Energy Storage Scheduling for Cost Minimization Using Deep Q-Learning

Yaju Rajbhandari
School of Engineering, University College Cork, Cork, Ireland,
MaREI Centre, Environmental Research Institute, University College Cork,
Cork, Ireland
rajbhandari.yaju@umail.ucc.ie

Dr. Barry Hayes
School of Engineering, University College Cork, Cork, Ireland,
MaREI Centre, Environmental Research Institute, University College Cork,
Cork, Ireland
barry.hayes@ucc.ie

2025 IEEE Kiel PowerTech | 979-8-3315-4397-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/PowerTech59965.2025.11180258

***Abstract*— This paper presents a Discrete-Action Deep Q-Network (DQN) approach for scheduling Energy Storage Systems (ESS) to optimize energy costs for residential households. The proposed model normalizes the state based on the maximum allowable actions defined by the policy, enabling effective state representation and decision-making. Simulation results validate the superiority of the proposed method, which achieves cost savings of 43% compared to a baseline with no rule-based strategy and 21% more than the average rule-based approach. Additionally, the proposed discrete-action DQN approach demonstrates an 11% improvement in cost efficiency over a three-level scheduling DQN method, highlighting the benefits of action discretization. These findings underscore the potential of the proposed approach to enhance energy management in smart grids, effectively optimizing storage system contributions for residential households while significantly reducing energy costs.**

Keywords— Energy Management, Optimization, Reinforcement Learning, Deep Q-Learning

I. Introduction

With growing demand for electric energy, and limitations on clean sources of energy, managing energy on the demand side has played a significant role in enabling sustainable growth of energy systems [1, 2]. Energy demand reduction or shifting from peak to off-peak period has played a significant role in avoiding instability in the power system and avoiding situations leading to blackouts. Common approaches used by service providers are the implementation of market-based demand response programs including price signal strategies such as time of use tariffs [3], real-time pricing of energy [4] , demand charges or critical and peak pricing [3, 5].

However, from the perspective of the user, it can be inconvenient, as shifting or shedding of energy usage requires their active participation to maintain economic benefits. A common approach to overcome this issue has been the integration of Distributed Energy Resources (DER), such as solar and ESS with an energy management system (EMS) on individual households [6]. This concept of managing energy from the consumer side has enabled users not only to consume energy efficiently but also to support the grid through generation [7, 8]. With the installation of DER and ESS, energy management can be optimized to meet household energy requirements or generate additional income by importing energy during low-cost periods and selling surplus energy back to the grid during high-price periods, thereby enhancing economic benefits. To this end, energy scheduling with EMS has been widely studied. Solving for optimum energy scheduling is inherently complex due to the interplay of several factors such as uncertainty of energy demand and renewable generation, the large-scale nature of the system, conflicting economic, strict technical and regulatory requirements. Various approaches have been discussed in the literature, for instance, linear programming-based approaches [9, 10] to heuristic-based Evolutionary Algorithm (EA) [11, 12] and even Machine Learning (ML) approaches [13-17].

In recent decades, ML approaches have been a major field of interest, particularly Deep Reinforcement Learning (DRL), which has shown promise in optimizing energy management systems for smart buildings. The application of RL has demonstrated promising results optimising the overall energy cost while maintaining thermal comfort for residential households[13]. The application of RL has expanded beyond just thermal management for residential households, research such as [13, 14], demonstrating the superiority of RL methods over heuristic models, explored RL for scheduling charging and discharging of ESS, considering the bidirectional flow of energy from residential households. To further reduce the computational complexity of controlling multiple appliances, the concept of multi-agent reinforcement learning (MARL) has emerged. In MARL-based systems, each controllable appliance is treated as an independent agent, allowing for decentralized learning and optimization. [15-17], implemented a multi-agent RL-based method for Home Energy Management Systems (HEMS), where each agent independently learns actions that align with a common optimization goal. This decentralization reduces computational burdens and allows for diverse actions such as power shifting, time shifting, and ESS charging/discharging can have distinct action sets, enabling energy scheduling solutions to increase overall profit.

