A Q-learning Algorithm for Optimizing On-load Tap Changer Operation and Voltage Control in Distribution Networks with High Integration of Renewable Energy Sources

Alessandro Bosisio, Francesca Soldan, Matteo Pisani, Enea Bionda, Federico Belloni, and Andrea Morotti

***Abstract*—Distribution networks have been experiencing significant changes under the pressure of the energy transition. The high integration of renewable energy sources combined with the electrification process introduces new challenges in managing distribution networks. Innovative solutions aimed at optimizing the control of complex problems, starting from historical data instead of a detailed system model, have been growing due to rapid development in artificial intelligence and machine learning. This paper proposes a Q-learning algorithm to control the tap setting of the on-load tap changer installed in primary substation transformers. The ultimate goal is to maintain voltage magnitudes at all busses of the medium-voltage distribution network within a safe range, simultaneously optimizing on-load tap changer operations. As a case study, the effectiveness of the proposed algorithm is assessed using a real medium-voltage distribution network with high penetration of renewable energy sources that supplies more than 2500 users/prosumers. The ability of the proposed algorithm to control bus voltages is tested in several scenarios characterized by significant variability and uncertainty. Outcomes show that the proposed algorithm is suitable for optimizing voltage control in distribution networks using a data-driven approach.**

***Index Terms*—Automatic voltage control, on-load tap changer, distribution network, renewable energy source, energy transition, reinforcement learning, artificial intelligence, smart grid.**

I. Introduction

THE energy transition is driving profound changes within power distribution networks. Massive integration of renewable energy sources (RESs) combined with electrification of consumption introduces new challenges in managing distribution networks. A medium-voltage (MV) distribution network typically has a radial topology with several feeders departing from the secondary busbar of the high-voltage (HV)/MV transformer. Compared with the HV grid, the voltages of distribution networks are also influenced by active power injections due to a higher resistance/reactance ratio of the lines. The distribution system operator (DSO) usually maintains the voltage magnitudes along the feeders within the admissible range, controlling the on-load tap changer (OLTC) transformer installed in the primary substation. In regular operation, the OLTC behaves as an automatic voltage regulator (AVR) based on the voltage measurement at the secondary busbar of the transformer [1]. Unareti, one of the leading Italian DSOs, traditionally acts similarly: when the voltage deviation at the secondary busbar of HV/MV transformer exceeds ±2.5% of the system nominal voltage, the AVR changes the OLTC position to control the voltage level.

The AVR set point is commonly set to comply with the voltage fluctuations in the transmission network and compensate for the voltage drop at the end of the lines. The control voltage typically ranges between 10% and 20% of the rated voltage at the MV busbar of the substation transformer. The widespread installation of distributed generation (DG) units and the increase in electricity demand can affect the quality of the voltage regulation control: the presence of DG alters the conventional structure of the network and causes inhomogeneity of generation/load profiles among the different feeders.

In recent years, many studies have developed algorithms to handle voltage regulation issues. In the majority of suggested approaches, however, the optimal control solution is to solve the optimal power flow (OPF) problem [2]-[4].Several models and algorithms are proposed, including mixed-integer nonlinear programming, quadratic programming, Theve-

Manuscript received: May 21, 2024; revised: November 26, 2024; accepted: April 21, 2025. Date of CrossCheck: April 21, 2025. Date of online publication: May 5, 2025.

This work was supported by the Research Fund for the Italian Electrical System under the Three-Year Research Plan 2022-2024 (DM MITE No. 337, 15.09.2022), in compliance with the Decree of April 16, 2018.

A. Bosisio (corresponding author) is with the Department of Electrical, Computer and Biomedical Engineering, University of Pavia, 27100 Pavia, Italy (e-mail: alessandro.bosisio@unipv.it).

F. Soldan and E. Bionda are with the Research Center, Ricerca sul Sistema Energetico (RSE S.p.A.), 20134 Milano, Italy (e-mail: enea.bionda@rse-web.it; francesca.soldan@rse-web.it).

M. Pisani was with the Energy Department, Politecnico di Milano, 20156 Milano Italy, and he is now with Edison S.p.A., 20121 Milano, Italy (e-mail: matteo1.pisani@mail.polimi.it).

F. Belloni and A. Morotti are with Unareti S.p.A., 20128 Milano, Italy (e-mail: federico.belloni@unareti.it; andrea.morotti@unareti.it).

DOI: 10.35833/MPCE.2024.000528

nin equivalent-based and recursive approaches, primal-dual decomposition, particle swarm optimization, forward/back ward sweep algorithm and others.

OPF models rely on accurate and complete network models, which is not always practical for the sizeable interconnected power systems nowadays with increasing complexity. Moreover, the lack of widespread real-time measurements that still exist in real distribution networks is another problem, which prevents their usage in real-world applications. In addition, OPF model usually cannot be used to make speedy decisions since it needs to solve a sizeable non-linear optimization problem through many iterations. The above-discussed criticalities can be addressed using a data-driven approach. Recently, developments in artificial intelligence (AI) and machine learning (ML) techniques have inspired new solutions where the control of complex problems is optimized by starting from historical data instead of a detailed system model. Within ML techniques, reinforcement learning (RL) has an outstanding potential to solve distribution network problems, since it can learn to make decisions by interacting with an environment to maximize cumulative reward. This paper exploits Q-learning algorithms, which enable agents to learn optional action-selection policies by iteratively updating Q-values, representing the expected cumulative rewards for taking a specific action in a given state.

The very first paper presenting an RL approach to control voltage magnitudes was published in 2004 [5]. The control variables are three tap changers and one switched capacitor. The optimal configuration of each controlling device has been obtained through a Q-learning algorithm, in which the set of possible actions is intrinsically finite, and the network states have been discretized. In [6], a Q-learning approach is proposed to manage the OLTC installed at the primary substation, using a large Brazilian MV/low-voltage (LV) network with high photovoltaic (PV) penetration as a case study. To make the application of a Q-table possible, the set of states refers to a few critical customers whose voltage magnitudes have been converted into discrete intervals. Whereas Q-learning algorithms work effectively to regulate OLTC equipment, artificial neural networks (ANNs) are essential to deal with reactive power control devices. An increasing share of literature articles proposes deep learning (DL) techniques to manage the reactive and active power of inverter-based DG units. In [7], the agent learns optimal set points of reactive power with a deep deterministic policy gradient (DDPG) method to minimize voltage deviations, PV active power curtailment, and power system losses simultaneously. On the other side, a deep Q network (DQN) algorithm, an evolution of Q-learning that implements ANNs, is proposed to control generator set points in [8]. After offline training, whenever the agent detects abnormal conditions in real time, it suggests some control actions to human operators, who will assess the value of the response and decide if it should be implemented. A comparison between the performances of the DQN and DDPG for voltage regulation of generator is given by [9]. The DDPG agent, which controls actions defined in a continuous space, achieves better performances than the DQN agent, based on a discrete action space. Reference [10] proposes a multiagent DDPG to regulate voltage set points of generator groups partitioned in different regional zones. Each agent will optimize actions in a specific zone according to local measurements after a training phase requiring a centralized communication network. In [11]-[13], voltage regulation is jointly achieved using reactive power control of DG units and static var compensators (SVCs). Reference [11] exploits a DDPG model. The relationship between the power injection and voltage fluctuation at the busses is retrieved in a supervised manner from historical data, instead of using a system model to run power flows. Reference [12] implements a soft-actor-critic (SAC) agent to perform the automatic voltage regulation, while in [13], a multiagent DDPG is used to control different portions of the network based on the latest local information. There are also studies based on deep RL models aimed at driving OLTC operations, as standalone control systems or in combination with other devices. A policy for optimal tap settings in the presence of one or more transformers is shown in [14], where a scheme based on linear function approximation called least square policy iteration (LSPI) has been used. The SAC agent of [15] selects the tap position of controllable tap changers and switches on/off capacitors simultaneously to minimize voltage deviation, losses, and unneeded switches. An application of OLTC tap control, optimized together with the DG units output, is presented in [16]. A DDPG scheme is proposed to minimize voltage deviation and active power curtailment, simultaneously reducing the wear of the tap mechanism. A couple of works [17], [18] discuss RL frameworks based on a two-timescale voltage control. On the slower timescale, shunt capacitors are configured in on/off state, while on the faster timescale, the generator power is regulated to minimize instantaneous voltage deviations, and the state of charge of battery energy storage systems (BESSs) is optimized [18]. In [17], a DQN algorithm is used within both timescale regulations, whereas in [18], DQN is used only for switched capacitors; while for DG and BESS outputs, DDPG is exploited.

