IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 7 September 2022, accepted 18 October 2022, date of publication 26 October 2022, date of current version 3 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3217497

RESEARCH ARTICLE

Intelligent Bidding Strategies for Prosumers in Local Energy Markets Based on Reinforcement Learning

GODWIN C. OKWUIBE[1,2], **(Member, IEEE), JEEL BHALODIA**[2], **AMIN SHOKRI GAZAFROUDI**[2], **THOMAS BRENNER**[2], **PETER TZSCHEUTSCHLER**[1], **AND THOMAS HAMACHER**[1]

[1]School of Engineering and Design, Technical University of Munich, 80333 Munich, Germany
[2]OLI Systems GmbH, 67376 Harthausen, Germany

Corresponding author: Godwin C. Okwuibe (godwin.okwuibe@tum.de)

The work of Godwin C. Okwuibe, Amin Shokri Gazafroudi, and Thomas Brenner was supported by the Federal Ministry for Economic Affairs and Energy (BMWi), Germany, as a part of the Blockchainbasiertes dezentrales Energiemarktdesign und Managementstruktur (BEST) Project under Grant 03EI4017D.

ABSTRACT Local energy markets (LEMs) are proposed in recent years as a way to enable local prosumers and community to trade their electricity and have control over their electrical related resources by ensuring that electricity is traded closer to where it is produced. However, literature is still scarce with the most optimal and effective trading strategies for LEM design. In this work, we propose two reinforcement learning based intelligent bidding strategies for prosumers and consumers trading within an LEM. Our proposed models were evaluated of their performance by testing them in a German real case scenario. The simulation results show that intelligent bidding strategies create additional self sufficiency and market savings to the local community compared to the baseline strategy where the agents make their trading decision randomly without an intelligent agent. Moreover, modelling the intelligent agents to perform towards a common goal creates more share of individual savings for the prosumers and consumers compared to the classical intelligent bidding strategies employed in this work.

INDEX TERMS Bidding strategy, energy community, local energy markets, Markov decision process, peer-to-peer, reinforcement learning.

I. INTRODUCTION

A. MOTIVATION AND BACKGROUND

Local energy markets (LEMs) were introduced in recent years as a means to curb the challenges resulting from increasing share of variable distributed energy resources at the distribution grid level and thus creating an avenue to get small-scale producers, prosumers and consumers involved in the electricity market [1]. By transacting electricity closer to where it is produced, producers, prosumers and consumers create additional benefits for each other compared to transacting electricity to a far distance prosumers/consumers or with the upstream grid [2]. However, residential and most commercial prosumers/consumers are lay users and have little or no knowledge of the electricity markets. Therefore, they may not be able to decide the appropriate bidding/offering price for their energy demand/supply considering the dynamics and complexity involved [3]. On the other hand, the local electricity market is a time series market platform. Hence, consumers and prosumers are required to consistently post their bid/offer containing their desired energy quantity and price every time slot [4]. This is inefficient and time consuming and thus, the need for an intelligent bidding/offering agent responsible for making the complex and dynamic decision involved in LEM trading. The agent is also responsible for selecting appropriate price for prosumers/consumers to make benefits from their electricity assets and automatically posting the bids/offers on behalf of consumer/prosumers who own the agent [3], [4]. Consequently, researchers have proposed different bidding strategies for LEM design [5], [6].

The associate editor coordinating the review of this manuscript and approving it for publication was Oussama Habachi.

B. LITERATURE REVIEW

According to Ref. [7], LEM bidding strategies can be classified either as zero-intelligence or intelligent agent bidding strategies. Zero-intelligence agents randomly select their bid/offer price within the limit range of the maximum selling and minimum buying price, which is usually the upstream grid and feed-in tariff price, respectively. Intelligent agents bidding strategy usually derive their trading price based on some optimization model, algorithms, game theoretic approach and/or learning experience. The authors of [8], proposed a linear bidding/offering strategy for LEM that allows agents to linearly decrease/increase their bid/offer price, respectively, in an LEM time slot. This strategy allows prosumer agents to send multiple bids/offers in a single time slot thereby increasing their chance of being matched in the LEM without increasing the computational power requirements of the agents. Ref. [9] proposed an optimization based prediction-integration strategy called surrogate market prediction model based on Extreme Learning Machine. The model was used to learn the relationship between prosumer bidding actions and market responses from historical transaction data and the outcome was used for bidding/offering in a peer-to-peer(P2P) LEM.

The effective metrics and criteria for evaluating the performance of an LEM bidding strategy was introduced by Ref. [5]. Furthermore, the authors modelled the behaviors of both risk-neutral and risk-averse agents selling energy to the LEM taking into account the expected profit and risk criteria to obtain an optimal multi-step energy quantity-price bidding strategies of risk-neutral and risk-averse agents. PV is considered a major source of energy during the day for most intra community and inter community energy trading. Thus, considering the uncertainties in solar irradiance and temperature can result in an optimal P2P LEM bidding strategy [6], [10]. Ref. [6] proposed a dual bidding strategy for multi-hierarchical P2P energy trading in an LEM considering uncertainties in solar irradiance and temperature. By considering the uncertainty of renewable energy resources such as solar, wind and consumer demands in an LEM, the authors of [10] proposed a bi-level optimization model for prosumers to appropriately take advantage of their distributed energy resources in an LEM. The optimal bidding curve which described the cost-minimizing buying/selling strategy of prosumers was used to guarantee the optimality of bidding decisions and to reduce the computation and communication overhead of bidding agents by [11]. Ref. [12] proposed a two-stage bidding strategy for P2P LEM design. The first stage considered the supply-demand relationship for two-step price predictor with the aim to promote the usage of local renewable energy within the LEM. In the second stage, a trading preference based simultaneous game-theoretic approach was introduced and used to optimize the market equilibrium and social welfare of the P2P LEM. In [13], an optimal bidding/offering strategy was proposed for prosumers in an LEM to improve their savings and further increase the overall social welfare of the local community.

Before the introduction of LEM in the last two-decades and in recent years, reinforcement learning is used by electricity producers for making decision on their offering price in a competitive electricity market. In [14], a modified continuous action reinforcement learning automata algorithm was proposed to help power suppliers bid with the limited information in an electricity market. Ref. [15] proposed an experience weighted attraction reinforcement learning algorithm for bidding in an electricity markets. The authors of [16] used a deep deterministic policy gradient reinforcement learning algorithm to develop a bidding agent for a uniform pricing electricity market. Ref. [17] developed a fuzzy Q-learning method and used it to model the electricity producer strategic bidding behavior in a competitive and computational electricity market. In [18], the deep deterministic policy gradient method was combined with a prioritized experience replay strategy and used to model the strategic bidding decisions in a deregulated electricity markets. In the same way, Ref. [19] combined reinforcement learning with belief learning that converts experience-weighted attraction in a learning model for describing and improving individual learning behavior for effective bidding in a double auction electricity markets.

Similar to the main electricity markets and as a result of its numerous advantages, reinforcement learning is currently gaining the interest of researchers on how it can be utilized for decision making in an LEM. By solving the deep reinforcement learning technology with experience replay mechanism, Ref. [20] modified the deep Q-learning for local energy trading algorithm from deep Q-network to facilitate the decision-making process of local energy prosumers with an intelligent system and further promote prosumers' willingness to participate in the LEM trading. Ref. [21] proposed an intelligent bidding strategy based on an adaptive reinforcement learning model for prosumers within a local grid. In Ref. [22], a deep learning based on data-driven approach was developed and used to model the transaction behaviour of prosumers and consumers based on public information in a two-stage P2P local electricity market. A Q-learning based intelligent bidding strategy was proposed by [23] for prosumers in a competitive two-sided pay-as-bid LEM. To further integrate electric vehicle trading in an LEM, [24] proposed a data analytics and deep reinforcement learning based bidding strategy for electric vehicle aggregators in an LEM.

C. CONTRIBUTION & ORGANIZATION

Currently, the literature contains only few studies proposing intelligent bidding strategies for LEM design. Moreover, there is still a gap in literature proposing and comparing different reinforcement learning models for LEM bidding strategies and further suggesting the most optimal strategy for the different local energy participants types. In this paper, by answering the research question, which trading strategies are most suitable for effective performance of local energy markets? We propose two novel reinforcement learning based intelligent bidding strategies for prosumers and consumers in an LEM. The proposed models use information/data initially acquired from the LEM and the participants to formulate the most optimal bidding/offering prices for consumers and prosumers within an LEM for them to make optimal benefits from their consumption's/productions. The models were implemented on an interface and open source code-base of the Grid Singularity bidding application programming interface [25]. Furthermore, we evaluate the models using performance indicators such as self-sufficiency, share of market savings and traded energy quantity of the prosumers/consumers in the LEM. The main contributions of the paper are summarized below:

- Proposing two novel reinforcement based intelligent bidding strategies for prosumers and consumers in an LEM.
- Implementation of the proposed intelligent bidding strategy models in a real case German community.
- Assessing the performance of the proposed bidding strategies using LEM and reinforcement learning performance indicators.

The remaining sections of this work are structured as follows. Section II introduces reinforcement learning taxonomy and the proposed reinforcement learning algorithms. The proposed reinforcement learning bidding strategies are described in Section III. The simulation case study, community set-up and simulation data are presented in Section IV. Section V presents the simulation results, while Section VI discusses the findings, insights and improvements of the models. Finally, the paper is concluded in Section VII.

II. REINFORCEMENT LEARNING

A. REINFORCEMENT LEARNING TAXONOMY

Reinforcement learning (RL) is a branch of machine learning that uses trial-and-error strategy to learn from previous actions to make future decisions [26]. The reward strategy which gives positive reward to successful events/actions and negative rewards to unsuccessful events/actions is used by the agents to improve its learning and decision making [27]. The term "value function" refers to prediction of future benefits based on the present condition or state [28]. By comparing the outcome of each decision made by the agents, the agents get experience on how to perform better on future actions of the model based on its prior experience with the environment. In RL, this experience-based decision making process which helps the agents to maximize their reward for each action taken against the environment is referred to as a policy [26]. The Markov Decision Process (MDP), is a probabilistic model based on sequential decision making and provides the mathematical foundation for RL process [29]. The MDP property claims that "the probability of the future is independent of the past given the present" [29]. However, the optimal action of an agent is usually obtained when the agent evaluate not just the immediate reward but also the long-term quality of the action(s). Because of the high accuracy when given more data, the action-value function is preferable in the long term RL model [30]. This is why an action-value function is more suitable for intelligent agent-based bidding strategies for LEM.

RL algorithm can be classified based on their access to the model as model based and model-free algorithm [31]. Model-based RL algorithm is a type of RL in which the agent is privileged to know all possible state transition probabilities and rewards [32]. This type of RL is heavily influenced by control theory, and the objective is to obtain the optimum behaviours using a control function. The major draw back of this RL method is that since they have access to all potential state-actions, storing all the probabilities becomes impractical as the number of states and actions increases exponentially [32]. Model-free RL is a type of RL model that develops its own optimum strategy based on its own experience with its surroundings, state-action pairings, and their associated rewards [33]. Model-free algorithms are classified into policy iteration and value iteration [31]. For policy iteration model-free algorithm, the agent directly learns the policy function that translates state to action using policy optimization approaches [32]. Thus the policy is decided without the use of a value function. The policies here can be either deterministic or stochastic [32]. Value iteration model-free RL algorithm gains knowledge of the action-value function, calculates the expected discounted rewards received for taking a particular action and determines how beneficial it is to behave in a specific state [31]. Thus, a scalar value called Q-value is assigned to an action based on its state and the ideal results are obtained when the action with the highest Q-value is chosen. Value iteration algorithm can be further classified into off-policy and on-policy algorithms. Off-policy RL algorithms use greedy policy and learns the best policy and acts based on a different policy [34]. The updated policy differs from the behaviour policy. The Q-learning algorithm is an example of an off-policy algorithm. On the other hand, for the on-policy strategy, the actor captures the best policy and applies it to its actions [34]. The major difference between off-policy and on-policy is that for on-policy, the policy for updating and acting is the same, while it is different for off-policy. On-policy tries to evaluate the same policy that is used to make decisions [34]. State Action Reward State Action (SARSA) algorithm is an example of on-policy algorithm.

