Adaptive Congestion Control for Electric Vehicle Charging in the Smart Grid

Abdullah Al Zishan, Moosa Moghimi Haji, Omid Ardakanian *Member, IEEE,*

***Abstract*—This paper proposes an adaptive control algorithm for plug-in electric vehicle charging without straining the power system. This control algorithm is decentralized and merely relies on congestion signals generated by sensors deployed across the network, e.g., distribution-level phasor measurement units. To dynamically adjust the parameter of this congestion control algorithm, we cast the problem as multi-agent reinforcement learning where each charging point is an independent agent which learns this parameter using an off-policy actor-critic deep reinforcement learning algorithm. Simulation results on a test distribution network with 33 primary distribution nodes, 3630 low voltage nodes, and 500 electric vehicles corroborate that the proposed algorithm tracks the available capacity of the network in real-time, prevents transformer overloading and voltage limit violation problems for an extended period of time, and outperforms other decentralized feedback control algorithms proposed in the literature. These results also verify that our control method can adapt to changes in the distribution network such as transformer tap changes and feeder reconfiguration.**

***Index Terms*—Electric car, congestion, reinforcement learning.**

I. Introduction

THE electric vehicle (EV) market has been growing rapidly in recent years. According to the IEA's Global EV Outlook 2019 [1], the number of private electric vehicle charging points, which are mostly connected to low-voltage distribution networks, was approximately 4.66 millions worldwide in 2018 showing a 48% increase compared to the year before. This trend is expected to continue in the future with increased global demand for electric passenger cars and electric light and medium-duty trucks.

Even at low voltage, EV chargers draw a large amount of power. This makes the reliable operation of the distribution system challenging, especially in neighbourhoods with high EV penetration [2]. Transformer overloading, voltage sag, increased losses, and reduced equipment lifetime are some of the problems created by the high EV charging demand [3], [4]. Although upgrading the distribution network can address these problems, it may not be practical due to the high costs involved. An alternative approach is to leverage the inherent flexibility of EV charging loads and schedule them to utilize the available network capacity without causing these problems.

The EV charging control algorithms proposed in related work can be divided into two categories: centralized and decentralized. In the first category, a central controller determines an admissible charge power for every connected EV after taking into account various constraints. These charge powers are then sent to the respective charging points. Since all EV related data and other measurements must be sent to the control centre and decisions must be sent back to charging points, these methods have high communication overhead and do not usually scale with the size of the network and the number of charging points. Additionally, the central node will become a single point of failure and sending data about individual EVs to a third party could raise privacy concerns. Decentralized EV charging control methods can address these shortcomings [5]–[7]. Most decentralized methods rely on a model (i.e., the admittance matrix) which relates EV charge powers to voltages and real power flows in different parts of the distribution network. But the distribution system model is often inaccurate or nonexistent in practice [8]. In the absence of an accurate model, it is possible to incorporate an approximate model which ignores losses and reactive power flows [9]. Nevertheless, due to these approximations, the resulting model cannot be used to control voltage and does not guarantee that available resources are fully utilized.

A promising alternative, inspired by the design of Internet congestion control algorithms, is to employ a feedback control algorithm which infers *congestion* (i.e., overloading and voltage problems) based on measurements of local or remote sensors rather than using the network model to determine if there is congestion in some part of the network. Specifically, the charging points can adjust their power based on the received feedback such that the overall EV charging demand tracks the available capacity of the network in real-time [10], [11]. These algorithms draw upon a rule-based control scheme, called Additive-Increase Multiplicative-Decrease (AIMD), which has been successfully adopted in TCP congestion control [12]. In AIMD, the control rules are fixed regardless of the changes in the network condition [13]. Employing fixed rules is not a major problem for fast-timescale control, but can lead to inefficiencies when control decisions are made on a slower timescale, e.g., on the order of seconds in the distribution grid. In this work we update the control rules dynamically to increase the overall network utilization and responsiveness [14].

Inspired by [13], we propose an Adaptive AIMD-like method (dubbed A-AIMD) for controlled EV charging. Specifically, we use the Multi-Agent Reinforcement Learning framework to adjust the additive and multiplicative rates of AIMD at each charging point assuming that it is a Reinforcement Learning (RL) agent. The RL agents are trained such that they can react to different conditions of the grid given the congestion signals they receive. Our control method is model-free and requires reliable transmission of congestion signals. It scales

As a part of the University of Alberta's Future Energy Systems research initiative, this research was made possible in part thanks to funding from the Canada First Research Excellence Fund.

A. A. Zishan (email: azishan@ualberta.ca), M. Moghimi (email: moghimih@ualberta.ca) and O. Ardakanian (email: ardakanian@ualberta.ca) are with the Department of Computing Science, University of Alberta, Canada.

with the size of the distribution network and has the same communication overhead as other decentralized feedback control algorithms that mimic TCP congestion control [10], [11]. We note that the proposed method offers 'best effort' service similar to the Internet, i.e., it tries to meet the user-specified deadlines but cannot guarantee that charging demands are fully met if the network is congested or deadlines are set too early. Instead, our goal is to maximize the total charge power of EVs without causing congestion in the network.

Our contribution is threefold:

- We propose an adaptive AIMD-like algorithm for controlled EV charging in the smart grid. An actor-critic reinforcement learning algorithm is used to adapt the parameters of AIMD to varying network conditions.
- We evaluate the efficacy of our control method in preventing transformer overloading and voltage deviation problems through simulations on a test distribution system with 33 primary nodes and 1850 low-voltage end nodes that supply residential customers. We use real EV and load demand data for residential customers.
- We compare our method with various EV charging control methods that avoid congestion and show that it outperforms these baselines.

This work extends our previous work [14] by (a) evaluating the performance of A-AIMD in a distribution network with EV chargers and uncontrolled loads, such as homes connected to low-voltage feeders; (b) presenting results for two types of congestion, namely transformer overloading and voltage limit violations, and examining the compatibility of our control algorithm with traditional voltage control mechanisms implemented by tap-changing transformers; (c) showing that RL agents can quickly adapt to changes in the distribution network model (e.g., feeder reconfiguration); and (d) studying the importance of using imitation learning for offline training.

II. Related Work

A. Congestion Control Algorithms for EV Charging

The success of TCP congestion control algorithms motivated researchers to apply them to control congestion in the power system. In [10] the authors propose a decentralized charging strategy based on the AIMD algorithm to maximize EV charging powers while considering fairness. Each EV increases its charge rate by an additive factor until the network gets congested; in that case it reduces its charge rate by a multiplicative factor. However, they do not consider practical power system constraints, such as transformer ratings and voltage constraints. In [15] the basic AIMD algorithm of [10] is extended to incorporate these constraints assuming a hierarchical communication network. Additionally, the authors propose a price-adjusted available power heuristic to encourage shifting the EV charging load to off-peak times.

More recently, a fully decentralized AIMD-like EV charging algorithm in proposed in [16]. This algorithm relies on local voltage measurements only and can be tuned to have near-optimal operation. However, fine-tuning the parameters is nontrivial. Since the method relies on local information, it may not be able to fully utilize the available capacity of the network. Moreover, the method does not consider fairness when allocating power to EVs. Reference [17] proposes AIMD-like EV charging methods that consider fairness. These methods require the knowledge of the system model which is not typically available in practice. Reference [11] proposes an EV charging controller inspired by the TCP slow start mechanism used in the Internet. This controller cannot adjust the amount of increase or decrease in the charge power of EVs based on real-time dynamics of the power system.

