Journal of Energy Storage 51 (2022) 104379

Contents lists available at ScienceDirect

Journal of Energy Storage

journal homepage: www.elsevier.com/locate/est

Research Papers

Reinforcement learning-based scheduling strategy for energy storage in microgrid

Kunshu Zhou [a], Kaile Zhou [a,*], Shanlin Yang [a,b]

[a] School of Management, Hefei University of Technology, Hefei 230009, China
[b] Key Laboratory of Process Optimization and Intelligent Decision-making of Ministry of Education, Hefei University of Technology, Hefei 230009, China

ARTICLE INFO

Keywords:
Microgrid
Energy storage scheduling
Deep learning
Reinforcement learning

ABSTRACT

Integrated energy microgrids (IEMs) have developed rapidly in the past years with the advancement of renewable energy and energy storage technologies. As a result, dealing with uncertainty on the source and load sides and optimizing energy storage scheduling in IEMs have become critical research issues. However, some existing methods have limitations in terms of solution accuracy and efficiency. In order to accurately grasp the uncertainty at both source and load sides while effectively reducing the cost and improving the computational efficiency, this study proposes an adaptive and lightweight algorithm to obtain the optimal scheduling strategy for energy storage. First, a modified deep learning method is proposed to predict the PV power and load demand. Then, based on the prediction results, a reinforcement learning algorithm is used to solve the energy storage scheduling model and obtain the optimal scheduling strategy. In addition, to further investigate the effects of greedy and non-greedy actions on the agent's training, this study compares the results under different action exploration policies and different time scales. The outcomes of this study were compared with the ones obtained by mixed-integer linear programming. The results show that the reinforcement learning algorithm reduces 61.17% of the solution time though loses 3.13% of the solution accuracy. The increase in computational efficiency is essential for the real-time energy storage applications.

1. Introduction

Energy storages are promising solutions to meet renewable energy consumption, reduce energy costs and improve operational stability for Integrated Energy Microgrids (IEMs) [1]. Particularly in the industrial park, the large-scale access to renewable energy represented by photovoltaic and the diversification of load types make the application of energy storage equipment more and more widespread [2]. However, the highly volatile and uncertain nature of the demand side of renewable energy generation and load poses a significant challenge to the operation of IEMs [3]. On the other hand, energy storage can divorce the temporal features of PV output from the cyclical variation of load demand, addressing the issues given by microgrids' unpredictability at both the source and load sides, balancing energy supply and demand at various times [4]. To enhance the efficiency of renewable energy consumption and lower total energy costs, it is critical to optimize the performance of energy storage in IEM.

Battery scheduling strategies have been addressed extensively in the literature with various design objectives. According to Wali et al. [5], the paradigm of energy storage and renewable energy integration is known to evolve quickly. Most research focused on experimental designs for energy storage capacity planning and operational optimization issues. Hannan et al. [6] summarized the methods for optimizing the operation of energy storage systems according to the different operational objectives of microgrids and discussed the advantages and shortcomings of batteries in their application. Lipu et al. [7] concluded that the effectiveness of battery management systems for battery control in electric vehicles depends mainly on the estimation of the efficiency of critical parameters and discuss the main concerns, research challenges and future progress of battery management systems in electric vehicles. Hannan et al. [8] adopted a deep convolutional god-general network to predict the state of charge of lithium batteries based on current and voltage data, which provides a basis for the safe and economic operation of batteries. Xiao et al. [9] used a dynamic planning approach to optimize the operation of a hybrid energy storage system in an integrated energy system. It is shown that the energy storage system can efficiently coordinate energy production and consumption. Nan et al. [10] used a data-driven dynamic planning algorithm to optimize a home rooftop PV

* Corresponding author.
E-mail address: zhoukaile@hfut.edu.cn (K. Zhou).

https://doi.org/10.1016/j.est.2022.104379
Received 14 January 2022; Received in revised form 20 February 2022; Accepted 5 March 2022
Available online 11 March 2022

system considering load and PV output uncertainty to minimize long-term power costs. The study showed that the proposed optimization algorithm can significantly improve the optimization results. Furthermore, the intelligent optimization algorithms have been frequently employed to handle energy storage optimization issues. Bouakkaz et al. [11] developed a particle swarm algorithm to optimize battery scheduling, demonstrating that an optimal scheduling strategy may save costs. Shan et al. [12] used an improved artificial swarm intelligence algorithm to optimize the operation of an IEM. The findings of the optimization revealed that the energy storage system could better balance the system's economy and environmental friendliness. Roslan et al. [13] proposed a microgrid energy controlling strategy based on a lightning search algorithm. The strategy overcomes the uncertainty problem in microgrid energy management, simplifies the complexity of constraints, and achieves the effect of cost reduction and environmental pollution reduction to a certain extent. However, most of the existing related studies are basic rule-based methods, including genetic algorithms and stochastic programming. It is difficult to choose appropriate models for an actual energy system and identify corresponding parameters that are energy storage dependent. Besides, many assumptions were made in these rule-based methods, which may do not apply to real-world situations.

With the development of artificial intelligence technology, deep learning (DL) and reinforcement learning (RL) have been widely used in decision-making problems. DL techniques have been widely used for PV generation and load demand forecasting, and specific methods include salp swarm algorithm-recurrent neural network-long short-term memory (SSA-RNN-LSTM) [14], gated recurrent unit (GRU) [15] and LSTM-convolutional neural network (LSTM-CNN) [16]. On the other hand, RL is widely used in studying energy storage arbitrage and energy storage controlling strategy issues. Han et al. [17] developed an energy storage arbitrage strategy based on reinforcement learning algorithms, taking into account the electricity price uncertainty. Xu et al. [18] proposed a hierarchical Q-learning algorithm to optimize the energy scheduling of electric vehicles. Cao et al. [19] used the NoisyNet-Dueling Deep Q-learning (NN-DDQN) algorithm to obtain an energy storage arbitrage strategy, taking into account battery degradation. Although RL has been used to solve energy storage controlling strategy issues, it is still worth investigating to reduce the uncertainty at both source and load ends and improve computational efficiency.

In general, existing research on energy storage scheduling strategies taken less account of the following three points. First, the commonly used rule-based algorithm has complex modeling problems, has slow solution efficiency, and easily falls into the local optimum. It cannot adapt to the actual application scenarios with variable operating conditions [20–22]. Second, the choice of greedy or non-greedy actions in the training process of the Q-learning algorithm has an impact on the training efficiency and needs to be investigated. Finally, the influence of the external environment, i.e., the uncertainty of both source and load, is less considered when obtaining the energy storage scheduling strategy [23–25].

To improve the computational efficiency of the scheduling algorithm, this study proposed a DGRU-QL algorithm capable of adaptive online learning to solve the optimal scheduling strategy for energy storage. The main contributions of this study compared to previous works are as follows.

(1) A modified deep learning approach using the swish activation function is used to accurately measure PV generation and load demand and to evaluate the influencing factors.
(2) A model-free, lightweight, data-driven adaptive reinforcement learning algorithm is proposed to solve the optimal scheduling strategy for energy storage, which satisfies the real-time online strategy solution for energy storage, reduces the influence of uncertainty at both source and load sides, and improves the solution efficiency.
(3) The effects of greedy and non-greedy actions on the training results are examined using two action exploration policies. Moreover, the proposed algorithms are evaluated from solution accuracy and solution efficiency perspectives.

The overall structure of the article is as follows, and Section 2 introduces the structure of the IEM, and relevant research methods. Section 3 presents the data, the process, and the results of the experiment. Section 4 gives the overall research conclusions.

2. DGRU-QL algorithm

2.1. Structure of the IEM

The IEM usually consists of distributed power sources, energy storage devices, energy conversion devices, loads, monitoring, and protection devices for small-scale power generation and distribution systems. The purpose and significance of its construction are to realize the flexible and efficient application of distributed power supply and solve the problems of high load demand and increased energy stability requirements [26]. To successfully verify the effectiveness of the algorithm proposed in this study for solving the energy storage scheduling strategy in a realistic scenario, a typical IEM containing a photovoltaic power plant, a combined heat and power (CHP) unit, and a lithium battery for energy storage is selected for this study. Fig. 1 shows the structure of the IEM.

Energy storage technology has developed rapidly in recent years. It has evolved from small-capacity, small-scale research and application to large-capacity, large-scale research and application of energy storage systems. Lithium-ion batteries have been widely chosen in IEMs for their flexibility in assembly, stable energy supply, fast response time, and low self-discharge rate compared with other energy storage technologies [27].