B. REINFORCEMENT LEARNING ALGORITHMS

1) OFF-POLICY ALGORITHM - Q-LEARNING

In Q-learning, the 'Q' stands for quality which refers to the usefulness of a certain action in obtaining a future reward which can be determined by the Q-value. Q-learning seeks to determine the best policy while pursuing a separate exploration strategy [35]. This class of algorithms updates the state or state-action values by calculating the difference between current and past estimations [33]. Eq. (1) is the general Q-learning function which states that the Q-value depends on the state-action combination [32].

$$\mathcal{Q} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{R} \tag{1}$$

where $\mathcal{S}$ is the state of the agent, $\mathcal{A}$ is the different actions that are available for the agent to take and $\mathcal{R}$ is the rewards the agents can receive for taking different actions. Eq. (2) represent the different states of an agent.

$$\mathcal{S}_t = \{[s_{1,t}^{x1}, s_{1,t}^{x2}, \ldots, s_{1,t}^{xm}], \ldots, [s_{n,t}^{x1}, s_{n,t}^{x2}, \ldots, s_{n,t}^{xm}]\} \tag{2}$$

where s_1 to s_n are the different states and x_1 to x_m are the state variables. The different state variables combine to form one sate s. n and m are the number of states and state variables, respectively. Therefore, at a time step t, the state of an agent is represented in Eq. (3),

$$s_t = s_t^{x1}, s_t^{x2}, \ldots, s_t^{xm}. \tag{3}$$

The different actions an agent can take at any time step is represented by (4).

$$\mathcal{A} = a_{1,t}, \ldots, a_{u,t} \tag{4}$$

where u is the number of actions. The available rewards for an agent at a time step is represented in Eq. (5),

$$\mathcal{R} = (-\infty, \infty). \tag{5}$$

Initializing Q-table: A Q-table is a matrix with the structure of [states, actions] used for Q-learning process as represented in Eq. (6).

$$\mathcal{Q}_{s\times a,t} = \begin{bmatrix} q_{1,1,t} & \cdots & q_{1,u,t} \\ \vdots & \ddots & \vdots \\ q_{n,1,t} & \cdots & q_{n,u,t} \end{bmatrix}, \forall t. \tag{6}$$

Q-table is first initialized with either 0 or 1 and updated every time step. It is the reference table for agents, which it uses to determine the optimal course of action taken.

Taking action: In order to interact with the environment, the agent needs to select an action. The agent chooses an action based on the maximum value in the Q-table or on a random basis [32]. The epsilon (ϵ) greedy approach is used by an agent to achieve a balance between exploration (chosen random action) or exploitation (chosen action with maximum Q-value) and to interact with its environment in either of the two ways throughout the experiment [36]. Eq. (7) represents that the total experiment time step is the sum of exploration and exploitation time represented as t^ϵ and $t^{1-\epsilon}$, respectively,

$$\mathcal{T} = \sum t^\epsilon + \sum t^{1-\epsilon}. \tag{7}$$

Eq. (8) represents how the agents select its action during time of exploration and exploitation. From Eq. (8), a_{u,t^ϵ} and $a_{u,t^{1-\epsilon}}$ represents actions selected by the agent during exploration and exploitation, respectively,

$$a_{u,t} = \left\{ \begin{array}{ll} a_{u,t^\epsilon}: & \epsilon = 1 \\ a_{u,t^{1-\epsilon}}: & \epsilon = 0 \end{array} \right\}, \forall t. \tag{8}$$

Eq. (9) represents that during exploration, the agent takes a random action by randomly selecting any action from the available actions at the time step. On the other hand, Eq. (10) represents that during exploitation, the agent takes action by selecting the action with the maximum Q-value,

$$a_{u,t^\epsilon} = rand.\{a_{1,t}, \ldots, a_{u,t}\}, \tag{9}$$

$$a_{u,t^{1-\epsilon}} = \arg\max \sum \mathcal{Q}_{s\times a,t}. \tag{10}$$

Updating Q-table: Q-values of Eq. (6) are updated every time step based on Bellman's equation as represented in Eq. (11) [32], [37].

$$Q_t(s_t, a_t) = \sum \left[r_{(t+1)} + \gamma \max_{a'} Q_t\left(s_{(t+1)}, a_{(t+1)}\right)\right] \tag{11}$$

where, s_t and a_t are the current state and action, $s_{(t+1)}$ and $a_{(t+1)}$ are the next state and action, respectively. r_{t+1} is the expected reward. γ is the discount factor and represents the amount of value the agent place on the future benefits. The discount rate ranges from 0 to 1 ($0 \leq \gamma \leq 1$). With a higher discount rate, the agent places a higher premium on future returns. Bellman's equation is used to calculate the value of a state and to estimate how beneficial it is to be in that state. The ideal state is the state that produces the optimal Q-value. Q-learning algorithms are based on the Bellman's equation used as the basic value of iteration update, based on the weighted average of the old and new Q-values and is defined according to Eq. (12) [37].

$$Q_{(t+1)}\left(s_t, a_t\right) \leftarrow Q_t\left(s_t, a_t\right) + \alpha \left[r_t + \gamma \max_a Q_t\left(s_{(t+1)}, a_t\right) - Q_t\left(s_t, a_t\right)\right] \tag{12}$$

where α is the learning rate and indicates the learning pace of the agent. This parameter gives information on how the agents' estimations should be updated considering the mistakes. The learning rate ranges between 0 and 1. A high learning rate adapts aggressively, which may result in variable — rather than converging — training outcomes. A low learning rate adapts slowly, which means that it will take longer time to converge. The terms in Eq. (12) are defined as follows:

- (1- α) $Q_t(s_t, a_t)$ is the current Q-value weighted by the learning rate.
- $\alpha\, r_t$ is the reward obtained in state s_t by taking action a_t, weighted by learning rate.

$$\alpha\gamma \max_a Q_t\left(s_{(t+1)}, a\right) \tag{13}$$

- Eq. (13) is the maximum reward to be obtained from the next state $s_{(t+1)}$.

$$r_t + \gamma \max_a Q_t\left(s_{(t+1)}, a\right) \tag{14}$$

- The term given in Eq. (14) which is derived from Eq. (12) is called the temporal difference target through which the estimated Q-value is adjusted.

The temporal difference is an estimate of the optimum Q-value which the agent strives to get and this varies as the agent is trained and the Q-value matrix updated. The Q-value of the agent's current state and action is updated by subtracting the previous Q-value and then adding the learned value. The learned value is a function of the reward for taking the current action in the current state and the discounted maximum reward from the subsequent state in which the present action is performed.

2) ON POLICY ALGORITHM : SARSA

SARSA is an on-policy reinforcement learning algorithm in which an action, A, is taken in the current state, S, and the agent receives a reward, R. The agent then moves to the next state, S', and performs action, A' in S'. As a result, the name SARSA is derived from the tuple (S, A, R, S', A'). SARSA is referred to as an on-policy algorithm because it adjusts the policy in response to actions and is part of temporal difference learning [36]. The algorithm's learning process is similar to Q-learning described in Section II-B1. The first step is to initialize the Q-table. Secondly, an action similar to that of Q-learning with its ϵ-greedy strategy is taken. Lastly, the Q-table is updated and this is where there is distinction between SARSA and Q-learning. Eq. (15) is used to update the Q-value for SARSA.

$$Q_{(t+1)}(s_t, a_t) \leftarrow Q_t(s_t, a_t) + \alpha \left[r_{(t+1)} + \gamma Q_t\left(s_{(t+1)}, a_{(t+1)}\right) - Q_t(s_t, a_t)\right] \quad (15)$$

The update equation of SARSA, Eq. (15) shows that the goal value in this case is dependent on the action the agent will take in the following state, s_{t+1}. Since this update is dependent on the next action a_{t+1}, which is determined by the current policy, this algorithm is termed on-policy [37]. While training the agent and the corresponding Q-value (and policy) is updated, the new policy may generate a different action for next time step $a_{(t+1)}$ for the same state $s_{(t+1)}$. It is not possible to improve the estimations by drawing on prior experiences. Hence, the algorithm utilizes each experience just once to update the Q-values and then discard it [34].

III. PROPOSED REINFORCEMENT LEARNING BIDDING STRATEGIES

In this Section, the proposed reinforcement bidding strategies is presented. Fig. 1 and 2 represents the process diagram and the flowchart, respectively, of the proposed bidding strategies. First, the state, actions and reward function are defined. Then, the Q-learning algorithm bidding strategy is introduced followed by SARSA bidding strategy.

A. STATE, ACTION AND REWARD DEFINITION

1) STATE

The state of the agent at any time t, is define by Eq. (16).

$$\mathcal{S}_t = \{[s^{\sqcup}_{1,t}, s^{\overline{p}}_{1,t}, s^{\overline{v}}_{1,t}, s^{\rho}_{1,t}], \ldots, [s^{\sqcup}_{n,t}, s^{\overline{p}}_{n,t}, s^{\overline{v}}_{n,t}, s^{\rho}_{n,t}]\} \quad (16)$$

where the state variable $\sqcup$ is the hour of the day where the trading occurs and this is split into 24 variables .i.e $\sqcup = \{0, 1, \ldots, 23\}$. $\overline{p}$ is the average market trade rate in cent/kWh for the previous market time step. The average trade rate has a range from feed-in tariff price ($p^{\perp}$) to the electricity buy price (p^b) from the upstream grid. In order to reduce the size of the Q-table for its optimal performance, the range of the state variable $\overline{p}$ is divided into six equal buckets as presented in Eq. (17)

$$\overline{p} = \{p^{\perp}, \frac{(4 \times p^{\perp} + p^b)}{5}, \ldots, p^b\}. \quad (17)$$

$\overline{v}$ is the average trade volume which is the average volume of the electricity traded internally within the local community without the help of the upstream grid in the last time step. The average trade volume is also divided into six buckets in order to have discrete data and to reduce the size of the Q-table as represented in Eq. (18)

$$\overline{v}_t = \{\overline{v}_{1,t}, \overline{v}_{2,t}, \ldots, \overline{v}_{6,t}\}. \quad (18)$$

ρ is the solar irradiation which gives information on the average solar radiation received per unit area of the simulation area. This state variable is divided into nine buckets to better refelect the different solar radiation intervals as given in Eq. (19)

$$\rho_t = \{\rho_{1,t}, \rho_{2,t}, \ldots, \rho_{9,t}\}. \quad (19)$$

Hence, at any time step t, the state of the agent is described as a tuple containing four state variables $\sqcup, \overline{p}, \overline{v}$ and ρ.

2) ACTION

This is a discrete set of potential bids/offers prices available to a consumer or prosumer. This varies from the lowest ($p^{\perp}$) to the highest price (p^b) allowed with the LEM. The range of bid/offer price is discretize into sixteen potential actions as represented in Eq. (20)

$$\mathcal{A} = \{p^{\perp}, \frac{(14 \times p^{\perp} + p^b)}{15}, \ldots, p^b\} \quad (20)$$

In RL, the agent learns what to do by itself and translate situations to actions to maximize a numerical reward signal [26]. The agent is not instructed on which actions to take; instead, through trial and error, it decides which action gives the maximum reward. For our model, ϵ greedy policy is used to handle the exploration/exploitation dilemma by applying simple strategy for balancing exploration and exploitation by randomly selecting between the two. For this, a random number λ is selected and the value compared with the given value of ϵ as represented in Eq. (21),

$$a_{u,t} = \left\{ \begin{array}{ll} a_{u,t^{\epsilon}}: & \lambda < \epsilon \\ a_{u,t^{1-\epsilon}}: & \lambda > \epsilon \end{array} \right\}, \forall t. \quad (21)$$

Hence, from Eq. (21), a random action (exploration) is selected if $\lambda < \epsilon$, while action leading to the maximum reward (exploitation) is selected if $\lambda > \epsilon$. The developed RL algorithm is a single agent RL method which contains a combined and evolved policy for both consumers and producers. Both consumers and prosumers take distinct actions as they have a different pricing strategy. However, their states and reward functions are identical. As a result, each actor uses a different Q-table inside a single RL agent, where one actor is not aware of the presence of the other actor.

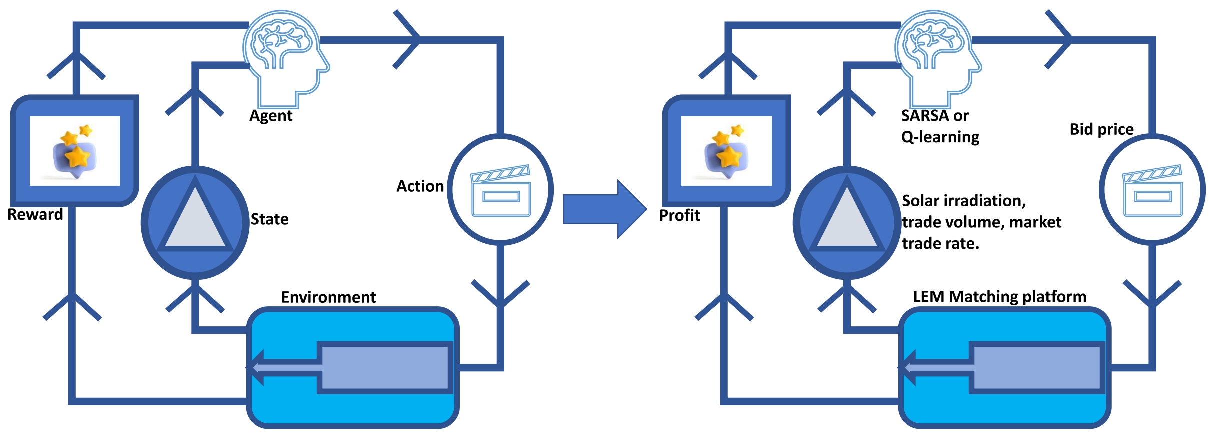

FIGURE 1. Proposed bidding strategies based on Q-learning and SARSA.

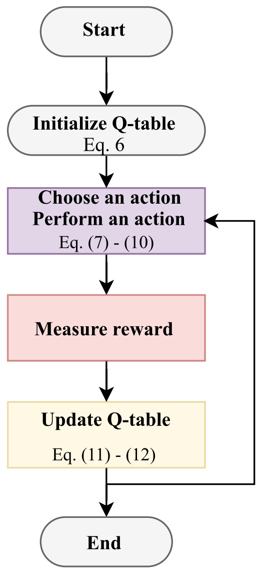

FIGURE 2. Learning process flowchart for Q-learning and SARSA.

