Decentralized Control in Active Distribution Grids via Supervised and Reinforcement Learning

Stavros Karagiannopoulos[a,b,*], Petros Aristidou[c], Gabriela Hug[b], Audun Botterud[a]

[a]*Laboratory for Information and Decision Systems (LIDS), Massachusetts Institute of Technology, Cambridge, USA*
[b]*EEH - Power Systems Laboratory, ETH Zurich, Physikstrasse 3, 8092 Zurich, Switzerland*
[c]*Department of Electrical Engineering, Computer Engineering & Informatics, Cyprus*

Abstract

While moving towards a low-carbon, sustainable electricity system, distribution networks are expected to host a large share of distributed generators, such as photovoltaic units and wind turbines. These inverter-based resources are intermittent, but also controllable, and are expected to amplify the role of distribution networks together with other distributed energy resources, such as storage systems and controllable loads. The available control methods for these resources are typically categorized based on the available communication network into centralized, distributed, and decentralized or local. Standard local schemes are typically inefficient, whereas centralized approaches show implementation and cost concerns. This paper focuses on optimized decentralized control of distributed generators via supervised and reinforcement learning. We present existing state-of-the-art decentralized control schemes based on supervised learning, propose a new reinforcement learning scheme based on deep deterministic policy gradient, and compare the behavior of both decentralized and centralized methods in terms of computational effort, scalability, privacy awareness, ability to consider constraints, and overall optimality. We evaluate the performance of the examined schemes on a benchmark European low voltage test system. The results show that both supervised learning and reinforcement learning schemes effectively mitigate the operational issues faced by the distribution network.

Nomenclature

A Action space of the environment in the Reinforcement Learning (RL) model.

*Corresponding author
Email addresses: stavroskarajan@gmail.com (Stavros Karagiannopoulos), petros.aristidou@cut.ac.cy (Petros Aristidou), hug@eeh.ee.ethz.ch (Gabriela Hug), audunb@mit.edu (Audun Botterud)

α'	Action of the RL method that maximizes the expected return in the next state s'.
α_t	Action of the RL method at time step (iteration) t.
$\alpha_{1,j,t}$	Active power action of DER at node j and time t.
$\alpha_{2,j,t}$	Reactive power action of DER at node j and time t.
$\beta j, k$	Regression coefficients for the j^{th} DER unit and k^{th} feature of the supervised learning method.
$\boldsymbol{\phi} j, k$	The k^{th} input measurement for the j^{th} inverter-based DER.
$\boldsymbol{\Phi} j$	Feature matrix of the supervised learning method for the j^{th} inverter-based DER.
$\boldsymbol{u}$	Vector of control variables in the centralized method.
Δt	Length of each time period in the centralized method.
η_μ	Learning rate for the actor network in the RL model.
$\eta_{\tilde{Q}}$	Learning rate for the critic network in the RL model.
γ	Discount factor of the RL method.
$\mathbb{E}(\cdot)$	Expectation operator used in the RL method.
$\mathbf{x}$	State vector representing bus voltage magnitudes and angles, except for the slack bus where the angle is set to 0 degrees and the magnitude is fixed.
$\mathbf{y}$	Constant parameter vector comprising the network topology, physical characteristics of the grid, and the thermal and voltage constraint limits.
$\mathcal{J}$	The set of inverter-based DERs in the distribution network.
$\mathcal{T}, \tau$	The set and number of time steps used in the optimization problems.
P	Transition function of the environment in the RL method.
μ	Actor network that maps the observed state into an action in the RL method.
μ'	Target actor network: a time-delayed copy of the actor network.
$\phi_{j,1,t}$	Net active power demand at node j and time t of the supervised learning method.
$\phi_{j,2,t}$	Local measured voltage at node j and time t of the supervised learning method.
$\phi_{j,3,t}$	Maximum active power capability of the inverter at node j and time t.

$\phi_{j,4,t}$	Feature combination of $\phi_{j,1,t}$ and $\phi_{j,2,t}$ at node j and time t of the supervised learning method.
π	Control policy of the RL method.
$\theta^{\mu'}$	Parameters of the target actor network of the RL method.
θ^{μ}	Parameters of the actor network of the RL method.
$\theta^{\tilde{Q}'}$	Parameters of the target critic network of the RL method.
$\theta^{\tilde{Q}}$	Parameters of the critic network of the RL method.
$\theta_{km,t}$	Voltage angle difference between buses k and m at time t.
θ_{slack}	Fixed reference slack bus voltage angle.
$\tilde{p}j,t$	Active power injection at node j and time t according to the supervised learning method.
$\tilde{Q}$	Critic network that uses the state-action pair to calculate the action value (Q-value).
$\tilde{Q}^*(\cdot)$	Optimal action value function (Bellman equation) of the RL method.
$\tilde{Q}^{\pi}(\cdot)$	Action value function (Q-function of the RL method) following policy π.
$\tilde{Q}'$	Target critic network: a time-delayed copy of the critic network.
$\tilde{V}^{\pi}(\cdot)$	Value function of the RL method.
$c(\cdot)$	General objective function representation of the centralized method.
c_1	Cost parameter of the RL method to penalize the local voltage constraint.
c_2	Cost parameter of the RL method to penalize the case where the apparent inverter power is violated.
C_P	Cost parameter to penalize curtailing active power.
C_Q	Cost parameter to penalize reactive power control.
$cos(\phi_{\max})$	Power factor corresponding to the acceptable inverter operational mode.
$f(\cdot)$	Power flow equations enforcing active and reactive power balances at each node.
$g_{\mathrm{DER}}(\cdot)$	DER technical inequality constraints and regulatory limitations in the centralized method.
$h_I(\cdot)$	Current (thermal) constraints referring to acceptable current magnitudes.

$h_V(\cdot)$	Voltage constraints referring to acceptable voltage magnitudes.
$h_{\text{DER}}(\cdot)$	DER technical equality constraints of the centralized method.
$I_{br,i,t}$	Current flowing at branch i and time t.
$I_{i,max}$	Maximum thermal limit at branch i.
N_b	Total number of network nodes in the system.
N_K	The number of features in the supervised learning method.
N_k	Random noise following the Ornstein-Uhlenbeck process in the RL method.
N_{hor}	Time horizon of the optimization problem in the centralized method.
N_{OPF}	The number of optimal setpoints obtained in the centralized method.
$P_{curt,j,t}$	Curtailed power of the unit connected at node j and time t.
$P_{g,j,t}^{\min}$	Lower limit for active power injection at node j at time t.
$P_{g,j,t}$	Active power injection of the unit connected at node j and time t.
$P_{g,j,t}^{max}$	Maximum available active power of the unit connected at node j and time t.
$P_{inj,j,t}$	Total power injection at node j and time step t.
$P_{l,j,t}$	Active power load at node j and time step t.
$Q_{ctrl,j,t}$	Reactive power output of unit connected at node j and time t.
$Q_{g,j,t}^{\max}$	Upper limit for reactive power of unit connected at node j and time t.
$Q_{g,j,t}^{\min}$	Lower limit for reactive power of unit connected at node j and time t.
$Q_{g,j,t}$	Reactive power injection/absorption of unit at node j and time t.
$Q_{inj,j,t}$	Total reactive power injection/absorption at node j and time t.
$Q_{l,j,t}$	Reactive power load at node j and time step t.
$R(\cdot)$	Reward function of the RL method.
$r(s,\alpha)$	Immediate reward of the RL method received for taking action α in state s
S	State space representation of the environment in the RL method. Current state denoted as s and next state as s'.
s_t	State of the RL method at time step t.

$S^{\max}_{\text{inv,j}}$	Capacity of the unit's inverter located at node j.
$S^{max}_{g,j,t}$	Maximum apparent power capability of unit at node j and time t.
t	Time index.
$V_{\min}/V_{\max}$	Minimum/Maximum acceptable voltage magnitude in the distribution grid.
$V_{j,t}$	Voltage magnitude at bus j at time t.
V_{slack}	Fixed reference bus voltage magnitude.
Y_{km}	Nodal admittance matrix of the distribution grid.

Acronyms

AI	Artificial Intelligence
BESS	Battery Energy Storage System
BaU	Bussiness as Usual
DDPG	Deep Deterministic Policy Gradient
DER	Distributed Energy Resource
DG	Distributed Generator
DN	Distribution Network
LV	Low Voltage
MAE	Mean Absolute Error
MSE	Mean Squared Error
ML	Machine Learning
NN	Neural Network
OPF	Optimal Power Flow
RL	Reinforcement Learning
RMSE	Root Mean Squared Error
PV	Photovoltaic

1. Introduction

1.1. Motivation & Background

Over the last years, Distribution Networks (DNs) have become more and more observable and controllable, due to the widespread installation of smart metering devices with control capabilities, and the deployment of numerous flexible Distributed Energy Resources (DERs) [1]. The high number of Distributed Generators (DGs), such as photovoltaic (PV) units or wind turbines, along with other DERs, such as electric vehicles, Battery Energy Storage Systems (BESSs), and flexible loads, are elevating the role of DN operators, but are also imposing substantial challenges to the DN operation [2]. Modern grids needs to operate *safely* under higher complexity, as they are not just sinks of power anymore, and increased uncertainty due to the intermittent nature of renewable-based DGs that cannot be predicted perfectly and are volatile. Therefore, controlling these resources is of crucial importance to guarantee safe and stable grid operation.

