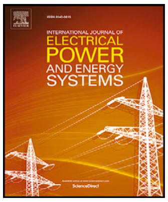
Electrical Power and Energy Systems 161 (2024) 110148

Contents lists available at ScienceDirect

International Journal of Electrical Power and Energy Systems

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



Real-time power system dispatch scheme using grid expert strategy-based imitation learning

Siyang Xu [a], Jiebei Zhu [a,*], Bingsen Li [a], Lujie Yu [a], Xueke Zhu [a], Hongjie Jia [a], Chi Yung Chung [b], Campbell D. Booth [c], Vladimir Terzija [d]

[a] *School of Electrical and Information Engineering, Tianjin University, Tian Jin, China*
[b] *Department of Electrical and Electronic Engineering, Hong Kong Polytechnic University, Hong Kong, China*
[c] *Electronic and Electrical Engineering, University of Strathclyde, Lanarkshire, United Kingdom*
[d] *School of Electrical and Electronic Engineering, The University of Manchester, Manchester, United Kingdom*

ARTICLE INFO

Keywords:
Real-time dispatch
Imitation learning
Grid export strategy
***N*-1** security operation
Reinforcement learning

ABSTRACT

With large-scale grid integration of renewable energy sources (RES), power grid operations gradually exhibit the new characteristics of high-order uncertainty, leading to significant challenges for system operational security. Traditional model-driven generation dispatch methods require large computational resources, whereas the widely concerned Reinforcement Learning (RL)-based methods lead to issues such as slow training speed due to the high complexity and dimension of processed grid state information. For this reason, this paper proposes a novel Grid Expert Strategy Imitation Learning (GESIL)-based real-time (5 min intervals in this paper) dispatch method. Firstly, a grid model is established based on the graph theory. Secondly, a pure rule-based grid expert strategy (GES) considering detailed power grid operations is proposed. Then, the GES is combined with the established model to obtain a GESIL agent using imitation learning by offline–online training, which can produce specific grid dispatch decisions for real-time. By designing a graph theory-based grid model, a model-driven purely rule-based GES, and embedding a penalty factor-based loss function into IL offline–online training, GESIL ultimately achieves high training speed, high solution speed, and strong generalization capability. A modified IEEE 118-node system is employed to compare the proposed GESIL to traditional dispatch method and RL method. Results show that GESIL has significantly improved computational efficiency by approximately 17 times and training speed by 14.5 times. GESIL can more stably and efficiently compute real-time dispatch decisions of grid operations, enhancing the optimization effect in terms of transmission overloading mitigation, transmission loading optimization, and power balancing control.

1. Introduction

With the rapid development of renewable energy sources (RES), low-carbon, intelligent, and friendly modern power systems are gradually taking shape [1]. However, the intermittency, volatility, and uncertainty of high-penetration RES lead to the “energy trilemma” among system security, economy, and sustainability. How to break through such a trilemma becomes a foremost task for system operators [2]. Power system dispatch strategies can be categorized by time scale into day-ahead dispatching, intra-day dispatching, and real-time dispatching. Among these, real-time dispatching has the highest requirements for computational timeliness, as it further corrects the results of day-ahead and intra-day dispatching based on precise ultra-short-term forecasts of renewable energy and load data. However, real-time dispatching has limited flexibility in mobilizing resources within a short timeframe, making it challenging to accurately achieve power balance and meet N-1 contingency requirements in scenarios with a high proportion of RES. Therefore, it is crucial to design new, safe, and efficient real-time dispatch methods for modern power systems.

Historically reported dispatch schemes are mainly based on model-driven methods, which can be classified as grid expert-based empirical strategies and mathematical optimization-based strategies. The grid expert-based dispatch schemes typically carry out offline contingency analysis to identify system operational risks, and then develop dispatch plans based on dispatchers’ empirical experiences [3]. However, these approaches have difficulties in accurately tackling with real-time operation issues promptly. On the other hand, the dispatch schemes based on mathematical optimization algorithms mainly involve robust optimization [4], stochastic programming [5], and chance-constrained

[*] Corresponding author.
E-mail address: jiebei.zhu@tju.edu.cn (J. Zhu).

https://doi.org/10.1016/j.ijepes.2024.110148
Received 15 April 2024; Received in revised form 20 June 2024; Accepted 17 July 2024
Available online 29 July 2024

Nomenclature

Abbreviations

DRO	Distributionally robust optimization
GES	Grid expert strategy
GESIL	Grid expert strategy imitation learning
IL	Imitation learning
MADRL	Multi-agent deep reinforcement learning
RES	Renewable energy sources
RL	Reinforcement learning
RUR	Renewable utilization rates
SAC	Soft actor–critic
SGs	Synchronous generators
TGES	Traditional grid expert strategy

Variables

$\mathit{\Delta} P_{(g,i),t}$	The amount of adjusted active power of generator *i*
$\mathit{\Delta}\boldsymbol{P}^{\max}_{G,t+1}, \mathit{\Delta}\boldsymbol{P}^{\min}_{G,t+1}$	The sets of upper and lower active power limits
$\mathit{\Delta} P^{\max}_{\mathrm{r},t+1}$	The adjusted upper limit summation of all RES generators' active power
$\mathit{\Delta} P^{max}_{\mathrm{T},t+1}$	The summation of the upper limit of active power adjustment of all the SGs
$\mathit{\Delta} P^{min}_{\mathrm{T},t+1}$	The summation of the lower limit of active power adjustment of all the SGs
$\mathit{\Delta} P^{\min}_{\mathrm{r},t+1}$	The summation of the lower limit of active power adjustment of all the RES
$\mathit{\Delta} P^{\mathrm{neg}}_{y}$	The adjusted active power of downstream generator *y*
$\mathit{\Delta} P^{\mathrm{pos}}_{x}$	The adjusted active power of upstream generator *x*
$\mathit{\Delta} P^{\mathrm{S}}_{g,i}$	The ramp power of standby generator *i*
$\mathit{\Delta} P^{t}_{\mathrm{load}}$	The load at current time step *t*
$\mathit{\Delta} P^{t+1}_{\mathrm{load}}$	The ultra-short-term load forecast value
C_i	The shutdown operation
$C^{\max}$	The maximum value of generator operating cost for the current dispatch time step
$\boldsymbol{D}_{g,i}$	The number of branches connected to the node where generator *i is located*
$\boldsymbol{G}$	The set of generators
$\boldsymbol{G}^{C}$	The set of offline generators
$\boldsymbol{G}^{O}$	The set of online generators
$\boldsymbol{L}$	The set of branch circuits
P_d	The RES full generation criterion
$\boldsymbol{P}_{\mathrm{G}}$	The set of generator active power outputs
$P_l / P^{\max}_l$	The active current/thermal capacity of branch *l*
$P^{\max}_{r,t}$	The upper limit of RES active power
$\boldsymbol{P}^{\max}_{r,t+1}$	The set of forecasted RES power outputs
$P^{\min/\max}_{g,i}$	The lower/upper limits of generator power
P^{npp}_{b}	The neutral power point
$P_{r,t}$	The summation of RES active power
$\boldsymbol{R}$	The set of branch loading ratios
$r_{(g,i)}$	Ramp rate
ratio	The proportional adjustment coefficient of the generator's active power
R_{C}	The normalized result of r_3
reward	The reward for real-time dispatch decision
$R_l / R^{\max}_l$	The current/maximum loading ratio of branch *l*
S_i	The startup operation
$\boldsymbol{T}_{\mathrm{C}}$	The set of remaining restart time for offline generators
$T^{\mathrm{C}}_{g,i}$	The shutdown time
$\boldsymbol{U}^{\mathrm{G}}$	The set of generator startup/shutdown states
U^{G}_{i}	The switching action of generator *i*
$\boldsymbol{U}^{\mathrm{L}}$	The set of branch on/off states
a_1, a_2, a_3	The reward weighting coefficients
a_i, b_i, c_i	The cost coefficients of generator *i*
r_1	The grid security operation evaluation coefficient
r_2	The RES utilization evaluation coefficients
r_3	The generator operation cost evaluation coefficients
θ	The penalty value of reward