This paper proposes a Q-learning algorithm to control the OLTC installed in primary substation transformers. The ultimate objective is to eliminate voltage issues within the entire distribution network, minimizing unnecessary OLTC tap switches simultaneously. The data-driven AVR will learn to master the OLTC operations accounting on a restricted number of measured voltages. To further increase the reliability of the control system, the voltage regulator learns to maintain bus voltages within a narrower band of admissible values during the training process. The effectiveness of the proposed algorithm is tested in two scenarios with different values of both demand and generation, characterized by significant variability and uncertainty, using the real three-phase MV distribution network of a primary substation in Vobarno, managed by the local DSO, Unareti.

The contribution of this paper can be summarized as follows.

1) RL is used to solve distribution network issues controlling OLTC installed in primary substation transformers.

2) A sensitivity analysis of the RL hyperparameters is con-

ducted, both in training and testing, to minimize the reward function.

3) The proposed algorithm is tested on a real distribution network with high penetration of RESs in the current and future scenarios, considering the increasing DGs and electrification of consumption.

The remainder of this paper is organized as follows. Section II briefly reports the theoretical background of Q-learning algorithm. Section III presents the problem formulation of the OLTC control, while Section IV shows the sensitivity analysis of RL hyperparameters. Section V presents the case study and numerical results obtained, where the proposed algorithm is compared with traditional control strategy. Concluding remarks and future work are given in Section VI.

II. Theoretical Background of Q-learning Algorithm

RL is one of the three main ML subsets, together with supervised learning and unsupervised learning. In RL, the control system learns how to act to maximize a numerical reward [19]. The learner discovers optimal actions through a trial-and-error search to obtain the largest reward in the long run. The main advantage of using an RL controller lies in its ability to learn the best control action from scratch and gradually master the system operation.

All the RL algorithms are formulated using the Markov decision process (MDP), as shown in Fig. 1. The key elements of MDP are hereafter presented.

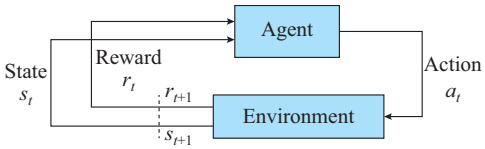

Fig. 1. Agent-environment interaction in an MDP.

1) Agent: the subject that performs actions to maximize the long-term reward.

2) Environment: the surrounding world in which the agent interacts and responds to the action selected.

3) State s_t: a set of significant variables aimed at representing the environment at that moment. All the possible values that can be assumed by this set of variables are called state space S. When the agent does not have access to all the state variables (fully observable environment), the known part is often called observation o_t.

4) Action a_t: the decision taken by the agent that belongs to the action space A, i.e., the set of all the possible actions that can be performed. When the number of possible actions is finite, the action space is discrete; vice versa, if it is infinite, the action space is continuous [20].

5) Reward r_t: the immediate feedback, usually a scalar number received from the environment and used to evaluate the last action performed by the agent.

The basic principle of MDP consists of an agent (the controller) interacting with a dynamic environment over a sequence of discrete steps. In each time step t, the agent receives a representation of the system through s_t and in response to that, the state selects a_t. Action is taken following a policy function $\pi(a|s)$ expressed as:

$$\pi(a|s)=\mathbb{P}[a_t=a|s_t=s] \tag{1}$$

The policy function maps the probability of selecting each possible action from the current state. As a consequence of the performed action, the environment sends back to the agent a reward signal r_{t+1} related to the new state s_{t+1} in which the system ends up. The goal of the agent is to learn a policy that maximizes the expected long-term reward. When a policy π is being used, the action-value function (also called Q-value) $Q_\pi(s,a)$ is given by:

$$Q_\pi(s,a)=\mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}\,\middle|\,a_t=a,s_t=s\right] \tag{2}$$

Equation (2) determines how good it is to take a particular action from a state s, and this can only be derived from experience. The discount rate γ (defined in the range of $0<\gamma<1$) is meant to decrease the worth of future rewards. Finding the optimal policy is equivalent to finding the optimal action-value function $Q_*(s,a)=\max_\pi Q_\pi(s,a)$.

Q-learning is a model-free technique; hence, a detailed model with the system dynamics is not needed. It is a tabular method since its working principle is based on a Q-table that stores a Q-value $Q(s,a)$ for each action-state pair. The Q-table has dimension $Q(S\times A)$. Since the algorithm relies on a conventional tabular structure, the number of elements that compose the matrix needs to be finite, hence both S and A are discrete spaces. An example of Q-table with dimension $m\times n$ defined for a set of states $S=\{s_1,s_2,\ldots,s_m\}$ and $A=\{a_1,a_2,\ldots,a_n\}$ is given in Table I.

TABLE I
Q-table with Dimension $m\times n$

S	Q-value			
	a_1	a_2	...	a_n
s_1	$Q(s_1,a_1)$	$Q(s_1,a_2)$	...	$Q(s_1,a_n)$
s_2	$Q(s_2,a_1)$	$Q(s_2,a_2)$	...	$Q(s_2,a_n)$
⋮	⋮	⋮		⋮
s_m	$Q(s_m,a_1)$	$Q(s_m,a_2)$	...	$Q(s_m,a_n)$

Upon each step, the agent chooses an action according to the Q-values. When implemented as a real-time controller, the agent always takes the action with highest Q-value and consequently the highest expected return from the current state, whereas during the learning process, it can also select random actions to explore more alternatives. Once a_t is performed, the agent receives r_{t+1} and falls in s_{t+1}. The value of $Q(s_t,a_t)$ within the Q-table is updated following (3).

$$Q(s_t,a_t)=(1-\alpha)Q(s_t,a_t)+\alpha\left(r_{t+1}+\gamma\max_{a_{t+1}}Q_*(s_{t+1},a_{t+1})\right) \tag{3}$$

where α is the learning rate ($0<\alpha\leq 1$) specifying the extent to which the action-value function is updated in each iteration [20]. The agent gradually enhances the Q-table estimations by interacting with the environment and updating Q-values following (3).

III. Problem Formulation of OLTC Control

The proposed Q-learning algorithm uses voltage magnitudes at some distribution network buses as input, and provides the best tap setting as output. This pattern generates the optimal performances to comply with specified reward requirements, relying on the Q-values stored within the Q-table. The voltage control problem is designed following an MDP formulation.

Q-learning agent is the AVR driving OLTC operations. OLTC control is applied within a distribution network, which is the environment. As depict in (4), at each step, the RL agent can regulate the OLTC settings following three possible actions:

$$\Delta Tap = \begin{cases} -1 & Tap_t \neq Tap_{\min} \\ 0 & \forall Tap_t \\ +1 & Tap_t \neq Tap_{\max} \end{cases} \tag{4}$$

where ΔTap is the variation of the tap position with respect to the current tap position; Tap_t is the current tap position; and $Tap_{\min}$ and $Tap_{\max}$ are the minimum and maximum taps, respectively. Action space A is intrinsically discrete and has dimension $n_a = 3$.

A state is defined by the combination of voltage magnitudes at n buses spotted to be critical for either undervoltages or overvoltages during preliminary analysis. Since the algorithm is based on the Q-table and voltage magnitudes are continuous, the values are discretized according to different ranges, which can be defined through uniform voltage steps of 0.01 p.u. between the lower ($V_{\min} = 0.95$ p.u.) and upper ($V_{\max} = 1.05$ p.u.) voltage limits, plus ranges $v \leq 0.95$ p.u. and $v \geq 1.05$ p.u., for a total of twelve steps.