2.2. Workflow of the DGRU-QL algorithm

The energy storage scheduling is a typical sequential decision problem suitable for solving with reinforcement learning algorithms [28]. In this study, the optimal scheduling process of energy storage is transformed into a Markov decision process (MDP). Considering that the scheduling time of this study is one day and the overall data volume is moderate, to ensure the solution efficiency and accuracy, this study adopts the classical model-free algorithm Q-learning in reinforcement learning to explore the optimal strategy of energy storage scheduling, and two action exploration policies are used for comparison. Compared to continuous reinforcement learning algorithms, Q-learning algorithms are more suitable for handling MDP problems with moderate amounts of data and simple action spaces and can improve computational efficiency without compromising computational accuracy [29]. This method can be better adapted to the difficult situation of system modeling during the daily operation of IEM and can reduce the computational complexity while improving the optimal scheduling effect. The user only needs to enter the IEM state to obtain the optimal charging and discharging strategy for the energy storage, without the need for a detailed description of the constraint formulas for each component in microgrid, as in traditional optimization algorithms before optimal scheduling. The strategies obtained through training can be adapted to the optimal IEM scheduling in various scenarios without recalculation when the operation scenario is changed, which is more flexible and more convenient for realizing the optimal scheduling microgrid in real-time [30]. In addition, unlike traditional scheduling algorithms, reinforcement learning algorithms are model-free scheduling methods that do not require a priori knowledge and specific models of the system, and can better solve the decision-making problem by only learning from the interaction between the agent and the environment, obtaining feedback from the environment during the learning process and maximizing the

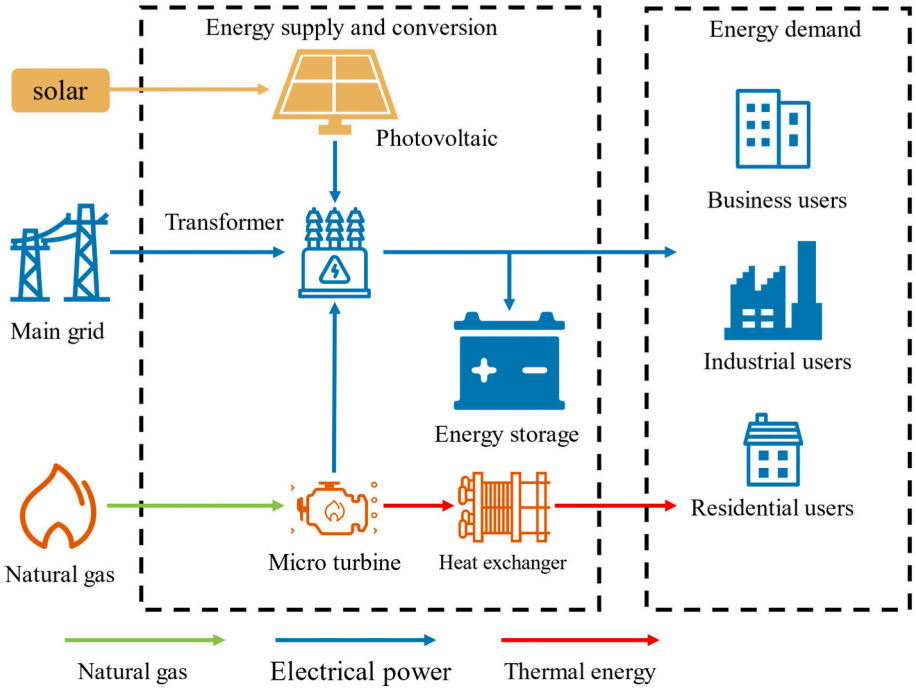

Fig. 1. Structure of the IEM of the industrial parks.

accumulated reward to obtain the optimal policy [31].

2.2.1. Mathematical framework

The DGRU-QL algorithm used in this study is implemented in two steps, firstly, a deep GRU neural network model with higher efficiency considering time dependence is used to predict the PV power and electric load demand, and the prediction results are applied to the process of solving the energy storage of microgrid scheduling strategy, and then the Q-Learning algorithm based on two action exploration policies is then used to solve the energy storage scheduling strategy. The overall flow of the algorithm is shown in Fig. 2.

Photovoltaic power generation is more susceptible to weather and climate conditions, and there is a certain periodicity and linear correlation between the two. The electric load demand is closely related to the electricity consumption behavior of users, and there is a strong transient and non-linear relationship between them, so the temporal correlation in the data needs to be considered when constructing the prediction model.

The LSTM is a particular structure of RNN, which solves the problems of gradient disappearance and gradient explosion that occur during the use of RNNs by introducing gate mechanisms and memory units [32]. GRU is a particular form of LSTM. Unlike LSTM with three gating units, GRU uses two gating units to control the information flow, i.e., update and reset gate. Input gate and forget gate in LSTM are combined into update gate in GRU, and update gate is used to control the update of the hidden state [33]. The reset gate is used to decide whether to ignore the previous hidden state. The structure diagram is shown in Fig. 3, and the GRU is updated in the following way.

$$u_t = \sigma(W_{xu} \bullet \mathrm{x_i} + \mathrm{W_{hu}} \bullet \mathrm{h_{t-1}} + \mathrm{b_u}) \tag{1}$$

$$r_t = \sigma(W_{xr} \bullet \mathrm{x_i} + \mathrm{W_{hr}} \bullet \mathrm{h_{t-1}} + \mathrm{b_{ur}}) \tag{2}$$

$$\widetilde{h}_t = tanh(W_{xh} \bullet \mathrm{x_i} + \mathrm{W_{hh}} \bullet (\mathrm{r_t} \odot \mathrm{h_{t-1}}) + \mathrm{b_h}) \tag{3}$$

$$h_t = u_t \bullet \mathrm{h_{i-1}} + (1 - \mathrm{u_t}) \odot \widetilde{\mathrm{h}}_\mathrm{t} \tag{4}$$

As a particular form of LSTM, GRU adopts a gated recurrent neural network structure with a more straightforward network structure, fewer training parameters, and a faster training process with similar prediction results [34].

Photovoltaic power generation and load capacity are affected by a variety of external factors [35], so actual data of influential factors such as temperature, humidity, visibility, body surface temperature, pressure, wind speed, rainfall, and rainfall probability are collected. The input variables of the final neural network were determined by calculating the Spearman correlation coefficients and Pearson correlation coefficients of each influencing factor and the predictor variables.

The Pearson correlation coefficient can effectively test the linear relationship between two continuous variables and is calculated as in Eq. (5).

$$r_{Pearson} = \frac{\sum_{i=1}^{N}(x_i - \bar{x}) \bullet (y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 \bullet \sum_{i=1}^{N}(y_i - \bar{y})^2}} \tag{5}$$

In Eq. (5), $\bar{x}$ and $\bar{y}$ denote the mean values of variables x and y, respectively.

The Spearman correlation coefficient detects correlations between variables that do not satisfy the conditions for using the Pearson correlation coefficient and is calculated as follows.

$$r_{Spearman} = \frac{\sum_{i=1}^{N}(R_i - \overline{R}) \bullet (S_i - \overline{S})}{\sqrt{\sum_{i=1}^{N}(R_i - \overline{R})^2 \bullet \sum_{i=1}^{N}(S_i - \overline{S})^2}} \tag{6}$$

In Eq. (6), R_i and S_i denote the ranks of the values taken for observation i, respectively, $\overline{R}$ and $\overline{S}$ denote the average ranks of variables x and y, respectively.

Reinforcement learning is based on the Markov decision process (MDP), i.e., the next state of the system is only related to the current state but not to the previous state [36]. In solving, the problem needs to be represented by a quadruple (S, A, R, π), where S denotes the set of states, A denotes the set of actions, R denotes the reward function, and π denotes the set of agent action policies. During the learning process, the agent interacts with the environment by trial and error to perform the learning of action policy sets that maximize the cumulative rewards the agent obtains during the interaction with the environment [37]. As shown in Fig. 4, in the initial stage, the agent is in state S_0, According to

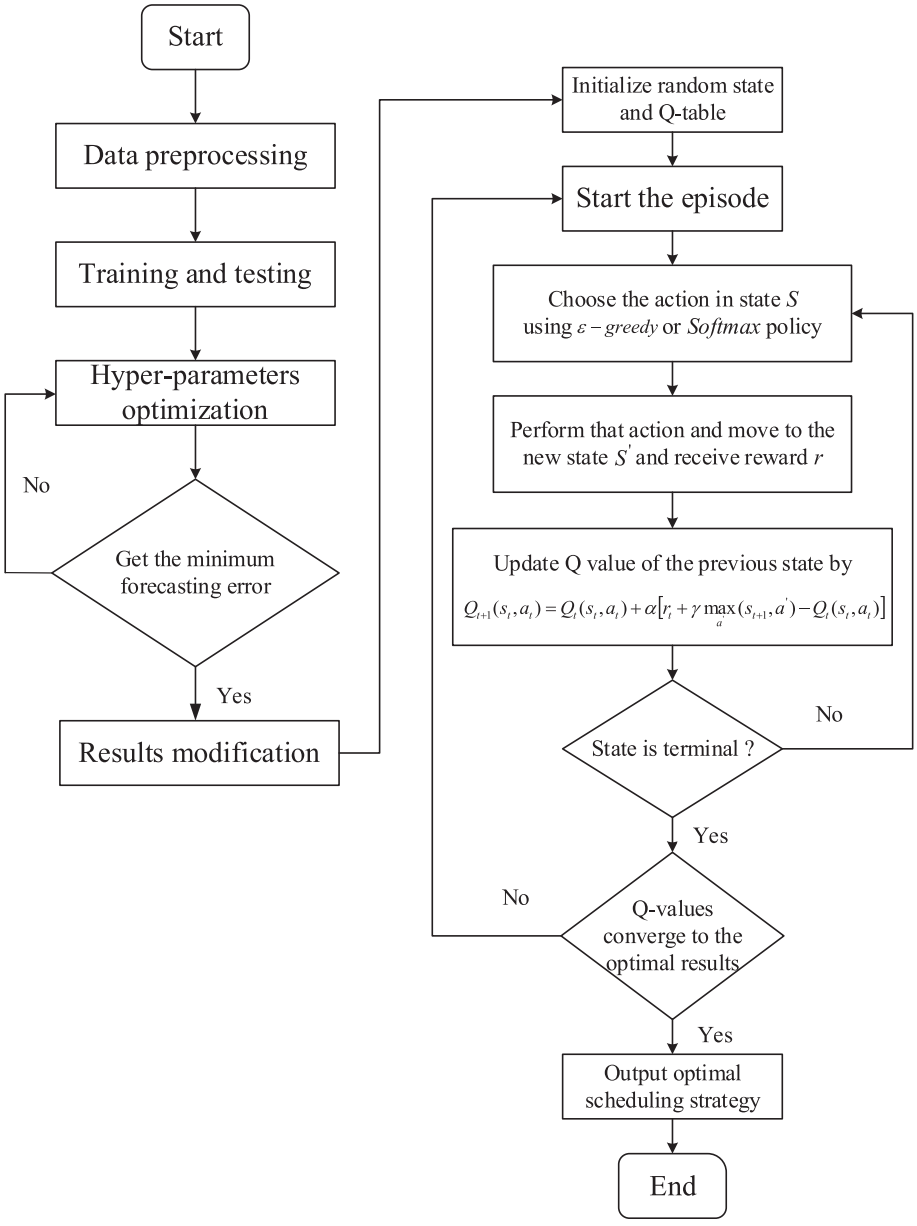

Fig. 2. Flowchart of DGRU-QL algorithm to solve the IEM energy storage scheduling strategy.

