

Parametric study of adaptive reinforcement learning for battery operations in microgrids

Deepak Kumar Panda^{a,d}, Saptarshi Das^{a,b,*}, Mohammad Abusara^c

^a Centre for Environmental Mathematics, Faculty of Environment, Science and Economy, University of Exeter, Penryn Campus, Cornwall, TR10 9FE, UK

^b Institute for Data Science and Artificial Intelligence, University of Exeter, Devon, Exeter, EX4 4QE, UK

^c Department of Engineering, Faculty of Environment, Science and Economy, University of Exeter, Penryn Campus, Cornwall, TR10 9FE, UK

^d School of Engineering and Applied Sciences, Cranfield University, Cranfield, MK430AL, UK

ARTICLE INFO

Keywords:

Microgrid management

Battery operations

Q-learning

Reinforcement learning convergence

Adaptive RL

ABSTRACT

Reinforcement learning (RL) has been increasingly used for efficient energy management systems (EMSs) in microgrids. The battery storage system in the microgrid can be controlled using efficient policies derived from RL. However, little attention has been paid so far to the parametric study, which is a fundamental step for efficient implementation of such RL algorithms. Unlike previous works which focused on the implementation of different RL algorithms, this paper mainly demonstrates the parametric sensitivity study of the RL algorithms. It involves investigating the effects of (1) controllable state discretization, (2) exogenous state discretization, (3) action discretization, (4) exploration and exploitation parameters, and (5) decision intervals. Moreover, the performance of the ϵ -greedy randomized RL algorithm is compared against the adaptive Q-learning, derived from the adaptive approximate dynamic programming (ADP). In many microgrids utilizing solar energy and battery storage, energy management still relies on manually tuned and inefficient algorithms. This is largely due to the sensitivity of RL algorithm parameters to factors such as the specific EMS problem, environment, action-state discretization, exploration parameter and time step. We show the univariate and multivariate kernel density estimate (KDE) plots to study the RL algorithms' performance concerning the rewards and variation of the battery state of charge (SoC) and the net power imported from the grid. Overall, the deterministic adaptive RL performs better as compared to the randomized ϵ -greedy algorithms in terms of rewards and simulation time. Higher discretization levels in the action space affect the convergence rate while lower discretization levels in the state space influence the performance of the algorithm. The proposed parametric analysis can be easily adapted to other EMS in more complex microgrids.

1. Introduction

With the increasing cost of fossil fuels and environmental concerns, it is essential to develop appropriate control strategies and efficient management of microgrids consisting of renewable energy resources (RES), battery energy storage systems (BESSs), and local load variations. Residential and commercial microgrids can maximize the utilization of locally produced RES power, and hence reduce the overall cost of electricity imported from the main grid. BESSs are deployed to reduce power fluctuations and the inherent volatility of RES [1]. They are also utilized by intelligent EMSs to charge during low tariff and discharge during high tariff within the day, thereby reducing costs while making

the best use of the available RES power [2]. Among competing optimization methods, RL is well known for its adaptability and ability for self-learning in model-free environments using random simulations.

1.1. Existing literature on RL in energy management

RL is a popular control and optimization algorithm and has been widely used in the management of microgrids [3–8], charging of electric vehicles (EVs) [9–11], energy management in hybrid electric vehicles [12–16] and demand-side management (DSM) for scheduling appliances usage. In microgrids, control algorithms for BESS focus on utilizing local RES to reduce power imported from the grid. In Refs. [6,17], power

This article is part of a special issue entitled: Energy Storage for Sustainability published in Renewable Energy.

* Corresponding author. Centre for Environmental Mathematics, Faculty of Environment, Science and Economy, University of Exeter, Penryn Campus, Cornwall, TR10 9FE, UK.

E-mail addresses: dp457@exeter.ac.uk (D.K. Panda), saptarshi.das@ieee.org (S. Das), m.abusara@exeter.ac.uk (M. Abusara).

<https://doi.org/10.1016/j.renene.2025.123250>

Received 25 April 2024; Received in revised form 22 April 2025; Accepted 24 April 2025

Available online 27 April 2025

0960-1481/© 2025 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

imported from the main grid is only considered while other works e.g. Refs. [3,5,7,18] focus on bidirectional exchange of power with the main grid. Continuous state space has been generally utilized for the control actions of the battery and hence the principle of function approximation has been used. Previous studies have considered control actions for EMS based on the availability of hourly data. However, in Ref. [17], the optimal policy of the state action pair is obtained from the Q-table where the control decisions are updated at 5-min intervals. In Ref. [18], the derived Q-learning policy is compared with the randomized Particle Swarm Optimization (PSO) algorithm, where the former had a better cost-saving performance. The performance of batch RL is analyzed considering the influence of the exogenous data like the solar power generation in Ref. [4]. The performance is benchmarked with that of the Mixed Integer Linear Programming (MILP) method. In Ref. [19], the tabular Q-learning method is compared with different variants of the algorithm, like retaining the previous action, minimum instantaneous reward, and hourly optimization. In Ref. [5], rewards are compared by changing the initial SoC, and the results are benchmarked against the reward obtained from greedy optimization algorithms where the performances are also analyzed by changing the hyper-parameters of the deep neural networks (DNNs) employed for function approximation in the case of non-stationary data. In Ref. [7], the performance of RL algorithm is analyzed with the centralized and decentralized control mode of the grid for the community BESS. The whole system is analyzed in the grid-connected and islanded mode of operation. Furthermore, the validity of the algorithm is tested on a longer time scale of 48 h. However, an in-depth sensitivity analysis of different parameters of RL algorithms, i.e., states, actions, exploration/exploitation rates, have not been explored in the previous RL-EMS literature for optimal microgrid management.

Recent advancements in deep reinforcement learning (DRL) for battery energy management in electric vehicles (EVs) have demonstrated significant adaptability to real-time driving environments, particularly under dynamic traffic conditions [20,21]. To handle the complex interactions within real-time distribution networks, multi-agent DRL (MADRL) techniques have been employed, showcasing enhanced scalability and coordination to optimize battery operational costs [22]. Additionally, battery operational costs have been integrated with battery swapping logistics, considering the optimization of truck routes [23]. Hierarchical coordination between Distribution System Operators (DSOs) and batteries in multi-microgrid setups have also been explored, optimizing operational costs by managing power imports from the grid under realistic constraints [22].

A cyclic time-dependent Markov Decision Process (MDP) is utilized in Ref. [24] to capture daily patterns in energy management, while conditional generative adversarial networks are applied in Ref. [25] to generate wind and solar power scenarios for DRL applications. The works in Refs. [26,27] incorporate battery operational metrics such as aging, optimal state of charge, and charge-discharge cycles into the reward functions of DRL algorithms. Safety aspects are further enhanced using projection techniques to improve the trustworthiness of DRL decisions [28]. Studies like [29] focus on cell balancing and switching for RES applications, while [30] integrates various elements, such as wind turbines and thermostatically controlled loads, with energy storage systems managed by DRL algorithms. Moreover, online RL techniques for grid-connected battery management, considering dynamic tariffs, have been explored to further refine energy management operations [31].

1.2. Contribution of this paper

Unlike previous works, the main contribution of this paper is a thorough parametric study of adaptive Q-learning along with traditional ϵ -greedy Q-learning algorithms. The analysis has been conducted on 24-h load demand and solar photovoltaic (PV) data, which can be forecasted using advanced algorithms [32,33]. Similar training of the RL on

a daily horizon of load demand and solar PV data has been adopted in Ref. [7]. Since the number of state-action pairs is finite and a Time of Use (ToU) tariff is used, a tabular Q-learning method is employed instead of a function approximator in Ref. [7]. Q-table can be used for framing policies with finite state space due to its simplicity and faster convergence during learning. With the use of Q-table for finding an optimal policy, it is imperative to investigate the discretization levels for different types of states and actions used, which have not been addressed in the existing RL-EMS literature. Previously the value of ϵ , which signifies the percentage of the time agent explores the action space randomly, have been considered arbitrarily.

This paper builds upon our previous work [34] but introduces a novel adaptive algorithm that deviates from the traditional exploration-exploitation trade-off for action selection. Instead, it employs a deterministic algorithm with dual-layer optimization, incorporating robust theoretical convergence guarantees, adapted from Ref. [35]. In addition, a detailed parametric study inspired from Ref. [12] is conducted, where the grid cost is equivalent to mileage in hybrid electric vehicles. However, in this paper, the novel contribution lies in the comprehensive analysis of RL algorithm performance across various discretization levels of the controlled state (SOC of the battery), exogenous state (the net power to be met by the battery) (P_{Net}) and the battery action space (charging/discharging commands) (P_b). The performance of the algorithm is also analyzed by varying ϵ , which controls the exploratory nature of the learning agent. In addition, the decision made by the EMS based on data available at different time intervals (Δt) is also analyzed, which has not been addressed in the previous RL based EMS literature. The reward function, in the deployed algorithms is modified to reflect the overall economic cost, and the penalties to prevent operation at extreme ends as presented in Ref. [7]. The adaptive Q-learning algorithm, derived from the ADP as given in Ref. [35], is utilized for the EMS for the first time, using an auxiliary table to update the main Q-table, thus accelerating convergence. The algorithm tunes the adaptive parameter for policy evaluation, which achieves a trade-off between the current and future Q-functions. The performance of Q-learning techniques, as described in Ref. [36], is compared for all the parametric variation along with the novel adaptive Q-learning algorithm for EMS. The parametric study is conducted using the univariate and bivariate KDE plots of the parameters obtained from the numerical simulations of the RL algorithm. Furthermore, due to the ToU tariff used, the performance of the algorithm has also been checked at various tariff states. Overall, the main contributions of this paper can be summarized as:

- Parametric study of different state and action discretization levels and decisions at control intervals for the adaptive and temporal difference (TD) type Q-learning algorithm. The adaptive Q-learning algorithm is implemented for EMS in a microgrid with solar PV generation, offering a trade-off between the current and future Q-function for policy evaluation.
- Comparison of the performance of TD-type Q-learning with different exploration parameters of the learning agent.
- Comparison of the performance of different variants of Q-learning algorithms like repeated trace and value iteration with the adaptive Q-learning algorithm.

The manuscript is structured into six sections. Section 2 explores the microgrid environment, incorporating battery storage and solar PV systems. Section 3 conceptualizes the microgrid as an MDP, detailing its state space, action space, transition dynamics, and reward function. Section 4 introduces the benchmark RL algorithms and elaborates on the adaptive RL strategy, supported by theoretical proofs. Section 5 presents the numerical implementation, results, and discussions, highlighting various algorithms across different parametric studies. Finally, section 6 provides the conclusion and future scope of research.

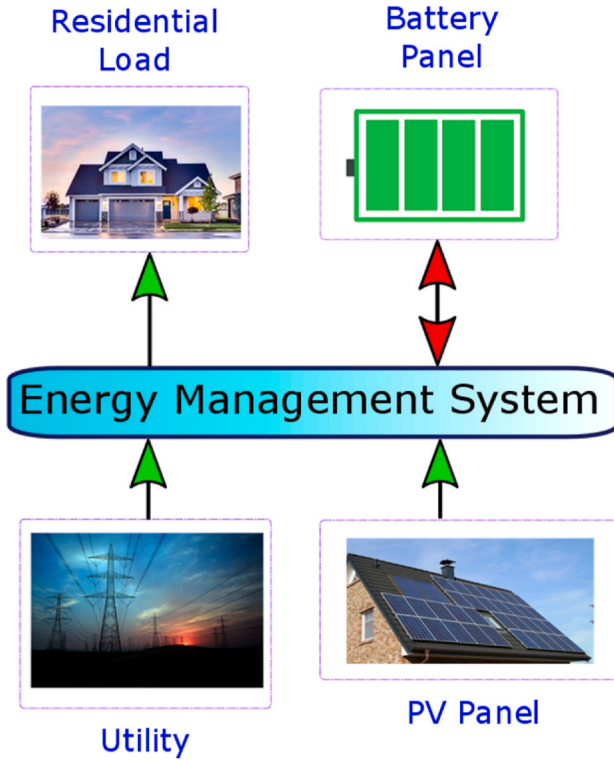


Fig. 1. Microgrid architecture along with the direction of power flow.

2. Microgrid with battery storage and solar PV

2.1. Description of the microgrid with battery and EMS

The system used in this study consists of a residential load, BESS, local PV generation, utility grid, and their interface with the EMS unit, as shown in Fig. 1. The load demand is met using the power generated from the solar PV, power discharged from the battery and the remaining power needs to be imported from the main grid. The power flow for the battery storage system is considered as bidirectional, based on the control commands of the EMS.

The utility price considered here is ToU tariff, described in Ref. [37] where the price of the electricity varies according to the general load

demand pattern. The tariff strategy is to provide monetary benefits to the consumers for shifting their demand in the day. The data used for the tariff is Green UK tariff policy, as described in Ref. [38]. Samples of the load demand, tariff and PV generation profiles used in this paper are shown in Fig. 2. As observed, three levels of the tariff are used, peak, mid-peak and off-peak as given in Refs. [38–40]. Peak tariff is observed when the load demand is considerably higher with a moderate amount of PV generation.

2.2. Battery state of charge (SoC) model

In a battery storage system, $P_b(t)$ represents the battery power, where it is positive when charging and negative when discharging. The SoC represents the percentage of the energy available within the battery. The SOC(t) of the battery is represented by the difference equation:

$$\text{SOC}(t) = \text{SOC}(t-1) + P_b(t) \cdot \Delta t / E_b, \quad (1)$$

where, E_b represents the energy capacity of the battery and Δt represents the action duration of the time step. However, the SOC of the battery must stay within the given limits to prevent over-charging and over-discharging. Hence it is constrained between the following interval:

$$\text{SOC}_{\min} \leq \text{SOC}(t) \leq \text{SOC}_{\max}, \quad (2)$$

where, $\text{SOC}_{\min}$ and $\text{SOC}_{\max}$ represent the minimum and maximum SOC limits, respectively. The rated power of the battery (P_{rated}) used in this study is 2 kW with the energy capacity (E_b) is 8 kWh. The values of $\text{SOC}_{\min}$ and $\text{SOC}_{\max}$ used here are 15 % and 100 %, respectively. The difference equation (1) is used to describe the next state of the environment once the agent has decided battery power $P_b(t)$ in the time interval Δt . The overall characteristics of the environment for the RL agent are described in the next section.

3. Description of the RL environment

In RL, the agent interacts with the environment by taking a certain action based on the given state and obtaining a certain reward. Hence the states, actions and the reward used in the study are described in the next subsections.

3.1. The state space

As described in Refs. [4,5,41], the state space (S) of the environment

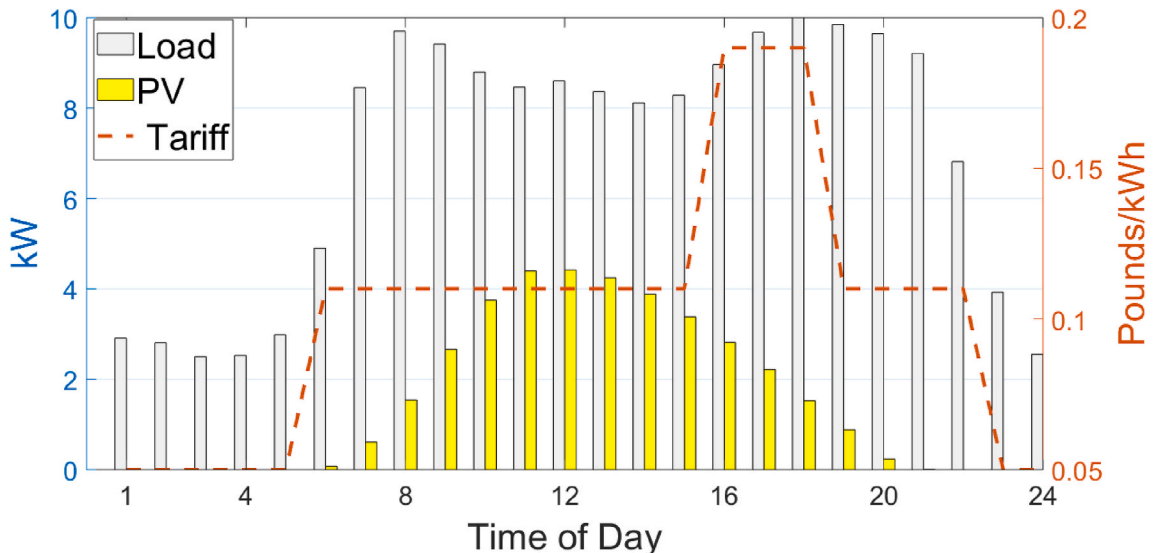


Fig. 2. Daily load demand, PV generation and ToU tariff used in the current data analysis and simulations.

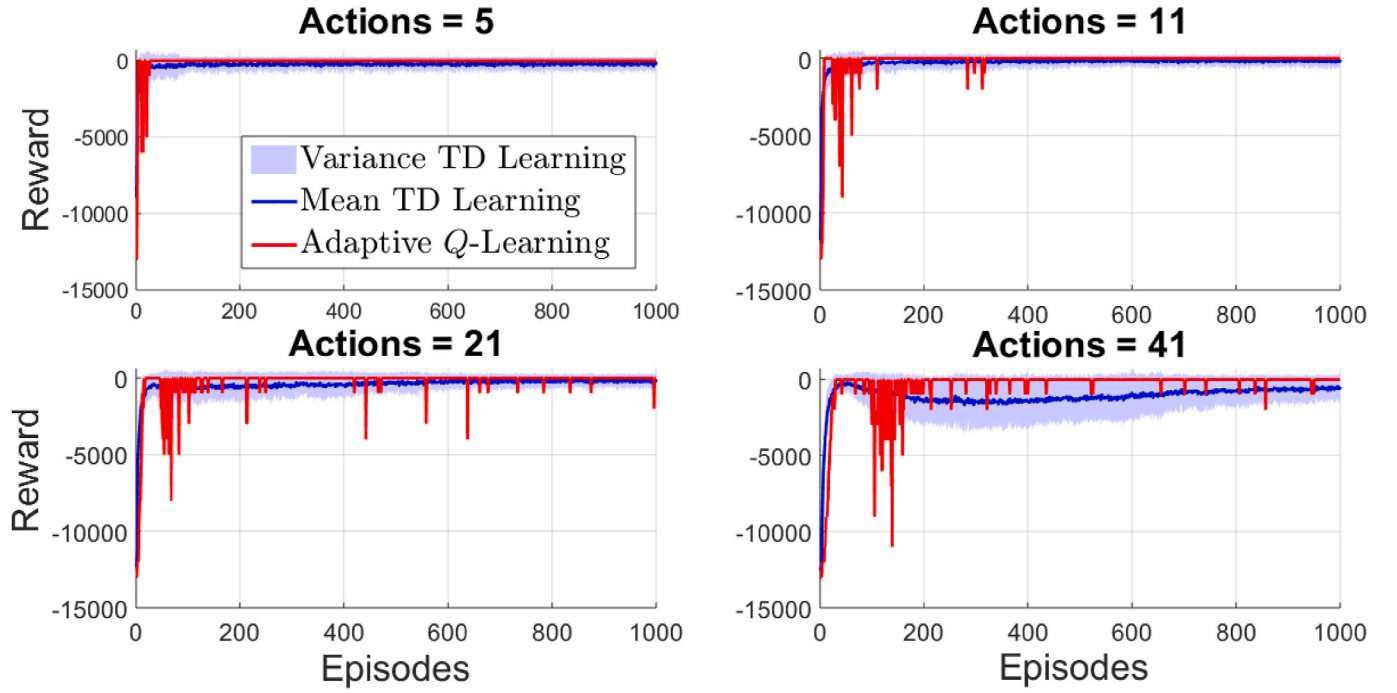


Fig. 3. Reward plots for different action discretization levels for TD (for 100 Monte Carlo runs) and adaptive Q-learning.

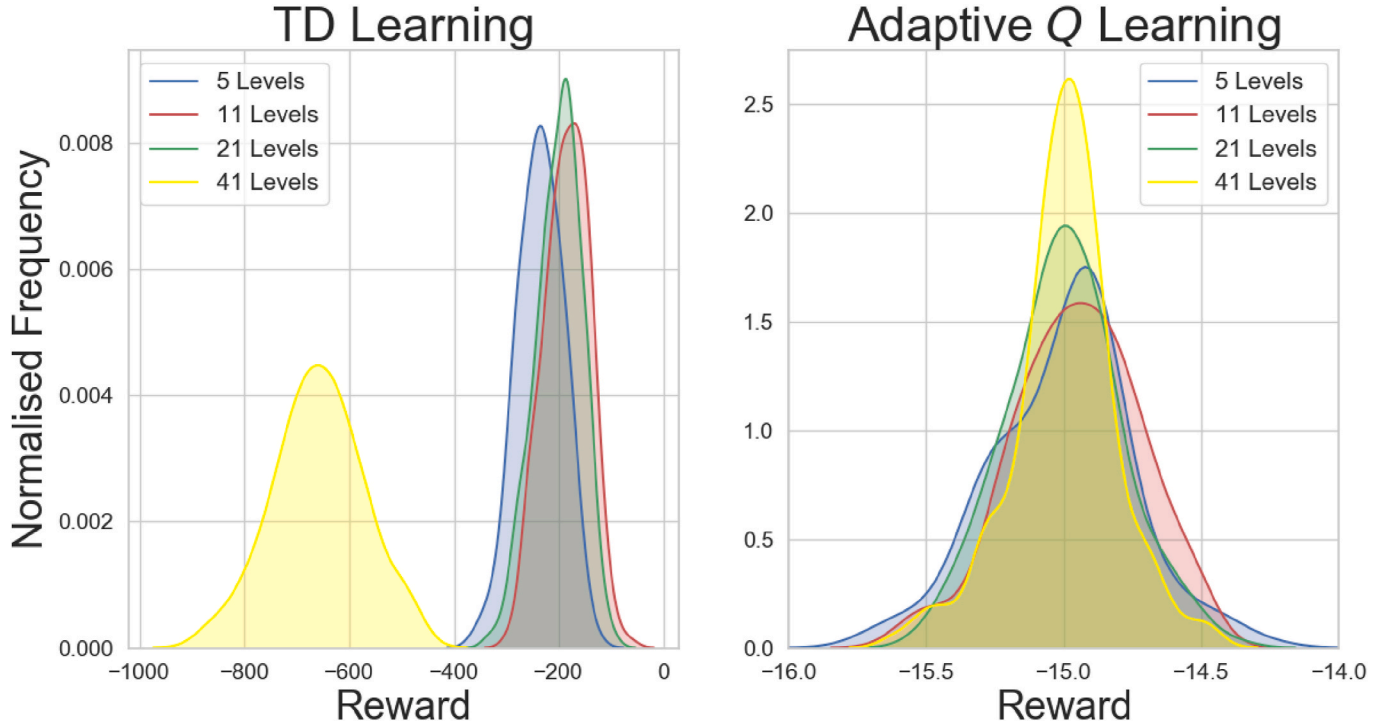


Fig. 4. Histograms of rewards for the last 200 episodes for different action discretization levels for temporal difference and adaptive Q-learning.

generally consists of three features, namely, timing component (S_t), controllable component (S_c) and non-controllable component (S_x). Since tabular Q table has been used for selecting control actions, the policy values are stored corresponding to the discrete value of the states. Hence, the discrete state space is represented as:

$$\mathbf{S} = S_t \times S_c \times S_x. \quad (3)$$

The timing component S_t is time-dependent, and it incorporates the environment's state information at a given time. Since the tariff is time

dependent, it is incorporated as the timing component. The ToU tariff used in the study consists of three states [39,40], as shown in Fig. 2. It is defined as:

$$S_t = \{\text{'Off - Peak'}, \text{'Mid - Peak'}, \text{'Peak'}\}. \quad (4)$$

The controllable component S_c consists of the information of the state, which is measured and controlled from the agent's actions.

