Transactive Electric Vehicle Agent: A Deep Reinforcement Learning Approach

Swastik Sharma
Department of Electrical Engineering
Indian Institute of Technology Kanpur
Kanpur, India
swastiks21@iitk.ac.in

Swathi Battula
Department of Electrical Engineering
Indian Institute of Technology Kanpur
Kanpur, India
swathi@iitk.ac.in

Sri Niwas Singh
Department of Electrical Engineering
Indian Institute of Technology Kanpur
Kanpur, India
snsingh@iitk.ac.in

***Abstract*—Bid-based Transactive Energy System (TES) designs for distribution systems have several merits, including fair compensation of participating Distributed Energy Resources (DERs) where they can voice their preferences using bids/offers. In this research, a Deep Reinforcement Learning (DRL)-based Proximal Policy Optimization (PPO) algorithm with recurrent neural networks has been employed to derive bids/offers from participating Electric Vehicle (EV) agents by modelling the problem as a Partially Observable Markov Decision Process (POMDP). This helps in preserving the non-linear characteristics of the problem, i.e. obtaining DERs' price-sensitive bids/offers while considering user goals and constraints, owing to the highly efficient function approximation characteristics of Deep Neural Networks (DNNs) employed in DRL-based algorithms. The obtained results demonstrate the convergence of the agent's policy in real-world data having dynamic price fluctuations.**

***Index Terms*—Transactive energy, reinforcement learning, electric vehicles, distribution systems, proximal policy optimization**

I. Introduction

Today, electric grids worldwide are witnessing rampant technological advancements changing how we perceive and value our energy requirements. Access to data analytics on energy usage using smart meters, conversion of consumers' role to that of prosumers, and the use of Information and Communications Technology (ICT) have revolutionised the way we interact as customers of electricity. With the advent of the Transactive Energy System (TES) designs – a system of economic and control mechanisms that allows value-based transactions to achieve dynamic demand-supply balance – customers are given a higher autonomy in the decisions regarding their energy usage [1]. The bid-based TES designs help customers owning Distributed Energy Resources (DERs) voice their goals/constraints through bids/offers [2].

TES-based distribution systems are well-suited for Electric Vehicle (EV) charging infrastructure due to their flexible energy requirements. Their ability to charge or discharge batteries based on price signals makes them ideal candidates for bid-based TES designs. Several studies have applied a TES-based architecture to distribution systems involving EVs for an array of problems that focus on charge-discharge scheduling [3], [4], voltage control & unbalance [5], [6], network-constrained coordination [7], [8]. In [3], [4], optimal allocation of charge-discharge schedules for the EVs is done by minimising the total cost incurred in a transactive framework. However, a customer-centric design that involves active participation through bids/offers is absent, which can aid in improving overall system operations. Studies by Hoque et al. in [5], [7] employed a model-based bid/offer function for EVs to charge/discharge in TES-based distribution system designs for voltage control and network-aware coordination respectively. Saber et al. in [6], [8] also used a similar model-based bid/offer function to optimise EVs' charge/discharge schedules. Current literature is limited to model-based bid/offer functions, facing several limitations. First, the inherent non-linearities in the problem are usually linearised. Second, the stochastic modelling techniques face difficulties in accurately modelling the randomness of EV behaviour. Finally, the optimization techniques employed are often time-intensive, hindering application in real-time decision-making and large-scale simulations [9]. Model-free techniques such as *Deep Reinforcement Learning (DRL)* can tackle these challenges as they combine the powerful & efficient data processing capability of Deep Neural Networks (DNNs) and the decision-making ability of Reinforcement Learning (RL) algorithms. Authors in [10]–[12] have employed DRL for control-based tasks such as charge/discharge scheduling for EVs and have shown improved performance. However, in [10], [11], charge/discharge schedules are determined using forecasted price information, and in [12], time-of-use prices are used for the same.