Though the RL method shows promising results, its practical application still faces challenges related to training time, scalability, and adaptability to dynamic environments. It generally requires extended exploration periods to reach acceptable performance levels in each environment. This makes direct implementation in real-world systems impractical. As [18] questions the robustness of the model-free approach under changing constraints, it shows how the predicted control can go off trajectory. To overcome this, pre-trained policy functions are often developed in simulated environments before being deployed in real systems, with each environment having its own policies.

This work has been supported by Taighde Éireann – Research Ireland under grant number 22/FFP-A/10455.

In this paper, the energy scheduling problem for an individual household is formulated as an Markov Decision Process (MDP) from a user perspective. The objective is to find the optimal scheduling of an hour ahead energy, to minimize the cost of energy utilized based on the varying price of the energy throughout the day. The proposed model relies on the hour ahead forecasted generation and load of the energy along with the import and export price and the current state of charge (SoC) of the battery, to schedule the energy for the next hour to be imported or exported by the ESS. The paper focuses on normalizing the simulation environment along with the policy developed to adapt to different system conditions based on the discretized actions of charging and discharging the ESS. The experimental results compare the optimization outcomes with three other approaches in HEMS.

The rest of the paper is organized as follows: The problem formulation is introduced in Section II further discussing the deep RL-based approach and its learning mechanism. In Section III, the simulation setup and comparative analysis are presented to demonstrate the working of the proposed approach, Finally, Section IV gives the conclusion.

II. Problem Formulation

The energy scheduling problem is formulated as a finite MDP with discrete time steps, where each time step represents one hour. At each time step t, the system observed the state s_t, which includes information about the hour ahead predicted system generation and demand estimating the surplus energy, the import and export prices of energy for the next hour and the current state of charge of the battery. Based on the information the user selects the optimal action a_t, which represents the amount of energy to be contributed by the ESS during the time interval. Positive values of a_t correspond to the energy import, while a negative value represents energy exported with respect to the ESS. After the execution of the action, the system then transitions to a new state s_{t+1}, where a new action a_{t+1} is selected for the next time step. Using the MDP, a mathematical architecture for modelling decisions can be implemented for different conditions, where during the training phase the result is partly random or partly under control obtained using greedy policy. The sequential decision-making process can be represented as five-tuple structures for the MDP $(s_t, a_t, P_t, r_t, \gamma)$[14]. The details on the MDP formulation are discussed below:

State definition: The system state s_t at time step t is represented as a vector $\left(surplus_t, SoC_t, I_t^{price}, E_t^{price}\right)$. The vector encapsulates 4 different pieces of information: $surplus_t$ indicates the difference in predicted generation and demand, $generation_t - demand_t$; SoC_t is the current state of charge of the battery; and I_t^{price}, E_t^{price} represented the import and export price of the energy for the next hour, here the price can be a predicted real-time price or time of use (ToU) tariff.

Action Space: Based on the state s_t, the a_t represents the decision to import or export energy for the next hour. The user can make money by importing the energy at a lower import price and the export or using battery when the import price is high. Action taken can be constrained as $-e_{export}^{max} \leq a_t \leq e_{import}^{max}$ where, $-e_{export}^{max}$ and e_{import}^{max} are the maximum allowable energy export and import, which is the maximum charging or discharging from the battery, respectively. As suggested by [14], the action is discretized into predefined steps. The input to the proposed architecture of the RL model has 9 possible actions which range from -1 to 1, such that the action space is normalized, however the action can range to n different actions that can be taken, and the total action steps can be written as:

$$a_t = \{\mathrm{a}_1, \mathrm{a}_2, \mathrm{a}_3 \dots \mathrm{a}_\mathrm{n}\} \tag{1}$$

