcement learning. In *The eleventh ACM international conference on future energy systems* (pp. 116–120).