For this study, SOC is considered as the controllable component. The SOC is controlled using the battery power action $P_b(t)$ at time t as

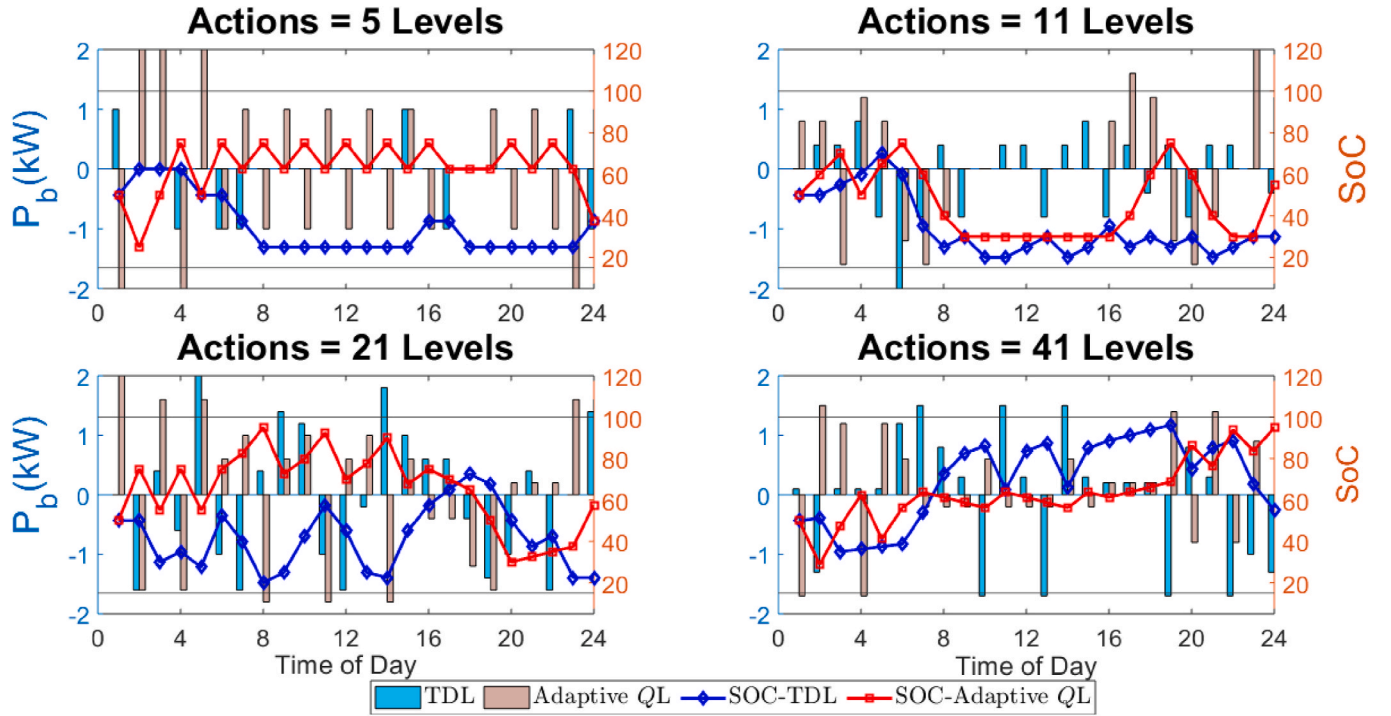


Fig. 5. Battery power at different action discretization level along with the SoC variation for TD-learning and adaptive Q-learning, considering the same initial Q-table initialization in the last episode.

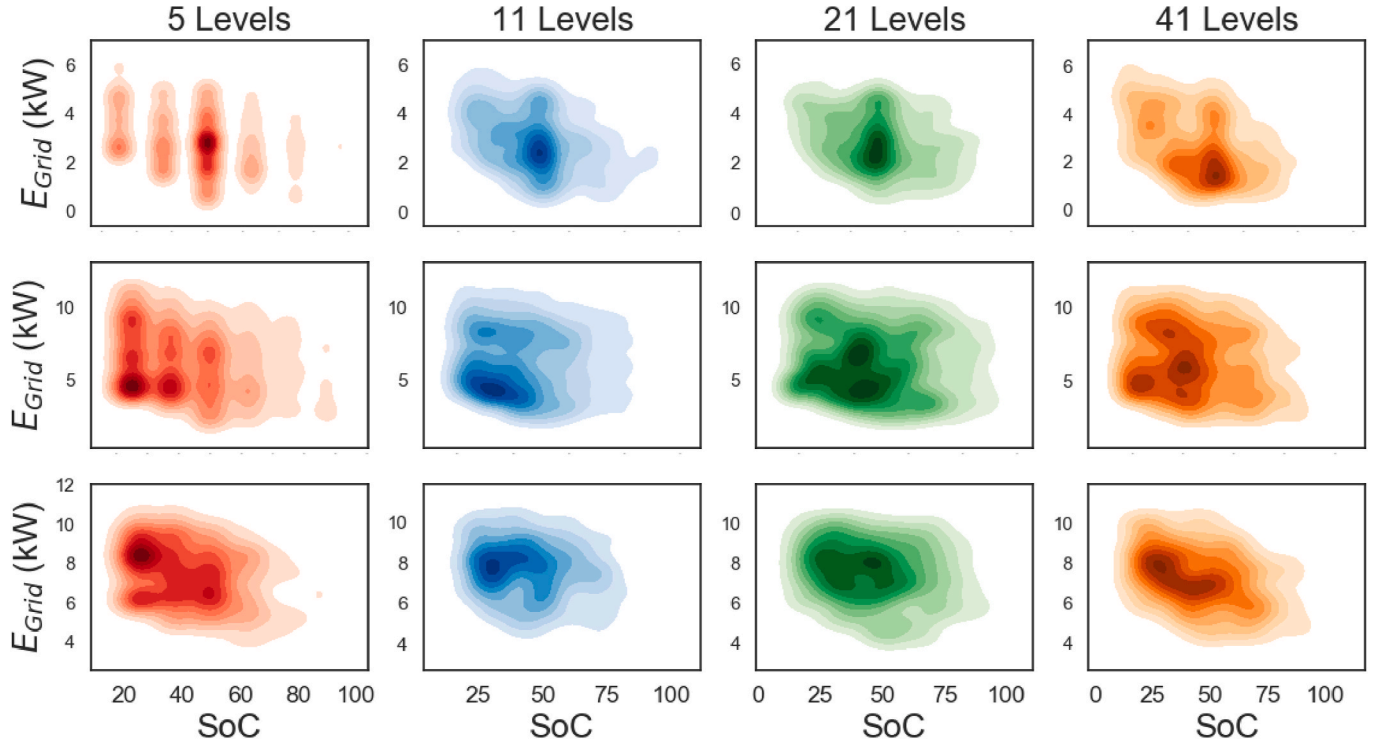


Fig. 6. Bivariate density plots of the power drawn from the main grid along with the SoC level variations for all the Monte Carlo simulations at different action discretization during different tariff periods. Row 1: off-peak, row 2: mid peak, row 3: peak.

defined in (1). However, the agent needs to guarantee the safety limits imposed in (2). The SOC value obtained after taking an action is rounded off to the nearest bin value to derive the appropriate policy. The non-controllable component S_x has a significant influence on the controllable component and the rewards obtained by the agent. However, it is

not influenced by the control actions taken by the agent. This feature is dependent on external conditions. The net power $P_{Net}(t)$ is used as non-controllable component and it is represented as:

$$P_{Net}(t) = P_L(t) - P_{PV}(t), \quad (5)$$

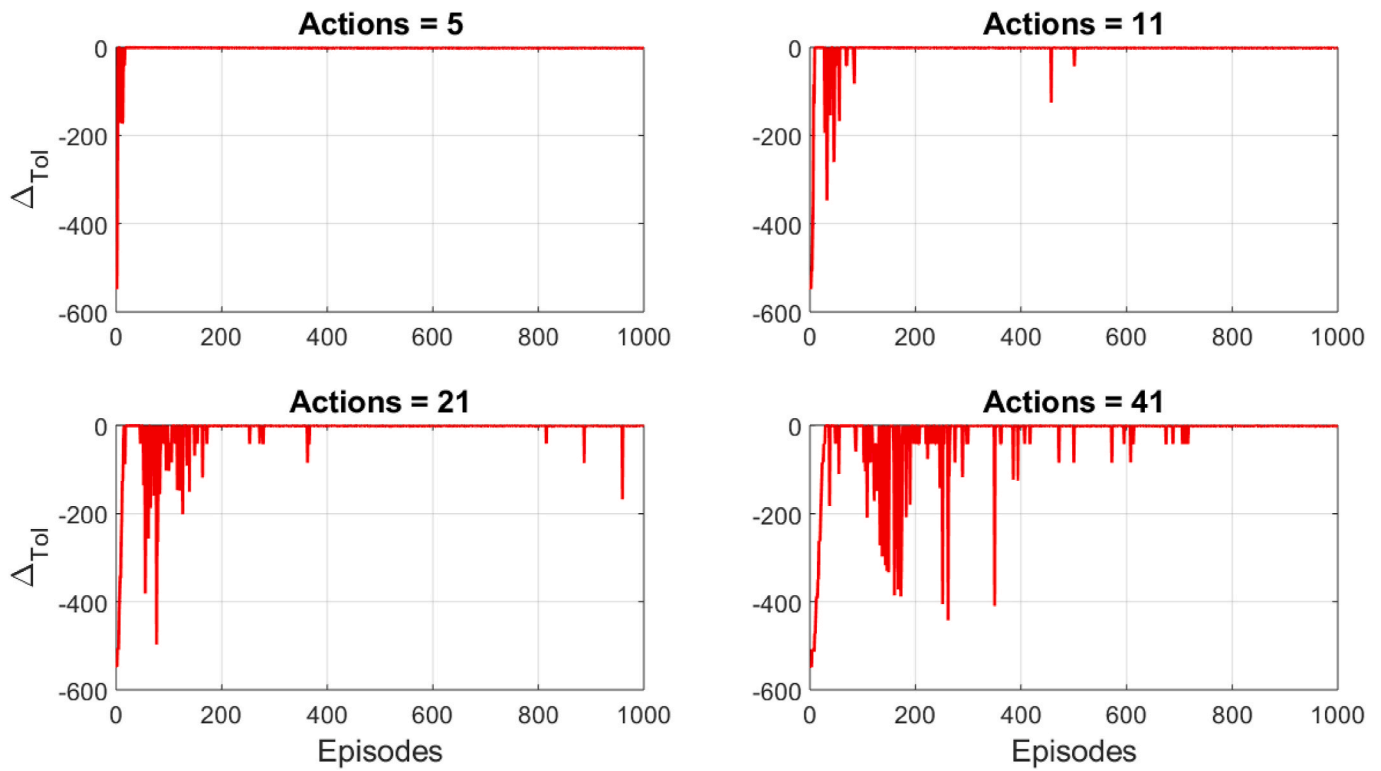


Fig. 7. Convergence characteristics of adaptive Q-learning for different episodes for different action discretization levels.

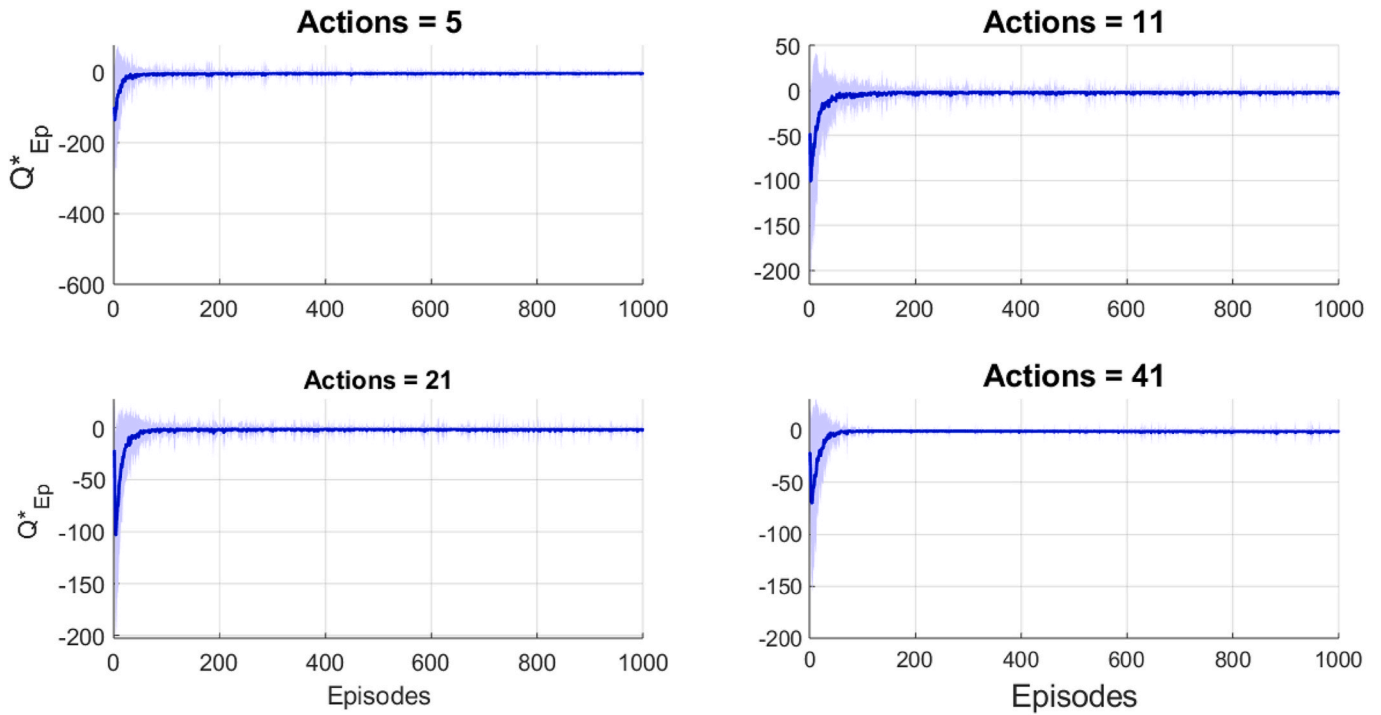


Fig. 8. Convergence characteristics of the TD-Learning after 100 Monte Carlo runs. For different action discretization levels.

where, P_L is the load demand and P_{PV} is the PV power generated. The $P_{Net}(t)$ is discretized to the nearest bin, to formulate the policy. With the availability of different states, it is essential to analyze the agent's actions while it interacts with the environment.

3.2. The action space

The agent takes an action $P_b(t)$, which is a supervisory command to the inverter for regulating the power flow of the battery. It depends on the given state s_t at time t . The action space P is discretized into n levels such as:

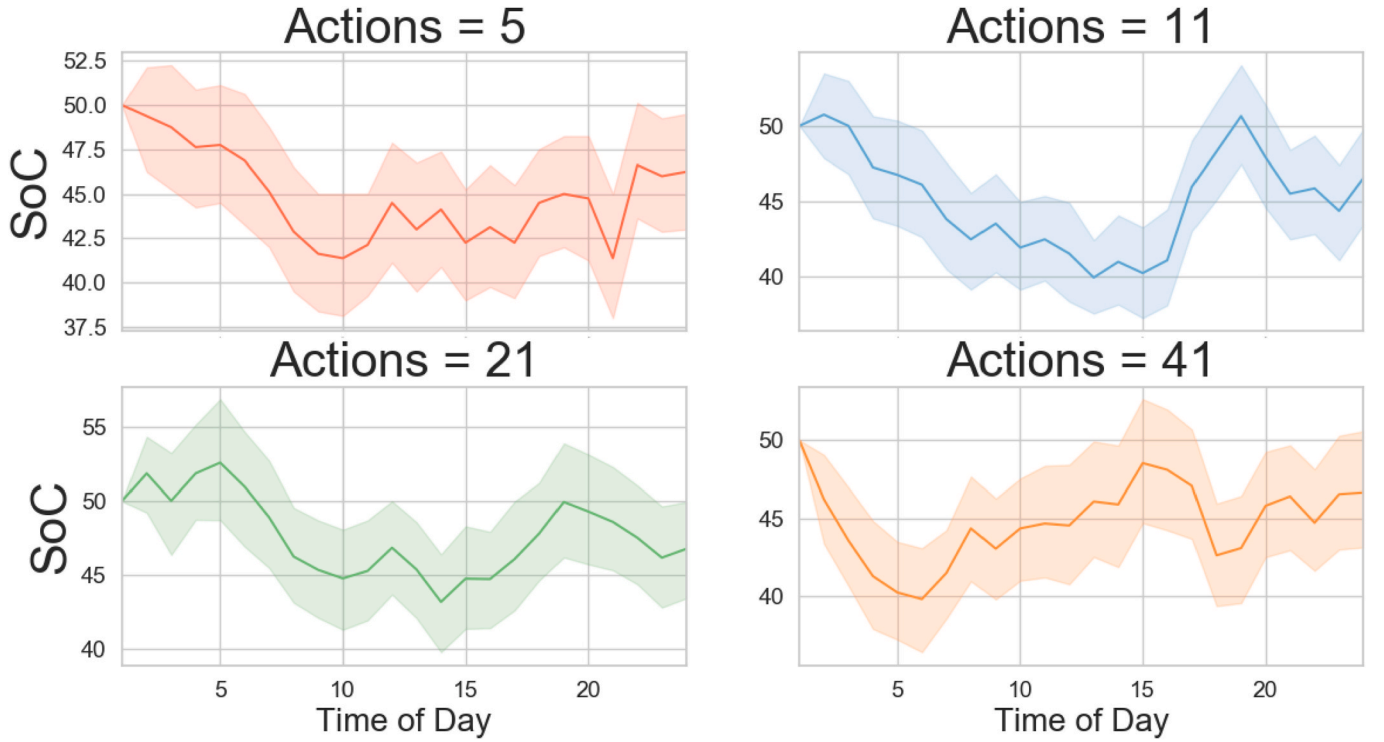


Fig. 9. SOC variation for different action level discretization levels for TD learning during different time of the day for 100 Monte Carlo runs.

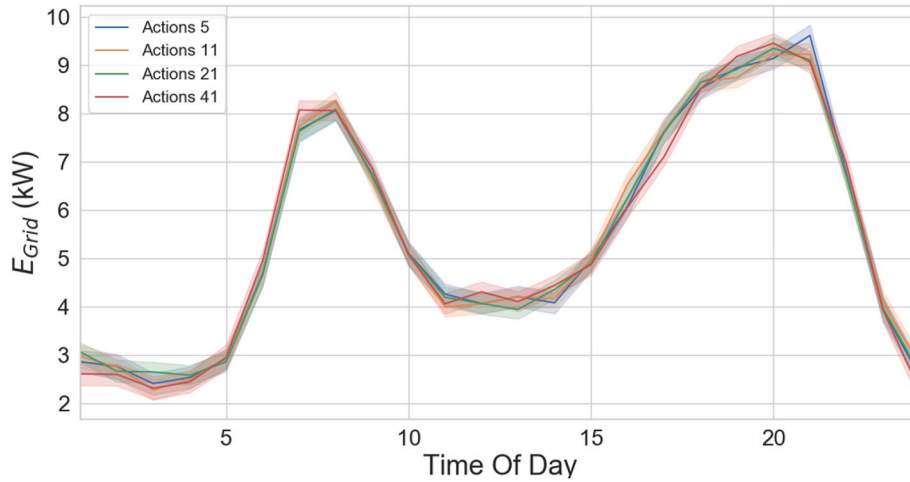


Fig. 10. The power imported from the grid at different action discretization levels for 100 Monte Carlo runs for temporal difference learning.

$$P = \{P_b^{(1)}, \dots, P_b^{(n)}, 0, -P_b^{(n)}, \dots, -P_b^{(1)}\}, \quad (6)$$

where, $P_b^{(i)} = [(n - i + 1) / n] \cdot P_{\text{bat}}$.

However, it is essential that the SOC limits, as defined in (2), are not exceeded. If the action determined by the policy would violate these limits, it is overridden. The override action depends on the timing component of the state S_t , i.e. the tariff. This dependency is created so that battery is charged during the less expensive tariff period, and the energy being conserved when the tariff is high. The override action when SOC goes below $\text{SOC}_{\min}$ can be defined as:

$$P_b^{\text{ov}}(t) = \begin{cases} 0, & \text{if } S_t = \{\text{'Peak'}\}, \\ P_{\text{bat}}, & \text{else.} \end{cases} \quad (7)$$

The idle action is preferred during peak period so that the battery is not charged during the expensive tariff. Similarly, the override action,

when SOC goes beyond, the $\text{SOC}_{\max}$ is defined as:

$$P_b^{\text{ov}}(t) = \begin{cases} 0, & \text{if } S_t = \{\text{'Off Peak'}\}, \\ -P_{\text{bat}}, & \text{else.} \end{cases} \quad (8)$$

The agent's reward is formulated based on the actions defined in Ref. [7], with the objective of deriving an optimal policy. Further details are provided in the following subsection.

3.3. The reward function

The overall objective of the RL agent is to maximize the future reward. Conventionally, the reward is related to the grid cost, which has to be effectively minimized. Thus, the reward function is defined as:

$$r_t(s_t, a_t) = -P_g \cdot C_{S_t}, \quad (9)$$

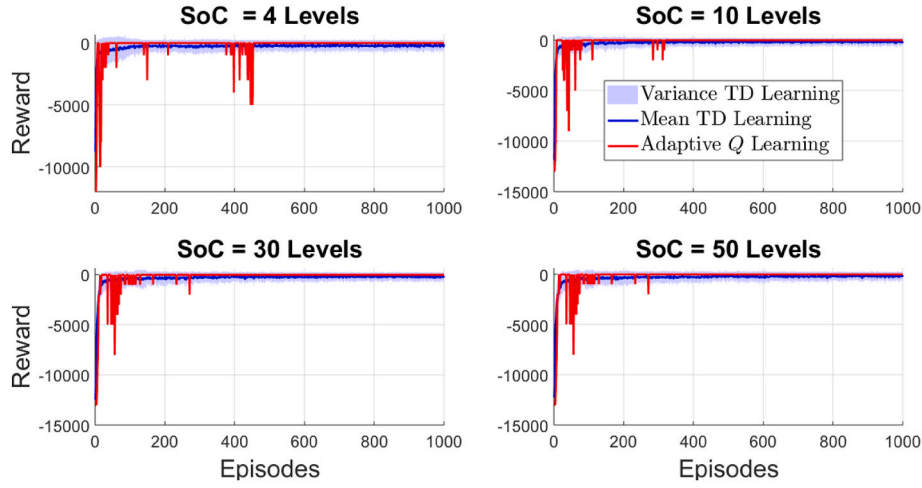


Fig. 11. Reward plots for different controllable state discretization levels for temporal difference (for 100 Monte Carlo runs) and adaptive Q-learning.