Reward Function: The reward r_t at each time step t consists of two components are defined as,

$$r_t = r_t^{cost} + r_t^{unbalance} \tag{2}$$

Where, r_t^{price} is the reward received from the cost of energy imported or exported, where imported cost is kept at negative while the export is the positive price which can be further expanded as;

$$r_t^{cost} = \begin{cases} -abs\left(P_t^{import} \cdot a_t\right), & a_t < 0\ [import] \\ abs(P_t^{export} \cdot a_t)\,, & a_t \geq 0\ [export] \end{cases} \tag{3}$$

Likewise, $r_t^{unbalance}$ is the reward received for the unbalance faced in the system, it is mover a punishment that indicates the action taken leads to higher/lower, import/export of energy required than the action taken. To penalize unnecessary high imports or exports, an unbalanced reward is defined as:

$$r_{unbalance} = -\frac{|u_t^{energy}|}{Normalizer} \tag{4}$$

Here, $\left|u_t^{energy}\right|$, is the absolute unbalance in energy, such that the unbalance can be negative or positive, one of the scenarios can be importing at the full SoC or exporting at zero surplus and zero SoC. The unbalance is normalized by the maximum charging to avoid fluctuation in the learning process and penalize poor decisions during the learning process.

A. Deep Reinforcement Learning

In general, Q-learning methods rely on a lookup table to represent the Q-value function, which maps state-action pairs to the expected rewards. While the table-based approach is effective for low-dimensional problems, it becomes infeasible as the dimensionality of the state and action space increases. In the case of an energy system, the observed state variables such as energy, state of charge and price are often continuous, which can result in a vast state-action space.

To overcome the limitation, one of the approaches is to use a function approximator, in the presented case the model uses a universal function approximator (Neural Network) to approximate the optimal Q value function $\mathrm{Q}^*(s, a)$. Fig. 1 illustrates the overall architecture of the proposed DRL-based energy scheduling framework. The state representation network considers various inputs, including generation, demand, SoC, and energy prices. These inputs