The focus of the aforementioned studies is not on EVs participating in a bid-based TES design for distribution systems - where establishing retail prices is a non-trivial task. Moreover, accurate predictions of these retail prices become challenging as they depend on several factors, including the DERs' bids/offers, which are intricately linked with the market clearing process. Therefore, in such sequential decision-making problems, traditional predict-then-optimize algorithms become impractical where we predict unknown parameters and then solve the optimization problem [13]. To address this, we employ *Paritally Observable Markov Decision Process* (POMDP) based formalism for modelling the problem, which is a special form of Markov Decision Process (MDP) where the agent has access to the environment but cannot directly observe the system dynamics for improved agent's decision making under uncertainty.

979-8-3503-8183-2/24/$31.00 ©2024 IEEE

The contributions of this research are enumerated below:

(a) Implementation of a model-free DRL-based EV agent to bid/offer on behalf of the customers and preserve the non-linear characteristics of the problem.
(b) Modelling the problem with POMDP-based formalism for robust decision-making of the DRL-based EV agent under uncertainty.
(c) Assessment of the proposed DRL-based EV agent using real-world data.

II. DRL-BASED EV AGENTS IN TES-BASED DISTRIBUTION SYSTEMS

A. Problem description

This research employs a bid-based TES design for distribution systems, where the bids/offers (x_t (¢/kWh)$, p_t$ (kW)) by a DRL EV agent go through a hierarchical aggregation, as depicted in Fig. 1. The bids/offers placed by the EV agents are aggregated at the charging station level, which is then sent to the Distribution System Operator (DSO). The DSO aggregates bids/offers from all such DERs/aggregators and determines retail prices (x_t^{ret}(¢/kWh)) after participating in the Wholesale Power Market (WPM). Here, the focus of this research is limited to the development of customer-centric DRL EV agents participating on behalf of the owners and performing the following to meet goals subject to constraints.

(a) Placing bids for charging to reach a specified desired State of Charge (SoC_{des}) by the end of plug-in duration.
(b) Placing offers for discharging EV battery for monetary benefits.

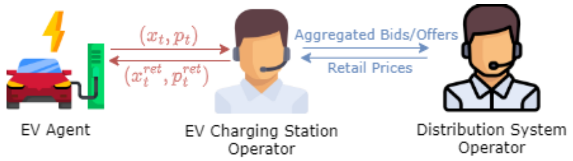

Fig. 1. Heirarchical TES-based distribution systems design [1]

B. Formulation of Markov decision process

We employ POMDP-based formalism to materialise the objective of the proposed problem. Since the proposed problem is a sequential decision-making problem where past decisions regarding charge, discharge, or not to charge influence future ones, we utilize *Long Short-Term Memory* (LSTM) Neural Networks (NNs). LSTMs help in preserving information about long-term dependencies, allowing them to extract valuable insights from the history of the data – a series of observations.

The environment which comprises an EV agent and the transactive framework through which DSO sets the retail prices can, therefore, be characterised as a discounted episodic POMDP defined by $(\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{P}, \mathcal{R}, \mathcal{Z}, \gamma)$, where $\mathcal{S}$ is defined as the state space; $\mathcal{A}$ is the action space; $\mathcal{O}$ is the observation space; $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the state transition probability distribution; $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function; $\mathcal{Z} : \mathcal{S} \times \mathcal{A} \times \mathcal{O} \rightarrow [0, 1]$ is the observation function denoting the probability of observing an observation after taking action for an agent transitioning from present state to the next state and $\gamma \in [0, 1]$ is the discount factor.

1) State: The state is a series of observations, where O_t is the agent's observation at the time-step t, given by:

$$O_t = (SoC_{t-1}, t_r, x_{t-1}^{ret}, SoC_{des}) \tag{1}$$

where SoC_{t-1} is the State of Charge (SoC) of the EV agent at the end of the time step $t-1$, t_r is the remaining time for the EV to disconnect from the charger, x_{t-1}^{ret} is the retail market price for time-step $t-1$, and SoC_{des} is the desired SoC at the end of plug-in duration.
Now state s_t can be denoted as:

$$s_t = (O_0, O_1, ..., O_t) \tag{2}$$

2) Action: The agent, for each time step, t, comes up with a bid/offer price, x_t (¢/kWh) and a bid/offer volume, p_t (kW) for its participation in TES design. The agent's action for t:

$$a_t = (x_t, p_t) \tag{3}$$

the action, a_t, has a continuous space, wherein x_t takes both negative and positive values denoting discharging and charging, respectively, and p_t takes positive values.

