IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 30, 2020, accepted November 9, 2020, date of publication November 17, 2020,
date of current version December 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3038735

Reinforcement Learning Techniques for Optimal Power Control in Grid-Connected Microgrids: A Comprehensive Review

ERICK O. ARWA, (Graduate Student Member, IEEE), AND KOMLA A. FOLLY, (Senior Member, IEEE)
Department of Electrical Engineering, University of Cape Town, Cape Town 7701, South Africa
Corresponding author: Erick O. Arwa (arweri001@myuct.ac.za)

This work was based on the research supported in part by the National Research Foundation of South Africa under Grant 118550.

ABSTRACT Utility grids are undergoing several upgrades. Distributed generators that are supplied by intermittent renewable energy sources (RES) are being connected to the grids. As RES get cheaper, more customers are opting for peer-to-peer energy interchanges through the smart metering infrastructure. Consequently, power management in grid-tied RES-based microgrids has become a challenging task. This paper reviews the applications of reinforcement learning (RL) algorithms in managing power in grid-tide microgrids. Unlike other optimization methods such as numerical and soft computing techniques, RL does not require an accurate model of the optimization environment in order to arrive at an optimal solution. In this paper, various challenges associated with the control of power in grid-tied microgrids are described. The application of RL techniques in addressing those challenges is reviewed critically. This review identifies the need to improve and scale multi-agent RL methods to enable seamless distributed power dispatch among interconnected microgrids. Finally, the paper gives directions for future research, e.g., the hybridization of intrinsic and extrinsic reward schemes, the use of transfer learning to improve the learning outcomes of RL in complex power systems environments and the deployment of priority-based experience replay in post-disaster microgrid power flow control.

INDEX TERMS Electric vehicle charging station, energy management, Markov decision process, microgrid, reinforcement learning.

I. INTRODUCTION

The power grids are experiencing a massive transition due to several technological advances. For instance, the world is gearing toward the use of electric vehicles (EVs) for transportation due to the economic, technical and environmental advantages associated with them [1]. However, the current global electricity generation mix is still predominated by fossil fuels. This negates the environmental benefits of electric mobility [2]. The use of renewable energy sources (RES) to charge EVs would significantly reduce greenhouse gas emissions [3]–[5]. Nevertheless, RES are intermittent and non-dispatchable. They may not be sufficient on their own to meet the power demands of fast charging. The deployment of grid-tied microgrids (GT-MGs) to supply electric vehicles (EVs) would be needed to guarantee continuous and reliable supply of power [6].

Optimal power control involves management of the power system variables to supply the load at minimum or reasonable cost while not violating the system constraints [7]. The main goals of the power optimization process are to minimize energy production and delivery costs, minimize power losses, reduce load shedding, and maximize system performance in general. The objective function captures the cost minimization while the constraints take care of the system health and load-generation balance [8], [9]. In particular, optimal power scheduling encompasses the temporal arrangement of the system's generation resources to achieve the system's objective and maintain its overall health [10].

Despite the advancement in research on renewable energy and smart grid control technologies, grid-tied microgrids still face challenges in the control of their operations. For instance, power management in modern GT-MGs is

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

challenging because of stochasticity in RES and load demand [11]–[13]. Thus, the current matters of great research interest in power systems control include: (a) the scheduling and control of generating units under a stochastic environment, (b) the development of demand response strategies that enable customers to cooperate with the grid in energy trading contracts, and (c) the design of modern voltage and frequency regulation methods to accommodate the increased in power conversion stages [11], [14].

There is a lot of published literature that deals with power systems scheduling. Traditional methods such as dynamic programming (DP), linear programming (LP), and their derivatives have been used to perform power scheduling tasks in GT-MGs for years [15]–[17]. Nevertheless, these approaches have been found to suffer from the infamous curse of dimensionality and are unable to adapt to the stochasticity of the optimization environment that contains unpredictable load profiles, grid tariff and RES generation. Therefore, such methods have limited scalability and versatility. Global search methods such as genetic algorithm (GA) and swarm intelligence (SI) have also been employed in solving the power management problems [7], [18]–[20]. However, these methods are generally slow, thus, they cannot operate online. Online operation enables more economical use of computing resources as it does not require a dedicated computer to do the offline optimization. The above algorithms do not have a learning component. Thus, optimization iterations are needed for every new load and generation profile, which is computationally expensive. Moreover, these approaches require a separate forecasting algorithm to predict the state variables.

Reinforcement learning (RL) algorithms offer a better alternative as they can be trained offline for a general load and generation profile and then be applied online for any load and generation profiles. Unlike other optimization methods such as numerical methods and soft computing techniques, RL does not require an accurate model of the optimization environment in order to arrive at an optimal solution. Furthermore the application of artificial neural networks (ANN) in modern reinforcement learning algorithms effectively eliminates the need for a separate forecasting model because of their capability to make more accurate predictions [10], [21].

Reinforcement learning (RL) is a bio-inspired machine learning technique employed for solving sequential decision-making problems. In this category of algorithms, a software agent is modelled as a decision-maker that learns by iterative trial and error through a carefully defined reward scheme [22]. The agent's key motivation is to maximize its total reward. The environment is characterized by states which the agent observes, selects an action from the set of possible actions and receives feedback on the value of the actions taken in each of the states [10]. RL algorithms require that the problem be expressed as a multi-stage decision problem (MDP). The MDP model offers a formal mathematical language describing sequential control operations. In such operations, the outcomes are partly uncertain and partly informed by the actions of the decision-making agent [23]. An MDP consists of a set of states within a definite state-space, a set of possible actions within a definite action-space, a reinforcement function and a state transition function or probability [21], [24]. The agent's objective is to maximize the total reward. The reward is any scalar quantity that can be used to implicitly communicate the objective of the learning activity to the agent [25]. Thus, suitable reward shaping is essential to achieve the desired objective [26].

RL has been applied for solving unit commitment problems, economic dispatch problems, microgrid energy management problems, energy trading in microgrids, etc [10], [27]–[29]. Although it takes longer to train an RL algorithm, the training can be done offline. After the training, optimal solutions are retrieved for the whole optimization horizon. This eliminates the need to iterate during online operations. Despite the merits, the application of RL in power management is still in its infancy.

There are several review papers on the application of RL in power systems operation and control. However, there are few reviews on the use of RL in power management in GT-MGs. The study in [30] focuses on the use of RL in demand response. Authors in [31] studied the application of RL to control energy flow in buildings. In [32], deep RL methods applied to smart grid control are reviewed. However, the above papers did not capture the recent developments in the field of RL such as policy optimization, intelligent reward schemes and the parallelization of learning agents were not captured. There is a need for a review paper that can comprehensibly capture recent developments in RL and provides valuable directions for future research in reinforcement learning applications in control of power in GT-MGs.

In this paper, a critical and comprehensive review of reinforcement learning (RL) approaches to power management in grid-tied microgrids is presented. The main contributions of the paper include:

(1) Providing new insights into the challenges associated with the control of power in GT-MGs using grid-tied RES-based EV charging stations as a case study.
(2) Synthesis of the different mathematical formulations of the power management problem and how each formulation affects the computational efficiency of RL solutions.
(3) A critical review of proposed solutions to the issues associated with current RL techniques such as instability in Q-learning type recursions applied to deep RL, data inefficiency and the bias-overfitting conundrum.
(4) Discussion of possible applications of recent RL techniques such as priority experience replay and transfer learning to address some of the emerging challenges in grid-tied microgrid power management.

The rest of this paper is organized as follows. In section II, the two major mathematical formulations of the power management problem are reviewed. Section III deals with the Markov Decision Process Models used to describe the GT-MGs operational environments. In section IV, the reinforcement learning solutions to the power management

problem are reviewed. Section V highlights the emerging reinforcement learning techniques such as transfer learning, hindsight experience replay, curiosity driven learning etc., and their possible applications in solving modern power management problems. Section VI discusses the strengths and limitations of the RL techniques and the possible improvements. In section VII gives the conclusions on contributions of the paper and future research directions.

II. MATHEMATICAL FORMULATIONS

A typical grid-tied microgrid-based EV CS consists of an AC or DC bus connected to the grid through an appropriate power converter. The bus is connected to the EV supply equipment (EVSE) via an appropriate power conversion apparatus. More details on the EV charger architectures and design considerations in grid-tied EV CSs can be found in [5], [33]–[35].

The objectives of optimal power management algorithms for GT-MGs is to minimize running cost and to maximize profit from energy sales while supplying the demand on the microgrid side [7], [18], [36]. For a battery behind meter systems, the essential variables to solve the problem include the DG power output, the load profile, the measured battery storage system (BSS) instantaneous state of charge (SoC) and the forecasted day-ahead grid tariff profile. The costs considered include the grid power purchase cost, the cost of degradation of the BSS as defined in [28], [29], [39] and the cost of power purchase from auxiliary sources such as vehicle-to-microgrid (V2M) [6], [40], [41]. In some cases, the grid tariff is constant like in [41] and in others, a stochastic tariff is considered as in [42], [43]. The two major mathematical formulations of this optimization problem are the optimal battery scheduling and unit commitment formulations [31]. The two formulations will be discussed in subsections II (A) and II (B).

A. OPTIMAL BATTERY SCHEDULING FORMULATION

This is the most common formulation for GT-MG scheduling tasks. In this formulation, the battery state of charge is the basis of optimization. The solver is designed to consider the load, the RES output and the grid tariff at every time step and find the optimal schedule of BSS SoC. The power balance equation is defined as given in (1) below:

$$P_b(t) = P_L(t) - P_g(t) - \sum_{i=0}^{N} P_{DG_i}(t) \tag{1}$$

where $P_b(t)$, $P_L(t)$ and $P_g(t)$ are the instantaneous values of power delivered to or drawn from the BSS, load demand and power delivered to or drawn from the utility grid respectively. $P_{DG_i}(t)$ is the instantaneous power generated by DG_i and N is the number of DGs. From (1) above, the instantaneous state of charge of the BSS is given by [7]:

$$SoC(t) = SoC(t - \Delta t) - \frac{P_b(t).\Delta t}{E_b} \tag{2}$$

where Δt is the duration of a time step and E_b is the rated battery capacity. The solver then finds the optimal SoC schedule for the BSS as $SoC_1, SoC_2, \ldots, SoC_T$ where T is the optimization horizon [7], [42]. A high-level illustration of a grid-tied PV powered EV charging station is shown in Figure 1.