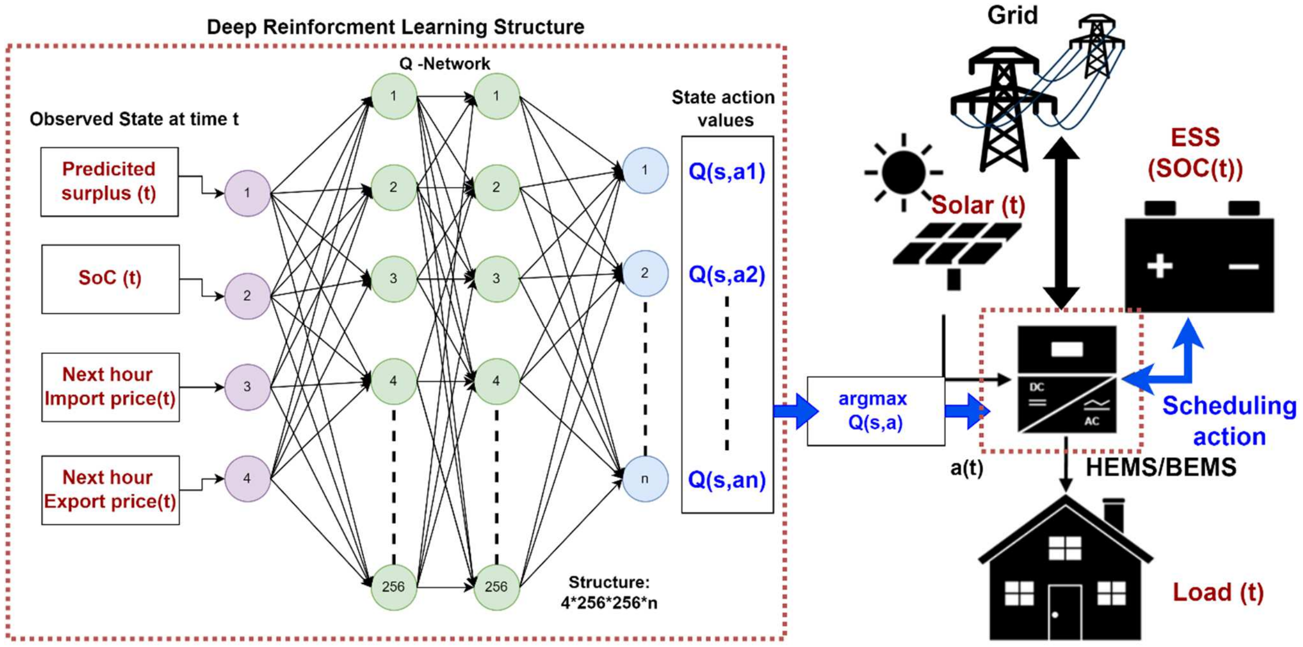

Fig. 1: Architecture of DQN based optimization

are fed into a deep neural network, which computes the Q-values for all possible schedules. The action corresponding to the highest Q-value is selected as the optimal decision, determining whether to import or export energy and by how much, which can be presented as $a_t = argmax_\pi Q^*(s, a)$. As shown in the equation 1, the action can range from 2 to n number of possible actions. In the presented case scenarios the import-export action can have varying ranges such that the proposed method applies 9 levels of scheduling from -1 to 0 to +1 with a 0.25 step difference, with 1 being the maximum energy that can be imported or exported from an ESS at a defined period. The action can be reduced to a required discrete state based on the system control capability, focusing only on discharging, doing nothing and charging which is one of the approaches further discussed in Section III.

B. *Algorithm for Learning Energy management:*

The EMS of the household relates to the PV and ESS, learn the energy scheduling policies focused on optimizing the cost the energy consumed. The proposed energy scheduling framework leverages a DRL. The process first initializes an experience replay memory M to store state transitions tuple. In step 2 the algorithm initialized the neural network Q with random parameters of weights w and biases b. Next, the training of the Q network is done at the end of the episode and the training is done for E number of episodes. For each training epoch, the building environment is reset to its initial state, and the system interacts with the environment until termination. Here the training is done with random data for generation, demand, and initial SoC. As all the states are normalized, the generation demand and state of charge are randomly selected. Such that surplus is the function of demand and generation. To train the policy, the algorithm initializes the import price between the maximum possible price and minimum price can be represented as:

$$Energy_{price}^{min} \le price_t^{import} < Energy_{price}^{max} \tag{5}$$

In the considered scenario the trained policy focuses on constant export price, kept at \$ 0.1. The range of import price is limited between \$ 0.4 and \$ 0.1. Further Q network is trained after each episode, where each episode is run for max 1 day which results in 24-time steps. At each time step t, the current state s_t is observed, and an action a_t is selected using an epsilon-greedy policy to balance exploration and exploitation. The epsilon-greedy policy is formulated as below:

$$\epsilon = \max(\epsilon - \Delta\epsilon, \epsilon_{\min}) \tag{6}$$

$$a_t = \begin{cases} \mathcal{A}^i \in A \mid i = \mathrm{random}(n) & \text{probability } \epsilon \\ \underset{\pi^*}{\mathrm{argmax}} Q(s, \mathrm{a}) & \text{otherwise} \end{cases} \tag{7}$$