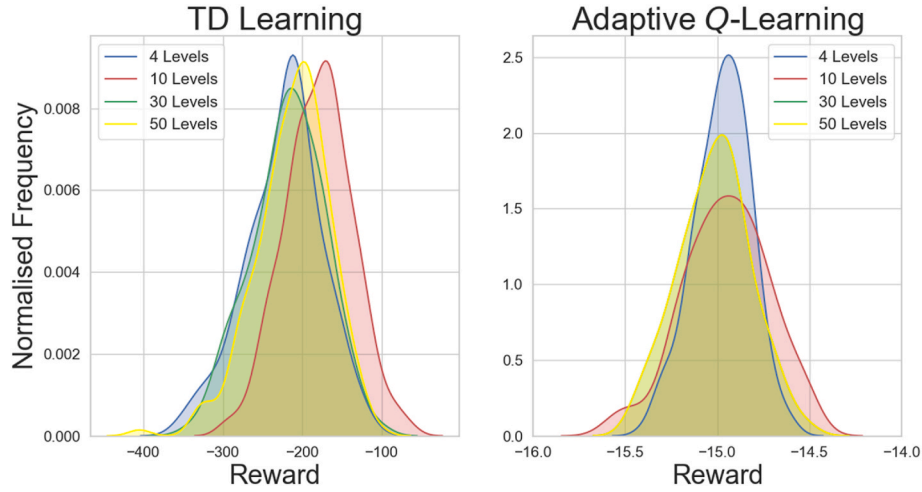


Fig. 12. Histograms of reward for last 200 episodes for different controllable state discretization levels for mean temporal difference and adaptive Q-learning algorithm considering the same initial Q-table initialization.

where, C_{S_t} is the tariff cost at time t , and $P_g = P_{Net}(t) - P_b(t)$ is the net power imported from the grid, and a_t represents the action taken by the agent as defined in the previous section. The battery power considered here in both positive and negative, considering charging and discharging nature of the battery. We have assumed that P_g remains positive throughout the day. A negative value is used so that the reward is effectively maximized, thus minimizing the overall grid cost. However, the override actions, (7) and (8), are sub-optimal since they are not derived from the policy. As given in Ref. [42], the life of the battery depends on its mode of operation. It should not operate at a high depth of discharge, which corresponds to a low state of charge. Moreover, the battery should avoid operating near the high state of charge mode, as it acts like a power-consuming device rather than a generating device, hence undermining the objective. Therefore, a high penalty term has been introduced, derived from Ref. [7] as:

$$r_t(s_t, a_t) = -1000. \quad (10)$$

The magnitude of the penalty for taking a sub-optimal action in (10) is considered high as compared to the penalty for consuming the power

from the grid, defined in (9). As we observe, the agent receives a penalty (10) and a corrective action defined in (7) and (8). Both the penalty and corrective mechanism guide the agent to converge towards an optimal policy. The following section outlines how this optimal policy is derived using different RL algorithms.

4. Reinforcement learning algorithms

The RL algorithm used in this paper is based on the Q-table due to its simplicity. Q-learning is an iterative algorithm that requires initialization of the Q-table. A high initial value encourages exploration of the action space and can lead to slow convergence. Zero-initialization of the Q-table tends to make the agent biased towards the unexplored states, causing premature convergence. In order to mitigate these issues, an initial positive bias is added to the reward function [12] so that it explores the whole space of state-action pairs. However, as per [43], if the decision making policy is initialized with non-zero initial rewards, the convergence characteristics become faster. Hence, in this study, Q-table is initialized using rewards obtained from randomly selected action values. Moreover, the max Q-value of the explored states at time t for the

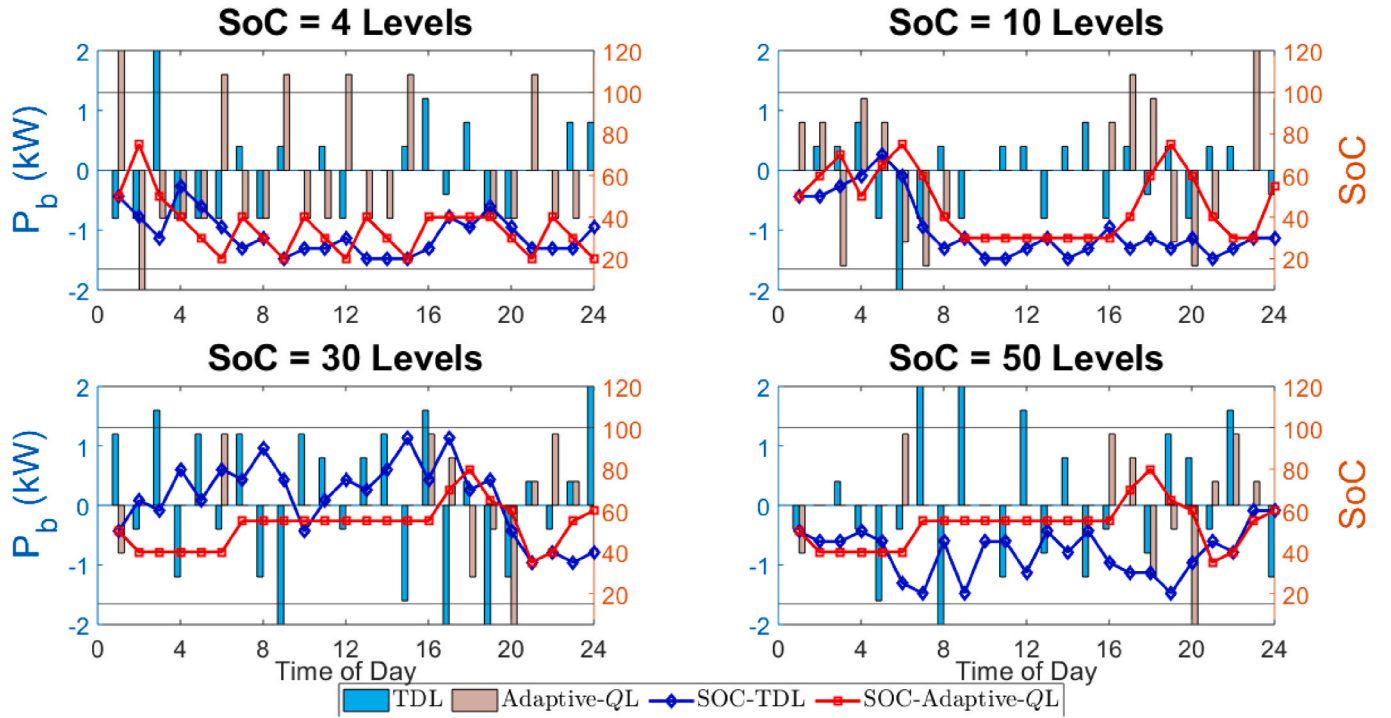


Fig. 13. Battery power at different controllable state discretization level along with the SoC variation for TD learning and adaptive Q-learning, considering the same initial Q-table initialization in the last episode.

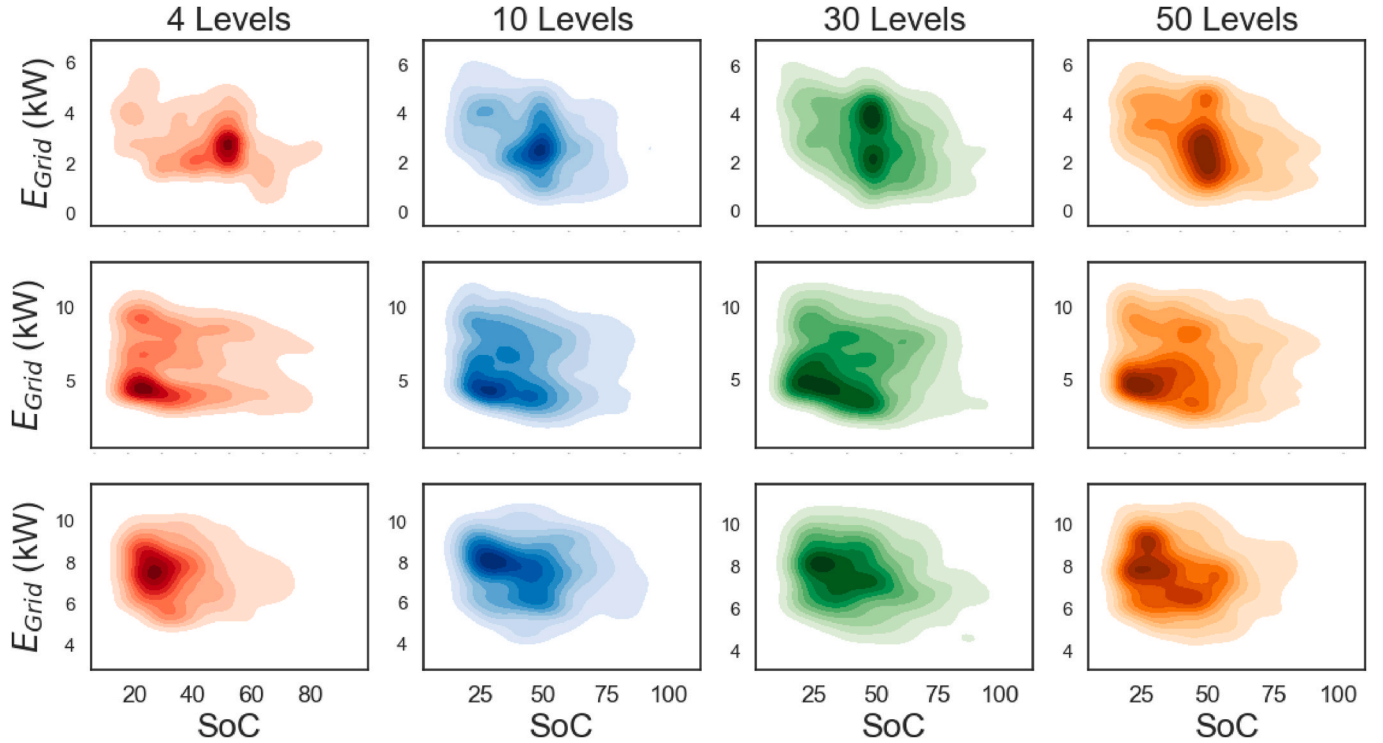


Fig. 14. Bivariate density plots of the power drawn from the main grid along with the SoC level variation for all the Monte Carlo simulation at different controllable state discretization during different tariff periods. Row 1: off-peak, row 2: mid peak, row 3: peak.

update equation has been used. The performance of the algorithm is dependent on the reward obtained from the randomly chosen value used for the initialization of the Q-table. Therefore, simulations are performed with different Q-table initialization from the rewards of different initial random actions.

4.1. Off-policy reinforcement learning value prediction

As given in Ref. [44], Monte Carlo and temporal difference methods rely on the experiences to solve the prediction problem. However, in the former method, the policy update takes place at the end of each episode,

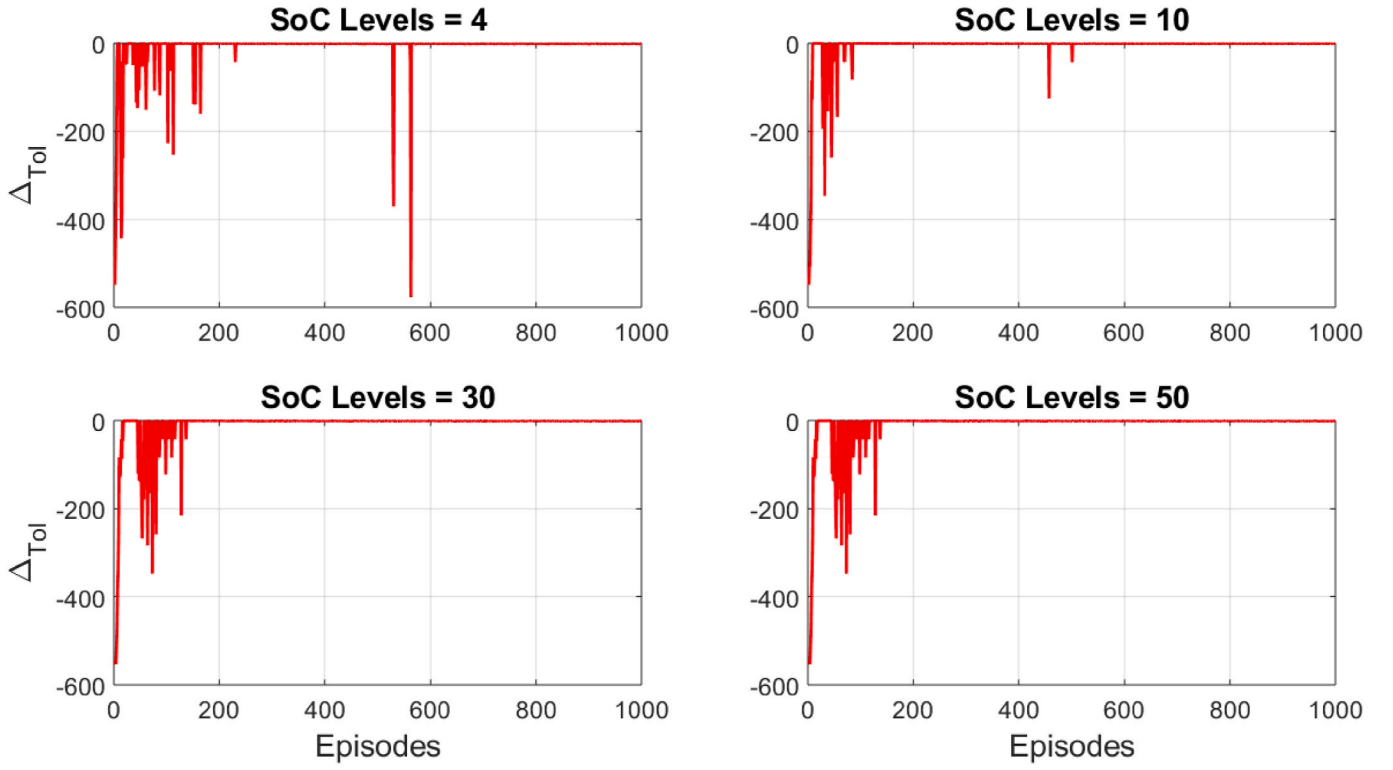


Fig. 15. The convergence characteristics at different SOC discretization levels for adaptive Q-learning algorithm.

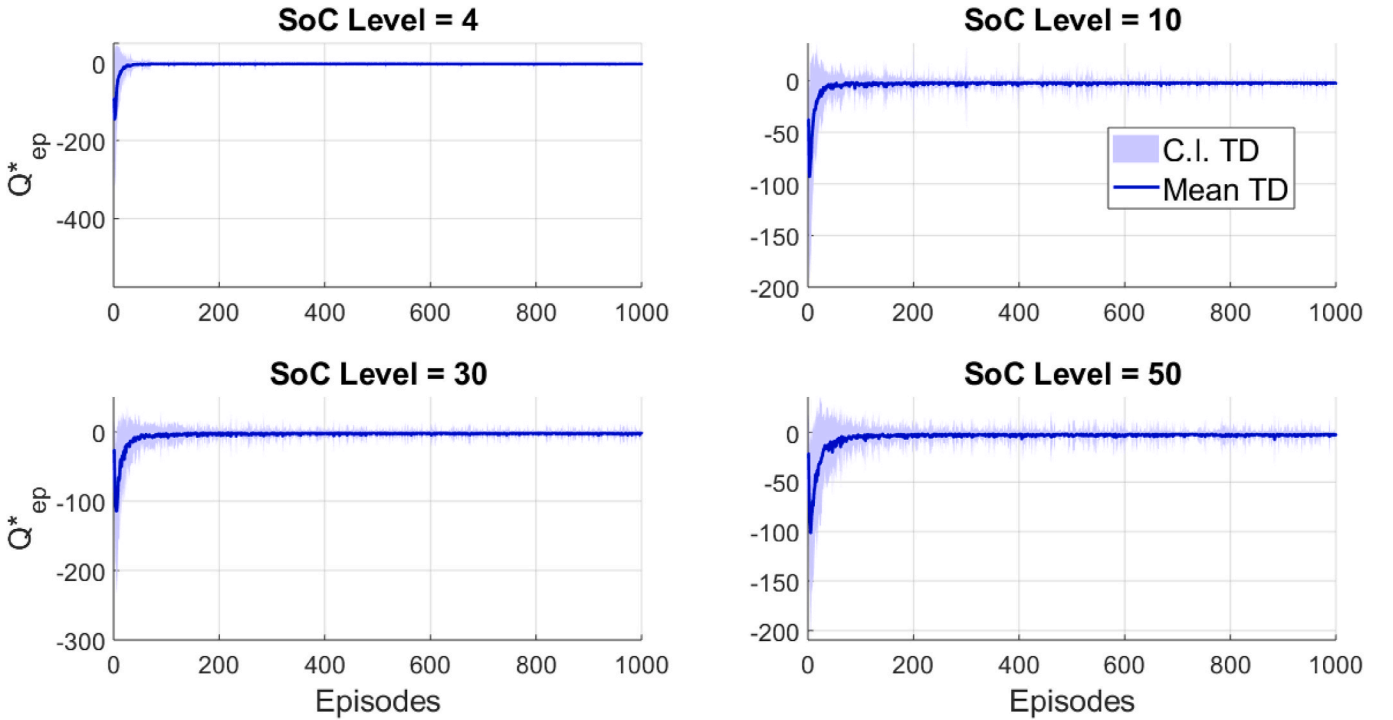


Fig. 16. Convergence characteristics of the TD Learning process for different SoC discretization levels for 100 Monte Carlo runs. Sky Blue-Variance, Blue-Mean.

after the rewards are obtained. While in the latter method, the learning is bootstrapped, i.e. estimated from an incremental experience. The exploration and exploitation of the action space are managed using ϵ greedy policy which is defined as:

$$a_t = \begin{cases} \operatorname{argmax}_a Q(s_t, a) & \text{with probability } (1 - \epsilon), \\ \text{random action P} & \text{with probability } \epsilon. \end{cases} \quad (11)$$

The ϵ parameter is used for exploration/exploitation where higher value promotes more exploration in the action space. Once the agent takes action a_t and obtains reward r_t , and the environment transitions

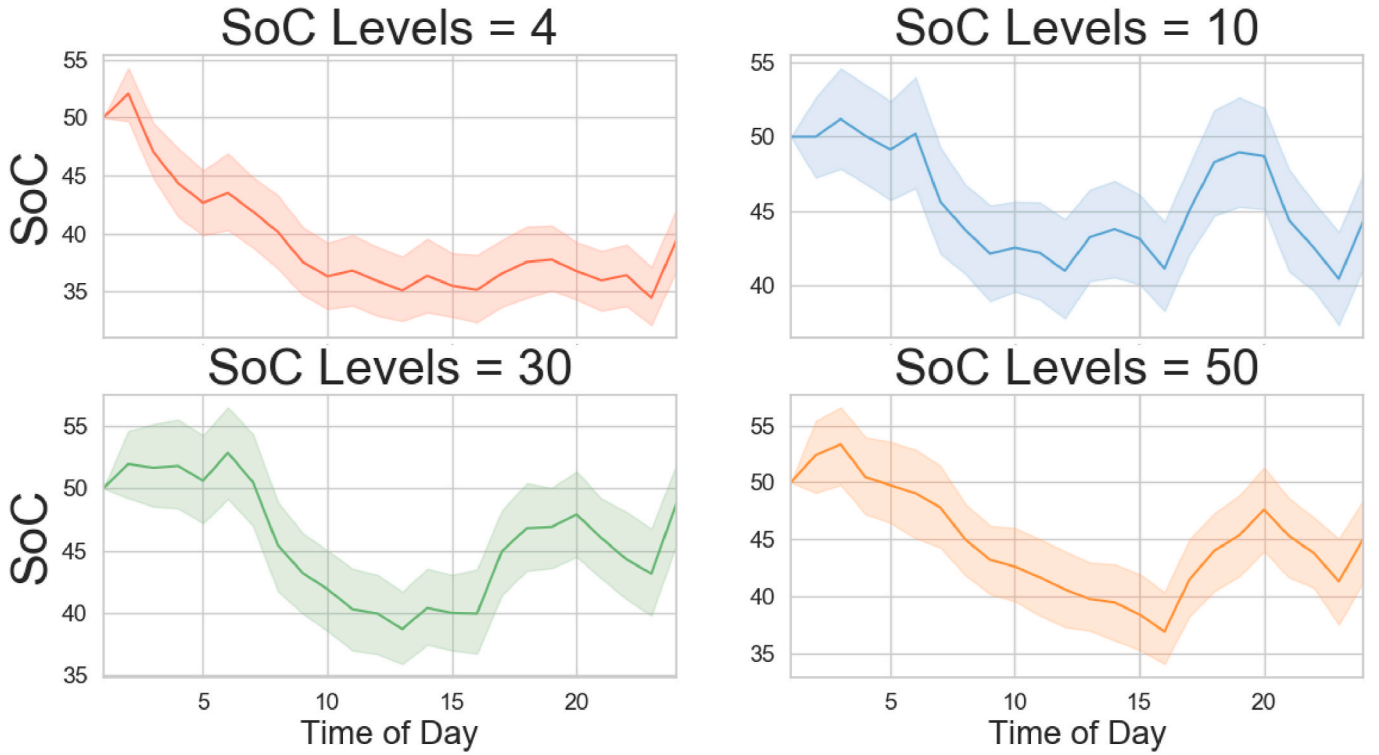


Fig. 17. SOC variation for different action level discretization levels for TD learning during different time of the day for 100 Monte Carlo runs.

from s_t to s'_t , the Q -table is updated as shown in (15). The value $\delta_{(s_t, a_t)}$ in (15) depends on the state-action pair (s_t, a_t) . For the temporal difference, it is defined as:

$$\delta_{(s_t, a_t)} = r_t + \gamma \max_{a_t} Q(s', a_{t+1}) - Q(s_t, a_t). \quad (12)$$

The convergence of (12) depends on the learning rate α as given in Refs. [45,46]. It follows the stochastic approximation principle of a convergent series as:

$$\sum_t \alpha_t(s_t, a_t) = \infty, \sum_t \alpha_t^2(s_t, a_t) < \infty. \quad (13)$$

The algorithmic convergence also depends on the number of times the state-action pair is visited $M(s_t, a_t)$, which is updated every time step as in (16). The relation of the step size with $M(s_t, a_t)$ as adapted from Ref. [46] is given as:

$$\alpha_t(s_t, a_t) = 1 / M(s_t, a_t). \quad (14)$$

Algorithm 1. Off-policy RL Algorithm based on Value Prediction

```

Initialize:  $Q(s, a)$ 
for episode = 1, N do
  Initialize initial state  $s_1$ 
   $s_1 = [C_s, SOC_{in}, P_{Net}(1)]$ 
  for t = 1, T do:
    Choose  $a_t$  using  $\epsilon$  greedy policy
    Observe: next state  $s'_t$  as per (1) and the data set.
    Reward  $r_t(s_t, a_t)$  as per (9) or (10).
    Update  $Q$ -table as per the equation
     $Q(s_t, a_t) = Q(s_t, a_t) + \alpha_{M(s_t, a_t)} \delta_{(s_t, a_t)} e_{(s_t, a_t)}$  (15)
     $M(s_t, a_t) = M(s_t, a_t) + 1$  (16)
     $s_t \leftarrow s'_t$ 
  end
end

```

The relation in (16), reduces the agent learning rate from the optimum Q -values for the next state s'_t as the state-action pair (s_t, a_t) is visited more often making the algorithm converging to an optimal

policy. The detailed mathematical proof of this is given in Refs. [45,46]. The initial value of α is set to 0.5. The variable $e_{(s_t, a_t)}$ defined in (15) is considered to be 1, in the case of temporal difference learning, since the decision is based on the reward of a single action. The value of γ is set to 0.95, which is standard in contemporary RL literature [3].

The update equation for value iteration Q -learning is based on the maximum Q -value of the next state s'_t [18]. The temporal difference target $\delta_{(s_t, a_t)}$ is defined as:

$$\delta_{(s_t, a_t)} = r_t + \gamma \max_a Q(s', a_t). \quad (17)$$

Since value iteration is a variant of temporal difference learning, the values of γ and $e_{(s_t, a_t)}$ are retained as previously defined. When future rewards are incorporated using n -step TD prediction, the learning process alternates between using average rewards in some steps and estimated rewards in others. The number of prediction steps depends on the value of λ . The TD learning deployed that incorporates future rewards is called TD(λ) prediction. However, the rewards obtained from previous states can be considered as future rewards as given in Ref. [44]. Hence it is formulated as an algorithm in the form of eligibility traces. However repeated trace algorithm in Ref. [47], performs better than the eligibility trace. Here, the variable $e_{(s_t, a_t)}$ is defined as:

$$e_t(s_t, a_t) = \begin{cases} 1 + \gamma \lambda e_{t-1}(s_t, a_t) & \text{if } s = s_t \text{ and } a = a_t; \\ 0 & \text{if } s = s_t \text{ and } a \neq a_t \quad \forall s, a; \\ \gamma \lambda e_{t-1}(s_t, a_t) & \text{if } s \neq s_t. \end{cases} \quad (18)$$

The number of times a state is visited has been captured already in $e_{(s_t, a_t)}$. Hence, the value of λ is set to 0.5 as it provides a balance between Monte Carlo simulation-based control and single-step TD learning. The general structure of the off-policy value prediction has been shown in Algorithm 1. In addition to the RL techniques derived from dynamic programming, Q -learning derived from the adaptive ADP is described in the next section.