B. Model-Free Control of EV Charging Load

The application of reinforcement learning to control EV chargers has gained attraction lately especially because the distribution system model is typically unavailable. The RL-based methods are model-free, learning the mapping from actions to states through interaction with the environment. One particular line of work focuses on scheduling a group of EVs, typically connected to an aggregator [18], [19]. Although the results reported in these studies are satisfactory, they solve the problem in a centralized fashion, hence suffer from the same drawbacks as other centralized control methods. Another line of work focuses on charging a single EV considering different objectives, such as providing vehicle-to-grid support or reducing long-term electricity costs [20], [21]. The proposed methods are decentralized, but they do not discuss coordination among multiple EVs connected to a power system. Unlike these methods, we focus on decentralized and coordinated scheduling of EV charging without making any specific assumption about how EVs are connected to the power system.

III. System Model

We study the EV charging problem in the context of a 33-bus test distribution network [22]. To have an accurate model, we consider the secondary (low voltage) feeders supplying EV chargers and inelastic loads (e.g., residential loads). We adopt the IEEE European low voltage test feeder [23] as the low-voltage feeders of the distribution system. Each of these low-voltage feeders is connected to a node of the primary distribution network via a step-down transformer. Figure 1 shows the low-voltage feeder connected to Node 25 of the primary distribution network. Similar feeders are connected to Nodes 2-33, but they are not depicted in this figure. The primary distribution network is connected to the substation via the substation transformer. We assume that both primary and secondary feeders have a radial operational structure.

We augment the system by adding two tie switches between Nodes 12 and 22, and Nodes 18 and 33. Both switches are normally open and can be operated to change the configuration of the system in our case study (Scenario B in Section V-C). The IEEE European low voltage test feeder has 906 nodes out of which 55 nodes are end nodes with single-phase loads connected to them. Since we have 32 of these low-voltage feeders connected to Nodes 2 to 33 of our primary distribution system, we have 1815 end nodes in total that supply the loads, including EV chargers and residential loads.

We aim to control the charge power of EV charging points to fully utilize the available capacity of the grid, avoid overloading of transformers, and keep nodal voltages in the primary

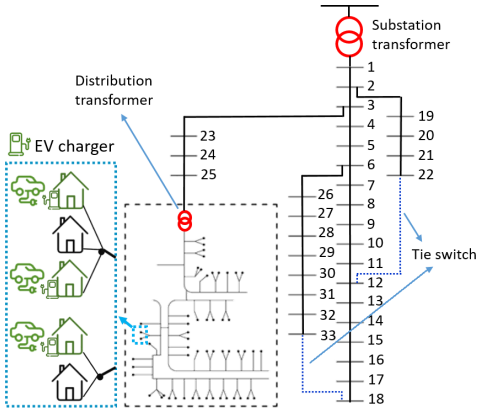

Fig. 1: The primary distribution network (Nodes 1-33) and secondary feeders (one of them is shown inside the dashed rectangle) supplying residential loads and EV chargers.

distribution system between 1.05 and 0.95 pu at all times. To control EV charging, we rely on time-synchronized measurements provided by distribution-level phasor measurement units (DPMUs) or other sensors installed on the secondary side of the transformers[1]. Although it is assumed in our simulation that all the distribution transformers are equipped with DPMUs, it is not a requirement of our method. The distribution system operator could install DPMUs only at critical locations which are likely to be congested. The DPMUs can measure three-phase voltage and current phasors at regular intervals. Once the transformer's voltage and current phasors are available, it is possible to compute its loading and subtract it from its short-time rating to estimate the congestion state in volt-ampere. For voltage congestion, the congestion state is expressed in per unit nodal voltage minus the lower voltage limit (i.e., 0.95). In either case, the congestion state indicates the available capacity of the distribution network. We assume there is an overlay communication network connecting each EV charging point to sensors (or DPMUs) installed upstream, i.e., at the substation and distribution transformers that supply its demand. Thus, charging points receive a congestion signal when their upstream transformers are overloaded or voltage limits are violated in the primary network.

We put a reinforcement learning agent at each charging point. This agent is responsible for adjusting the charge power of the corresponding charger according to the (perceived) congestion state of its upstream nodes. Keep in mind that there is no direct communication between the agents. When a congestion is detected by a sensor, a congestion signal is sent to all agents downstream of the sensor's location and all of the notified agents will react to the congestion.

IV. Control Methodology

In this section, we describe how a residential charging point (an RL agent) learns the state-action mapping from interaction with the power system. The agent's action determines the parameter of the A-AIMD algorithm and therefore affects the

[1]Since we do not have real sensor data from the selected distribution network to use in our simulations, we run power flow analysis (using the accurate distribution network model) to calculate the quantities that would be measured by sensors in the real world and treat them as sensor measurements.

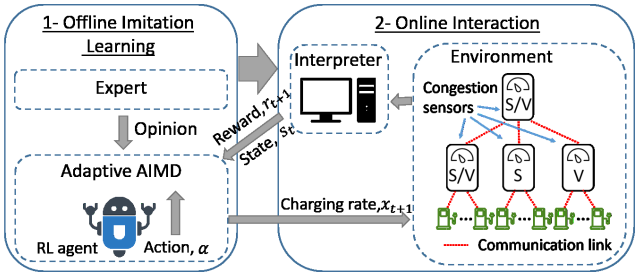

Fig. 2: An overview of the proposed methodology.

charge power of the corresponding EV charging point. We first present the optimization problem that was traditionally solved in a centralized fashion. Then, we present our multi-agent framework followed by the description of the adaptive AIMD control method. The details of our RL agents are discussed next. As shown in Figure 2, RL agents are trained in two phases: offline training via imitation learning and online interaction with environment. In the offline training phase, the agent is initialized by imitating the centralized controller presented later in this section. In the online (interactive) phase, the agent interacts with the environment and updates its policy using a policy gradient algorithm. We explain the offline and online phases for training the RL agents as well as the details of the policy gradient method at the end of this section.

A. Optimization Problem Formulation

Our goal is to maximize the total power delivered to EVs while avoiding transformer overloading and voltage limit violations. Thus, the optimization problem can be written as

$$\begin{aligned} \text{maximize} \quad & \sum_{i\in\mathcal{E}} x_t^i \qquad (1)\\ \text{subject to:} \quad & \text{Power Flow Eqs.} \\ & L_t^j \le \bar{L}^j, && \forall j \in \mathcal{L} \\ & V_{min} \le |V_t^k| \le V_{max}, && \forall k \in \mathcal{N} \\ & 0 \le x_t^i \le M^i, && \forall i \in \mathcal{E} \end{aligned}$$

where $\mathcal{E}$ is the set of active charging points, x_t^i is the charge power of charging point i at time t, M^i is its maximum charge power, L_t^j is the loading of transformer j at time t, $\bar{L}^j$ is its rated capacity, $\mathcal{L}$ is the set of transformers in the distribution network, V_t^k is the voltage magnitude at Node k at time t, V_{min} and V_{max} are respectively the minimum and maximum voltage magnitude limits, and $\mathcal{N}$ is the set of primary nodes.

Knowing the admittance matrix and the demand of uncontrolled loads (e.g., homes), all nodal voltages and the transformer loading can be determined using the nonlinear power flow equations. This problem is a non-convex optimization problem [24]. One way to deal with it is to use approximation and make it a convex problem as presented later in this section.