Where, ϵ is the epsilon value, which drives the policy selection, such that a random value v is selected from a uniform distribution between 0 to 1 and if the $\epsilon \ge v$, then random policy is selected, else the action is selected using greedy policy π as shown in equation 7. The value of ϵ is slowly decreased with an increase in number of episodes run for training as shown in equation 6, where, the value of ϵ is decreased by Δϵ. The selected action is executed, yielding an immediate reward r_t, and the environment transitions to a new state s_{t+1}. The transition tuple (s_t, a_t, r_t, s_{t+1}) is stored in the replay memory M. Once the replay memory size exceeds a predefined threshold D_{train}, the algorithm starts to train the NN using mini batch gradient descent method. For this, a random mini batch of transitions is sampled from memory D_{memory} , and the expected cumulative reward yi for each transition is calculated using the Bellman equation:

$$y_i = r_i + \gamma Q(s_{i+1}, a_{i+1} | \pi *) \tag{8}$$

Here, i is the i^{th} sample on the mini batch sampled, γ is the discount factor that determines the importance of future rewards for the policy π ∗. Further, the neural network is updated by minimizing the loss function which is represented as means square error loss. Adam optimizer is applied to adjust the neural network weights, improving the policy over time[19]. The algorithm iteratively refines the scheduling policy by learning from interactions with the environment, enabling the DRL model to make optimal energy import/export decisions under varying system states.

III. SIMULATION AND ANALYSIS

To analyse the effectiveness of the proposed model, the model has been compared with 3 different approaches, no rule, and average price rule-based approach considering the SoC and DQN model with 3 levels of scheduling.

A. No Rule and Average Price Rule based approach

The simple rule-based approach where the charging and discharge of the battery scheduling is done based on the last 24 hrs average price i.e., importing if the price is lower than the average price and exporting at the higher price of the energy combined with the discharge bandgap which is defined from 30% to 80% such the unbalance condition is avoided under all conditions. Whereas no rule only applies the bandgap to avoid unbalance and over-charging and discharging of the battery.

B. DQN for 3 level scheduling

The three-action DQN approach limits the system's actions to charging, discharging, or remaining idle, similar actions to a rule-based strategy. In contrast to the proposed discrete actions, here the DQN for 3 actions is in between the proposed and conventional rule-based method, the action is defined by trained policy, such that the action steps can be written as:

$$a_t = \{1,\ 0, -1\} \tag{9}$$

where -1 is the battery's discharging action with max power, 0 refers to no input from the battery, and 1 refers to the battery's charging action with the max power system.

C. Proposed DQN approach with 9 level scheduling

The proposed scheduling methods focus on discretizing the action to how much energy the battery contributes in the defined period. The input of the proposed architecture of the RL approach has 9 possible discrete actions which range from -1 to 1, such that the action steps can be seen in equation 10.

$$a_t = \{1, 0.75, 0.5, 0.25, 0, -0.25, -0.5, -0.75, -1\} \tag{10}$$

Where -1 is the discharging action of the battery with max power, 0 refers to no input from the battery and 1 refers to charging of the battery with max power system. For -0.5 the battery contributes 50% of the max power which for 1000W normalizer will be 500W over a period and 0.75 refers to 75% and so on.

D. Environment setup

The environmental setup plays a crucial role in reinforcement learning (RL), as the agent learns its policy based on the system's response to its actions. Unlike early RL applications in gaming [20], where environmental variables remain constant, energy systems present dynamic and variable conditions. These variations arise from differences in consumer energy requirements, making it impractical to generalize a single policy for all scenarios. In model-free RL, policies must adapt to each unique environment. One effective approach is to normalize the input and output, enabling the RL policy to scale effectively across varying environmental conditions. In this context, the environment is defined by a normalizing value, representing the maximum energy that can be imported or exported from the battery within a defined period, such as an hour, half-hour, or 15 minutes. For the simulation, one hour is considered, and the residential household is modelled with a normalizing value of 1 kWh for simplicity.

E. Simulation

To test the proposed model, a simulation test is performed and compared with the other three approaches. This tests how the battery can contribute to residential load management and reduce the overall cost of the energy consumed considering the varying import price at a constant export price. Fig. 2 shows the result of 1-day simulation data, generation, consumption and price variation data. We can observe in Fig. 2(a) and 2(b), the demand and generation from the first day of the simulation. The average price ranges from $ 0.32 to $ 0.25 per unit kWh. Based on this the charging and discharging action changes. Fig. 2(c) shows the change in action taken by the proposed DQN approach with changing price, the agent can learn the policy to charge the battery during lower price hours and contribute energy during higher price hours, such that the action and price are opposing in most cases. However, with the unbalance penalty, it also charges the battery even in higher price hours, such as in hours 11 and 12.