Considering i as the number of existing ranges, the state space dimension n_s is equal to (5).

$$n_s = i^n \tag{5}$$

Since n_s grows exponentially as the function of critical buses, n could be as small as possible. In fact, it is better to reduce the size of the Q-table to update each Q-value more times, so that it can converge to an accurate estimate of the real unknown value.

The ultimate objective of the proposed Q-learning algorithm is to eliminate voltage issues within the distribution network, minimizing OLTC operations, which causes long-term wear and tear on the device. The agent will learn to master system operations according to a reward function defined as in (6).

$$r_{t+1} = \begin{cases} -C_{TC}\left|\Delta Tap_t\right| + R_{neg}(V_{\min} - V_{n,t}) & \forall V_{n,t} < V_{\min} \\ -C_{TC}\left|\Delta Tap_t\right| + R_{neg}(V_{n,t} - V_{\max}) & \forall V_{n,t} > V_{\max} \end{cases} \tag{6}$$

where C_{TC} is the cost for each tap movement throughout the entire life of the device; $V_{n,t}$ is the voltage of bus n at time t; and R_{neg} is the penalty coefficient associated with voltage constraints. On the one hand, a penalty is assigned when modifying the tap position ($\Delta Tap_t \neq 0$). On the other hand, all the buses in which voltage limits $[V_{\min}, V_{\max}]$ are violated receive a penalty proportional to the voltage deviation. The penalty given to nodes outside the permissible range grows when voltage deviation increases, which is scaled by $R_{neg} = \Delta t\lambda$, where λ is the cost if the voltage level remains out of range for one hour. Even if Italian technical standards still do not prescribe automatic compensations for users supplied with unacceptable voltage levels, a high value of λ has been assigned to push the agent to avoid this condition. For the case study, $\lambda = 100$, thus $R_{neg} = 25$ when using a quarter-an-hour resolution.

Equation (6) is used to train and subsequently evaluate the performances of agent, although reward parameters have different values in the two phases. The penalty parameter referred to tap movements C_{TC} will be heuristically tuned during training, in order to find the optimal setting that avoids any voltage problem with the minimum number of OLTC operations. One of the main concerns about implementing data-driven control systems in critical infrastructures is the safety of using the algorithm in an environment with high variability and uncertainty. Hence, the Q-learning agent will learn the optimal strategy for complying with stricter voltage limits. The safety principle is also followed to evaluate the trained agent: the designed RL controller first needs to avoid any voltage violation during the test phase; only if the safety principle has been fulfilled will the performances be numerically assessed using the reward signal (6).

During the offline learning process, the data-driven agent explores a wide range of power system states and improves its policy through multiple simulations. In a real-time control process, the calibrated controller will exploit its functional map from state to action learned during training, directly applying the optimal control action for that specific power system condition. The RL agent is hence not interested in complex power system dynamics like in OPF problems, making it practical for real systems and much faster than an OPF controller. To train and subsequently test the RL agent, Q-learning algorithm is implemented in Python using Gym-ANM library [21], [22]. The simulation environment that models the actual system has been designed to exploit and modify the Gym-ANM framework. The structure of the adapted Gym-ANM framework shown in Fig. 2 is based on the classical MDP described above [23].

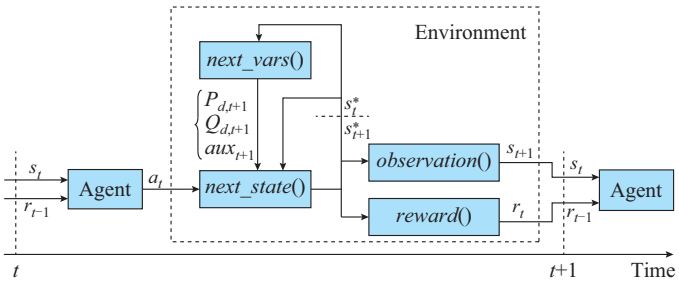

Fig. 2. Gym-ANM framework used to train RL agent.

At each step t, the agent receives information about the current state s_t, i.e., the voltage magnitudes, and the reward r_t resulting from the previous interaction with the environment. The selected action a_t, i.e., the tap-setting, is sent to the environment, in which a series of internal variables is sampled using the *next_vars*() block. *next_vars*() allows the

transition from step t to $t+1$ conditioned on the current state of the environment, fully described by s_t^* (including bus voltages, branch currents, active and reactive power exchanges at each bus, etc). The two types of variables processed by *next_vars*() are:

1) The temporal evolution of active power $P_{d,t+1}$ and reactive power $Q_{d,t+1}$ withdrawn or absorbed by each device, either related to loads or DG units.

2) An auxiliary variable aux_{t+1} modelling other temporal factors that influence the outcomes of $P_{d,t+1}$ and $Q_{d,t+1}$. For the case study, the auxiliary variable represents the time of the day.

Internal variables are then passed to *next_state*() block along with a_t and s_t. In *next_state*(), the action taken is applied to the environment, and the new full state of the system s_{t+1}^* is simulated. This built-in function computes all the system variables through a Newton-Raphson power flow method. s_{t+1} containing input measurements is used to train the RL agent, which can be directly extracted from s_{t+1}^* by means of the observation function. In addition, the return r_{t+1} is computed following the pre-specified reward parameters.

The Q-learning agent will interact with simulated data for an entire year, among which a large part is used to train the agent, while a smaller share is meant to evaluate the performance of the model. Training is performed for 335 days, referred to as episodes, and each episode finishes after training 96 samples of one day (quarter-an-hour data, thus $\Delta t = 0.25$ hour). The test phase is instead based on 30 episodes (one month) with the same time resolution.

On the one hand, daily time series $P_l\left[1,2,\ldots,\frac{24}{\Delta t}\right]$, $Q_l\left[1,2,\ldots,\frac{24}{\Delta t}\right]$, $P_g\left[1,2,\ldots,\frac{24}{\Delta t}\right]$ of each load and generator will be exploited by the aforementioned function *next_vars*(). On the other hand, the voltage level at the HV busbar of the primary substation is modeled through daily profiles $v_{sb}\left[1,2,\ldots,\frac{24}{\Delta t}\right]$.

A data-driven agent can hence interact with the developed environment to learn the optimal strategy for regulating voltages within the MV distribution network at the minimum cost. The goal is to determine the tap-setting leading to the highest return when a set of discretized voltages is observed. At each step, the agent should improve its knowledge of how the environment responds when a certain action is executed. In Q-learning models, this is achieved by updating $Q(s_t, a_t)$ (the Q-value referred to state-action of that step) using (3) at each iteration. The training process of the Q-learning algorithm is given in Algorithm 1.

Once the simulation framework is ready, the data-driven agent can start learning by interacting with the environment in different situations. At each time step, the Q-learning agent chooses $a_t \in A$ following an ϵ-greedy strategy, defined as in (7).

$$\pi(a|s) = \begin{cases} a = \arg\max_a Q_*(s,a) & \text{w.p. } 1-\epsilon \\ a = \text{random}(A) & \text{w.p. } \epsilon \end{cases} \tag{7}$$

where $0 \le \epsilon \le 1$ points out the probability of taking an action randomly; and "w.p." represents "with probability". Hence, a_t will depend on both the Q-values associated with $s_t \in s$ and on ϵ, usually higher in the first episodes to favor exploration. The use of an ϵ-greedy strategy aims at finding a trade-off between exploration (when a random action is selected) and exploitation (when the agent takes a decision that maximizes the expected return according to the already learned Q-values). Especially at the beginning, introducing an ϵ-greedy strategy prevents always taking the same route, encouraging the agent to explore alternatives to achieve better performances in the future. For this reason, policies in which ϵ decreases are often effective choices. The simulation environment will receive a_t and a full estimate of the grid s_t^* as input, and output s_{t+1}^* (which is a function of load and generation injections and slack bus voltage at step t), as shown in Fig. 2. Immediate feedback r_t is calculated according to the training reward parameters, while the vector of observed voltages is directly retrieved from s_{t+1}^*. Voltages are converted using specified voltage ranges to obtain the discrete state s_{t+1}. Given r_t, s_t, s_{t+1}, and hyperparameters α, γ, Q-table is updated in position $Q(s_t, a_t)$ following the off-policy technique [23] of (3). This is the key moment of the RL agent learning process. The control system knowledge about the environment dynamics is indeed enhanced: if the action taken is good in terms of immediate reward as well as of value of the new system state, $Q(s_t, a_t)$ increases and a_t will be more likely to be selected when acting greedy. At the end of step t, the output s_{t+1} becomes s_t that is the input of the next MDP loop. Once the last step T has been performed, hyperparameters with variable value across episodes, e.g., ϵ and α, are updated. This process is repeated throughout all the days of the training set for N_{train} times.