B. Multi-Agent Reinforcement Learning (MARL)

A reinforcement learning agent learns a *policy* (a sequence of actions) that maximizes the expected reward received over time from interaction with the environment at discrete time steps. The optimal policy learned by the agent is the solution

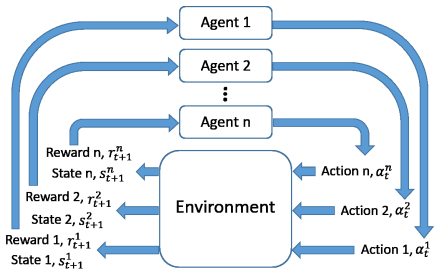

Fig. 3: The agent-environment interplay in a multi-agent reinforcement learning problem.

to a stochastic optimal control problem which can be framed using a Markov Decision Process (MDP). In each time step, the agent i receives some representation of the environment's state, s_t, and takes an action, α^i_t, which changes the environment's state to s_{t+1}. It then receives a numerical reward, r_{t+1}, from the resulting environment. In multi-agent reinforcement learning, N independent agents try to maximize a reward signal they simultaneously receive from the environment. Figure 3 shows interaction between agents and the environment in the multi-agent setting. The rewards received by different agents may not be the same and the agents cannot directly communicate with each other.

There are three types of multi-agent reinforcement learning frameworks: *fully cooperative*, *fully competitive*, and *mixed* [25]. In the fully cooperative framework, agents receive the same reward signal and cooperate to achieve a common goal by optimizing a common return, whereas in the competitive framework, each agent receives a different reward and the sum of the return of the agents is usually zero. The mixed framework includes both cooperative and competitive agents, with general-sum returns.

Our problem is decentralized and cooperative in nature. Here, each charging point is considered as an independent agent. All agents receive the same reward signal, i.e., loading of the substation. They collectively maximize the substation loading by communicating with sensors that generate congestion signal. Specifically, they adjust their charge power using an adaptive AIMD approach which is described next.

C. Adaptive AIMD

AIMD is a feedback control algorithm which is known for its use in TCP congestion control [12]. In this algorithm, the end nodes increase their rate (in our case their charge power) in an additive fashion until the network gets congested. Once congestion is signalled by intermediate nodes (in our case the sensors) to the end nodes, they decrease their rate multiplicatively to quickly alleviate congestion. A-AIMD is an adaptive control algorithm which mimics the AIMD method used in TCP congestion control, but unlike AIMD, it does not have fixed additive increase and multiplicative decrease rates. It learns these parameters through interaction with the environment (see Algorithm 1). In particular, each RL agent learns the mapping between the congestion signals and a continuous action, $-1 \leq \alpha \leq 1$. This action determines the additive increase factor when it is positive and the multiplicative decrease factor otherwise.

TABLE I: List of reinforcement learning parameters.

RL Parameter	Brief Description
state: s^i_t	a 3-tuple: (t, x^i_t, Λ^i_j)
action: α^i_t	parameter of A-AIMD algorithm ($-1 \leq \alpha \leq 1$)
reward: r	instantaneous loading of the substation transformer (no congestion) or negative of this value (congestion)
policy: π_θ	Gaussian policy with parameters μ_θ and σ_θ

D. Reinforcement Learning Agents

In our reinforcement learning problem, the action-space is continuous and each charging point is an agent that takes an action which determines the parameter of the A-AIMD algorithm that modulates its charge power. Specifically, agent i samples an action $-1 \leq \alpha^i_t \leq 1$ at time step t as follows:

$$\alpha^i_t \sim \pi^i_\theta(\alpha|s^i_t) = \mathrm{P}(A^i_t|S^i_t = s^i_t; \theta) \tag{2}$$

where π^i is the policy parameterized by θ, and s^i_t is the environment state perceived by the agent. We consider a Gaussian policy with parameters $\mu_\theta, \sigma_\theta$ given by neural networks:

$$\pi^i_\theta(\alpha|s^i_t) = \frac{1}{\sqrt{2\pi\sigma^2_\theta}} \exp\left(-\frac{(\alpha-\mu_\theta)^2}{2\sigma^2_\theta}\right) \tag{3}$$

To update the policy parameter, θ, we use the soft actor-critic algorithm (introduced later). We divide the agent-environment interaction into a number of episodes where each episode spans one day. Upon convergence, we find a policy that maximizes the expected reward.

We define the partial state of the environment perceived by agent i at time t as a tuple $s^i_t = (t, x^i_t, \Lambda^i_t)$, where x^i_t is the charge power of the charging point controlled by agent i, and $\Lambda^i_t = \{\lambda_j | j \in \mathcal{L}_i\}$ compactly represents the congestion signals received by agent i at time t where

$$\lambda_j = \begin{cases} 1, & \text{if congestion is signalled by sensor } j \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

We remark that agent i will receive congestion signals only from the sensors located on its path to the substation (i.e., members of the set $\mathcal{L}_i$). Table I summarizes the RL parameters.

When agent i takes action α^i_t, it is translated into a new charge power for the next time slot following the A-AIMD rules. Charging the EV at this rate causes a transition from s_t to s_{t+1}. The agent then receives a reward, r_{t+1}, from the sensing node installed at the substation. Due to the Markov property, the reward and the next state are independent of previous states given the current state. In our problem formulation, the reward received by each agent is the instantaneous loading of the substation transformer when there is no congestion, and the negative of that value when at least one of the sensors on its path to the substation detects congestion:

$$r = \begin{cases} Load, & \text{if no congestion} \\ -Load, & \text{otherwise} \end{cases} \tag{5}$$

Note that the measured substation load is a function of the total power drawn by all EV charging points (i.e., $\sum_i x^i_t$). Thus, the agents receive a higher reward when the substation's utilization increases as long as there is no transformer overloading or undervoltage problem on their path to the substation. When at least one of these problems occurs, the agents who might have contributed to the problem will receive a negative reward.

E. Offline Training Phase

Although RL agents are supposed to learn the state-action mapping through online interaction, it is shown in previous work that prior offline training can significantly lower sample complexity by reducing random exploration during the online interaction [26]. Thus, in the offline training phase we use an imitation learning approach [27] which exposes the RL agents to trajectories generated by an expert, in this case the solution to the optimization problem presented earlier. This enables the agents to learn from these demonstrations.

As mentioned earlier, controlling the charge power of EVs can be cast as a constrained optimization problem where the constraints depend on the structure of the power system and locations of charging points. We use the optimization framework for controlled EV charging in a distribution network that was originally introduced in [9]. In this framework, an approximate model that ignores losses and reactive power flows is utilized to relate EV charge powers to real power flow at the feeder head. We solve this optimization problem in a centralized fashion for each time slot and use the solution as expert demonstration in offline training of the RL agents.