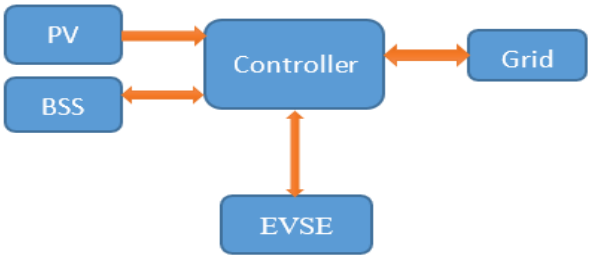

FIGURE 1. A high-level illustration of a grid-connected PV-powered electric vehicle (EV) charging station. It consists of a battery storage system (BSS) and vehicle-to-microgrid (V2M) enabled EV supply equipment (EVSE). The optimizer is embedded in the controller which makes all the power management decisions and relays control signals to the relevant power conversion equipment.

The SoC schedule determines the amount of power to be supplied to or taken from the battery at every time step. The solver then considers the load and commits the remaining RES output to supply the load. The excess RES generator output is then sent to the grid whereas if there is a deficit, power is purchased from the grid. This formulation can be found in [7], [44] and [45]. A major challenge with this method is that if there are more than two sources whose costs are to be considered then resultant schedule may not be optimal. This is because once the battery power has been optimally determined, a suboptimal rule-based method is to be used to determine the schedule for the remaining sources. However, this formulation performs better where there is just one DG involved. It is also less computationally intensive than the unit commitment formulation for the same quantization step size for the battery SoC.

B. UNIT COMMITMENT FORMULATION

This formulation considers each source as a separate unit with the battery as a special unit that can absorb or generate power (prosumer). It is derived from the unit commitment formulation as described in [46]. At every time step, the solver searches the combination of all the available power from sources that minimize the objective function while meeting the constraints. The optimal schedule is determined for all the units simultaneously as described in [5], [31], [47] and [48]. The advantage of this method is that it can accommodate more sources than the battery scheduling approach. Therefore, it allows the combination of several RES to supply the load as well as the integration of V2M technology. However, this formulation is computationally expensive. Table 1 shows a summary of the mathematical formulations with their strengths and weaknesses.

TABLE 1. Summary of mathematical formulations.

Formulation	References	Strengths	Weaknesses
Battery scheduling	[7], [42], [44-45]	Computationally efficient	Sub-optimal results for setups with many sources
Unit commitment	[5], [31], [47-48]	Can handle many sources effectively	Computationally inefficient

III. MARKOV DECISION PROCESS MODELS

A Markov Decision Process (MDP) is usually expressed as a tuple of four elements, (S, A, P, R), where S is the state space, A is the defined action space, P is the state changeover probability and R is the immediate reward obtained in taking a given control action in state S. The process is said to be Markovian if the state space is such that the next state is determined using the current state variables and without the need for the memory of previous events that lead to the current state [18], [19], [21], [22], [38]. Some predictive Markovian processes are modelled using Markov Chains where states are represented as nodes and are linked with directed branches that show the possible transition. EV charging and discharging process models using Markov chains are found in [50]–[52]. MDPs are an essential construction for sequential control problems such as in reinforcement learning methods [21]. Details on how to develop MDPs and model problems can be found in [10] and [21].

A. STATE AND STATE SPACE

The state of a reinforcement learning environment is the set of all information that an agent requires to make the correct choice of the control action to take [53]. The state space is the set of all states in the environment. Depending on the nature of the problem and the amount of accuracy desired on the solution, the state-space may be continuous or discrete. A continuous state-space may also be quantized to make solutions simple and computationally efficient [54]. The question of the amount of discretization required to achieve the desired efficiency within acceptable correctness is still an open problem.

In BBM schemes, the load, grid tariff, BSS energy or SoC and the RES generation are the state variables that may be continuous or discrete. If the agent is provided with all these variables, then the state is said to be fully-observed and deterministic [55]. In some cases, the agent is provided with limited information on the states. The agent uses that information to approximate the likelihood of the environment being in the allowable states and treats these probabilities as the actual state [24]. Such are partially observable MDPs (PO-MDPs) [56], [57]. An example of this formulation in BBM systems can be found in [27]. Stochastic models have also been developed to represent load demand in EV CSs using the "*Spatio-temporal characteristics*" of EV driving and arrivals at the CS [58]. Such models may be essential in approximating the state transitions in an EV charging station environment.

B. ACTION AND ACTION SPACE

The control action is chosen and executed in every state the agent gets to. In BBM systems, the action may be an SoC value in the battery scheduling formulation. As regards this formulation, some researchers only model the actions to three decisions, namely, charge, idle or discharge as given in [59] and [60]. An action in the unit commitment formulation may be a vector of power schedules for each unit as modelled in [48]. The action space is the union of all possible action sets for all the states in the environment.

C. STATE TRANSITION FUNCTION/PROBABILITY

For deterministic environments, the same action in the same state will always lead to the same next state [61]. However, in a stochastic Markovian environment, state transitions are probabilistic. Thus, they may be represented using a transition matrix whose elements are the transition probabilities of various possible next states [62]. Besides, the transitions can be indicated using expectation operators on state transition probability distributions [49]. In EV charging stations, the state transition may include changes in the number of EV arrivals, the SoCs of their various battery packs, fluctuations in PV generator output with irradiance, the amount of energy available in the CS's BSS and the value of the grid tariff.

D. REWARD FUNCTION

A reward is therefore defined as a scalar quantity that can be used to implicitly but effectively communicate the objective of the learning process to the agent [23]. The reward is a function of the action, the state in which it is taken and the state it leads to. Suitable "reward engineering" is needed to link actions with the purposes of the agent in order to learn the optimal policy, i.e., the rule that maps every state to the optimal action [26]. Depending on the objective of the power management algorithm, the reward is defined such that in maximizing the total reward, the agent meets the objective. For instance, to minimize cost, the reward may be defined as the inverse of that cost. The reward may also be expressed as the negative of the cost. However, care must be taken on the effect of the sign of the reward on the stability of the learning process.

IV. REINFORCEMENT LEARNING SOLUTIONS TO THE POWER MANAGEMENT PROBLEM

A. BRIEF INTRODUCTION TO REINFORCEMENT LEARNING

Reinforcement learning (RL) is a computerized reward-directed trial and error method of solving MDPs [63]. In RL, the agent learns by sequentially interacting with the states in

the environment. The goal of the agent is to maximize its total reward. By experience and through a well-designed reward scheme, the agent acquires the knowledge of a policy that maps every state to the best action [15], [28]. The reinforcement learning process is shown in Figure 2. A policy is represented by $\pi(a, s)$, which is the probability that action a will be taken in state s and this may be stochastic or deterministic.

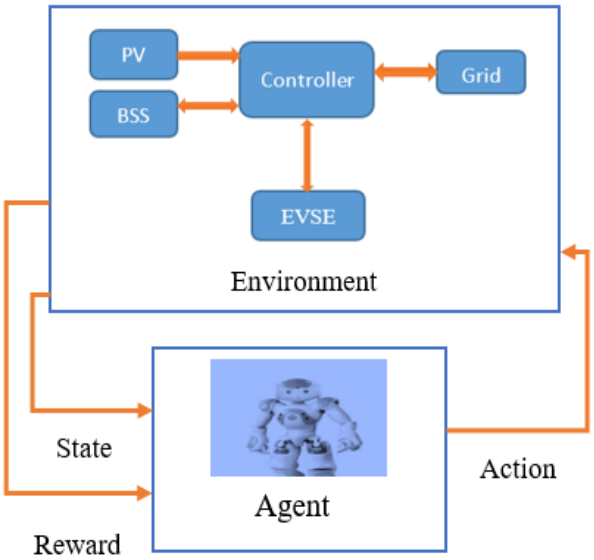

FIGURE 2. An illustration of Reinforcement Learning Agent-Environment interaction. The environment has states that the agent visits sequentially. When the agent takes an action, the environment returns a reward and changes to the next state.

If an agent visits a state x, takes an action suggested by a rule, π, and hence move to next state y, the state-value function in that state is given by:

$$V^{\pi}(x) = r(\pi(x)) + \gamma \sum_{y} P\left[\pi(x)\right] V^{\pi}(y), \tag{3}$$

where $r(\pi(x))$ is the immediate reward, P is the state transition probability from state x to state y, $V^{\pi}(y)$ is the value of state y and $\gamma \in (0, 1)$ is the discount factor that governs the amount of future returns valued in the current state. Bellman [54] established that there is at least one optimal policy which recommends the action that maximizes the value, $V^{\pi}(x)$ in the state x, such that [19], [31]:

$$V^{*}(x) = V^{\pi^{*}}(x) = \frac{max}{a}\left\{ r(a) + \gamma \sum_{y} P\left[\pi(x)\right] V^{\pi^{*}}(y) \right\} \tag{4}$$

is the best value the agent can achieve in the state x, with $r(a)$ as the recommended action. This finding has been the basis of most RL algorithms.

B. EXPLORATION EXPLOITATION CONUNDRUM

During learning, the agent needs to balance exploitation and exploration in its choice of actions to avoid being trapped in a local extremum. The exploration-exploitation conundrum poses a challenge because the agent needs to accumulate experience to take optimal action, and to gather that much data requires that computational resources must be expended. The action that gives the most reward is the greedy action, a_g. Strategies for solving the dilemma of exploration and exploitation have been developed. The strategies include epsilon-greedy [31], the pursuit algorithm [64], the SoftMax function [30], random [65] and deterministic selection [45]. In the ϵ-greedy method, the greedy action is selected by a probability, 1-ϵ, for $\epsilon \in (0, 1)$, in any state x_k, while all other actions in the action space, $\mathcal{A}_k$, are explored by a probability, ϵ [31]. The exploration rate, ϵ, is typically initialized with a value close to 1. It is then gradually decreased as the learning proceeds. That process requires hand tuning of ϵ, which may be inaccurate. In the pursuit algorithm, an action a_k in state x_k has the probability P_k of being selected. P_k is initialized by a constant value for all actions and is updated as given in (5).

$$P_x^{n+1}(a_k) = \begin{cases} P_x^n(a_k) + \beta[1 - P_x^n(a_k)], & a_k = a_g \\ P_x^n(a_k) - \beta P_x^n(a_k), & a_k \neq a_g \end{cases} \tag{5}$$

where $\beta \in (0, 1)$ is a constant and n is the episode number. Therefore, in every learning episode, the pursuit algorithm gradually increases the probability of the greedy action being chosen while slightly reducing the probability of choosing the rest of the actions. Thus, if β is very small, as the number of episodes increases, the probability of choosing a_g in every state will get close to unity while that of choosing all other actions will collapse to near zero [46]. In the softmax method, every action has a probability of being chosen expressed as a function of its Q-value. This is done using the Gibb's distribution function as described in [66]. The probability function is given by (6):

$$p(a|s) = \frac{exp\left\{\frac{Q(s,a)}{\tau}\right\}}{exp\left\{\sum_{a_g \in \mathcal{A}} \frac{Q(s,a')}{\tau}\right\}} \tag{6}$$

where $\tau \in (0, 1)$ is referred to as the temperature. A low value of τ increases the probability of selecting a_g and high value gives all actions an equal chance of being chosen. Tokik [66] hybridized ϵ-greedy with the softmax method and came up with a more adaptive method of exploration called value difference based exploration (VDBE). In this method, the value of ϵ was represented as a function of temporal difference error (TD-error). The author showed that the hybrid method was more robust than the ϵ-greedy and softmax methods.