Algorithm 1: training of Q-learning algorithm

Initialize parameters of model: number of training episodes N_{train}, time resolution Δt, reward parameters (C_{TC}, λ, $V_{\min}$, $V_{\max}$), hyperparameters (α, γ, ϵ), and discretization ranges
Initialize the Q-table ($n_s \times n_a$) with arbitrary values, i.e., all zeros
for $ep = 1, 2, \ldots, N_{train}$ **do**
  Retrieve $P_l\left[1,2,\ldots,\frac{24}{\Delta t}\right]$, $Q_l\left[1,2,\ldots,\frac{24}{\Delta t}\right]$, $P_g\left[1,2,\ldots,\frac{24}{\Delta t}\right]$ of each load and generator
  Retrieve $v_{sb}\left[1,2,\ldots,\frac{24}{\Delta t}\right]$
  Receive initial discrete observation s_1
  for $t = 1, 2, \ldots, \frac{24}{\Delta t}$ **do**
    Choose a_t from s_t following policy derived by Q-values, e.g., ϵ-greedy of (7)
    Execute a_t in power flow environment to transfer input state s_{t+1} and obtain reward r_t
    Discretize s_{t+1} using discrete voltage ranges
    Update Q-table:
    $$Q(s_t, a_t) \leftarrow (1-\alpha) Q(s_t, a_t) + \alpha\left(r_t + \gamma \max_{a_{t+1}} Q_*(s_{t+1}, a_{t+1})\right)$$
    Update $s_t \leftarrow s_{t+1}$
  end for
  Update hyperparameters ϵ and α (only if needed)
end for

After training, the Q-learning agent should be able to map the best action for any feasible state of the system. In practical applications, the data-driven voltage control system will receive a combination of real-time voltage measurements monitored at the same busses used for training as input. Once the actual voltage magnitudes are converted into a discrete state, the AVR selects a tap-setting that maximizes the long-term return according to Q-values learned from training data and stored in the Q-table. The strategy followed by the Q-learning agent when used as real-time voltage controller is hence a greedy policy ($\pi(a|s)=\arg\max_a Q_*(s,a)$), which exploits experience to optimize the reward in the long run. This process continues for an indefinite time whenever the agent observes a state of the distribution network, so there is neither a constant Δt nor fixed episodes made of $\frac{24}{\Delta t}$ time steps.

IV. Sensitivity Analysis of RL Hyperparameters

During the training phase, a sensitivity analysis of RL hyperparameters (ϵ, α, γ) has been performed. ϵ and α are respectively represented by exponential functions, whereas the discount rate remains constant. The set of hyperparameters initially used will be called Set 0. Numerical expressions of parameters in Set 0 are given as: $\epsilon=\underline{\epsilon}+(\bar{\epsilon}-\underline{\epsilon})e^{-\tau_\epsilon \cdot ep}$ (with $\underline{\epsilon}=0.01$, $\bar{\epsilon}=1$, $\tau_\epsilon=0.01$), $\alpha=\underline{\alpha}+(\bar{\alpha}-\underline{\alpha})e^{-\tau_\alpha \cdot ep}$ (with $\underline{\alpha}=0.01$, $\bar{\alpha}=0.1$, $\tau_\alpha=0.01$), and $\gamma=0.95$, where *ep* is the episode.

The results are summarized as follows.

1) ϵ: the trade-off between explorations and exploitation is determined. In addition to Set 0, two other sets have been tested. ① Set 1: the same as Set 0, but with $\tau_\epsilon=0.005$. In this case, the decay rate of the exponential function has been halved; therefore, the agent explores for a longer time and has more significant fluctuations. ② Set 2: $\epsilon=0.01$. In this case, the degree of exploration is reduced. Results show that the agent trained with Set 1 hyperparameters performs similarly to the agent trained with parameters in Set 0, while worse rewards are obtained in Set 2. In addition, voltage limits have not been passed using the epsilon of Set 0 and Set 1, whereas Set 2 agent causes some voltage issues. Finally, increasing ϵ may cause lower rewards during the training period, but once the trained agent is implemented as a real-time controller, it can perform better.

2) α: the learning rate α determines the extent to which Q-values are updated at each iteration following (3) in this paper. In addition to Set 0, two other sets have been tested: ① Set 3: $\alpha=0.5$; ② Set 4: $\alpha=0.001$. Using a large learning rate leads to almost double tap switches (208 instead of 115), but effectively eliminates voltage violations during the test period. On the other hand, a small learning rate causes remarkable voltage violations, but it drastically reduces the OLTC operations (42 instead of 115). Since the main agent requirement is the reliability of voltage control, the α-function of Set 4 is not considered better than the initial setting. Instead, the user will decide if voltage limits can be passed without negative consequences. In this case, the choice of Set 0 is still the best for a trade-off between the number of tap switches and voltage stability.

3) γ: the discount rate γ aims to teach the agent its primary goal. Lower γ privileges immediate rewards, whereas values close to one are used to maximize the long-term return. In Set 0, γ has been set equal to 0.95, while in the last configuration (Set 5), $\gamma=0.5$ has been used. Training the agent with a lower discount rate reduces tap switches (46 versus 115) but causes voltage problems at some buses. This is the expected result of reducing the discount rate: the agent will not change tap position even if some bus voltages get closer or slightly beyond the limits, because it would immediately receive a large penalty due to its action.

The number of tap switches for all the considered sets is shown in Fig. 3, in which cases bringing voltage issues are highlighted in orange color.

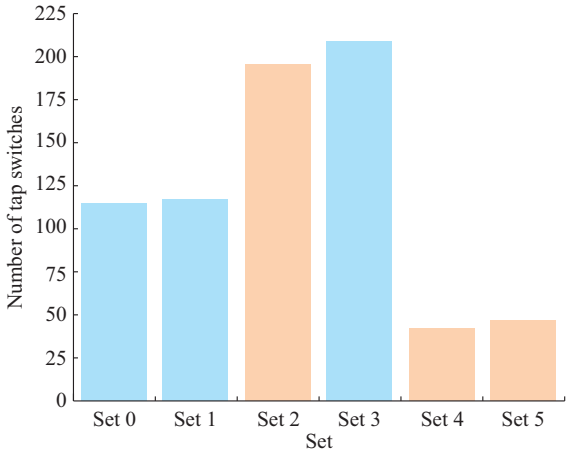

Fig. 3. Number of tap switches for all considered sets.

V. Case Study and Numerical Results

The effectiveness of the proposed Q-learning algorithm is assessed using a real MV distribution network located near the Garda Lake in Italy, operated by the local DSO, Unareti. The primary substation Vobarno supplies the distribution network, which consists of 2728 domestic users. Moreover, 180 RESs, of which 176 are solar PV plants and four are run-of-the-river hydroelectric plants, are also connected. The distribution network has a radial topology and is comprised of 274 buses, one HV/MV transformer with electronic OLTC, and 59 MV/LV transformers. The 132 kV HV transmission grid supplies the 40 MVA transformer with OLTC. This transformer features a nominal turn ratio $k_{nom}=V_{1,rat}/V_{2,rat}=$ 132 kV/15 kV, where $V_{1,rat}$ and $V_{2,rat}$ are the rated voltages on the two sides of the transformer. The regulation between 83.5% and 116.5% of $V_{2,rat}$ is discretely achieved through 23 tap positions, in which the voltage step Δv between consecutive taps is 1.5% of $V_{2,rat}$.