Solving the following convex optimization problem yields the charge power of every charger (x_t) for the interval $[t, t+1)$:

$$\begin{aligned} \text{maximize} \quad & \sum_{i\in\mathcal{E}} x_t^i && (6)\\ \text{subject to:} \quad & \sum_i T_{ij} x_t^i \le c_j, && \forall j \in \mathcal{L} \\ & 0 \le x_t^i \le M^i, && \forall i \in \mathcal{E} \end{aligned}$$

Here, x_t^i is the charge power of charging point i at time t, T_{ij} is a matrix that encodes the topology of the distribution network ($T_{ij} = 1$ if EV i is supplied by transformer j), c_j is the available capacity of the transformer j after subtracting the household demands from its rated capacity, $\mathcal{L}$ is the set of transformers installed in the distribution network, and M^i is the maximum charge power of charger i. The optimal solution is an allocation that uses up the available capacity of the network at time t. Note that this optimization framework cannot prevent voltage limit violation incidents from happening as it ignores reactive power flow in the distribution network. It is one of the problems of using an optimization based approach that can be resolved by a model-free controller.

We model this convex optimization problem in CVXPY and solve it using MOSEK to obtain a baseline. The problem is similar to the centralized approach used in [9] except that it maximizes the sum of charge powers rather than the sum of the logarithm of each charge power. Using the logarithmic utility function allows to achieve *proportionally fairness* while allocating power to EV charging points. However, it complicates the design of the RL-based controller. Throughout this paper, we refer to this approach as the *centralized* control method.

F. Online Interaction Phase

Algorithm 1 presents the online interaction phase of A-AIMD for agent i with hyperparameters $B \ge 1$ and $A > 0$. After perceiving the state, it samples action α (line 4) from the policy $\bar{\pi}_\theta$ (initially the policy trained offline). In line 5, it checks for congestion by checking the sign of α. The negative sign of α indicates congestion and the positive sign indicates normal operation. If congestion is detected, it lowers the charge power by multiplying it by $0 < B^\alpha \le 1$ (line 6). When there is no congestion (α is positive) the charge power is increased by an additive factor $A\alpha$ (line 8). The algorithm then receives the next state and reward after charging an EV at rate x_{t+1} (lines 11-12). We use *soft actor-critic* to update the policy parameter (line 14) given a replay buffer $\mathcal{D}$. The replay buffer is the collection of current state, action, next state, and reward (line 13). Note that this algorithm updates the EV charge power in real time.

Algorithm 1: Adaptive AIMD: Online Interaction

input : Hyperparameters: $A > 0$ and $B \ge 1$;
Pre-trained policy, π_θ^i;

1 **for** $e = 1, 2, \ldots$,*# of episodes* **do**
2 $s_1^i \longleftarrow (1, x_1^i, \Lambda_1^i)$
3 **for** $t = 1, 2, \ldots$,*# of steps* **do**
4 $\alpha_t^i \sim \pi_\theta^i(s_t^i)$
5 **if** $\alpha_t^i \le 0$ **then** `// if congested`
6 $x_{t+1}^i \longleftarrow x_t^i \times B^{\alpha_t^i}$
7 **else**
8 $x_{t+1}^i \longleftarrow \min\{M^i, x_t^i + A\alpha_t^i\}$
9 **end**
10 Implement(x_{t+1}^i) `// charge EV at` x_{t+1}^i
11 $(r_{t+1}, \Lambda_{t+1}^i) \longleftarrow$ env.step(x_{t+1}^i)
12 $s_{t+1}^i \longleftarrow (t+1, x_{t+1}^i, \Lambda_{t+1}^i)$
13 $\mathcal{D}^i \longleftarrow \mathcal{D}^i \cup \{(s_t^i, \alpha, s_{t+1}^i, r_{t+1})\}$
14 $\theta \longleftarrow SAC(\mathcal{D})$
15 **end**
16 **end**

We use a policy gradient algorithm for updating the policy as it is the natural choice for continuous action/state space problems [28]. In a policy gradient method, the policy is represented by a parameterized function (possibly a neural network) π_θ. The agent updates the policy parameter towards the direction of the gradient of a performance function: $\theta \longleftarrow \theta + \gamma \nabla_\theta J(\theta)$.

The SAC algorithm [29] is a new policy gradient method which incorporates the entropy measure of the policy into the reward to encourage exploration. Concretely, it is an off-policy actor-critic model following the maximum entropy reinforcement learning framework which tries to learn a policy that acts as randomly as possible while still succeeding at the task. The policy is trained with the objective to maximize the expected return and entropy at the same time:

$$J(\theta) = \sum_{t=1}^{T} \mathbb{E}_{(s_t,\alpha_t)\sim\rho_\pi}[r_t + \eta\mathcal{H}(\pi(.|s_t))]. \quad (7)$$

Here ρ_π is the marginal distribution of state-action pairs, i.e., (s_t, α_t) induced by policy π, $\mathcal{H}$ is the entropy measure, and η determines the importance of the entropy term (aka the temperature parameter).

The SAC aims to learn three functions: the policy π_θ, the soft Q-value function Q_ψ, and the soft state value function V_ϕ. Both the Q-function and the state value function, V, can

Algorithm 2: Soft Actor Critic (SAC)

input : Initialized parameters: $\phi, \bar{\phi}, \psi, \theta$, Mixture Coefficient: τ
output: Policy Parameter: θ
for *each gradient step* **do**
  $\phi \longleftarrow \phi - \gamma_v \nabla_\phi J_V(\phi)$
  $\theta \longleftarrow \theta - \gamma_\pi \nabla_\theta J_\pi(\theta)$
  $\psi \longleftarrow \psi - \gamma_q \nabla_\psi J_Q(\psi)$
  $\bar{\phi} \longleftarrow \tau\phi + (1-\tau)\bar{\phi}$
end

be instantiated as neural networks [29]. The soft state value function is trained to minimize the mean squared error:

$$J_V(\phi) = \mathbb{E}_{s_t \sim \mathcal{D}} \left[\tfrac{1}{2} (V_\phi(s_t) - \mathbb{E}_{\alpha_t \sim \pi_\theta} [Q_\psi(s_t, \alpha_t) - \log \pi_\theta(\alpha_t | s_t)])^2 \right], \tag{8}$$

with a stochastic gradient

$$\nabla_\phi J_V(\phi) = \nabla_\phi V_\phi(s_t)(V_\phi(s_t) - Q_\psi(s_t, \alpha_t) + \log \pi_\theta(\alpha_t | s_t)). \tag{9}$$

Here D is the distribution of previously visited (s_t, α_t) pairs. Similarly, the Q-function is trained to minimize the mean squared loss

$$J_Q(\psi) = \mathbb{E} \left[\tfrac{1}{2} (Q_\psi(s_t, \alpha_t) - (r_t + \gamma \mathbb{E}[V_{\bar{\phi}}(s_{t+1})]))^2 \right], \tag{10}$$

which can be optimized with stochastic gradients

$$\nabla_\psi J_Q(\psi) = \nabla_\psi Q_\psi(s_t, \alpha_t)(Q_\psi(s_t, \alpha_t) - r_t + \gamma V_{\bar{\phi}}(s_{t+1})). \tag{11}$$

Here $\bar{\phi}$ is the parameter of the target value network, which is the exponential moving average of ϕ.

The policy could be a Gaussian with mean and variance given by neural networks. Instead of directly maximizing the performance given in Equation (7), SAC uses KL-divergence to update its policy parameter θ. An approximation for the policy gradient, i.e., $\nabla_\theta J_\pi(\theta)$, can be found in [29].

Algorithm 2 shows different steps of SAC. We call it in Algorithm 1 to update the policy parameters.