In terms of controlling approaches, the recent advances in computer science and computational power have fostered research on machine learning (ML) and artificial intelligence (AI). These fields offer a large variety of suitable methods for power systems algorithms that can make use of historical data and real-time measurements to learn parameters and design sophisticated control functions. Traditional fit-and-forget control approaches of (mostly) overdimensioned grids can now be substituted by more sophisticated mathematical approaches based on ML which, with the additional support of the communication links, integrate DERs safely into the existing infrastructure and optimize the operation of the grid. Operational schemes are categorized as centralized, distributed, and decentralized (or local) based on the communication infrastructure available to govern the DERs. In the rest of this section, we summarize the available methods and provide a literature review on the various control approaches. Since most of our previous work focused on centralized OPF-based control and supervised learning methods to control DERs in distribution networks, e.g., [3–6], we only provide a summary of these methods. Then, we focus on the contributions of this paper performing a detailed state-of-the-art literature review on the RL-based methods for voltage control. Finally, we distinguish our work against the revised literature and summarize our contributions.

1.2. Related Works

1.2.1. Centralized OPF-based methods

Centralized methods need an extensive monitoring and communication infrastructure, and they often rely on sophisticated optimization-based control approaches. Their benefit stems from the fact that they allow for system-wide optimal functioning through coordinated control of DERs [7]. Because of recent advances in computational power, wireless communication, and new theoretical discoveries in handling the nonlinear AC power flow equations, [8, 9], centralized control has gained a lot of attention. Nevertheless, the infrastructure necessary for this form of control is rarely present in DNs, and the financial advantage of investing in such capabilities remains debatable.

1.2.2. Conventional decentralized and distributed methods

Decentralized control techniques, e.g., [10], use local measurements to address power quality and security issues. These sorts of controls have been integrated into a number of grid codes, and they are presently the most commonly used approach in DNs. The simplicity and low implementation costs of these systems are their key advantages. Because no communication infrastructure is required, the required investment is kept to a minimum. Decentralized approaches, however, normally take a one-size-fits-all approach, with the same control settings used in all DNs, regardless of generator type or operating conditions. In a fast-changing environment, such an approach might lead to unanticipated challenges, such as stability issues [3, 4]. Finally, distributed techniques, e.g., [11, 12], rely on minimal communication among DERs to coordinate and produce near-optimal results. For a detailed survey of the latest literature on distributed algorithms on the areas of optimization and control of power systems we refer the interested reader to [12] where distributed algorithms for optimal power flow (OPF) problems are compared. While distributed approaches attempt to bridge the gap between local and centralized systems, they still need some communication infrastructure and often use consensus-based control algorithms that are susceptible to communication delays and errors.

1.2.3. Decentralized methods based on supervised learning

Lately, data-driven methods based on machine learning have attracted a lot of attention in the power systems area. In terms of *supervised* learning, most works use regression methods to represent the optimal behavior obtained by offline optimal power flow calculations [3, 5, 6, 13–16]. In [3, 5], we perform three-phase optimal power flow calculations based on historical data to design local controls based on segmented regression and support vector machines that emulate the optimal behavior without the use of any communication. The proposed approach in [13, 16] uses non-linear control policies to calculate the real-time reactive power injections of inverter-based DGs. It uses linearized grid modeling, assumes balanced operation, and focuses on reactive power control, exploiting the flexibility of various kernel functions to model complex and non-linear behaviors. In [14] and [15], multiple linear regression models are used in an open-loop fashion to calculate a function for each inverter that maps its local historical data to pre-calculated optimal reactive power injections. These works focus on reactive power control, not considering possible combinations with other available controls, and [15] assumes a balanced DN, i.e. using a single-phase representation. However, these approaches work well with legacy equipment and do not require a stability analysis like [5] which is addressed in [4]. The interested reader is referred to [17] for a review on learning to control power systems, with an emphasis on guidelines for concrete safety problems.

1.2.4. Decentralized methods based on reinforcement learning

Finally, in terms of (deep) *Reinforcement Learning* (RL), we restrict ourselves to works related to power systems that are in normal operating state, i.e., the control policies are learned and applied under normal operating conditions. Such works are related to frequency

regulation, e.g., [18–20], demand response, e.g., [21, 22], and the main application of this paper, voltage control, e.g., [23–31].

In [23], a two-timescale approach is used to regulate voltages in distribution systems: on the faster timescale, a voltage tracking problem decides on the DER setpoints, while deep RL is used on the slower timescale to control capacitor banks for long-term voltage stability. The control is based on switching on/off the capacitors to regulate voltage. In [24], voltage control is achieved by deciding on the generator voltage setpoints. Deep Q-network and Deep Deterministic Policy Gradient (DDPG) are used to perform voltage control, the latter of which showing better long-term performance due to a larger range of exploration. A similar RL algorithm is proposed in [26] where the authors reduce the system's losses by controlling the reactive power of smart transformers. However, a data-driven network model is used to create the training data for the RL algorithm without interactions with the actual distribution grid.

Reference [27] proposes a voltage sensitivity based DDPG method to compute analytically the gradient of the value function instead of using the critic neural network. The control relies also on reactive power only, but considers a multi-agent approach. In [29], the authors propose a two timescale hybrid voltage control strategy based on mixed-integer optimization and multi-agent reinforcement learning. The focus of the fast timescale is to mitigate short-term voltage fluctuations by reactive power control of smart PV inverters and active power of electric vehicles. Reference [28] is based also on two different DERs for control; it uses reactive power from PV and active power from a BESS to mitigate voltage issues. Another RL-based method that controls the tap position of voltage regulating transformers and capacitor banks is presented in [30], whereas the authors of [31] utilize active power curtailment of PV units and reactive power control of static var compensators. In both references, a single-agent approach is used in a centralized manner, i.e., the RL agent requires full knowledge of the environment.

Many of the revised papers, i.e., [23, 24, 26] apply an RL algorith to one control measure to regulate voltages. The references that consider multiple measures, i.e., [28–31], rely on separate DERs for each offered control measure. In our paper, we focus on decentralized control of a single agent to optimally regulate voltage, without need for communication network. In contrast to the revised literature, we use both active power curtailment and reactive power control from the same inverter-based DG. We perform a steady-state analysis and we consider the inverter's capability curve to decide on the reactive and active power control setpoints.

Finally, we refer the interested reader to two review papers; an extensive review in [32], summarizing the use of RL in power system control with respect to normal, emergency, and restorative control applications; and Ref. [33] providing a comprehensive review on recent RL-based methods to control voltages in power systems, where the different methods are compared in terms of the used environment, state space and action space representation, reward function, constraints, and challenges.

1.3. Contributions

In this paper, we compare state-of-the-art decentralized control approaches in active distribution grids using AI-based methods. We consider reactive power control and active power curtailment from the same inverter-based DG and the decentralized schemes are derived by supervised learning and deep RL methods. More specifically, the contributions of this paper can be summarized as follows:

- The proposal of a deep reinforcement learning algorithm for voltage control in active distribution grids based on deep deterministic policy gradient. The single agent represents any inverter-based distributed energy resource and controls both active and reactive power using the inverter's capability curve.
- A quantitative comparison of the state-of-the-art approaches to control distributed energy resources in active distribution grids, i.e., based on purely local control laws, optimal power flows, supervised learning and reinforcement learning, using error metrics.
- A qualitative evaluation of the different options/methods to control DERs, including optimality, constraint consideration, needed training effort, safety, capability to adapt to changes, privacy and scalability that can assist decisions of operators and researchers.

The remainder of the paper is organized as follows: In Section 2, we present the mathematical formulation of the decentralized DER controllers using a) the supervised learning algorithms based on offline optimal DER setpoints, and b) RL-based schemes. Then, in Section 3, we introduce the case study and simulation results that show the performance of the optimized controllers according to the different methods. Finally, we draw conclusions in Section 4.

2. Supervised Learning and RL-based methods

2.1. Supervised Learning based on off-line data

Supervised learning requires the offline computation of many optimal DER setpoints that represent different operating conditions. These can be obtained via OPF calculations, under specific objectives, such as system losses minimization or reference voltage tracking. Since the focus of this paper does not lie on OPF formulations, a generic high-level description is provided here. For a full three-phase OPF formulation in distribution grids, we refer the interested reader to [34]. System safety and power quality considerations can be incorporated by including appropriate constraints in the optimization problem.

Offline OPF. An optimal power flow problem calculates the most efficient settings of the power system control variables to minimize the total cost of generating and transmitting electricity while satisfying the operational constraints. As inputs it uses the system's topology, its parameters, load demand and the generators' and lines' operational constraints that

need to be satisfied. The outputs of an OPF problem are the controllable units' setpoints and the objective function value that is optimized. We refer the interested reader to [35] for a detailed analysis of the power flow equations and their use in an optimization setup. Formally, the OPF problem is represented as

$$\min_{\mathbf{u}} \; c(\mathbf{x}, \mathbf{u}) \tag{1a}$$

$$\text{s.t.} \quad f(\mathbf{x}, \mathbf{u}, \mathbf{y}) = 0, \tag{1b}$$

$$h_V(\mathbf{x}, \mathbf{u}, \mathbf{y}) \leq 0, \tag{1c}$$

$$h_I(\mathbf{x}, \mathbf{u}, \mathbf{y}) \leq 0, \tag{1d}$$

$$h_{\text{DER}}(\mathbf{x}, \mathbf{u}, \mathbf{y}) \leq 0, \tag{1e}$$

$$g_{\text{DER}}(\mathbf{x}, \mathbf{u}, \mathbf{y}) = 0. \tag{1f}$$

The control vector $\mathbf{u}$ represents the controllable entities, e.g. the DER active and reactive power setpoints; the state vector $\mathbf{x}$ refers to the bus voltage magnitudes and angles (except for the slack bus, where the angle is set to 0 degrees and the magnitude is fixed); the constant parameter vector $\mathbf{y}$ comprises the network topology, physical characteristics of the grid, and the thermal and voltage constraint limits; finally, the function $c(\mathbf{x}, \mathbf{u})$ represents the various objectives, including the operational cost of the used measures, electricity cost/revenue by exchanging energy with the upper voltage networks, revenues from ancillary service provision, etc. The power flow equations are represented by equation (1b) which enforces active and reactive power balances at each node. Constraints (1c) - (1d) correspond to power quality constraints, referring to acceptable voltage and current magnitudes. Finally, the DER models and constraints are incorporated via (1e) - (1f) which model the DER technical and regulatory limitations, such as curtailing PV power, setting limits for the battery state of charge, updating the BESS energy capacity at each time, etc. The simulation outcomes of many operational conditions are provided to the data-driven design stage, where the obtained optimal data are used to design local DER controls for real-time DN operation based on ML techniques.