Several RL algorithms have been developed to solve MDPs. These include Monte Carlo (MC) learning, temporal difference (TD) methods, Q-learning and policy gradient techniques [67], [68], to name but a few. Some of the RL techniques used in solving microgrid scheduling are discussed below.

C. Q-LEARNING

Q-learning is a method of learning the decision-making rule, π, without a model. This method was proposed by Watkins [63]. It is the most widely used RL algorithm due to its simplicity [30]. In Q-learning, the action-value function, $Q(x, a)$, is a measure of the goodness of taking an action a in state x, such that [25]:

$$Q(x, a) = r_x(a) + \gamma \sum_y P[\pi(x)] V^{\pi}(y). \tag{7}$$

The optimal Q-value for the optimal policy is such that:

$$Q^*(x, a) = Q^{\pi^*}(x, a), \quad \forall x \in \chi, \ \forall a \in \mathcal{A} \tag{8}$$

The best Q-value in any state, x, is given by:

$$Q^*(x, a) = \frac{min}{\pi} Q^{\pi}(x, a), \quad \forall x \in \chi, \ \forall a \in \mathcal{A}. \tag{9}$$

In every episode, the agent gets into states x_k, with an action space, $\mathcal{A}_k$, and using some strategy, chooses an action a_k and as a result, transits to the next state x_{k+1}, receiving an immediate reward, $r = g(x_k, a_k, x_{k+1})$. Then the Q-values are updated in the Bellman's recursive temporal difference fashion as shown below [25].

$$Q^{n+1}(x, a) \leftarrow Q^n(x, a) + \alpha [g(x_k, a_k, x_{k+1}) + \gamma max_{a_{k+1}} Q^n(x_{k+1}, a_{k+1}) - Q^n(x, a)] \tag{10}$$

where $\alpha \in (0, 1)$ is the learning rate determining the extent of modification of Q-values, $Q^n(x, a)$ is the current Q-value, $Q^{n+1}(x, a)$ is the next Q-value while $\gamma \in (0, 1)$ is the discount factor. The Q-learning algorithm is described in [10].

As shown in (10) above, although the greedy policy recommends the action that maximizes the action-value function for the next state, the agent does not have to take that action. As such, Q-learning is said to be an off-policy method of learning since the agent is not bound by the policy in acting [25], [63].

Q-learning is a very common algorithm in grid-tied microgrid power management. Kuznetsova *et al.* [45] implemented a two-step ahead Q-learning algorithm with a deterministic exploration method to schedule energy storage in a GT-MG with wind-powered DG. The study in [47] developed an asynchronous Q-learning technique to manage power in a grid-tied PV/battery EV charging station. The study showed that the asynchronous Q-learning produced a power schedule with about 14% lower global cost and a showed a more stable learning behaviour than the conventional Q-learning method. Leo *et al.* [69] designed a 3-step ahead learning to schedule energy in a BSS for a grid-tied PV/battery system. The authors reported an improvement in the utilization of the on-site generated PV power and the BSS. Foruzan *et al.* [70] developed a multi-agent scheme to manage energy trading between customers and energy suppliers including the utility grid, diesel and wind generators. Each entity trading with the grid was modelled as an agent, with each agent learning to improve on defined performance parameters. The algorithm converged to a policy that reduced energy interchange with the grid by 14%. Other applications of Q-learning in energy scheduling in microgrids may be found in [65], [71] and [72].

Q-learning methods use a Q-table to track the learning process. As state-action pairs increase, the Q-table size also increases, thus, the process suffers from the curse of dimensionality just like dynamic programming methods [23], [73]. Therefore, conventional Q-learning algorithms necessitate large discretization steps which make the results suboptimal. Also, Q-learning cannot handle stochastic policies since the Q-function is deterministically computed.

One of the methods of overcoming this curse of dimensionality is to use artificial neural networks (ANNs) to estimate the Q-function based on statistical regression. There are several function approximation methods such as decision trees and multivariable regression techniques, but ANNs are chosen for reinforcement learning due to the following reasons [23]:

1) They can deal with time-varying target functions.
2) They can learn effectively using data acquired by incremental means.

Therefore, instead of learning the action-value function $Q(x, a)$, the algorithm learns the parameterized function $Q(x, a, \theta)$ in a process called fitted Q-iteration. Thus, to get the optimal value of the Q function, the function approximator finds the parameter θ^* such that $Q(x, a, \theta^*)$ best estimates $Q^*(x, a)$ [74]. Then in every state, the action that maximizes the approximate value function returned by the ANN is selected during policy retrieval. This learning method is called fitted Q-iteration.

To arrive at the optimal value of the parameter θ^*, a training algorithm is needed as in the deep learning method. Many training algorithms have been developed to get the neural network to arrive at the optimal θ^*, namely, momentum, back propagation, Levenberg–Marquardt algorithm, etc., [75]. The most common of them is backpropagation that uses the gradient descent technique to arrive at the optimal weight vector. The algorithm calculates the gradient of a loss function with respect to each element in the weight vector θ^*. The objective of the algorithm is to minimize the loss starting from the output layer backward [75].

In Q-learning, the Q-value of a state-action pair is substituted by the old value plus some error in the estimation of the Q-value. The Q-value estimate is computed by adding the immediate reward to the maximum possible Q-value that may be found in the next state if the current policy were obeyed. This current estimate is called the ''target'' (T_k^Q) and the old value is called the ''prediction''. Thus, the target is calculated by (11).

$$T_k^{Q^n} = g(x_k, a_k, x_{k+1}) + \gamma max_{a_{k+1}} Q(x_{k+1}, a_{k+1}, \theta_n) \tag{11}$$

In ''stochastic gradient descent'', the ANN is trained to minimize a squared error (loss) [76] that is given by (12):

$$L(\theta) = \left(Q(x_k, a_k, \theta_n) - T_k^{Q^n}\right)^2 \tag{12}$$

Then the weights vector is adjusted using chain rule according to the stochastic gradient descent method [76], as given below:

$$\theta_{n+1} = \theta_n + \alpha \left(Q(x_k, a_k, \theta_n) - T_k^{Q^n} \right) \nabla_{\theta_n} Q(x_k, a_k, \theta_n) \tag{13}$$

The update is done iteratively so that the parameters are modified in the direction that minimizes the loss function until the parameter vector θ converges to θ^*.

However, as Sutton [74] observed, this method does not lead to proper convergence and diverge in some cases because the optimum policy is stochastic rather than deterministic as stipulated by (7). Therefore, a very small change in the value function for a particular action may well prevent it from being chosen. This occurs even in cases where the action itself is the optimal one for that state. Also, there are instabilities associated with Q-learning type recursion when it is applied to neural networks. The instabilities are caused by two issues:

1) Significant correlations within the state transitions, which is because the state transitions occur sequentially. Therefore, every state has some correlation with its predecessor and successor states [70].
2) Fitted Q-iteration does not employ a true gradient descent. Instead, the algorithm updates the weights of a neural network based on a loss with respect to a target which also depends on the very weights [77], [78].

Since policy is the logic of an agent, an ANN can be modeled as an agent to directly parameterize and approximate this policy through the policy gradient theorem [60], [78]. The policy gradient theorem is defined as:

$$\frac{\partial P}{\partial \theta} = \sum_s d^{\pi}(s) \sum_a \frac{\partial \pi(s,a)}{\partial \theta} Q^{\pi}(s,a), \tag{14}$$

where θ is the vector of the parameters of the current prevailing policy, P is the average reward obtained if that policy is obeyed in every step and $d^{\pi}(s)$ is the fixed distribution of states under the prevailing policy. Since $Q^{\pi}(s,a)$ is also unknown, it may also be estimated or better represented by an N-step return, i.e., the total expected discounted reward for N stages (say taking 5 future states). This is possible because the absolute value of $Q^{\pi}(s,a)$ is not needed but a numerical indication of whether there is progress in reaching the optimal policy or not [79]. Using (14), the ANN can be trained to adjust its parameters in the direction of better policy performance using the gradient ascent method. Thus, the parameter θ is updated as follows [78]:

$$\theta_{n+1} = \theta_n + \alpha \frac{\partial P}{\partial \theta_n}, \tag{15}$$

where α is the learning rate. This update increases the probabilities of selecting actions that give positive rewards and reduces the probabilities of actions that return negative rewards. Otherwise, if all the rewards are positive, then the policy gradient update leads to a faster increase in the probability of picking actions with higher rewards. It has been shown that learning a policy in this manner is easier and converges faster than learning a value function [74]. In the next subsection, deep reinforcement learning algorithms which apply either policy gradient theorem or advanced value-based techniques are discussed. They include batch reinforcement learning (BRL), actor-critic (AC) algorithms, deep Q-network (DQN), etc., [43], [57], [80].

D. BATCH REINFORCEMENT LEARNING

One of the major challenges experienced with traditional Q-learning is that it is data inefficient, thus it does not produce robust performance with stochastic policies [67]. Also, very complex environments with high stochasticity like BBM systems come with complex time series which may cause instability in training the ANNs [57]. To increase stability and data efficiency, batch reinforcement learning (BRL) is employed. BRL applies the Q-learning technique of recursive Q-value updates. However, unlike Q-learning that updates Q-values every time an action is taken, it sums up the experiences (or transitions) of the environment before updating the values [77], [81]. A single experience consists of the tuple (state, action, reward, next state). The agent approximates the optimal policy using a batch of its earlier experiences in a method called batch gradient descent. Thus, BRL is said to converge quicker than conventional fitted Q-iteration and Q-learning that discard observations after each Q-value update. The data flow diagram for a BRL algorithm is shown in Figure 3 [77].

Claessens *et al.,* [82] used a combination of BRL and ANN Q-function estimation to achieve up to 60% peak load reduction in a grid-tied PV/wind system, using the softmax function for exploration. Shi *et al.,* [83] used an echo state neural network (ESN) with BRL and the ϵ-greedy exploration strategy to obtain a 71% reduction in energy cost in a grid-connected PV/battery microgrid.