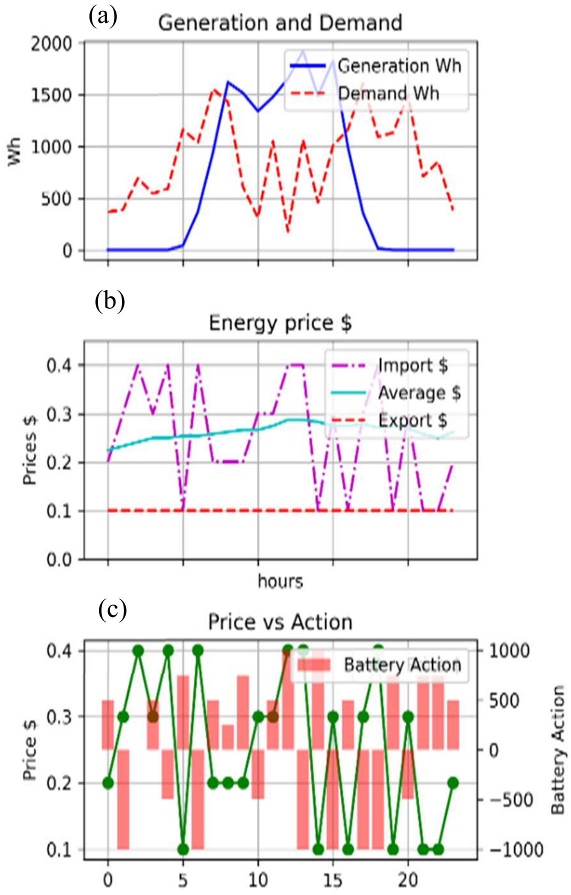

Fig. 2: 1-day simulation results of proposed DQN (a) Generation and Demand; (b) Energy price variation, (c) Import price vs Action

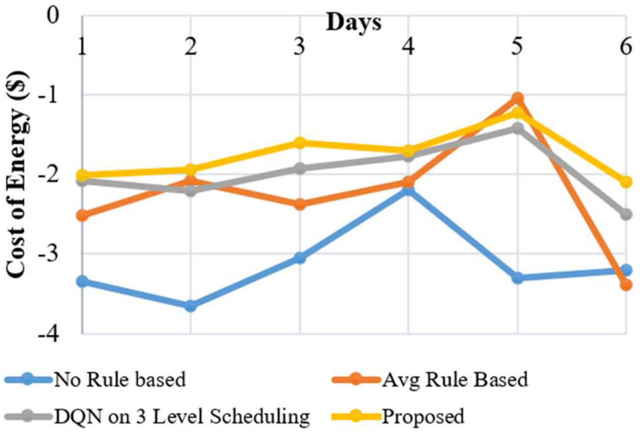

Fig. 3: 6-day energy cost comparison

F. *Results*

Fig. 3 provides a comparative analysis of energy costs over six days for 3 methods and the "Proposed DQN Discrete Scheduling." It is evident that the proposed DQN discrete scheduling method consistently outperforms the others, showcasing its superiority in optimizing energy costs over time. For the “No Rule” case, energy costs are consistently higher across all six days. This method lacks any optimization mechanism, resulting in inefficient energy utilization and the highest overall costs.

The lack of adaptability and control in this method highlights the need for advanced scheduling strategies. The average rule-based method provides a noticeable improvement over the no rule-based approach Its energy costs remain relatively stable throughout the 6 days, indicating its ability to introduce some level of optimization. Especially on day 5 as it outperforms others by sight margin, however, its lack of dynamic adaptability prevents further cost reductions on alternate days, keeping it from achieving significant improvements.