Data-driven Control Design. Figure 1 summarizes the procedure to generate the optimized local DER control models used in the real-time decision making. As input data, each DER considers only local information, e.g., local solar radiation, active and reactive demand, voltage magnitude, maximum active and reactive power capability of the inverter, the local time, and interaction terms among these. The scope of the supervised ML approach is to create a model that given these local inputs, will respond similarly to the OPF behaviour it is trained from.

In order to design robust ML-based control schemes, the training dataset should consider both expected as well as unexpected operational conditions. It should consider the seasonal variation of load and generation, the various valid network configurations and intermittent renewable-based energy resources. For instance, when exploring the summer season, one needs to consider both sunny and cloudy days in the training dataset, typical consumption patterns, and all possible DN topologies. It is important to note that all the considered

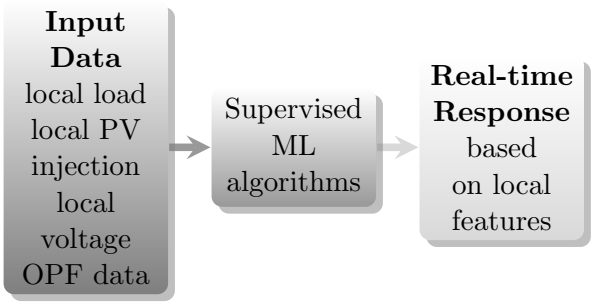

Figure 1: Data-driven control design based on supervised learning.

conditions do not need to be present in the historic realized behavior. Instead, they can be artificial, i.e., simulated, to be used as inputs in the design of the ML-based schemes and lead to robust and safe data-driven behavior.

There are various regression and classification ML algorithms, such as segmented and multiple regression, support vector machines, and decision trees, that can be used according to the required complexity and accuracy in terms of mimicking the offline OPF response. The interested reader is referred to [5] for a detailed discussion of such models. Although the mathematical models differ from each other, they all try to map the observed OPF behaviour as close as possible using local features. The real-time response of the j^{th} inverter-based DER ($j \in \mathcal{J}$) in terms of active power control $p_{j,t}$ is derived from the N_{OPF} optimal setpoints ($t \in \mathcal{T}$) obtained in the offline calculations, and the final rules depend only on local features. The feature matrix $\mathbf{\Phi}_j \in \mathbf{R}^{N_{OPF} \times N_K}$ contains as columns the N_K features and as rows the N_{OPF} observations of the k^{th} input measurement $\boldsymbol{\phi}_{j,k} \in \mathbf{R}^{N_K}$, i.e. $\mathbf{\Phi}_j = [\boldsymbol{\phi}_{j,1}, \boldsymbol{\phi}_{j,2}, \ldots, \boldsymbol{\phi}_{j,N_K}]^T$.

2.2. Deep RL-based methods

In this section, we explain the main principles of deep RL, and we present in detail the mathematical model of the specific algorithm applied in this paper.

2.2.1. Basics of (deep) RL methods

Reinforcement learning differs from supervised learning in the sense that the focus is not on labelled input and output data, but rather on trying out different control actions and evaluating them according to a reward system. Thus, a balance is sought between exploration of uncharted territory and exploitation of gained knowledge based on the observed states and rewards. The agents interact with the environment, which is typically modelled as a (partially observable) Markov decision process following the principles of dynamic programming.

The environment of our RL model is defined by: a (continuous or discrete) state space S; a (continuous or discrete) action space A; an environment transition function $\mathrm{P} : S \times \mathrm{A} \to S$; a reward function $R : S \times \mathrm{A} \to R$; and a discount factor $\gamma \in [0, 1]$. The main principle is summarized as follows; At each time step t, the agent (e.g., a DER unit) observes the state $s_t \in S$, performs an action $\alpha_t \in \mathrm{A}$ following a control policy π (e.g., PV curtailment, BESS charging or discharging, injecting/absorbing reactive power) that alters the environment,

and receives the corresponding reward signals $r \in R$ (e.g., revenues from injecting power, penalties for causing voltage issues). The goal is to learn and apply the optimal action based on the current state in order to maximize the accumulated reward over time. This is given by $R(t) = \sum_{t=0}^{T} \gamma^t r_t$.

Let us define the value function $\tilde{V}^\pi : S \to \mathbb{R}$:

$$\tilde{V}^\pi(s) = \mathbb{E}[\sum_{t=0}^{T} \gamma^t r_t | S_0 = s], \, \forall s \in S \tag{2}$$

and the action value function (Q-function) $\tilde{Q}^\pi : S \times A \to \mathbb{R}$:

$$\tilde{Q}^\pi(s, \alpha) = \mathbb{E}[\sum_{t=0}^{T} \gamma^t r_t | S_0 = s, A_0 = \alpha], \, \forall s \in S, \forall \alpha \in \mathrm{A} \tag{3}$$

which models the expected discounted return when the agent takes the action α in state s and then follows the policy π.

The Q-function is updated by the Bellman equation, i.e., an iterative algorithm given by

$$\tilde{Q}^*(s, \alpha) = \mathbb{E}[r(s, \alpha) + \gamma \max_{a' \in \mathrm{A}} \tilde{Q}^*(s', \alpha')]. \tag{4}$$

where α' refers to the action that maximizes the expected return in the next state s'. This algorithm will converge to the optimal solution $\tilde{Q}^*(s, \alpha)$ as $t \to \infty$ as long as the state signals fulfil the Markov property.

The traditional Q-learning requires discretization of the observation space, since it relies on tabular methods. This introduces computational concerns when the dimensions are large, leading to memory issues and prolonged training stages. Furthermore, it is sometimes difficult to discretize realistic environments which include continuous variables. For these reasons, deep RL models have gained a lot of attention lately, due to their efficiency in approximating the policies and Q-functions. They leverage the historic agent-environment interactions to extract the optimal policies, which may be based on simulated or real events. Typically, deep RL models use Neural Networks (NN) to estimate Q-values. Their efficiency and stability are increased by using a targeted Q-network apart from the current Q-network, and by using experience replay where samples of the agent's experience in terms of actions and rewards, i.e., mini-batches, are stored and used to train the Q-network.

The available deep RL models can be categorized into model-free and model-based algorithms. The former are easier to tune and implement, but their efficiency depends on the samples. The latter are constructed using a physical/mathematical model but are more difficult to formulate. We explored several (deep) RL models and algorithms that can be found in the literature, such as asynchronous advantage actor-critic, proximal policy optimization, trust region policy optimization, Deep Deterministic Policy Gradient (DDPG), and twin delayed DDPG [36]. After considering the overall performance in terms of average reward per episode and convergence stability for our use case, we observed that the DDPG showed the best behavior with the default hyperparameters, and thus, in the rest of the

paper we focus on the DDPG algorithm. Comparing different RL methods is outside the scope of this paper, since one would need to compare the mathematical models, the various parameter sets of each method and convergence characteristics. The selected method is a model-free, off-policy, and actor-critic deep RL algorithm that includes continuous state and action spaces.

2.2.2. Deep deterministic policy gradient

In this section, we present the RL method, based on DDPG, to control DERs in active distribution grids. The selected method follows an 'actor-critic' architecture, which involves two neural networks. The actor network is responsible for learning the optimal control policy π that maps the state of the distribution network to the corresponding control actions of the DERs, while the critic network estimates the Q-function (4), i.e., the expected cumulative reward obtained from following the policy. It is 'model-free', i.e., it does not require any prior knowledge of the distribution network or the dynamics of the DERs, and is an 'off-policy' method, i.e., the agents learn from data generated by any policy, not the optimal policy necessarily. DDPG algorithms can work with continuous control variables, such as PV curtailment and reactive power control actions, as considered in this paper.

The procedure and the main principles of the DDPG algorithm of the networks are given in Fig. 2. First, the actor network, denoted as $\mu(s|\theta^{\mu})$ and shown as block 1 in Fig. 2 maps the observed state into an action α, and then the critic network denoted as $\tilde{Q}(s, \alpha|\theta^{\tilde{Q}})$ and shown as block 2 in Fig. 2, uses the state-action pair to calculate the action value (Q-value). The parameters of the neural networks are represented by θ^{μ} for the actor and $\theta^{\tilde{Q}}$ for the critic network. To foster exploration and avoid getting stuck in local solutions, the random