The other common approach is to randomly sample experiences instead of using an ANN directly [81]. This is called the ''experience replay'' technique [30]. In this method, every transition is saved in a replay buffer or memory as a tuple. The updates on the Q-values are done using a sample (mini-batch) from the replay buffer like in Monte Carlo learning. This is unlike TD-learning where updates are done using the most recent transition. This technique has been used by Mbuwir *et al.,* [27] for optimal battery scheduling for energy trading in a PV/battery/grid set-up. As will be seen in the following subsections, most modern deep reinforcement learning techniques apply experience replay method in updating the weights of the deep neural networks. Though BRL is data-efficient, its accuracy heavily relies on the experiences gathered. The algorithm cannot learn policies that have not appeared in its learning history. Also, since BRL algorithms use ANNs to estimate the Q function, they display instability in their learning process. The instability comes because the updates on the ANNs are not true gradient descent. The training of the ANNs is done by updating their parameters based on an error with respect to a target which itself depends

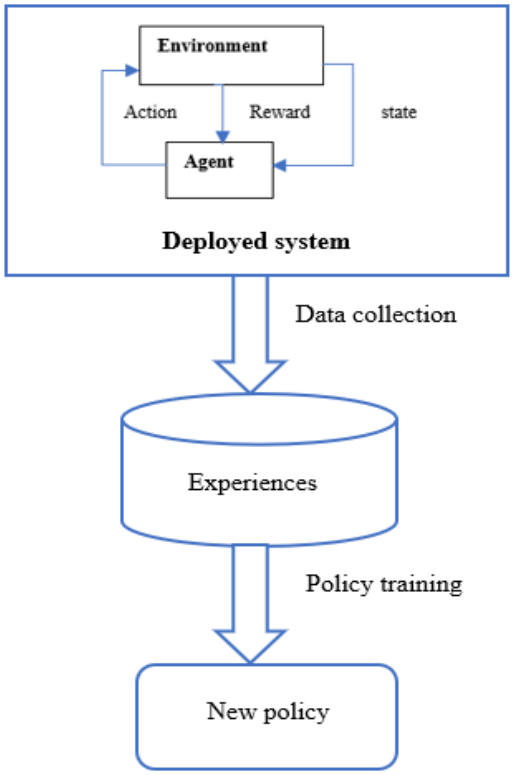

FIGURE 3. A data flow diagram for a batch reinforcement learning algorithm. A memory location is dedicated to storing transitions from which policy updates are done [77].

on the very parameters [77], [78]. It is like a dog chasing its tail.

E. DEEP Q-NETWORK

Deep Q-network is a deep reinforcement learning algorithm that combines techniques of both supervised learning and reinforcement learning [57]. In deep learning, ANNs with several hidden layers are used to approximate functions that represent data ensembles. Deep Q-network (DQN) is a framework of RL that incorporates the deep learning methods into Q-learning iterations using batch reinforcement learning techniques [57].

In the DQN, several experiences are gathered and saved in a replay buffer. A random sample of the experiences in the buffer is taken and used to update the deep neural network [84]. The DQN algorithm employs two neural networks, namely, the prediction network and the target network. The prediction network ($Q(s, a; \theta)$) estimates the current Q-value, while the target network ($Q(s, a; \theta^{-}$) hosts the old parameters used to estimate the next Q-value [73]. The current parameters (θ) are updated using a random sample of experiences (batch) from the replay buffer and after a set number of episodes. Then, after a given number of prediction steps, the parameters of the prediction network are copied into the target network as shown in Figure 4 below [85]. Since the experiences from the replay buffer are randomly sampled, their sequential occurrence during learning which

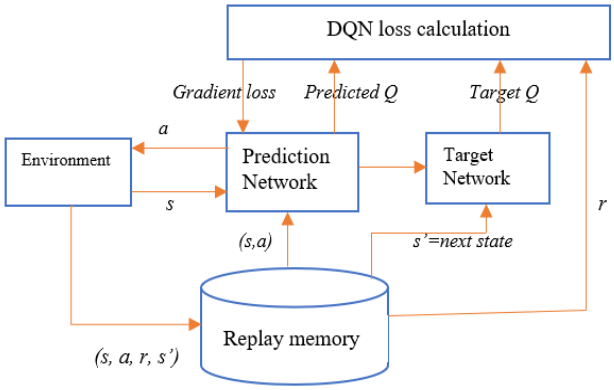

FIGURE 4. A data flow diagram for a DQN with a replay buffer and a target network [85].

makes them correlated is interrupted. As such, the experience replay technique breaks the correlation between sequentially observed experiences thus reducing oscillations or even possible divergence of the action values returned by the Q-network. Besides, the use of a separate target network from the prediction network helps to achieve stability as opposed to previous methods where the same network was used. More details including relevant equations on the DQN algorithm may be found in [57], [84].

Minh *et al.,* [86] established that DQN, apart from being more stable than normal policy gradient techniques, is more efficient even when used with many input-output nodes of neural networks. François-Lavet *et al.,* [60] implemented a DQN-based algorithm to schedule BSS and hydrogen storage for a microgrid. The authors used convolutional neural networks to learn a general policy for scheduling the storage under unpredictable demand and generation environment. Lu *et al.*, [29] used the DQN strategy for energy trading between a microgrid and a power plant and achieved a 22. 3% improvement in self-consumption of the MG generated power. Ji *et al.,* [48] describe the use of the DQN method to schedule microgrid energy generation and consumption with the inclusion of demand and generation prediction. The authors found that DQN returned a 20.75% reduction in energy cost compared to 13.12% obtained using a fitted Q-iteration.

As observed in the above studies, DQN cannot work with continuous action spaces. Besides, it still suffers from instability issues as dimensionality increases. This is because DQN merely assigns a score to every possible action in the action space. It then selects the action with the best score for execution. This is not possible if the action space is continuous or very large. Also, the algorithm employs the experience replay technique which is associated with memory inefficiency, takes a very long time to train and is limited to off-policy methods like Q-iterations [87]. There is also the problem of overestimation. This results from using the same neural network parameters for both estimating policy and evaluating the policy [88]. Bui *et al.,* [88] proposed a double-deep Q-network. In this approach, the action selection

from action evaluation is disassociated by using separate neural networks. The challenge with training time may be addressed using parallel learning and asynchronous techniques as reported by Nair *et al.,* [85] and applied in robotics in [89].

F. ACTOR-CRITIC ALGORITHMS

In the actor-critic (AC) method, two deep neural networks are involved. The first is the policy network (or the actor). The actor takes the environment states as inputs and gives a control action (or a policy). The second neural network is the value function estimator (or the critic). The critic observes the environment state and the reward obtained from the actor's control action and returns the estimated value of the action. The actor-network uses a gradient ascent method to maximize the objective function $\mathcal{P}(\theta)$ according to the policy gradient theorem in (14). The critic network uses gradient descent to minimize error in the value function estimation [90]. The actor takes the environment's state as input and produces a probability distribution (policy), $\pi(\omega)$, by which an action is selected. ω represents the weights of the actor network. The critic takes the same state and estimates the value function $V(k, \theta_n).\theta_n$ is the weight matrix for the value function network. The selected action is executed, the next state is generated from the environment and the reward $r(k)$ is computed. The next state is forwarded to the critic network to get the value of the next state $V(x_{k+1}, \theta_n)$. TD error $\partial(k)$ is then computed using equation (16) and used to update the critic network [90], [91].

$$\partial(k) = r + \gamma V(s_{k+1}, \theta_n) - V(s_k, \theta_n) \tag{16}$$

γ the discount factor. Figure 5 shows the data flow in an AC environment [92]. However, this TD learning approach comes with a high bias that may lead to the agent being trapped in a local optimum. This is common when the AC is applied in stochastic environments. Monte Carlo method, in which updates are done using sampled data from several episodes are a better alternative. But that will require that learning waits until the episodes are completed, and it is also memory intensive. Furthermore, pure Monte Carlo learning is prone to overfitting (or variance). An N-step return (where N is the number of transitions) may be used for updates to reduce variance. N transitions are performed under the policy estimated by the actor-network and the total discounted return is used to update the critic network [91], [93].

If the policy is deterministic, the AC method is called deep deterministic policy gradient (DDPG) [73]. Whether the policy is stochastic or deterministic, the architecture is the same except that in DDPG, the state-action-value (Q) is used to update the critic while in other actor-critic cases, the state-value (V) is used to update the critic network. AC methods have attracted significant applications in BBM schemes. Fuseli *et al.* [94] implemented an actor-critic algorithm using ANNs trained using particle swarm optimization (PSO) to obtain optimal schedules of energy resources in a smart home with better performance in terms of convergence than

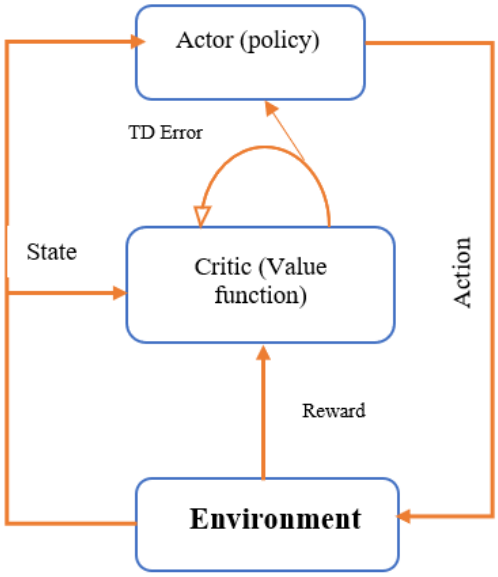

FIGURE 5. Illustration of actor-critic architecture. The actor-network represents the policy and is updated via gradient ascent to follow a policy that maximizes the total reward. The critic network represents the value function and is updated via gradient descent to minimize the value estimation error [90].

a general PSO. In [95], an AC algorithm is used to optimize power allocation in a heterogenous power network with wind turbines and PVs, leading to an increase in energy efficiency. Wan *et al.,* [96], schedule energy storage in a smart home leading to a significant energy cost reduction. DDPG is known to perform better with continuous action spaces than classical AC [97]. Chen *et al.* [98] used the DDPG technique for battery scheduling for energy trading in a grid-tied PV/battery microgrid returning a 55% profit increment compared to the system without the optimal scheduling. Odonkor and Lewis [99] developed a DDPG-based controller for shared energy storage devices for building clusters to obtain a reduction in the peak demand. Yu *et al.*, [100] designed and simulated a DDPG algorithm to control energy storage and heating, ventilation and air conditioning (HVAC) devices attaining a reduction in energy cost by 10%. However, DDPG does not work with stochastic policies.

The other challenge with such policy gradient-based algorithms is that they may not handle more than one action at a time. The actor may only return one action or the probability of taking one action at a time in a particular state. Mocanu *et al.* [101] developed a novel deep policy gradient (DPG) technique that was able to deal with more actions per state and converged faster than normal DDPG. Specifically, the authors used the DPG algorithm for both energy cost minimization in HVAC devices and peak reduction for residential application of the devices. It was observed that this algorithm performed better than DQN in achieving the peak reduction and energy optimization. A summary of key applications of reinforcement learning in microgrid technologies are given in Table 2.