G. Baseline Algorithms

Centralized method [9]: solves the centralized optimization problem explained in Section IV-E.
AIMD [17]: is the basic AIMD algorithm with fixed rates for additive increase and multiplicative decrease parts. The algorithm is similar to Algorithm 1 except that α is fixed. In particular, when there is no congestion it increases the charge power by one unit. If congestion is detected, it halves the current charge power.
Direct Rate Learning [14]: is an algorithm that tries to learn the charge power directly from interaction with the environment instead of learning the additive increase and multiplicative decrease rates. This algorithm also has two training phases similar to the proposed A-AIMD algorithm but the learned action is the charge power x rather than the parameter α.
Earliest Deadline First (EDF) [30]: which simply prioritizes charging EVs based on their deadlines; thus, EVs with earlier deadlines are charged first at the maximum power supported by their charging points. To fully utilize the network at any given time, we admit as many EVs as possible without causing congestion.
Least Laxity First (LLF) [30]: is similar to EDF except that it prioritizes charging EVs based on their *laxity* i.e., the amount of time left to meet the charging demand divided by the deadline should the EV be charged at the maximum power. EVs with lower laxity are charged first at the maximum power supported by their charging points. At any point in time, we charge as many EVs as possible without causing congestion.

V. Case Study

We used PyTorch to implement our RL agents and OpenAI Gym [31] wrapper to implement the environment. We considered 50 and 30 episodes for offline pre-training and online interaction, respectively. We witnessed that after only 25 episodes (i.e., 25×144 time steps) A-AIMD learns the optimal policy. However, DRL needs more episodes to converge. For A-AIMD, we set A and B to 2 using the grid search method. In all scenarios, unless otherwise stated, we run simulation for 1 month (30 days) and report the average result.

We consider Gaussian policy with μ and σ as neural networks. Both networks have one hidden layer (besides input and output layers) with 256 nodes, ReLU activation function, and mean squared error loss function.

A. Performance Metrics

We use the following performance metrics to evaluate the A-AIMD control algorithm:
Jain index is used to evaluate fairness of a power allocation scheme (a higher value suggests a fairer scheme). It is defined as $J(x) = \frac{\left(\sum_i x_i\right)^2}{n \sum_i x_i^2}$ where x is a vector of charge powers of all charging points in a given time slot and n is the number of active charging points. The index is calculated for every time slot and then averaged over all time slots in one day.
Percentage of EVs with a certain SoC is the percentage of connected EVs with a state-of-charge (SoC) greater than or equal to a certain threshold at departure time.
Resource utilization is the loading of a transformer (in MVA).
Turnaround time is the interval between the arrival of an EV and its charge completion time.

B. Performance Evaluation

Consider the 33-bus system [22] shown in Figure 1. We connect one or more homes to each of the 1815 end nodes of the low-voltage feeders to create the base load. Each of these homes can have one EV, hence multiple EVs may be connected to a single end node. To model the home loads, which are inelastic and cannot be controlled, we utilize sample household demands (with 1-second resolution) from the ADRES-CONCEPT[2] project [32]. The dataset contains power consumption of 30 households over two weeks, hence

[2]The data was generated in the research project "ADRES-Concept" (EZ-IF: Development of concepts for ADRES- Autonomous Decentralized Regenerative Energy Systems, project no. 815 674). This project was funded by the Austrian Climate and Energy Fund and performed under the program "ENERGIE DER ZUKUNFT".

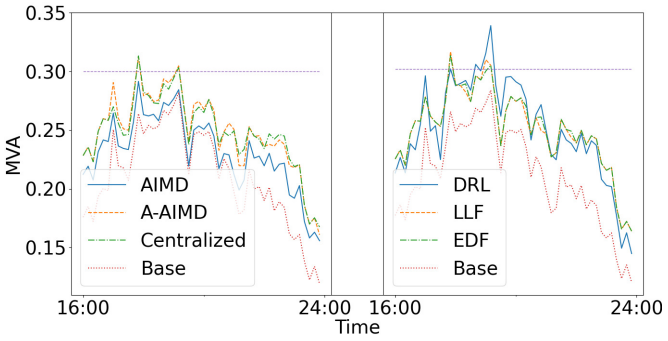

Fig. 4: Loading of a distribution transformer under centralized, A-AIMD and AIMD algorithms (right figure), and DRL, LLF and EDF strategies (left figure). The dashed horizontal line is the rated capacity and the red curve is the base load.

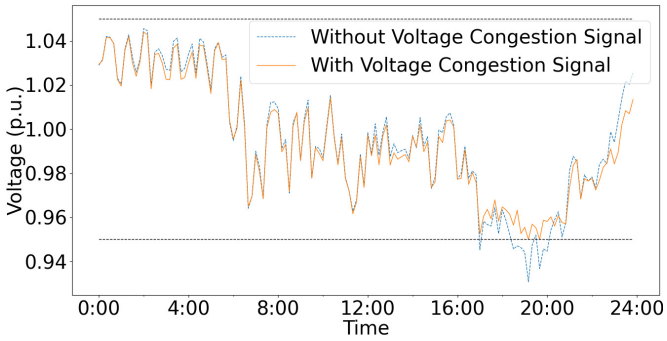

Fig. 5: The voltage of one of the primary nodes of the distribution network using A-AIMD with and without considering voltage congestion signals. The horizontal lines indicate $\pm 5\%$ of the nominal voltage.

we will get 420 unique load profiles when they are split into 1-day segments. We chose this dataset because it has both active and reactive power consumption of homes which allows for running more realistic simulations. To obtain more unique load profiles, we add white Gaussian noise with 10% standard deviation to each sample to create a total of 8400 unique household load data.

We consider a maximum of 500 EVs in our distribution system, although the actual number of connected EVs depends on their mobility and SoC. The arrival times of EVs are generated from the data provided in the Pecan Street dataset [33]. Since there are only about 80 EVs in this dataset, we take the daily arrival times of these EVs over one year (2016) and fit a Gaussian Mixture Model (GMM) to the probability distribution function and sample the daily arrival times from the fitted GMM. The dataset does not include EV departure times, so we randomly sample the sojourn time of each EV from a Gaussian distribution with $\mu = 8$ and $\sigma = 2$ (in hours). We then calculate the departure time of each EV by adding their sojourn time to their arrival time. We disconnect EVs when they are fully charged or manually unplugged (whichever happens first). We randomly select the battery size of each EV from four of the popular models in the market: a) $16kWh$ for Chevy Volt and Mitsubishi iMiEV, b) $30kWh$ for Nissan Leaf, c) $42kWh$ for BMW i3, and d) $75kWh$ for Tesla Model 3. The initial SoC of each EV at the arrival time is sampled uniformly between 0 and 0.1. We consider 10-minute time intervals and run a day-long simulation for each episode.

1) Addressing Transformer Overloading Problem: Figure 4 shows loading of a distribution transformer during the peak time when charging points are controlled using the centralized, A-AIMD, AIMD, DRL, LLF, and EDF algorithms. We observe that in all cases except AIMD, there are some small overshoots which we attribute to changes in residential loads. The reason that there are overshoots in case of A-AIMD but no overshoots in case of AIMD is that when there is no congestion, A-AIMD tries to increase the charge power more aggressively than AIMD. However, as soon as the network is congested, A-AIMD quickly responds to the congestion signal and we do not have any sustained overloading in the system. We see larger overshoots for DRL, which we attribute to the large number of episodes required to train the RL agents.