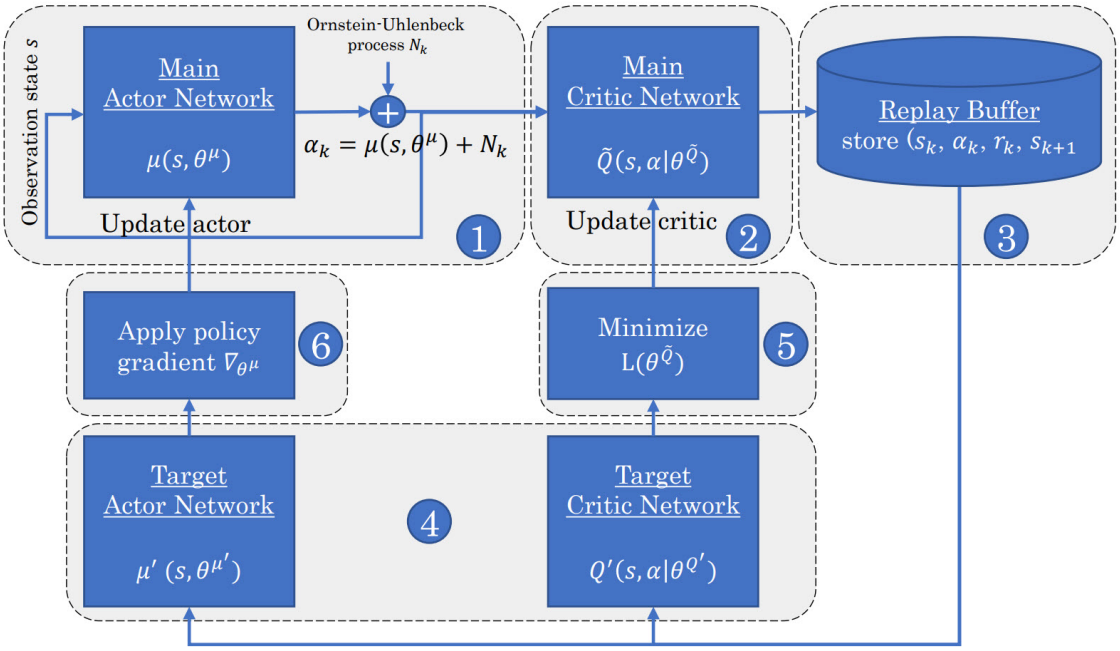

Figure 2: High-level flowchart overview of the DDPG algorithm.

noise N_k following the Ornstein-Uhlenbeck process [37] is added to the policy. In each iteration, denoted by the index k, the tuple $(s_k, \alpha_k, r_k, s_{k+1})$ is stored in the replay buffer B shown as block 3 in Fig. 2.

A time-delayed copy of these two networks is also used to improve the training stability. This is achieved by introducing the so-called target networks that are denoted by $\tilde{Q}'(s, \alpha|\theta^{\tilde{Q}'})$ and $\mu'(s|\theta^{\mu'})$, and shown as block 4 in Fig. 2; their parameters $\theta^{\tilde{Q}'}$ and $\theta^{\mu'}$ track smoothly the main networks by

$$\theta^{\tilde{Q}'} \leftarrow \tau\theta^{\tilde{Q}} + (1-\tau)\theta^{\tilde{Q}'}, \qquad \theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'}. \tag{5}$$

From the replay buffer B that serves as a data-pool, N samples are derived to train the neural networks. More specifically, to first train the target critic network, we calculate for each sample i the sum of the immediate reward and the expected Q-function value of the next state, given by

$$y_i = r_i + \gamma\tilde{Q}'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{\tilde{Q}'}). \tag{6}$$

Then, as shown in block 5 of Fig. 2, the critic network is updated by minimizing the averaged loss function between the estimated reward calculated in (6) and the expected reward of the derived N samples from the critic network, given by

$$L(\theta^{\tilde{Q}}) = \frac{1}{N}\sum_{i}^{N}(y_i - \tilde{Q}(s_i, \alpha_i|\theta^{\tilde{Q}}))^2. \tag{7}$$

Thus, the parameters of the critic network are updated by

$$\theta^{\tilde{Q}} \leftarrow \theta^{\tilde{Q}} - \eta_{\tilde{Q}}\nabla_{\theta^{\tilde{Q}}}L(\theta^{\tilde{Q}}), \tag{8}$$

where $\eta_{\tilde{Q}}$ is the learning rate for the critic network and $\nabla_{\theta^{\tilde{Q}}}L(\theta^{\tilde{Q}})$ is the gradient of the averaged loss function (7) with respect to $\theta^{\tilde{Q}}$.

Finally, the actor network is updated via a gradient ascent approach shown in block 6 of Fig. 2 that maximizes the average policy performance of the used N samples, given by

$$\nabla_{\theta^{\mu}}J(\theta^{\mu}) \approx \frac{1}{N}\sum_{i}^{N}\nabla_{\alpha}\tilde{Q}(s, \alpha|\theta^{\tilde{Q}}))|_{s=s_i, \alpha=\mu(s_i)}\nabla_{\theta^{\mu}}\mu(s|\theta^{\mu})|_{s_i}. \tag{9}$$

The parameters of the actor network are updated by

$$\theta^{\mu} \leftarrow \theta^{\mu} + \eta_{\mu}\nabla_{\theta^{\mu}}J(\theta^{\mu}), \tag{10}$$

where η_{μ} is the learning rate for the actor network and $\nabla_{\theta^{\mu}}J(\theta^{\mu})$ is the gradient of the average policy performance with respect to θ^{μ}.

The update equations (8) and (10) represent the gradient steps taken to update the neural network parameters. A descent step is used for the critic network to minimize the loss function, and an ascent step for the actor network to maximize the policy performance. By moving in the negative direction of the gradient of the loss function, the critic neural network learns to minimize the mismatch between the estimated and expected rewards. Similarly, by taking steps in the positive direction of the gradient of the average policy performance, the actor network learns to take better actions, improving the policy. Both learning rates, i.e., $\eta_{\tilde{Q}}$ and η_{μ}, influence the convergence and stability of the learning processes, by controlling the size of the update steps. They assess the trade-off between exploration and exploitation and the careful tuning of them affects the overall DDPG performance and avoids oscillations.

Algorithm 1 presents the procedure of training a single DER agent based on the DDPG model. We define an *episode* as one day with 24 hourly steps, i.e. $T = 24$. In total, M episodes are considered in the training stage, corresponding to different solar radiation and loading data that represent different operating conditions. Hence, the power flow calculations result in different solutions, i.e. complex voltages, that trigger different actions from the DER-agent.

First, all the networks and memory buffer are initialized. Then, for each episode, at the beginning of each hour, the agent senses the *environment*, which is represented by measuring the local voltage and solar radiation and performs *actions*, such as the curtailment of active power and reactive power control. To decide on the actions, rewards are received based on the electricity exchange and local constraints' satisfaction. Subsequently, a power flow calculation is performed, the system state is updated, and the reward signals are sent to the DER agent anew. All the simulated cases are stored in the memory buffer used to improve the parameters of the actor and critic networks, following the last algorithm steps.

3. Case Study - Results

3.1. Network Description and Input Data

In order to compare the different decentralized control methods, we use a typical European radial LV grid [38], sketched in Fig. 3. The installed PV capacity is expressed as a

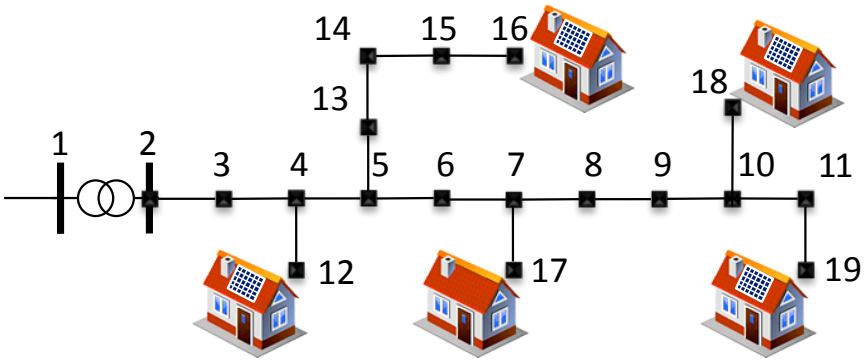

Figure 3: Benchmark European LV grid [38].

Algorithm 1 DER-agent training based on the DDPG-model.

Randomly initialize actor network $\mu(s|\theta^{\mu})$ and critic network $\tilde{Q}(s, \alpha|\theta^{\tilde{Q}})$ with parameters $\theta^{\tilde{Q}}$ and θ^{μ}, respectively.
Initialize target networks $\mu'(s|\theta^{\mu'})$ and $\tilde{Q}'(s, \alpha|\theta^{\tilde{Q}'})$, with main networks' parameters, i.e., $\theta^{\mu'} \leftarrow \theta^{\mu}$ and $\theta^{\tilde{Q}'} \leftarrow \theta^{\tilde{Q}}$.
Initialize the experience replay buffer B.
for $episode = 1, ..., M$ **do**
    Initialize random process N_k.
    Initialize observation state s_1.
    for $t = 1, ..., T$ **do**
        Perform action $\alpha_k = \mu(s|\theta^{\mu}) + N_k$, run power flow calculations and observe reward r_k and new state s_{k+1} of the system.
        Store tuple $(s_k, \alpha_k, r_k, s_{k+1})$ in memory buffer B.
        Randomly sample N tuples from B.
        Calculate (6) and update critic network by minimizing (7) (one step of gradient descent).
        Update the actor policy by the sampled policy gradient of (9) (one step of gradient ascent).
        Update the parameters of the target networks using (5).
    end for
end for

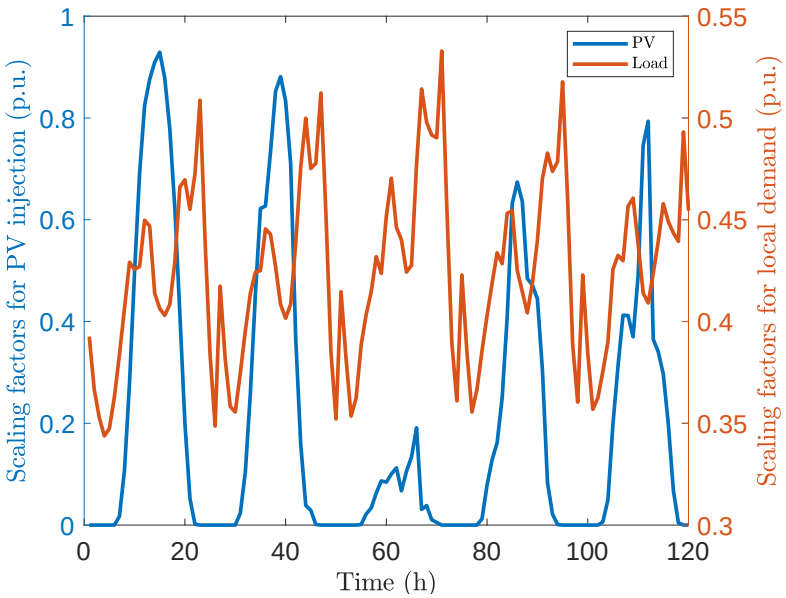

Figure 4: Scaling factors for the PV injections and the load.

percentage of the total peak load as follows: PV nodes = [12, 16, 18, 19], PV share (%) = [25, 45, 35, 25]. This work considers balanced, single-phase system operation, but the framework can be extended to three-phase unbalanced networks following [34].