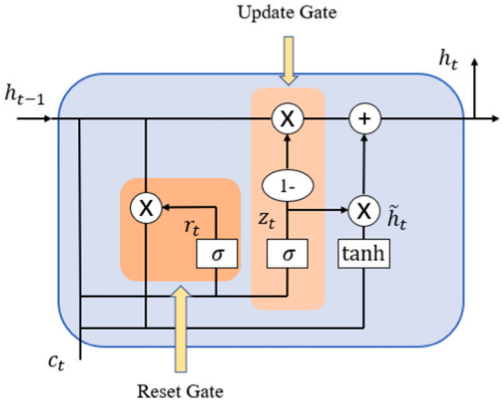

Fig. 3. GRU recurrent neural network unit.

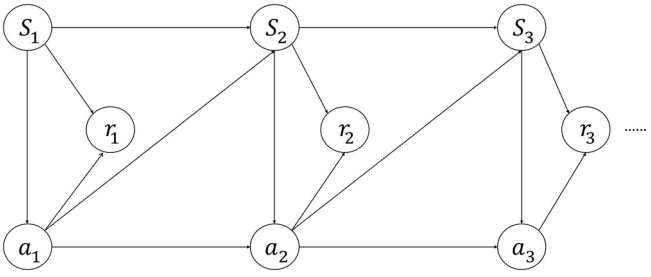

Fig. 4. Markov decision process.

the policy, action a_0 is selected. After executing action a_0, the agent moves to the next state S_1 and gets the reward r_0. The following action a_1 is executed until the set conditions are satisfied. Reinforcement learning can determine the optimal action choice of agent by calculating the maximum cumulative reward obtained during its continuous interaction

with the environment, enabling the optimization of sequential decisions in a restricted feedback process. Because of its advantages of no model dependency and scalability, reinforcement learning is gradually widely used in energy system management problems. Zoltowska et al. [38] explored an energy storage of microgrid scheduling strategy containing PV systems using reinforcement learning algorithms. Abedi et al. [4] calculated the impact of reinforcement learning-based scheduling strategies on the overall cost of electricity consumption in households. Jayaraj et al. [39] gives a 24-hour energy storage scheduling scheme for microgrids containing PV generation systems and energy storage using reinforcement learning algorithms to achieve economic scheduling and reduce the net transaction cost of electricity.

Q-learning is a model-free, value-based reinforcement learning algorithm [40] that evaluates the merit of a strategy in terms of a state-value function $Q(s,a)$. When the current state $s$and action a are known, the state-value function can be expressed as.

$$Q(s,a) = E_\pi\left(\sum_{t=1}^{T}\gamma^t r_t | s_t = s, a_t = a\right) \tag{7}$$

In Eq. (7), $E_\pi(\bullet)$ denotes the long-term expected return under strategy π, t denotes the number of iteration steps, γ is the discount factor, which usually takes a value between 0 and 1 and indicates the importance of future returns relative to current returns. r_t denotes the immediate reward at step t.

During the learning process, the Q values are updated according to the Bellman equation, i.e.

$$Q_{t+1}(s_t,a_t) = Q_t(s_t,a_t) + \alpha\left[r_t + \gamma \underset{a'}{max}(s_{t+1},a') - Q_t(s_t,a_t)\right] \tag{8}$$

In Eq. (8), α denotes the learning rate, $r_t + \gamma \underset{a'}{max}(s_{t+1},a')$ denotes the maximum Q value that can be obtained in the next state, which is usually called the target Q value, and $Q_t(s_t,a_t)$ is called the estimate of Q. By iteratively updating Eq. (8) continuously, the Q value converges to the convergence.

In order to enable the agent to both make full use of what has been learned and thoroughly explore the unknown environment during training, different exploration policies are usually used to select the actions to be performed during training, and the commonly used policy is the *ε-greedy* [41], which takes the following form.

$$a = \begin{cases} argmax_{a\in A}, \rho \geq 1-\varepsilon \\ random(A)\ , \rho < \varepsilon \end{cases} \tag{9}$$

In Eq. (9), ε denotes the exploration probability, which takes values between 0 and 1. ρ is a random number ranging from 0 to 1, and *random* (A) denotes a randomly selected action in the action space A.

Intuitively, the *ε-greedy* policy gives a more significant probability to the action corresponding to the most prominent value function, i.e., $1-\varepsilon+\frac{\varepsilon}{|A|}$, while the other actions are sampled with equal probability, i.e., $\frac{\varepsilon}{|A|}$, regardless of the size of the corresponding value function. However, theoretically, there are good and bad non-greedy actions, and actions with large correspondence value functions should have a higher probability of being sampled than those with small correspondence value functions. Therefore, this study uses the *Softmax* exploration policy to compare the results with the *ε-greedy* policy. The *Softmax* policy can soft process the probability of the action according to the value function, and the specific *Softmax* policy for the probability of action selection is as follows [42].

$$p(a_i) = \frac{exp\left(\frac{Q(a_i)}{\tau}\right)}{\sum_{j=1}^{k}\left(\frac{Q(a_i)}{\tau}\right)} \tag{10}$$

In Eq. (10), τ indicates the temperature regulation coefficient, which regulates the proportion of exploration and utilization. The smaller τ, the more significant the proportion of actions selected to correspond to a large value function. The larger τ, the closer to uniform sampling.

2.2.2. Modified deep learning and optimization model

The neural network structure proposed in this study contains an input layer, fully connected layer, GRU layer, dropout layer, and an output layer, as shown in Fig. 5 [43]. The input layer contains 96 neurons, each of which is a sequence of 96 in length containing time variables and weather data. The dropout layer changes the network structure by removing a certain percentage of neurons, thus adaptively adjusting the learning ability of the model and making the learning effect more robust. The output layer also has 96 neurons and is used to output the prediction results.

The implementation of the prediction model is mainly divided into three steps: data pre-processing, model training, and model evaluation. In order to obtain better prediction results, firstly, the training data are normalized to reduce the data dimensionality and improve the computational efficiency [44]. For a given variable x, the normalization process can be expressed as

$$x_i' = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{11}$$

In Eq. (11), x_{min} denotes the minimum value of variable x, x_{max} denotes the maximum value of variable x, and x_i' denotes the result of normalization of variable x_i, The dropout layer is also added to the model to prevent overfitting. Finally, an Adam optimizer is used to train the model.

In this study, three indicators were selected to assess the predictive effect of the model, i.e. root mean square error (RMSE), mean absolute error (MAE), and Pearson correlation coefficient (PCC). Their definitions are shown in Eq. (12), Eq. (13) and Eq. (5) respectively.

$$\mathrm{RMSE} = \sqrt{\frac{1}{m}\sum_{m=1}^{M}\left(y_{actual} - y_{forecasting}\right)^2} \tag{12}$$

$$\mathrm{MAE} = \frac{1}{m}\sum_{m=1}^{M}\left|y_{actual} - y_{forecasting}\right| \tag{13}$$

In Eq. (12) and Eq. (13), M denotes the time step, y_{actual} denotes the actual value in the test set, and $y_{forecasting}$ denotes the corresponding predicted value.

In this study, the charging and discharging strategy of the battery is optimized to minimize the operating cost of the microgrid while meeting the energy demand. Therefore, this research problem can be viewed as an optimization problem with the following objective function.

$$min\left[p_{g,t}\bullet\sum_{t=1}^{T}\left(\mathrm{V_{chp,t}}\right) + \mathrm{p_t}\bullet\sum_{t=1}^{T}\left(\mathrm{P_{grid,t}}\right) + \mathrm{C_{cs}}\bullet\sum_{t=1}^{T}\left(\left|\mathrm{p_{dis,t}}\right| + \left|\mathrm{p_{char,t}}\right|\right)\right] \tag{14}$$

Eq. (14) represents the operating cost of the IEM, which contains three items. The first item is the gas purchase cost, where $p_{g,t}$ in Eq. (14) denotes the gas price at time t and $V_{chp,t}$ denotes the gas purchase volume. The second term is the cost of electricity purchase, p_t represents the price of electricity at time step t, and $P_{grid,t}$ represent the amount of electricity purchased at time step t. The third term is the cost of battery charging and discharging. C_{cs} denotes the charging and discharging cost per unit of battery, $p_{dis,t}$ denotes the amount of battery discharging at time step t, and $p_{char,t}$ denotes the amount of battery charging at time step t.

To ensure the safety and reliability of energy use, the IEM needs to meet the power balance constraint and battery capacity constraint at all times during the operation process, as expressed below.