Figure 4 shows the topology of the considered distribution network, highlighting critical buses regarding voltage issues, used as input by the AVR. Bus 1 immediately reflects fluctuations of the HV voltage provided by the TSO, and additionally, a voltage measurement device is already installed at the secondary side of the HV/MV transformer, thus not implying additional costs to the DSO. Bus 204 and Bus 92 are located on feeders that present the minimum bus voltages, respectively, in

the current and future scenarios that will be presented later, while Bus 140 is the connection point of the largest hydroelectric unit, which significantly impact the voltage profile.

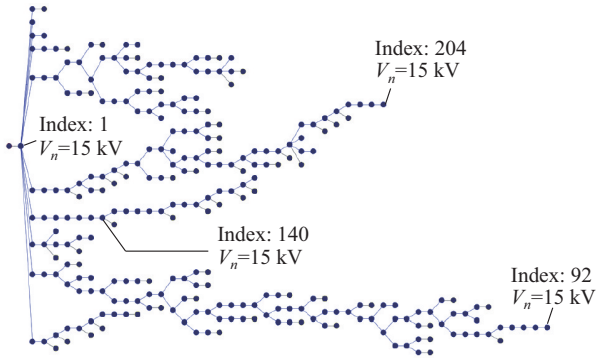

Fig. 4. Topology of considered distribution network.

Considering those four critical buses, the Q-table has three columns ($\Delta Tap=-1$, $\Delta Tap=+1$, and $\Delta Tap=0$) and a number of rows equal to 20736 (computed as $n_s=i^n=12^4=20736$), where voltage steps are 12 (interval of 0.01 p.u. from 0.95 p.u. to 1.05 p.u. plus ranges $v\leq 0.95$ p.u. and $v\geq 1.05$ p.u.. The number of critical buses n is 4.

Figure 5 shows the voltage magnitudes of all the network buses in the current scenario for two extreme cases, namely the cases of minimum demand and maximum demand.

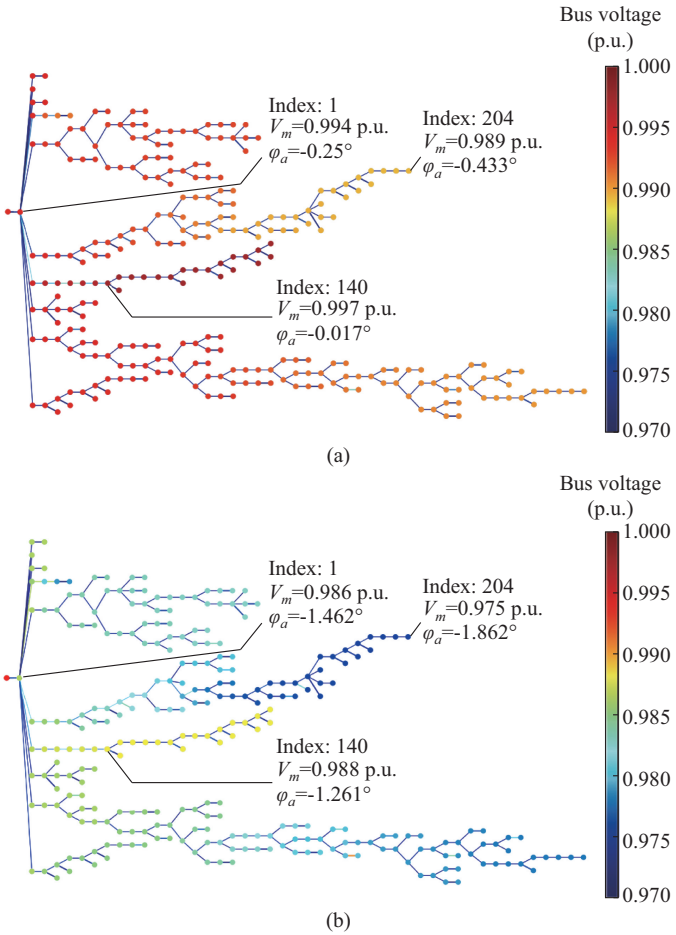

Fig. 5. Voltage magnitudes of all network buses in current scenario. (a) Minimum demand case. (b) Maximum demand case.

High-resolution measurements of load demand and DG production are unavailable, as still what usually happens in distribution networks. However, the yearly power profile of distribution network can be simulated by scaling the nominal values of each load and generator via standard curves. The key information used for domestic loads is the average annual power P_{avg}, whereas for DG units, it is the nominal apparent power A_{ref}. Load behavior is approximated through eight consumption patterns, both for active power and reactive power, related to season and day type (weekday or holiday). As an example, the load profiles assigned to domestic users to scale P_{avg} during a year are shown in Fig. 6. The generation pattern of DG units depends on the power plant technology: solar PV production profiles are determined by seasonal solar radiation, while for the sake of simplicity, run-of-the-river hydroelectric generators show a constant curve throughout the year. The generation profiles assigned to DG units to scale A_{ref} during a year are shown in Fig. 7. Finally, the yearly profile of HV voltage at the primary side of the substation transformer is built based on a dataset of measurements collected by Unareti. Voltage profiles represented in Fig. 8 have been determined for the same eight typical days defined for domestic users.

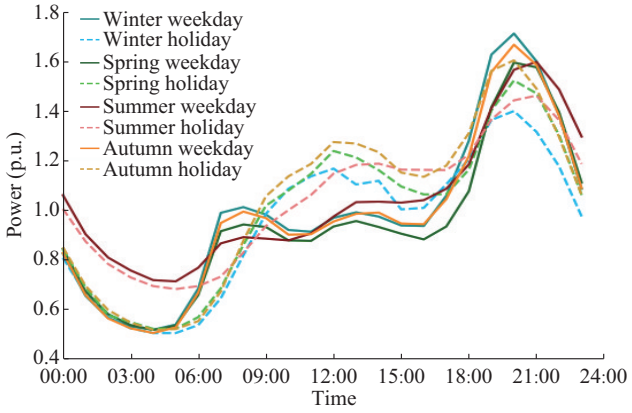

Fig. 6. Load profiles assigned to domestic users to scale P_{avg} during a year.

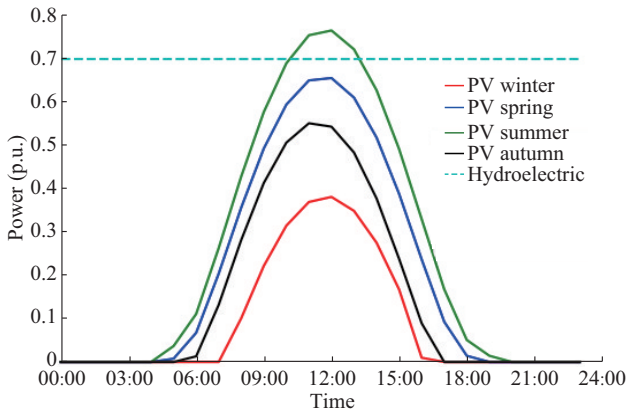

Fig. 7. Generation profiles assigned to DG units to scale A_{ref} during a year.

The effectiveness of the AVR is tested in two scenarios: the current scenario exploits the set of current values of P_{avg} and A_{ref} provided by Unareti, but these base values will be increased in the future scenario. The purpose is to model the

state of the distribution network in the future, where the increase of DGs and the electrification of consumption, e. g., electric vehicles and district heating, will cause an increase in power flows. The specifications of the test network in current and future scenarios is evidenced in Table II.