planning [6]. The optimization-based dispatch method focuses on finding the best solution based on mathematical algorithms, ensuring efficiency and accuracy. A robust optimization method is proposed in [4] to establish an integrated electricity and natural gas system model with a special consideration of RES outputs uncertainty. However, such a method is overly conservative in dispatch decision-making due to the strict grid-operation security constraints in the solving process, and it cannot fully utilize the potential of the system for RES accommodation. In [5], a stochastic multi-objective economic dispatch model is proposed to minimize the total operating costs of generators and spinning reserves. However, the probabilistic modeling scale of this method is extremely large due to the uncertainty of RES outputs. In [6], a chance-constrained dispatch method is proposed, with a Gaussian mixture model to capture RES power uncertainty. But this method requires establishing the probability distribution for wind power output, which could potentially result in errors in the generated dispatch plans. A newly proposed distributionally robust optimization (DRO) method, which accounts for the true distribution of uncertainties lies in an ambiguity set, can significantly improve the accuracy of dispatch plans [7]. The general DRO method usually adopts the probability distribution ambiguity set based on distance [8] and moment information [9], but the resulting NP-hard problem is hard to solve [10]. In addition, there is literature that employs model predictive control (MPC) to achieve microgrid energy management integrated with battery energy storage [11]. However, MPC faces challenges such as computational real-time.

Nowadays, real-time dispatch is gaining growing importance as it serves as the ultimate mechanism for rectifying dispatching outcomes prior to the power grid entering the closed-loop control stage. However, the model-driven methods are generally unable to process the high volume of system state data on such a real-time scale [12]. In contrast, the data-driven solution has excellent performance in computing efficiency, accuracy, and decision-making rationality, without the requirement for accurate modeling of RES outputs uncertainty [13,14]. Meanwhile, the big data collected from a power system can be leveraged to produce data-driven dispatch solutions, thereby strengthening the timeliness of decision-making. As the grid operation environments may be modeled as a Markov chain, it is possible to transform the grid dispatch strategy into a sequential decision problem to solve [15]. Reinforcement learning (RL) is among the suitable methods for solving sequential decision problems. Literature [16] introduces a dual-depth Q-learning-based system dispatch method that uses a deep neural network as a function approximator, thereby improving the efficiency of solving

Table 1
Comparison of GESIL with model-driven methods and RL.

Features	Proposed GESIL	Model-driven method	RL
Training time	Short training time based on expert system demonstration	No training required	Long training time due to random exploration of action space to obtain rewards
Solution speed	Fast	Slow	Fast
Computational complexity	Low	High	Low
Generalization capability	Strong generalization ability based on efficient expert system and offline + online training	Good generalization ability based on optimized dispatching model.	Weak generalization ability in facing multi-scenario dispatching

large grid state space. However, this method has to solve large volumes of discrete data in both the state and action space, potentially leading to the curse of dimensionality. To overcome this issue, deep reinforcement learning (DRL), which combines RL with deep learning technologies, has been studied and adopted in power system dispatch in the past few years [17–19]. Literature [20] utilizes a depth-deterministic policy gradient algorithm to convert discrete variables into continuous variables, effectively reducing the issues caused by dimensional catastrophe in the solving process, but the agent training may still be inefficient. A soft actor–critic (SAC)-based policy gradient algorithm is utilized in [21] to reduce the dimensionality of the dispatch model, but the method does not consider the typical grid N-1 operational requirement in the reward function design. To improve the learning efficiency, more advanced multi-agent deep reinforcement learning (MADRL) is applied in autonomous voltage control [22], distribution networks [23], and microgrids [24,25], et al. However, the low sample efficiency and high operational data collection cost make it difficult to train MADRL algorithms in practice for large-scale power systems [26]. In general, the aforementioned RL-based system dispatch methods proposed in [16–25] all use a random search to explore the action space and obtain rewards. However, when applied to large power systems, they can suffer from time-consuming training processes and difficulty in converging to optimal dispatch solutions, due to the large dimensionality of the explored action space [27]. Compared to RL, imitation learning (IL) can achieve faster and more accurate sequential decision results with less data. Furthermore, the sample complexity exponentially decreases [28–30]. An IL algorithm is applied to a cloud resource dispatch scheme in [31], and the training efficiency of the IL-optimized agent is significantly improved compared to the RL algorithms. The above literature [28–31] demonstrated that IL can be solved for better dispatching results while taking, into account the advantages of RL, and therefore, it is necessary to apply IL to the field of real-time power system dispatching. However, the successful application of IL hinges on a responsive and well-established expert system to guide its training process. When IL is deployed in the real-time dispatching of large-scale power systems, existing model-driven methods encounter challenges in fulfilling the role of expert systems for imitation learning, primarily due to their iterative optimization nature.

To enhance the solution speed and dispatch accuracy of real-time power system dispatch methods, thereby ensuring the secure and stable operation of the power system, this paper proposes a real-time power system dispatch scheme using Grid Expert Strategy-based Imitation Learning (GESIL). An efficient rule-based grid expert strategy (GES) is designed, diverging from traditional model-driven methods characterized by iterative optimization. By embedding rule design and a graph theory grid model based purely on mathematical formulations, GES can efficiently guide IL training without the need for iterative solutions. Moreover, for generator switching, a loss function for IL incorporating penalty factors is designed, and the fusion of GES and IL is realized through an offline–online training approach. Table 1 is a taxonomy table to compare the features of GESIL with model-driven methods and RL. The main contributions of this paper are as follows.

1. A graph theory-based grid model is proposed, which establishes the relationships among connection nodes of transmission branches, generators, and loads based on pure mathematical formulations. With such a model, the prompt identification of weak grid risks is attainable, bypassing the necessity for traditional fault analysis relying on power flow iterations and contingency scanning. This advancement significantly enhances the efficiency of model-driven solving processes.
2. Based on the established graph theory-based grid model, a rule-based Grid Expert Strategy (GES) is proposed to effectively guide the IL training. With the proposed GES, active power outputs of regulated generators are adjusted to reduce overloading risks, optimize branch loading ratio, and facilitate power balancing control. Importantly, GES achieves these objectives without the need for iterative solving, as is typically required in traditional model-driven dispatching methods.
3. Embedding the penalty factor-based loss function into offline–online training, the GES and IL are integrated to create a model-data-driven GESIL real-time dispatch scheme. This scheme enhances the computational efficiency and generalization capability of the real-time GESIL-based dispatch, as well as the dispatch accuracy.

The rest of the paper is organized as follows. Section 2 introduces the fundamentals of grid operation strategy and GESIL-related technologies. Section 3 presents the design of the GES scheme, including the proposed graph theory-based grid modeling and dispatch expert strategy. Section 4 presents the implementation framework of GESIL, including the specific IL design and the GESIL realization in real-time power system dispatch. Section 5 performs numerical validations and comparisons in a modified IEEE 118-node system. Section 6 concludes the paper.

2. Basic grid operation rules and GESIL related fundamentals

This section presents basic grid operation rules and GESIL related fundamentals. As the basis for the design of GES and IL action spaces, Sections 2.1 and 2.2 introduce power system operation constraints and active power regulation actions, respectively. Sections 2.3 and 2.4 respectively delve into the following topics: graph theory as the basis for grid modeling in Section 3.1, IL and its training methods which are crucial for the GES and IL fusion process in Section 4. Additionally, Sections 2.5 and 2.6 introduce the proposed comparative SAC algorithm, evaluation metrics, and reward functions, respectively.

2.1. Power system operating constraint

The operational constraints on the active power of generators and the thermal capacities of transmission branches can be expressed as [32]:

$$\sum_{(g,i)\in \boldsymbol{G}} \Delta P_{(g,i),t} = \Delta P_{\mathrm{load}}^{t+1} - \Delta P_{\mathrm{load}}^{t} \tag{1}$$

$$P_{g,i}^{\min} \le P_{g,i} \le P_{g,i}^{\max}, \quad \forall (g,i) \in \boldsymbol{G} \tag{2}$$

$$R_l = \frac{P_l}{P_l^{\max}} \le R_l^{\max}, \quad \forall l \in \boldsymbol{L} \tag{3}$$

where $t \in [1, T]$ is the time index.