3) Rewards: At the end of any time-step $t \leq t_{end}$, where t_{end} represents the last time-step of the plug-in duration, the rewards are given as:
if bid/offer is `cleared`:

$$r_t = \begin{cases} -p_t \times (x_t + \phi \cdot \eta_c) \times \Delta t, & \text{Charging} \\ -p_t \times (x_t + \frac{\phi}{\eta_d}) \times \Delta t, & \text{Discharging} \end{cases} \tag{4}$$

if bid/offer is `not cleared`:

$$r_t = -|p_t \times x_t| \cdot \Delta t \tag{5}$$

where Δt is the duration of a time-step. The parameter, ϕ, is the degradation cost associated with the battery in ¢/kWh. It can be represented as:

$$\phi = \phi_0 + \phi_u \tag{6}$$

where ϕ_0 (¢/kWh) represents the actual degradation cost of the battery, and ϕ_u (¢/kWh) represents the additional utility the user derives from the battery. η_c and η_d are the charging and discharging efficiencies of the battery, respectively.
At $t = t_{end}$, additional rewards are given to the agent as below:

$$r_t = \begin{cases} -(SoC_t - SoC_{des})^2 \times \beta, & SoC_t \leq SoC_{des} \\ 0, & SoC_t > SoC_{des} \end{cases} \tag{7}$$

Here, β is the weight factor that represents the agent's maximum willingness to pay for the EV to be at SoC_{des} at t_{end}.

The SoC is updated at the end of t by comparing x_t and x_t^{ret} as follow:
if $x_t \geq x_t^{ret}$:

$$SoC_t = SoC_{t-1} + \eta_c P_t \Delta t \tag{8}$$

else if $x_t \leq 0$ and $x_t \geq -x_t^{ret}$:

$$SoC_t = SoC_{t-1} - \frac{P_t}{\eta_d} \Delta t \tag{9}$$

where, $P_t = min(max(p_t, 0), P_{max})$

C. *Proximal policy optimisation with recurrent networks*

To address the continuous and high dimensional state and action spaces, a policy gradient algorithm called Proximal Policy Optimization (PPO) is employed. PPO directly predicts the policy the agent must follow to reap higher rewards. The policy is represented as $\pi(a_t|s_t,\theta)$ where $\pi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, is parameterised by θ shows the probability distribution of taking action, a_t, based on the state, s_t, of the agent. PPO is highly sample efficient and can learn near-to-optimal policy with only a few interactions with the environment. It is based on the actor-critic architecture, where the actor decides the actions to take, and the critic estimates the performance of these actions. In PPO, a clipped surrogate objective is used to limit the policy updates to a "trust region", preventing the policy, $\pi(a_t|s_t,\theta)$ from deviating too much from the old policy $\pi(a_t|s_t,\theta_{old})$ ensuring gradual and stable learning [14].

In the proposed implementation, the actor-critic network takes the state shown in (2) as input and outputs ($\mu(s_t,\theta)$, $\sigma(s_t,\theta)$), and $V_\pi(s_t,\theta)$, where $\mu(s_t,\theta)$ is the mean vector and $\sigma(s_t,\theta)$ is the standard deviation vector of the $\pi(a_t|s_t,\theta)$, which is characterised by a normal distribution, $\mathcal{N}(\mu(s_t,\theta),\sigma(s_t,\theta))$, and $V_\pi(s_t,\theta)$, is the total expected reward starting from state s_t and following policy π thereafter.

The representation of the actor-critic network is depicted in Fig. 2. LSTMs, as discussed earlier, are used to extract information regarding the long-term dependencies of the history of observations. A shared parameter-based NN architecture where the output is policy (actor) and value function (critic) is employed to reduce memory and computation time.