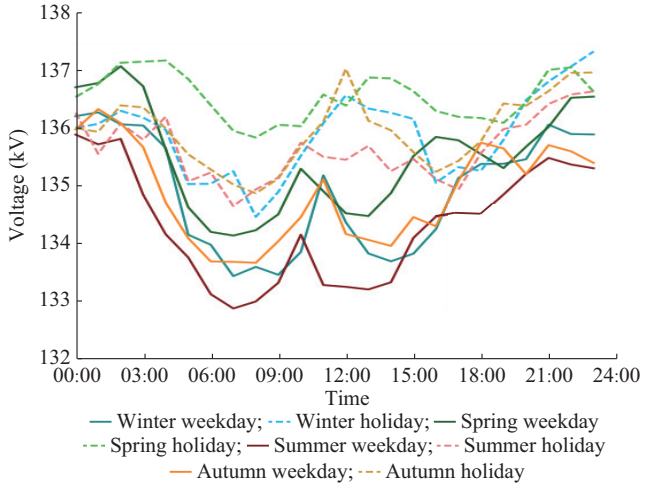

Fig. 8. Voltage profiles used to scale V_{nom} (132 kV) on primary side of OLTC during a year.

TABLE II
SPECIFICATIONS OF TEST NETWORK IN CURRENT AND FUTURE SCENARIOS

Scenario	PV A_{ref} (MVA)	Hydro A_{ref} (MVA)	Demand P_{avg} (MW)
Current	3.16	4.4	7.44
Future	17.47	4.4	15.33

As already mentioned, the training is performed using a dataset of 335 days (episodes), whereas a 30-day dataset addresses the test phase. The RL agent interacts with the environment 96 times per day (with quarter-an-hour resolution), and at the end of each episode, it receives a reward $R_{ep}=\sum_{t=1}^{96} r_t$.

To further increase the variability within the considered distribution network, random errors sampled from two normal distributions ($\varepsilon_{l,g}\sim N(0,\sigma_{l,g}^2)$ with $\sigma_{l,g}=0.1$ for loads and DG units, and $\varepsilon_{sb}\sim N(0,\sigma_{sb}^2)$ with $\sigma_{sb}=1/75$ for the slack bus voltage) are added to loads, DG units, and the slack bus voltage during each episode. The process used to build daily time series of load and generator injections during each episode is schematized, as given in Algorithm 2, whereas the voltage curves $v_{sb}\left[1,2,\ldots,\frac{24}{\Delta t}\right]$ of slack bus used for training the Q-learning agent are built following the steps of Algorithm 3.

It is not the best option to train the agent with daily data that are organized chronologically. On the one hand, during the first episodes, the RL agent would be unaware of the whole environment, and following an ϵ-greedy strategy searches for actions that can lead to the highest return. Q-values related to power system state typical of this first period could not yet be their best approximations because the agent would be more focused on trying multiple alternatives instead of consolidating knowledge of the optimal ones. On the other hand, an opposite problem could arise for scenarios occurring in the last months. In final stages, the model aims at exploiting information learnt and stored in the Q-table, hence the agent may not be aware of better actions for typical states of that period. Thus, days have been sorted in a random order during the training process to overcome these issues. The same procedure has been followed during the test phase but with a different goal: to assess model performances under as many cases as possible.

Algorithm 2: daily time series of load and generator injections during each episode

For loads, initialize the interpolated daily cluster profiles of active and reactive power consumption referred to the day processed during the episode; initialize $P_{avg,l}$ of each load l
For DG units, initialize the interpolated daily generation profiles of active and reactive power consumption referred to the day processed during the episode; initialize $A_{ref,g}$ of each generator g
Initialize $N(0,\sigma_l^2)$, $N(0,\sigma_g^2)$
for $l=1,2,\ldots,N_l$ **do**
  Draw a random sample $\varepsilon_{l,ep}$ from $N(0,\sigma_l^2)$
  Compute $P'_{avg,l}=P_{avg,l}+\varepsilon_{l,ep}$
  Retrieve the cluster assigned to l
  for $t=1,2,\ldots,\frac{24}{\Delta t}$ **do**
    Compute $P_{l,t}$ scaling time stamp t of active power consumption pattern by $P'_{avg,l}$
    Compute $Q_{l,t}$ scaling time stamp t of reactive power consumption pattern by $P'_{avg,l}$
  end for
end for
for $g=1,2,\ldots,N_g$ **do**
  Retrieve the technology of generator g
  if a unit with the technology of g has never been processed **then**
    Draw a random sample $\varepsilon_{tec,ep}$ from $N(0,\sigma_g^2)$
  else
    Take $\varepsilon_{tec,ep}$ received by the previous unit that shares the same technology of generator g
  end if
  Compute $A'_{ref,g}=A_{ref,g}+\varepsilon_{tec,ep}$
  for $t=1,2,\ldots,\frac{24}{\Delta t}$ **do**
    Compute $P_{g,t}$ scaling time stamp t of generation pattern by $A'_{ref,g}$
  end for
end for

Algorithm 3: voltage time series at slack bus during each episode

Initialize the previously built hourly voltage profiles of the HV busbar referred to the day processed during the episode; initialize the nominal voltage $V_{HV,n}$ of slack bus n
Initialize $N(0,\sigma_{sb}^2)$
for $t=1,2,\ldots,24$ **do**
  Draw a random sample $\varepsilon_{sb,t}$ from $N(0,\sigma_{sb}^2)$
  Convert $\varepsilon_{sb,t}$ to an actual voltage deviation $\varepsilon_{HV,sb,t}=\varepsilon_{sb,t}V_{HV,n}$
  Compute $v_{sb,t}$ adding $\varepsilon_{HV,sb,t}$ to voltage level at time stamp t of the standard HV curve
end for
Interpolate voltage values between t and $t+1$ to obtain a voltage curve with $\frac{24}{\Delta t}$ time tamps

As discussed above, the main criterion for evaluating the performance of Q-learning algorithm is to eliminate any voltage issue. The best algorithm setting will be the one maximizing the reward function (6) throughout $N_{test}=30$ test episodes. To solve the problem related to the number of OLTC operations associated with the upper and lower voltage limits, a sensitivity analysis of the parameter C_{TC} and voltage thresholds have been performed. In the current scenario, different settings of C_{TC}, $V_{\min}$, and $V_{\max}$ are shown in Table III.

TABLE III
DIFFERENT SETTINGS IN CURRENT SCENARIO

Voltage control system	Average of R_{ep} throughout 30 test episodes	Total average tap switches
$C_{TC}=5$, $V_{\min}=0.96$ p.u., $V_{\max}=1.04$ p.u.	−3.90	78
$C_{TC}=40$, $V_{\min}=0.97$ p.u., $V_{\max}=1.03$ p.u.	−5.65	113
Traditional	−6.65	133

Based on the sensitivity analysis performed, the optimal performances in the current scenario are obtained by training the RL agent with $C_{tc}=5$, $V_{\min}=0.96$ p.u., and $V_{\max}=1.04$ p.u.. The agent deals with input voltages discretized in twelve uniform voltage ranges from 0.95 p.u. to 1.05 p.u.. Performances achieved by data-driven AVR are compared with the traditional control strategy, which is currently adopted by Unareti. The OLTC is driven to keep the voltage at the secondary side of the HV/MV transformer within a fixed deadband centered around a target value. The deadband is equal to ±2.5% of the reference voltage of the AVR, and the target voltage has been set equal to $V_{2,rat}=15$ kV. Figure 9 compares the daily rewards throughout the test period.

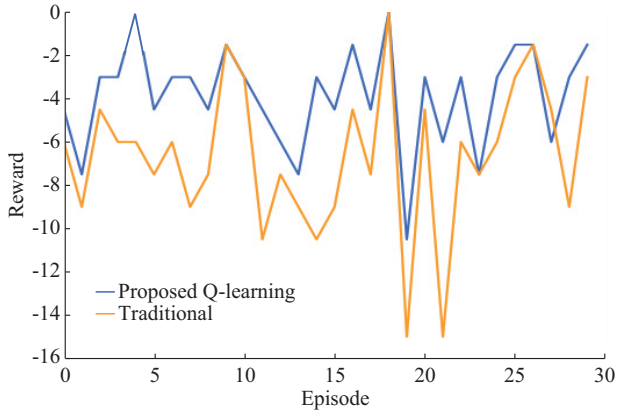

Fig. 9. Comparison of daily rewards throughout test period.