2.2. Power system active regulation action

The main active power regulation actions of a power system include generator active power adjustment and generator switching, with the following rules [33]:

$$\Delta P_{(g,i),t} - r_{g,i} \leq \Delta P_{(g,i),t} \leq \Delta P_{(g,i),t} + r_{g,i} \quad \forall (g,i) \in \boldsymbol{G} \tag{4}$$

where $r_{g,i}$ is the ramp rate.

The generator switching rules [33] can be expressed as:

$$\begin{cases} C_i = 0 & \text{if } P_{g,i} > P_{g,i}^{\min} \\ C_i = 1 & \text{if } P_{g,i} \leq P_{g,i}^{\min} \end{cases} \quad \forall (g,i) \in \boldsymbol{G} \tag{5}$$

$$\begin{cases} S_i = 0 & \text{if } T_{g,i}^{c} \leq T_c \\ S_i = 1 & \text{if } T_{g,i}^{c} \geq T_c \end{cases} \quad \forall (g,i) \in \boldsymbol{G} \tag{6}$$

when the active power $P_{g,i}$ of the generator i is less than or equal to the lower output limit $P_{g,i}^{\min}$, C_i is 1 or 0 indicating that the shutdown operation is allowed or not allowed, respectively. When the shutdown time $T_{g,i}^{c}$ of the shutdown generator i is larger than or equal to the set value T_c, S_i is 1 or 0 indicating that the startup operation is allowed or not allowed, respectively.

2.3. Graph theory

Graph theory can be used to integrate a grid model by extracting the connection node relationship among generators, grid branches, and loads. The connection point of each unit in a power grid can be generally regarded as a node in the network [34]. Consider a weighted directed graph $G = (V, \varepsilon, A)$ composed by a finite node set $V = \{1, 2, \ldots, n\}$, a finite edge set $\varepsilon \subseteq V \times V$, and a weighted adjacency matrix $A = \left[a_{ij}\right]_{n \times n}$ with $a_{ij} > 0$ if $(j,i) \in \varepsilon$ and $a_{ij} = 0$ otherwise. The ordered pair $(j,i) \in \varepsilon$ represents the capability of node i to receive information from node j. Let two sets $N_i^{\text{in}} = \{j \in V \mid (i,j) \in \varepsilon\}$ and $N_i^{\text{out}} = \{j \in V \mid (i,j) \in \varepsilon\}$ denote the in- and out-degree neighbor set of nodes i, respectively. A directed path refers to a sequence of edges with the form $\left(v_1, v_2\right), \left(v_2, v_3\right), \ldots, \left(v_{n-1}, v_n\right)$. A directed strongly connected graph is one such that, for every node in the graph, there is at least a directed path to every other node in the graph.

2.4. IL and its training methods

IL is a learning mode characterized by imitating the behavior of experts, which refers to an agent's acquisition of knowledge by observing an expert demonstrating a given task, as depicted in Fig. 1 [30]. A key distinction between RL and IL is that RL learns mostly through trial-and-error while IL aims to learn through direct mappings from states to actions, particularly by utilizing an expert replay buffer via supervised learning [35]. IL possesses the advantage of its high learning efficiency in terms of sample complexity and training time [36].

The most widely used training methods for IL are behavioral cloning (BC) and dataset aggregation (DAgger) [36]. BC uses supervised learning to train the optimal policy trajectory by labeling the expert data to an agent. DAgger is a data aggregation method that addresses the issue of error accumulation by online interaction between an agent and its environment. It utilizes expert demonstrations to showcase new states encountered by an agent and updates a training set in real-time. This paper prepares to use the above training methods for offline–online learning of GES by IL in Section 4 (see Fig. 1).

2.5. SAC algorithm for proposed comparison

SAC algorithm is one of the most efficient off-policy reinforcement learning methods, which incorporates the maximized entropy into the objective search at the same time to prevent the agent from converging to the local optimum prematurely, making it widely used in the field of sequential decision-making [37]. In this paper, GESIL is compared to the grid dispatch scheme based on the SAC algorithm [21].

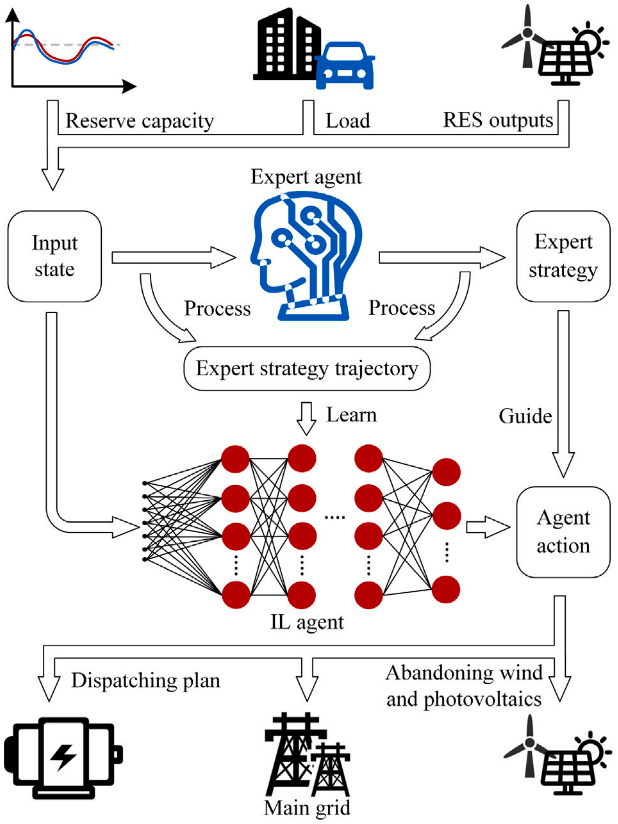


Fig. 1. Schematic diagram of imitation learning.

2.6. Evaluation metrics and reward functions

To compare the dispatch results with other schemes in Section 5, dispatch evaluation metrics are employed as in [15]:

$$r_1 = 1 - \frac{1}{L} \sum_{l \in L} \min\left(R_l, 1\right) \tag{7}$$

$$r_2 = \frac{P_{\mathrm{r},t}}{P_{\mathrm{r},t}^{\max}} \tag{8}$$

$$r_3 = -\sum_{i \in G} \left(a_i P_{g,i}^2 + b_i P_{g,i} + c_i\right) \tag{9}$$

To train the dispatch agent, reward functions are expressed as:

$$R_C = \frac{r_3}{C^{\max}} \tag{10}$$

$$reward = \begin{cases} \theta, \text{ if grid not convergence} \\ a_1 r_1 + a_2 r_2 + a_3 R_c, \text{ else} \end{cases} \tag{11}$$

Since safe integration of RES is the task priority in power systems, a_2 in this paper is set 2 3 times larger than the remaining weights, namely a_1 and a_3.

Based on (7)–(9), Section 5.1 will validate the algorithm performance of GESIL, including training convergence, generalization capability, and single-step decision dime. In Section 5.2, comparisons will be made concerning grid operation optimization, which includes averaging grid loading ratios, addressing grid overloading scenarios, and optimizing slack generation regulation margin.

3. New GES design solution

To reduce the risk of grid overloading under contingencies, the novel GES is proposed in this subsection, which consists of 5 modules, employing a graph theory-based grid model which will be described in Section 3.1 to adjust generator power output for overloading risk alleviation as well as for system power balance. The designed GES will be used as an expert strategy imitated by GESIL in Section 4.

3.1. GES graph theory-based grid modeling module

As introduced in Section 1, traditional grid contingency analysis, based on detailed power system models, scans all possible grid contingencies, and identifies grid operational risks. In real-time timescale for grid dispatch, such an analysis can be computationally intensive and difficult to immediately obtain analysis results. To address these challenges, this subsection proposes a graph theory-based grid modeling module, which establishes the relationships among connection nodes of transmission branches, generators, and loads.