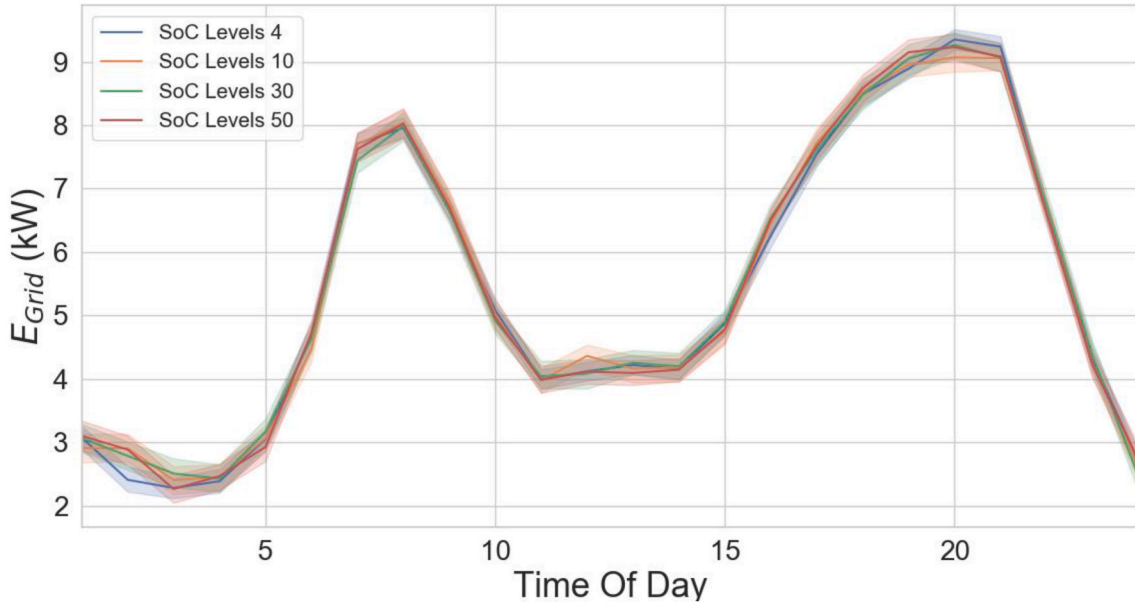


Fig. 18. The power imported from the grid at different SOC discretization levels for 100 Monte Carlo runs for temporal difference learning.

4.2. Off-policy adaptive Q-learning

Algorithm 2 represents an off-policy adaptive Q-learning, where the update equations (21) and (23) depend on the actions taken from the next state s'_t . Hence, the Q-table must be fully initialized for all state-action-pairs (s_t, a_t) , making the learning process deterministic. An auxiliary table is used to update the main Q-table, hence accelerating the convergence of the algorithm. The algorithm also tunes the adaptive parameter for policy evaluation, enabling a trade-off between the current and future Q-function. The policies derived from the actions are not updated within the same episode. This adaptive Q-learning procedure is adapted from the algorithm described in Ref. [35] and applied to energy management in microgrids.

Algorithm 2. Adaptive Q-Learning

```

Initialize:  $Q^0(s, a), Z^0(s, a)$ 
for j = 1, N do
  Initialize initial state  $s_1$ ,
   $s_1 = [C_{s_1}, SOC_{in}, P_{Net}(1)]$ 
  for t = 1, T do:
     $a_t = \text{argmax}_a Q^{(j-1)}(s_t, a)$  (19)
    Observe: next state  $s'_t$  as per (1) and the data set.
    Reward  $r_t(s_t, a_t)$  as per (9) and (10).
     $a'_t = \text{argmax}_a Q^{(j-1)}(s'_t, a)$  (20)
    Update Z-table as per the equation
     $Z_t^{(j-1)}(s_t, a_t) = r(s_t, a_t) + \alpha_j Z_t^{(j-1)}(s'_t, a'_t) + (1 - \alpha_j) Q^{(j-1)}(s'_t, a'_t)$  (21)
     $s_t \leftarrow s'_t$ 
  end
   $Q^{(j)}(s_t, a_t) = Z_T^{(j-1)}(s, a)$  (22)
  for t = 1, T do:
     $Q^{(j)}(s_t, a_t) = r(s_t, a_t) + \alpha_j Q^{(j)}(s'_t, a'_t) + (1 - \alpha_j) Q^{(j-1)}(s'_t, a'_t)$  (23)
  end
   $Z^{(j)}(s, a) \leftarrow Q^{(j)}(s_t, a_t)$ 
  j ← j + 1
end

```

As seen in Algorithm 2, the actions are taken based on the maximum Q-value obtained from (19). Thus, the agent does not explore the input action space as described in the previous off-policy Q-learning procedure in (11). The Q-update procedure is hierarchical in this method, where one form of the update process is performed within the episode, using $Z(s, a)$ as shown in (21). However, convergence proof has not yet been established for the custom microgrid model. Therefore, the

reassignment of the intermediate variable is performed using the theory presented in Ref. [35]. Applying Theorem 1.4. in Ref. [35] we get,

$$\lim_{t \rightarrow T} Z^{(t)}(s_t, a_t) = Q^{(j)}. \quad (24)$$

Thus the update equation defined in (21), uses the Q-values from the previous episode $Q^{(j-1)}$ to initialize the Q-value for its update as given in (22). The $Z_T^{(j-1)}(s, a)$ obtained at the final step is used for initialization. The Q-value is further updated using the equation in (23). The updated Q-value becomes the initial value of $Z(s, a)$ in the next iteration. As per Theorem 2.2 in Ref. [35], the value of $Q^{(j)}$ converges to:

$$\lim_{i \rightarrow \infty} Q^{(i)}(s, a) = Q^*(s, a). \quad (25)$$

Hence, the stopping criteria $\|Q^{(i+1)}(s, a) - Q^{(i)}(s, a)\|_{\infty}$ is to be used as the criterion of convergence. As per Corollary 1 in Ref. [35], the step size increases with the episode for the convergence of the optimal policy. Hence, it is defined as:

$$\alpha_j = \tanh(b \ln(j + 1)). \quad (26)$$

Here, j represents the episode number, since the number of episodes considered is 1000 to benchmark with the algorithm, described in the previous subsection. Therefore, the value of b considered here is set to 0.005, so that α_j approaches 1 with the episodes.

5. Results and discussions

The RL algorithms described in the previous section are implemented on the microgrid model shown in Fig. 1, using daily load demand and solar PV dataset. All simulations are carried out in MATLAB. The initialization of the Q-table for the off-policy RL algorithm as described in Algorithm 1, is based on initial random action. Then, Monte Carlo simulations were performed 100 times based on the rewards of the different initial random actions selected in every simulation. The simulation of Q-learning algorithm is performed for 1000 episodes, and the rewards and the parameters of the environment are analyzed accordingly. A single day simulation constitutes one episode, and the rewards are accumulated over several steps. The simulation is continued over several episodes until convergence in episodic rewards is achieved. The values corresponding to the state-action pair in the updated Q-table can be used to take the actions in real time.

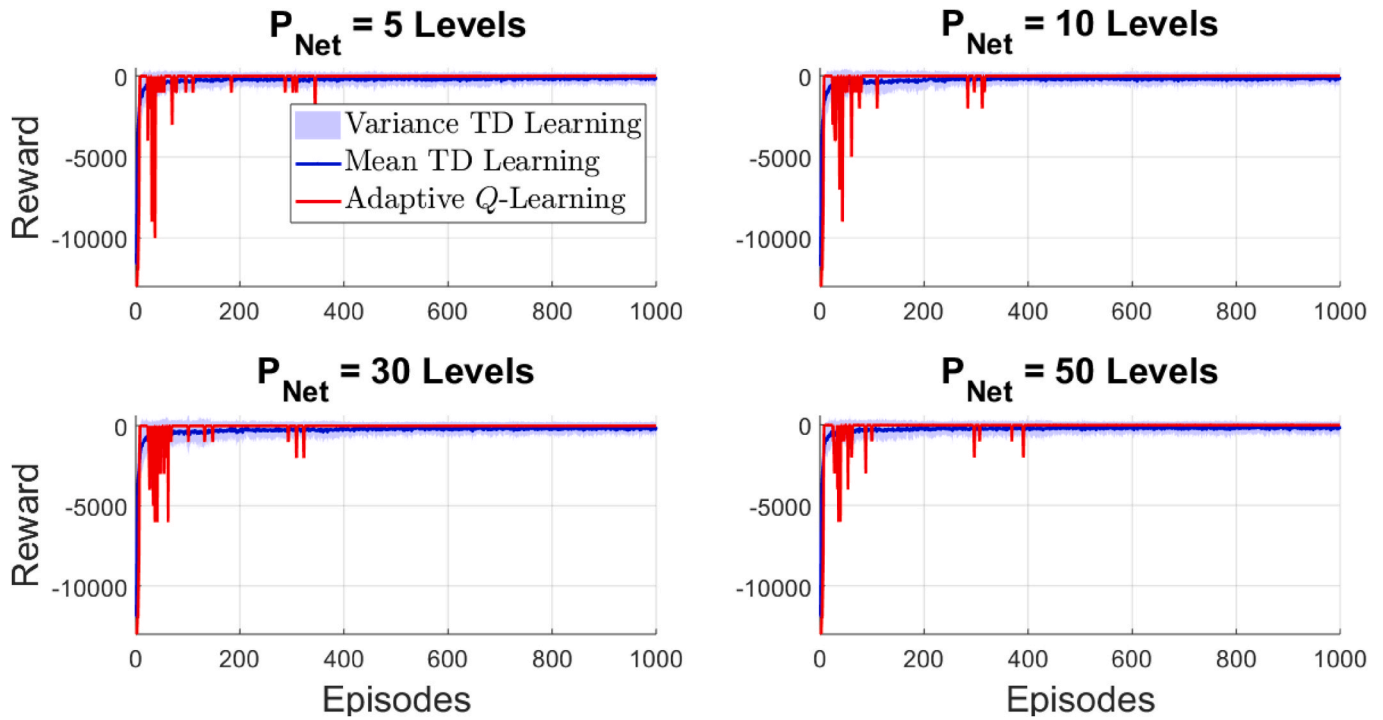


Fig. 19. Reward plots for different exogenous state discretization levels for temporal difference (for 100 Monte Carlo runs) and adaptive Q-learning.

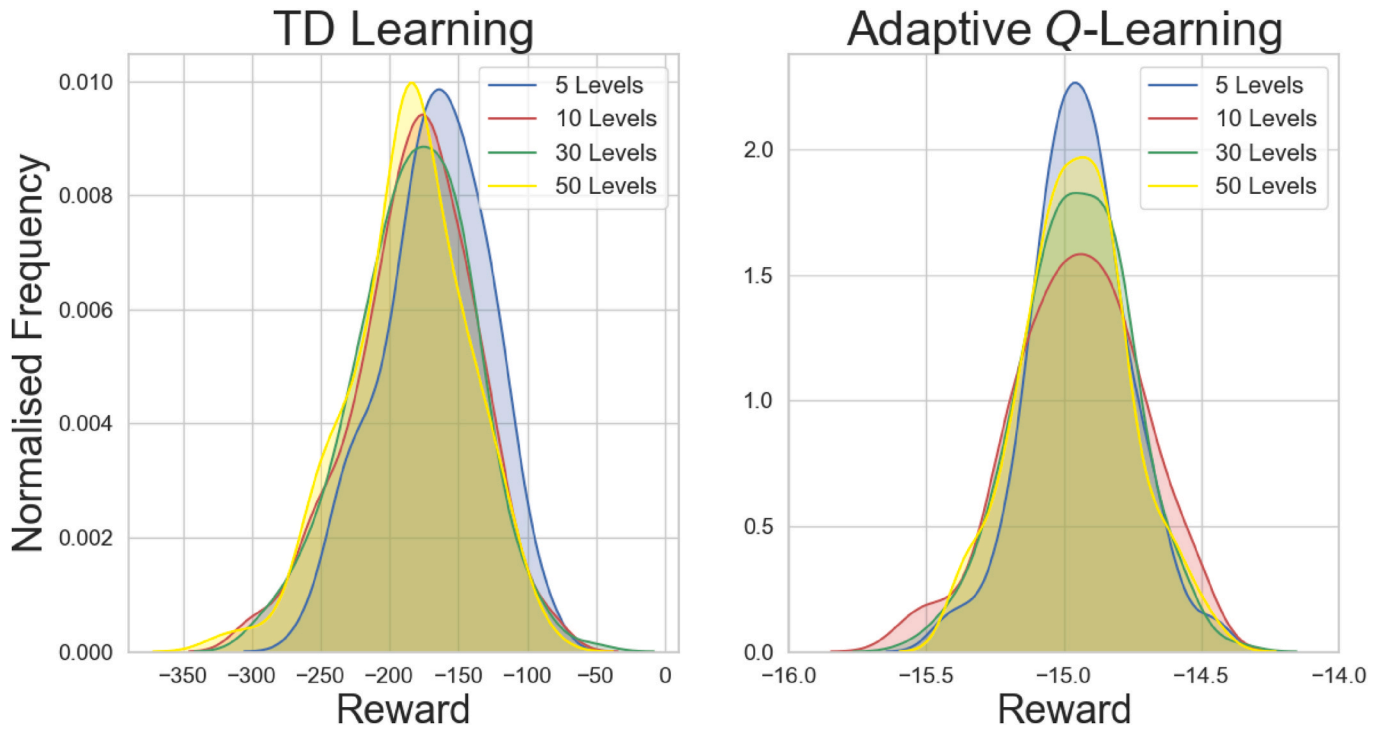


Fig. 20. Histograms of the reward for the last 200 episodes for different exogenous state discretization level for temporal difference and adaptive Q-learning algorithm considering the same initial Q-table initialization.

The initial state of charge for every episode is set to 50 %. The value of the converged rewards and the relationship between the power drawn from the grid and the variation of SOC during the day is analyzed using univariate and multivariate KDE plots. These plots are created using the function `kdeplot()` in the Seaborn library in Python [47]. The univariate plot of the reward is considered after 800 episodes which is equivalent to visualization of the posterior samples after a burn-in period for Markov

Chain Monte Carlo (MCMC) family of algorithms. The kernel density is estimated at default parameters of the function where the grid-size is 100, with Gaussian kernel and Scott method of determining the kernel size, scalar factor, and scale for each dimension. Since the number of samples in the case of adaptive Q-learning is small, the bivariate plots are generated by interpolating between the samples using the `interp()` function in MATLAB with the cubic-spline method. The rewards for all

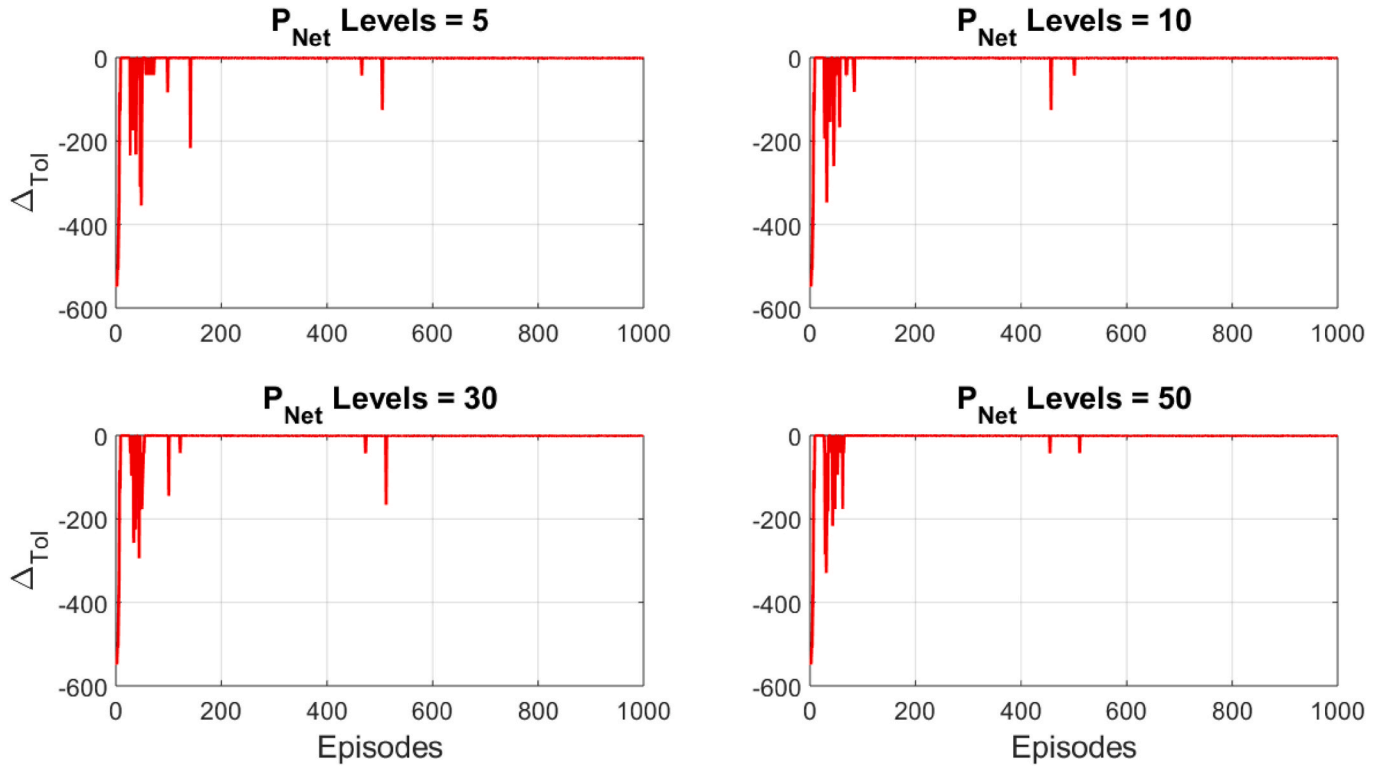


Fig. 21. Convergence characteristics of the adaptive Q-learning for different exogeneous state level discretization.

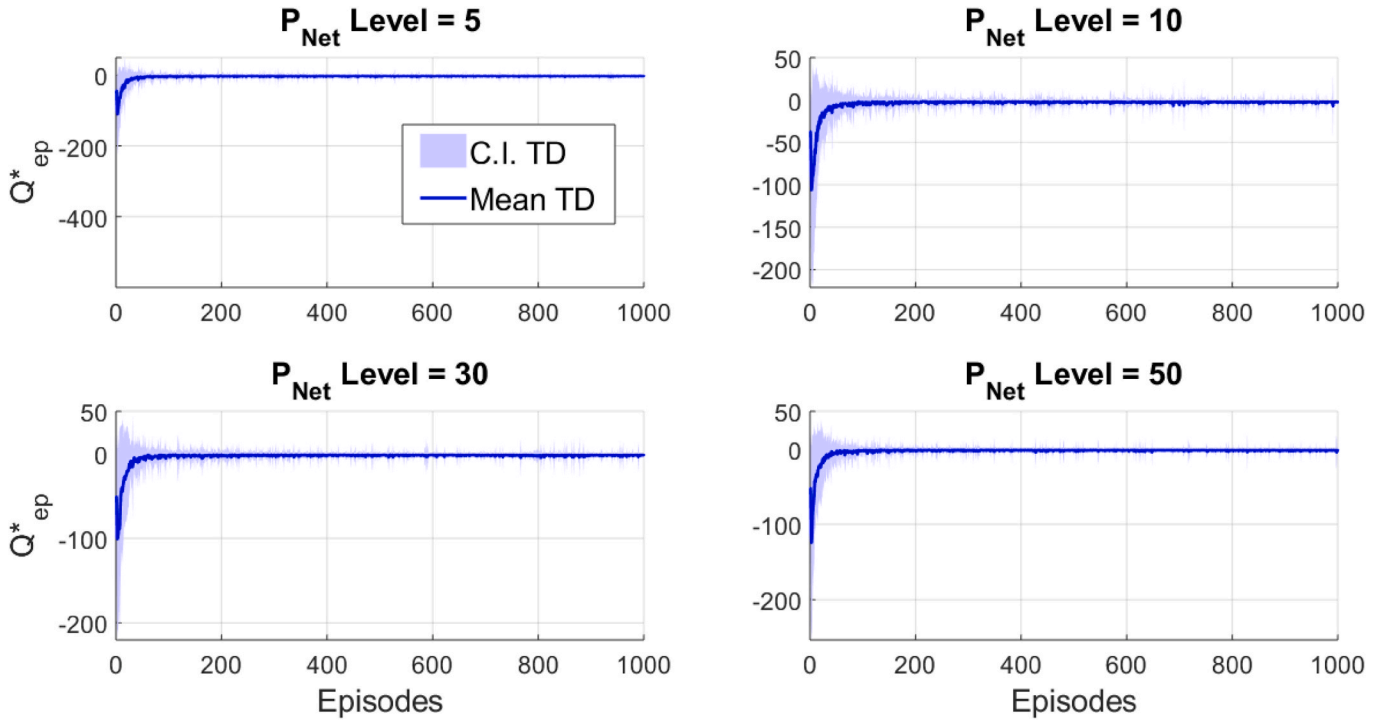


Fig. 22. Convergence characteristics of the TD-learning algorithm for different exogeneous state discretization levels.

the on-policy RL algorithms are analyzed, considering the samples obtained from the Monte Carlo simulation runs on different Q-table initialization, where the mean and the confidence intervals of one standard deviation around the mean are plotted. The univariate KDE plot for the same is plotted, considering the mean variation of the rewards, obtained for all the episodes.

The operation of the battery considering the variation of its SOC and the power imported from the grid, is analyzed with the same Q-table initialization for different parametric study under consideration. The adaptive Q-learning algorithm is deterministic as per the Q-table initialization and the action selection in (19) and (20). Therefore, we obtain a single plot for the reward when the kernel density is estimated.

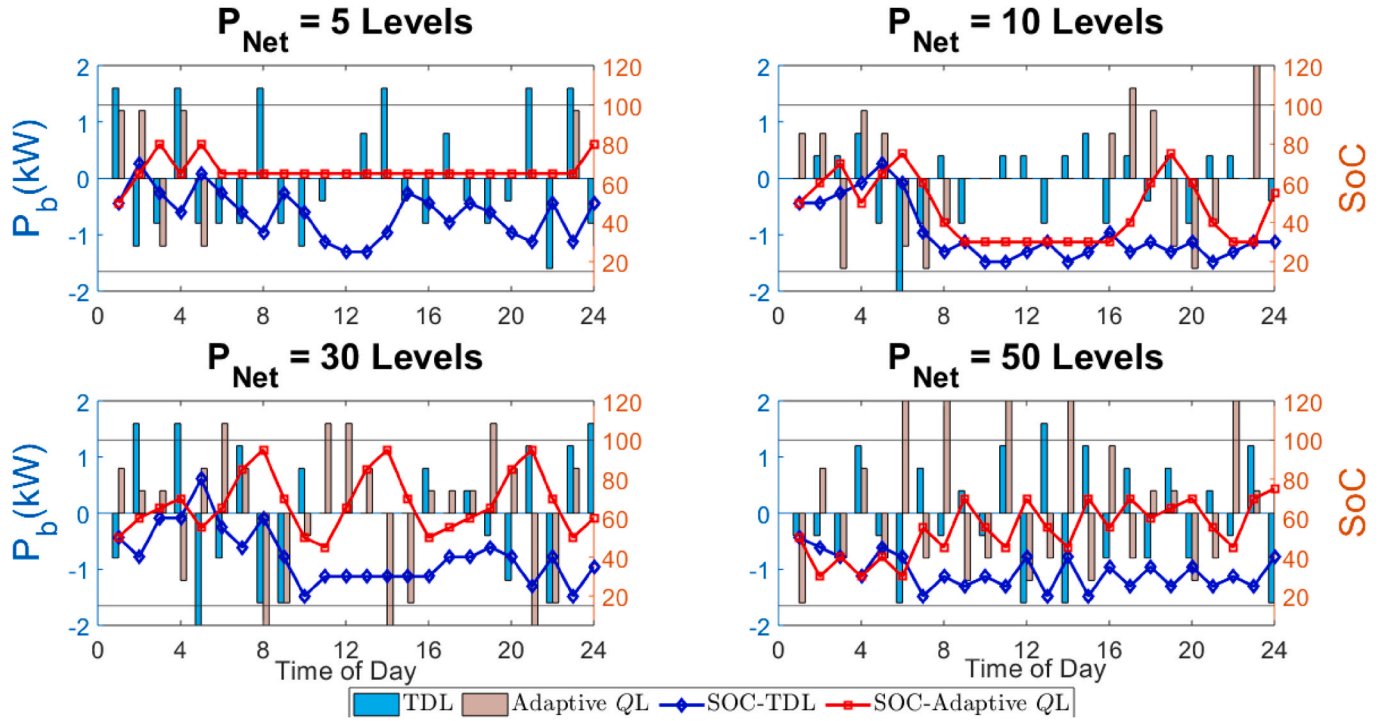


Fig. 23. Battery power at different exogenous state discretization level along with the SoC variation for TD learning and adaptive Q-learning considering the same initial Q-table initialization in the last episode.