In order to create multiple episodes, we consider different loading conditions taken from [38], and solar radiation profiles [39] for specific areas in Switzerland covering the time period of 2 summer months. Figure 4 shows the scaling factors for five summer days taken from the testing dataset. We assume a perfect spatial correlation, i.e., the PV scaling factors are the same at all nodes, which is a logical assumption for the examined neighborhood. These scaling factors are used to test the behavior of the different methods, i.e., they are not considered in the training dataset. In terms of operational limits, we assume a maximum (resp. minimum) acceptable voltage of 1.04 p.u (resp. 0.96 p.u.) and cable current magnitude of 1 p.u. on the cable base.

The implementation was done in MATLAB R2021a for the supervised learning and in Python 3.7 for the deep RL. For the OPF-based control, YALMIP R20210331 [40] was used as the modeling layer and Gurobi 9.1.2 [41] as the solver. For the deep RL modeling, we used OpenAI Gym 0.18 [36]. The results were obtained on an Intel Core i7-2600 CPU and 16 GB of RAM.

3.2. Supervised learning based on off-line data

3.2.1. OPF formulation

The selected objective function of the centralized OPF minimizes the control cost for all network nodes (N_b) for the time horizon (N_{hor}), and is given by

$$\min_{\boldsymbol{u}} \sum_{t=1}^{N_{hor}} \left\{ \sum_{j=1}^{N_b} \left(C_P \cdot P_{curt,j,t} + C_Q \cdot Q_{ctrl,j,t} \right) \right\} \cdot \Delta t, \tag{11}$$

where $\boldsymbol{u}$ is the vector of control variables and Δt is the length of each time period. The curtailed power of the DGs connected at node j and time t is given by $P_{curt,j,t} = P^{max}_{g,j,t} - P_{g,j,t}$, where $P^{max}_{g,j,t}$ is the maximum available active power and $P_{g,j,t}$ the active power injection of the DGs. The use of reactive power support $Q_{ctrl,j,t} = |Q_{g,j,t}|$ for each DG connected to node j and time t is also minimized; $Q_{g,j,t}$ represents the DG reactive power injection or absorption. The coefficients C_P and C_Q represent, respectively, the DG cost of curtailing active power and providing reactive power support. The assumption that $C_Q \ll C_P$ is made, which prioritizes the use of reactive power control over active power curtailment.

The power injections at every node j and time step t are given by

$$P_{inj,j,t} = P_{g,j,t} - P_{l,j,t}, \tag{12a}$$

$$Q_{inj,j,t} = Q_{g,j,t} - Q_{l,j,t}, \tag{12b}$$

where $P_{l,j,t}$ and $Q_{l,j,t}$ are the active and reactive node demands of constant power type. The nodal power balance equations using the full, non-linear AC power flow are given by

$$P_{inj,j,t} = |V_{k,t}| \sum_{m=1}^{N_b} |V_{m,t}| \left(G_{km} \cos\theta_{km,t} + B_{km} \sin\theta_{km,t} \right), \tag{13a}$$

$$Q_{inj,j,t} = |V_{k,t}| \sum_{m=1}^{N_b} |V_{m,t}| \left(G_{km} \sin\theta_{km,t} + B_{km} \cos\theta_{km,t} \right), \tag{13b}$$

where $Y_{km} = G_{km} + jB_{km}$ is the nodal admittance matrix, $|V_{k,t}|$, $|V_{m,t}|$ are the voltage magnitudes at buses k and m respectively at time t, and $\theta_{km,t} = \theta_{k,t} - \theta_{m,t}$ is the voltage angle difference between these buses at time t.

The constraint for the current magnitude for branch i at time t is given by

$$|I_{br,i,t}| \leq I_{i,max}, \tag{14}$$

where $I_{br,i,t}$ is the branch current, and $I_{i,max}$ is the maximum thermal limit. Similarly, the voltage constraints for each bus j and for each time step t are given by

$$V_{min} \leq |V_{j,t}| \leq V_{max}, \qquad |V_{slack}| = 1, \quad \theta_{slack} = 0, \tag{15}$$

where V_{max} and V_{min} are respectively the upper and lower acceptable voltage limits for the magnitudes of the bus voltages $|V_{j,t}|$, and $|V_{slack}|$, θ_{slack} are the fixed reference slack bus voltage magnitude and angle, respectively.

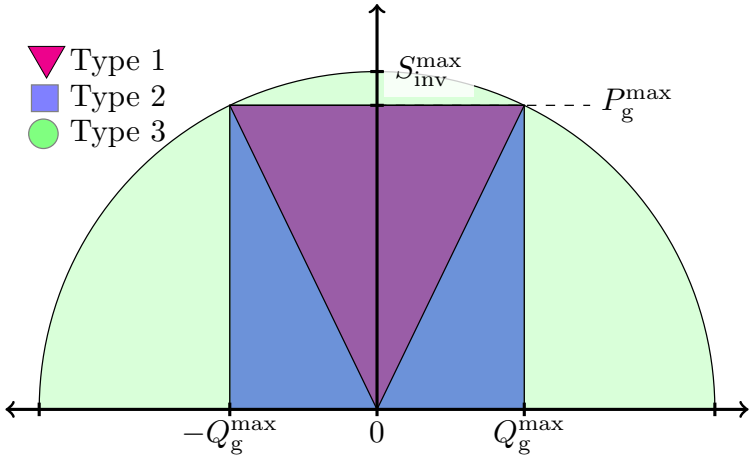

Figure 5: Different operational types within the P-Q inverter capability curve.

In this work, we only consider inverter-based DGs, such as PVs. Their output power limits are thus given by

$$P_{g,j,t}^{min} \leq P_{g,j,t} \leq P_{g,j,t}^{max}, \quad Q_{g,j,t}^{min} \leq Q_{g,j,t} \leq Q_{g,j,t}^{max}, \tag{16}$$

where $P_{g,j,t}^{min}$, $P_{g,j,t}^{max}$, $Q_{g,j,t}^{min}$ and $Q_{g,j,t}^{max}$ are the upper and lower limits for active and reactive DG power at each node j and time t. These limits vary depending on the type of the DG and the control schemes implemented.

Inverter-based DERs operate within the P-Q inverter capability curve, depicted in Fig. 5. We assume that the inverter is over-dimensioned by 10%, e.g. $S_{inv,j}^{max} = 1.1 \cdot P_{g,j}^{max}$ to allow for reactive power control even when the DER is operating at the maximum active power. There are different operational modes, such as a) the "triangular" mode ($\bigtriangledown$) which imposes an operational minimum power factor, b) the "rectangular" mode ($\square$) which allows reactive power control at times of low active power injections, and finally c) the semi-circle capability mode ($\bigcirc$) described by (17c) which represents the full capability region of the DER, without any constraint on the power factor. Thus, the limits are given by

$$(\bigtriangledown): -tan(\phi_{max})P_{g,j,t} \leq Q_{g,j,t} \leq tan(\phi_{max})P_{g,j,t}, \tag{17a}$$

$$(\square): -tan(\phi_{max})P_{g,j,t}^{min} \leq Q_{g,j,t} \leq tan(\phi_{max})P_{g,j,t}^{max}, \tag{17b}$$

$$(\bigcirc): \quad Q_{g,j,t}^2 \leq (S_{inv,j}^{max})^2 - P_{g,j,t}^2. \tag{17c}$$

The optimization problem (11)-(17) can be solved using the interior point algorithm based on the barrier method through the IPOPT solver [42] and provides the optimal set-points that are needed in the supervised learning method.

3.2.2. Supervised Learning

As base features for the active and reactive power DER control we use the active power demand $\phi_{j,1,t} = P_{l,j,t}$, the local measured voltage $\phi_{j,2,t} = V_{j,t}$, and the maximum active

power capability of the inverter $\phi_{j,3,t} = P^{max}_{g,j,t}$. Combinations of these features can also be considered, e.g., $\phi_{j,4,t} = \phi_{j,1,t} \cdot \phi_{j,2,t}$ or $\phi_{j,5,t} = (\phi_{j,1,t})^2$ if they increase the accuracy metrics. Finally, the feature matrix is given by $\mathbf{\Phi_{j,1}} = [\phi_{j,1,t}, \phi_{j,2,t}, \phi_{j,3,t}, \phi_{j,4,t}]^T$. Using the least squares method that can be found in [43] with many other supervised learning methods, the local model for active power control is derived by solving

$$\min_{\boldsymbol{\beta}} \sum_{t \in N_{OPF}} (P_{g,j,t} - \tilde{p}_{j,t})^2, \tag{18a}$$

$$\tilde{p}_{j,t} = \beta_{j,0} + \sum_{k \subset K} \beta_{j,k} \cdot \mathbf{\Phi_{j,1}}, \tag{18b}$$

where $\beta_{j,k}$ are the $k+1$ regression coefficients of the j^{th} unit for the $k \subset N_K$ features. A similar model for reactive power control is derived.