The proposed Q-learning algorithm performs better than the traditional control strategy. The average daily return of the RL AVR is $R_{ep}=-3.9$ against $R_{ep}=-6.65$ of the traditional control strategy adopted by Unareti. Since voltage issues are effectively eliminated with both the proposed Q-learning algorithm and the traditional control strategy, the reward is directly proportional to the number of tap switches during the test period, as given by (6). As shown in Fig. 10, the proposed Q-learning algorithm can reduce the OLTC operations. The total number of tap switches is 78 using the Q-learning algorithm against 133 with the traditional control strategy.

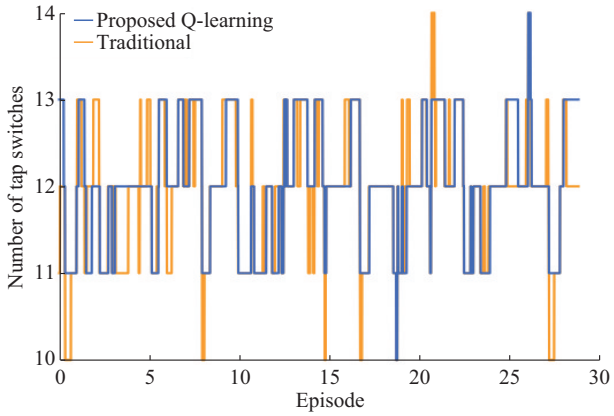

Fig. 10. Number of tap switches in current scenario using proposed Q-learning algorithm and traditional control strategy.

The situation will be different in the future scenario. The traditional control strategy fails to fulfill the voltage requirements, as shown in the boxplot of all bus voltages during the test phase, as displayed in Fig. 11. In fact, large demand leads to significant voltage drops along the distribution feeders.

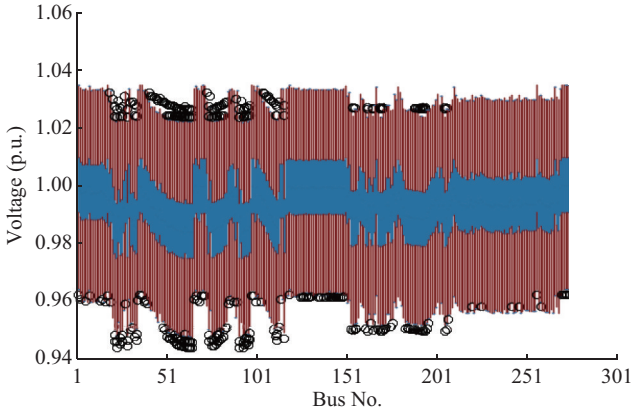

Fig. 11. Bus voltage boxplot using traditional control strategy in future scenario.

The data-driven AVR can instead handle the voltage issue in future scenario. Bus voltages are maintained within the admissible deadband, as shown in Fig. 12. In the future scenario, different settings of C_{TC}, $V_{\min}$, and $V_{\max}$ are shown in Table IV. The 2nd, 3rd, and 6th configurations bring voltage issues.

This time, the optimal performances are achieved by tuning training reward parameters as follows: $C_{tc}=2.5$, $V_{\min}=0.96$ p.u., and $V_{\max}=1.04$ p.u.. The agent is trained to learn through a lower penalty associated with tap movements since the main challenge is the possibility of performing an effective regulation within the system.

The key feature of the proposed Q-learning algorithm is its flexibility and that it is not limited by fixed schemes. Users can freely customize model parameters to meet their requirements. In distribution networks with tiny voltage drops on the secondary transformers and along LV distribution feeders, it is possible to expand the admissible voltage band to drastically reduce the OLTC operations, as shown in Fig. 13, where the agent has the goal of maintaining bus voltages within a deadband of $0.94\text{ p.u.} < V_{bus} < 1.06\text{ p.u.}$.

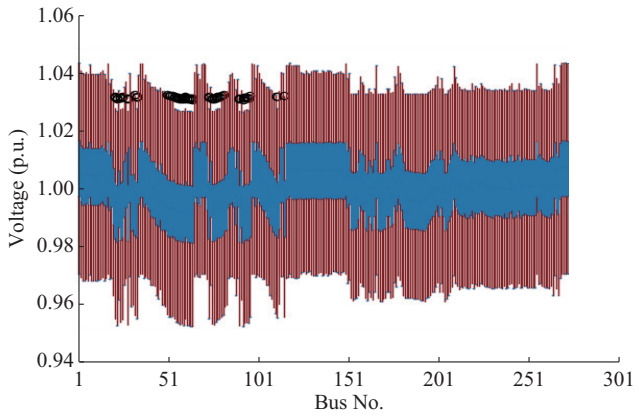

Fig. 12. Bus voltage boxplot using proposed Q-learning algorithm in future scenario.

TABLE IV
DIFFERENT SETTINGS IN FUTURE SCENARIO

Voltage control system	Average of R_{ep} throughout 30 test episodes	Total average tap switches
$C_{TC}=1.5$, $V_{min}=0.97$ p.u., $V_{max}=1.03$ p.u.	–27.60	552
$C_{TC}=5$, $V_{min}=0.97$ p.u., $V_{max}=1.03$ p.u.	–20.71	414
$C_{TC}=10$, $V_{min}=0.97$ p.u., $V_{max}=1.03$ p.u.	–16.51	321
$C_{TC}=1$, $V_{min}=0.96$ p.u., $V_{max}=1.04$ p.u.	–9.65	193
$C_{TC}=2.5$, $V_{min}=0.96$ p.u., $V_{max}=1.04$ p.u.	–9.35	187
$C_{TC}=5$, $V_{min}=0.96$ p.u., $V_{max}=1.04$ p.u.	–7.65	153

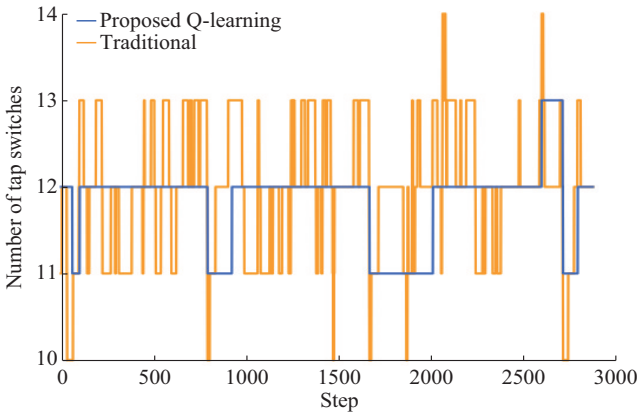

Fig. 13. Number of tap switches using proposed Q-learning algorithm compared with traditional control strategy adopted by Unareti.

The DSO can also update Q-values periodically, even when the Q-learning agent is installed as a real-time controller. This pattern can prove to be helpful both in network reconfigurations and in the cases of variations of power injections by DG units and withdrawn by loads. Moreover, input voltages used by the RL agent are not pre-defined. The DSO can carry out an analysis to assess whether the available measurement equipment is sufficient to train a highly efficient RL agent.

VI. CONCLUSION AND FUTURE WORK

This paper proposes a Q-learning algorithm for handling the voltage control problem in distribution networks with growing variability and uncertainty. The proposed Q-learning algorithm bypasses the need for accurate system models to regulate voltage effectively. The data-driven AVR determines the optimal tap-setting that eliminates voltage issues and minimizes unnecessary device operations based only on a few voltage measurements. Numerical simulations on a real MV network validate the effectiveness of the proposed Q-learning algorithm. In addition, users can freely customize model parameters to meet their requirements. Future developments of this paper could include introducing elements that have not been considered yet, such as network reconfigurations, reactive power injections by DG units, and compensation devices. Moreover, implementing deep RL techniques based on ANNs could help cope with the scalability issues of the Q-table and allow the development of a coordinated control system between the OLTC and the reactive power voltage control that synchronous and inverter-based DG units can perform.