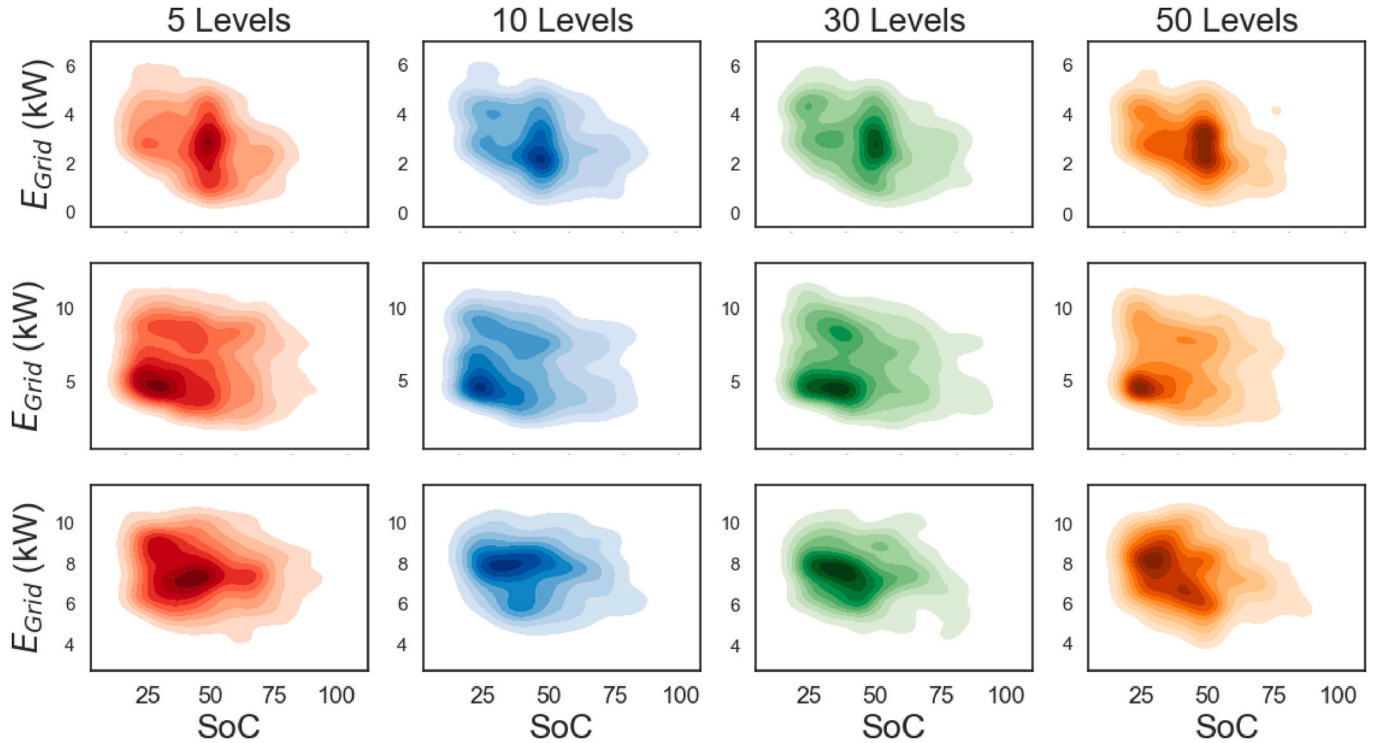


Fig. 24. Bivariate density plots of the power drawn from the grid along with the SoC level variation for all the Monte Carlo simulation at different exogenous state discretization during different tariff periods. Row 1: off-peak, row 2: mid peak, row 3: peak.

The nominal simulation is conducted considering 11 action levels and 10 states of the controllable and exogeneous states discretization levels for comparing the performance of algorithms at different exploratory policies and decision intervals.

5.1. Analysis at different discretization levels

The RL algorithm has been analyzed at different discretization levels of the actions and the state space. The analysis is conducted by keeping one parameter constant and varying the discretization level of the other

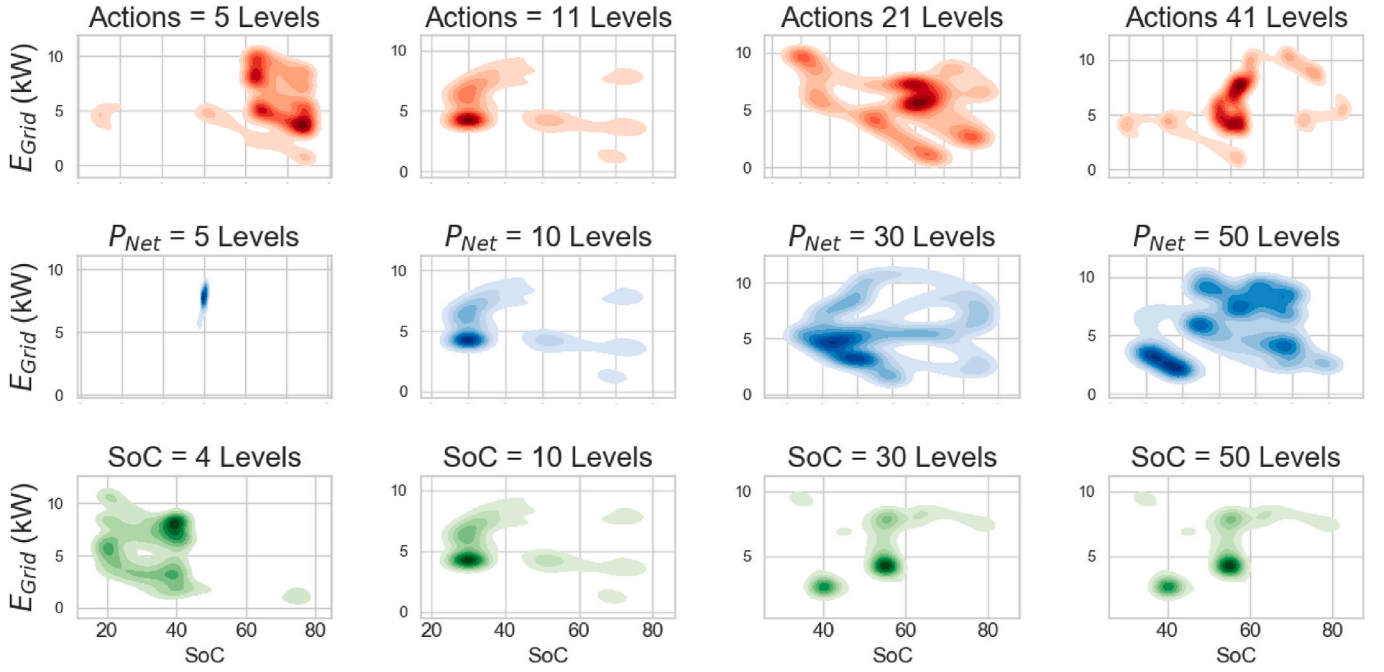


Fig. 25. Bivariate density plots of the power drawn from the grid and SoC variations for adaptive Q-learning algorithm at different discretization levels.

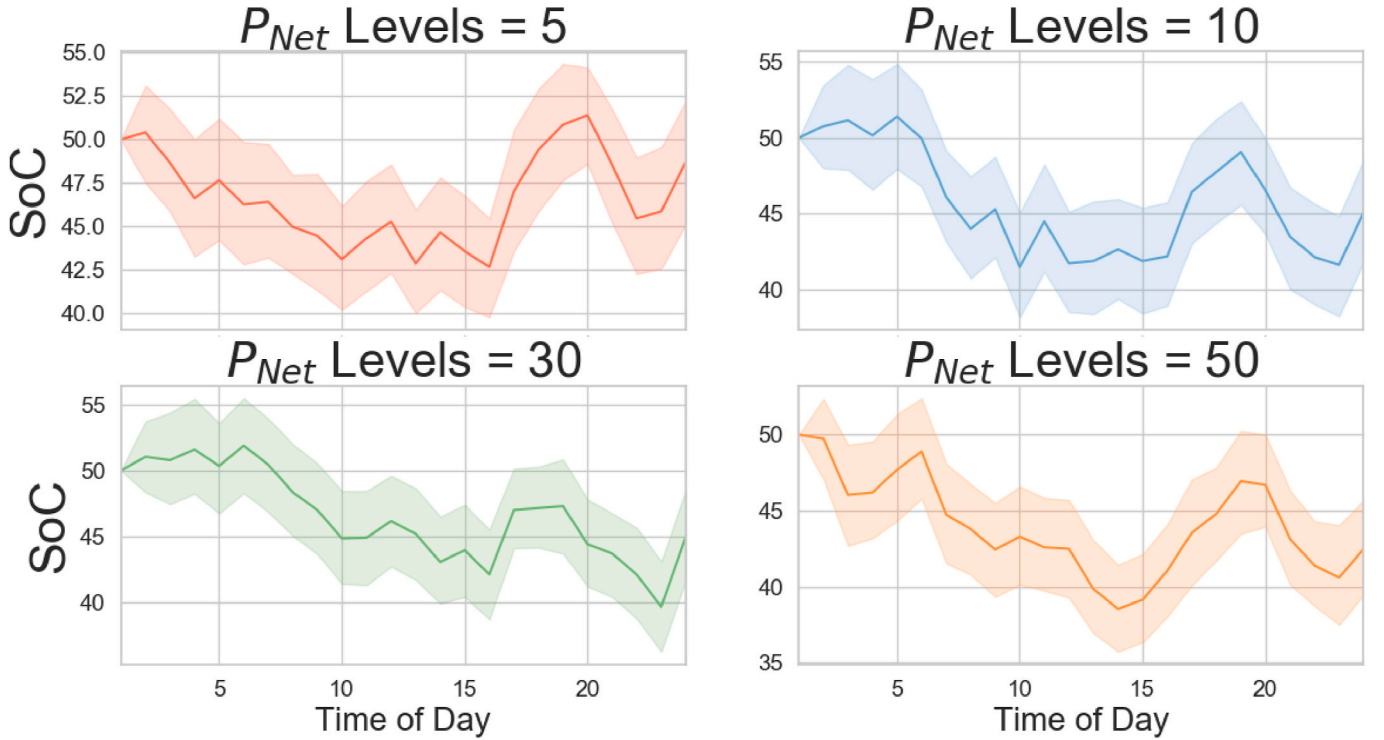


Fig. 26. Variation of SOC throughout the day for different exogenous state discretization levels for 100 Monte Carlo runs.

state variable. The value of ϵ in the greedy algorithm for the value iteration is set to 0.05, which is standard in many recent RL literatures.

5.1.1. Effect of action discretization

In this simulation, the action space discretization level is varied, while the discretization levels of the controllable and exogenous states are kept fixed. It is observed from Fig. 3 that the reward variation increases with random initialization of the Q-table with the increase in the number of decisions in the action space. Moreover, it is also evident that

the convergence time is faster in the case of smaller action space. In adaptive Q-learning, an increase in the size of the action space leads to the selection of more sub-optimal actions. As a result, penalties may still be observed even after the algorithm has converged. The higher initial spikes observed before convergence suggest that the battery operates mostly at extreme limits during the initial episodes.

The univariate reward distribution shown in Fig. 4 suggests that the converged rewards for the Adaptive Q-learning algorithm are better than those of the TD. The univariate mean in the case of adaptive Q-

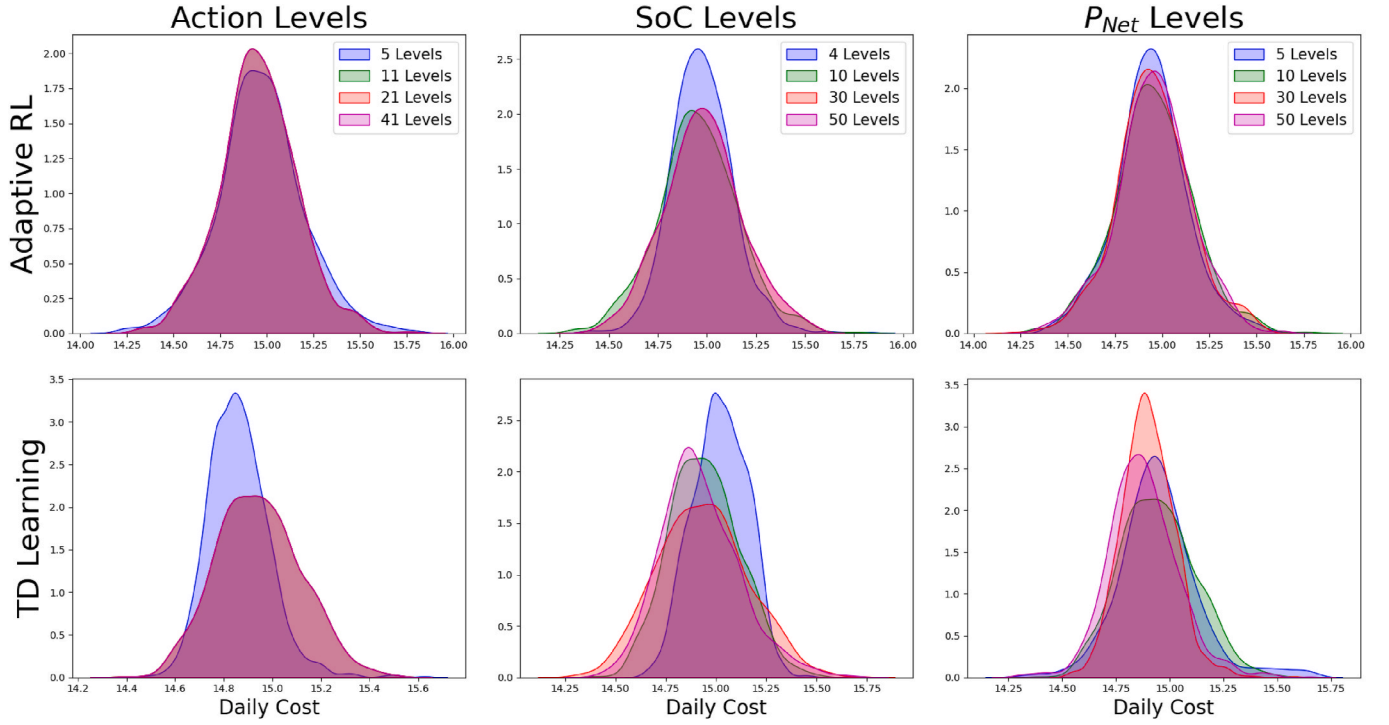


Fig. 27. Comparison of the daily episodic cost (in pounds) for adaptive RL and TD learning for different action level, SoC level and P_{Net} level discretization.

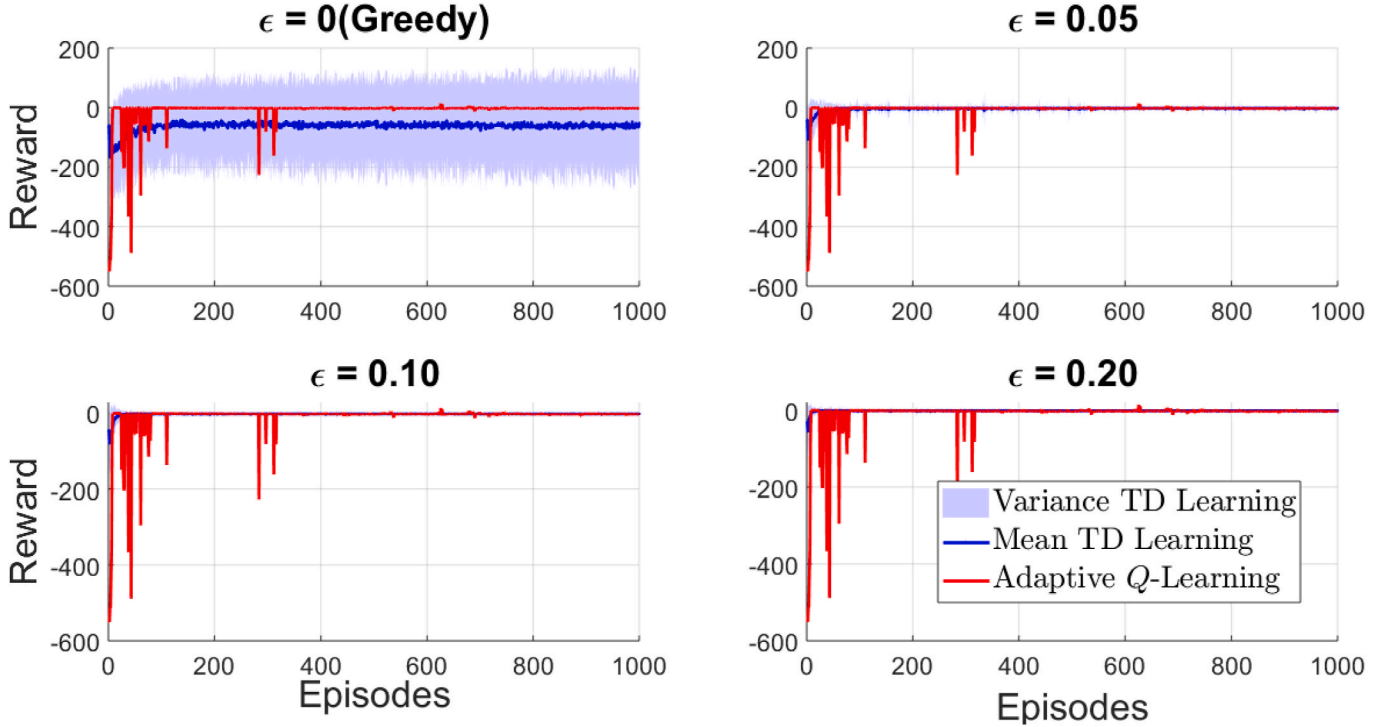


Fig. 28. Reward plots for different exploration parameter levels for temporal difference (from 100 Monte Carlo runs) and adaptive Q-learning.

learning is much lower than the one observed in TD learning. Moreover, in the case of TD-learning, the absolute amount of the rewards is less than the penalty defined in (10). This is because the penalty is induced for some initialization of Q -values. Hence, the average value becomes lower than the specified one in (10), when the overall rewards average. It also suggests that the deterministic algorithm is more likely to operate within the given battery SOC limits compared to stochastic algorithms. It

is mainly due to the maximization step, which restricts the agent's exploration of the action space.

In Fig. 5, it is observed that the battery operation shows greater oscillations or extended idle periods in the case of adaptive Q-learning process. This can be detrimental to long-term battery life due to the oscillatory nature of battery actions at lower action levels.

The reward for higher action states in the case of TD learning is lower

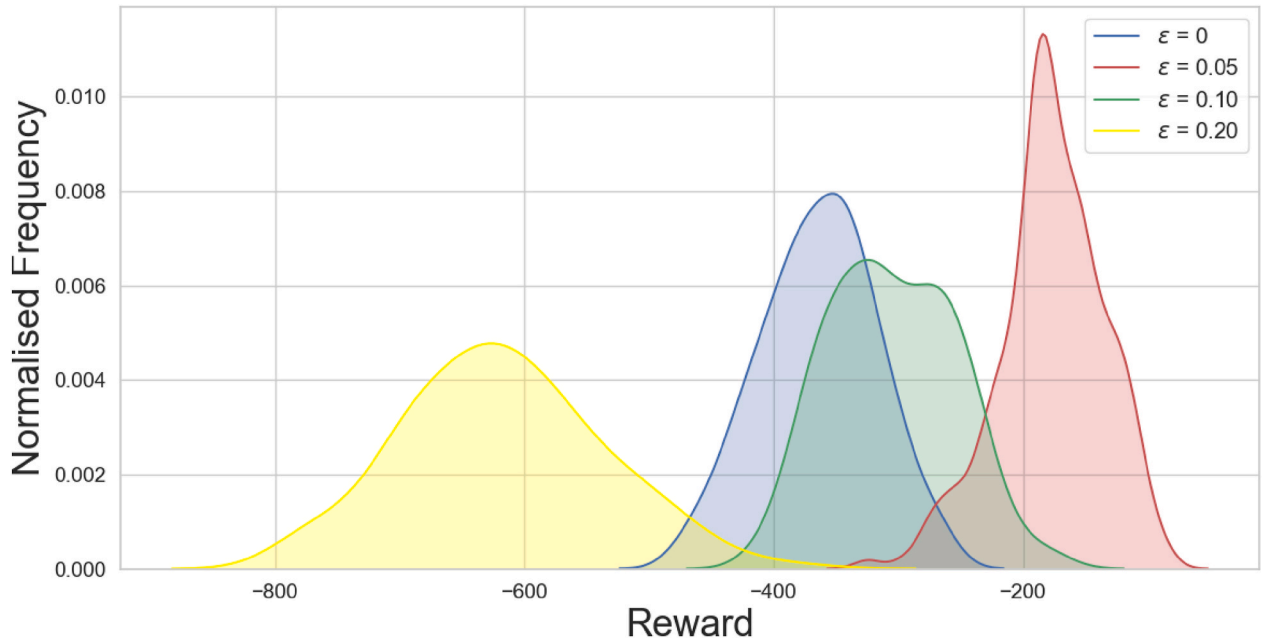


Fig. 29. Histograms of the reward for the last 200 episodes for different exploration parameters considering same initial Q-table initialization.