The graph theory-based grid model primarily consists of three matrices: the association matrix of generator-node $\boldsymbol{V}$, the adjacency matrix $\boldsymbol{B}$, and the association matrix of node-branch connectivity $\boldsymbol{C}$. Matrix $\boldsymbol{V}$ distinguishes between generator nodes and non-generator nodes, matrix $\boldsymbol{B}$ provides the adjacency relationships of grid nodes, and matrix $\boldsymbol{C}$ represents the connectivity between nodes and branches. Based on the three matrices $\boldsymbol{V}$, $\boldsymbol{B}$, and $\boldsymbol{C}$, mathematical transformations yield (12) and (15). (12) can be used in *GES Contingency Analysis and Optimization Module*, and (15) can be applied in *GES Branch Loading Ratio Averaging Module* and *GES Generator Power Ramping and Switching Module*.

The connection information of the upstream/downstream generators of a random branch can be immediately obtained through:

$$\boldsymbol{I} = \boldsymbol{V} \cdot \boldsymbol{B}^{n} \cdot \boldsymbol{C} \tag{12}$$

where $\begin{matrix} \boldsymbol{I} = \begin{bmatrix} I_{11} & \dots & I_{1L} \\ \dots & \dots & \dots \\ I_{G1} & \dots & I_{GL} \end{bmatrix} \boldsymbol{V} = \begin{bmatrix} V_{11} & \dots & V_{1N} \\ \dots & \dots & \dots \\ V_{G1} & \dots & V_{GN} \end{bmatrix} \\ \boldsymbol{B} = \begin{bmatrix} B_{11} & \dots & B_{1N} \\ \dots & \dots & \dots \\ B_{N1} & \dots & B_{NN} \end{bmatrix} \boldsymbol{C} = \begin{bmatrix} C_{11} & \dots & C_{1L} \\ \dots & \dots & \dots \\ C_{N1} & \dots & C_{NL} \end{bmatrix} \end{matrix}$, $\boldsymbol{I}$ is the association matrix of generator-branch power flow relationships in the range of n paths (e.g., a positive/negative/zero I_{ij} indicates a positive/negative/no power flow correlation between generator i and branch j), $\boldsymbol{V}$ is the association matrix of a generator-node connectivity (e.g., $V_{ij} = 1$ or 0 means generator i is connected or unconnected with node j), $\boldsymbol{B}$ is an adjacency matrix weighting the proximity between two random grid nodes of all grid nodes (e.g., $B_{ij} = 1$ or 0 indicates node i is immediately adjacent with node j via only 1 branch or isolated by at least 2 branches, respectively), and $\boldsymbol{C}$ is an association matrix of node-branch connectivity (e.g., $C_{ij} = 1$, -1, or 0 means node i is the beginning, the end of the branch j or isolated with branch j, respectively).

Fig. 2 visualizes the IEEE 39-node system model established using graph theory, which is realized by python programming based on (12). This model acts as a model foundation to be applied with the proposed GES in the next subsection.

3.2. GES contingency analysis and optimization module

As introduced in Section 1, to ensure grid operational security, system operators typically perform N-1 contingency analysis to identify overloading risks and optimize grid loading beforehand. A contingency analysis and optimization module is designed to adjust the active power of generators with highest sensitivities on branch l prone to overloading, expressed as:

$$\begin{cases} \Delta P_{x}^{\text{pos}} = -\alpha P_{x}^{\max}, I_{x,l} > 0, x \in X \\ \Delta P_{y}^{\text{neg}} = \alpha P_{y}^{\max}, I_{y,l} < 0, y \in Y \end{cases} \tag{13}$$

$$\Delta P_{\text{imb1}} = -\left(\sum_{x \in X} \Delta P_{x}^{\text{pos}} + \sum_{y \in Y} \Delta P_{y}^{\text{neg}} \right) \tag{14}$$

where α is a power adjustment factor with a fixed increment (in this paper α is set as 0.05 p.u.), X and Y are the sets of the upstream and downstream generators within n paths to the overloading branch (in this paper n is set as 2), respectively, obtained from generator-branch association matrix $\boldsymbol{I}$ of (12), and ΔP_{imb1} is a total system power imbalance resulting from generator active power adjustments above.

3.3. GES branch loading ratio averaging module

To enhance transmission branch utilization and reduce transmission power losses in real time, a branch loading ratio averaging module is designed here to average branch loading as weighted by branch loading ratio. Firstly, the number of branches connected with generator nodes is obtained a connectivity degree matrix $\boldsymbol{D}$ as:

$$\boldsymbol{D} = \boldsymbol{V} \cdot \operatorname{diag}\left(\sum_{u \in N} B_{u1}, \sum_{u \in N} B_{u2}, \dots \sum_{u \in N} B_{uN} \right) \tag{15}$$

where $\boldsymbol{D}$ indicates the number of branches connected to a generator node, obtained by diagonalizing the summations of each row of $\boldsymbol{B}$ and multiplying $\boldsymbol{V}$.

The higher value in $\boldsymbol{D}$ a specific generator has, the more branches the generator is directly connected with. Therefore, the larger the generator power output shall be adjusted to change the power flow of a certain branch connected with the generator.

3.4. GES power balancing control module

To ensure system power balancing, it is essential to accurately account for power system imbalances, including the grid loading optimization in (14), the prevailing system imbalance in (16), and the slack generator margin optimization in (17). In response, a power balance control module is embedded in the GES.

Firstly, the initial power imbalance ΔP_{imb2} can be obtained by comparing the maximum forecasted load increase with the largest generation infeed loss $\Delta P_{sg}^{\max}$ at the next time step, as:

$$\Delta P_{\text{imb2}} = \max\left\{ \Delta P_{sg}^{\max}, \Delta P_{\text{load}}^{t+1} - \Delta P_{\text{load}}^{t} \right\} \tag{16}$$

Then, the prevailing regulating power margin of the slack generator is obtained as:

$$\Delta P_{\text{imb3}} = P_{b} - P_{b}^{\text{npp}} \tag{17}$$

where ΔP_{imb3} is the power deviation from the neutral power point of the slack generator, P_{b} is its actual power point, and P_{b}^{npp} is its neutral power point.

Finally, the system power imbalance ΔP_{imb} can be determined, expressed as:

$$\Delta P_{\text{imb}} = \Delta P_{\text{imb1}} + \Delta P_{\text{imb2}} + \Delta P_{\text{imb3}} \tag{18}$$

3.5. GES generator power ramping and switching module

To realize the functionality of the above modules, this module drives synchronous generators (SGs) to adjust in proportion with their respective connectivity degrees as given in (19) derived from (12)–(18). With extremely large RES proportions in the system, to avoid RES curtailments as much as possible, GES mandates that all the SGs are required to decrease their power output to their respective lower limits. The specific adjustment method of generator active power can be expressed as:

$$\begin{cases} \Delta P_{g,i} = ratio\left(\Delta P_{g,i}^{\max} - \Delta P_{g,i}^{\min} \right) \frac{\boldsymbol{D}_{g,i}}{\overline{\boldsymbol{D}}} + \Delta P_{g,i}^{\min} \\ ratio = \begin{cases} \frac{\left(\Delta P_{\text{imb}} - \Delta P_{\text{r},t+1}^{\max} \right) - \Delta P_{\text{T},t+1}^{\min}}{\Delta P_{\text{T},t+1}^{\max} - \Delta P_{\text{T},t+1}^{\min}}, P_{\text{d}} \le \Delta P_{\text{imb}} \\ \frac{\left(\Delta P_{\text{imb}} - \Delta P_{\text{T},t+1}^{\min} \right) - \Delta P_{\text{r},t+1}^{\min}}{\Delta P_{\text{r},t+1}^{\max} - \Delta P_{\text{r},t+1}^{\min}}, P_{\text{d}} > \Delta P_{\text{imb}} \end{cases} \\ P_{d} = \Delta P_{\text{r},t+1}^{\max} + \Delta P_{\text{T},t+1}^{\min} \end{cases} \tag{19}$$