3) REWARD FUNCTION

The reward function is used to compare the value of actions taken by the agent at different states. To encourage prosumer behaviour that results in greater consistency with their trading goals the reward function of our model is a mix of the monetary value of the bid/offer and how early trade occurs in a market time step. For this, the market model based on the work of Ref. [8] which allow prosumers and consumers to submit multiple bids and offers at a time step. Hence, each market time step is further divided into market ticks t_t^* as represented in Eq. (22).

$$t_t^* = \{t_{1,t}^*, t_{2,t}^*, \dots, t_{k,t}^*\}, \tag{22}$$

where k is the number of ticks in a time step t. The trade rate of prosumer agent i, at a time step is compared with the markets median trade rate at the same time step to obtain the monetary value of reward function. The median is used to prevent outliers from having impact if the mean was used as the measure. The monetary reward function of consumers is represented in Eq. (23).

$$r_t^{b,\$} = \left\{ \begin{array}{cc} 0.7 \times (\widehat{p}_t - p_{i,t}^b)/(p^b - \widehat{p}_t): & p_{i,t}^b > \widehat{p}_t \\ 0: & p_{i,t}^b = \widehat{p}_t \\ 0.7 \times (\widehat{p}_t - p_{i,t}^b)/(\widehat{p}_t - p^{\perp}): & p_{i,t}^b < \widehat{p}_t \end{array} \right\}, \forall t, \tag{23}$$

where $\widehat{p}_t$ is the median trade rate at time step t and $p_{i,t}^b$ is the trade rate of consumer agent i, at the same time step. Eq. (23) shows that if a consumer's trade rate is lower than the community's median rate $\widehat{p}_t$, the consumer's agent receives a positive reward since he/she (the consumer) paid less than the other members of the community and outperformed them. When a consumer's trade rate exceeds the market's median rate, the consumer's agent earns a negative reward. Eq. (24) represents the prosumer (producing) reward function.

$$r_t^{s,\$} = \left\{ \begin{array}{cc} 0.7 \times (p_{j,t}^s - \widehat{p}_t)/(p^b - \widehat{p}_t): & p_{j,t}^s > \widehat{p}_t \\ 0: & p_{j,t}^s = \widehat{p}_t \\ 0.7 \times (p_{j,t}^s - \widehat{p}_t)/(\widehat{p}_t - p^{\perp}): & p_{j,t}^s < \widehat{p}_t \end{array} \right\}, \forall t, \tag{24}$$

where $p_{i,t}^s$ is the trade rate of prosumer (producing) agent j, at time step t. Eq. (24) shows that if a prosumer's trade rate while producing is lower than the community's median rate $\widehat{p}_t$, the prosumer agent receives a negative reward since he/she (the prosumer) get less money compared to the other members of the community. When a prosumer's trade rate while producing exceeds the market's median rate, the agent earns a positive reward because the prosumer gets more money compared to other community members. The tick at which the trade takes place is compared to the total number of ticks per time slot to get the accuracy reward function. This is used to know how accurate and efficient the agent is in making their bids/offers at a time step. The degree of accuracy reward function is represented in Eq. (25).

$$r_t^{\mu} = 0.3 \times (t_{k,t}^* - t_{\mathcal{K},t}^*), \tag{25}$$

where $\mathcal{K}$ is the time tick at which trade took place. This means that the greater the disparity from the total number of ticks per time step, the greater the reward. This parameter does not result in a negative reward; rather, it results in a positive reward that is either increased or decreased. Eqs. (24) and (25) shows that the model places 70% value on monetary reward while the remaining 30% is on accuracy. The general reward function is the sum of the monetary and accuracy reward function. Eqs. (26) and (27) represent the total reward function for a consumer and prosumer (producing), respectively,

$$r_t^b = r_t^{b,\$} + r_t^{\mu}, \tag{26}$$

$$r_t^s = r_t^{s,\$} + r_t^{\mu}. \tag{27}$$

B. Q-LEARNING STRATEGY

1) STAGE I: INITIALIZE ALGORITHM

a: STEP I: INITIALIZE Q-TABLE

For the Q-learning algorithm, the defined state and actions acts as inputs to the Q-table. The defined state contains four state variables made of 24 time variables, 6 average trade variables, 6 average trade volume variable, and 9 solar irradiation variable. Consequently, we have a total of 7776 (24 × 6 × 6 × 9) defined states for the Q-table. Eq. (28) represents the initialization of Q-table with ones for the 7776 defined states and 16 available action according to Eq. (20) at time step t. The Q-table is defined separately for consumers and producing prosumers because of their different trading interests,

$$\mathcal{Q}_{s\times a,t} = \begin{bmatrix} 1_{1,1,t} & \dots & 1_{1,16,t} \\ \vdots & \ddots & \vdots \\ 1_{7776,1,t} & \dots & 1_{7776,16,t} \end{bmatrix}, \forall t. \tag{28}$$

b: STEP II: SELECT INITIAL ACTION

At the first time step $t = 0$, a random action is selected by the agent from Eq. (20) at tick $t^*_{k^*,t}$. The action is a bid or offer price (p^b_{i,t,k^*} or p^s_{j,t,k^*}) depending on if the agent is a consumer or producer agent, respectively and $p^{\perp} < p^b_{i,t,k^*}, p^s_{j,t,k^*} < p^b$. If the agent was unable to make a trade at time tick k^*, another action is selected by the agent in the next time tick (k^*+1), following Eq. (29) or (30) for a buyer or seller agent, respectively.

$$a^b_{u,t^\epsilon} = \begin{Bmatrix} p^b_{i,t,(k^*+1)}: & k^*+1 < k \\ p^b: & k^*+1 = k \end{Bmatrix}, \forall t, \tag{29}$$

Subject to:

$$p^b_{i,t,(k^*)} < p^b_{i,t,(k^*+1)} < p^b$$

$$a^s_{u,t^\epsilon} = \begin{Bmatrix} p^s_{j,t,(k^*+1)}: & k^*+1 < k \\ p^{\perp}: & k^*+1 = k \end{Bmatrix}, \forall t, \tag{30}$$

Subject to:

$$p^s_{j,t,k^*} > p^s_{j,t,(k^*+1)} > p^{\perp}.$$

Here, a^b_{u,t^ϵ} and a^s_{u,t^ϵ} are the random actions taken by a buyer and seller agents, respectively. Eqs. (29) and (30) show that the agents buy/sell from/to the upstream grid at the last market tick k using the upstream grid buying/selling price $p^b/p^{\perp}$ if they are unable to buy from the local community. Assuming that trade took place at $\mathcal{K}$ time tick, then, $p^b_{i,t,\mathcal{K}}$ and $p^s_{j,t,\mathcal{K}}$ are the trade price for buyer and seller agents, i and j, respectively. Since this is the first market time step, there is no previous market results to define the current state, therefore, there is no need calculating the reward and updating the Q-table. However, the process of calculating reward and updating the Q-table after every action is done every time step except the first time step which is the initialization time step.

2) STAGE II: REWARD CALCULATION, TAKING ACTION, AND Q-TABLE UPDATE

a: STEP I: REWARD CALCULATION

At the next time step, $t + 1$, the result of the previous market time step containing the average market trade rate $\overline{p}$, average trade volume $\overline{v}$, the median trade rate $\widehat{p}_t$ and the agent trade price is used to calculate the reward functions as represented in Eqs. (31) and (32) for the buyer and seller agents, respectively.

$$r_t^{b',\$} = \begin{Bmatrix} 0.7 \times (\widehat{p}_t - p^b_{i,t,\mathcal{K}})/(p^b - \widehat{p}_t): & p^b_{i,t,\mathcal{K}} > \widehat{p}_t \\ 0: & p^b_{i,t,\mathcal{K}} = \widehat{p}_t \\ 0.7 \times (\widehat{p}_t - p^b_{i,t,\mathcal{K}})/(\widehat{p}_t - p^{\perp}): & p^b_{i,t,\mathcal{K}} < \widehat{p}_t \end{Bmatrix}, \forall t, \tag{31}$$

$$r_t^{s',\$} = \begin{Bmatrix} 0.7 \times (p^s_{j,t,\mathcal{K}} - \widehat{p}_t)/(p^b - \widehat{p}_t): & p^s_{j,t,\mathcal{K}} > \widehat{p}_t \\ 0: & p^s_{j,t,\mathcal{K}} = \widehat{p}_t \\ 0.7 \times (p^s_{j,t,\mathcal{K}} - \widehat{p}_t)/(\widehat{p}_t - p^{\perp}): & p^s_{j,t,\mathcal{K}} < \widehat{p}_t \end{Bmatrix}, \forall t, \tag{32}$$

Consequently, Eqs.(26) and (27) is updated as Eqs. (33) and (34), respectively.

$$r_t^{b'} = r_t^{b',\$} + r_t^{\mu}, \tag{33}$$

$$r_t^{s'} = r_t^{s',\$} + r_t^{\mu}. \tag{34}$$

b: STEP II: TAKING ACTION

In order to balance the exploration/exploitation challenges, the ϵ greedy strategy is used for taking action. Eq. (35) is used to select an action by first selecting a random number λ,

$$a_{u,t+1} = \begin{Bmatrix} a_{u,(t+1)^\epsilon}: & \lambda < \epsilon \\ a_{u,(t+1)^{1-\epsilon}}: & \lambda > \epsilon \end{Bmatrix}, \forall t. \tag{35}$$

If $\lambda < \epsilon$, a random (exploration) action is taken by applying Eq. (29) or (30) for a buyer or seller agent, respectively. On the other hand, if $\lambda > \epsilon$, the action with the maximum Q-value is selected by applying Eq. (36).

$$a_{u,(t+1)^{1-\epsilon}} = \arg\max \sum \mathcal{Q}_{s\times a,t}. \tag{36}$$

c: STEP III: UPDATING Q-TABLE

In the same way, the Q-value is updated for both buyers and sellers agents as represented in Eqs. (37) and (38), respectively.

$$\mathcal{Q}_{(t+1)}\left(s_t, a^b_{u,t^\epsilon}\right) \leftarrow \mathcal{Q}_t\left(s_t, a^b_{u,t^\epsilon}\right) + \alpha$$

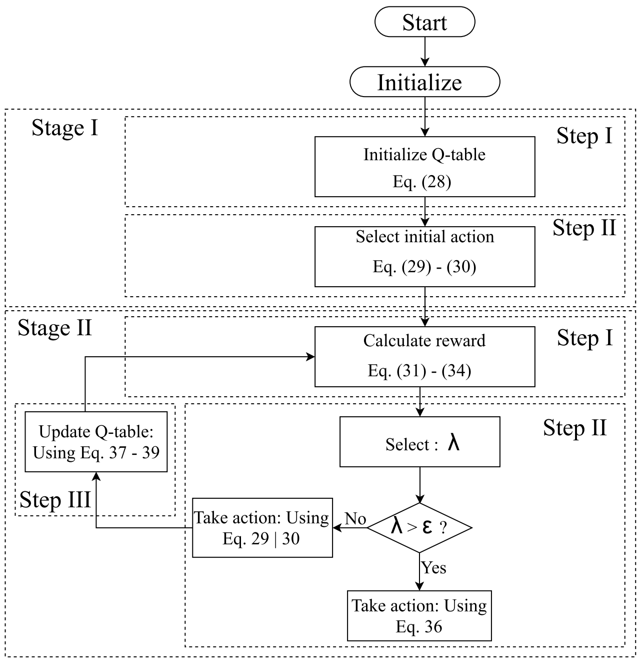

FIGURE 3. Flowchart for Q-learning algorithm.

$$\left[r_t^{b'} + \gamma \max_a Q_t\left(s_{(t+1)}, a^b_{u,t^\epsilon}\right) - Q_t\left(s_t, a^b_{u,t^\epsilon}\right)\right], \tag{37}$$

$$Q_{(t+1)}\left(s_t, a^s_{u,t^\epsilon}\right) \leftarrow Q_t\left(s_t, a^s_{u,t^\epsilon}\right) + \alpha \left[r_t^{s'} + \gamma \max_a Q_t\left(s_{(t+1)}, a^s_{u,t^\epsilon}\right) - Q_t\left(s_t, a^s_{u,t^\epsilon}\right)\right], \tag{38}$$

Here, a^b_{u,t^ϵ} and a^s_{u,t^ϵ} are the actions for the previous time step, t for buyer and seller agent, respectively and does not mean that the initial action is used for all time steps. Eqs. (37) and (38) are used to update the Q-table as represented in Eq. (39).

$$Q_{s\times a,t} = \begin{bmatrix} q_{1,1,t} & \cdots & q_{1,16,t} \\ \vdots & \ddots & \vdots \\ q_{7776,1,t} & \cdots & q_{7776,16,t} \end{bmatrix}, \forall t. \tag{39}$$

Stage II is repeated every time step until end of the experiment. The pseudocode and flowchart of the proposed Q-learning bidding strategy is represented in Algorithm 1 -Appendix A and Fig. 3, respectively.