(a) Electricity supply and demand balance

$$P_{grid,t} + P_{chp,t} + P_{pv,t} + \left|p_{dis,t}\right| - \left|p_{char,t}\right| = P_{load,t} \tag{15}$$

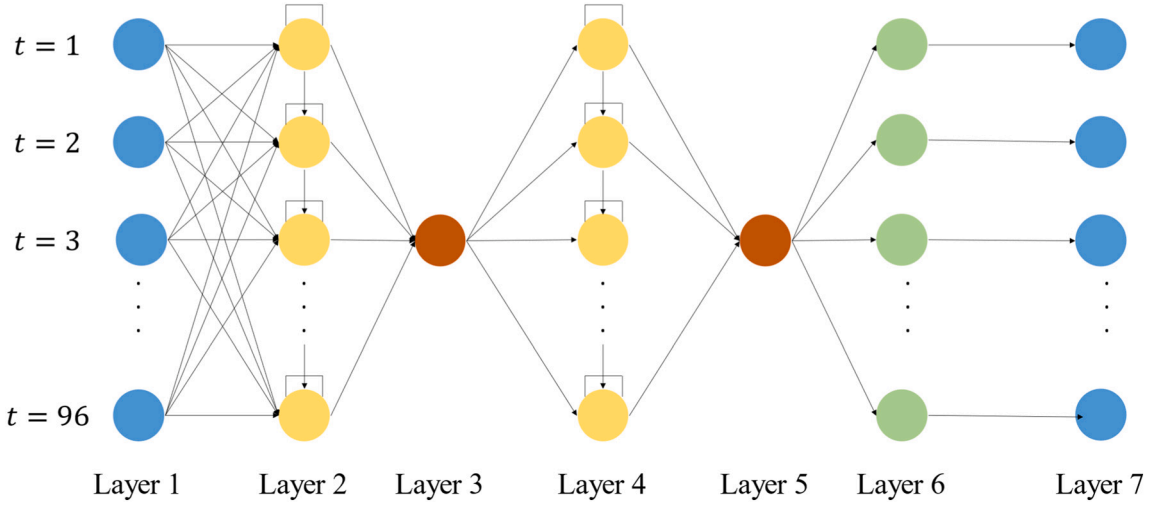

Fig. 5. Neural network structure for prediction.

In Eq. (15), $P_{load,\ t}$ denotes the microgrid electrical load demand at time t, and $P_{chp,\ t}$ denotes the generation capacity of the CHP unit at time step t. The formula is calculated as follows.

$$P_{chp,t} = V_{chp,t} \bullet q_{NG} \bullet \eta_{chp} \tag{16}$$

In Eq. (16), q_{NG} denotes the low calorific value of natural gas, and η_{chp} denotes the electrical efficiency of the CHP unit.

(b) Heat supply and demand balance

$$V_{chp,t} \bullet q_{NG} \bullet \left(1 - \eta_{chp}\right) = H_{load,t} \tag{17}$$

In Eq. (17), $H_{load,\ t}$ denotes the microgrid heat load demand at time step t.

(c) battery charging and discharging capacity constraint

$$SOC^{min} \le SOC_t \le SOC^{max} \tag{18}$$

In Eq. (18), SOC_t represents the capacity state of the battery at time step t, and SOC^{min} and SOC^{max} represent the upper and lower limits of the battery capacity, respectively.

In this study, the above optimization problem is transformed into a data-driven, model-free, value-based Q-learning algorithm solution problem, which learns the optimal action (i.e., optimal scheduling strategy) through the continuous interaction between the agent and the environment to explore the charging and discharging actions of the battery at each scheduling time step of the day in order to achieve the lowest operating cost. As shown in Fig. 6, the agent represents the energy storage, and the microgrid represents the environment where the agent is located. In each scheduling time step, the agent receives the state information provided by the environment, takes the corresponding charging and discharging actions, and gives feedback to the current microgrid environment. The optimal charging and discharging strategy are finally obtained by continuous iterative learning of the state action-value function.

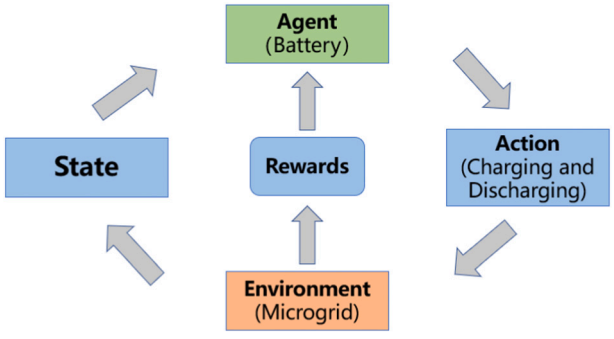

Fig. 6. Schematic diagram of reinforcement learning based battery scheduling strategy optimization.

When the Q-learning algorithm is used to obtain the battery charging and discharging strategy, the elements in the MDP quaternion (S, A, R, π) correspond as follows.

The state space S should contain the external information perceived by the agent in the training environment. The construction of the state space affects the convertibility of the algorithm and the final optimization results. For the IEM, the state space provided by the environment to the agent generally includes the battery SOC, PV output information, electricity price information, natural gas price information, and electricity and heat load information. Therefore, the state space in this study can be defined as follows.

$$S = \left[P_{grid,t}, SOC_t, P_{pv,t}, P_{load,t}, H_{load,t}, p_t, p_{g,t}\right] \tag{19}$$

In Eq. (19), $P_{pv,\ t}$ denotes the PV power at time step t.

The action space A consists of the battery charging and discharging action, which can be defined as follows.

$$A = \left[p_{ess,t}\right] \tag{20}$$

The Q-learning algorithm requires discretization of the action space when processing continuous actions, i.e.

$$\begin{cases} p_{ess,t} = \left[p^1_{ess,t}, p^2_{ess,t}, \cdots p^K_{ess,t}\right] \\ p^1_{ess,t} = p^{max}_{dis,t}, p^K_{ess,t} = p^{max}_{char,t} \\ p^K_{ess,t} - p^{K-1}_{ess,t} = \dfrac{\left|p^{max}_{dis,t}\right| + \left|p^{max}_{char,t}\right|}{K-1} \end{cases} \tag{21}$$

In Eq. (21), $p_{ess,\ t}{}^K$ indicates the Kth charge and discharge action in the discrete action space, $p_{ess,\ t}{}^1$, $p_{ess,\ t}{}^2$, $\cdots p_{ess,\ t}{}^K$ in ascending order, the first $p_{ess,\ t}{}^1$ value is the maximum discharge power, and the last $p_{ess,\ t}{}^K$ value is the maximum charge power.

The reward function R represents the feedback obtained by an agent after executing an action in the environment and is the key to guiding the agent to achieve its goal. The reward function should be set to include the system reward when the agent performs the correct action and the system penalty when it performs the wrong action and satisfies the fundamental constraints of each device in the microgrid. Therefore, the reward function of the Q-learning-based optimization model for the

charging and discharging strategy of battery is as follows.

$$r = -(C + D + E) \tag{22}$$

In Eq. (22), C denotes the gas purchase cost and the battery charging and discharging cost is as follows.

$$C = p_{g,t} \bullet \sum_{t=1}^{T} (V_{chp,t}) + C_{cs} \bullet \sum_{t=1}^{T} (|p_{dis,t}| + |p_{char,t}|) \tag{23}$$

In Eq. (23), D represents the penalty when the energy supply and demand do not match during the operation of the microgrid. In this study, the microgrid is dispatched by the "determining power by heat" strategy, and when the total load demand is greater than the total power supply, the power needs to be purchased from the grid. When the total load demand is less than the total power supply, the excess power supply is stored in the battery through charging action. The penalty can be equated to the power purchased by the microgrid from the larger grid. When PV and battery power cannot meet the microgrid electrical load demand, the microgrid meets the usage demand by purchasing power from the larger grid is as follows.

$$D = \begin{cases} P_{grid,t} \times p_t \ , \ P_{load,t} - (P_{pv,t} + P_{chp} - |p_{dis,t}| + |p_{char,t}|) \\ > 0\,0 \ , \ P_{load,t} - (P_{pv,t} + P_{chp} - |p_{dis,t}| + |p_{char,t}|) \le 0 \end{cases} \tag{24}$$

In Eq. (24), E denotes the penalty when the battery SOC exceeds the limit. The battery must always meet the upper and lower capacity limits, with a penalty function of the specific form as follows.

$$E = \begin{cases} 0, SOC^{min} \le SOC_t \le SOC^{max} \\ 100, SOC_t < SOC^{min} \text{ or } SOC_t > SOC^{max} \end{cases} \tag{25}$$

Eq. (25) indicates that the penalty is 0 when the battery capacity is kept in a reasonable operating range and a more enormous value of 100 when the agent selects a wrong action that causes the battery capacity to be lower than the minimum capacity or higher than the maximum capacity to ensure proper battery operation.

3. Results and discussion

3.1. Data

The data used for PV power prediction are from the public dataset DKA Solar Center, which contains complete PV data and weather information, including temperature, humidity, global horizontal radiation, diffuse horizontal radiation, wind direction, wind speed, rainfall, radiation global tilted, and radiation global tilted radiation, wind direction, wind speed, rainfall, and other actual data. A total of 124 days of data were used in the experiment, with the first 95 days as training data and the last 29 days as test data. The load data were obtained from 363 days of actual data from an industrial park, of which the first 353 days of data were used as training set data for training neural network parameters, and the last 10 days of data were used as test set data.

By calculating the correlation coefficients between the influencing factors and the predicted objects, the more important influencing factors can be further selected for prediction. The results of the two correlation coefficient calculations are shown in Fig. 7.

The Fig. 7(a) shows the Pearson correlation coefficient calculation results between PV power and influencing factors. The Fig. 7(b) shows the results of the Spearman correlation coefficient calculation between PV power and influencing factors. From the calculation results, temperature, humidity, wind speed, global horizontal radiation, diffuse horizontal radiation, and radiation global tilted can be filtered as input variables for the final prediction neural network. In addition, the prediction time interval was chosen to be 15 min. In addition, the prediction time interval is determined to be 15 min.

The results of the correlation coefficient calculation between the load and the influencing factors are shown in Fig. 8. The results make it possible to identify temperature, humidity, visibility, body surface temperature, and wind speed as input variables for the neural prediction network. The same 15-minute prediction interval is chosen.

The time-of-use electricity and gas tariffs are shown in Fig. 9. The experimental design scheduling cycle is one day, and the whole scheduling cycle is divided into three time intervals of 24, 48, and 96. The battery scheduling strategies for the three scheduling time intervals are solved and compared respectively.

3.2. Parameters setup

This study's deep learning and reinforcement learning parts were performed using python 3.6, 2.6 GHz Intel Core i5 processor and 16 GB RAM, and the MILP part was done using Matlab software with the same hardware conditions. The search space of the relevant hyperparameters in the predictive neural network is given in Table 1.