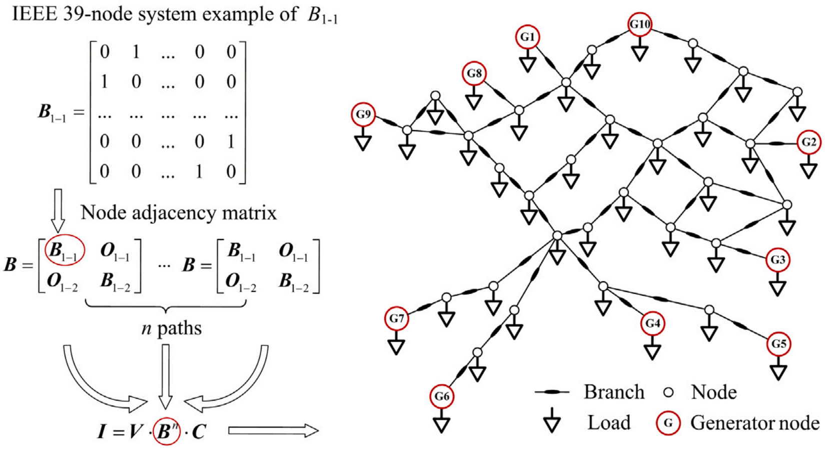

IEEE 39-node system example of $B_{1\text{-}1}$
$\boldsymbol{B}_{1-1}=\begin{bmatrix}0&1&\ldots&0&0\\1&0&\ldots&0&0\\\ldots&\ldots&\ldots&\ldots&\ldots\\0&0&\ldots&0&1\\0&0&\ldots&1&0\end{bmatrix}$
Node adjacency matrix
$\boldsymbol{B}=\begin{bmatrix}\boldsymbol{B}_{1-1}&\boldsymbol{O}_{1-1}\\\boldsymbol{O}_{1-2}&\boldsymbol{B}_{1-2}\end{bmatrix}\cdots\boldsymbol{B}=\begin{bmatrix}\boldsymbol{B}_{1-1}&\boldsymbol{O}_{1-1}\\\boldsymbol{O}_{1-2}&\boldsymbol{B}_{1-2}\end{bmatrix}$
n paths
$\boldsymbol{I}=\boldsymbol{V}\cdot\boldsymbol{B}^{n}\cdot\boldsymbol{C}$
G1
G10
G8
G9
G2
G3
G7
G4
G5
G6
Branch
Node
Load
Generator node

Fig. 2. IEEE 39-node system model establishment via graph theory.

where $\overline{\boldsymbol{D}}$ denotes the average value of $\boldsymbol{D}$.

Even after these modules, the system may still have unresolved imbalances due to limitations in its balancing capacity. To maintain the largest margin of the slack node and reduce/eliminate the demand for primary frequency response services, this paper employs pure SGs switching actions to tackle the system remaining power imbalance P_{s} as follows:

$$P_{\mathrm{s}}=\begin{cases}\Delta P_{\mathrm{imb}}-\sum_{i\in G}U_i^{\mathrm{G}}\Delta P_{g,i}^{\max}, & \Delta P_{\mathrm{imb}}>0\\ \Delta P_{\mathrm{imb}}-\sum_{i\in G}U_i^{\mathrm{G}}\Delta P_{g,i}^{\min}, & \Delta P_{\mathrm{imb}}<0\end{cases} \tag{20}$$

$$U_i^{\mathrm{G}}=\begin{cases}1, \quad S_i=1\cap\min\left|P_{\mathrm{s}}-\Delta P_{g,i}^{\mathrm{s}}\right|, & \forall i\in\boldsymbol{G}_{\mathrm{c}}\\ 0, \quad C_i=1\cap\min\left|P_{\mathrm{s}}-\Delta P_{g,i}^{\min}\right|, & \forall i\in\boldsymbol{G}_{\mathrm{o}}\end{cases} \tag{21}$$

where U_i^{G} is 1 or 0 indicates startup or shutdown of generator i.

Overall, the GES is summarized in Algorithm 1, where $T=288$ denotes one episode of 24 h with a time resolution of 5 min and a total of 288 time steps, and L is the total number of branches within the entire system. The specific steps of the GES are as follows: (1) Updating the grid model using (12) in Section 3.1. (2) optimizing the grid overload using (13)–(14) in Section 3.2. (3) Averaging branch loading ratio using (15) in Section 3.3. (4) Quantifying power imbalance using (16)–(18) in Section 3.4. (5) Optimizing SGs power ramping and switching using (19)–(21) in Section 3.5. (6) Integrating actions to generate a dispatch solution $\Delta P_{g,i}$, U_i^{G}.

Algorithm 1 Grid Expert Strategy

```
Input: $P_{\mathrm{G}}, R, \Delta P_{G,t+1}^{\max}, \Delta P_{G,t+1}^{\min}, T_{\mathrm{c}}, P_{r,t+1}^{\max}, \Delta P_{\mathrm{load}}^{t+1}, U^{\mathrm{G}}$, and $U^{\mathrm{L}}$
Output: $\Delta P_{g,i}$ and $U_i^{\mathrm{G}}$
for $t$=1 to $T$ do
    for $l$=1 to $L$ do
        Process (12) to update the grid model
    end for
    Process (13)–(14) to optimize grid loading
    Process (15) to average branch loading ratio
    Process (16)–(18) to quantify power imbalance
    Process (19)–(21) to optimize SGs power ramping and switching
end for
return $\Delta P_{g,i}$, $U_i^{\mathrm{G}}$
```

4. GESIL training and implementation

As introduced in Section 1, to improve the computational efficiency and generalization capability of the GESIL-based dispatch, in this paper, GES is used as the expert strategy, and IL is employed for offline–online learning. This section includes the specific design of IL agent as well as the GESIL application in a practical power system.

4.1. IL agent design principle for GESIL

4.1.1. The state-space and action-space of proposed IL agent

The state space of IL agent for the proposed GESIL is designed to identify the key information influencing dispatch decisions. Aimed at the grid's state information utilized by GES in (12)–(21), the state space $\boldsymbol{S}$ of IL can be established as:

$$\boldsymbol{S}=\Big[\boldsymbol{P}_G,\boldsymbol{R},\Delta\boldsymbol{P}_{G,t+1}^{\max},\Delta\boldsymbol{P}_{G,t+1}^{\min},\boldsymbol{T}_c,\boldsymbol{P}_{r,t+1}^{\max}, \\ \Delta\boldsymbol{P}_{\mathrm{load}}^{t+1},\boldsymbol{U}^G,\boldsymbol{U}^L,t\Big] \tag{22}$$

Based on (2)–(6), the action space $\boldsymbol{A}$ of the IL agent satisfying the grid operation constraints can be constructed as:

$$\boldsymbol{A}=\left[U_1^{\mathrm{G}}\Delta P_{g1},U_2^{\mathrm{G}}\Delta P_{g2},\ldots,U_i^{\mathrm{G}}\Delta P_{g,i}\right] \tag{23}$$

4.1.2. The loss function of proposed IL agent

Using the state $\boldsymbol{s}$ corresponding to the state space $\boldsymbol{S}$ as the input data and the action $\boldsymbol{a}$ corresponding the action space $\boldsymbol{A}$ as the label data, the IL agent training can be formulated as a regression problem. The policy trajectory $\pi(s)$ of the IL agent can be trained by the first-order optimization method of stochastic gradient descent [38]. Compared to the modules of graph theory-based grid modeling, contingency analysis and optimization, branch loading ratio averaging, power balancing control, and generator power ramping, which can be addressed by a conventional loss function, to embed the generator switching module as proposed in Section 3.5, a specific loss function $L(\theta)$, with an additional penalty factor added in the second term of a traditional loss function, is designed to process (23) and (24), expressed as:

$$L(\theta)=\frac{1}{J}\sum_{j=1}^{J}\left\|\pi\left(\boldsymbol{s}_j\right)-\boldsymbol{a}_j\right\|_2^2+ \\ \frac{\lambda}{J}\sum_{j=1}^{J}\left\|\left(\mu_j-\boldsymbol{U}_j^{\mathrm{G}}\right)\left(\pi\left(\boldsymbol{s}_j\right)-\boldsymbol{a}_j\right)\right\|_2^2+\frac{\beta}{2}\|\theta\|_2^2 \tag{24}$$

where θ denotes the neural network parameters of the IL agent to be trained, the first term of the loss function captures the deviations between the GES and IL agent, with $\pi(\boldsymbol{s}_j)$ as the IL dispatch policy trajectory, $\boldsymbol{a}_j$ as the GES dispatch policy, J as the IL training batch size, and $\|\cdot\|^2$ as the square of the euclidean parametrization; the second term of the loss function caters for the system power balancing, with λ as the penalty factor and μ_j as an array of values 1 at the same length of $\boldsymbol{U}_j^{\mathrm{G}}$; the third term of the loss function is a regular term to prevent the IL agent from being overfitted during training, with β as a constant value (generally greater than 0 and is recommended as 0.02 in this paper).