REFERENCES

[1] R. Caldon, M. Coppa, R. Sgarbossa *et al.*, "A simplified algorithm for OLTC control in active distribution MV networks," in *Proceeding of 2013 AEIT Annual Conference*, Palermo, Italy, Oct. 2013, pp. 1-6.
[2] B. Robbins, H. Zhu, and A. Dominguez-Garcia, "Optimal tap setting of voltage regulation transformers in unbalanced distribution systems," *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 256-267, Jan. 2016.
[3] C. Murphy and A. Keane, "Optimised voltage control for distributed generation," in *Proceeding of 2015 IEEE Eindhoven PowerTech*, Eindhoven, Netherlands, Jun. 2015, pp. 1-6.
[4] J. F. Franco, L. F. Ochoa, and R. Romero, "AC OPF for smart distribution networks: an efficient and robust quadratic approach," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4613-4623, Sept. 2018.
[5] J. G. Vlachogiannis and N. D. Hatziargyriou, "Reinforcement learning for reactive power control," *IEEE Transactions on Power Systems*, vol. 19, no. 3, pp. 1317-1325, Aug. 2004.
[6] G. Custodio, L. F. Ochoa, F. C. L. Trindade *et al.*, "Using Q-learning for OLTC voltage regulation in PV-rich distribution networks," in *Proceedings of 2020 International Conference on Smart Grids and Energy Systems*, Perth, Australia, Nov. 2020, pp. 482-487.
[7] H. Liu, C. Zhang, and Q. Guo, "Data-driven robust voltage/var control using pv inverters in active distribution networks," in *Proceeding of 2020 International Conference on Smart Grids and Energy Systems*, Perth, Australia, Nov. 2020, pp. 314-319.
[8] R. Diao, Z. Wang, D. Shi *et al.*, "Autonomous voltage control for grid operation using deep reinforcement learning," in *Proceeding of 2019 IEEE PES General Meeting*, Atlanta, USA, Aug. 2019, pp. 1-5.
[9] J. Duan, D. Shi, R. Diao *et al.*, "Deep-reinforcement-learning-based autonomous voltage control for power grid operations," *IEEE Transactions on Power Systems*, vol. 35, no. 1, pp. 814-817, Jan. 2020.
[10] S. Wang, J. Duan, D. Shi *et al.*, "A data-driven multi-agent autonomous voltage control framework using deep reinforcement learning," *IEEE Transactions on Power Systems*, vol. 35, no. 6, pp. 4644-4654, Nov. 2020.
[11] D. Cao, J. Zhao, W. Hu *et al.*, "Model-free voltage control of active distribution system with PVs using surrogate model-based deep reinforcement learning," *Applied Energy*, vol. 306, p. 117982, Jan. 2022.
[12] H. Liu and W. Wu, "Two-stage deep reinforcement learning for inverter-based volt-var control in active distribution networks," *IEEE Transactions on Smart Grid*, vol. 12, no. 3, pp. 2037-2047, May 2021.
[13] D. Cao, J. Zhao, W. Hu *et al.*, "Attention enabled multi-agent DRL for decentralized volt-var control of active distribution system using PV inverters and SVCs," *IEEE Transactions on Sustainable Energy*, vol. 12, no. 3, pp. 1582-1592, Jul. 2021.
[14] H. Xu, A. D. Domínguez-García, and P. W. Sauer, "Optimal tap setting of voltage regulation transformers using batch reinforcement learning," *IEEE Transactions on Power Systems*, vol. 35, no. 3, pp. 1990-2001, May 2020.
[15] W. Wang, N. Yu, Y. Gao *et al.*, "Safe off-policy deep reinforcement

learning algorithm for volt-var control in power distribution systems," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3008-3018, Jul. 2020.
[16] J. F. Toubeau, B. B. Zad, M. Hupez *et al*., "Deep reinforcement learning-based voltage control to deal with model uncertainties in distribution networks," *Energies*, vol. 13, no. 15, p. 3928, Aug. 2020.
[17] Q. Yang, G. Wang, A. Sadeghi *et al*., "Two-timescale voltage control in distribution grids using deep reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2313-2323, May 2020.
[18] J. Zhang, Y. Li, Z. Wu *et al*., "Deep-reinforcement-learning-based two-timescale voltage control for distribution systems," *Energies*, vol. 14, no. 12, p. 3540, Jun. 2021.
[19] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction.* Cambridge: MIT Press, 2018.
[20] T. Simonini. (2020, Oct.). An introduction to deep reinforcement learning. [Online]. Available: https://thomassimonini. medium. com/an-introduction-to-deep-reinforcement-learning-17a565999c0c
[21] S. Kansal and B. Martin. (2018, Jan.) .Reinforcement Q-learning from scratch in Python with OpenAI Gym. [Online]. Available: https://www. learndatasci. com/tutorials/reinforcement-q-learning-scratch-python-openai-gym/
[22] R. Henry and D. Ernst, "Gym-ANM: reinforcement learning environments for active network management tasks in electricity distribution systems," *Energy and AI*, vol. 5, p. 100092, Sept. 2021.
[23] R. Henry and D. Ernst. (2024, Nov.). Gym-ANM. [Online]. Available: www.github.com/robinhenry/gym-anm

Alessandro Bosisio received the M.S. and the Ph.D. degrees in electrical engineering from Politecnico di Milano, Milano, Italy, in 2011 and 2015, respectively. He is currently an Assistant Professor with the University of Pavia, Pavia, Italy. His primary teaching activities range from the basics of power systems to smart grids and renewable energy sources. He is involved in several industrial projects in collaboration with distribution system operators (DSOs), transmission system operators (TSOs), and energy service companies. His main research interests include distribution network operation and planning, electric power system optimization, machine learning, and operation of distribution network and smart grid.

Francesca Soldan received the M.S. degree in physics from the Università degli Studi di Milano, Milano, Italy, in 2018. She is currently a Researcher at Ricerca sul Sistema Energetico (RSE S.p.A.) within the Information and Communications Technology for Energy Systems Group, where she focuses on the application of artificial intelligence algorithms to distribution networks. Her main research interests include use of semantic technologies and standards to improve interoperability across multi-energy system, as well as development of machine learning and graph computing techniques for network analysis.

Matteo Pisani received the master's degree in electrical engineering from Politecnico di Milano, Milano, Italy, in 2021. His research interests include development of grid-scale plants that incorporate renewable technologies, innovative solutions, and traditional system reimagined with a modern approach.

Enea Bionda graduated in Engineering of Computing Systems at Politecnico di Milano. Since 2024, he has been the Coordinator of the Working Group on Artificial Intelligence at Ricerca sul Sistema Energetico (RSE s.P.A.). Furthermore, he holds the position of technical manager for the Smart Grid Innovation Accelerator (SGIA) platform, a project developed under Mission Innovation (MI) – Innovation Challenge 1 on Smart Grids (IC1). His research interests include cloud/fog/edge computing, and artificial intelligence applied to energy sector.

Federico Belloni received the M.S. and Ph.D. degrees in physics from Università degli Studi di Milano, Milano, Italy, in 2003 and 2006, respectively. He is Head of Control Rooms in Unareti based in Milano and Brescia, Italy. His main research interests include real-time management of electric and gas networks of Unareti, and innovation and technological projects with impacts on real-time management of network.

Andrea Morotti received the M.S. degree in electrical engineering from Politecnico di Milano, Milano, Italy, in 2013. He is Head of Electricity Grid Development at Unareti, Milano, Italy. His main research interests include network planning, smart grid, distributed and flexible resource, storage, sustainability, network resiliency, protection and automation.