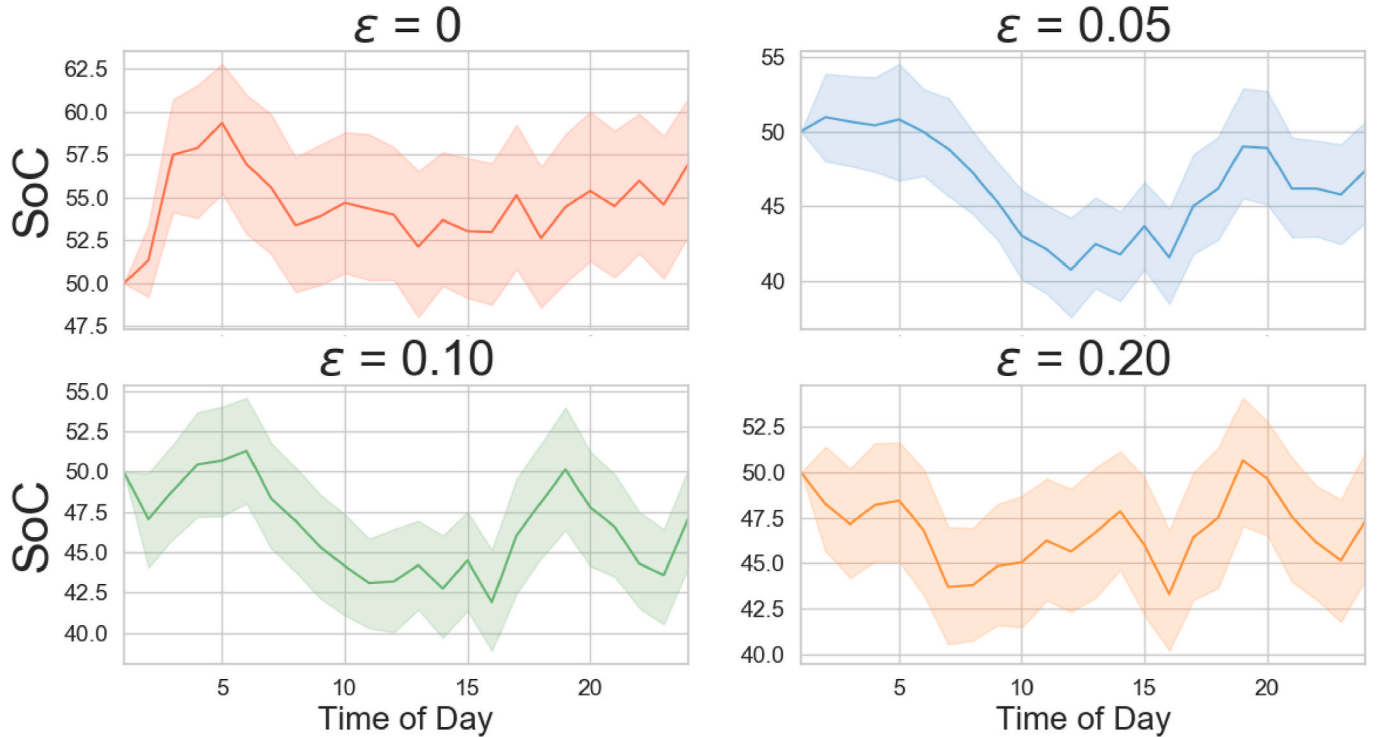


Fig. 30. SOC variation for different exploration parameters in TD learning during different time of the day for 100 Monte Carlo runs.

compared to other cases. This suggests that higher action discretization can cause the agent to take more sub-optimal decisions. When the action discretization level is low, the battery SOC in the case of adaptive Q-learning tends to oscillate during periods of high solar PV generation throughout the day. In contrast, in the case of TD learning, SOC tends to stay mostly idle for extended periods, indicating limited battery utilization. A similar phenomenon is observed when the action space is discretized to 11 levels, where the battery is mostly utilized when the solar PV generation is high. However, the spike in SOC is observed during peak period, which is due to the suboptimal actions taken in that

period. However, the ideal usage of the battery is observed at 21 action levels where the battery is charged during the low-price period of the day, charging and discharging during the PV period and discharged during the higher peak period. However, when the action discretization level is even higher, the battery operates primarily at high SOC mode taking up more power from the grid. This also explains the slow convergence of the rewards.

Bivariate density plot of the power imported from the grid and the SOC variation for Monte Carlo simulation is shown in Fig. 6 for off-peak, mid-peak and peak periods. It is observed that the spread in SOC

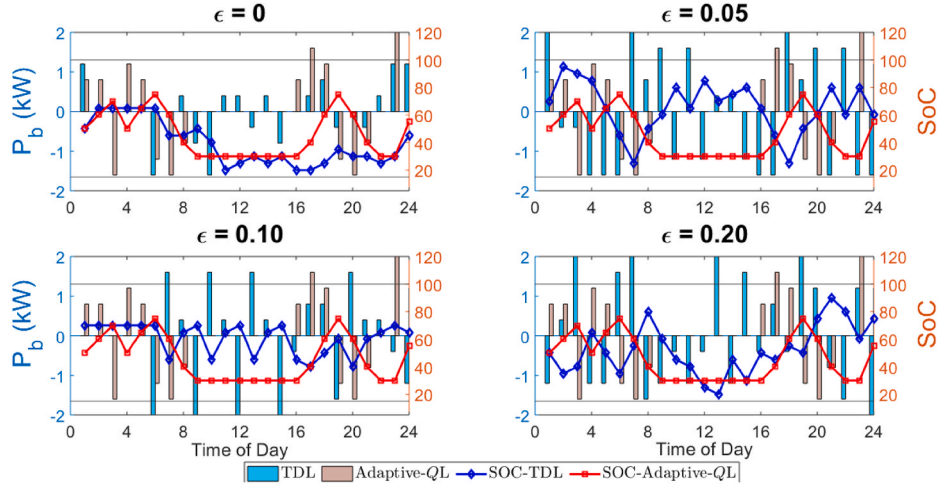


Fig. 31. Battery power at different exogenous state discretization level along with the SoC variation for TD learning and adaptive Q-learning for the same Q-table initialization for the last episode.

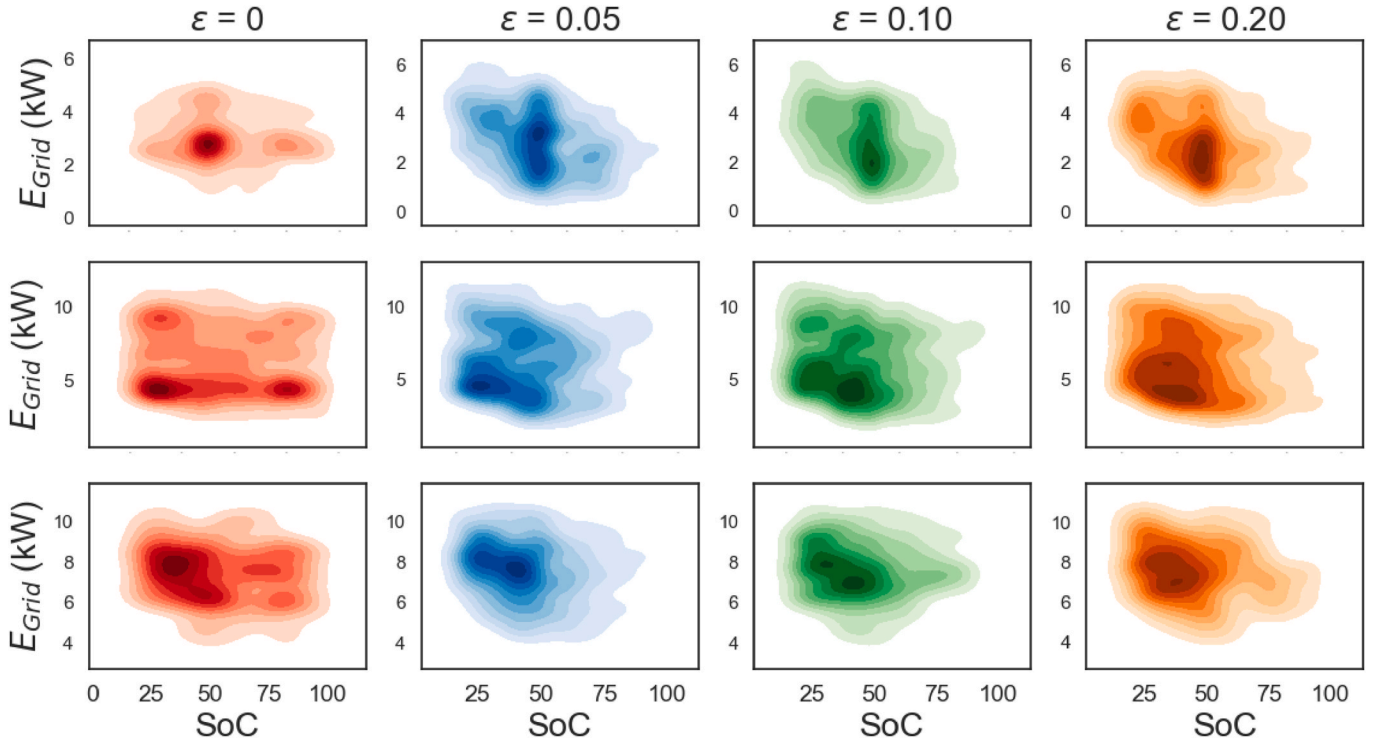


Fig. 32. Bivariate density plots of the power drawn from the main grid along with the SoC level variation for all the Monte Carlo simulation at different exploration parameters, during different tariff periods. Row 1: off-peak, row 2: mid peak, row 3: peak.

becomes more significant with higher action discretization levels. This implies that lower action discretization level results in a narrower range of SOC variation. Beside the convergence characteristics of the reward in Fig. 3, the convergence of other variables are shown in Figs. 7 and 8, for four levels of action discretization.

Similarly, we also observe the variation in SOC with change in action discretization levels in Fig. 9, and correspondingly the power imported from the grid in Fig. 10.

5.1.2. Effect of controllable state discretization

Next, the controllable state discretization space is varied while keeping the number of exogenous states and action space constant. The variation of rewards is shown in Fig. 11. The convergence characteristics of the tolerance and optimal Q-value is shown in Figs. 15 and 16

respectively for different SOC discretization levels. Unlike the pattern observed in the previous case, higher discretization causes a fewer violations of the SOC limits. The violation is due to the inefficient approximation of the SOC to the nearest bin. Hence, it causes the agent to take sub-optimal actions inducing penalties for the agent, which is not observed in the coarse discretization levels. This trend is also visible in the rewards for adaptive Q-learning. It is observed in Fig. 12 that the average reward is better at the discretization level of 10 in the case of TD learning, which implies that the optimum performance is obtained at a specific SOC discretization level, as lower discretization causes poor approximation of the state. In contrast, a higher discretization level causes over-approximation of the state.

In the case of adaptive Q-learning algorithm, the performance remains similar at high discretization levels. Hence, similar rewards are

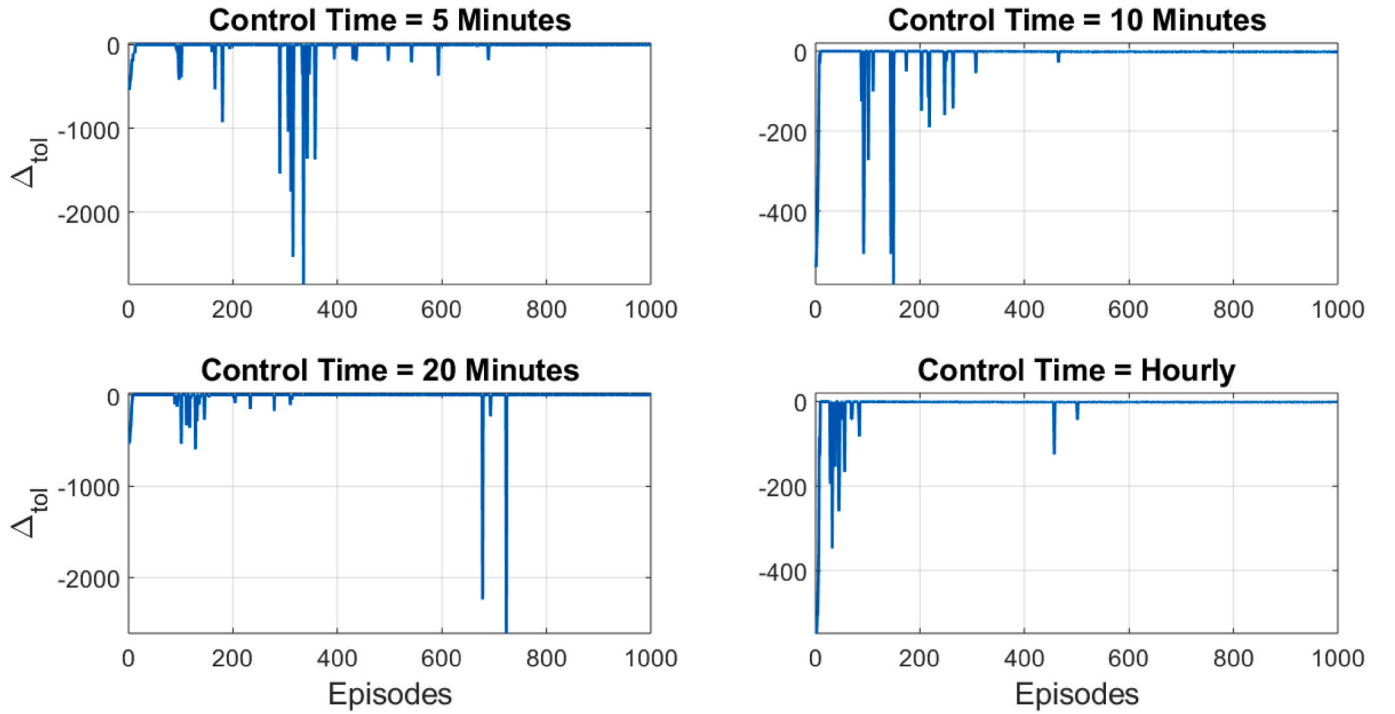


Fig. 33. Convergence characteristics of adaptive Q-learning for different episodes for different control time iteration.

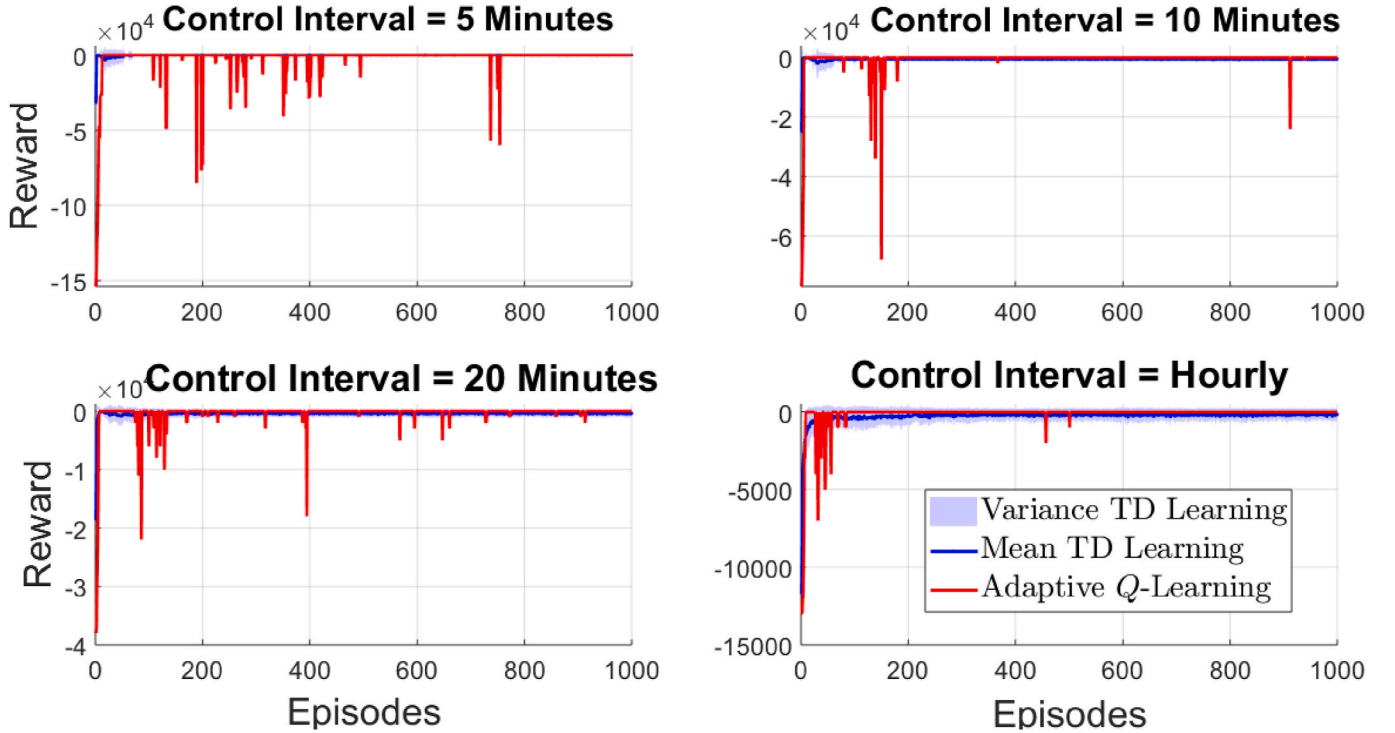


Fig. 34. Reward plot for different control interval for temporal difference (for 100 Monte Carlo runs) and adaptive Q-learning.

observed for 30 and 50 discretization levels. This is a result of optimal actions defined in (19) as there are no variation in the rewards with the increase in discretization states. However, in the case of TD-learning, we observe slight variation in the rewards with the increase in the discretization levels.

In the SOC plots, shown in Fig. 13, for adaptive Q-learning algorithm, the response is oscillatory in the duration when PV generation is high,

but with the increase in discretization level, the battery remains idle most of the time. However, as the discretization level increases the SOC space is explored more by the TD learning as compared to the adaptive Q-learning algorithm. At higher discretization levels, the agent follows the typical behaviour, i.e. charging during the off-peak period and discharging during the peak period. From the bivariate plots in Fig. 14 obtained for TD algorithm, we observe that the SOC variation is quite

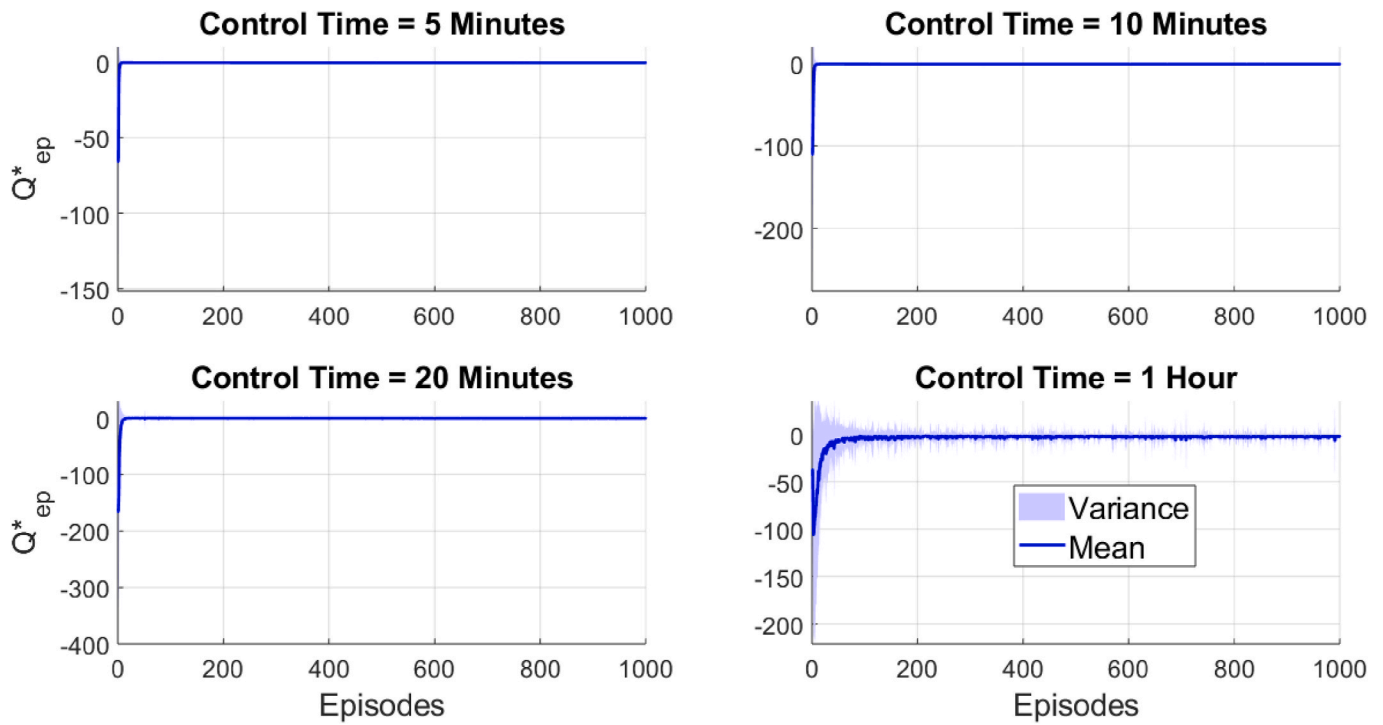


Fig. 35. Convergence characteristics of TD learning at different control decision interval for 100 Monte Carlo runs.

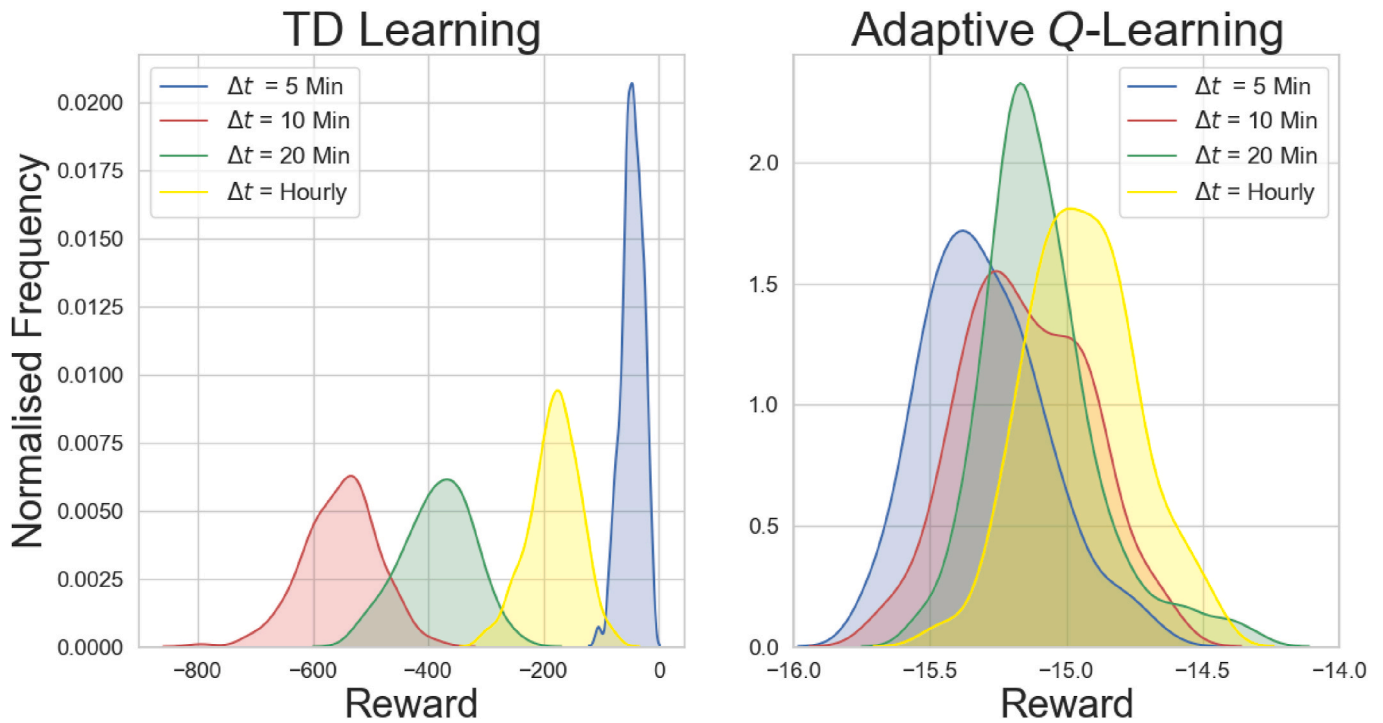


Fig. 36. Histograms of reward for last 200 episodes for different controllable state discretization levels for temporal difference/adaptive Q-learning.

high when the discretization level increases. The analysis can also be reflected by the SOC variation and the power imported from the grid in Figs. 17 and 18 for various SOC discretization levels.

5.1.3. Effect of exogenous state discretization

Next, the exogenous state discretization space is varied, keeping the number of controllable states and action space constant, and the variation of the reward is shown in Fig. 19. The convergence of the algorithm

with respect to different exogenous state discretization levels is shown in Figs. 21 and 22. Lower variation is observed for the different discretization levels of the exogenous variable. However, a decrease in reward overshoot for the adaptive Q-learning algorithm is observed with increase in the discretization levels. It signifies that there is a higher chance that the battery operates near to the SOC limits due to the inaccurate approximation of the load demand and PV generation.