C. SARSA STRATEGY

The methodology for the SARSA bidding/offering strategy is similar to the Q-learning strategy. The discrepancy between the two models is that in the SARSA strategy the Q-value is updated based on the temporal difference using SARSA update equation as represented in Eqs. (40) and (41) for buyer and seller agent, respectively. This is equally reflected on the flowchart of Fig. 3. The pseudocode of the proposed SARSA bidding strategy is represented in Algorithm 2 in Appendix A.

$$Q_{(t+1)}\left(s_t, a^b_{u,t^\epsilon}\right) \leftarrow Q_t\left(s_t, a^b_{u,t^\epsilon}\right) + \alpha \left[r_t^{b'} + \gamma \max_a Q_t\left(s_{(t+1)}, a_{u,(t+1)}\right) - Q_t\left(s_t, a^b_{u,t^\epsilon}\right)\right], \tag{40}$$

$$Q_{(t+1)}\left(s_t, a^s_{u,t^\epsilon}\right) \leftarrow Q_t\left(s_t, a^s_{u,t^\epsilon}\right) + \alpha \left[r_t^{s'} + \gamma \max_a Q_t\left(s_{(t+1)}, a_{u,(t+1)}\right) - Q_t\left(s_t, a^s_{u,t^\epsilon}\right)\right], \tag{41}$$

IV. SIMULATION CASE STUDY

A. SIMULATION FRAMEWORK AND DATA

The proposed LEM reinforcement bidding strategy model was developed as a Python code and implemented in the bidding agent application programming interface (API) of the open-source Grid Singularity Exchange (GSy-E) [8], [38]. The bidding agent API is used for creating the bids and offers and posting them to the exchange engine. The exchange matches the bids and offers using a two-sided pay-as-bid clearing mechanism and send the results with market information back to the bidding agents [38]. Bids and offers not matched at the local community are traded with the upstream grid. In our simulation model, each prosumer agent communicates its bids or offers individually to the exchange engine every 15 minute time slot before the energy exchange time. Each 15-minute time slot has 4 ticks, therefore, prosumer agents are able to update their bids/offers three times within a market slot. At the fourth market tick, the agent bids/offers the upstream grid price/feed-in tariff price for PV, respectively. The model is verified in a simulation case study of a community with 43 participants consisting of 26 households with only consumption devices and 17 households with consumption and production devices. Load profiles used are taken from [39]. Moreover, the PV production profiles are generation data from Renewables Ninja [40], [41] using Stuttgart region as a community and scaled down from hourly resolution to 15-minutes slot. To ensure that all PV of the same capacity do not produce exactly the same quantity of electricity, the PV system losses are varied between 5% and 15% with a constant tilt angle of 35°. Because of the cost of household electricity in Germany, the cost of buying electricity from the upstream grid is capped at 31.5 ct/kWh while the sell price to upstream grid is 11.00 ct/kWh which is the PV feed-in tariff price [42].

B. ASSUMPTIONS AND SIMULATION METHODS

In order to analyze the behaviour of the individual agents and to verify the implication of the agents working towards the same goal or benefits, the developed Q-learning and SARSA algorithms are verified in three different forms namely ''classical'', ''standalone actor'' and ''shared rewards algorithms'' The performance of the different reinforcement learning algorithms are compared to the baseline to establish a lower limit on the performance and complexity of the subsequent models. A random bidding agent is used as a baseline model

TABLE 1. Simulation parameters.

S/N	Parameter	Symbol
1	Learning rate	α
2	Exploration/exploitation rate	ϵ
3	Discount factor	γ
4	Episodes	Epi.
5	Simulation time	T

TABLE 2. Parameter for classical Q-learning and SARSA algorithms.

Environment	Algorithm	α	ϵ	γ	Epi.	T
Training	SARSA	0.1	0.8	0.4	182	17446
Testing	SARSA	0.1	0	0.4	56	5368
Training	Q-learning	0.1	0.8	0.4	182	17446
Testing	Q-learning	0.1	0	0.4	56	5368

TABLE 3. Parameter details for the Q-learning and SARSA standalone actor algorithms in training environment.

Algorithm	Actor	α	ϵ	γ	Epi.	T
SARSA	Pros.	0.1	0.8	0.9	182	17446
SARSA	Cons.	0.1	0.8	0.9	182	17446
Q-learning	Pros.	0.1	0.8	0.9	182	17446
Q-learning	Cons.	0.1	0.8	0.9	182	17446

in this research to compare it with the intelligent agent models. The agent randomly makes bids or offers based on the amount of energy required or available. The random strategy for generation (producer) begins with an arbitrary price and steadily decreases as the number of ticks increase. However, the consumer agent strategy steadily increases until a match is discovered. For a classical Q-learning and SARSA algorithm, the model is the traditional Q-learning and SARSA algorithm described in Section III. Table 1 displays the simulation parameters for training and testing of the simulations. T is the total simulation 15-minutes time steps and Epi. is the number of episodes.

Table 2 shows the simulation parameters for the classical Q-learning and SARSA algorithms for training and testing of the simulations. The standalone actor algorithm is used to get a better understanding of individual prosumers and consumers agents. Hence, for this algorithm, consumers trade using the intelligent bidding strategy while the prosumers use the baseline strategy and vice versa. This algorithm helps isolate each actor's policy and get a more refined knowledge into single actor's profits. Furthermore, it is evident that when two intelligent agents trade as a single entity, a difference between each of their profits is anticipated, unlike when just a single agent trades with an unintelligent agent. It is expected that in the first case that the model strikes a balance between the two, but in the latter, the model considers the relevance of just one actor. Here, the purpose is to experiment and analyse both scenarios. Table 3 shows the parameter details for training of Q-learning and SARSA standalone actor algorithms.

The goal of the shared reward case is to develop a model that will simultaneously benefit both prosumers and

TABLE 4. Parameter details for the Q-learning and SARSA shared rewards algorithm.

Environment	Algorithm	α	ϵ	γ	Epi.	T
Training	SARSA	0.01	0.8	0.9	364	34892
Training	SARSA	0.1	0.8	0.9	364	34892
Testing	SARSA	0.01	0.8	0.9	62	5933
Testing	SARSA	0.1	0.8	0.9	62	5933
Training	Q-learning	0.1	0.8	0.9	364	34892
Testing	Q-learning	0.1	0.8	0.9	62	5933

consumers since using the classical SARSA and Q-learning algorithms was unable to do that. Additionally, for our model, the consumer and prosumer actors do not communicate with one another and are unaware of each other's existence. To solve this, we define an element of cooperation between the actors for them to communicate information not directly but by working towards a certain goal. Consequently, the classical algorithms are modified to develop a shared-reward concept that allows agents to learn dynamically and coordinate with one another to motivate them to cooperate on the global reward aim, that is, to benefit the entire local community. Thus, the global reward is the sum of the individual prosumers and consumers agents rewards within the community. All agents now work towards maximizing this global reward. Table 4 shows the parameter details for shared reward algorithm.

C. PARAMETER TUNING

After development of the SARSA and Q-learning model, parameter tuning was performed to determine the value of the input parameters that leads to optimal rewards. The three parameters used for the SARSA and Q-learning model are the learning rate (α), exploration/exploitation rate (ϵ) and discount factor (γ). The value of each of these parameters range from 0 to 1. The parameter tuning procedure is divided into 3 stages. First, sensitivity analysis was used to analyze the entire range of values of the parameters to determine the few optimal values of the parameters. In the second stage, the initial parameter tuning is carried out by using the optimal values selected from sensitivity analysis to repeat the simulations and the results compared to determine the parameters that outperforms the other. The optimal values from the second stage are then used to perform the final tuning simulations and the results compared. The best results are then determined as the optimal parameters of the model. As Q-learning and SARSA showed similar features, the parameter tuning was performed only for the SARSA algorithm.

V. RESULTS

In this Section, the simulation results are presented and discussed briefly. The first paragraph presents the general simulation analysis, afterwards, the results are analysed based on LEM performance indicators. Lastly, the simulations is analysed to determine the optimal parameters for the bidding agents.

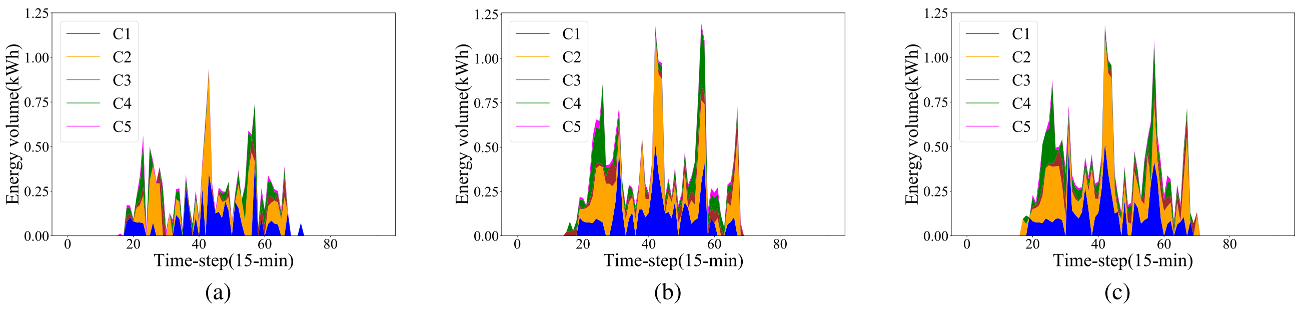

FIGURE 4. Selected consumers internal traded energy for (a) baseline, (b) classical Q-learning and (c) classical SARSA strategies.

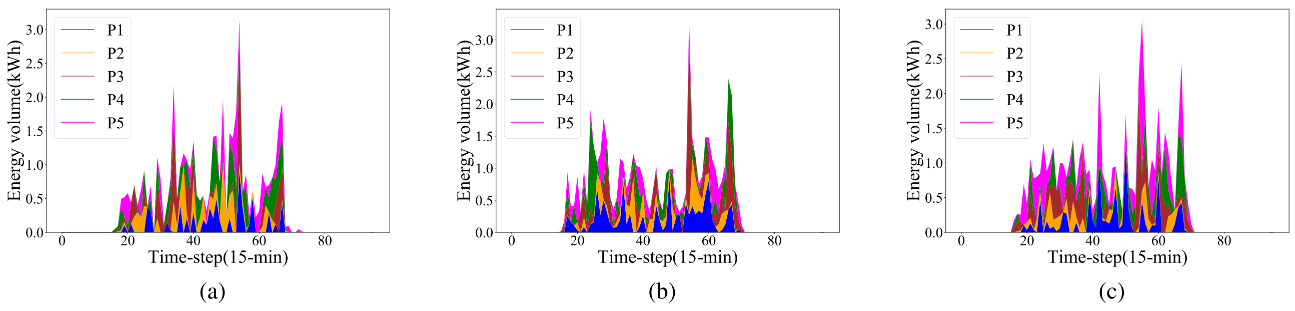

FIGURE 5. Selected prosumers internal traded energy for (a) baseline, (b) classical q-learning and (c) classical SARSA strategy.

A. GENERAL SIMULATION ANALYSIS

The results of the simulation are analyzed based on the quantity of energy traded between the prosumers and consumers, energy imported from the upstream grid to the community and energy exported from the local community to the upstream grid. Fig. 4 displays the internal traded energy of five selected consumers for the (4a) baseline, (4b) classical q-learning and (4c) classical SARSA strategies of a selected day with the training environment. The internal traded energy of the consumers is the energy bought by the consumers from prosumers within the local community. From Fig. 4, for all the scenarios, energy is traded majorly during the day. This is evident because the major source of energy within the community is PV and therefore, energy is generated and traded within the community mainly during the day. Furthermore, for the five selected consumers, the consumers trade more energy within the local community in the classical Q-learning and SARSA strategy compared to the baseline strategy. It is evident that the intelligent strategies (Q-learning and SARSA) provide opportunity for consumers to trade more energy within the local community compared to the baseline strategy.

Fig. 5 displays the internal traded energy of five selected prosumers for the (5a) baseline, (5b) classical q-learning and (5c) classical SARSA strategies of a selected day with the training environment. The internal traded energy of the prosumers is the energy sold/bought by the prosumers to/from consumers/prosumers within the local community. Similar to the consumers, as shown in Fig. 5, for all the scenarios, energy is traded majorly during the day because, the major source of energy within the community is PV. Therefore, energy is generated and traded within the community mainly during the day. Furthermore, for the five selected prosumers, the prosumers trade more energy within the local community in the classical Q-learning and SARSA strategy compared to the baseline strategy. This provides further evidence to support that the intelligent strategies create more opportunity for prosumers to trade energy within the local community compared to the baseline strategy.

Fig. 6 displays the community (6a) traded energy, (6b) energy import from the upstream grid and (6c) energy export to the upstream grid using the baseline, classical SARSA, shared Q-learning, shared SARSA, standalone Q-learning, and standalone SARSA strategies for a single day in the training environment. From Fig. 6a, the prosumers and consumers trade less energy using the baseline strategy compared to other strategies. This is because the internal traded energy of the baseline strategy is less compared to other strategies. Also, using the baseline strategy, the consumers and prosumers import more energy from the upstream grid compared to other strategies as shown in Fig. 6b. Furthermore, the prosumers export more energy to the upstream grid while using the baseline strategy compared to other strategy - Fig 6c.