The PPO Agent performs several rollouts in the environment for a fixed number of time steps, T using $\pi(a_t|s_t,\theta_{old})$ and saves the trajectories $(O_0, a_0, r_0, \cdots, O_T, a_T, r_T)$ in a buffer with size, `buffer_size`. These trajectories are sampled as mini-batches to train the policy network by using the clipped PPO objective ($L_t^\pi(\theta)$) given by:

$$L_t^\pi(\theta) = \hat{\mathbb{E}}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)] \quad (10)$$

where $r_t(\theta)$ is the probability ratio of policies after and before the parameter updates, given by:

$$r_t(\theta) = \frac{\pi(a_t|s_t,\theta)}{\pi(a_t|s_t,\theta_{old})} \quad (11)$$

ϵ is the clipping parameter that ensures that r_t, doesn't move outside the trust region interval of $[1-\epsilon, 1+\epsilon]$. A minimum of the clipped and unclipped objectives is taken to provide a pessimistic bound over the unclipped objective function [14]. $\hat{A}_t$ is the Generalised Advantage Estimates (GAE) [15] given by:

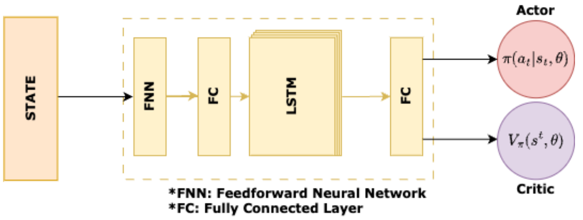

Fig. 2. Schema of the actor-critic network

$$\hat{A}_t(\gamma,\lambda) = \sum_{l=0}^{T-t-1} (\gamma\lambda)^l(\delta_{t+l}) \quad (12)$$

here, δ_{t+l} is the Temporal Difference (TD) residual of the value function, V, with discount factor γ given by:

$$\delta_t = r_t + \gamma V_\pi(s_{t+1}) - V_\pi(s_t) \quad (13)$$

The $\lambda \in [0,1]$ controls the bias and variance trade-off [15].

The loss function contains two more terms with the clipped surrogate objective at (10). First is the value function loss ($L_t^V(\theta)$) given by:

$$L_t^V(\theta) = \hat{\mathbb{E}}_t(V_\theta(s_t) - V_t^{tgt})^2 \quad (14)$$

where $V_t^{tgt} = r_t + \gamma V_\theta(s_t)$. The other term is entropy, which helps provide sufficient exploration, given by $S[\pi_\theta](s_t)$. The loss functions and the entropy are combined, and the expectation in (10) is replaced by its monte-carlo estimate as follows:

$$L_t^{\pi+V+S}(\theta) = \frac{1}{T}\sum^{T}[L_t^\pi(\theta) - c_v L_t^V(\theta) + c_s S[\pi_\theta](s_t)] \quad (15)$$

here, c_v, c_s are coefficients that control the participation of the value function and entropy, respectively. This loss function $L_t^{\pi+V+S}(\theta)$ is optimised for N optimization epochs using learning rate α, with sampled mini-batch data from the buffer. This process is continued until convergence.

III. DRL Framework related Preliminaries

1) Performance enhancement for PPO: First, to enhance the learning efficiency of the agent since the environment is complex and non-stationary, an early stopping method based on *Kullback-Leibler* (KL) divergence is adopted. This approach halts the optimization epochs when the KL divergence between the old policy and new policy-based probability distribution ($D_{KL}(\pi_{\theta_{old}}|\pi_\theta)$ exceeds a specified threshold (β_{KL}):

$$D_{KL}(\pi_{\theta_{old}}|\pi_\theta) \leq \beta_{KL} \quad (16)$$

which ensures that learning remains gradual and stable.

Second, normalisation of rewards is done to ensure the scale doesn't hinder the agent's learning. This is particularly useful because the rewards in the environment are not constant under different iterations. Therefore, the normalization of rewards from the entire batch of data is carried out as follows:

$$r_i' = \frac{r_i - r_\mu}{r_\sigma + \epsilon_r} \quad (17)$$

Here, r_i' is the normalised reward, r_i is the reward at the end of i^{th} step of a batch of multiple episodes, and r_μ and r_σ are the mean and standard deviation of rewards for that batch of data. The ϵ_r avoids having a zero or a very small number in the denominator. The hyperparameter settings for PPO are shown in Table I.