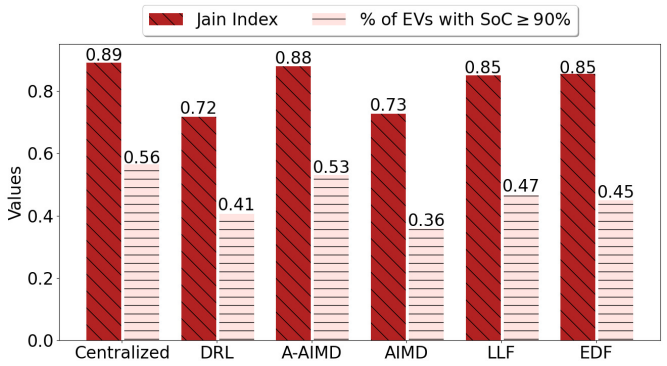

Fig. 6: Comparing the performance of A-AIMD (with imitation learning) and baseline algorithms.

2) Voltage Control: Figure 5 shows the voltage of one of the primary nodes under A-AIMD with and without voltage congestion signals. A congestion signal is sent to the RL agents when the nodal voltage level goes below 0.95 p.u. or above 1.05 p.u.. As shown in the figure, A-AIMD can effectively respond to voltage congestion signals by reducing the charge power of charging points downstream of the node with the voltage problem. As discussed earlier, the centralized optimization based controller is not capable of dealing with voltage problems as it ignores reactive power flows. The same observation can be made for EDF and LLF algorithms.

3) Fairness & User Satisfaction: Figure 6 presents the comparison between different algorithms in terms of fairness and the number of EVs charged up to 90% of their battery capacity. It can be seen that A-AIMD closely follows the centralized method and outperforms all the other methods. We also notice that both AIMD and DRL fail to outperform EDF and LLF, which merely rely on an admission control scheme.

4) Adaptive Nature of A-AIMD: Figure 7 illustrates the adaptive nature of A-AIMD. The top figure shows the charge power of an EV charger when controlled by AIMD and A-AIMD algorithms. The middle figure shows the values of the parameter α learned by A-AIMD. The bottom figure shows the power loading and rated capacity (dashed line) of one of the phases of the transformer that feeds this EV and many other EVs. The EV arrives and connects to the charging point around 6:00pm. Given the available capacity, both A-AIMD

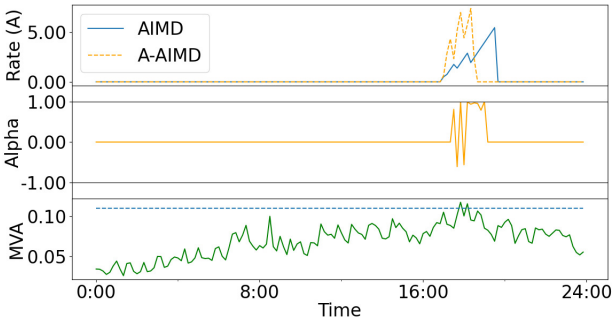

Fig. 7: Comparing rate changes under AIMD and A-AIMD.

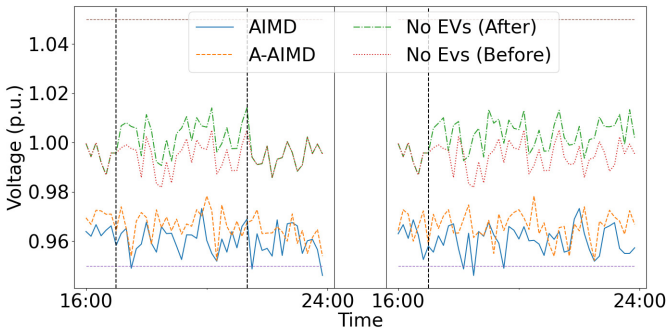

Fig. 8: Voltage profile of Node 18 under A-AIMD and AIMD algorithms. In the left plot the substation voltage increases between 5:00pm and 9:00pm (marked by dashed vertical lines). In the right plot the configuration is changed (reconfiguration A) at 5:00pm (marked by the dashed vertical line).

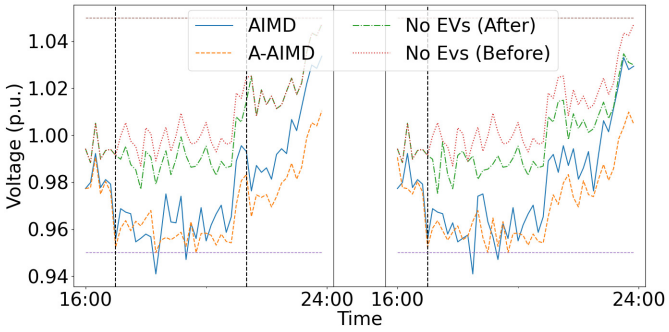

Fig. 9: Voltage profile of Node 33 under A-AIMD and AIMD algorithms. In the left plot the substation voltage decreases between 5:00pm and 9:00pm (marked by dashed vertical lines). In the right plot the configuration is changed (reconfiguration B) at 5:00pm (marked by the dashed vertical line).

and AIMD start to increase the charge power upon the arrival of this EV. However, A-AIMD quickly increases its additive increase rate. We witness that when there is congestion, α becomes negative and A-AIMD reduces the charge power of the EV to relieve congestion. AIMD also decreases the charge power multiplicatively to avoid congestion. The differences in additive and multiplicative rates cause A-AIMD to fully charge the EV battery much earlier than AIMD.

5) Resource Utilization: Results obtained for a day-long simulation period indicate that A-AIMD increases the total energy delivered to EVs by 4% on average compared to AIMD by better utilizing the available capacity of the system. Both methods overshoot the rated capacity of transformers for short periods of time, but quickly react to the received congestion signal by reducing the charge power of connected EVs. The A-AIMD algorithm results in a few more overshoots compared to AIMD, but none of them causes a sustained overloading which could damage the transformer.

6) Turnaround Time: Table II compares the average turnaround times of different control algorithms. The average is taken over all EV charging sessions in one month. As expected, EDF and LLF yield lower average turnaround time than AIMD and A-AIMD algorithms as they are sorting-based power allocation algorithms that charge EVs according to their deadline. Nevertheless, the difference between the turnaround times of A-AIMD and EDF is less than 90 minutes on average. DRL has lower turnaround time than A-AIMD because it frequently exceeds the rated capacity of the transformer (as it cannot avoid congestion at all times).

C. Adaptability to Changes in Distribution Network

Distribution system operators frequently adjust the tap position of tap-changing transformers or change the configuration of the network for voltage regulation, loss reduction, etc. Thus, we consider two scenarios to study how our method can adapt to such changes in the network.

1) Scenario A – Adjusting Substation Voltage: We change the voltage level of the substation transformer during a certain period of time to see whether A-AIMD can still adapt its update rules to avoid voltage problems. For the first case, the voltage of the substation is increased between 5:00pm and 9:00pm. This will increase the voltage level, so we expect to see an increase in charge power. For the second case, the voltage of the substation is decreased between 5:00pm and 9:00pm to create more congestion in the network. Figure 8 (left panel) and Figure 9 (left panel) show the voltage profile of one of the primary nodes of the distribution network for A-AIMD and AIMD algorithms in the two cases. To show how the changes would have affected the voltage profiles without the EV charging load, we plot the voltage magnitude of Nodes 18 and 33 without having any EVs in the system, before and after changing the substation voltage (left panels) and reconfiguration (right panels). It can be seen that both AIMD and A-AIMD can avoid congestion during this time. We observe some minor voltage limit violations for AIMD which are not present for A-AIMD. The reason is that, unlike AIMD, A-AIMD learns to update the charge power less aggressively.