TABLE 2. Summary of reviewed reinforcement learning algorithms.

Algorithm	Microgrid applications	Strengths	Weaknesses	Possible improvements
Q-learning [54, 56]	• Battery energy scheduling [45, 65, 69] • MG energy trading [70-72]	• Easy to implement • Versatile	• Suffers from the curse of dimensionality • Data inefficient • Learning value functions is difficult.	• Use of function approximators • Use of experience replay
BRL [77, 81]	• Battery energy scheduling [27] • Peak load shaving [82] • MG power dispatch [83]	• Better convergence than normal Q-learning. • Data efficient	• Memory intensive. • Unstable • Poor convergence.	• Use of target network • Priority sampling of experiences.
DQN [84-88]	• MG energy trading [29] • MG power dispatch [48] • Control of multi-level energy storage [60]	• Performs well in continuous state spaces where action space is discrete and small.	• Cannot handle continuous or large action spaces. • Prone to overestimation. • Takes long to train.	• Separation of prediction and target networks • Parallel learning method with different neural networks • Priority experience replay.
AC [90-93]	• Smart home energy control [94, 96] • Power dispatch for heterogeneous network [95]	• Faster convergence. • It is easier to learn a policy than a value function. • Performs well with both discrete and continuous environments.	• Sample inefficient • Hard to select and tune hyperparameters for the actor and the critic. • May have bias and be trapped in a local optimum	• Use of experience replay method. • Parallel learning • Adding entropy noise
DDPG [97]	• MG energy trading [98] • Control of shared energy storage [99] • HVAC control [100-101]	• Sample efficient • Performs well with continuous environments.	• Problems with convergence under discrete stochastic environments. • Deals with one action at a time	• Application of policy optimization techniques.

V. EMERGING REINFORCEMENT LEARNING TECHNIQUES AND RESEARCH DIRECTIONS

A. SYNHRONOUS AND ASYNCHRONOUS LEARNING

As previously noted, there are instabilities associated with the Q-learning type recursions when applied to neural networks. To address this, Google DeepMind Technologies developed the asynchronous advantage actor-critic (A3C) algorithm [87]. In this method, several neural network agents (worker networks) are trained with different copies of the environment (asynchronously). The worker networks then update a master agent after a given number of steps until the master converges to the optimal policy. The study showed that this parallelization of learning and asynchrony in A3C algorithm performed better than a DQN method in playing standard video games. In 2017, OpenAI (a USA-based artificial intelligence research company) demonstrated that, despite its complexity, asynchrony did not produce any specific advantage. Same or even better results could be achieved without the complexity of asynchrony. The company therefore released an easy to implement and an efficient method called the synchronous advantage actor-critic (A2C) technique [102]. The A2C uses the N-step return technique that ensures a better bias-overfitting balance than conventional AC and Monte Carlo learning. There was no paper found in current literature at the time of this study on the use of the A2C algorithm in solving the power management problem in BBM schemes. This could be a possible future research direction.

B. MULTI-AGENT REINFORCEMENT LEARNING

Conventionally, reinforcement learning techniques involves the modelling of a single agent to learn an optimal control policy. A multi-agent reinforcement learning algorithm (MARLA) has more than one learning agent interacting with the environment, i.e., both the environment's state occurrences an the emergence of the reward signals are occasioned by the combined actions of all the agents [103]. Although the MARLA concept is not new, the recent advancements in single-agent systems have brought them into greater focus. In the recent times, microgrid power management systems are becoming more decentralized. Thus, single-agent algorithms are becoming less popular. Particularly, MARLA has been applied with visible success in demand response, real-time demand response and management of distributed energy storage systems [104]–[106]. The multi-agency technology is such an attractive research area that can be implemented for microgrid power management schemes. These include interconnected microgrids, peer-to-peer energy trading and optimal grid-tied microgrid power scheduling to enhance system healing in the post-disaster or post-cyber-physical attach scenarios.

C. TRANSFER LEARNING AND ITS APPLICATION IN REINFORCEMENT LEARNING

Transfer learning is the process of using the knowledge acquired by a model in performing a task in solving another problem that is not exactly related to the previous task. Conventionally, this technique was used in deep learning models where historical data is limited [107]. The study in [108] used transfer learning technique to forecast a PV generator output profile. Also, transfer learning has been applied to fasten learning in classical reinforcement learning algorithms [109].

Transfer learning would significantly improve the modern deep reinforcement learning (DRL) based microgrid power management algorithms. This is because most of the DRL techniques have been more successful in playing computer games than in solving most power systems problems. The knowledge acquired by a DRL agent in gaming may easily be transferred to microgrid power management environments. However, to apply this technology, the tasks being dealt with must have some reasonable similarity. In [110] the authors explored the application of transfer learning in a DQN algorithm and demonstrated that task similarity increases the probability of success in transferring learned policy.

However, the application of this technique in grid-tied microgrid power management is still open to research.

D. PRIORITY EXPERIENCE REPLAY

The advantages of experience replay in enhancing stability in BRL and DQN algorithms has been explored in [70], [84]. The other advantage of experience replay is that it allows for retrieval of highly beneficial experiences to increase the speed of learning. The biased sampling of the past experiences of an RL agent to achieve a given learning objective is called ''priority experience replay''. Authors in [111] applied a priority experience replay method to improve the performance of a DQN. The results of the study showed that the DQN with priority experience replay performed better than a DDPG in terms of both speed and convergence.

Also, some experiences are very rare to come by, thus, it is important to have a buffer for such experiences for future reuse. Prioritized sampling of experiences may be applicable in scheduling microgrid power in post-disaster response scenarios. The experiences of an agent when a disaster strikes may be saved and replayed to the agent to improve its reaction when the disaster strikes again. Furthermore, actor-critic algorithms do not make use of this significant technique. In the development of the A2C algorithm, experience replay was introduced to AC algorithm with significant success [102]. The combination of the stability enhancement by experience replay and the robustness of the actor-critic methods significantly improved the global convergence and the speed of the A2C method. However, this research is still open to exploration.

E. EXTRINSIC AND INTRINSIC MOTIVATION IN LEARNING

Classical RL techniques reward an agent for performing correct transitions toward the goal of the learning process [23]. Such rewards are called immediate rewards. They elicit extrinsic motivation, i.e., a motivation to earn rewards or avoid punishment external to the agent. The rewards are developed from the dynamics of the environment by means of hard coding through reward functions. Thus, they are not scalable. Furthermore, in some environments, the impact of actions may not be immediately clear. In more complex environments like modern power networks, reward functions generated by the agent itself independent of the state transitions in the environment may be more appropriate [92]. Such a reward scheme generates an intrinsic motivation, i.e., a motivation to achieve goals without external signals. There are two learning methods that produce intrinsic motivation in the agent, namely, curiosity-driven learning and hindsight experience replay [92].

1) CURIOSITY-DRIVEN LEARNING

Curiosity is defined as the error in the agent's prediction of the state transition or the consequences of the agent's actions [112]. Such a reward is defined to motivate the agent to reduce the uncertainty in the prediction of its actions. The upshot is that the uncertainty is higher in the parts of the environment that has not been visited by the environment. Actions that lead to predictable transitions get higher intrinsic rewards [113]. Although this concept is becoming popular in robotics and computer gaming environments, it has not been implemented in power management algorithms.

2) HINDSIGHT EXPERIENCE REPLAY

Hindsight experience replay (HER) is an improvement of experience replay used with off-policy RL algorithms. In this technique, the reward function is a function of the current state, the action (derived from the current policy) and a goal state (instead of the next state) [114]. The reward is detached from the state transitions in the environment. Thus, the agent learns beneficial policies from both bad and good transitions. The learning mechanism is built on the fact that even bad policies can produce experiences that are beneficial to the learning process [114].

F. POLICY OPTIMIZATION METHODS

Normally, an AC algorithm has a policy that changes according to the gradient descent-based updates. However, defining the step size of these updates is a major challenge. Large steps poses the risk of going too far in the wrong direction that the agent may not correct regardless of the amount of experiences gathered [115]. This is more detrimental if there is a high probability of the agent gathering misleading experiences (bad data). Policy optimization methods are advancements of the AC algorithms that have been developed to overcome this challenge. Two optimization methods have been developed,

namely, trust-region policy optimization (TRPO) and proximal policy optimization (PPO).

1) TRUST-REGION POLICY OPTIMIZATION

In TRPO, a ''surrogate'' objective is optimized with respect to a boundary in the step size of the updates. Thus, the updates are limited to a ''trust region'' to avoid accumulating misleading experiences [116]. The surrogate objective is linear while the approximation to the constraint on the policy update step sizes is defined using a quadratic function. A recent study implemented a multi-agent TRPO algorithm to perform real-time dynamic demand response [105]. There is still need for further studies on this algorithm in microgrid power management.

2) PROXIMAL POLICY OPTIMIZATION

Proximal policy optimization (PPO) is a simpler version of TRPO that linearizes both the surrogate objective and the step sizes approximation. PPO has been shown to perform better than most algorithms with the actor-critic architecture in solving multi-dimensional continuous environments [116]. The linearization of both the surrogate objective function and the step sizes makes PPO simpler and more robust. It also makes the algorithm easier to tune.

Since most of the smart grid power management problems have such characteristics of multidimensionality and continuity of the state space, TRPO and PPO techniques will be of significant application in microgrid and the modern utility grids.

VI. DISCUSSIONS

Reinforcement learning is a powerful tool that can be used for power systems scheduling in highly stochastic environments such as grid-tied RES with BBM architecture. RL has been used for battery energy scheduling, MG power dispatch, control of HVAC, energy management in smart homes, management of multi-level energy storage, control of shared energy storage devices in building clusters, etc. It has been observed that the efficiency of the RL-based method depends on the way the problem is formulated. There is a need to develop a formulation that produces better optimality with considerable efficiency.

One major challenge in RL application is the development of an MDP that best represents the optimization environment. In the development of deterministic MDPs, the stochasticity of the environments being modelled are ignored. Such assumptions have been found to adversely affect the accuracy of the solutions obtained. Furthermore, deterministic MDPs occupy a large memory space. Moreover, in the case of PO-MDPs, it is difficult to provide adequate observations that can permit the learning agent to estimate the states correctly. Current methods incorporate aspects of supervised learning in which simulated environments are used to train ANNs that make a better approximation of states than simple Markov Chains and occupy less memory space than deterministic MDPs.

The reviewed literature has revealed that traditional challenges experience with RL algorithms such as the curse of dimensionality and exploration-exploitation conundrum have been competently addressed. It has been noted that the methods used to address these challenges introduce new issues such as instability in the DQN and pure policy gradient techniques. Although experience replay has been applied to improve stability in value-based methods, it introduces variance in policy-based techniques. The reviewed literature has pointed out that the separation of the policy and the value function networks produces better results. This separation is the major reason for the success of AC algorithms. The fact that AC architecture hybridizes policy gradients with value-based methods is a major reason why it is more robust in microgrid power management. It has been noted that modern RL algorithms take the form of this architecture in one way or the other.