The univariate density plots in Fig. 20 show slight differences in the

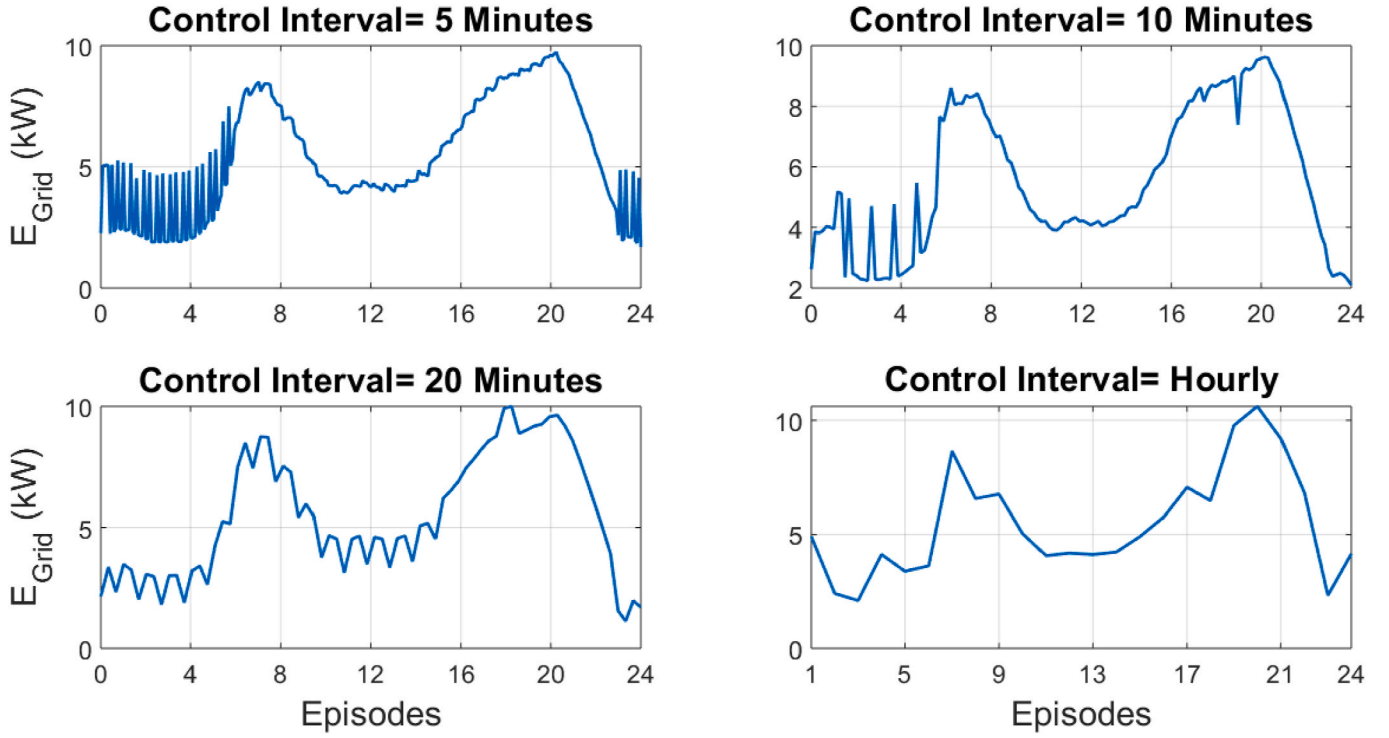


Fig. 37. Power imported from the grid at different control intervals for adaptive Q-learning.

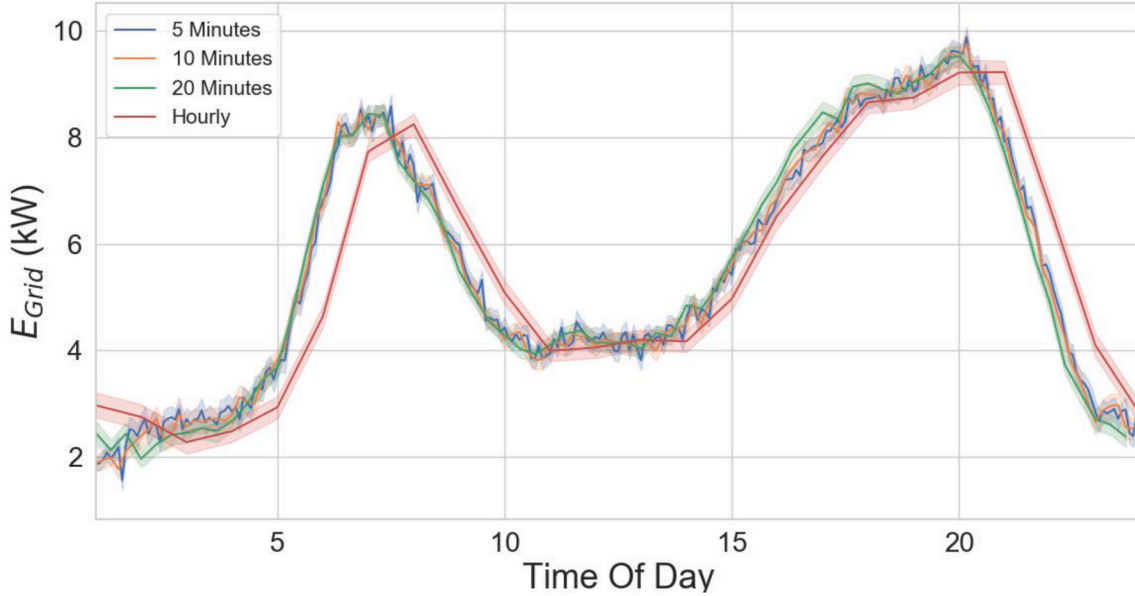


Fig. 38. Variations of the power imported from the main grid for different control time intervals for TD learning obtained from 100 Monte Carlo runs.

converged rewards across different RL parametric variations. However, for the adaptive Q-learning algorithm in the case of low state discretization, the battery operation has fewer fluctuations in battery commands as observed in Fig. 23. With further increase in the discretization levels, the operation becomes more oscillatory due to high PV generation during the day, as shown in Fig. 23. Moreover, the SOC variation in the case of TD algorithm is minimal, and the battery operates near the lower SOC limits.

In the bivariate density plot shown in Fig. 24, it is observed that more power is drawn at lower SOC which means more charging actions is preferred during the peak period of the day at lower discretization

levels. The number of samples with higher SOC decreases as the discretization level increases for all periods. However, in the case of adaptive Q-learning algorithm, the spread in SOC increases with the increase in discretization levels, as shown in Fig. 25.

The variation of the SOC with different exogenous state discretization values as shown in Fig. 26 does not affect the daily cost significantly. A comparison of the costs across different algorithms in Fig. 27 shows that the final cost remains similar regardless of the discretization levels of action and SoC. However, we observe a slightly lower cost for lower action levels in the TD learning and a slightly higher cost for the lower SoC discretization levels. Hence, we can conclude that the

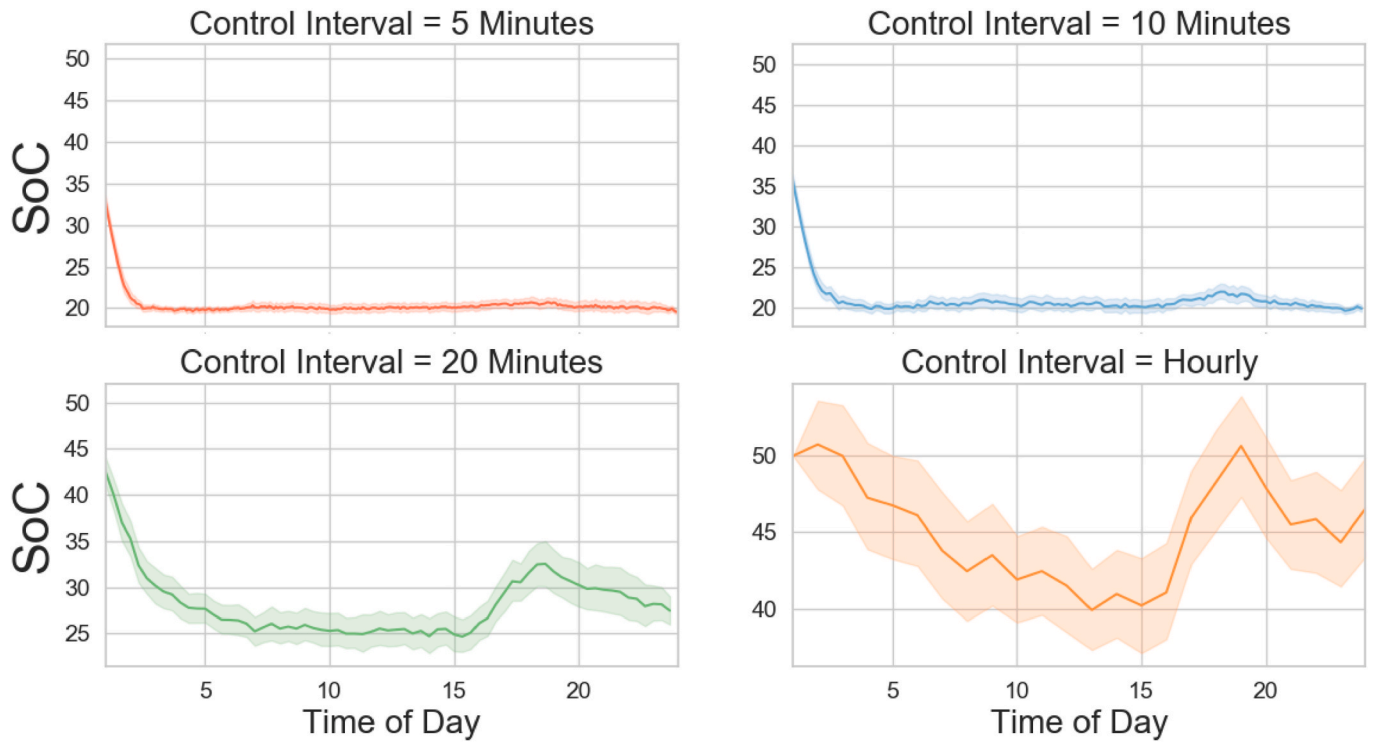


Fig. 39. Variations of SoC throughout the day in the case of TD learning, obtained from 100 Monte Carlo runs.

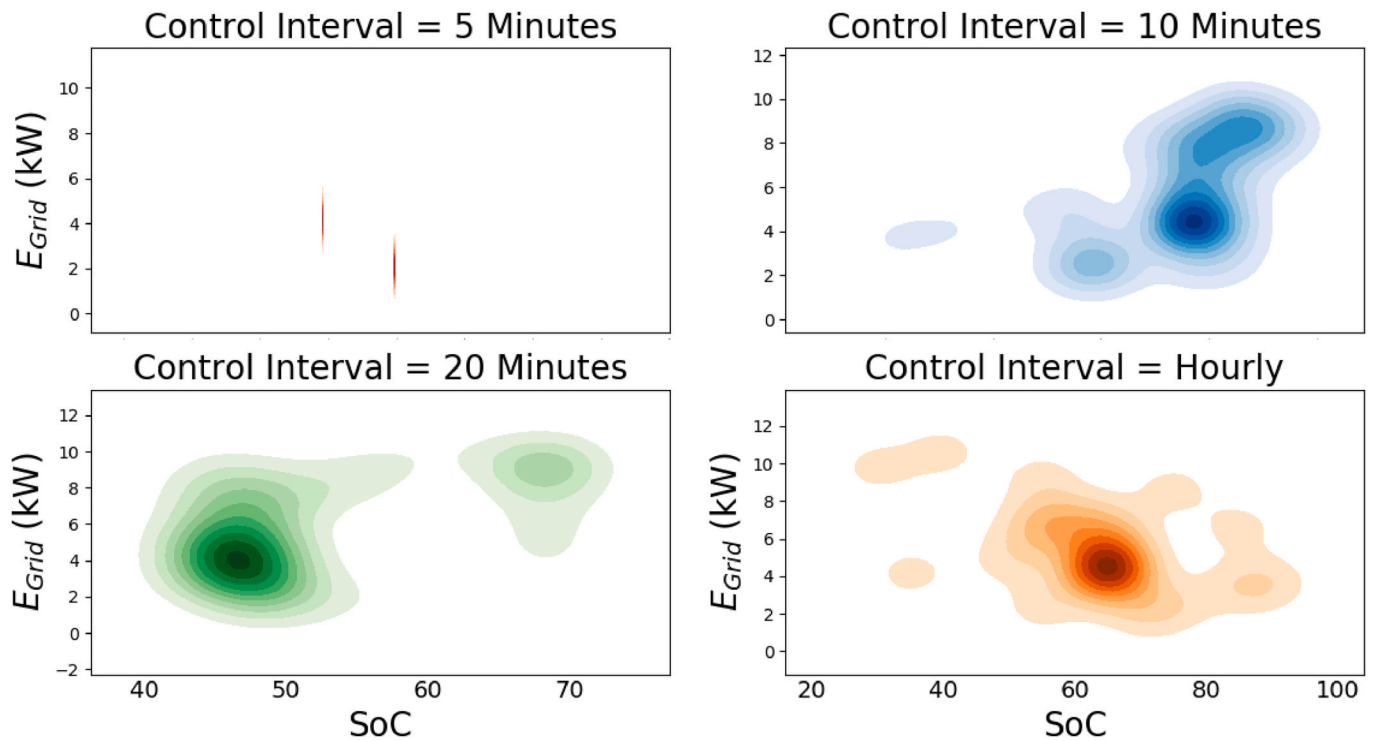


Fig. 40. Bivariate density plots of the power imported from the main grid and the power variation throughout the day for adaptive Q-learning.

discretization levels and algorithms may slightly affect the daily cost, unlike the rewards or the operational characteristics.

5.2. Comparison with different exploration parameters

Fig. 28 shows that the uncertainty in the reward plot in the case of exploitation policy is quite high, making it highly susceptible to the Q-

value initialization. With $\epsilon = 0.05$, the reward of the system is considered optimal, as shown in Fig. 29. This is consistent with most RL literature, which suggests that this value offers a good balance between exploration and exploitation strategies. Fig. 31 shows that the SOC variation in the case of the greedy policy is very similar to the one obtained from the adaptive Q-learning. However, the variation increases with greater agent exploration.

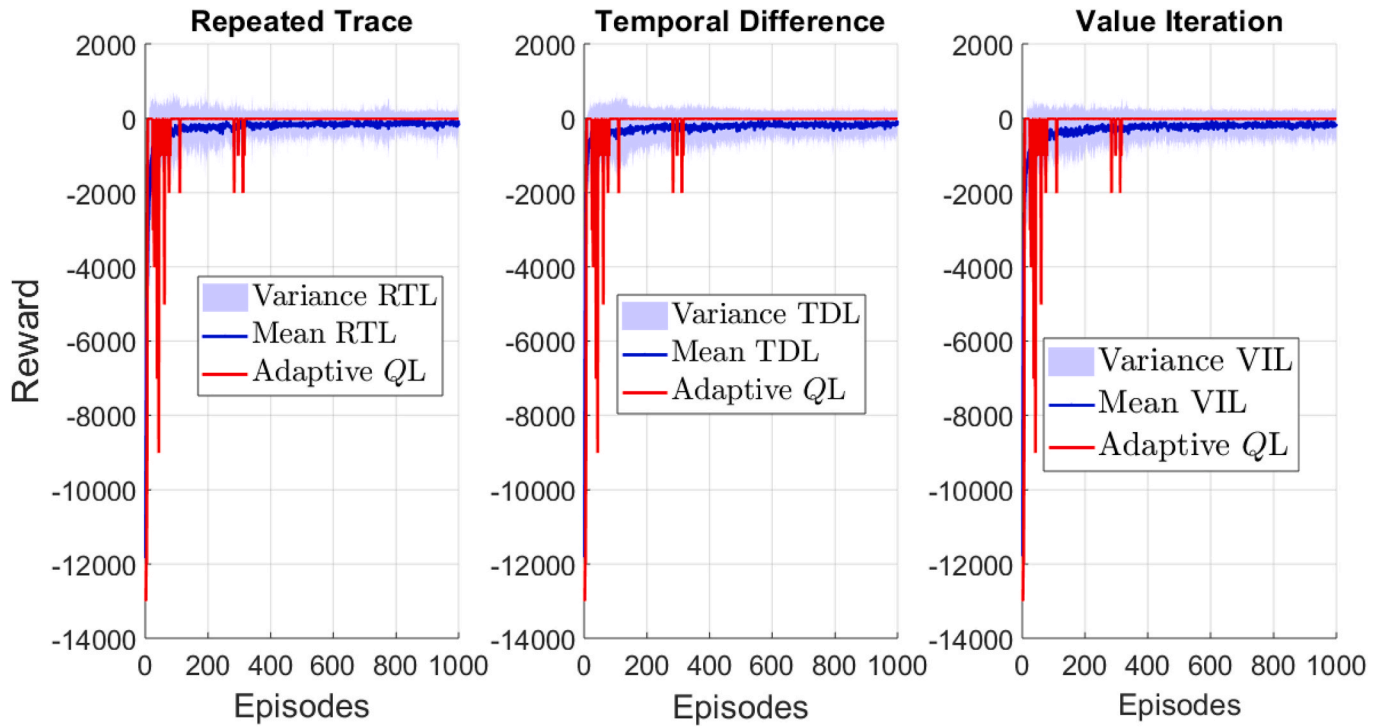


Fig. 41. Reward plots for different exploration parameters level for temporal difference and adaptive Q-learning.

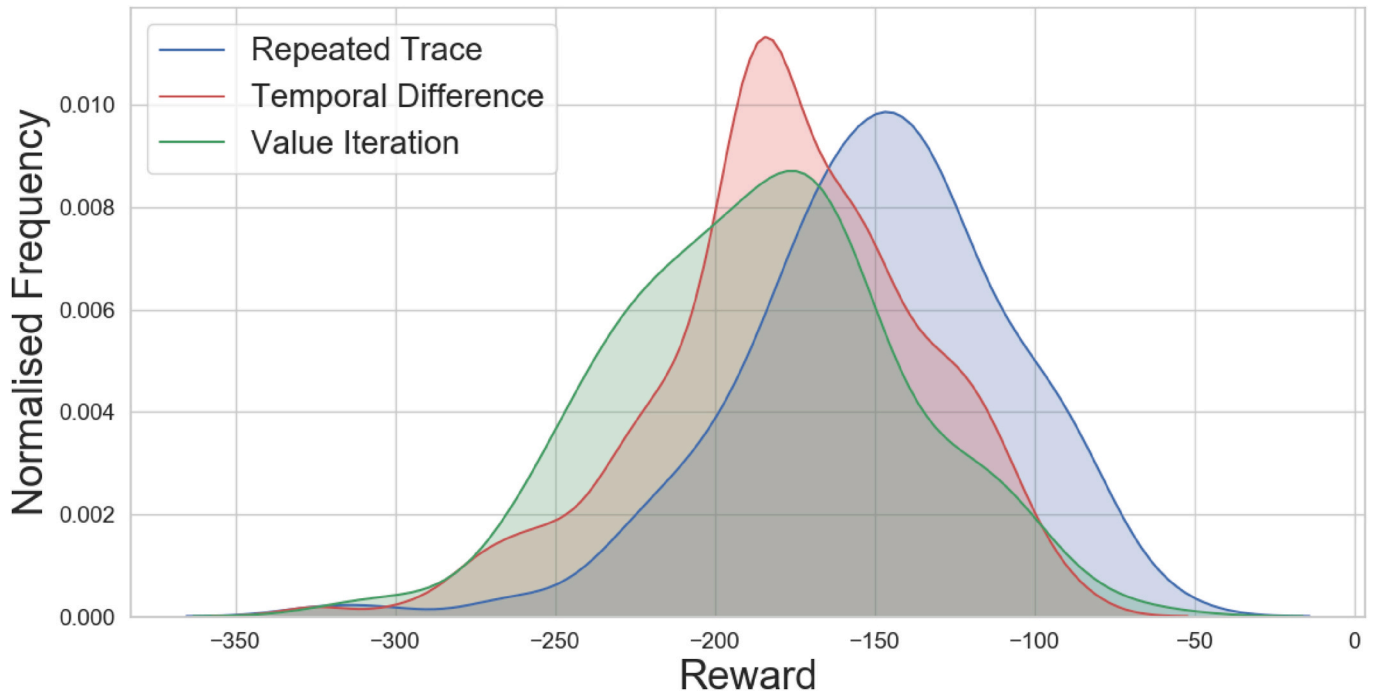


Fig. 42. Histograms of rewards for last 200 episodes for different controllable state discretization level for temporal difference/adaptive Q-learning.

A higher spread in SOC variation is observed with the increased agent exploration, which is consistent with the pattern seen in Fig. 31. Similarly, we can observe how the SOC varies throughout the day with different exploration parameters as shown in Fig. 30. Additionally, from the bivariate plot in Fig. 32, it can be observed that the pattern becomes more rectangular as the tariff cost increases when the agent follows the greedy policy.

5.3. Comparison with decision at different time step

Fig. 34 shows that an increase in the decision time interval (Δt) leads to a decrease in the reward variation across different Q-value initialization. This is due to the noise-filtering effect of averaging when the time interval increases. Noise plays a vital role in the adaptive Q-learning algorithm. The noise also plays a key role in the convergence characteristics of the algorithm as shown in Figs. 33 and 35.

Here, it is observed that the agent is highly penalized for operating

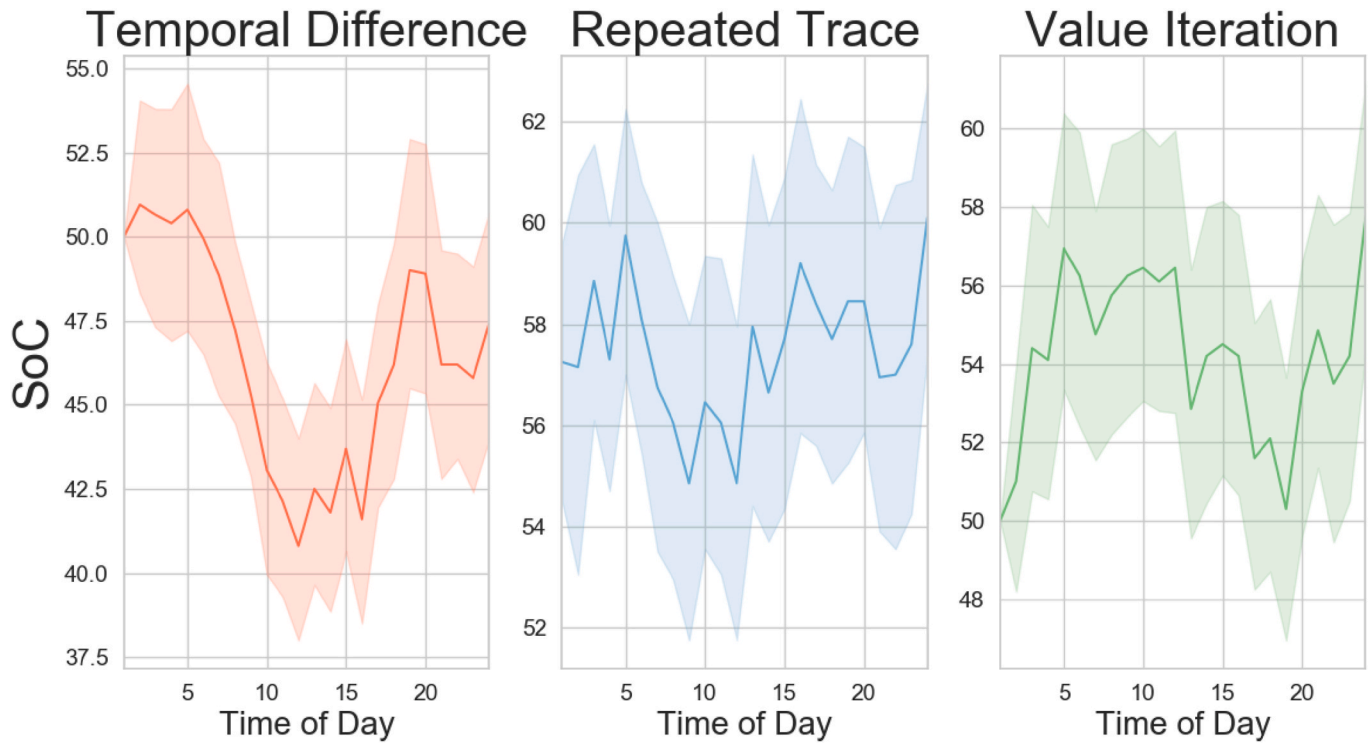


Fig. 43. The variation of SOC during the day with the mean and the confidence bound for different variant of Q-learning for 100 Monte Carlo runs.