Figure 10 (left panel) shows the performances of AIMD and A-AIMD algorithms with and without the voltage increase at the substation. As the increase in voltage level makes more room for charging EVs, the performances of both algorithm improves in this case. Figure 11 (left panel) shows the performances of AIMD and A-AIMD algorithms with and without the voltage decrease at the substation. As the decrease in voltage level reduces the room for charging EVs, the performances of both algorithm degrades in this case. Expectedly, A-AIMD has better performance than AIMD before and after the tap change in both cases.

2) Scenario B – Reconfiguration: To further evaluate the ability of A-AIMD to adapt to varying network conditions, we consider two simple reconfiguration events. In both cases,

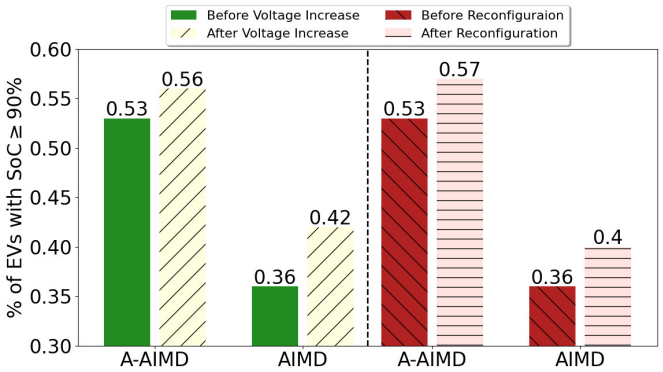

Fig. 10: Performances of A-AIMD and AIMD before and after voltage increase (left panel) and reconfiguration A (right panel). Note that the y-axis is truncated.

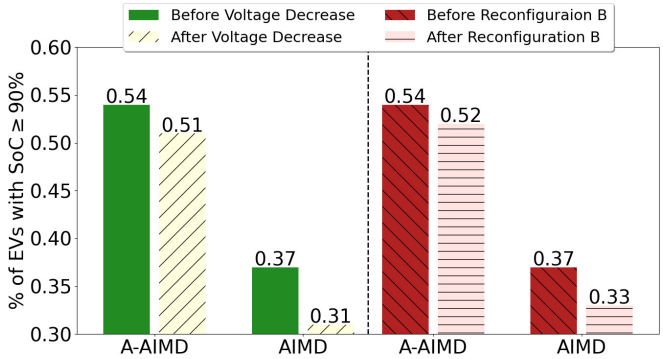

Fig. 11: Performances of A-AIMD and AIMD before and after voltage decrease (left panel) and reconfiguration B (right panel). Note that the y-axis is truncated.

TABLE II: Average turnaround time (in hours) for all algorithms. The average is taken over all EV charging sessions.

Algorithms	A-AIMD	AIMD	DRL	Centralized	LLF	EDF
Turnaround Time	5.9	6.2	5.1	5.8	4.9	4.6

the reconfiguration changes the network topology from what was used to train the RL agents. In the first event (which we refer to as *Reconfiguration A*), we close the (normally open) tie switch between Nodes 12 and 22 at 5:00 pm and open the (normally closed) sectionalizing switch between Nodes 11 and 12 to maintain the tree structure. In the second reconfiguration (*Reconfiguration B*), we close the (normally open) tie switch between Nodes 18 and 33 at 5:00pm and open the (normally closed) sectionalizing switch between Nodes 31 and 32.

Figure 8 (right panel) and Figure 9 (right panel) respectively show the voltage profile of one of the phases at Node 18 and Node 33 under AIMD and A-AIMD control algorithms with Reconfiguration A and B. While both algorithms avoid congestion successfully, A-AIMD quickly perceives the updated state and learns to update the charge power less aggressively.

Figure 10 (right panel) depicts the performance of AIMD and A-AIMD algorithms before and after Reconfiguration A. As this reconfiguration improves the overall voltage profile of the network, there is more room for charging EVs. Thus, the performance of both algorithms improves. Figure 11 (right panel) depicts the performance of AIMD and A-AIMD algorithms before and after Reconfiguration B. This time, as the reconfiguration exacerbates the voltage congestion problem, there is less available capacity for charging EVs. Thus, the performance of both algorithms degrades. We note that in both reconfiguration scenarios A-AIMD outperforms AIMD as it learns to update the additive and multiplicative factors.

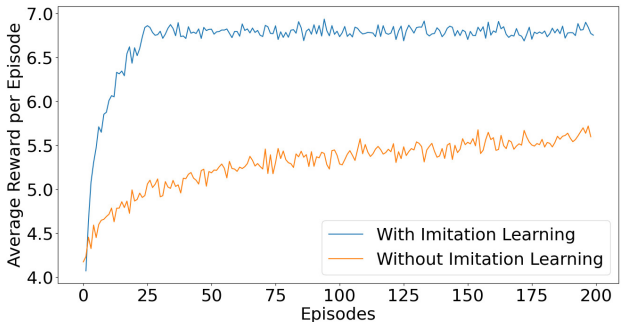

Fig. 12: Average reward per episode (in MVA) of A-AIMD achieved with and without imitation learning. Here the average is taken over all time steps of an episode and all active agents.

D. Imitation Learning & Reducing Sample Complexity

Figure 12 shows the average reward per episode required by A-AIMD in the online interaction phase with and without imitation learning in the offline training phase. It can be seen that with imitation learning, A-AIMD requires around 25 episodes to converge. But without using imitation learning in the offline training phase the agent does not achieve a stable reward after 200 episodes and more episodes will be needed to achieve a comparable reward. Considering the performance metrics presented in Section V-A, we observe that after only 30 episodes, A-AIMD with imitation learning can attain a Jain index of 0.88 and bring the SoC of 53 percent of all the EVs to 90% or above by their departure time. But without imitation learning, A-AIMD achieves comparable performance at 200 episodes, i.e., attaining a Jain index of 0.86 and increasing the SoC of 52 percent of EVs to 90% or above by their departure time. This underlines the importance of using imitation learning to speed up the learning process.

VI. Conclusion and Future Work

Rule-based algorithms mimicking the behaviour of Internet congestion control schemes have been successfully applied to control charging of EVs in the power grid. These control algorithms are decentralized, do not assume the knowledge of the distribution network model, and only require measurements obtained from sensors installed beyond the substation and a trivial signalling mechanism; these properties make them ideal for real-world development. Yet there are fundamental differences between the Internet and the power grid, e.g., the timescale of control, which call for improving the performance by adjusting the control rules dynamically.

In this work we proposed a novel approach based on multi-agent reinforcement learning to adapt the parameter of the modified AIMD algorithm to the network condition. Our simulation results confirmed that the resulting algorithm (A-AIMD) tracks the available capacity of the network in real

time, prevents sustained overloading and voltage limit violation problems, and outperforms decentralized feedback control algorithms (DRL and AIMD) and sorting-based algorithms (EDF and LLF) in terms of fairness and utilization. Additionally, it can adapt to changes in the distribution network such as transformer tap changes and feeder reconfiguration events.