Recently, some grid-tied microgrids are equipped with new technologies such distributed power dispatch, decentralized energy storage system, intelligent real-time load scheduling and demand response, interconnection of microgrids and peer-to-peer power sharing. Such technologies call for more advanced power management algorithms. There have been improvements to the reinforcement learning techniques to adapt to the new challenges in the area of microgrid power management. Multi-agent RL is seen as a one of the most powerful RL technologies for power management in a distributed dispatch scheme and interconnected microgrids. Also, increased dimensionality in networked microgrids can be handled by policy optimization methods such as TRPO and PPO techniques due to their ability to seamlessly optimize objectives more efficiently in highly uncertain, continuous and multi-dimensional environments. It has been observed that the policy optimization algorithms are also easier to implement and tune. Moreover, the application of transfer learning in DRL methods may help to effectively bring the success of DRL methods in gaming to modern power management environments. Furthermore, new emerging issues such as smart grid vulnerability to cyber-physical attacks and natural disasters could be addressed using priority experience replay. Replaying previous high impact low probability events to an agent may better adapt it to react to them more appropriately when they recur. Also, transfer learning may be implemented to maximize the performance of the agent in cases where the task has high similarity with one that has been solved by the agent before. Additionally, intrinsic motivation methods of learning such as curiosity driven learning and hindsight experience replay have the potential of reducing the complexity of reward engineering in complex systems. This is because these methods detach environment's transitions from the reinforcement function. The main objective of introducing intrinsic motivation in RL is to improve scalability of the algorithms by detaching the environment's dynamics from the reward function. While this is desirable, it is still important to take advantage of the existing knowledge of the system dynamics to improve the performance of the

algorithms. Therefore, hybridization of extrinsic and intrinsic motivation techniques may produce more stable learning in complex power systems environments. Such a hybrid reward scheme may benefit from both the designer's knowledge of the environment's dynamics and the agent's own experiences during the learning process. Table 3 shows a summary of the emerging reinforcement learning technologies and their possible applications in microgrid power management.

TABLE 3. Summary of emerging reinforcement learning technologies and their possible applications in microgrid power management.

RL Technique	Possible applications
A3C and A2C [87], [102]	• Battery scheduling and microgrid energy dispatch
Multi-agent RL [103]-[106]	• General smart grid power scheduling. • Distributed power dispatch • Intelligent demand response
Transfer learning [107]-[110]	• Cyber-physical attach detection and mitigation. • Improvement of forecasting models for real-time power management applications.
Priority Experience Replay [111]	• Post-disaster microgrid power management.
TRPO $ PPO [105], [115], [116]	• General microgrid power scheduling. • Peer-to-peer energy trading
Intrinsic Motivation [112]-[114]	• Networked microgrids, distributed power dispatch

VII. CONCLUSION

In this paper, RL approaches applied to the scheduling of power and energy in grid-tied microgrids has been reviewed. The use of RL has been found to optimize energy in smart homes, heterogenous power networks and to maximize profit in peer-to-peer energy trading schemes. It has been observed that the drawback of RL techniques such as the curse of dimensionality and the exploration-exploitation dilemma has significantly been overcome using the modern techniques that combine RL and supervised learning. The current methods such as AC and DDPG can be used online because of their high efficiency and speed. However, modern deep RL techniques still experience challenges such as data inefficiency and instability. The application of BRL, experience replay and target neural networks have been proposed in various literature to address such difficulties with visible success.

Nonetheless, there are still outstanding issues in the application of RL in control of power systems that require more research attention. First, practicality and scalability are major challenges especially when RL is applied in dynamic settings. Practicality of RL in power systems control is severely affected by the switching speed of the power conversion equipment. Furthermore, most RL algorithms are developed and tested using computer games, thus, adapting and scaling them to the large power systems environment is still a challenge. Finally, the difficulty in the tuning of RL algorithmic hyperparameters affects reproducibility, reusability and versatility of the developed algorithms. The issue of complex balancing and tuning of algorithmic hyperparameters is a significant limiting factor in the design of the RL algorithms. Hand-coded tuning of intelligent algorithms may limit the extent of learning the algorithms could achieve. If the reinforcement learning algorithms are to be trained to perform more complex tasks such as power management in modern and future power systems, it would be important that the tuning of their core hyperparameters be learnt by the algorithms too. Also, conventional ways of reward engineering assume the human designer understands the system dynamics. However, that is not the case with complex systems such as modern and future power systems. In order to scale the current RL algorithms to handle the complexity of power systems, intrinsic reward mechanisms such as curiosity and hindsight experience replay will be instrumental. It is this authors' view that such methods could be hybridized with the current reward design techniques to produce better algorithmic learning outcomes in the future.

REFERENCES

[1] W. E. Forum, *Electric Vehicles for Smarter Cities: The Future of Energy and Mobility*. Geneva Canton, Switzerland: World Econ. Forum, Jan. 2018.
[2] *2018 World Energy Outlook Report*, International Energy Agency, Paris, France, 2018.
[3] J. Larminie and J. Lowry, *Electric Vehicle Technology Explained*. Oxford, U.K.: Wiley, 2012.
[4] A. R. Bhatti, Z. Salam, M. J. B. A. Aziz, and K. P. Yee, "A comprehensive overview of electric vehicle charging using renewable energy," *Int. J. Power Electron. Drive Syst.*, vol. 7, no. 1, pp. 114–123, 2016.
[5] C. Youssef, E. Fatima, E. S. Najia, and A. Chakib, "A technological review on electric vehicle DC charging stations using photovoltaic sources," *IOP Conf. Series, Mater. Sci. Eng.*, vol. 353, no. 1, 2018, Art. no. 012014.
[6] U. Abronzini, C. Attaianese, M. D'Arpino, M. Di Monaco, A. Genovese, G. Pede, and G. Tomasso, "Optimal energy control for smart charging infrastructures with ESS and REG," in *Proc. Int. Conf. Electr. Syst. Aircr., Railway, Ship Propuls. Road Vehicles Int. Transp. Electrific. Conf.*, Nov. 2016, pp. 1–6.
[7] M. O. Badawy and Y. Sozer, "Power flow management of a grid tied PV-battery system for electric vehicles charging," *IEEE Trans. Ind. Appl.*, vol. 53, no. 2, pp. 1347–1357, Apr. 2017.
[8] S. Parhizi, H. Lotfi, A. Khodaei, and S. Bahramirad, "State of the art in research on microgrids: A review," *IEEE Access*, vol. 3, pp. 890–925, 2015.
[9] T. Khalili, S. Nojavan, and K. Zare, "Optimal performance of microgrid in the presence of demand response exchange: A stochastic multi-objective model," *Comput. Electr. Eng.*, vol. 74, pp. 429–450, Mar. 2019.
[10] E. A. Jasmin, "Reinforcement learning approaches to power system scheduling," Ph.D. dissertation, School Eng., Cochin Univ. Technol., Kochi, Kerala, 2008.
[11] D. E. Olivares, A. Mehrizi-Sani, A. H. Etemadi, C. A. Cañizares, R. Iravani, M. Kazerani, A. H. Hajimiragha, O. Gomis-Bellmunt, M. Saeedifard, R. Palma-Behnke, G. A. Jiménez-Estévez, and N. D. Hatziargyriou, "Trends in microgrid control," *IEEE Trans. Smart Grid*, vol. 5, no. 4, pp. 1905–1919, Jul. 2014.
[12] B. Yu, J. Guo, C. Zhou, Z. Gan, J. Yu, and F. Lu, "A review on microgrid technology with distributed energy," in *Proc. Int. Conf. Smart Grid Electr. Autom. (ICSGEA)*, May 2017, pp. 143–146.

[13] A. Jafari, T. Khalili, H. G. Ganjehlou, and A. Bidram, "Optimal integration of renewable energy sources, diesel generators, and demand response program from pollution, financial, and reliability viewpoints: A multi-objective approach," *J. Cleaner Prod.*, vol. 247, Feb. 2020, Art. no. 119100.

[14] N. Bizon, N. Mahdavi Tabatabaei, and H. Shayeghi, *Analysis, Control and Optimal Operations in Hybrid Power Systems*. London, U.K.: Springer, 2014.

[15] Y. Riffonneau, S. Bacha, F. Barruel, and S. Ploix, "Optimal power flow management for grid connected PV systems with batteries," *IEEE Trans. Sustain. Energy*, vol. 2, no. 3, pp. 309–320, Jul. 2011.

[16] L. Ngoc An and T. Quoc-Tuan, "Optimal energy management for grid connected microgrid by using dynamic programming method," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Jul. 2015, pp. 1–5.

[17] D. Zhang, "Optimal design and planning of energy microgrids," Ph.D. dissertation, Dept. Chem. Eng., Univ. College London, London, U.K., 2013.

[18] M. Hijjo, F. Felgner, and G. Frey, "PV-Battery-Diesel microgrid layout design based on stochastic optimization," in *Proc. 6th Int. Conf. Clean Electr. Power (ICCEP)*, Jun. 2017, pp. 30–35.

[19] U. B. Tayab, F. Yang, M. El-Hendawi, and J. Lu, "Energy management system for a grid-connected microgrid with photovoltaic and battery energy storage system," in *Proc. Austral. New Zealand Control Conf. (ANZCC)*, Dec. 2018, pp. 141–144.

[20] A. T. Eseye, D. Zheng, J. Zhang, and D. Wei, "Optimal energy management strategy for an isolated industrial microgrid using a modified particle swarm optimization," in *Proc. IEEE Int. Conf. Power Renew. Energy (ICPRE)*, Oct. 2016, pp. 494–498.

[21] Ma. L. Puterman, *Markov Decision Processes*. British CO, USA: Wiley, 2005.

[22] A. O. Erick and K. A. Folly, "Reinforcement learning approaches to power management in grid-tied microgrids?: A review," in *Proc. Clemson Univ. Power Syst. Conf.*, 2020, pp. 1–6.

[23] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[24] R. S. Sutton, "On the significance of Markov decision processes," in *Artificial Neural Networks—ICANN'97* (Lecture Notes in Computer Science), vol. 1327, W. Gerstner, A. Germond, M. Hasler, and J. D. Nicoud, Eds. Berlin, Germany: Springer, 1997, doi: 10.1007/BFb0020167.

[25] C. J. Watkins and P. Dayan, "Technical note: Q-learning," *Mach. Learn.*, vol. 8, no. 3–4, pp. 279–292, May 1992.