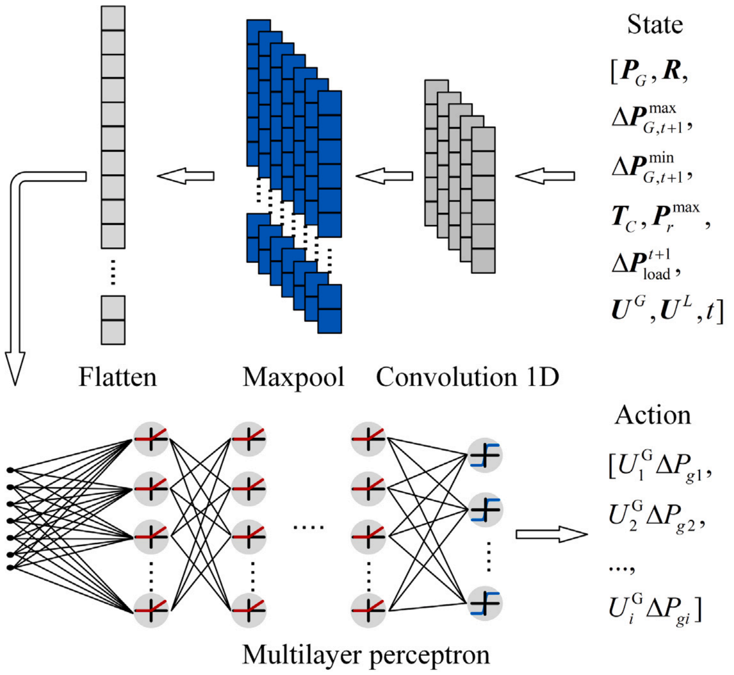

State
$[\boldsymbol{P}_G, \boldsymbol{R},$
$\Delta\boldsymbol{P}_{G,t+1}^{\max},$
$\Delta\boldsymbol{P}_{G,t+1}^{\min},$
$\boldsymbol{T}_C, \boldsymbol{P}_r^{\max},$
$\Delta\boldsymbol{P}_{\text{load}}^{t+1},$
$\boldsymbol{U}^G, \boldsymbol{U}^L, t]$
Flatten
Maxpool
Convolution 1D
Action
$[U_1^G \Delta P_{g1},$
$U_2^G \Delta P_{g2},$
$\ldots,$
$U_i^G \Delta P_{gi}]$
Multilayer perceptron

Fig. 3. IL model of GESIL.

4.1.3. *The structure of proposed IL agent*

Fig. 3 shows the structure of the proposed IL agent. The grid operation states are firstly classified by convolutional neural network (CNN), and the states after a flattening process are then input to multi-layer perceptron (MLP) for the IL agent to generate the dispatch decisions. The activation functions of rectified linear unit (ReLU) and hyperbolic tangent (Tanh) [39], are used in the MLP's hidden layer and the output layer, respectively.

4.2. *GESIL training and its power system application*

As introduced in Section 1, the proposed GESIL, which is essentially a trained IL agent by the five GES modules as proposed in Sections 3.1–3.5, are illustrated in the flow chart of Fig. 4. The specific training steps of the GESIL are as follows: (1) GES interacts with the grid environment to collect expert experiences via (12)–(21), which are stored in the expert replay buffer as $\boldsymbol{D}^m$; (2) IL agent offline training using BC algorithm via (24); (3) IL agent interacts with the grid environment to generate $\pi(\boldsymbol{s}_m)$, and returns to the grid environment for $\boldsymbol{s}_{m+1}$ as in (22)–(23); (4) GES processes $\boldsymbol{s}_{m+1}$, generates $\boldsymbol{a}_{m+1}$, and updates $\boldsymbol{D}^{m+1}$ in real-time to realize the online demonstration and correction via (12)–(21); (5) IL agent online training using $\boldsymbol{D}^{m+1}$ and the DAgger algorithm as in (24). These steps (3)–(5) are repeated until the required number of iterations is met. Such an offline–online training process can significantly reduce the training time and improve the generalization capability of the proposed GESIL.

5. Case study

To validate the proposed GESIL scheme, case studies are conducted using the IEEE 118-node test system [40] as shown in Fig. 5, which represents a portion of American Electric Power Company containing 54 generators, 186 branches, and 99 load nodes, and it can be divided into three regions by enveloping curves. To model a high proportion of RES generation, as shown in Fig. 5, the SGs at nodes 1, 10, 12, 25, 26, 46, 80, and 100 are replaced by individual PV plants, whereas the SGs at nodes 4, 6, 15, 18, 19, 27, 31, 32, 73, 99 and 116 are replaced by individual wind farms, with the rated capacities of the individual RES generators 1.4 times of the original SGs, achieving 37.4% RES generation of the total system installed generation capacity. All the SG ramp rate as introduced in (4) is set as $r_{g,i} = 0.05P_{g,i}^{\max}$ per step of 5 min [41], and their minimum shutdown time steps to fulfill unit commitment as introduced in (6) are simulated as $T_c = 40$ (200 min in total). The adopted power profiles of RES generations and loads are publicly available in [42] for specific dates to be selected in the following case studies. The PYPOWER [43], a Python version of the widely used MATPOWER [44], is employed in this section as a simulator for grid environment modeling.

The GESIL training parameters is set up as: $\theta = -10$, $a_1 = 1$, $a_2 = 2$, and $a_3 = 1$ as introduced in (11), $n = 2$ as introduced in (12), $\lambda = 2$ and $\beta = 0.02$ as introduced in (24), respectively. The remaining parameters are presented in Appendix A.1. The dynamic curves of the loss function in the training phase utilizing both the BC algorithm and the DAgger algorithm are depicted in Figs. A.1 and A.2 in Appendix A.2. The specific SAC parameters are set out in Appendix A.3.

The following case studies will compare the proposed GESIL with the traditional grid expert strategy (TGES) [5] and the SAC-based grid dispatch method [21] in terms of algorithm performance and effectiveness in grid operation optimization and power balancing control.

5.1. *Algorithm performance*

Algorithm performance is primarily evaluated by their training convergence and generalization capability.

5.1.1. *Training convergence*

To train GESIL agent, for one dispatch day, the power profiles of RES generations and loads, with 5 min intervals, are selected from 25 randomly-selected dispatch days from each month of March, June, September, and December 2021. The GESIL and SAC use the same state space as in (22) and action space as in (23) to train.