3.3. Deep RL

We define for the active power the *action* space as $\alpha_{1,j,t} \in [0, 1]$ where zero indicates complete active power curtailment and one indicates no curtailment. Thus, the active power injection of the DG at the node j is given by $P_{g,j,t} = \alpha_{1,j,t} P^{max}_{g,j,t}$. Regarding reactive power, the *action* space is defined as $\alpha_{2,j,t} \in [-1, 1]$, and the reactive power injection is given by $Q_{g,j,t} = \alpha_{2,j,t} S^{max}_{g,j,t}$. We use the semicircle DER capability of (17c) which needs to be respected by the agent. As *states* we consider the time t, the maximum available PV power $P^{max}_{g,j,t}$, and the local voltage magnitude $V_{j,t}$. The injected active and reactive power of the agent at time t will thus be given by

$$P_{inj,j,t} = \alpha_{1,j,t} P^{max}_{g,j,t} - P_{l,j,t}, \tag{19a}$$

$$Q_{inj,j,t} = \alpha_{2,j,t} S^{max}_{g,j,t} - Q_{l,j,t}. \tag{19b}$$

Considering the curtailment costs, the operational constraints regarding local voltage magnitudes and the inverter capabilities, the reward function at time t of the agent for the DER at bus j is given by

$$\begin{aligned} R(t) = -\sum_{t=0}^{T} \Big(& C_P (1 - \alpha_{1,j,t}) P^{max}_{g,j,t} + C_Q \alpha_{2,j,t} S^{max}_{g,j,t} + c_1 \cdot \max(0, V_{j,t} - V_{max}) \\ & + c_1 \cdot \max(0, V_{min} - V_{j,t}) + c_2 \cdot \max(0, P^2_{g,j,t} + Q^2_{g,j,t} - (S^{max}_{inv,j})^2) \Big) \end{aligned} \tag{20}$$

where c_1 is a parameter to penalize the local voltage constraint, and c_2 is a parameter to penalize situations that the apparent inverter power is not respected. Overall, the goal is to avoid local constraint violations in terms of voltage by controlling reactive power and by curtailing active power. Similar to the formulation of Section 3.2.1, by selecting the reactive power cost coefficient lower than the corresponding for active power curtailment, we prioritize the use of reactive power control. Thus, reactive power is expected to be utilized first to alleviate limit violations before active power is curtailed.

Table 1: Hyperparameters in the deep RL training stage.

Hyperparameter in DDPG	**Value**
Mini-batch size	120
Actor learning rate	1.00E-04
Critic learning rate	1.00E-03
Gradient Threshold	1
Sample time (hour)	1
Target smooth factor	1.00E-03
Experience buffer length	1.00E-05
Noise variance	1.00E-01
Noise variance decay rate	1.00E-06
Discount factor gamma	0.99
Optimizer	Adam

After the actions of the agent are defined, a power flow calculation is performed in order to derive the voltages at all nodes and obtain the new states from the environment. In case the agent's action does not respect the inverter's capability curve, the active and reactive power are decreased proportionally to satisfy the constraint.

In this paper, we use OpenAI Gym [36] as the modeling platform to test our deep RL model. Each episode is modeled as one day, i.e. 24 hours. Parameter tuning is a very important step in the implementation of RL algorithms. The choice of optimal parameter values depends on the specific mathematical problem and environment, and may require experimentation and tuning. Typical methods to tune the parameters of RL algorithms include grid search, random search [44], and Bayesian optimization. Grid search is a simple method where we define a set of values for each parameter and then evaluate the performance of the algorithm for all possible combinations of parameter values. In random search [44], the performance is evaluated for each combination of parameters after randomly sampling the parameter space. Bayesian optimization is a more sophisticated method because it constructs a probabilistic model to predict the performance of the algorithm for each combination of parameters and then uses it to choose the next set of parameters that evaluate [45]. Other recent methods that show promising results include genetic algorithms and heuristics [46] showing that the optimal parameter selection is an active research question. We refer the interested reader to [47] for a review and comparison of hyperparameter optimization techniques.

In this paper, we have used a combination of grid search and manual tuning to select the hyperparameters for DDPG in the context of our specific problem. We performed a sensitivity analysis of the most important parameters to investigate the effect of them on the stability and performance of the algorithm. More specifically, we compared the average reward over the training period, convergence, overall stability, and robustness. The needed hyperparameters for the implementation of the DDPG RL algorithm are listed in Table 1.

3.4. Comparison of Methods and Discussion

To compare the different decentralized methods, we investigate the following approaches:

- Method 0 - Business as usual (BaU): This represents a current local Volt/VAr control scheme defined in grid codes, e.g., [48], and adopts a one-solution-fits-all approach for the design of the static control laws irrespective of the location of each DG or of the specific grid specifications.
- Method 1 - OPF: An AC OPF solution is performed at each time step, representing the optimal benchmarking behavior. Active power curtailment and reactive power control using (17c) are allowed;
- Method 2 - Supervised Learning: The optimized local control schemes are derived in the training stage following Section 2.1, and are tested using new samples regarding solar radiation and load. An AC PF solution is performed for each time step, with the DG agent behaving according to their supervised data-driven schemes;
- Method 3 - Deep RL: In this method, the agent behaves according to the DDPG method described in Section 2.2. An AC PF solution is performed for each time step after the agent's actions have taken place.

We organize the remaining section into three parts. First, we provide detailed results for 5 summer days in order to get intuition about the outcome of each method, and understand their strengths and limitations. Then, we provide summarized monthly results, provide error metrics compared to the optimal method, and compare the different methods in terms of error metrics. Finally, we provide a comparative evaluation of the examined decentralized control methods, and we assess the trade-offs between privacy, needed computational effort, constraint consideration, scalability, and optimality.

3.4.1. Detailed Results over 5 *summer days*

Figure 6 presents the voltage magnitude evolution of Node 16, over the period of 5 summer days. The scaling factors for the DG unit (PV) and the nodal demand are given in Fig. 4. Considering the installed DG capacity and nominal load, Node 16 is mostly injecting power to the grid (apart from the third day) which leads to high voltages. It is obvious that the current industrial practice (Method 0) can be insufficient in cases of large DER penetration as severe voltage limit violations still occur. This is happening because this method does not consider the location of each DG or of the specific requirements for both active power curtailment and reactive power control to reduce the voltages. Method 1 corresponds to the centralized solution where the optimal behavior is achieved assuming that a bi-directional communication and control network exist. The optimal behavior indicates that the minimum control effort is used in terms of active and reactive power control is used to satisfy the operational voltage constraint at its acceptable limit.

On the other hand, Methods 2 and 3 mimic the optimal centralized solution (Method 1) without communication needs, i.e., they are based on decentralized approaches. We observe

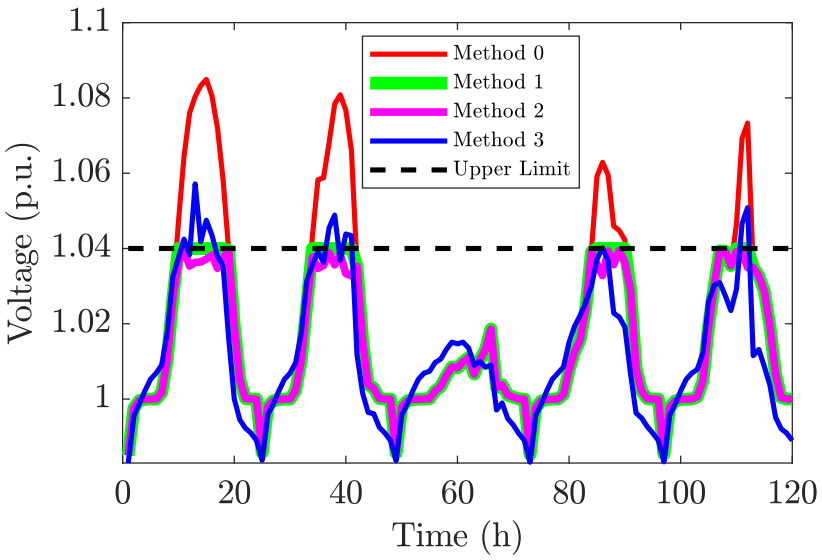

Figure 6: Voltage magnitude evolution for all examined methods over 5 summer days.

that Method 3 results in power being utilized ("activated") almost continuously, even when this is not needed. For instance, in contrast to all other methods that correctly do not control power on Day 3, Method 3 utilizes reactive power that influences the voltage trajectory. This indicates that a larger training dataset is needed so that the RL-agent learns not to utilize any control when it is not needed.

By design, Method 3 tries to avoid problematic instances (see reward function (20)), and shows a more conservative behavior, i.e., more active and reactive power control is activated to reduce the voltage back to acceptable values. However, both Methods 2 and 3 track the OPF behavior with marginal inefficiencies in terms of more activated control (Method 2), or short-term violations (Method 3).

As illustrated in Fig. 7, at periods of high voltages, an inductive behavior of the controller in Method 3 reduces the local voltage. When the maximum reactive power consumption is not enough, active power curtailment is also used during the first 2 days. Since this is penalized more in the reward function, the algorithm prioritizes reactive over active power control. However, the existence of some overvoltages indicate the need for more training data for the deep RL scheme.

3.4.2. Monthly Evaluation Results at Node 16

In this section, we summarize the results from applying the four decentralized control methods in real-time operation for the test period of one summer month. We compare first power quality values, i.e., the system's losses, the maximum observed voltage magnitude at Node 16 and cable (branch 5-16) loading, as well as active power curtailment. Then, we calculate error metrics to compare the different decentralized approaches against the centralized one.