[26] D. Dewey, "Reinforcement learning and the reward engineering principle rewards in an uncertain world," in *Proc. AAAI*, 2014, pp. 1–8.

[27] B. V. Mbuwir, F. Ruelens, F. Spiessens, and G. Deconinck, "Battery energy management in a microgrid using batch reinforcement learning," *Energies*, vol. 10, no. 11, pp. 1–19, 2017.

[28] A. O. Erick and K. A. Folly, "Energy trading in grid-connected PV-battery electric vehicle charging station," in *Proc. Int. SAUPEC/RobMech/PRASA Conf.*, Jan. 2020, pp. 1–6.

[29] X. Lu, X. Xiao, L. Xiao, C. Dai, M. Peng, and H. V. Poor, "Reinforcement learning-based microgrid energy trading with a reduced power plant schedule," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10728–10737, Dec. 2019.

[30] J. R. Vázquez-Canteli and Z. Nagy, "Reinforcement learning for demand response: A review of algorithms and modeling techniques," *Appl. Energy*, vol. 235, pp. 1072–1089, Feb. 2019.

[31] K. Mason and S. Grijalva, "A review of reinforcement learning for autonomous building energy management," *Comput. Electr. Eng.*, vol. 78, pp. 300–312, Sep. 2019.

[32] D. Zhang, C. Electric Power Research Institute, X. Han, C. Deng, T. University of Technology, and C. Electric Power Research Institute, "Review on the research and practice of deep learning and reinforcement learning in smart grids," *CSEE J. Power Energy Syst.*, vol. 4, no. 3, pp. 362–370, Sep. 2018.

[33] S. Bai, D. Yu, and S. Lukic, "Optimum design of an EV/PHEV charging station with DC bus and storage system," in *Proc. IEEE Energy Convers. Congr. Expo.*, Sep. 2010, pp. 1178–1184.

[34] M. Yilmaz and P. T. Krein, "Review of battery charger topologies, charging power levels, and infrastructure for plug-in electric and hybrid vehicles," *IEEE Trans. Power Electron.*, vol. 28, no. 5, pp. 2151–2169, May 2013.

[35] M. Yilmaz and P. Krein, "Review of Charging Power Levels and Infrastructure for Plug-In Electric and Hybrid Vehicles and Commentary on Unidirectional Charging," *IEEE Trans. Power Electron.*, vol. 28, no. 5, pp. 2151–2169, 2012.

[36] Q.-T. Tran, N. A. Luu, and T. L. Nguyen, "Optimal energy management strategies of microgrids," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2016, pp. 1–6.

[37] A. Hoke, A. Brissette, K. Smith, A. Pratt, and D. Maksimovic, "Accounting for lithium-ion battery degradation in electric vehicle charging optimization," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 2, no. 3, pp. 691–700, Sep. 2014.

[38] B. Xu, "Degradation-limiting optimization of battery energy storage systems operation," M.S. thesis, Power Syst. Lab. Fed. Inst., Technol. Zurich, Dättwill, Switzerland, 2013.

[39] C. Ju, P. Wang, L. Goel, and Y. Xu, "A two-layer energy management system for microgrids with hybrid energy storage considering degradation costs," *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 6047–6057, Nov. 2018.

[40] U. Abronzini, C. Attaianese, M. D'Arpino, M. Di Monaco, A. Genovese, G. Pede, and G. Tomasso, "Multi-source power converter system for EV charging station with integrated ESS," in *Proc. IEEE 1st Int. Forum Res. Technol. Soc. Ind. Leveraging*, Sep. 2015, pp. 427–432.

[41] U. Abronzini, C. Attaianese, M. D'Arpino, M. D. Monaco, and G. Tomasso, "Power converters for PV systems with energy storage: Optimal power flow control for EV's charging infrastructures," in *Proc. Int. Exhib. Conf. Power Electron., Intell. Motion, Renew. Energy Manage.*, May 2016, pp. 1–7.

[42] B. Jeddi, Y. Mishra, and G. Ledwich, "Dynamic programming based home energy management unit incorporating PVs and batteries," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Jul. 2017, pp. 1–5.

[43] M. O. Badaway, "Grid tied PV/battery system architecture and power management for fast EV charging," Ph.D Thesis, Dept. Elect. Eng., The University of Akron, Akron, OH, USA, 2016.

[44] B. V. Mbuwir, F. Spiessens, and G. Deconinck, "Self-learning agent for battery energy management in a residential microgrid," in *Proc. IEEE PES Innov. Smart Grid Technol. Conf. Eur. (ISGT-Europe)*, Oct. 2018, pp. 1–6.

[45] E. Kuznetsova, Y.-F. Li, C. Ruiz, E. Zio, G. Ault, and K. Bell, "Reinforcement learning for microgrid energy management," *Energy*, vol. 59, pp. 133–146, Sep. 2013.

[46] J. E. A., I. A. T. P., and J. R. V. P., "Reinforcement learning solution for unit commitment problem through pursuit method," in *Proc. Int. Conf. Adv. Comput., Control, Telecommun. Technol.*, Dec. 2009, pp. 324–327.

[47] A. O. Erick and K. A. Folly, "Power flow management in electric vehicles charging station using reinforcement learning," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2020, pp. 1–8.

[48] Y. Ji, J. Wang, J. Xu, X. Fang, and H. Zhang, "Real-time energy management of a microgrid using deep reinforcement learning," *Energies*, vol. 12, no. 12, p. 2291, Jun. 2019.

[49] A. W. Moore, "Markov systems, Markov decision processes, and dynamic programming," *Science*, vol. 50, no. 20, pp. 1–18, 2004.

[50] J. Song, V. Krishnamurthy, A. Kwasinski, and R. Molina, "Analysis of the energy storage operation of electrical vehicles with a photovoltaic roof using a Markov chain model," in *Proc. IEEE Veh. Power Propuls. Conf.*, 2012, pp. 820–825.

[51] S. Kumar and R. Y. Udaykumar, "Markov chain based stochastic model of electric vehicle parking lot occupancy in vehicle-to-grid," in *Proc. IEEE Int. Conf. Comput. Intell. Comput. Res. (ICCIC)*, Dec. 2015, pp. 1–6.

[52] T. AlSkaif, W. Schram, G. Litjens, and W. van Sark, "Smart charging of community storage units using Markov chains," in *Proc. IEEE PES Innov. Smart Grid Technol. Conf. Eur. (ISGT-Europe)*, Sep. 2017, pp. 1–6.

[53] M. M. Kokar and S. A. Reveliotis, "Reinforcement learning: Architectures and algorithms," *Int. J. Intell. Syst.*, vol. 8, no. 8, pp. 875–894, 1993.

[54] R. Bellman, *The Theory of Dynamic Programming*. Santa Monica, CA, USA: The Rand Corporation, pp. 1–550, 1954.

[55] Y. Wan, M. Zaheer, M. White, and R. S. Sutton, "Model-based reinforcement learning with non-linear expectation models and stochastic environments," in *Proc. FAIM Workshop Predict. Generative Model. Reinforcement Learn.*, 2018, pp. 1–5.

[56] A. Potapov and M. K. Ali, "Convergence of reinforcement learning algorithms and acceleration of learning," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 67, no. 2, Feb. 2003, Art. no. 267061.

[57] V. François-Lavet, "Contributions to deep reinforcement learning and its applications in smartgrids," Ph.D Thesis, Dept. Elect. Eng. Comput. Sci., Univ. of Liege, Liège, Belgium, 2017.

[58] X. Yang, L. Zhu, Z. Zhang, L. Li, and H. Wang, "Electric vehicles charging and discharging control strategy based on independent DC microgrid," in *Proc. IEEE 3rd Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Oct. 2018, pp. 969–973.