TABLE I
PPO HYPERPARAMETER SETTINGS

Hyperparameter	Value	Hyperparameter	Value
α	3e-4	GAE's λ	0.1
Hidden Layers	4	N	25
Hidden Size	128	β_{KL}	0.015
ϵ	0.1	buffer_size	3840
γ	0.99	c_v	0.5
		c_s	0.01

TABLE II
ENVIRONMENT RELATED PARAMETERS

Parameter	Value	Parameter	Value
η_c	0.95	ϕ_u	2.5 ¢/kWh
η_d	0.95	P_{max}	20 kW
B^{cap}	40 kWh	β	80
ϕ_0	10 ¢/kWh	Δt	0.25 h

2) Environment details: To simulate the retail price setting process, the ERCOT's Real Time Market (RTM) Price Data[1] from 2018 for training and 2019 for testing is used since retail price data coming from a transactive design is not readily available. Also, normalisation of data using *min-max scaling* is done before giving it as input to NN. The EV is assumed to come to a charging station at an office location at 10 A.M. and leave at 6 P.M., making the total plug-in duration 8 hours. Since the RTM price data is available for every 15-minute interval, 32 intervals (8 × 4) are there for submitting bids and offers. To introduce stochasticity in the environment and generalise the performance, the EV's initial SoC (SoC_0) and desired SoC (SoC_{des}) are assumed to follow a Gaussian distribution with mean and variance as given below:

$$\begin{aligned} SoC_0 &\sim \mathcal{N}(0.45, 0.05^2) \\ SoC_{des} &\sim \mathcal{N}(0.8, 0.05^2) \end{aligned} \tag{18}$$

Environment-related parameters are given in Table II.

IV. RESULTS AND DISCUSSION

The simulations were conducted on a PC with an Intel i9-12900K CPU, NVIDIA 3060Ti GPU, and 32 GB of RAM. The 32 intervals of the plug-in time, constituting one daily simulation, are considered as one episode. During training, the PPO agent took 124 minutes to reach convergence.

1) Episodic reward progression: The PPO agent took approximately 50k episodes to converge to an average reward per episode of -0.8 to -0.5, as shown in Fig. 3 during training with the data of the year 2018. The simulation outcomes during testing with the data of the year 2019 are presented in Fig. 4. As evident from the figure, rewards per episode stay mostly around -1.0 to -0.4, with a minimum of -4. The peaks in Fig. 4 represent times when the prices in the testing dataset were high for the entire plug-in duration, which caused the agent to have a lower reward. The testing results in Fig. 4 demonstrate the generalizability of the proposed DRL EV agent, as it performs well on unseen data.

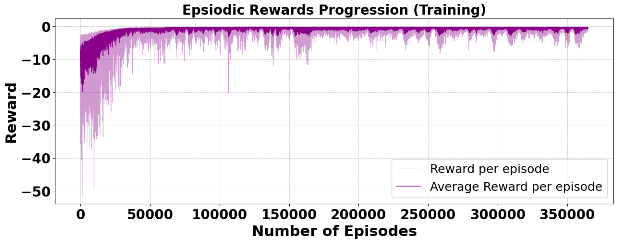

Fig. 3. Average episodic rewards progression (training)

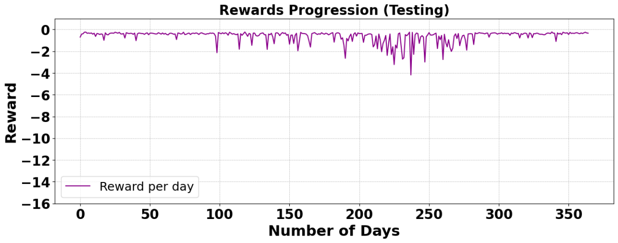

Fig. 4. Daily rewards progression (testing)

2) Further analysis of DRL-based EV agent's performance using testing data: Seven days of results were extracted from Fig. 4 from different months to demonstrate the DRL-based EV agent's performance to varied retail price fluctuations, and shown in Fig. 5. The results of day 4 and day 6 are magnified for improved readability. As can be seen, the agent is placing bids and offers for charging and discharging the EV under different SoC_0 and the SoC_{des} values. Guided by its policy, it tries to reach the SoC_{des} by adapting its bidding/offering behaviour. Generally, it places offers to discharge in early intervals when it knows it has time to charge and reach SoC_{des} or when it is at $SoC_t \geq SoC_{des}$ (see day 7). The figure also illustrates that when the disparity between SoC_{des} and the SoC_t is substantial, the agent limits offers for discharging, especially when it anticipates SoC_{des} cannot be reached by t_{end}. However, in cases where the agent's SoC_t is close to or above SoC_{des}, it places offers for discharging (see day 4). Moreover, when the prices are almost constant and low, the agent charges to an $SoC_{t_{end}} > SoC_{des}$ (see day 6), showcasing its adaptive behaviour based on varying charging requirements and market prices.