B. ANALYSIS OF SIMULATION BASED ON PERFORMANCE INDICATORS

Fig. 7 and 8 display the average and cumulative reward, respectively, of consumers and prosumers actors for the classical, standalone and shared reward strategies for the entire training period. Fig. 7 displays the average reward over the

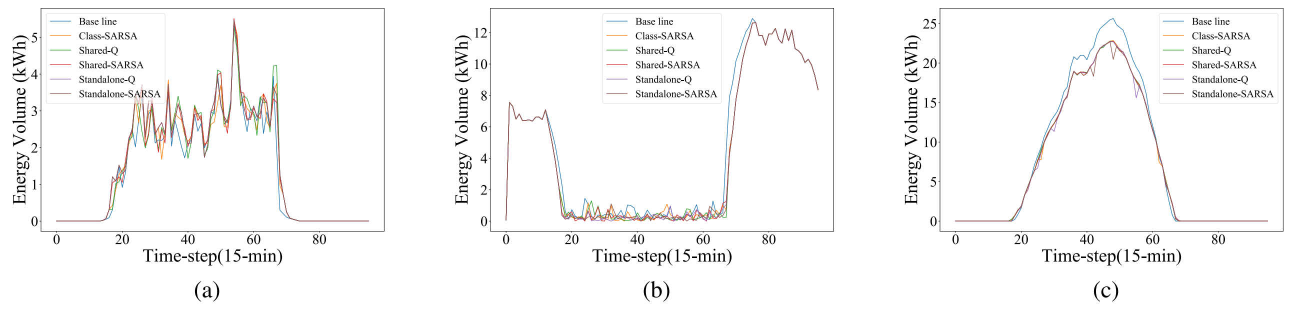

FIGURE 6. Community (a) traded energy (b) energy import and (c) export from/to the upstream grid.

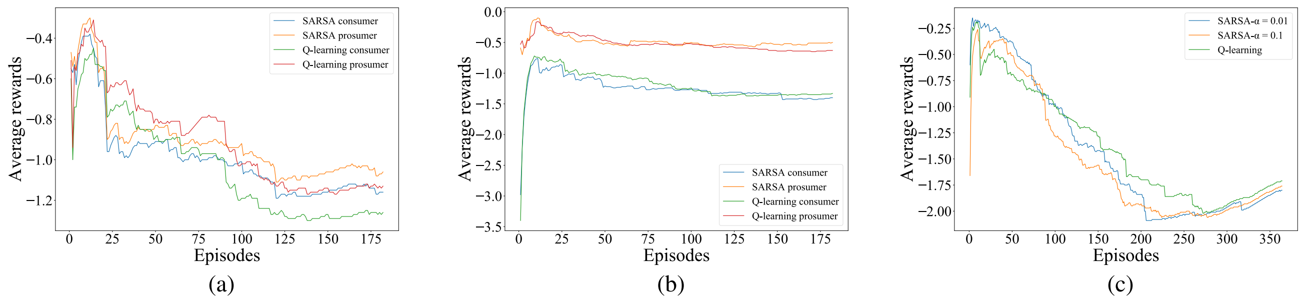

FIGURE 7. Average reward for (a) classical (b) standalone and (c) shared reward strategy.

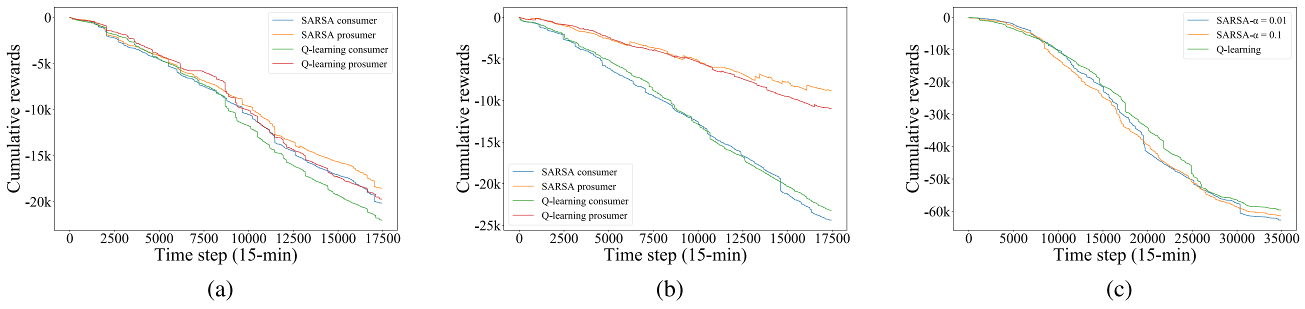

FIGURE 8. Cumulative average reward for (a) classical (b) standalone and (c) shared reward strategy.

entire episodes while Fig. 8 displays the cumulative reward over the entire 15-mins time step. The higher the average reward, the better the algorithm. Fig. 7a and Fig. 8b show that for the classical strategies, prosumers gather more rewards compared to the prosumers. At the same time, SARSA strategy gather more rewards compared to the Q-learning strategy. This shows that SARSA strategy has the capability of producing more benefits to the LEM compared to the Q-learning strategy while using the classical algorithm. Also, the classical algorithms has the capability of yielding more benefits for the prosumers compared to the consumers. From Fig. 7b and 8b, similar to the classical algorithms (Fig. 7a), the prosumers gather more rewards compared to consumers for the standalone strategies. However, the consumers gather more rewards using the Q-learning algorithm compared to the SARSA algorithm. On the other hand, prosumers gather more reward using the SARSA algorithm compared to the Q-learning algorithm. For the shared reward, the prosumers and consumers agents work towards a common goal of increasing their global reward. Therefore, the rewards of Fig. 7c and 8c are the global average and cumulative rewards, respectively. The Q-learning algorithm gathers more rewards compared to the SARSA algorithm. It is evident that Q-learning can provide more benefits to both consumers and prosumers while helping them to achieve a common goal. Generally comparing the classical, standalone, and shared reward strategies rewards show that the agents gathers most rewards with the standalone strategy and least reward with the shared reward strategy.

Table 5 presents the varied bidding strategies verified using the LEM performance indicators and their accompanying symbols. The performance indicators used are self sufficiency, market revenue, individual average consumer savings, individual average prosumer savings and average

TABLE 5. Varied strategies.

S/N	Simulation scenario	Symbol
1	Baseline	L_0
2	Classical Q-learning	L_1
3	Classical SARSA	L_2
4	Standalone SARSA - Consumer	L_3
5	Standalone SARSA - Prosumer	L_4
6	Standalone Q-learning - Consumer	L_5
7	Standalone Q-learning - Prosumer	L_6
8	Shared reward SARSA- $\alpha = 0.01$	L_7
9	Shared reward SARSA- $\alpha = 0.1$	L_8
10	Shared reward Q-learning	L_9

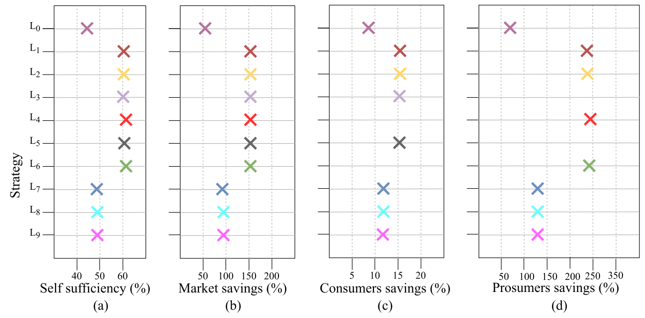

FIGURE 9. Performance indicators ((a) Self-sufficiency, (b) share of market savings, (c) share of consumer savings, and (d) share of prosumer savings) for varying scenarios in training environment.

trade rate. Fig. 9 and 10 display the self-sufficiency (SS), share of market savings (SMS), share of consumers savings (SCS), and share of prosumers savings for the different strategies L_0 to L_9, in training and testing environment, respectively. SMS is the percentage of savings made by the prosumers and consumers for trading within the LEM compared to without LEM. In the same way, the share of consumer/prosumer savings is the percentage of savings made by the consumer/prosumer for trading within the LEM compared to without LEM. The standalone strategies are not experimented with the testing environment because it is not obtainable in practice to keep some group of agents intelligent in a community while others are not. Therefore, the standalone strategies is used to verify the individual capabilities of the agents. For the training environment (Fig. 9), the community SS is higher with the classical and standalone strategies. This can be explained as consequent upon the rewards gathered by the classical and standalone strategy during training which is higher compared to the shared reward strategy. The maximum SS is obtained with the L_4 strategy. For this strategy, the prosumers use intelligent strategy based on SARSA while the consumers use the baseline strategy. Since trading energy within the community is more beneficial compared to the upstream grid, this strategy ensures that the prosumers trade all their energy within the community thereby making the community to be sufficient. This further results in higher SMS of the local community and share of prosumer savings. The baseline strategy (L_0) shows the least SS, SMS, SCS as well as share of prosumer savings. This is because the agents are not intelligent and do not bid/offer strategically to make the optimum benefit from the LEM. The

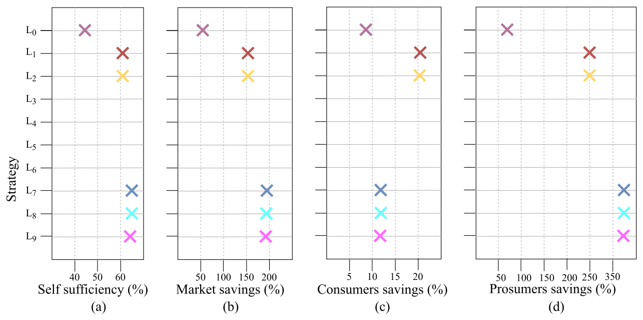

FIGURE 10. Performance indicators ((a) Self-sufficiency, (b) share of market savings, (c) share of consumer savings, and (d) share of prosumer savings) for varying scenarios in testing environment.

shared reward strategies (L_7 to L_9) have the least value of SS, SMS, SCS and share of prosumer savings compared to other intelligent strategies. This is highlighted by the least rewards collected by the shared reward strategies during training.

From Fig. 10, the SS and SMS of the shared reward strategies (L_7 to L_9) is higher compared to other strategies. It is evident that the shared strategy require more training time for the prosumers and consumers agents to determine their optimal trading strategy. Furthermore, trading towards a common community goal (global reward) results in higher SS and SMS of the community. However, the SCS of L_7 to L_9 is lower compared to other intelligent strategies. On the other hand, the share of prosumer savings is higher with L_7 to L_9 compared to the other strategies. Since the consumers and prosumers agents of the L_7 to L_9 strategies work towards a common goal of increasing the global reward, however, the offering/bidding strategy that results in more global reward benefits the prosumers compared to the consumers. This is because the prosumers own the PV's and batteries and thereby invested into the market, therefore, trading more of the energy from the PV in the LEM creates more benefits to the prosumers.

Fig. 11 shows the comparison of the different LEM strategies using LEM performance indicator in the training environment. At this stage, a new strategy termed the hybrid (H1) strategy is introduced. The Hybrid strategy is similar to the classical strategies. Hence, it is a combination of the Q-learning and SARSA classical algorithm strategies. For the hybrid strategy, the prosumers agents are modelled using the classical SARSA strategy while the consumers are modelled using the classical Q-learning strategy. From Fig. 11, the H1 strategy shows better SMS and SS compared to other strategies. However, the SCS of the H1 strategy is lower compared to the L_1 and L_2 strategies. By integrating the features of the SARSA and Q-learning strategies, the H1 strategy was able to leverage the features of the two strategies to create technical and economic benefits for the LEM.

Fig. 12 shows the comparison of the different LEM strategies using LEM performance indicator in the testing environment. Unlike the training environment, the L_7 to L_9 strategies show better SS, SMS and SCS compared to the L_1 and L_2 strategies. However, the H1 strategy still shows higher

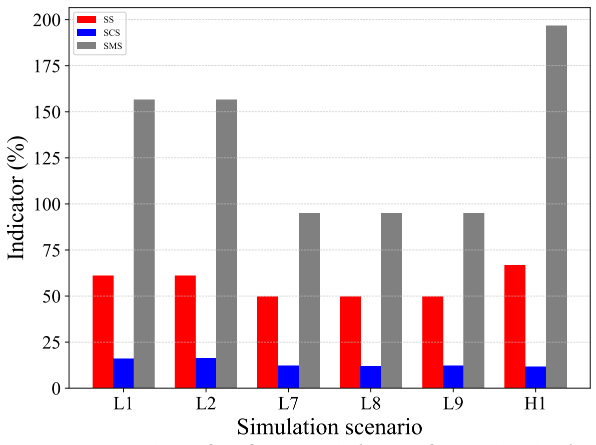

FIGURE 11. Comparison of performance indicators for varying simulation scenarios in training environment.

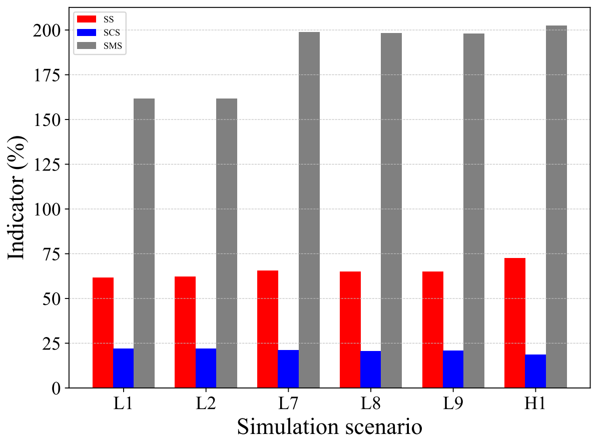

FIGURE 12. Comparison of performance indicators for varying simulation scenarios in testing environment.