The number of neurons in each layer of the neural network was set to range between 96 and 288, depending on each layer's input and output length. To avoid the gradient disappearance and gradient explosion problems, GRU is chosen for layer two and layer four neuron types to consider the temporal correlation in the data. Meanwhile, to improve the prediction accuracy, the Swish function is chosen as the activation function for the remaining layers except for the last fully connected layer where the sigmoid function is chosen. Swish activation function with the original formula:$f(x) = x \bullet sigmoid(x)$, which has the characteristics of the lower bound, no upper bound, and non-monotonic [45], also outperforms other activation functions in the experiments. In addition, the

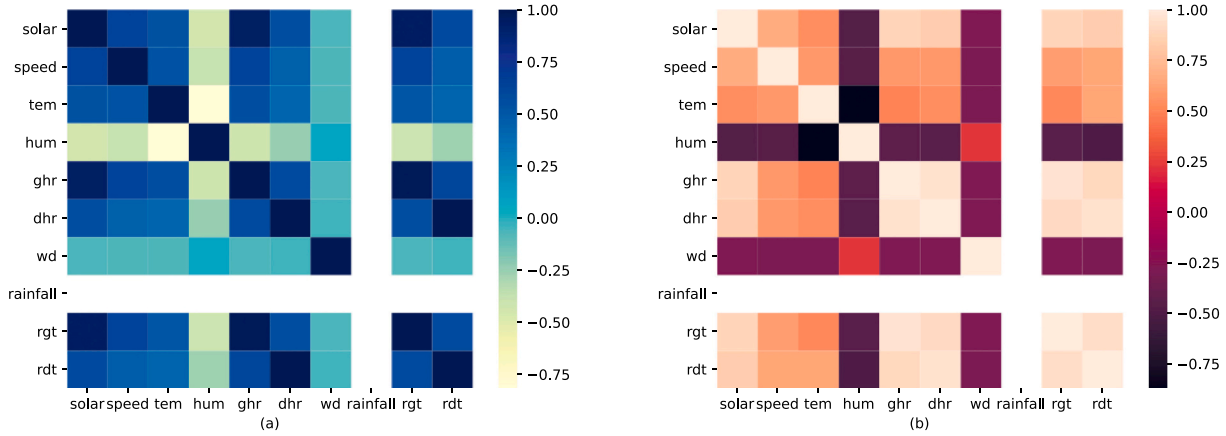

Fig. 7. PV power prediction correlation coefficient.

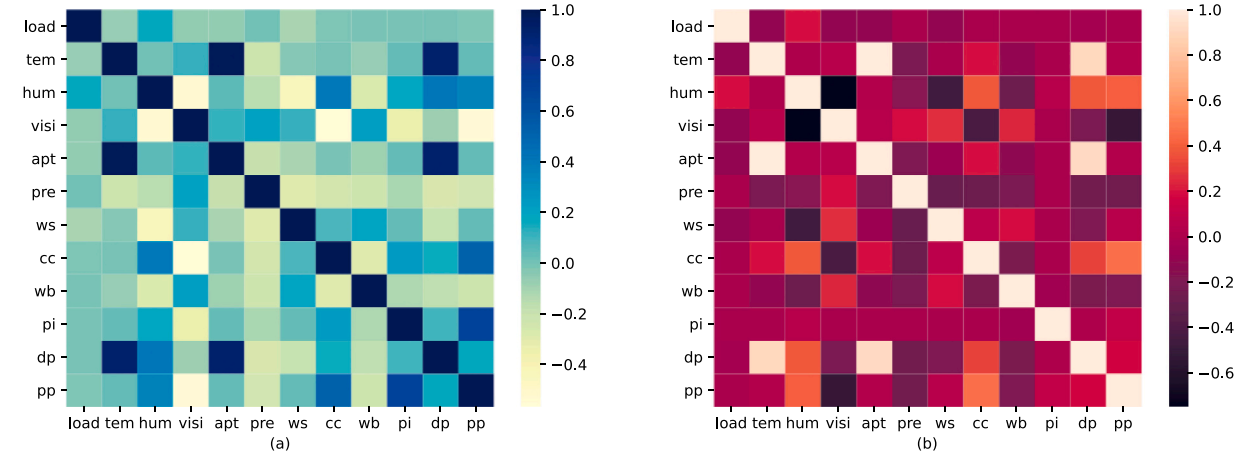

Fig. 8. Load prediction correlation coefficient.

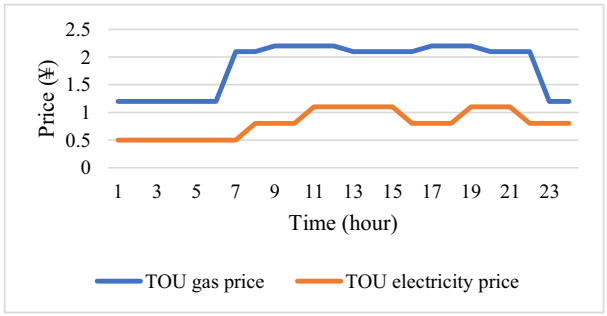

Fig. 9. Time of use gas and electricity price.

Table 1
Ranges of hyper-parameters.

Hyper-parameters	Range
Neuron numbers of layers	[96, 288]
Neurons type of layers 2 and 4	[GRU, Simple RNN]
Activation function of layers	[swish, sigmoid]
Dropout of layers 2, 3, 4 and 5	[0.2, 0.5]

learning rate decay technique is used in the model training to set the learning rate to 40% of the initial learning rate every two hundred steps to reduce the risk of model overfitting further [46].

The parameters of the devices in the IEM are shown in Table 2.

Table 3 gives the parameter settings for the training process of the Q-Learning algorithm. Since the scheduling period is short in this experiment, the discount factor γ takes the value of 0.99, the learning rate α takes 0.1, ε takes 0.1 in the ε-*greedy* exploration policy, the temperature coefficient τ takes 10 in the *Softmax* exploration policy. The number of iterative rounds takes 30,000.

Since the Q-learning algorithm requires the action space to be discretized during calculation, the battery charge and discharge power in this study are set to the rated power of 2 kW and two times the rated power of 4 kw, and according to Eqs. (20) and (21), the action space can be discrete into five actions.

$$A = [-4\ ,\ -2\ ,\ 0\ ,\ 2\ ,\ 4] \tag{26}$$

Also, the battery SOC can be discrete into 10 states according to the discrete form of the action space. The battery parameters and Eq. (19) can be discrete into 10 states.

$$\mathrm{SOC} = [6\ ,\ 8\ ,\ 10\ ,\ 12\ ,\ 14\ ,\ 16\ ,\ 18\ ,\ 20\ ,\ 22\ ,\ 24] \tag{27}$$

Table 2
Industrial parks IEM equipment parameters.

Parameters	Value
Battery capacity (kWh)	30
Charging efficiency	0.8
Discharging efficiency	0.8
Unit charge and discharge cost (¥/kWh)	0.01
Efficiency of CHP	0.35
Calorific value of natural gas (kWh/m^3)	9.7

Table 3
Q-learning algorithm training parameters.

Parameters	Value
Discount factor γ	0.99
Learning rate α	0.1
Exploration ratio ε	0.1
Softmax temperature parameter τ	10
Episodes	30,000

3.3. Forecasting results

The DGRU-NN model used in this study forecasts the power generation of a photovoltaic plant located in the Alice Spring Desert region of Australia. A total of 124 days with 15 min of data per day were collected, and the first 95 days of data were used as the training set data to train the model, and the last 29 days of data were used as the validation set data to verify the model effect. Fig. 10 shows the trend of the loss function during the model's training and the model's forecasting effect on the test set. From Fig. 10, it can be seen that the loss function decreases faster after the model starts training and converges for 200 steps, which proves that the model is trained well. In addition, it can be seen that the PV power generation data has a strong periodicity. Generally, in the absence of light during 20:00–05:00, PV output power is 0. At about 12:00, PV output power reached the maximum. On the other hand, it can be seen from the figure that there are fluctuations in the PV power curve during some periods under a general trend with a strong periodicity, and the more likely cause of such fluctuations in the cloud cover shading.

The forecasted load data are obtained from 363 days of actual data of an industrial park, of which the first 353 days of data are used as training set data to train the neural network parameters, and the last 10 days of data are used as test set data to verify the training results. Fig. 11 shows the trend of the loss function during the model's training and the model's forecasting effect on the test set. It can be seen that the loss function decreases rapidly after starting the training and tends to converge since 200 steps. The electric load curve is more stochastic and volatile

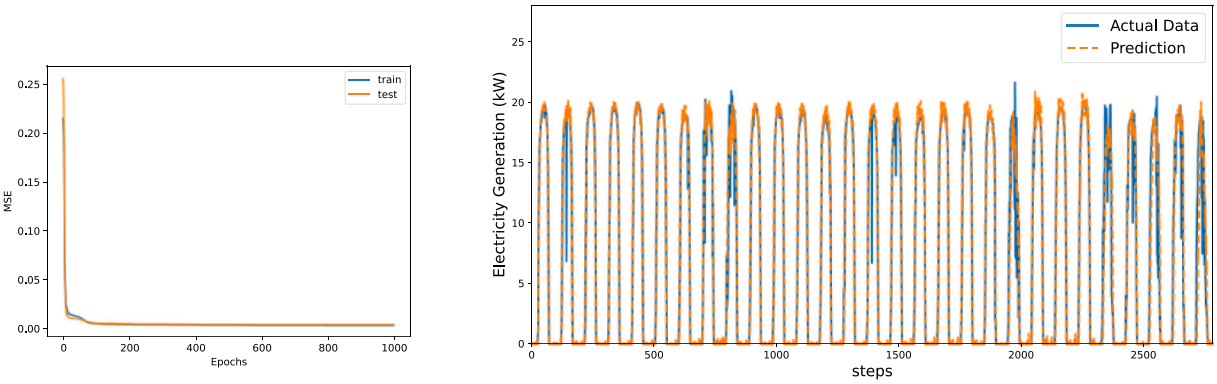

Fig. 10. Trends in PV power forecasting model loss functions and forecasting results.