The values for parameters β and ϕ_u can be adjusted to capture the user's range anxiety and discomfort for premature battery replacement. Additionally, users can set β to limit the maximum willingness to pay for EV charging and ϕ_u to limit the minimum acceptance price for EV discharging. From the testing data, it is observed that the EV agent's maximum bid price for charging is 10.41 ¢/kWh, while the maximum offer price to discharge is 5.25 ¢/kWh. It is important to note that the training and testing data utilize wholesale RTM prices, which typically exhibit greater volatility. Overall, it can be concluded that the agent effectively adapts to the data.

[1] https://ercot.com/content/rtm/rtm.htm

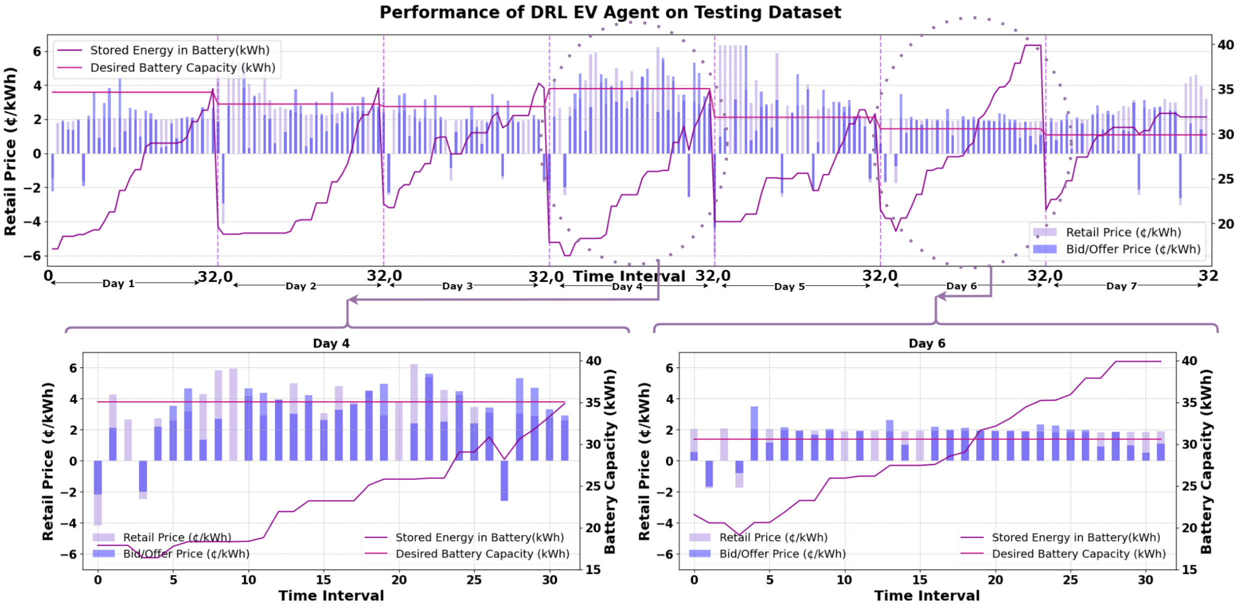

Fig. 5. Performance of DRL-based EV agent on a week of testing data

V. Conclusion and Future Work

This research presents a DRL-based EV agent that empowers customers to participate in a bid-based TES design for distribution systems. The problem is formulated as a POMDP to enable the agent to make better decisions under uncertainty. To extract sequential and temporal information from the partially observable environment, LSTM networks are employed and trained using the PPO technique. The results demonstrate the convergence of the policy for placing bids and offers that align with user goals and constraints while adapting to dynamic price fluctuations and different charging requirements. Notably, the agent learns not to place offers for discharging when prices are constant or low; rather, charge to an $SoC_t \geq SoC_{des}$. Also, when $SoC_t \geq SoC_{des}$ or prices are higher, it gets benefited by offering to discharge. The future work will focus on integrating DRL-based EV agents into a bid-based TES design operated by a DSO. A comprehensive impact assessment will be conducted to evaluate the performance of the DRL-based EV agents and the DSO's utilization of the flexibility of these bids/offers to improve overall system operations.