The "DQN on 3-Level Scheduling" method demonstrates the benefits of reinforcement learning by consistently achieving lower energy costs than the rule-based approach. The method strikes a balance between optimization and computational complexity, resulting in better performance. However, its reliance on a limited action space restricts its ability to fully leverage the available opportunities for cost reduction. The proposed DQN with a discrete scheduling approach outperforms all other approaches across the 6 days, with a minor setback on day 5, which allows major recovery by savings $1.3 on the last day. This method not only achieves the lowest energy costs but also demonstrates a more consistent improvement trend. By employing fine-grained discretization for scheduling, it enables more precise and effective decision-making, maximizing cost savings. Its adaptability to dynamic energy usage patterns and optimized control over distributed energy resources (DERs) make it the most efficient approach.

TABLE I. OVERALL COST OF ENERGY

Approach	Total Energy Cost ($)
No Rule	-18.75
Avg Rule Based	-13.51
DQN on 3 Level Scheduling	-11.92
Proposed	-10.59

Similarly, results presented in Table 1 highlight the 6 days overall cost of energy across four methods: "No Rule," "Average Rule-Based," "DQN on 3-Level Scheduling," and "Proposed DQN Discrete Scheduling." Each method demonstrates distinct levels of performance in terms of energy cost optimization. The "No Rule" approach, with a cost of $−18.75, which is the highest energy cost. This result emphasizes the inefficiency of energy management without any form of optimization strategy. The "Average Rule-Based" method shows a substantial improvement, reducing the cost to $−13.51. This represents a 28% reduction compared to the "No Rule" approach. It highlights that even basic rule-based strategies, such as predefined guidelines for energy usage, can significantly enhance cost efficiency in energy management systems. By incorporating reinforcement learning, the "DQN on 3-Level Scheduling" method reduces the cost further to $−11.92, achieving a 12% improvement over the "Average Rule-Based" approach. This result demonstrates the effectiveness of reinforcement learning in optimizing energy scheduling and addressing the limitations of conventional rule-based methods. The "Proposed DQN Discrete Scheduling" approach achieves the best performance, with the lowest cost of energy at $−10.59. This method demonstrates an 11% improvement over the "DQN on 3-Level Scheduling" and a 43% reduction compared to the "No Rule" approach. The results indicate that finer discretization in scheduling, as implemented in the proposed method, allows for more granular decision-making, thereby enhancing cost efficiency.

IV. CONCLUSION

The proposed discrete-action DQN method for scheduling battery contributions over defined periods was modelled and analysed through simulations. The results demonstrate that the DQN model effectively learns and adapts to optimize energy management, outperforming rule-based methods and alternative three-level scheduling DQN approaches. Simulation results reveal that implementing the DQN approach achieves a cost savings of 43% compared to a baseline without any rule-based approach and 21% more than the average rule-based strategy. Furthermore, the findings indicate that finer discretization of actions for scheduling can enhance cost efficiency, with an 11% improvement over the three-level DQN approach.

Overall, these results validate that optimizing the contribution of batteries to residential energy loads using the discrete-action DQN method can significantly reduce costs. However, the current analysis assumes deterministic load and generation data. Future work should focus on extending the model to handle the stochastic nature of real-world energy loads and generation, where forecasting uncertainties can impact performance. This will further enhance the robustness and practical applicability of the proposed approach in dynamic and unpredictable environments.

REFERENCES

[1] G. Strbac, "Demand side management: Benefits and challenges," Energy policy, vol. 36, no. 12, pp. 4419-4426, 2008.

[2] L. Steg, R. Shwom, and T. Dietz, "What drives energy consumers?: Engaging people in a sustainable energy transition," IEEE Power and Energy Magazine, vol. 16, no. 1, pp. 20-28, 2018.