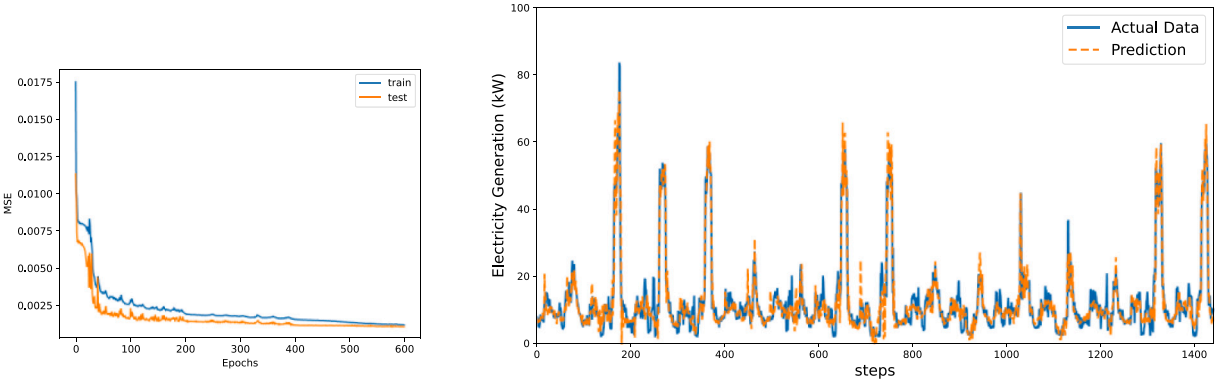

Fig. 11. Trends in load demand forecasting model loss functions and forecasting results.

compared to the PV generation curve. In terms of trend, the electric load demand is higher during the day and lower at night, and the forecast curve fits better with the actual value curve.

The evaluation indexes of the forecasting model are shown in Table 4, and the model is effective in predicting the PV generation and load demand in the IEM.

Overall, the proposed DGRU-NN model can effectively forecast the PV power generation and electric load demand at both the supply and demand sides of the IEM, which can provide the possibility of integrating renewable energy and reducing the energy cost for microgrid users. The forecasting results derived from the experiments in this section will provide the basis for solving the battery scheduling strategy, thus effectively promoting the interaction and balance between the supply and demand sides of the IEM.

3.4. Energy storage scheduling strategy optimization results

An accurate scheduling strategy enables the battery to sense the changes of electricity price, gas price, load, and PV output in the IEM in real-time and adjust the battery charging and discharging behavior according to the changes promptly to minimize the total operating cost. This section will explore the battery scheduling strategy from two exploration policies and two-time scale perspectives based on the previous prediction results to accurately grasp the PV output and load change patterns.

Table 4
Forecasting model evaluation metrics.

Index	RMSE	MAE	PCC
PV forecasting model	1.195	0.682	0.98285
Load forecasting model	3.399	2.489	0.95288

3.4.1. Action exploration policy description

To thoroughly investigate the solution results of the Q-learning algorithm based on two different action selection policies for the battery scheduling strategy and to evaluate the performance of obtaining scheduling results under different time scale conditions, this experiment examined six different strategies, which are as follows.

Strategy 1: Select the *ε-greedy* action exploration policy with 15 min scale.
Strategy 2: Select the *ε-greedy* action exploration policy with 30 min scale.
Strategy 3: Select the *ε-greedy* action exploration policy with 1 h scale.
Strategy 4: Select the *Softmax* action exploration policy with 15 min scale.
Strategy 5: Select the *Softmax* action exploration policy with 30 min scale.
Strategy 6: Select the *Softmax* action exploration policy with 1 h scale.

3.4.2. The results of battery scheduling strategy

Based on the definition of the above six strategies, the optimal battery scheduling strategy for different policies and different time scale scenarios is determined by successive iterations of the simulation. The agent first accepts the energy demand profile, PV power, power purchase price, and gas purchase price from the scheduling start phase. Then, it calculates the immediate reward value (Eq. (22)) obtained after selecting the action for each scheduling period based on the two action exploration policies and then adaptively adjusts the battery action selection until the maximum reward (minimum operating cost) is obtained for subsequent scheduling moments.

The convergence process of the cumulative reward during the training of the six strategies is shown in Fig. 12. The curves are drawn from the cumulative rewards after every 500 rounds of training, and the

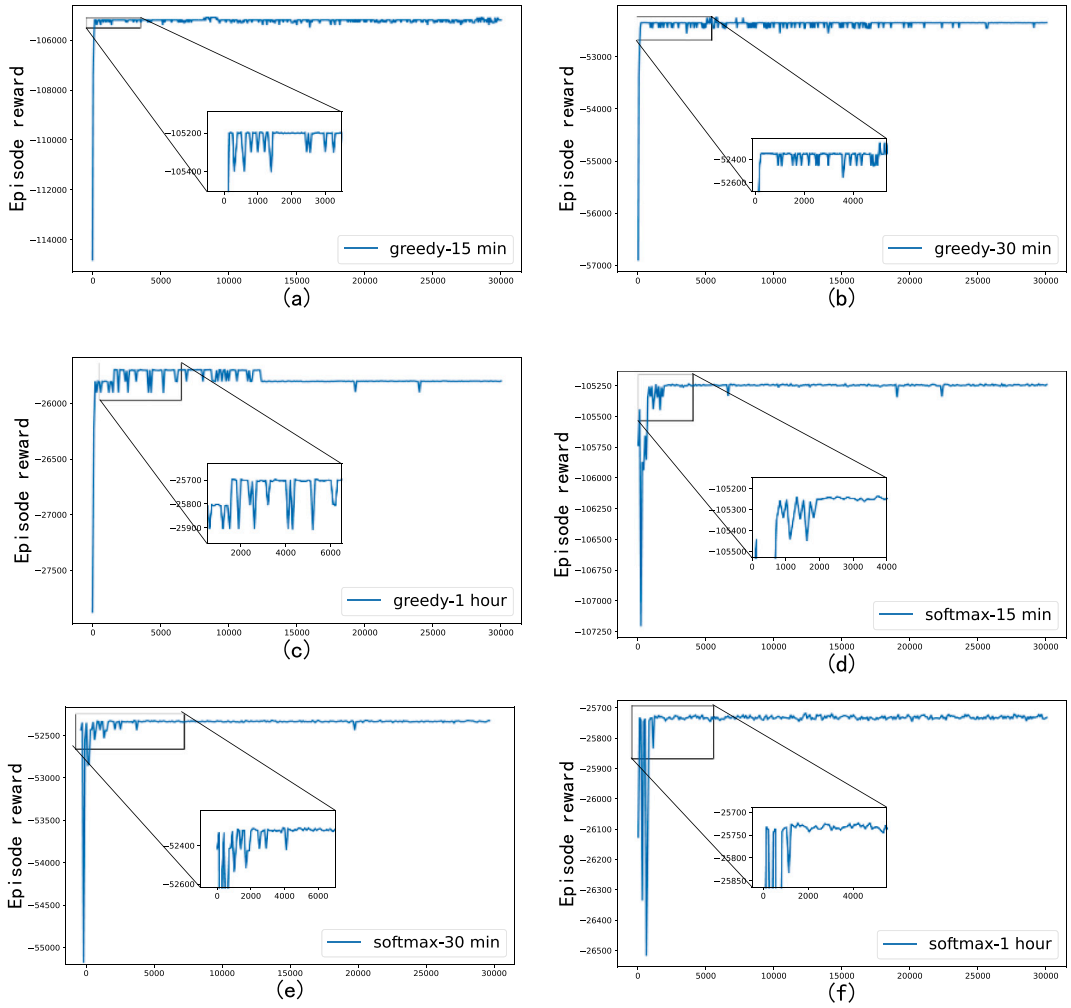

Fig. 12. Trends in reward functions during training of 6 strategies.

six graphs correspond to the above six strategies. 30,000 iterations complete the whole training process, and it can be seen from Fig. 13 that the training process of the Q-learning algorithm can be divided into two stages: exploration and convergence. Taking the first graph as an example, the 0th to 500th is the exploration phase. The agent selects and tries actions according to the established exploration policy, and the cumulative reward curve fluctuates widely in this phase. The cumulative reporting gradually converges after the 500th iteration. At this stage, the agent has successfully learned the optimal battery scheduling strategy in the stochastic environment to maintain the cumulative reward. The occasional fluctuation in the convergence part of the curve is due to the agent exploring with a small probability. However, it does not affect the whole learning process.

After 30,000 iterations, the battery learns the charging and discharging action selection at each scheduling moment guided by six strategies. Fig. 13 shows the results of each device output, battery SOC, and battery charging and discharging action selection for the IEM under the guidance of six battery scheduling strategies at different tariff times.

Fig. 13 shows in detail the changes in the electric power output of CHP units, the power output of PV generation, the power purchased by the microgrid from the larger grid, and the SOC of batteries under 6 different storage charging and discharging strategies for each period of the time-of-use tariff for IEM while maintaining power balance. In terms of the overall trend, 00:00 to 07:00 is the electricity price off-peak. At the same time, all 6 strategies choose to discharge action to reduce the purchasing power during this period to cost down the electricity bill. On the other hand, due to the low electricity price during this period, the battery chooses to recharge after discharging to replenish the power and reserve power for the subsequent high electricity price period. 07:00 to 10:00 tariff is in mid-peak time, and the PV output increases during this time to gradually meet the users' demand. The IEM purchases less electricity from the grid, and the battery SOC remains the same during this time. 10:00 to 14:00 tariff is on-peak hours, and the load demand is enormous during this time, the sunshine is sufficient, and the PV output is more during this time. It can be seen that only through the PV power generation can it meet the customer's electricity demand. In addition, the SOC curve in Fig. 13 shows that the battery can effectively store the excess power of PV generation during the time of sufficient sunlight, which provides a good guarantee for the electricity consumption of customers in the subsequent time. 14:00 to 16:00 is the flat segment tariff, there is sufficient sunlight during this time, PV still has sound power output, which protects users' electricity demand, users only need to purchase a small amount of power from the grid, batteries are mainly idle, and SOC fluctuations are negligible. The tariff from 17:00 to 19:00 is the on-peak tariff. During this time, users mainly purchase electricity from the grid to meet the energy demand. The tariff is higher during this time, the battery is more inclined to idle and discharge, and the SOC fluctuates less. After 19:00, the tariff is mid-peak, there is no sunlight during this period, and the PV output is 0. The figure shows that the user's load demand is low during this period, and the user mainly purchases electricity from the grid. On the other hand, the battery prefers to choose the discharge action to meet the user's electricity demand by discharging the power reserved during the daytime when there is sufficient sunlight to reduce the user's electricity purchase cost.