This work has several limitations we plan to address in future work. We do not have theoretical results to show that A-AIMD yields a fair allocation to connected EVs. It is assumed that a voltage congestion signal is only sent to downstream chargers, but other chargers can be notified too to address voltage limit violations. We also focused on maximizing the total charge power without considering the electricity price. We intend to incorporate dynamic pricing in our future work. Lastly, we believe that with a simple modification this algorithm can deal with overvoltage problems in distribution grids with a high penetration of solar photovoltaics.

REFERENCES

[1] I. E. Agency. (2019) Global EV Outlook 2019. [Online]. Available: https://www.iea.org/reports/global-ev-outlook-2019
[2] R. Moghaddas *et al.*, "Smart control of fleets of electric vehicles in smart and connected communities," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6883–6897, 2019.
[3] R. Mehta *et al.*, "Hybrid planning method based on cost-benefit analysis for smart charging of plug-in electric vehicles in distribution systems," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 523–534, 2019.
[4] M. Yilmaz and P. T. Krein, "Review of the impact of vehicle-to-grid technologies on distribution systems and utility interfaces," *IEEE Tran. Power Electronics*, vol. 28, no. 12, pp. 5673–5689, 2013.
[5] M. Liu *et al.*, "Customer-and network-aware decentralized EV charging control," in *Power Systems Computation Conf.* IEEE, 2018, pp. 1–7.
[6] ——, "Decentralized charging control of electric vehicles in residential distribution networks," *IEEE Trans. Control Systems Technology*, vol. 27, no. 1, pp. 266–281, 2019.
[7] A. Zishan *et al.*, "Reputation-based fair power allocation to plug-in electric vehicles in the smart grid," in *Proceedings of the 11th Int. Conf. on Cyber-Physical Systems*. ACM/IEEE, 2020, pp. 63–74.
[8] O. Ardakanian *et al.*, "On identification of distribution grids," *IEEE Trans. Control of Network Systems*, vol. 6, no. 3, pp. 950–960, 2019.
[9] O. Ardakanian, S. Keshav, and C. Rosenberg, "Real-time distributed control for smart electric vehicle chargers: From a static to a dynamic study," *IEEE Trans. Smart Grid*, vol. 5, no. 5, pp. 2295–2305, Sept 2014.
[10] S. Stüdli *et al.*, "AIMD-like algorithms for charging electric and plug-in hybrid vehicles," in *Int. Electric Vehicle Conf.* IEEE, 2012, pp. 1–8.
[11] A. Alyousef and H. de Meer, "Design of a TCP-like smart charging controller for power quality in electrical distribution systems," in *10th Int. Conf. on Future Energy Systems*. ACM, 2019, pp. 128–138.
[12] D. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Computer Networks and ISDN Systems*, vol. 17, no. 1, pp. 1–14, 1989.
[13] W. Li *et al.*, "QTCP: Adaptive congestion control with reinforcement learning," *IEEE Trans. Network Science and Engineering*, vol. 6, no. 3, pp. 445–458, 2019.
[14] A. Zishan, M. Moghimi, and O. Ardakanian, "Adaptive control of plug-in electric vehicle charging with reinforcement learning," in *Proceedings of the 11th Int. Conf. on Future Energy Systems*. ACM, 2020.
[15] M. Liu and S. McLoone, "Enhanced AIMD-based decentralized residential charging of EVs," *Trans. of the Institute of Measurement and Control*, vol. 37, no. 7, pp. 853–867, 2015.
[16] E. Ucer, M. Kisacikoglu, and M. Yuksel, "Analysis of AIMD algorithm for EV charging," in *Proceedings of the Tenth ACM Int. Conf. on Future Energy Systems*. ACM, 2019, pp. 436–438.
[17] E. Ucer *et al.*, "An internet-inspired proportional fair EV charging control method," *IEEE Systems Journal*, vol. 13, no. 4, pp. 4292–4302, 2019.
[18] N. Sadeghianpourhamami *et al.*, "Definition and evaluation of model-free coordination of electrical vehicle charging with reinforcement learning," *IEEE Trans. Smart Grid*, vol. 11, no. 1, pp. 203–214, 2020.
[19] S. Vandael *et al.*, "Reinforcement learning of heuristic EV fleet charging in a day-ahead electricity market," *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 1795–1805, 2015.
[20] J. L. A. Chis and V. Koivunen, "Reinforcement learning-based plug-in electric vehicle charging with forecasted price," *IEEE Trans. Vehicular Technology*, vol. 66, no. 5, pp. 3674–3684, 2017.
[21] W. Shi and V. Wong, "Real-time vehicle-to-grid control algorithm under price uncertainty," in *Int. Conf. on Smart Grid Communications (SmartGridComm)*. IEEE, 2011, pp. 261–266.
[22] M. E. Baran and F. F. Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Trans. Power Delivery*, vol. 4, no. 2, pp. 1401–1407, 1989.
[23] AMPS Distribution System Analysis Subcommittee. IEEE PES distribution systems analysis subcommittee, radial test feeders. [Online]. Available: http://sites.ieee.org/pes-testfeeders/resources/
[24] S. Bansal *et al.*, "Plug-and-play model predictive control for electric vehicle charging and voltage control in smart grids," in *53rd Conf. on Decision and Control*. IEEE, 2014, pp. 5894–5900.
[25] K. Zhang *et al.*, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *arXiv:1911.10635*, 2019.
[26] W. Sun, J. A. Bagnell, and B. Boots, "Truncated horizon policy search: Combining reinforcement learning & imitation learning," *Int. Conf. on Learning Representations*, 2018.
[27] Y. Yue and H. M. Le, "Imitation learning," online, 2018, Tutorials of the Thirty-fifth Int. Conf. on Machine Learning.
[28] R. S. Sutton and A. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
[29] T. Haarnoja *et al.*, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *arXiv preprint arXiv:1801.01290*, 2018.
[30] Z. Lee, D. Johansson, and S. Low, "ACN-Sim: An open-source simulator for data-driven electric vehicle charging research," in *Int. Conf. on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2019, pp. 1–6.
[31] G. Brockman *et al.*, "OpenAI Gym," *arXiv preprint arXiv:1606.01540*, 2016.
[32] ADRES-Concept. ADRES dataset. https://www.ea.tuwien.ac.at/projects/adres_concept/EN/. [Online].
[33] P. Street. Pecan Street dataset. https://dataport.pecanstreet.org/. [Online].

Abdullah Al Zishan completed his MSc degree in Computing Science from the University of Alberta, Edmonton, Canada. Currently, he is a Research Assistant in the Department of Computing Science at the University of Alberta. His research interest spans scheduling problems, resource allocation and storage systems in the smart-grid. He has several publications in IEEE and ACM conferences.

Moosa Moghimi Haji received his Ph.D. degree in energy systems from the University of Alberta in 2017. He joined the Computing Science department of the University of Alberta as a postdoctoral fellow in 2019. His research interests include power distribution systems analysis and control, application of artificial intelligence in power systems, and smart grid.

Omid Ardakanian (M'15) received his PhD degree from the University of Waterloo, ON, Canada. He is currently an Assistant Professor with the Department of Computing Science, University of Alberta, Edmonton, AB, Canada. He served as a Guest Editor of IEEE Transactions on Smart Grid, Special Section on Theory and Application of PMUs in Power Distribution Systems, and is currently on the Executive Committee of ACM Special Interest Group on Energy (SIGEnergy). He received several best paper awards from IEEE and ACM conferences.