SS and SMS compared to the L_7 to L_9. This is evident that utilizing SARSA strategy with prosumer agents while using Q-learning strategy for consumer agents creates more benefits to the local community compared to other strategies tested in this work.

C. PARAMETER TUNING

Appendix B-A (Table. 6, Fig. 13, and Fig. 14) contains the results of the sensitivity analysis of α, ϵ and γ parameter tuning. Four optimal values of α, ϵ and γ were selected from the sensitivity analysis and used for initial parameter tuning. The results of the initial parameter tuning is presented in Appendix B-B (Table. 7, Fig. 15, and Fig. 16). After the initial parameter tuning, two best values of α, ϵ and γ are then selected and used for the final parameter tuning. The results of the final parameter tuning are presented in Appendix B-C (Table. 8, Fig. 17, Table. 9, and Fig. 18) and a detailed discussion of the parameter tuning is presented in Section VI-D.

VI. DISCUSSION

The Section presents discussion about findings, insights and improvements of the model.

A. CLASSICAL MODELS COMPARISON

Firstly, the baseline model is used as a reference to assess the LEM performance indicators such as self sufficiency(SS), share of market savings (SMS), share of consumer savings (SCS), and share of prosumers savings. The baseline model and reinforcement learning (RL) algorithms are trained and compared, followed by testing of the RL algorithms. In general, the baseline model has satisfactory results of the LEM performance indicators with a SS of 44.75%, SMS of 56.99% and average trade rate of 23.63 ct./kWh. The SS and SMS increase significantly while the average trade rate decreased by deploying RL algorithm bidding strategy in the local community. The RL agent learns to bid/offer according to the current state which completely describes the observation needed to bid/offer. In the training environment, in comparison to the baseline model, average SS and SMS of the RL models increased from 44.75% to 61.20% and 56.99% to 156.65%, respectively, while the average trade rate of electricity is lowered from 23.63 ct./kWh to 22.01 ct./kWh. The SS and SMS in a testing environment compared to the training environment increase from 61.20% to 62.0% and 156.65% to 161.61%, respectively. However, the major changes is witnessed with the average trade rate which reduced from 22.01 ct./kWh to 19.75 ct./kWh. A lower community average trade rate is beneficial to the consumers and provides opportunity for them to buy locally produced electricity. This further results in additional technical and economic benefits of the community witnessed with the increase in SS and SMS, respectively. Comparing the increase in the share of savings of prosumers and consumers from baseline model to intelligent agents models reveals that the share of prosumers savings increase significantly higher than the consumers share of savings. Thus, the prosumers are the major contributors of the community SMS. This can be attributed to the investment made by the prosumers to the market by providing the PV and battery which are the source of power in the local community.

The RL rewards are the second criteria used for analysis and comparison of the model performance. However, throughout the comparison and evaluation, it is observed that the change in the RL rewards is higher compared to the changes in LEM performance indicators and so RL rewards are used as the deciding factors in most cases. Here, the ZI model is out of comparison because of lack of an agent. The consumers and prosumers act as a separate entity in a single agent model and so it is necessary to analyse their rewards differently. The average and cumulative rewards are negative in the training environment and can be attributed to be resulting from the high exploration rate (ϵ) used during the simulation. As the agent tries to explore new actions, it is possible that there are very few actions that result in positive rewards and so the agent is not able to compensate for the wrong actions received during the training process. This changes in the testing environment where no exploration is used, and the agent only chooses the action that leads to higher and positive rewards. Another observation to note is that the average reward per episode for consumer is higher than the average reward for the prosumer. This could be because the trades (demand) needed for the consumers are higher than the trades (supply) of the prosumers. In the training environment, the rewards collected by the SARSA algorithm is higher than Q-learning for both consumers and prosumers. In the testing environment, this is not the case. SARSA algorithm collects higher rewards for the consumer actor whereas Q-learning algorithm gains higher rewards for the prosumer. This means that one single-agent model does not work equally for both actors.

B. STANDALONE ACTORS

The standalone actor models are developed to isolate the behaviour of each actor and analyse their behaviour separately as the only intelligent agent in the system. Four models are developed in the training environment for this purpose. The two prosumer models have higher SS and SMS compared to the two consumer models. The average trade rate is higher for the prosumer models and lower for the consumer models showing the presence of only a single actor rather than two. The results of SS, average trade rate and SMS for the consumer models are close to the classical SARSA and Q-learning models, which means that the classical RL models are more influenced by the consumers rather than prosumers. The prosumer models are beneficial for the welfare of the community as it results in higher SS and SMS compared to the consumer models. The standalone consumer model is responsible for the lower average trade rate whereas the standalone prosumer model helps to increase the economic benefits.

The rewards gathered by the consumer and prosumer actors in the training environment are negative. The Q-learning algorithm provides opportunity to collect higher rewards compared to SARSA for the consumer models. However, for the prosumer models, the SARSA algorithm collect higher rewards compared to Q-learning model. It is important to note that the standalone actor models converged faster over time than the classical RL models. The average rewards gained by the prosumer models for the standalone algorithms are also comparatively higher than the prosumer rewards in classical RL models. This suggests that the standalone prosumer model performs better than the classical single-agent RL models in some criteria. This is because two entities in a single agent model might hinder each other's progress and can perform better as a standalone actor.

C. SHARED REWARDS CONCEPT

Shared rewards concept is implemented in SARSA and Q-learning models to analyze the possibility of prosumers and consumers to pursue the same goal in a single experiment. Two SARSA models with different learning rate and one Q-learning model are developed to analyse the shared rewards concept. The values of LEM performance indicators are lower in the training environment than the testing environment. The training was done for one year of simulation that includes both winter and summer season. That is one of the reasons why the SS and SMS is lower, as there is less solar production. The values of the LEM KPIs are higher in the SARSA algorithm rather than Q-learning algorithm with hardly noticeable difference which can results in the observed lower performance indicators. On the other hand, the RL rewards have a different outcome. For both the training and testing environment, Q-learning has constantly gained higher rewards than the SARSA models. Thus, there is no consistent model outperforming the other in both criteria. The graph of the global average rewards in the training environment is still increasing by the end of the training period which suggests that the algorithm might not have converged yet and would perform better if the training time was increased. However, because of the huge training time and lack of data, the training is only performed for 1 year. The lack of convergence can also be seen from the global average rewards in the testing environment which are negative. The LEM performance indicators in the testing environment show better performance compared to the testing environment of the classical RL models. The individual share of savings of both consumers and prosumers are substantially increased. The higher values of the LEM performance indicators of the shared reward models is evident that the shared reward models outperforms the classical models. However, the performance in the RL rewards is not as expected because of the insufficient training time. If the model was given enough training time for it to learn the cooperation between the agents, the shared reward model will outperform the classical RL models in terms of the obtained rewards.

D. PARAMETER TUNING

The three steps used are sensitivity analysis, initial parameter tuning and then, final parameter tuning. Starting with the sensitivity analysis, the change in the learning rate (α) does not result in significant change of the performance indicators. The lower values of α such as 0.1, 0.2, and 0.3 along with some higher values such as 0.7, result in comparatively higher LEM performance indicators. However, the RL reward criterion shows values of the learning rate such as 0.4 and 0.9 can help the algorithm collect more rewards compared to other values. Therefore, one optimum value from each criterion and one default and also most common learning rate are selected for the initial parameter tuning. The only parameter that has a linear relationship with the LEM performance indicator is the exploration rate (ϵ). The lower the values, the better the result. However, there are some outliers to this relationship in the RL rewards criteria. For example, ϵ equal to 0.6 and 0.7 collects higher rewards than 0.5 which is true for the consumer and prosumer rewards because of the exploration vs exploitation trade-off. Thus, the equal probability for both exploitation and exploration is not beneficial for the algorithm. To verify this anomaly further, ϵ equal to 0.2, 0.5 and 0.9 were chosen to compare further. The discount factor (γ) shows the most non-linear behaviour out of the three parameters. There is no rule of thumb on choosing γ, on the other hand, the value of 0.9 or 0.99 helps the algorithm converge faster. Considering

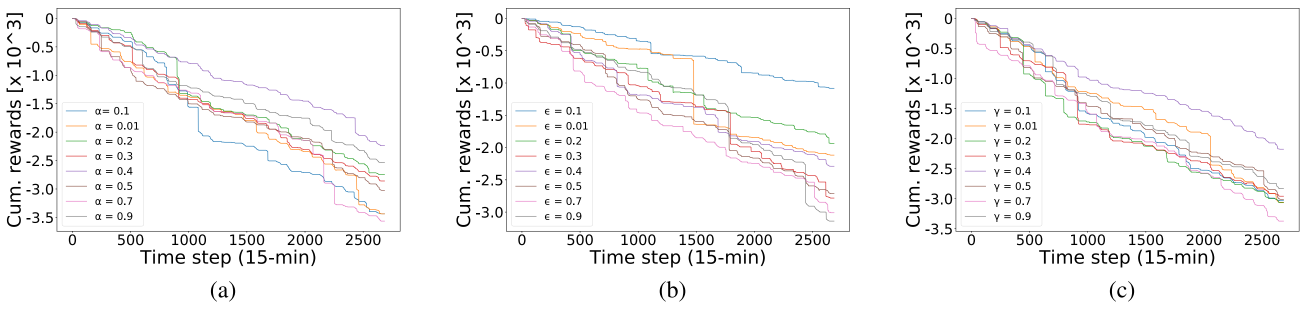

FIGURE 13. Sensitivity analysis of consumers rewards for (a) α, (b) ϵ and (c) γ.

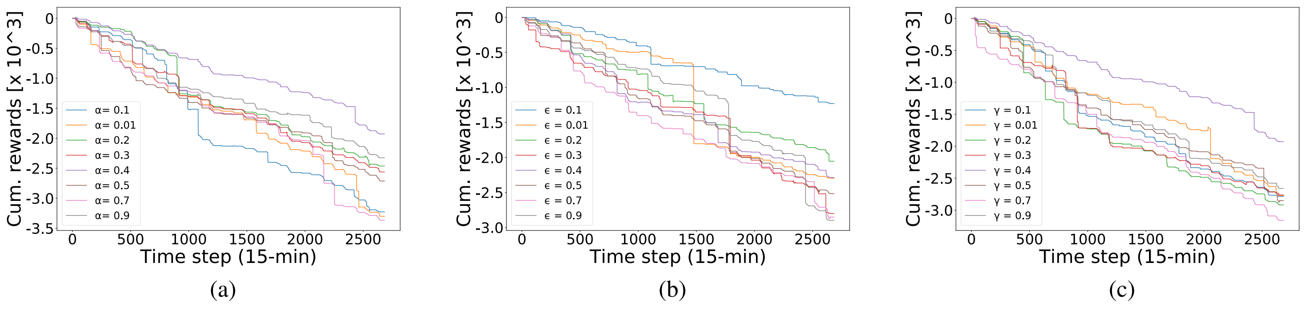

FIGURE 14. Sensitivity analysis of prosumers rewards for (a) α, (b) ϵ and (c) γ.

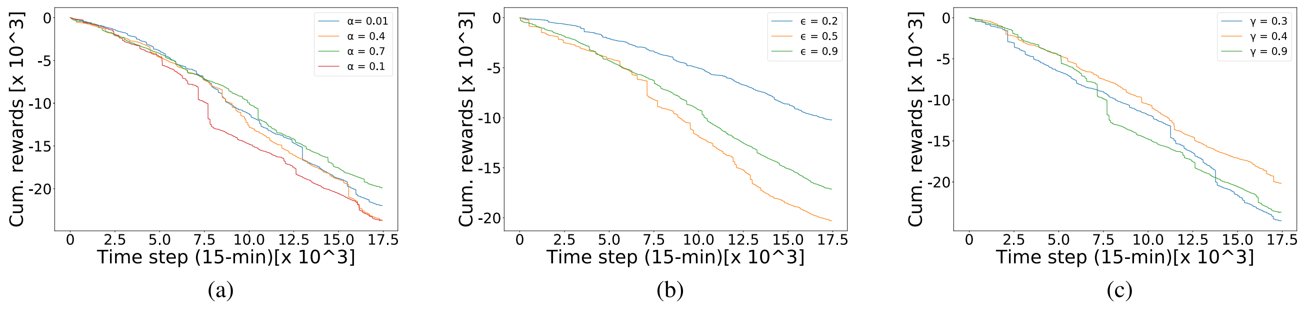

FIGURE 15. Initial parameter tuning of consumers rewards for (a) α, (b) ϵ and (c) γ.