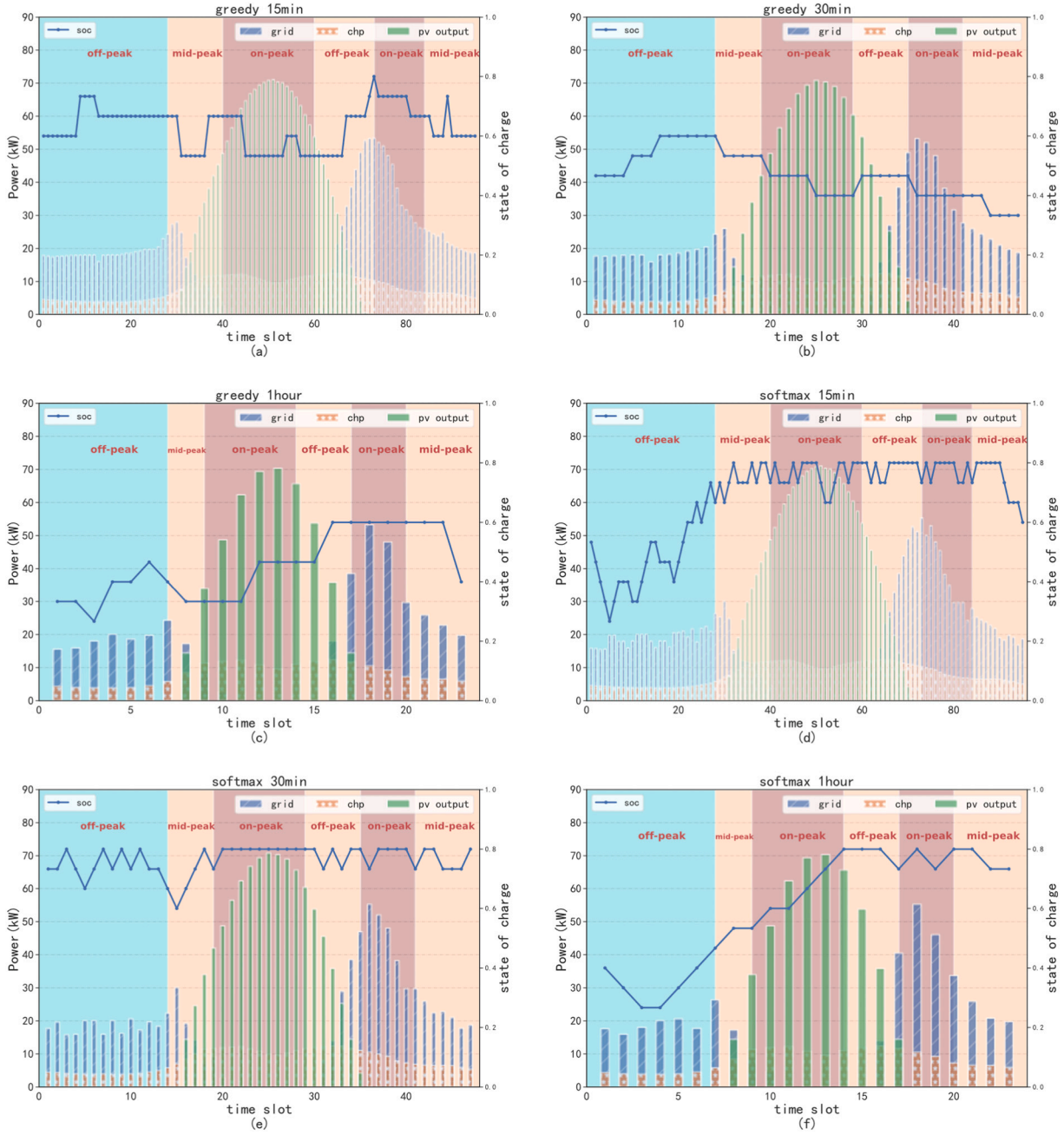

Fig. 13. Results of microgrid operation under 6 scheduling strategies.

The operating cost and solution time under the six battery scheduling strategies obtained by solving with the Q-learning algorithm are given in Table 5. In addition, for comparison with the conventional MILP solution results, the MILP calculation results and solution times obtained using the Matlab commercial solver are also provided in Table 5.

Table 5
Total cost under different strategies.

Strategies	Action exploration policy	Time step	Total cost (¥)	Time spent solving
Strategy 1	*ε-greedy*	15 min	459.914	6.35 s
Strategy 2	*ε-greedy*	30 min	458.431	2.51 s
Strategy 3	*ε-greedy*	1 h	455.702	1.06 s
Strategy 4	*Softmax*	15 min	463.150	7.51 s
Strategy 5	*Softmax*	30 min	462.171	2.97 s
Strategy 6	*Softmax*	1 h	456.458	2.01 s
MILP		1 h	441.858	2.73 s

As shown in Table 5, the results of the six reinforcement learning solutions have the best results for the *ε-greedy* action exploration policy with 1 h time scale, the lowest cost of ¥455.702, and the shortest solution time spent of 1.06 s. Moreover, the theoretical optimal solution MILP is ¥441.858 and takes 2.73 s to solve. In comparison, the reinforcement learning solution results are 3.13% less accurate but 61.17% more computationally efficient, which is closer to and more in line with the practical use of real-time response scheduling of energy storage.

3.4.3. *Effect of action exploration policy on battery scheduling results*

From the different action exploration policies of the Q-learning algorithm, the scheduling strategies of the three time scales using the *ε-greedy* exploration policy outperform the scheduling results of the corresponding time scales using the *Softmax* action exploration policy, both in terms of the accuracy of the solution and the computational

efficiency. It can be seen from Fig. 13 that the battery scheduled with the *ε-greedy* exploration policy has fewer charging and discharging actions than the one with the *Softmax* action exploration policy during the entire time scheduling process. It shows that when the battery fully understands the reported values obtained by each action through learning, it is more beneficial to choose the charge/discharge action with an enormous immediate reward to minimize the energy cost without exploring the remaining non-greedy actions. Therefore, compared with the *Softmax* exploration policy, the *ε-greedy* exploration policy can achieve the training goal more quickly and directly in applications with lower dimensions of state space and action space when the agent has acquired the corresponding Q values of each action through learning. Conversely, the *Softmax* exploration policy may perform better when facing higher state space and action space dimensions.

3.4.4. Effect of scheduling interval on the results of battery scheduling

In terms of different scheduling time scales, the longer the battery scheduling interval, the lower the overall operating cost of the IEM and the higher the computational efficiency, given the same action exploration policy. It shows that when the PV power generation can meet the energy demand of users, the reduction of battery charging and discharging times can effectively reduce the energy cost. On the other hand, the shorter the selected scheduling interval, the battery takes more charge/discharge times in the whole scheduling cycle. From the experimental results, the number of charges and discharges for the three scheduling time scales with *ε-greedy* action exploration policy are 14, 8, and 7 times. The number of charges and discharges for the three scheduling time scales with *Softmax* action exploration policy is 52, 26, and 15 times, respectively. In terms of current energy storage technology applications, batteries' unit charge and discharge costs are still high, so the increase in the number of battery charges and discharges will directly lead to the rise in the cost of energy use.

4. Conclusion

In this study, the battery charging and discharging strategies of IEM are explored in depth. Firstly, a modified deep learning algorithm is used to forecast the PV power and the load demand to accurately grasp the impact of source-load bilateral uncertainty and volatility on the microgrid. The reinforcement learning algorithm is then used to solve the optimal scheduling strategy of batteries from two perspectives of different action exploration policies and different time scales for the battery. The experimental results show that the energy storage scheduling strategy with *ε-greedy* policy and on a 1 h time scale can effectively reduce the operating cost of the IEM. Although the cost increases by 3.13% compared with the optimal solution, the solution efficiency is substantially improved by 61.17%. This result is more in line with the demand for real-time energy storage scheduling in practical application scenarios. The experimental results show that the charging and discharging cost of the battery can influence the agent's action selection in the scheduling process to a large extent, which in turn affects the total cost of microgrid operation. Therefore, with the development of energy storage technology and the reduction of battery charging and discharging costs, energy storage will achieve a more significant and prominent role in the operation of the microgrid in the future.

In addition, the Q-learning algorithm used in this study can significantly reduce the operation time and speed up the convergence of the reward function when dealing with discrete state spaces and action spaces. In future work, we will focus more on coping with microgrid environments that contain more devices coupled to operate and related problems containing higher-dimensional state spaces and action spaces.

CRediT authorship contribution statement

Kunshu Zhou: Methodology, Data curation, Software, Visualization, Validation, Writing – original draft. **Kaile Zhou:** Conceptualization, Methodology, Writing – review & editing, Resources, Supervision. **Shanlin Yang:** Supervision.

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.

Acknowledgements

This work is supported by the Natural Science Foundation of Anhui Province (No. 2008085UD05).