Firstly, the GESIL is trained offline for 600 iterations using the BC algorithm and the loss function in (24). Secondly, the GESIL is further trained online using the DAgger algorithm and also the loss function in (24). As GESIL is trained offline during the BC algorithm training process and cannot provide real-time feedback on reward values, this

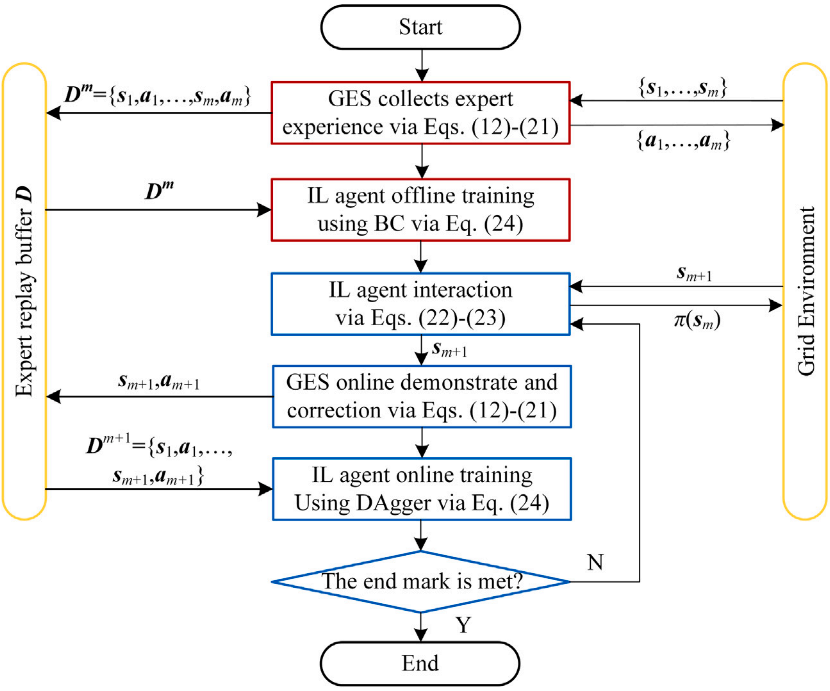

Start
GES collects expert experience via Eqs. (12)-(21)
$D^m=\{s_1,a_1,\ldots,s_m,a_m\}$
$\{s_1,\ldots,s_m\}$
$\{a_1,\ldots,a_m\}$
D^m
IL agent offline training using BC via Eq. (24)
IL agent interaction via Eqs. (22)-(23)
s_{m+1}
$\pi(s_m)$
s_{m+1}
GES online demonstrate and correction via Eqs. (12)-(21)
s_{m+1},a_{m+1}
$D^{m+1}=\{s_1,a_1,\ldots,s_{m+1},a_{m+1}\}$
IL agent online training Using DAgger via Eq. (24)
The end mark is met?
N
Y
End
Expert replay buffer D
Grid Environment

Fig. 4. Flow chart of GESIL in real-time grid dispatch.

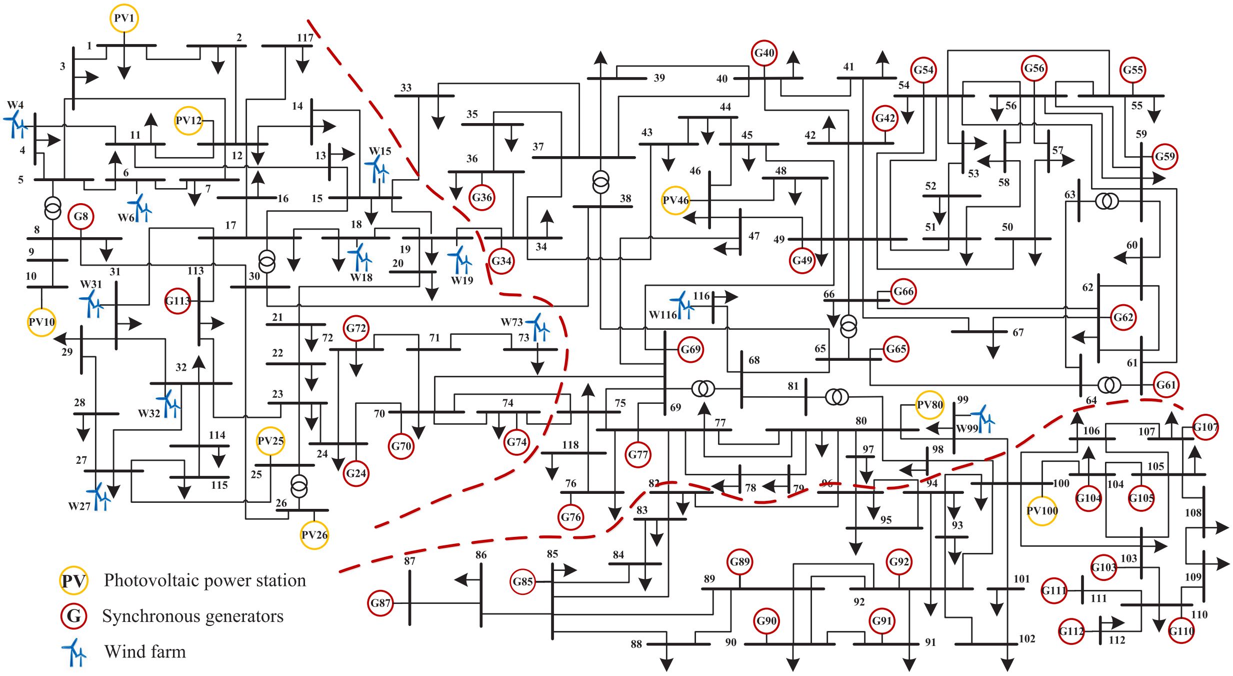


Fig. 5. Modified IEEE 118-node system model.

subsection focuses on the convergence comparisons of GESIL during the DAgger-based online training and SAC.

To demonstrate the stability of the GESIL during the training process, it is trained for the same extended duration as the SAC, which undergoes trainings for 5×10^5 episodes. Each episode consists of 288 steps and is run on a randomly-selected dispatch day, as shown in Fig. 6. Each reward value shown in the figure represent the average reward for one episode. GESIL, which is pre-trained using the BC algorithm, initially achieved a high reward value of 1.6, and gradually improves the reward value to 1.72 via the subsequent DAgger-based training within 4000 episodes as per the training procedures designed in Section 4.2 and Fig. 4. In contrast, the SAC-based training resulted in an average reward of 1.32, significantly lower than GESIL in the entire training results of long 5×10^5 episodes.

In terms of training time, GESIL takes 3.2 h to reach a stable convergence (approximately 4000 episodes), whereas SAC takes 46.5 h to complete the training, 14.5 times slower than GESIL.

5.1.2. Generalization capability

To compare GESIL's generalization capability with those of the SAC and TGES schemes, ten dispatch days are selected, and a different randomly occurring event is designed for each dispatch day,

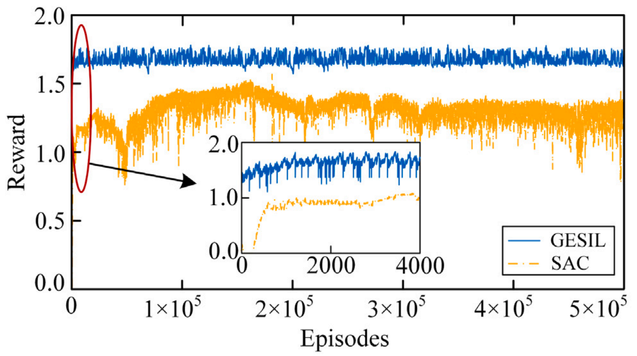


Fig. 6. Comparison of training convergence.

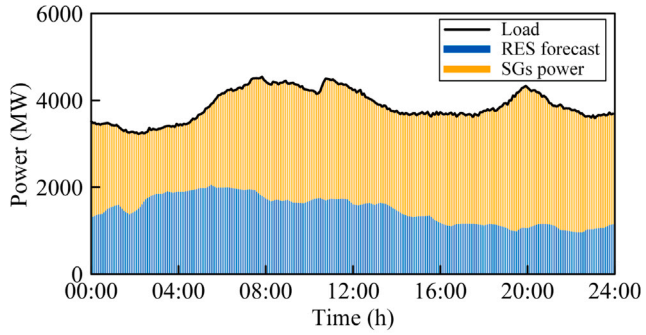


Fig. 8. System operating data on 9 Jun. 2021.

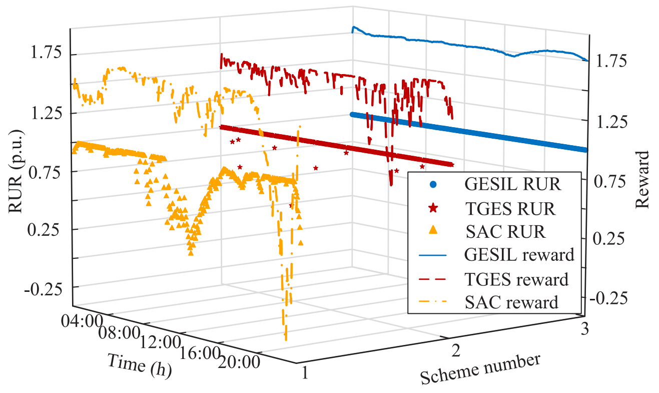


Fig. 7. Comparison of generalization capability on a dispatch day of 5 Mar. 2021.