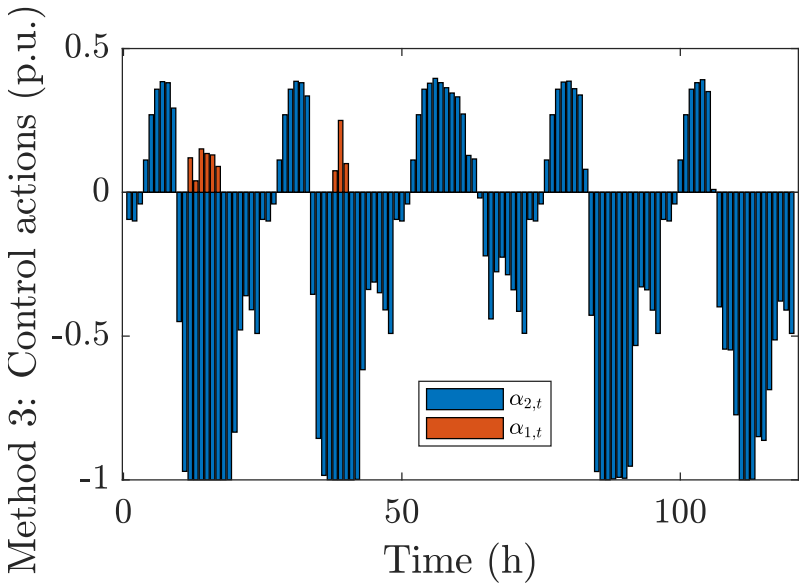

Figure 7: Active ($\alpha_{1,j,t}$) and reactive ($\alpha_{2,j,t}$) control action for the RL-based control scheme.

Table 2: Summarized monthly power quality results for all examined methods.

Method	0	1	2	3
P_{curt} (%)	1.510	4.340	7.235	4.785
Losses (%)	6.110	4.720	4.310	5.215
$\lvert V\rvert_{\max}$ (p.u.)	1.085	1.040	1.045	1.057
$\lvert I\rvert_{\max}$ (%)	125.940	100	104.570	108.130

Power quality quantities. Table 2 summarizes the monthly results in terms of power quality. The power losses and the active power curtailment refer to the whole month, whereas the voltage magnitude and cable flow only to the maximum value observed.

As benchmark we use the optimal case, i.e., Method 1 that satisfies the operational constraints at the minimum cost defined by the objective function (11). Method 0 (standard industry practice) shows higher losses than the OPF-based approach, due to less active power curtailment, i.e., more power is injected into the network. Method 2 shows higher total active power curtailment than the optimal case, but the worst power quality values over the whole testing month result in marginal violations. Finally, Method 3 keeps the total curtailment value closer to the optimal case with slightly higher violations than Method 2. Furthermore, it shows higher losses than Method 2 for two reasons; it curtails less active power and therefore more active power is injected, and it shows most of the time an inductive behavior by absorbing reactive power as can be seen in Fig. 7. Both Methods 2 and 3 mimic Method 1 without communication needs.

Error Metrics. In this part, we compare the performance of the examined methods using

common statistical error metrics to calculate how close the examined decentralized methods follow the optimal response. Thus, the voltage values obtained by Method 1, i.e., V_{m1} represent the 'true' values, and the time-series of Methods 0 (V_{m0}), 2 (V_{m2}) and 3 (V_{m3}) the 'predicted' values.

We will use the following error metrics:

- Mean Absolute Error (MAE): The MAE represents the average absolute difference between Method 1 and the estimated values of the other methods. It is given by: $MAE = \frac{1}{n}\sum_{t=1}^{n}|V_{m1,t} - V_{m,t}|$ where n is the number of data points (720 for the test month), $V_{m1,t}$ is the true voltage value at time t from Method 1, and $V_{m,t}$ is the estimated voltage value at time i from the other methods.

- Mean Squared Error (MSE): The MSE represents the average of the squared differences between Method 1 and estimated values of the other methods. It is given by: $MSE = \frac{1}{n}\sum_{t=1}^{n}(V_{m1,t} - V_{m,t})^2$.

- Root Mean Squared Error (RMSE): The RMSE is the square root of the MSE, which is easier interpretable being expressed in p.u. as the used voltage values. It is given by: $RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(V_{m1,t} - V_{m,t})^2}$.

- Coefficient of Determination (R^2): R^2 measures the proportion of the variance in Method 1 that is predictable from the other methods. It is given by: $R^2 = 1 - \frac{\sum t=1^n (V_{m1,t}-V_{m,t})^2}{\sum t=1^n (V_{m1,t}-\bar{V_{m1}})^2}$, where $\bar{V_{m1}}$ is the mean of the optimal voltage values in Method 1.

Table 3 shows the performance metrics of Methods 0, 2 and 3 in terms of voltage magnitude compared with the centralized OPF-based solution of Method 1. In absolute terms, the error metrics are very low because the behavior of all methods are the same or really close at non-problematic hours, e.g., at night, or at operational conditions that are not close to the limits. Thus, the relative comparison of the metrics is more relevant for the scope of this paper.

The results of the comparison showed that Method 2 resulted in the lowest error values across all four metrics, indicating that it is the closest to the benchmark optimal case. On the contrary, the current BaU shows the worst behavior in terms of these metrics showing that there is potential for improvement by applying more sophisticated decentralized methods. Method 3 shows intermediate error values closer to Method 1 than Method 0. The main reason for that is the continuous reactive power control (see Fig. 7), that changes the voltage compared to the OPF case even at non-problematic time periods.

3.4.3. Qualitative evaluation

In this part, we discuss the strengths and limitations of each method. The currently implemented local control schemes (i.e., Method 0), can scale easily to very large networks, do not show privacy concerns and do not require training effort. However, they often result in poor performance as they cannot achieve global optimality and can only consider local constraints. On the other side of the spectrum, the centralized OPF-based method (i.e.,

Table 3: Performance metrics of the examined methods compared to the optimal centralized solution.

	MAE	MSE	RMSE	R^2
V_{m0}	0.005340	0.000171	0.013059	0.457739
V_{m2}	0.000740	0.000003	0.001848	0.889138
V_{m3}	0.007145	0.000080	0.008947	0.745467

Method 1) works as a benchmark in terms of optimality, it can consider both local and global constraints, and can take into consideration any objective. However, it shows significant drawbacks regarding privacy, as all information is collected by one entity, and scalability. It requires a reliable bi-directional communication network, it is sensitive to modeling errors and uncertainties and is costly for large-scale systems. On the other hand, state-of-the-art ML-based methods, i.e., the supervised learning (Method 2) and RL approach (Method 3) examined in this paper, show a good trade-off between high performance with some training effort trying to combine the best of the previous two worlds. Method 2 could imitate very well the optimal response under expected normal conditions, while Method 3 has the advantage of being capable of adapting to changes in the system over time. For example, if another agent, such as a PV unit, is installed in the network, it will learn from experience and adapt to changing conditions without requiring re-training that is necessary for Method 2. A main difference of the ML-based methods is related to the way they try to approach the optimality of the central OPF scheme. The supervised learning Method 2 modifies the well-known and widely used local control schemes in order to customize them based on their location and grid challenges. The deep RL scheme of Method 3 aims at constructing a reward function to imitate the objective function of the OPF case. In this paper, only local information is used to allow a fair comparison of the studied decentralized methods.

Finally, Table 4 summarizes the findings and provides a qualitative evaluation of the used decentralized control methods in terms of computational burden for the training stage, scalability potentials, privacy-preserving capabilities, suitability to consider local and/or global constraints, and optimality. Overall, each method shows different characteristics in terms of limitations and strengths. The choice of method depends on the system's control requirements, constraints, data availability, available communication network, anticipation of frequent system's changes, and objectives of the controlled distribution system.

4. Conclusion

The increasing controllability and observability in distribution networks call upon more advanced control schemes that are scalable, optimal, and can consider privacy characteristics and the system's operational constraints. Although centralized solutions perform great in terms of optimality, they are associated with high costs and show robustness and privacy concerns. In this paper, we quantitatively compare the centralized OPF behavior against two machine-learning-based schemes based on supervised and reinforcement learning. We have demonstrated through a case study that in the absence of a full communication and monitoring infrastructure, the distribution grids can still optimize their grid operation safely,

Table 4: Qualitative evaluation of the examined control methods.

	Local Schemes (BaU)	**Central OPF**	**Supervised Learning**	**Deep RL**
Computational effort for training	✓	✓✓	✓✓✓	✓✓✓
Scalability	✓✓✓✓	✓	✓✓	✓✓
Privacy Preservation	✓✓✓✓	✓	✓✓✓	✓✓✓
Constraints' consideration	Local: ✓ Global: ✗	Local: ✓ Global: ✓	Local: ✓ Global: ✓	Local: ✓ Global:✓
Optimality	✓	✓✓✓✓	✓✓✓	✓✓✓
Adaptation to changes	✓✓	✓✓✓✓	✓	✓✓✓✓

by applying decentralized controllers based on supervised and reinforcement learning (RL). Furthermore, we provide intuition regarding the strengths and limitations of each method, highlighting the overall good performance of ML-based schemes across all examined criteria. For the specific benchmark European low-voltage grid and the available generating and consumption data, the supervised approach achieved the best results in terms of approaching the centralized optimal solution. It showed the maximum R^2 and the lowest MAE, MSE, and RMSE values. The RL method is also very promising since it can continuously adapt to changes when the operational conditions or the system's topology change. Future work will consider 3 different dimensions: first, incorporating more training data and controllable agents, such as electric vehicles, battery energy storage systems, and shiftable loads that require incorporating time constraints increasing the needed computational effort; then, it would be beneficial to use larger networks with real-world data and evaluate the control schemes in a real larger system considering the system's dynamics, interactions with the network's controlling devices, such as voltage regulating transformers, and implementation challenges; finally, further research is required in terms of reinforcement learning to study how to optimally tune the algorithm's parameters, investigate the multi-agent behavior where multiple agents have similar reward functions and explore the interactions between transmission and distribution voltage levels.