References

[1] B. Hong, W. Zhang, Y. Zhou, J. Chen, Y. Xiang, Y. Mu, Energy-internet-oriented microgrid energy management system architecture and its application in China, Appl. Energy 228 (2018) 2153–2164.
[2] E. Bullich-Massagué, F. Cifuentes-García, I. Glenny-Crende, M. Cheah-Maé, O. Gomis-Bellmunt, A review of energy storage technologies for large scale photovoltaic power plants, Appl. Energy 274 (2020), 115213.
[3] A. Aa, B. Ke, B. Sz, B. Ez, Dynamic energy management for photovoltaic power system including hybrid energy storage in smart grid applications, Energy 162 (2018) 72–82.
[4] S. Abedi, W.Y. Sang, 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.
[5] S.B. Wali, M. Hannan, M. Reza, P.J. Ker, R. Begum, M. Abd Rahman, M. Mansor, Battery storage systems integrated renewable energy sources: a biblio metric analysis towards future directions, J. Energy Storage 35 (2021), 102296.
[6] A. Mah, A. Sbw, B. Pjk, A. Msar, M. A, A. Vkr, C. Kmm, D. Tmim, E. Zyd, Battery energy-storage system: a review of technologies, optimization objectives, constraints, approaches, and outstanding issues, J. Energy Storage 42 (2021) 103023.
[7] M.H. Lipu, M. Hannan, T.F. Karim, A. Hussain, M.H. Saad, A. Ayob, M.S. Miah, T. Mahlia, Intelligent algorithms and control strategies for battery management system in electric vehicles: progress, challenges and future outlook, J. Clean. Prod. 292 (2021), 126044.
[8] M. Hannan, D. How, M.H. Lipu, P.J. Ker, Z. Dong, M. Mansur, F. Blaabjerg, Soc estimation of li-ion batteries with learning rate-optimized deep fully convolutional network, IEEE Trans. Power Electron. 36 (7) (2020) 7349–7353.
[9] Y. Xiao, W. Sun, L. Sun, Dynamic programming based economic day-ahead scheduling of integrated tri-generation energy system with hybrid energy storage, J. Energy Storage 44 (2021), 103395.
[10] Zhang Nan, D. Benjamin, Leibowicz, A. Grani, Hanasusanto, Optimal residential battery storage operations using robust data-driven dynamic programming, IEEE Trans. Smart Grid 11 (2) (2019) 1771–1780.
[11] A. Bouakkaz, A.J.G. Mena, S. Haddad, M.L. Ferrari, Efficient energy scheduling considering cost reduction and energy saving in hybrid energy system with energy storage, J. Energy Storage 33 (2021), 101887.
[12] J.N. Shan, R.X. Lu, in: Multi-objective Economic Optimization Scheduling of CCHP Micro-grid Based on Improved Bee Colony Algorithm Considering the Selection of Hybrid Energy Storage System vol.7, 2021, pp. 326–341.
[13] M. Roslan, M. Hannan, P.J. Ker, R. Begum, T.I. Mahlia, Z. Dong, Scheduling controller for microgrids energy management system using optimization algorithm in achieving cost saving and emission reduction, Appl. Energy 292 (2021), 116883.
[14] M.N. Akhter, S. Mekhilef, H. Mokhlis, R. Ali, M. Usama, M.A. Muhammad, A.S. M. Khairuddin, A hybrid deep learning method for an hour ahead power output forecasting of three different photovoltaic systems, Appl. Energy 207 (2021), 118185.
[15] A. Du Plessis, J. Strauss, A. Rix, Short-term solar power forecasting: investigating the ability of deep learning models to capture low-level utility-scale photovoltaic system behaviour, Appl. Energy 285 (2021), 116395.
[16] P. Kumari, D. Toshniwal, Long short term memory–convolutional neural network based deep hybrid approach for solar irradiance forecasting, Appl. Energy 295 (2021), 117061.
[17] G. Han, S. Lee, J. Lee, K. Lee, J. Bae, Deep-learning-and reinforcement-learning-based profitable strategy of a grid-level energy storage system for the smart grid, J. Energy Storage 41 (2021), 102868.
[18] B. Xu, Q. Zhou, J. Shi, S. Li, Hierarchical Q-learning network for online simultaneous optimization of energy efficiency and battery life of the battery/ultracapacitor electric vehicle, J. Energy Storage 46 (2022), 103925.
[19] J. Cao, D. Harrold, Z. Fan, T. Morstyn, D. Healey, K. Li, Deep reinforcement learning-based energy storage arbitrage with accurate lithium-ion battery degradation model, IEEE Trans. Smart Grid 11 (5) (2020) 4513–4521.
[20] M. Zhang, Q. Wu, J. Wen, Z. Lin, Q. Chen, Optimal operation of integrated electricity and heat system: a review of modeling and solution methods, Renew. Sust. Energ. Rev. 135 (2021), 110098.

[21] C. Cai, X. Dou, S. Cao, X. Chen, N. Wang, Energy optimization based on model predictive control for combined heating and power microgrid in industrial park, Electr. Power Constr. 40 (2019) 27–33.
[22] S. Ferahtia, H. Rezk, M.A. Abdelkareem, A. Olabi, Optimal techno-economic energy management strategy for building's microgrids based bald eagle search optimization algorithm, Appl. Energy 306 (2022), 118069.
[23] B. Hca, G. Lin, Z.C. Zhong, Multi-objective optimal scheduling of a microgrid with uncertainties of renewable power generation considering user satisfaction, Int. J. Electr. Power Energy Syst. 131 (2021), 107142.
[24] A.R. Jordehi, Economic dispatch in grid-connected and heat network-connected CHP microgrids with storage systems and responsive loads considering reliability and uncertainties, Sustain. Cities Soc. 73 (2021), 103101.
[25] M. Shafiee, M. Rashidinejad, A. Abdollahi, A. Ghaedi, A novel stochastic framework based on PEM-DPSO for optimal operation of microgrids with demand response, Sustain. Cities Soc. 72 (April) (2021), 103024.
[26] N. Cui, J. Mu, Application of New Energy Microgrid System in Industrial Park: Proceedings of PURPLE MOUNTAIN FORUM 2019-International Forum on Smart Grid Protection and Control, 2020.
[27] X. Sui, S. He, S.B. Vilsen, J. Meng, D.I. Stroe, A review of non-probabilistic machine learning-based state of health estimation techniques for Lithium-ion battery, Appl. Energy 300 (3) (2021), 117346.
[28] Y. Shang, W. Wu, J. Guo, Z. Ma, W. Sheng, Z. Lv, C. Fu, Stochastic dispatch of energy storage in microgrids: an augmented reinforcement learning approach, Appl. Energy 261 (2020), 114423.
[29] Z. Wang, Y. Liu, Z. Ma, X. Liu, J. Ma, LiPSG: lightweight privacy-preserving Q-learning-based energy management for the IoT-enabled smart grid, IEEE Internet Things J. 7 (5) (2020) 3935–3947.
[30] T. Nakabi, P. Toivanen, Deep reinforcement learning for energy management in a microgrid with flexible demand, Sustain. Energy Grids Netw. 25 (3) (2021), 100413.
[31] R. Sutton, A. Barto, 2nd ed., MIT Press, Cambridge, MA, USA, 2018.
[32] B. Gu, H. Shen, X. Lei, H. Hu, X. Liu, Forecasting and uncertainty analysis of day-ahead photovoltaic power using a novel forecasting method, Appl. Energy 299 (MAR) (2021), 117291.
[33] H. Yang, K.R. Schell, Real-time electricity price forecasting of wind farms with deep neural network transfer learning and hybrid datasets, Appl. Energy 299 (2021), 117242.
[34] Y. Liu, Probabilistic spatiotemporal wind speed forecasting based on a variational Bayesian deep learning model, Appl. Energy 260 (15) (2020), 114259.
[35] L. Li, C.J. Meinrenken, V. Modi, P.J. Culligan, Short-term apartment-level load forecasting using a modified neural network with selected auto-regressive features, Appl. Energy 287 (147) (2021), 116509.
[36] G. Pinto, D. Deltetto, A. Capozzoli, Data-driven district energy management with surrogate models and deep reinforcement learning, Appl. Energy 304 (2021), 117642.
[37] Y. Wang, P. He, X. Tan, Greedy Multi-step Off-Policy Reinforcement Learning, 2021 arXiv:2102.11717v4.
[38] I. Zoltowska, P. Cichosz, W. Kolodziejczyk, Real-time energy purchase optimization for a storage-integrated photovoltaic system by deep reinforcement learning, Control. Eng. Pract. 106 (2021), 104598.
[39] S. Jayaraj, I.A. TP, Application of reinforcement learning algorithm for scheduling of microgrid, in: [C]//2019 Global Conference for Advancement in Technology (GCAT), IEEE, 2019, pp. 1–5.
[40] D. Lee, Entropy-Augmented Entropy-Regularized Reinforcement Learning and a Continuous Path from Policy Gradient to Q-Learning, 2020 arXiv preprint arXiv: 2005.08844.
[41] Alexandre dos Santos Mignon, Ricardo Luis de Azevedo da Rocha, An adaptive implementation of ε-greedy in reinforcement learning, Procedia Comput. Sci. 109 (Jan. 2017) 1146–1151.
[42] K. Asadi, M.L. Littman, An alternative softmax operator for reinforcement learning, in: Proc. 34th Int. Conf. Mach. Learn vol. 70, JMLR.org, 2017, pp. 243–252.
[43] V.K. Kurmi, V.K. Subramanian, V.P. Namboodiri, Exploring dropout discriminator for domain adaptation, Neurocomputing 457 (2021) 168–181.
[44] S. Patro, K.K. Sahu, Normalization: A Preprocessing Stage, IARJSET, 2015 arXiv preprint arXiv:1503.06462.
[45] P. Ramachandran, B. Zoph, Q.V. Le, Searching for Activation Functions, 2017 arXiv preprint arXiv:1710.05941.
[46] H. Zhao, F. Liu, H. Zhang, Z. Liang, Research on a learning rate with energy index in deep learning, Neural Netw. 110 (2019) 225–231.