[3] N. Nezamoddini and Y. Wang, "Real-time electricity pricing for industrial customers: Survey and case studies in the United States," Applied energy, vol. 195, pp. 1023-1037, 2017.

[4] P. Finn and C. Fitzpatrick, "Demand side management of industrial electricity consumption: Promoting the use of renewable energy through real-time pricing," Applied Energy, vol. 113, pp. 11-21, 2014.

[5] N. Javaid, A. Ahmed, S. Iqbal, and M. Ashraf, "Day ahead real time pricing and critical peak pricing based power scheduling for smart homes with different duty cycles," Energies, vol. 11, no. 6, p. 1464, 2018.

[6] M. R. Alam, M. St-Hilaire, and T. Kunz, "Computational methods for residential energy cost optimization in smart grids: A survey," ACM Computing Surveys (CSUR), vol. 49, no. 1, pp. 1-34, 2016.

[7] M. Khalid, "Smart grids and renewable energy systems: Perspectives and grid integration challenges," Energy Strategy Reviews, vol. 51, p. 101299, 2024.

[8] E. Sarker et al., "Progress on the demand side management in smart grid and optimization approaches," International Journal of Energy Research, vol. 45, no. 1, pp. 36-64, 2021.

[9] O. Babacan, E. L. Ratnam, V. R. Disfani, and J. Kleissl, "Distributed energy storage system scheduling considering tariff structure, energy arbitrage and solar PV penetration," Applied energy, vol. 205, pp. 1384-1393, 2017.

[10] F. V. Cerna and J. Contreras, "A MILP model to relieve the occurrence of new demand peaks by improving the load factor in smart homes," Sustainable Cities and Society, vol. 71, p. 102969, 2021.

[11] T. Logenthiran, D. Srinivasan, and T. Z. Shun, "Demand side management in smart grid using heuristic optimization," IEEE transactions on smart grid, vol. 3, no. 3, pp. 1244-1252, 2012.

[12] Y. Rajbhandari et al., "Enhanced demand side management for solar - based isolated microgrid system: Load prioritisation and energy optimisation," IET Smart Grid, vol. 7, no. 3, pp. 294-313, 2024.

[13] E. Mocanu et al., "On-line building energy optimization using deep reinforcement learning," IEEE transactions on smart grid, vol. 10, no. 4, pp. 3698-3708, 2018.

[14] Z. Wan, H. Li, H. He, and D. Prokhorov, "Model-free real-time EV charging scheduling based on deep reinforcement learning," IEEE Transactions on Smart Grid, vol. 10, no. 5, pp. 5246-5257, 2018.

[15] M. Ahrarinouri, M. Rastegar, and A. R. Seifi, "Multiagent reinforcement learning for energy management in residential buildings," IEEE Transactions on Industrial Informatics, vol. 17, no. 1, pp. 659-666, 2020.

[16] X. Xu, Y. Jia, Y. Xu, Z. Xu, S. Chai, and C. S. Lai, "A multi-agent reinforcement learning-based data-driven method for home energy management," IEEE Transactions on Smart Grid, vol. 11, no. 4, pp. 3201-3211, 2020.

[17] M. Shin, D.-H. Choi, and J. Kim, "Cooperative management for PV/ESS-enabled electric vehicle charging stations: A multiagent deep reinforcement learning approach," IEEE Transactions on Industrial Informatics, vol. 16, no. 5, pp. 3493-3503, 2019.

[18] A. Nagy, H. Kazmi, F. Cheaib, and J. Driesen, "Deep reinforcement learning for optimal control of space heating," arXiv preprint arXiv:1805.03777, 2018.

[19] Z. Liu, Z. Shen, S. Li, K. Helwegen, D. Huang, and K.-T. Cheng, "How do adam and training strategies help bnns optimization," in International conference on machine learning, 2021: PMLR, pp. 6936-6946.

[20] V. Mnih, "Playing atari with deep reinforcement learning," arXiv preprint arXiv:1312.5602, 2013.