[59] G. Kumar Venayagamoorthy, R. K. Sharma, P. K. Gautam, and A. Ahmadi, "Dynamic energy management system for a smart microgrid," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 8, pp. 1643–1656, Aug. 2016.
[60] V. François-lavet, D. Taralla, E. Damien, and R. Fonteneau, "Deep Reinforcement Learning Solutions for Energy Microgrids Management," in *Eur. Workshop Reinforcement Learn.*, vol. 2016, pp. 1–7.
[61] R. S. Sutton, *Reinforcement Learning: Past, Present and Future*. Florham Park, CA, USA: AT&T Labs, 1999.
[62] R. H. M. Zargar and M. H. Yaghmaee Moghaddam, "Development of a Markov-Chain-Based solar generation model for smart microgrid energy management system," *IEEE Trans. Sustain. Energy*, vol. 11, no. 2, pp. 736–745, Apr. 2020.
[63] C. J. C. H. Watkins, "Learning from Delayed Rewards," Ph.D. dissertation, Dept. Comput. Sci., Univ. Cambridge, Cambridge, U.K., 1989.
[64] M. A. L. Thathachar and P. S. Sastry, *Networks of Learning Automata: Techniques for Online Stochastic Optimization*. New York, NY, USA: Springer, 2004.
[65] Y. Xi, L. Chang, M. Mao, P. Jin, N. Hatziargyriou, and H. Xu, "Q-learning algorithm based multi-agent coordinated control method for microgrids," in *Proc. 9th Int. Conf. Power Electron. ECCE Asia (ICPE-ECCE Asia)*, Jun. 2015, pp. 1497–1504.
[66] M. Tokic, "Adaptive ϵ-greedy exploration in reinforcement learning based on value differences," in *KI 2010: Advances in Artificial Intelligence* (Lecture Notes in Computer Science), vol. 6359, R. Dillmann, J. Beyerer, U. D. Hanebeck, and T. Schultz, Eds. Berlin, Germany: Springer, 2010, doi: 10.1007/978-3-642-16111-7_23.
[67] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, no. 1, pp. 237–285, Jan. 1996.
[68] R. M. Hamid, "Gradient temporal-difference learning algorithms," Ph.D. dissertation, Dept. Comput. Sci., Univ. Alberta, Edmonton, AB, Canada, 2011.
[69] R. Leo, R. S. Milton, and S. Sibi, "Reinforcement learning for optimal energy management of a solar microgrid," in *Proc. IEEE Global Hum. Technol. Conf.*, Sep. 2014, pp. 183–188.
[70] E. Foruzan, L.-K. Soh, and S. Asgarpoor, "Reinforcement learning approach for optimal distributed energy management in a microgrid," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 5749–5758, Sep. 2018.
[71] S. Kim and H. Lim, "Reinforcement learning based energy management algorithm for smart energy buildings," *Energies*, vol. 11, no. 8, p. 2010, Aug. 2018.
[72] P. Kofinas, G. Vouros, and A. I. Dounis, "Energy management in solar microgrid via reinforcement learning using fuzzy reward," *Adv. Building Energy Res.*, vol. 12, no. 1, pp. 97–115, Jan. 2018.
[73] J. Fan, Z. Wang, Y. Xie, and Z. Yang, "A theoretical analysis of deep Q-learning," in *Proc. 2nd Conf. Learn. Dyn. Control*, vol. 120. PMLR, 2020, pp. 486–489.
[74] R. S. Sutton, S. Singh, and D. McAllester, "Comparing policy-gradient algorithms," AT&T Shannon Lab., Florham Park, CA, USA, Tech. Rep. NJ 07932, 2001.
[75] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019.
[76] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," *Found. Trends Mach. Learn.*, vol. 11, nos. 3–4, pp. 219–354, 2018, doi: 10.1561/2200000071.
[77] S. Lange, T. Gabel, and M. Riedmiller, "Batch reinforcement learning," *Adapt. Learn. Optim.*, vol. 12, pp. 45–73, 2012.
[78] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," *JMLR*, vol. 32, pp. 605–619, Dec. 2014.
[79] R. Lincoln, S. Galloway, B. Stephen, and G. Burt, "Comparing policy gradient and value function based reinforcement learning methods in simulated electrical power trade," *IEEE Trans. Power Syst.*, vol. 27, no. 1, pp. 373–380, Feb. 2012.
[80] A. Barto, P. Thomas, and R. Sutton, "Some recent applications of reinforcement learning," in *Proc. Workshop Adapt. Learn. Syst.*, 2017, pp. 1–5.
[81] M. R. S. Lange and T. Gabel, "S. Lange, T. Gabel, M. Riedmiller, "Batch reinforcement learning," in *Reinforcement Learning. Adaptation, Learning, and Optimization*, vol. 12, M. Wiering, M. van Otterlo, Eds. Berlin, Germany: Springer, 2012.
[82] B. J. Claessens, S. Vandael, F. Ruelens, K. De Craemer, and B. Beusen, "Peak shaving of a heterogeneous cluster of residential flexibility carriers using reinforcement learning," in *Proc. IEEE PES ISGT Eur.*, Oct. 2013, pp. 1–5.
[83] G. Shi, D. Liu, and Q. Wei, "Echo state network-based Q-learning method for optimal battery control of offices combined with renewable energy," *IET Control Theory Appl.*, vol. 11, no. 7, pp. 915–922, Apr. 2017.
[84] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.
[85] A. Nair, P. Srinivasan, S. Blackwell, C. Alcicek, R. Fearon, A. De Maria, V. Panneershelvam, M. Suleyman, C. Beattie, S. Petersen, S. Legg, V. Mnih, K. Kavukcuoglu, and D. Silver, "Massively parallel methods for deep reinforcement learning," 2015, *arXiv:1507.04296*. [Online]. Available: http://arxiv.org/abs/1507.04296
[86] M. R. Volodymyr Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, and D. Wierstra, "Playing Atari with deep reinforcement learning," *Deep. Technol.*, vol. 15, pp. 1–9, Dec. 2013.
[87] K. K. Volodymyr Mnih, A. P. Badia, M. Mirza, A. Graves, T. Harley, P. Timothy Lillicrap, and D. Silver, "Asynchronous methods for deep reinforcement learning," in *Proc. 33rd Int. Conf. Mach. Learn.*, New York, NY, USA, vol. 48, 2016, pp. 1–11.
[88] Y.-H. Bui, A. Hussain, and H.-M. Kim, "Double deep *Q*-learning-based distributed operation of battery energy storage system considering uncertainties," *IEEE Trans. Smart Grid*, vol. 11, no. 1, pp. 457–469, Jan. 2020.
[89] T. Tongloy, S. Chuwongin, K. Jaksukam, C. Chousangsuntorn, and S. Boonsang, "Asynchronous deep reinforcement learning for the mobile robot navigation with supervised auxiliary tasks," in *Proc. 2nd Int. Conf. Robot. Autom. Eng. (ICRAE)*, Dec. 2017, pp. 68–72.
[90] Y. P. Awate, "Policy-gradient based actor-critic algorithms," in *Proc. WRI Global Congr. Intell. Syst.*, vol. 3, 2009, pp. 505–509.
[91] V. Veeriah, H. Van Seijen, and R. S. Sutton, "Forward actor-critic for nonlinear function approximation in reinforcement learning," in *Proc. Int. Joint Conf. Auto. Agents Multiagent Syst.*, vol. 1, 2017, pp. 556–564.
[92] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017.
[93] J. Fernando and H. Garcia, "Unifying n-step temporal-difference action-value methods," M.S. thesis, Dept. Comput. Sci., University of Alberta, Edmonton, AB, Canada, 2019.
[94] D. Fuselli, F. De Angelis, M. Boaro, S. Squartini, Q. Wei, D. Liu, and F. Piazza, "Action dependent heuristic dynamic programming for home energy resource scheduling," *Int. J. Electr. Power Energy Syst.*, vol. 48, pp. 148–160, Jun. 2013.
[95] Y. Wei, Z. Zhang, F. R. Yu, and Z. Han, "Power allocation in HetNets with hybrid energy supply using actor-critic reinforcement learning," in *Proc. IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–5.
[96] Z. Wan, H. Li, and H. He, "Residential energy management with deep reinforcement learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–7.
[97] T. Hirata, D. B. Malla, K. Sakamoto, Y. Okada, and T. Sogabe, "Smart grid optimization by deep reinforcement learning over discrete and continuous action space," *Bull. Netw., Comput. Syst. Softw.*, vol. 8, no. 1, pp. 19–22, 2019.
[98] P. Chen, M. Liu, C. Chen, and X. Shang, "A battery management strategy in microgrid for personalized customer requirements," *Energy*, vol. 189, pp. 1–18, Dec. 2019.
[99] P. Odonkor and K. Lewis, "Control of shared energy storage assets within building clusters using reinforcement learning," in *Proc. 4th Des. Autom. Conf.*, Aug. 2018, pp. 1–11.
[100] L. Yu, W. Xie, D. Xie, Y. Zou, D. Zhang, Z. Sun, L. Zhang, Y. Zhang, and T. Jiang, "Deep reinforcement learning for smart home energy management," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2751–2762, Apr. 2020.
[101] E. Mocanu, D. C. Mocanu, P. H. Nguyen, A. Liotta, M. E. Webber, M. Gibescu, and J. G. Slootweg, "On-line building energy optimization using deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3698–3708, Jul. 2019.
[102] OpenAI. (2020). *OpenAI Baselines ACKTR A2C*. Accessed: Apr. 5, 2020. [Online]. Available: https://openai.com/blog/baselines-acktr-a2c/
[103] K. Zhang, Z. Yang, and T. Bar, "Multi-Agent reinforcement learning: A selective overview of theories and algorithms," 2019, *arXiv:1911.10635*. [Online]. Available: https://arxiv.org/abs/1911.10635
[104] R. Lu, S. H. Hong, and M. Yu, "Demand response for home energy management using reinforcement learning and artificial neural network," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6629–6639, Nov. 2019.

[105] H. Li, Z. Wan, and H. He, "Real-time residential demand response," *IEEE Trans. Smart Grid*, vol. 11, no. 5, pp. 4144–4154, Sep. 2020, doi: 10.1109/TSG.2020.2978061.
[106] T. Morstyn, B. Hredzak, and V. G. Agelidis, "Control strategies for microgrids with distributed energy storage systems: An overview," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3652–3666, Jul. 2018.
[107] H. Liang, W. Fu, and F. Yi, "A survey of recent advances in transfer learning," in *Proc. IEEE 19th Int. Conf. Commun. Technol. (ICCT)*, Oct. 2019, pp. 1516–1523.
[108] S. Zhou, L. Zhou, M. Mao, and X. Xi, "Transfer learning for photovoltaic power forecasting with long short-term memory neural network," *2020 IEEE Int. Conf. Big Data Smart Comput. BigComp*, no. 51707026, pp. 125–132, 2020.
[109] H. Wang, S. Fan, J. Song, Y. Gao, and X. Chen, "Reinforcement learning transfer based on subgoal discovery and subtask similarity," *IEEE/CAA J. Automatica Sinica*, vol. 1, no. 3, pp. 257–266, Jul. 2014.
[110] R. Glatt, F. L. Da Silva, and A. H. R. Costa, "Towards knowledge transfer in deep reinforcement learning," in *Proc. 5th Brazilian Conf. Intell. Syst. (BRACIS)*, Oct. 2016, pp. 91–96.
[111] X. Cao, H. Wan, Y. Lin, and S. Han, "High-value prioritized experience replay for off-policy reinforcement learning," in *Proc. IEEE 31st Int. Conf. Tools with Artif. Intell. (ICTAI)*, Portland, OR, USA, 2019, pp. 1510–1514.
[112] I. M. De Abril and R. Kanai, "Curiosity-driven reinforcement learning with homeostatic regulation," *Int. J. Neural Netw.*, vol. 1, pp. 1–6, Jul. 2018.
[113] Y. Burda, A. Storkey, T. Darrell, and A. A. Efros, "Large-scale study of curiosity-driven learning," in *Proc. 7th Int. Conf. Learn. Represent.*, vol. 2019, pp. 1–15.
[114] M. Andrychowicz, "Hindsight experience replay," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2017, pp. 5049–5059.
[115] D. S. Ratcliffe, K. Hofmann, and S. Devlin, "Win or learn fast proximal policy optimisation," in *Proc. IEEE Conf. Games (CoG)*, Aug. 2019, pp. 1–4.
[116] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*. [Online]. Available: http://arxiv.org/abs/1707.06347

ERICK O. ARWA (Graduate Student Member, IEEE) received the B.Sc. degree in electrical and electronic engineering from the University of Nairobi, Nairobi, Kenya, in 2016. He is currently pursuing the master's degree in electrical engineering with the University of Cape Town, Cape Town, South Africa, under the Mandela Rhodes Foundation Scholarship. In 2015, he interned at Powerhive East Africa Ltd., as a Research Assistant in solar energy systems design and deployment. His research interests include operation and control of intelligent microgrids, machine learning, and power systems optimization.

KOMLA A. FOLLY (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1989 and 1993, respectively, and the Ph.D. degree in electrical engineering from Hiroshima University, Japan, in 1997. From 1997 to 2000, he worked with the Central Research Institute of Electric Power Industry (CRIEPI), Tokyo, Japan. He is currently a Professor with the Department of Electrical Engineering, University of Cape Town, Cape Town, South Africa. In 2009, he received a Fulbright Scholarship and was Fulbright Scholar with the Missouri University of Science and Technology, Rolla, MO, USA. His research interests include power system stability, control and optimization, HVDC modeling, grid integration of renewable energy, application of computational intelligence to power systems, smart grids, and power system resilience. He is a member of the Institute of Electrical Engineers of Japan (IEEJ), and a Senior Member of the South African Institute of Electrical Engineers (SAIEE).