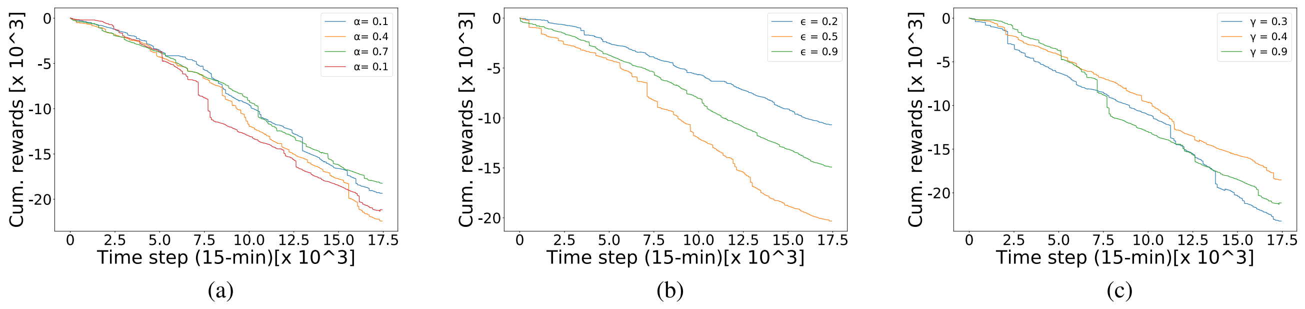

FIGURE 16. Initial parameter tuning of prosumers rewards for (a) α, (b) ϵ and (c) γ.

the LEM performance indicators, comparatively lower values of γ such as 0.1, 0.3 and 0.4 are better whereas for the RL rewards, higher values of gamma such as 0.5, 0.7 and 0.9 shows better performance. The value of the discount factor is usually higher compared to the learning rate and so three values, 0.3, 0.4 and 0.9 are used to further provide the comparison in the next step.

From the result of the initial parameter tuning for α, the default rate of 0.1 lead to the worst performance in the RL rewards criteria. Nevertheless, the LEM performance

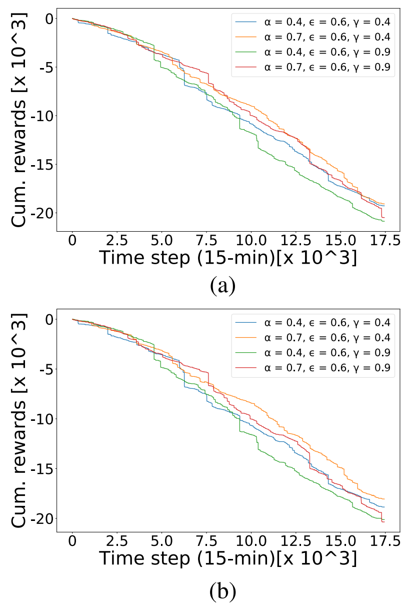

FIGURE 17. Final parameter tuning of rewards for (a) consumers and (b) prosumers in training environment.

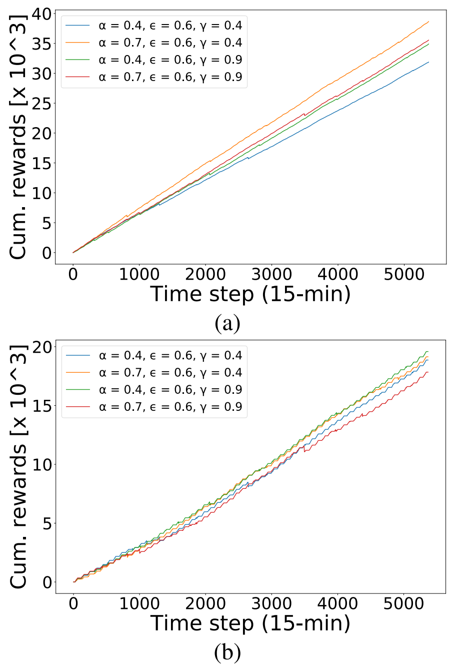

FIGURE 18. Final parameter tuning of rewards for (a) consumers and (b) prosumers in testing environment.

indicators for α equal to 0.1 and 0.4 were almost comparable. The initial parameter tuning of ϵ has the similar outcomes to its sensitivity analysis. $\epsilon = 0.5$ has lower rewards compared to $\epsilon = 0.9$. Given that the relationship of epsilon is quite straightforward with the evaluation criteria, only one value of epsilon 0.6 is used for the final comparison. The initial parameter tuning of gamma showed that there is no correlation between gamma and any of the KPIs. The two best values of 0.4 and 0.9 are selected from the criteria for the final

Algorithm 1 Q-Learning Algorithm Bidding/Offering Strategy

```
Require: $\mathcal{S}_t, \alpha, \epsilon, \gamma$ .        ▷ Initialize parameters and states
Initialize: $\mathcal{Q}_{s\times a,t}$ according to Eq. (28)
Initial action: according to Eq. 29 & 30

while (Epi > 0) do
    Calculate reward: according to Eq. (34) | 33
    Take action: according to Eq. 36 & 35
    Update Q-table: $Q_{(t+1)}\left(s_t, a^b_{u,t^\epsilon}\right) \leftarrow Q_t\left(s_t, a^b_{u,t^\epsilon}\right)$
        $+\alpha\left[r^{b'}_t + \gamma \max_a Q_t\left(s_{(t+1)}, a^b_{u,t^\epsilon}\right)\right.$
        $\left.-Q_t\left(s_t, a^b_{u,t^\epsilon}\right)\right]$        ▷ For buyers
        $Q_{(t+1)}\left(s_t, a^s_{u,t^\epsilon}\right) \leftarrow Q_t\left(s_t, a^s_{u,t^\epsilon}\right)$
        $+\alpha\left[r^{s'}_t + \gamma \max_a Q_t\left(s_{(t+1)}, a^s_{u,t^\epsilon}\right)\right.$
        $\left.-Q_t\left(s_t, a^s_{u,t^\epsilon}\right)\right]$        ▷ For sellers
        Update Eq. 39
end while
```

tuning. Hence, it is certain from here, that the values of α, ϵ, and γ do not follow the common and default values used for them. Therefore, the behaviour of each value and parameter to the algorithm is challenging to interpret. Another important and common observation is that the best values of α, ϵ, and γ for the consumer and prosumer models are in most cases different from each other which also explains the reason for the development of standalone actor algorithms.

The final parameter tuning consists of two α and γ values and one ϵ value to determine the final optimal values. The results obtained from the training and testing environment are different from each other. Learning rate (α) of 0.7 and γ of 0.4 has the highest value in terms of LEM performance indicators and the RL rewards in the training environment. However, for the testing environment, the model with the best LEM performance indicators has $\alpha = 0.4$ and $\gamma = 0.9$. However, this is not the same for rewards collected by consumers. The model with the best rewards collected by consumer has $\alpha = 0.7$ and $\gamma = 0.4$. Therefore, the model with the most optimal bidding/offering strategy is the single-agent SARSA model with $\alpha = 0.4$, $\epsilon = 0.6$, and $\gamma = 0.9$.

VII. CONCLUSION

In this paper, two novel reinforcement learning based intelligent bidding strategies, Q-learning and SARSA, were proposed for effective trading of distributed energy resources for prosumers and consumers in an LEM. The proposed models were tested in a German real case scenario and simulated for 45 German households. The simulations results show that the intelligent bidding strategies create additional technical and economic benefits to the local consumers and prosumers compared to the baseline strategy. Furthermore, the proposed models reveal that when the intelligent agents within the local community work towards a common goal, in this case shared reward strategy, the model create additional benefits for the

TABLE 6. Comparison of participant revenue for sensitivity analysis.

Parameter	$\alpha = 0.1$	$\alpha = 0.01$	$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.4$	$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 0.9$
Consumer 1	16.48%	15.91%	15.94%	16.18%	16.36%	15.95%	16.43%	15.99%
Prosumer 1	261.24%	261.54%	261.43%	261.59%	261.51%	261.41%	261.27%	261.50%
Parameter	$\epsilon = 0.2$	$\epsilon = 0.3$	$\epsilon = 0.4$	$\epsilon = 0.5$	$\epsilon = 0.6$	$\epsilon = 0.7$	$\epsilon = 0.8$	$\epsilon = 0.9$
Consumer 1	22.90%	21.89%	20.38%	18.81%	18.09%	16.82%	16.0%	14.44%
Prosumer 1	259.57%	259.64%	259.92%	260.24%	260.27%	260.81%	261.49%	261.89%
Parameter	$\gamma = 0.1$	$\gamma = 0.3$	$\gamma = 0.4$	$\gamma = 0.5$	$\gamma = 0.7$	$\gamma = 0.9$	$\gamma = 0.99$	$\gamma = 0$
Consumer 1	16.16%	16.14%	15.89%	16.30%	15.83%	16.00%	15.99%	15.81%
Prosumer 1	261.52%	261.65%	261.22%	261.20%	261.28%	261.49%	261.49%	261.35%

Algorithm 2 SARSA Algorithm Bidding/Offering Strategy

```
Require: $\mathcal{S}_t, \alpha, \epsilon, \gamma$ .        ▷ Initialize parameters and states
Initialize: $\mathcal{Q}_{s \times a,t}$ according to Eq. (28)
Initial action: according to Eq. 29 & 30

  while (Epi > 0) do
      Calculate reward: according to Eq. (34) | 33
      Take action: according to Eq. 36 & 35
      Update Q-table: $Q_{(t+1)}\left(s_t, a^b_{u,t^\epsilon}\right) \leftarrow Q_t\left(s_t, a^b_{u,t^\epsilon}\right)$
          $+\alpha\left[r^{b'}_t + \gamma \max_a Q_t\left(s_{(t+1)}, a_{u,(t+1)}\right)\right.$
          $\left.-Q_t\left(s_t, a^b_{u,t^\epsilon}\right)\right]]$        ▷ For buyers
          $Q_{(t+1)}\left(s_t, a^s_{u,t^\epsilon}\right) \leftarrow Q_t\left(s_t, a^s_{u,t^\epsilon}\right)$
          $+\alpha\left[r^{s'}_t + \gamma \max_a Q_t\left(s_{(t+1)}, a_{u,(t+1)}\right)\right.$
          $\left.-Q_t\left(s_t, a^s_{u,t^\epsilon}\right)\right]]$        ▷ For sellers
          Update Eq. 39
  end while
```

TABLE 7. Comparison of participant revenue for initial parameter tuning.

Parameter	$\alpha = 0.01$	$\alpha = 0.1$	$\alpha = 0.4$	$\alpha = 0.7$
Consumer 1	16.09%	16.39%	16.05%	16.44%
Prosumer 1	244.23%	244.03%	244.02%	244.11%
Parameter	$\epsilon = 0.2$	$\epsilon = 0.5$	$\epsilon = 0.9$	-
Consumer 1	20.92%	18.91%	15.29%	-
Prosumer 1	242.87%	243.37%	244.42%	-
Parameter	$\gamma = 0.3$	$\gamma = 0.4$	$\gamma = 0.9$	-
Consumer 1	16.32%	16.29%	16.44%	-
Prosumer 1	244.14%	244.18%	244.11%	-

community compared to the classical strategies. The most optimal strategy is the hybrid strategy which is a combination of the classical Q-learning and SARSA strategies. In future work, we will investigate other artificial intelligent models such as deep learning and how they can be modelled for LEM trading and how the model will be implemented in a distributed blockchain platform to ensure efficient preservation of prosumers' privacy and conformation to data protection laws.

APPENDIX A ALGORITHMS PSEUODO CODES

See Algorithm 1 and Algorithm 2.

APPENDIX B FURTHER SIMULATION RESULTS

A. TABLES AND FIGURES FOR SENSITIVITY ANALYSIS

See Table 6, Figures 13 and 14.

TABLE 8. Comparison of participant revenue for final parameter tuning (training env).

Parameters	$\alpha = 0.4$ $\gamma = 0.4$	$\alpha = 0.7$ $\gamma = 0.4$	$\alpha = 0.4$ $\gamma = 0.9$	$\alpha = 0.7$ $\gamma = 0.9$
Consumer 1	18.10%	18.32%	18.07%	18.36%
Prosumer 1	243.52%	243.59%	243.65%	243.54%

TABLE 9. Comparison of participant revenue for final parameter tuning (testing env).

Parameters	$\alpha = 0.4$ $\gamma = 0.4$	$\alpha = 0.7$ $\gamma = 0.4$	$\alpha = 0.4$ $\gamma = 0.9$	$\alpha = 0.7$ $\gamma = 0.9$
Consumer 1	20.56%	20.26%	19.77%	19.54%
Prosumer 1	260.35%	260.39%	260.94%	260.59%

B. TABLES AND FIGURES FOR INITIAL PARAMETER TUNING

See Table 7, and Figures 15 and 16.

C. TABLES AND FIGURES FOR FINAL PARAMETER TUNING

See Tables 8 - 9 and Figures 17 - 18.

ACKNOWLEDGMENT

The authors would like to thank their anonymous reviewers.

REFERENCES

[1] E. Mengelkamp, J. Diesing, and C. Weinhardt, "Tracing local energy markets: A literature review," *Inf. Technol.*, vol. 61, nos. 2–3, pp. 101–110, Apr. 2019, doi: 10.1515/ITIT-2019-0016.

[2] C. Weinhardt, E. Mengelkamp, W. Cramer, S. Hambridge, A. Hobert, E. Kremers, W. Otter, P. Pinson, V. Tiefenbeck, and M. Zade, "How far along are local energy markets in the DACH+ region? A comparative market engineering approach," in *Proc. 10th ACM Int. Conf. Future Energy Syst.*, New York, NY, USA, Jun. 2019, pp. 544–549.

[3] A. Wörner, V. Tiefenbeck, F. Wortmann, A. Meeuw, L. Ableitner, E. Fleisch, and I. Azevedo, "Bidding on a peer-to-peer energy market: An exploratory field study," *Inf. Syst. Res.*, vol. 33, no. 3, pp. 794–808, Sep. 2022.