References

[1] R. B. Melton, "Gridwise transactive energy framework (v.1.1)," 2019. [Online]. Available: https://www.osti.gov/biblio/1968036

[2] S. Battula, L. Tesfatsion, and Z. Wang, "A Customer-Centric Approach to Bid-Based Transactive Energy System Design," *IEEE Transactions on Smart Grid*, vol. 11, no. 6, pp. 4996–5008, 2020.

[3] H. Saber, H. Ranjbar, S. Fattaheian-Dehkordi, M. Moeini-Aghtaie, M. Ehsan, and M. Shahidehpour, "Transactive Energy Management of V2G-Capable Electric Vehicles in Residential Buildings: An MILP Approach," *IEEE Transactions on Sustainable Energy*, vol. 13, no. 3, pp. 1734–1743, 2022.

[4] A. Singhal, B. Bhattarai, F. B. dos Reis, H. Reeve, and R. Pratt, "Transactive Electric Vehicle Agent: Design and Performance Evaluation," in *2021 IEEE Power & Energy Society General Meeting (PESGM)*, 2021, pp. 1–5.

[5] M. M. Hoque, M. Khorasany, R. Razzaghi, H. Wang, and M. Jalili, "Transactive Coordination of Electric Vehicles With Voltage Control in Distribution Networks," *IEEE Transactions on Sustainable Energy*, vol. 13, no. 1, pp. 391–402, 2022.

[6] H. Saber, M. Ehsan, M. Moeini-Aghtaie, H. Ranjbar, and M. Lehtonen, "A User-Friendly Transactive Coordination Model for Residential Prosumers Considering Voltage Unbalance in Distribution Networks," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 5748–5759, 2022.

[7] M. M. Hoque, M. Khorasany, R. Razzaghi, M. Jalili, and H. Wang, "Network-Aware Coordination of Aggregated Electric Vehicles Considering Charge–Discharge Flexibility," *IEEE Transactions on Smart Grid*, vol. 14, no. 3, pp. 2125–2139, 2023.

[8] H. Saber, M. Ehsan, M. Moeini-Aghtaie, M. Fotuhi-Firuzabad, and M. Lehtonen, "Network-Constrained Transactive Coordination for Plug-In Electric Vehicles Participation in Real-Time Retail Electricity Markets," *IEEE Transactions on Sustainable Energy*, vol. 12, no. 2, pp. 1439–1448, 2021.

[9] Y. Tao, J. Qiu, and S. Lai, "Deep reinforcement learning based bidding strategy for evas in local energy market considering information asymmetry," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 6, pp. 3831–3842, 2022.

[10] H. Li, Z. Wan, and H. He, "Constrained EV Charging Scheduling Based on Safe Deep Reinforcement Learning," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2427–2439, 2020.

[11] Z. Wan, H. Li, H. He, and D. Prokhorov, "Model-free real-time ev charging scheduling based on deep reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5246–5257, 2019.

[12] Z. Ye, Y. Gao, and N. Yu, "Learning to Operate an Electric Vehicle Charging Station Considering Vehicle-Grid Integration," *IEEE Transactions on Smart Grid*, vol. 13, no. 4, pp. 3038–3048, 2022.

[13] O. E. Balghiti, A. N. Elmachtoub, P. Grigas, and A. Tewari, "Generalization Bounds in the Predict-then-Optimize Framework," *CoRR*, vol. abs/1905.11488, 2019. [Online]. Available: http://arxiv.org/abs/1905.11488

[14] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," 2017.

[15] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-Dimensional Continuous Control Using Generalized Advantage Estimation," 2018.