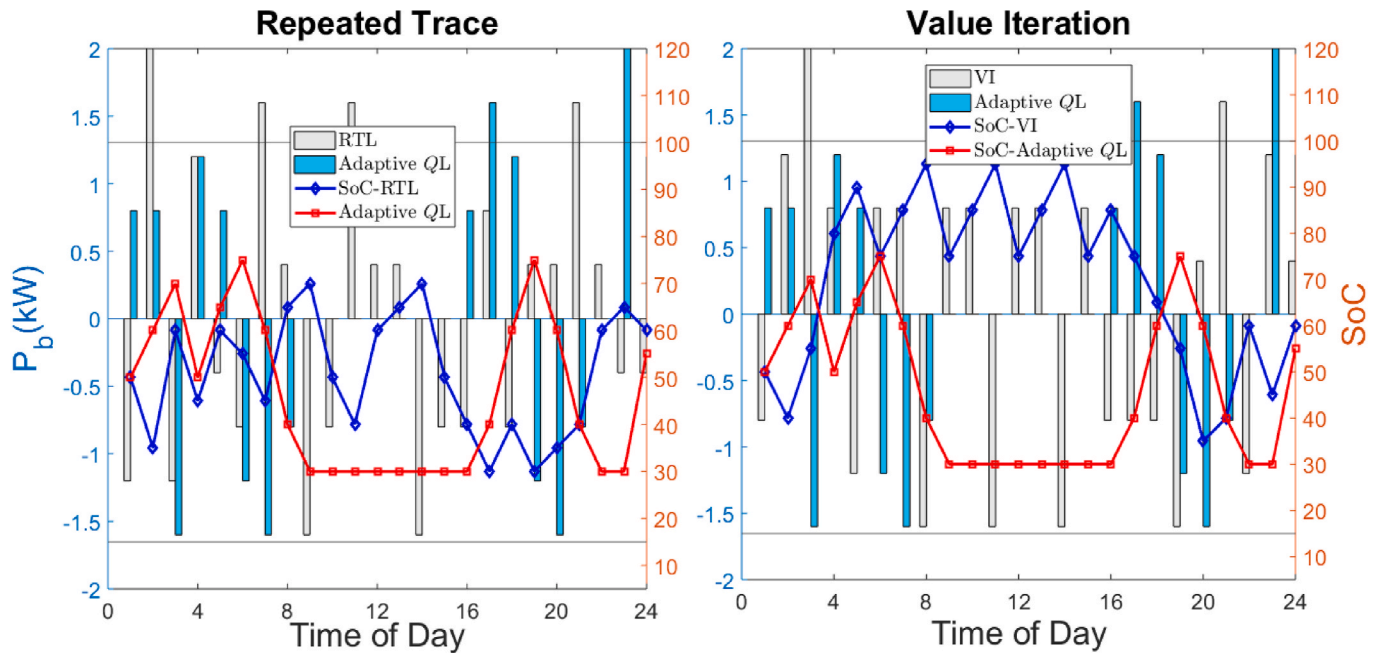


Fig. 44. Battery power at different exogenous state discretization level along with the SoC variations for TD learning and adaptive Q-learning.

the battery near the SOC limits. Hence, the convergence time is longer. However, the convergence characteristics improve when the agent observes the smoothed-out data. However, the averaging effect plays an essential role in the case of TD-learning, where the variation of the reward is higher. We observe that the converged reward is better when the data is not averaged. However, the performance of the algorithm, in terms of rewards, is similar to the adaptive Q-learning (see Fig. 36). The smoothness and better reward characteristics, considering original data is attributed due to the noisy and oscillatory nature of the power

imported from the grid, as shown in Fig. 38. Similar characteristics is also observed in the case of SOC variation, where the battery operates near the lower limit with more noisy data, as shown in Fig. 39. It is also observed, that with the increase in Δt , the confidence interval of the variation of the SOC becomes wider. We also observe a similar pattern in the power exported from the grid for the adaptive RL algorithm as shown in Fig. 37.

Thus, it can be concluded that data averaging and the decision time interval play a vital role in the variation of the parameters with different

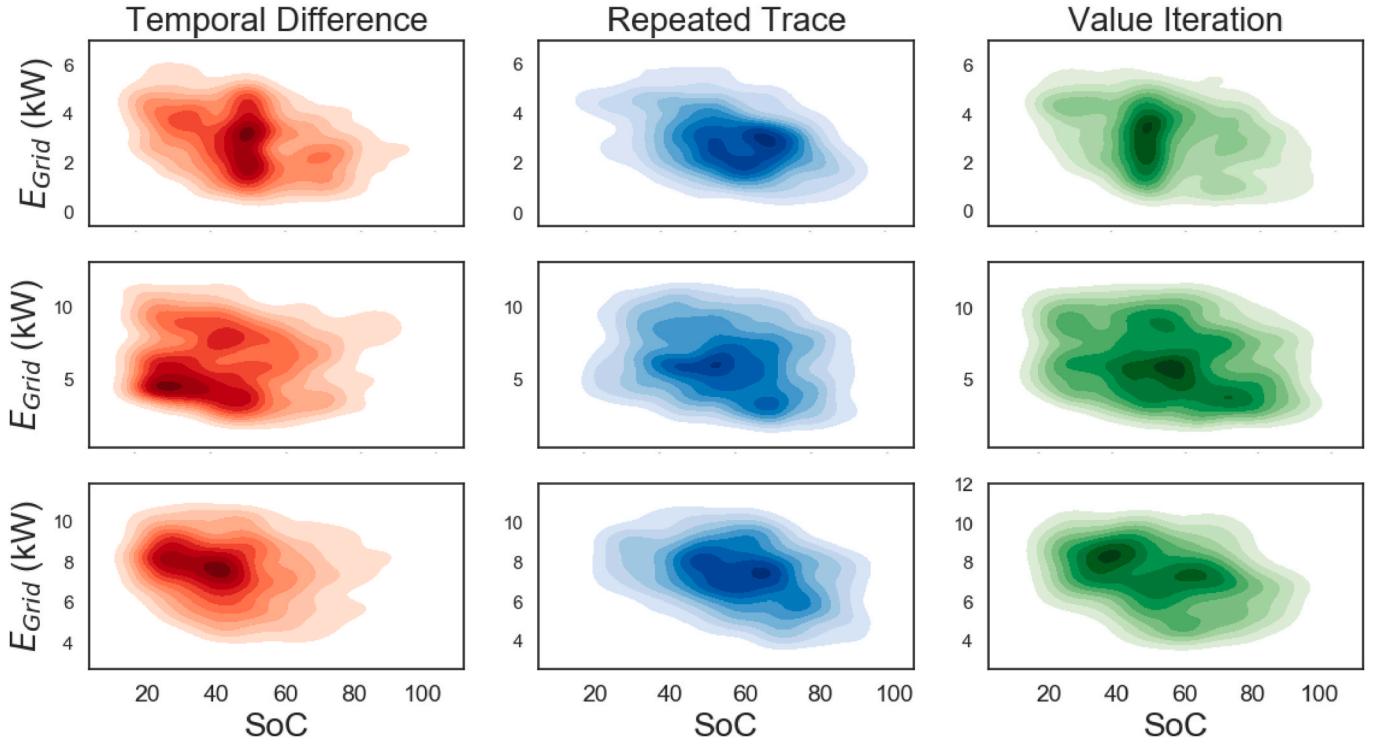


Fig. 45. Bivariate density plots of the power imported, and state of charge variation obtained from 100 Monte Carlo runs during different tariff period. Row 1: off-peak, row 2: mid peak, row 3: peak.

initialization of the Q-table. The bivariate plot of the power drawn from the grid along with the SOC characteristics, as shown in Fig. 40, also exhibits a similar trend. Higher spread in SOC has been observed as the decision control interval increases. However, violation of the limits causes the adaptive Q-learning algorithm to keep the battery operation idle as shown in Fig. 39. Hence, the battery operates near the initial SOC value. The characteristics of the rewards for different variants of Q-learning like repeated trace and value iteration are shown in Fig. 41. It shows slight variability of the rewards for different Q-value initializations. However, the univariate plot of the converged rewards in Fig. 42 suggests that the variation of the converged mean reward in the case of repeated trace is higher, as compared to the other two algorithms. We observe in Fig. 44, that in case of the value iteration algorithm, the SOC variation oscillates when the PV generation is high. The battery in both algorithms operate in discharge mode during the peak period. The bivariate density plot in Fig. 45 shows that the battery in TD algorithm operates at lower SOC ranges as compared to the other variants. The SOC variation in the case of value iteration algorithm is smoother as compared to the other two variants of the algorithm. Similar analysis can also be extended to the variation of SOC as shown in Fig. 43.

We can therefore summarize our overall findings as:

- Higher action levels cause slower convergence of the algorithm in the case of TD learning. Lower mean reward is obtained with higher variability to the Q-table initialization. At lower action levels multimodal distribution is obtained between the SOC bivariate density plots.
- In the case of controllable state discretization, lower values cause an error in approximation. Hence, higher variation is obtained with lower mean reward to Q-initialization. With an increase in the discretization level, the performance becomes similar. At higher discretization level we observe a lower SOC fluctuation in adaptive Q-learning compared to TD-learning as shown in Fig. 13.
- The variation of the reward is similar with an increase in discretization level of the exogenous state. However, it causes the

battery to operate near the minimum SOC limit. At lower discretization level, we observe a lower SOC fluctuation in adaptive Q-learning compared to TD-learning as shown in Fig. 23.

- Optimum performance in terms of reward is obtained when ϵ is set to 0.05. Rewards in exploitation policy are prone to high variability to different Q-table initialization. We observe higher fluctuation in battery commands as the value of ϵ increases as shown in Fig. 31, which is not observed in adaptive Q-learning as it is a deterministic algorithm.
- The rewards obtained in longer control decision intervals on the original data are better as compared to the shorter time intervals with averaged data. The noise present in the data is evident from the high-frequency oscillations in power imported from the grid and the operation of the battery near the lower SOC limit.

5.4. Assumptions and limitations of the study

The work presented in this paper is conducted in a simplified environment to facilitate a deeper analysis and testing of the novel adaptive learning framework. It does not account for rapidly fluctuating tariffs, power flow interactions across multiple microgrids, or hierarchical co-ordination with a distribution system operator (DSO). Although the adaptive Q-learning approach theoretically achieves optimal solutions, it can become trapped in local optima due to noise in load demand, PV generation, and tariff inputs. Therefore, these realistic complexities are excluded, and the study is conducted for academic interests, focusing on the effect of various discretization levels in the state and action spaces and comparing traditional RL with the adaptive Q-learning framework.

6. Conclusion

In this paper, a thorough parametric study has been conducted considering different state and action discretization levels, agent exploration and exploitation properties and the performance at different control time intervals. The algorithms used in the simulations are

adaptive Q-learning and temporal difference learning. A detailed comparison has been made based on the fluctuation in battery commands, convergence speed, and the induced rewards and penalties. Monte Carlo ensembles of the rewards, obtained from random Q-table initialization for TD-learning, is compared with the adaptive RL algorithm. The performances of the algorithms are also benchmarked with other variants of Q-learning e.g. the value iteration and repeated trace. The final rewards and induced penalties obtained for the adaptive Q-learning algorithm is better than that of the Q-learning, since optimal actions are implemented on the environment instead of agent exploration in the latter case. In terms of fluctuation of the battery commands, adaptive RL is better than the TD learning, as it induces lower fluctuation in the battery commands than the latter, which is helpful in the battery life in the long run. In the case of TD learning, the SOC variation is obtained across the given range with better utilization of the battery. The rewards are marginally better in the case of repeated trace RL. Higher oscillations are observed due to peak PV generation during the day for value iteration algorithm as compared to other variants of RL.

This study can be extended in the future to incorporate testing on larger datasets over longer time intervals with different non-stationary characteristics and testing with different tariff variants. Also, further analysis may include formulation and comparison of different reward functions for the adaptive Q-learning and deep Q-learning algorithms extending the findings in Ref. [34], along with their computational complexity vs. online implementation issues in real-time control of RES fed grid connected microgrids.

CRedit authorship contribution statement

Deepak Kumar Panda: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Saptarshi Das:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Funding acquisition, Conceptualization. **Mohammad Abusara:** Writing – review & editing, Validation, Supervision, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was partially supported by the ESIF ERDF Cornwall New Energy project, project number: 05R16P00282. SD was partially supported by the ERDF Deep Digital Cornwall project number: 05R18P02820. For the purpose of open access, the author has applied a 'Creative Commons Attribution (CC BY) license to any author accepted manuscript version arising.

References

- [1] M.F. Zia, E. Elbouchikhi, M. Benbouzid, Microgrids energy management systems: a critical review on methods, solutions, and prospects, *Appl. Energy* 222 (2018) 1033–1055.
- [2] C. Zhang, J. Wu, Y. Zhou, M. Cheng, C. Long, Peer-to-Peer energy trading in a microgrid, *Appl. Energy* 220 (2018) 1–12.
- [3] V.-H. Bui, A. Hussain, H.-M. Kim, Q-learning-based operation strategy for community battery energy storage system (CBESS) in microgrid system, *Energies* 12 (9) (2019) 1789.
- [4] B.V. Mbuwir, F. Ruelens, F. Spiessens, G. Deconinck, Battery energy management in a microgrid using batch reinforcement learning, *Energies* 10 (11) (2017) 1846.
- [5] Y. Ji, J. Wang, J. Xu, X. Fang, H. Zhang, Real-time energy management of a microgrid using deep reinforcement learning, *Energies* 12 (12) (2019) 2291.
- [6] G. Shi, D. Liu, Q. Wei, Echo state network-based Q-learning method for optimal battery control of offices combined with renewable energy, *IET Control Theory & Appl.* 11 (7) (2017) 915–922.
- [7] V.-H. Bui, A. Hussain, H.-M. Kim, Double deep Q-learning-based distributed operation of battery energy storage system considering uncertainties, *IEEE Trans. Smart Grid* 11 (1) (2019) 457–469.
- [8] E. Kuznetsova, Y.-F. Li, C. Ruiz, E. Zio, G. Ault, K. Bell, Reinforcement learning for microgrid energy management, *Energy* 59 (2013) 133–146.
- [9] N. Sadeghianpourhamami, J. Deleu, C. Develder, Definition and evaluation of model-free coordination of electrical vehicle charging with reinforcement learning, *IEEE Trans. Smart Grid* 11 (1) (2019) 203–214.
- [10] Z. Wan, H. Li, H. He, D. Prokhorov, Model-free real-time EV charging scheduling based on deep reinforcement learning, *IEEE Trans. Smart Grid* 10 (5) (2018) 5246–5257.
- [11] H. Ko, S. Pack, V.C. Leung, Mobility-aware vehicle-to-grid control algorithm in microgrids, *IEEE Trans. Intell. Transport. Syst.* 19 (7) (2018) 2165–2174.
- [12] B. Xu, D. Rathod, D. Zhang, A. Yebi, X. Zhang, X. Li, Z. Filipi, Parametric study on reinforcement learning optimized energy management strategy for a hybrid electric vehicle, *Appl. Energy* 259 (2020) 114200.
- [13] H. Sun, Z. Fu, F. Tao, L. Zhu, P. Si, Data-driven reinforcement-learning-based hierarchical energy management strategy for fuel cell/battery/ultracapacitor hybrid electric vehicles, *J. Power Sources* 455 (2020) 227964.
- [14] T. Liu, X. Hu, W. Hu, Y. Zou, A heuristic planning reinforcement learning-based energy management for power-split plug-in hybrid electric vehicles, *IEEE Trans. Ind. Inf.* 15 (12) (2019) 6436–6445.
- [15] X. Han, H. He, J. Wu, J. Peng, Y. Li, Energy management based on reinforcement learning with double deep Q-learning for a hybrid electric tracked vehicle, *Appl. Energy* 254 (2019) 113708.
- [16] J. Wu, H. He, J. Peng, Y. Li, Z. Li, Continuous reinforcement learning of energy management with deep Q-network for a power split hybrid electric bus, *Appl. Energy* 222 (2018) 799–811.
- [17] E. Mocanu, D.C. Mocanu, P.H. Nguyen, A. Liotta, M.E. Webber, M. Gibescu, J. G. Sloopweg, On-line building energy optimization using deep reinforcement learning, *IEEE Trans. Smart Grid* 10 (4) (2018) 3698–3708.
- [18] Q. Wei, D. Liu, G. Shi, A novel dual iterative Q-learning method for optimal battery management in smart residential environments, *IEEE Trans. Ind. Electron.* 62 (4) (2014) 2509–2518.
- [19] S. Kim, H. Lim, Reinforcement learning based energy management algorithm for smart energy buildings, *Energies* 11 (8) (2018) 2010.
- [20] W. Li, H. Cui, T. Nemeth, J. Jansen, C. Uenluebayir, Z. Wei, L. Zhang, Z. Wang, J. Ruan, H. Dai, Deep reinforcement learning-based energy management of hybrid battery systems in electric vehicles, *J. Energy Storage* 36 (2021) 102355.
- [21] X. Tang, J. Zhang, D. Pi, X. Lin, L.M. Grzesiak, X. Hu, Battery health-aware and deep reinforcement learning-based energy management for naturalistic data-driven driving scenarios, *IEEE Trans. Electr.* 8 (1) (2021) 948–964.
- [22] C. Guo, X. Wang, Y. Zheng, F. Zhang, Optimal energy management of multi-microgrids connected to distribution system based on deep reinforcement learning, *Int. J. Electr. Power Energy Syst.* 131 (2021) 107048.
- [23] Y. Liang, Z. Ding, T. Zhao, W.-J. Lee, Real-time operation management for battery swapping-charging system via multi-agent deep reinforcement learning, *IEEE Trans. Smart Grid* 14 (1) (2022) 559–571.
- [24] S. Abedi, S.W. Yoon, S. Kwon, Battery energy storage control using a reinforcement learning approach with cyclic time-dependent markov process, *Int. J. Electr. Power Energy Syst.* 134 (2022) 107368.
- [25] Y. Cui, Y. Xu, Y. Li, Y. Wang, X. Zou, Deep reinforcement learning based optimal energy management of multi-energy microgrids with uncertainties, *CSEE J. Power Energy Syst.* (2024), <https://doi.org/10.17775/CSEEJPES.2023.05120>.
- [26] A. Hedayatnia, J. Ghafourian, R. Sepehrzad, A. Al-Durad, A. Anvari-Moghaddam, Two-stage data-driven optimal energy management and dynamic real-time operation in networked microgrid based deep reinforcement learning approach, *Int. J. Electr. Power Energy Syst.* 160 (2024) 110142.
- [27] D. Dominguez-Barbero, J. Garcia-González, M.A. Sanz-Bobi, A. Garcia-Cerrada, Energy management of a microgrid considering nonlinear losses in batteries through deep reinforcement learning, *Appl. Energy* 368 (2024) 123435.
- [28] J. Zhu, D. Li, Y. Chen, J. Chen, Y. Luo, Parallel hybrid deep reinforcement learning for real-time energy management of microgrid, *J. Mod. Power Syst. Clean Energy* (2024), <https://doi.org/10.35833/MPCE.2024.000662>.
- [29] Y. Tavakol-Moghaddam, M. Boroushaki, M. Astaneh, Reinforcement learning for battery energy management: a new balancing approach for Li-ion battery packs, *Res. Eng.* 23 (2024) 102532.
- [30] T.A. Nakabi, P. Toivanen, Deep reinforcement learning for energy management in a microgrid with flexible demand, *Sustain. Energy Grids Networks* 25 (2021) 100413.
- [31] Q. Meng, S. Hussain, F. Luo, Z. Wang, X. Jin, An online reinforcement learning-based energy management strategy for microgrids with centralized control, *IEEE Trans. Ind. Appl.* 61 (1) (2025) 1501–1510.
- [32] L. Wen, K. Zhou, S. Yang, X. Lu, Optimal load dispatch of community microgrid with deep learning based solar power and load forecasting, *Energy* 171 (2019) 1053–1065.
- [33] M. Sun, T. Zhang, Y. Wang, G. Strbac, C. Kang, Using Bayesian deep learning to capture uncertainty for residential net load forecasting, *IEEE Trans. Power Syst.* 35 (1) (2019) 188–201.
- [34] D.K. Panda, O. Turner, S. Das, M. Abusara, Prioritized experience replay based deep distributional reinforcement learning for battery operation in microgrids, *J. Clean. Prod.* 434 (2024) 139947.
- [35] B. Luo, Y. Yang, D. Liu, Adaptive Q-learning for data-based optimal output regulation with experience replay, *IEEE Trans. Cybern.* 48 (12) (2018) 3337–3348.
- [36] C.J. Watkins, P. Dayan, Q-learning, *Mach. Learn.* 8 (1992) 279–292.

- [37] G.R. Newsham, B.G. Bowker, The effect of utility time-varying pricing and load control strategies on residential summer peak electricity use: a review, *Energy Policy* 38 (7) (2010) 3289–3296.
- [38] R. Carmichael, R. Gross, R. Hanna, A. Rhodes, T. Green, The demand response technology cluster: accelerating UK residential consumer engagement with time-of-use tariffs, electric vehicles and smart meters via digital comparison tools, *Renew. Sustain. Energy Rev.* 139 (2021) 110701.
- [39] G. Fridgen, M. Kahlen, W. Ketter, A. Rieger, M. Thimmel, One rate does not fit all: an empirical analysis of electricity tariffs for residential microgrids, *Appl. Energy* 210 (2018) 800–814.
- [40] A.J. Pimm, T.T. Cockerill, P.G. Taylor, Time-of-use and time-of-export tariffs for home batteries: effects on low voltage distribution networks, *J. Energy Storage* 18 (2018) 447–458.
- [41] F. Ruelens, B.J. Claessens, S. Quaiyum, B. De Schutter, R. Babuska, R. Belmans, Reinforcement learning applied to an electric water heater: from theory to practice, *IEEE Trans. Smart Grid* 9 (4) (2016) 3792–3800.
- [42] K.C. Divya, J. Ostergaard, Battery energy storage technology for power systems—An overview, *Elec. Power Syst. Res.* 79 (4) (2009) 511–520.
- [43] H. Shteingart, T. Neiman, Y. Loewenstein, The role of first impression in operant learning, *J. Exp. Psychol. Gen.* 142 (2) (2013) 476.
- [44] R.S. Sutton, A.G. Barto, *Reinforcement Learning: an Introduction*, MIT Press, 2018.
- [45] R.S. Sutton, Learning to predict by the methods of temporal differences, *Mach. Learn.* 3 (1988) 9–44.
- [46] T. Jaakkola, M. Jordan, S. Singh, Convergence of stochastic iterative dynamic programming algorithms, *Adv. Neural Inf. Process. Syst.* 6 (1993).
- [47] M.L. Waskom, Seaborn: statistical data visualization, *J. Open Source Softw.* 6 (60) (2021) 3021.



DSPACE

<https://dspace.org/>

Parametric study of adaptive reinforcement learning for battery operations in microgrids

Panda, Deepak Kumar; Das, Saptarshi; Abusara, Mohammad

2025-09-01

Elsevier

<https://doi.org/10.1016/j.renene.2025.123250>

Panda DK, Das S, Abusara M. (2025) Parametric study of adaptive reinforcement learning for battery operations in microgrids. Renewable Energy, Volume 250, September 2025, Article number 123250