References

[1] Q. Sun, H. Li, Z. Ma, C. Wang, J. Campillo, Q. Zhang, F. Wallin, and J. Guo, "A comprehensive review of smart energy meters in intelligent energy networks," *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 464–479, 2015.

[2] N. Hatziargyriou, O. Vlachokyriakou, T. Van Cutsem, J. Milanović, P. Pourbeik, C. Vournas, M. Hong, R. Ramos, J. Boemer, P. Aristidou, V. Singhvi, J. dos Santos, and L. Colombari, "Task Force on Contribution to Bulk System Control and Stability by Distributed Energy Resources connected at Distribution Network," IEEE PES, Tech. Rep., 2017.

[3] S. Karagiannopoulos, R. Dobbe, P. Aristidou, D. Callaway, and G. Hug, "Data-driven control design schemes in active distribution grids: Capabilities and challenges," in *2019 IEEE Milan PowerTech*. IEEE, 2019.

[4] A. Eggli, S. Karagiannopoulos, S. Bolognani, and G. Hug, "Stability analysis and design of local control schemes in active distribution grids," *IEEE Transactions on Power Systems*, 2020.

[5] S. Karagiannopoulos, P. Aristidou, and G. Hug, "Data-driven local control design for active distribution grids using off-line optimal power flow and machine learning techniques," *IEEE Transactions on Smart Grid*, 2019.

[6] S. Karagiannopoulos, P. Aristidou, and G. Hug, "Hybrid approach for planning and operating active distribution grids," *IET Generation, Transmission & Distribution*, pp. 685–695, Feb 2017.

[7] P. Fortenbacher, M. Zellner, and G. Andersson, "Optimal sizing and placement of distributed storage in low voltage networks," in *Proceedings of the 19th Power Systems Computation Conference (PSCC), Genova*, Jun 2016.

[8] J. Lavaei and S. H. Low, "Zero duality gap in optimal power flow problem," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 92–107, 2012.

[9] D. K. Molzahn and I. A. Hiskens, "Sparsity-Exploiting Moment-Based Relaxations of the Optimal Power Flow Problem," *IEEE Transactions on Power Systems*, vol. 30, no. 6, pp. 3168–3180, Nov 2015.

[10] P. Kotsampopoulos, N. Hatziargyriou, B. Bletterie, and G. Lauss, "Review, analysis and recommendations on recent guidelines for the provision of ancillary services by Distributed Generation," in *IEEE IWIES*, 2013, pp. 185–190.

[11] S. Bolognani and S. Zampieri, "A distributed control strategy for reactive power compensation in smart microgrids," *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2818–2833, Nov. 2013.

[12] D. K. Molzahn, F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.

[13] A. Garg, M. Jalali, V. Kekatos, and N. Gatsis, "Kernel-Based Learning for Smart Inverter Control," July 2018. [Online]. Available: https://arxiv.org/pdf/1807.03769.pdf

[14] R. Dobbe, O. Sondermeijer, D. Fridovich-Keil, D. Arnold, D. Callaway, and C. Tomlin, "Data-Driven Decentralized Optimal Power Flow," 2018. [Online]. Available: http://arxiv.org/abs/1806.06790

[15] O. Sondermeijer, R. Dobbe, D. Arnold, and C. Tomlin, "Regression-based Inverter Control for Decentralized Optimal Power Flow and Voltage Regulation," *IEEE PES General Meeting*, 2016.

[16] M. Jalali, V. Kekatos, N. Gatsis, and D. Deka, "Designing reactive power control rules for smart inverters using support vector machines," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1759–1770, 2020.

[17] R. Dobbe, P. Hidalgo-Gonzalez, S. Karagiannopoulos, R. Henriquez-Auba, G. Hug, D. S. Callaway, and C. J. Tomlin, "Learning to control in power systems: Design and analysis guidelines for concrete safety problems," *Electric Power Systems Research*, vol. 189, p. 106615, 2020.

[18] J. Sun, Z. Zhu, H. Li, Y. Chai, G. Qi, H. Wang, and Y. H. Hu, "An integrated critic-actor neural network for reinforcement learning with application of ders control in grid frequency regulation," *International Journal of Electrical Power & Energy Systems*, vol. 111, pp. 286 – 299, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0142061518336767

[19] F. Daneshfar and H. Bevrani, "Load-frequency control: a ga-based multi-agent reinforcement learning," *IET Generation, Transmission Distribution*, vol. 4, no. 1, pp. 13–26, January 2010.

[20] M. Abouheaf, W. Gueaieb, and A. Sharaf, "Model-free adaptive learning control scheme for wind turbines with doubly fed induction generators," *IET Renewable Power Generation*, vol. 12, no. 14, pp. 1675–1686, 2018.

[21] B. J. Claessens, P. Vrancx, and F. Ruelens, "Convolutional neural networks for automatic state-time feature extraction in reinforcement learning applied to residential load control," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3259–3269, July 2018.

[22] F. Ruelens, B. Claessens, P. Vrancx, F. Spiessens, and G. Deconinck, "Direct load control of thermostatically controlled loads based on sparse observations using deep reinforcement learning," *CSEE*

Journal of Power and Energy Systems, vol. 5, 07 2017.
[23] Q. Yang, G. Wang, A. Sadeghi, G. Giannakis, and J. Sun, "Real-time voltage control using deep reinforcement learning," 04 2019.
[24] J. Duan, D. Shi, R. Diao, H. Li, Z. Wang, B. Zhang, D. Bian, and Z. Yi, "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.
[25] D. Wu, X. Zheng, D. Kalathil, and L. Xie, "Nested reinforcement learning based control for protective relays in power distribution systems," 2019.
[26] P. Kou, D. Liang, C. Wang, Z. Wu, and L. Gao, "Safe deep reinforcement learning-based constrained optimal control scheme for active distribution networks," *Applied Energy*, vol. 264, p. 114772, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261920302841
[27] X. Sun and J. Qiu, "Two-stage volt/var control in active distribution networks with multi-agent deep reinforcement learning method," *IEEE Transactions on Smart Grid*, vol. 12, no. 4, pp. 2903–2912, 2021.
[28] P. Li, J. Shen, M. Yin, Y. Zhang, and Z. Wu, "A deep reinforcement learning voltage control method for distribution network," in *2022 IEEE 5th International Electrical and Energy Conference (CIEEC)*, 2022, pp. 2283–2288.
[29] D. Hu, Z. Ye, Y. Gao, Z. Ye, Y. Peng, and N. Yu, "Multi-agent deep reinforcement learning for voltage control with coordinated active and reactive power optimization," *IEEE Transactions on Smart Grid*, vol. 13, no. 6, pp. 4873–4886, 2022.
[30] W. Wang, N. Yu, Y. Gao, and J. Shi, "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, 2020.
[31] D. Cao, J. Zhao, W. Hu, F. Ding, N. Yu, Q. Huang, and Z. Chen, "Model-free voltage control of active distribution system with pvs using surrogate model-based deep reinforcement learning," *Applied Energy*, vol. 306, p. 117982, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S030626192101285X
[32] M. Glavic, "(deep) reinforcement learning for electric power system control and related problems: A short review and perspectives," *Annu. Rev. Control.*, vol. 48, pp. 22–35, 2019.
[33] J. Suchithra, D. Robinson, and A. Rajabi, "Hosting capacity assessment strategies and reinforcement learning methods for coordinated voltage control in electricity distribution networks: A review," *Energies*, vol. 16, no. 5, 2023. [Online]. Available: https://www.mdpi.com/1996-1073/16/5/2371
[34] S. Karagiannopoulos, P. Aristidou, and G. Hug, "A Centralised Control Method for Tackling Unbalances in Active Distribution Grids," in *in Proceedings of the 20th Power Systems Computation Conference (PSCC), Dublin*, June 2018.
[35] A. Gomez-Exposito, A. J. Conejo, and C. Cañizares, *Electric energy systems: analysis and operation*. CRC press, 2018.
[36] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.
[37] G. E. Uhlenbeck and L. S. Ornstein, "On the theory of the brownian motion," *Physical review*, vol. 36, no. 5, p. 823, 1930.
[38] K. Strunz, E. Abbasi, C. Abbey, C. Andrieu, F. Gao, T. Gaunt, A. Gole, N. Hatziargyriou, and R. Iravani, "Benchmark Systems for Network Integration of Renewable and Distributed Energy Resources," *CIGRE, Task Force C6.04*, no. 273, pp. 4–6, 4 2014.
[39] "MeteoSwiss - Federal Office of Meteorology and Climatology." [Online]. Available: http://www.meteoswiss.admin.ch/
[40] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in matlab," in *In Proceedings of the CACSD Conference*, Taiwan, 2004.
[41] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2016. [Online]. Available: http://www.gurobi.com
[42] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm

for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 2006.
[43] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference and prediction*. Springer, 2009.
[44] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization." *Journal of machine learning research*, vol. 13, no. 2, 2012.
[45] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems*, vol. 25, 2012.
[46] A. Dhiman, A. Kaushik, C. Rajak, and F. Naaz, "A comprehensive review on deep reinforcement learning in robotics," *Robotics and Autonomous Systems*, vol. 146, p. 103785, 2021.
[47] A. A. Chowdhury, A. Das, K. K. S. Hoque, and D. Karmaker, "A comparative study of hyperparameter optimization techniques for deep learning," in *Proceedings of International Joint Conference on Advances in Computational Intelligence: IJCACI 2021*. Springer, 2022, pp. 509–521.
[48] IEEE 1547-2018, "Standard for interconnection and interoperability of distributed energy resources with associated electric power systems interfaces," Standard, 2018.