[4] E. Mengelkamp, J. Gärttner, and C. Weinhardt, "Intelligent agent strategies for residential customers in local electricity markets," in *Proc. 9th Int. Conf. Future Energy Syst.*, New York, NY, USA, Jun. 2018, pp. 97–107.

[5] R. Ghorani, M. Fotuhi-Firuzabad, and M. Moeini-Aghtaie, "Optimal bidding strategy of transactive agents in local energy markets," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5152–5162, Sep. 2019.

[6] T. Gokcek, I. Sengor, and O. Erdinc, "A novel multi-hierarchical bidding strategy for peer-to-peer energy trading among communities," *IEEE Access*, vol. 10, pp. 23798–23807, 2022.

[7] E. Mengelkamp, P. Staudt, J. Garttner, and C. Weinhardt, "Trading on local energy markets: A comparison of market designs and bidding strategies," in *Proc. 14th Int. Conf. Eur. Energy Market (EEM)*, Jun. 2017, pp. 1–6.

[8] G. C. Okwuibe, A. S. Gazafroudi, S. Hambridge, C. Dietrich, A. Trbovich, M. Shafie-khah, P. Tzscheutschler, and T. Hamacher, "Evaluation of hierarchical, multi-agent, community-based, local energy markets based on key performance indicators," *Energies*, vol. 15, no. 10, p. 3575, May 2022. [Online]. Available: https://www.mdpi.com/1996-1073/15/10/3575

[9] K. Chen, J. Lin, and Y. Song, "Trading strategy optimization for a prosumer in continuous double auction-based peer-to-peer market: A prediction-integration model," *Appl. Energy*, vol. 242, pp. 1121–1133, May 2019.

[10] F. Lezama, J. Soares, R. Faia, P. Faria, and Z. Vale, "Bidding in local energy markets considering uncertainty from renewables and demand," in *Proc. IEEE Int. Conf. Environ. Electr. Eng. IEEE Ind. Commercial Power Syst. Eur.*, Sep. 2021, pp. 1–6.

[11] A. Yu, C. Zhang, and Y.-J. A. Zhang, "Optimal bidding strategy of prosumers in distribution-level energy markets," *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 1695–1706, May 2020.

[12] Z. Zhang, H. Tang, P. Wang, Q. Huang, and W.-J. Lee, "Two-stage bidding strategy for peer-to-peer energy trading of nanogrid," *IEEE Trans. Ind. Appl.*, vol. 56, no. 2, pp. 1000–1009, Mar. 2020.

[13] Y. Zhou and J. Wu, "Simulating ancillary service provision from a peer-to-peer energy trading community—Data," Tech. Rep., 2020, doi: 10.17035/D.2020.0114344992.

[14] Q. Jia, Y. Li, Z. Yan, C. Xu, and S. Chen, "A reinforcement-learning-based bidding strategy for power suppliers with limited information," *J. Mod. Power Syst. Clean Energy*, vol. 10, no. 4, pp. 1032–1039, 2022.

[15] J. Wang, J. Wu, and X. Kong, "Multi-agent simulation for strategic bidding in electricity markets using reinforcement learning," *CSEE J. Power Energy Syst.*, early access, Nov. 9, 2021, doi: 10.17775/CSEEJPES.2020.02820.

[16] A. J. M. Kell, M. Forshaw, and A. S. McGough, "Exploring market power using deep reinforcement learning for intelligent bidding strategies," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2020, pp. 4402–4411.

[17] M. Rahimiyan and H. R. Mashhadi, "An adaptive *Q*-learning algorithm developed for agent-based computational modeling of electricity market," *IEEE Trans. Syst., Man, Cybern., C, Appl. Rev.*, vol. 40, no. 5, pp. 547–556, Sep. 2010.

[18] Y. Ye, D. Qiu, M. Sun, D. Papadaskalopoulos, and G. Strbac, "Deep reinforcement learning for strategic bidding in electricity markets," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1343–1355, Mar. 2020.

[19] Q. Yu, Y. Liu, D. Xia, and L. Martinez, "The strategy evolution in double auction based on the experience-weighted attraction learning model," *IEEE Access*, vol. 7, pp. 16730–16738, 2019.

[20] T. Chen and W. Su, "Local energy trading behavior modeling with deep reinforcement learning," *IEEE Access*, vol. 6, pp. 62806–62814, 2018.

[21] J. Zhou, K. Wang, W. Mao, Y. Wang, and P. Huang, "Smart bidding strategy of the demand-side loads based on the reinforcement learning," in *Proc. IEEE Conf. Energy Internet Energy Syst. Integr. (EI2)*, Nov. 2017, pp. 1–5.

[22] D. Peng, H. Xiao, W. Pei, H. Sun, and S. Ye, "A novel deep learning based peer-to-peer transaction method for prosumers under two-stage market environment," *IET Smart Grid*, vol. 2022, pp. 1–10, Jun. 2022.

[23] G. C. Okwuibe, M. Wadhwa, T. Brenner, P. Tzscheutschler, and T. Hamacher, "Intelligent bidding strategies in local electricity markets: A simulation-based analysis," in *Proc. IEEE Electr. Power Energy Conf. (EPEC)*, Nov. 2020, pp. 1–7.

[24] Y. Tao, J. Qiu, and S. Lai, "Deep reinforcement learning based bidding strategy for EVAs in local energy market considering information asymmetry," *IEEE Trans. Ind. Informat.*, vol. 18, no. 6, pp. 3831–3842, Jun. 2022.

[25] A. Trbovich, S. Hambridge, D. van den Biggelaar, E. Hesse, and F. Sioshansi, "D3A energy exchange for a transactive grid," in *Behind Beyond Meter*. Amsterdam, The Netherlands: Elsevier, 2020, pp. 267–284.

[26] T. Mildenberger, "Stephen marsland: Machine learning. an algorithmic perspective," *Stat. Papers*, vol. 55, no. 2, p. 575, 2014.

[27] E. Alpaydin, *Introduction to Machine Learning*. Cambridge, MA, USA: MIT Press, 2020.

[28] D. Silver. (2015). *Lectures on Reinforcement Learning*. Accessed: Apr. 10, 2022. [Online]. Available: https://www.davidsilver.uk/teaching/

[29] M. van der Ree and M. Wiering, "Reinforcement learning in the game of Othello: Learning against a fixed opponent and learning from self-play," in *Proc. IEEE Symp. Adapt. Dyn. Program. Reinforcement Learn. (ADPRL)*, Apr. 2013, pp. 108–115.

[30] M. Tokic and G. Palm, "Value-difference based exploration: Adaptive control between epsilon-greedy and softmax," in *Proc. Annu. Conf. Artif. Intell.* Berlin, Germany: Springer, 2011, pp. 335–346.

[31] D. Silver. (2018). *Open AI Spinning Up: Key Concepts in RL*. Accessed: Jun. 22, 2022. [Online]. Available: https://spinningup.openai.com/en/latest/

[32] R. Moni. (2019). *Reinforcement Learning Algorithms—An Intuitive Overview*. Accessed: Mar. 13, 2022. [Online]. Available: https://tinyurl.com/2p8vuywy

[33] L. Owen. (2020). *Bird's-Eye View of Reinforcement Learning Algorithms Taxonomy*. Accessed: Mar. 13, 2022. [Online]. Available: https://tinyurl.com/vwn7yutd

[34] C. Kowshik. (2020). *Off-Policy vs on-Policy vs Offline Reinforcement Learning Demystified!*. Accessed: Mar. 13, 2022. [Online]. Available: https://tinyurl.com/58j5pzka

[35] C. J. C. H. Watkins and P. Dayan, "*Q*-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.

[36] M. V. Otterlo and M. Wiering, "Reinforcement learning and Markov decision processes," in *Reinforcement Learning*. Berlin, Germany: Springer, 2012, pp. 3–42.

[37] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[38] *Grid Singularity Technical Approach*, Grid Singularity, Berlin, Germany. Accessed: May 19, 2022. [Online]. Available: https://gridsingularity.github.io/gsy-e/technical-approach/

[39] T. Tjaden, J. Bergner, J. Weniger, and V. Quaschning, "Representative electrical load profiles of residential buildings in Germany with a temporal resolution of one second," Dataset, HTW Berlin University of Applied Sciences, Feb. 2022. [Online]. Available: https://solar.htw-berlin.de/elektrische-lastprofile-fuer-wohngebaeude/

[40] S. Pfenninger and I. Staffell. *Renewables Ninja*. Accessed: Dec. 7, 2021. [Online]. Available: https://www.renewables.ninja/

[41] S. Pfenninger and I. Staffell, "Long-term patterns of European PV output using 30 years of validated hourly reanalysis and satellite data," *Energy*, vol. 114, pp. 1251–1265, Nov. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544216311744

[42] *What German Households Pay for Power*. Accessed: Apr. 14, 2022. [Online]. Available: https://tinyurl.com/2re6knnj

GODWIN C. OKWUIBE (Member, IEEE) received the B.Eng. degree in electrical engineering from the University of Nigeria, Nsukka, Nigeria, in 2013, and the M.Sc. degree in power engineering from the Technical University of Munich, Munich, Germany, in 2019, where he is currently pursuing the Ph.D. degree. He is also a Researcher with OLI Systems GmbH, Stuttgart, Germany. His research interests include energy markets, game theory, integration of renewable energy to the power grid, distributed generation, and the application of blockchain technology to energy markets.

JEEL BHALODIA received the B.Tech. degree in computer engineering from Charusat University, Gujarat, India, in 2019, and the M.Sc. degree in data science from the Berliner University of Applied Sciences, Berlin, Germany, in 2022. She is currently a Working Student at OLI Systems GmbH, Stuttgart, Germany. Her research interests include trustworthy AI, Bayesian model-based reinforcement learning, transfer learning, FinTech regulations, and anti-money laundering using network modeling.

AMIN SHOKRI GAZAFROUDI received the B.Sc. and M.Sc. degrees in power electrical engineering, in 2011 and 2013, respectively, and the Ph.D. degree in computer science from the University of Salamanca, Salamanca, Spain, in 2019. From October 2019 to December 2021, he was a Postdoctoral Researcher involving in CoNDyNet2 Project with the Energy System Modeling Research Group, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany. He is currently an Innovation Manager at OLI Systems GmbH, Stuttgart, Germany, to draw technology road-map for OLI ecosystem products in energy communities, local energy and flexibility markets, peer-to-peer (P2P) energy transactions, smart grids, smart homes, electric vehicle (EV) charging strategies, and coordinating research and development national and international projects with OLI Systems GmbH. His research interests include power system and electricity market modeling, power flow and contingency analysis, local energy and flexibility markets design, peer-to-peer energy trading in smart grids, market-based coordination mechanisms, decentralized energy management systems, bidding strategies for autonomous home energy management systems, planning and operation of integrated energy systems, and application of machine learning algorithms on price and demand forecasting.

THOMAS BRENNER received the M.Sc. degree in interdisciplinary sciences from ETH Zürich, Switzerland, and the Ph.D. degree from Cambridge University, U.K., with a thesis on the physics of polymer solar cells.

He headed the Junior Research Group "Hybrid Optoelectronics," University of Potsdam, Germany, until he moved to Dr. Langniß–Energie & Analyse as a Senior Consultant, in 2015. He was responsible for the European Smart Grid Project "CALLIA" and was actively involved in the SINTEG project "C/sells." At OLI Systems, he is responsible for the interface between the energy sector on one hand and the design and development of hard- and software on the other.

PETER TZSCHEUTSCHLER received the Diploma degree in electrical engineering and information technology from the Technical University of Munich (TUM), in 1998, and the Ph.D. (Dr.-Ing.) degree for his doctoral thesis on "The Global Potential of Solar-thermal Electricity Generation," in 2005.

He joined the Chair of Energy Economy and Application Technology, TUM, as a Research Associate. From 2009 to 2011, he was a Team Leader of a Group of the TUM International Graduate School for Science and Engineering on the topic "Building–User–Climate." From 2011 to 2014, he was an Operating Agent of Annex 54 "Integration of Microgeneration and Related Technologies in Buildings" an international research group within the framework of the Energy in Buildings and Communities Program of the International Energy Agency. He is currently working at the TUM Chair for Energy Economy and Application Technology. His research interests include decentralised and renewable energy systems with a special focus on building energy supply, cogeneration, energy management, and local energy markets.

THOMAS HAMACHER received the Ph.D. degree from the University of Hamburg, Hamburg, Germany, in 1994, for his work on baryonic beta decay after studying physics in the University of Bonn, Bonn, Germany, RWTH Aachen University, Aachen, Germany, and Columbia University, New York, NY, USA.

He has been with the Max Planck Institute for Plasma Physics, Garching bei München, Germany, since 1996, most recently as the Head of the Energy and System Studies Group. From 2010 to 2013, he worked as the Acting Head of the Chair of Energy Management and Application Technology, Technical University of Munich (TUM), Munich, Germany. In 2013, he was appointed as a Full Professor of renewable and sustainable energy systems at the TUM. He is also the Director of the Munich School of Engineering, Garching bei München. He is also a member of the Environmental Science Centre (WZU), University of Augsburg, Augsburg, Germany. His research interests include energy and systems analysis, urban energy systems, integration of renewable energy into the power grid, and innovative nuclear systems (including fusion). Other focuses of his work are the methods and fundamentals of energy models.

Prof. Hamacher is also a member of the Energy Working Group of the European Physical Society (EPS).