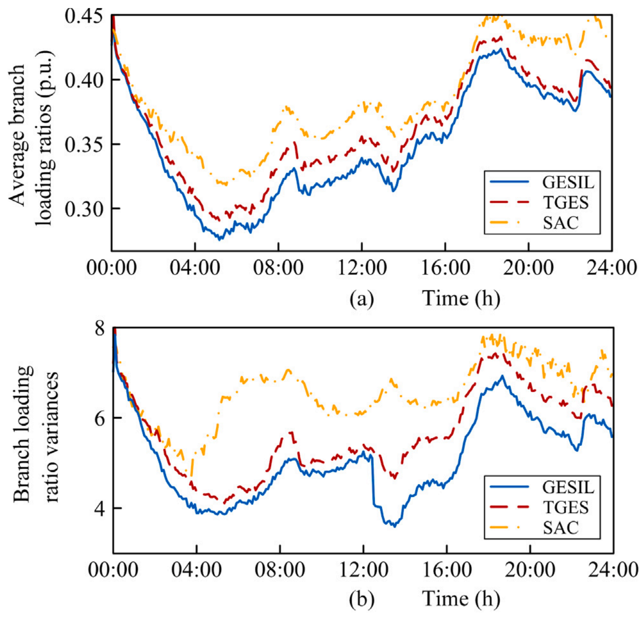


Fig. 9. Comparison of loading ratio averaging. (a) Average branch Loading ratios. (b) Branch loading ratio variances.

Table 2
Generalization capability comparisons in 10 dispatch days.

	GESIL	TGES	SAC
Days with power flow convergence	10 days	8 days	7 days
Average reward	1.73	1.56	1.26
RUR	99.98%	99.96%	92.37%

respectively. These ten events included: circuit disconnections of the branches 43, 44, 113, 114, and 118 lasting for 80 mins, a 15% steep rise in total RES generation, a 15% steep fall in total RES generation, a 10% step increase in total system load, the disconnection of generator 5 occupying 11.3% of the total generation power output, and the disconnection of generator 16 occupying 2% of the total generation.

Under these dispatch days with various grid disturbance events, the grid simulator solves power flows based on different dispatch decisions returned by the three aforementioned schemes, respectively. Fig. 7 shows the detailed renewable utilization rates (RUR) and average rewards under the three schemes for the dispatch day of 5 Mar. 2021, embodying GESIL's superior generalization capability compared to TGES and SAC. The results of the three schemes, in terms of power flow convergence, average reward, and RUR in the corresponding ten dispatch days, are detailed in Table 2. The convergences of SAC, TGES and GESIL are 7, 8 and 10, respectively, with GESIL being successfully convergent in all the ten dispatch days. The average reward values of TGES, SAC and GESIL are 1.56, 1.26 and 1.73, respectively, with GESIL having the highest average reward. Both GESIL and TGES achieve a relatively high RUR of 99.98% and 99.96%, respectively, whereas the value of SAC only reaches 92.37%.

5.1.3. *Single-step decision time*

To evaluate algorithm operating efficiency, this subsection first analyzes and compares the computational complexities of the three aforementioned schemes using the big O notation. The CC of GESIL is $O\left(n^2\right)$, TGES is $O\left(2^n n \log n\right)$ [45], and SAC is $O\left(n^2\right)$. Subsequently, the elapsed time of single-step decision-making is compared under the three aforementioned schemes. The computing server consists of a 12-core CPU at 2.50 GHz and 8 GB memory. As a result, TGES has an average single-step decision time of 2.2 s, whereas GESIL and SAC take 0.13 s and 0.15 s, respectively. This indicates GESIL outperforms TGES by approximately 17 times. Although GESIL and SAC have similar decision times, as concluded in Sections 5.1.1 and 5.1.2, the GESIL algorithm generally has a superior performance in terms of training efficiency and generalization capability over SAC.

5.2. *Grid operation optimization*

This subsection evaluates the grid operation optimization performance of the studied three schemes in averaging grid loading ratios, handling grid overloading scenarios, and slack generation regulation margin optimization. The historical energy scenario on 5 Mar. 2021 is adopted as shown in Fig. 8.

5.2.1. *Grid real-time loading ratio averaging*

Fig. 9(a) shows the average loading ratio values across all 186 branches in the 118-node system during a dispatch day. GESIL efficiently diverts power flows to the branches with larger transmission

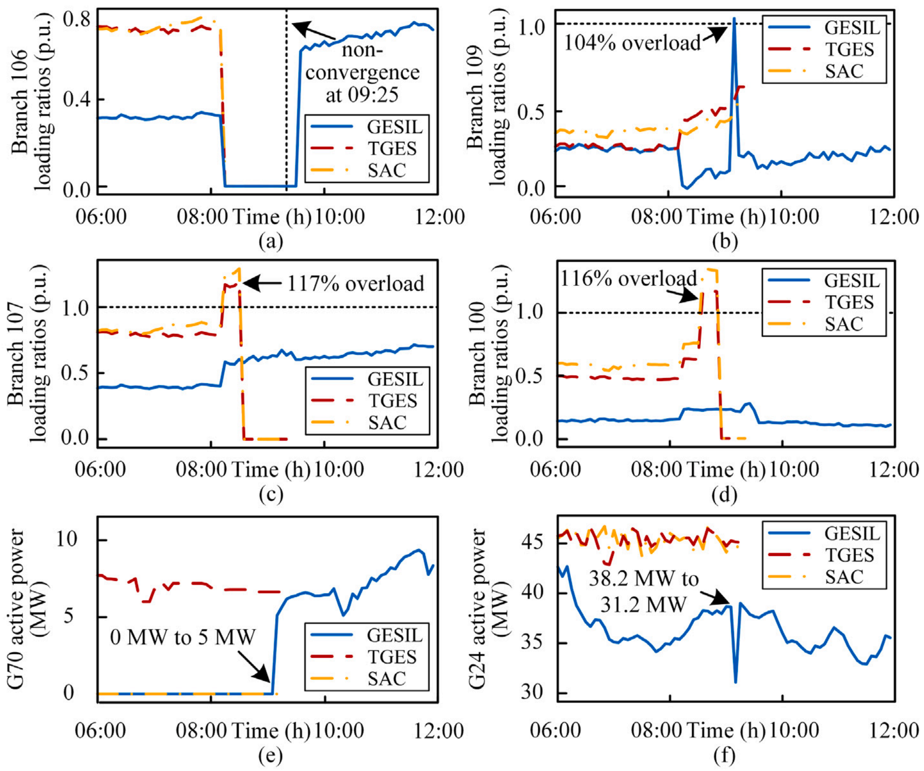


Fig. 10. Grid overloading in 06:00–12:00. (a) Branch 106 loading ratios. (b) Branch 109 loading ratios. (c) Branch 107 loading ratios. (d) Branch 100 loading ratios. (e) G70 active power. (f) G24 active power.

capacity via (19), resulting in a significantly lower loading ratio compared to TGES and SAC. Fig. 9(b) shows the branch loading ratio variances, that the lower the loading ratio variance the more evenly the power loadings are distributed among all the branches. As can be seen, GESIL achieves a significantly lower branch load ratio variance than TGES and SAC.

5.2.2. Grid overloading optimization

To compare the grid overloading handling capability, branch 106 is disconnected at 08:20 and restored at 09:40, As shown in Fig. 10(a), the loading ratios of branch 106 are all decreased immediately to zero at 08:20 with GESIL, TGES, and SAC.

As shown in Fig. 10, GESIL is capable of maintaining the average grid branch load ratio overall at a low level via (19) at the pre-contingency phase, preventing potential branch overloading due to the contingency. However, at 09:15 when the branch 106 is not restored in operation yet, the branch 109 becomes cascadedly overloading under the sharp rise in loading, as depicted in Fig. 10(b). During the contingency, the proposed GESIL uses the grid model based on graph theory to obtain G70 downstream, G24 and G72 upstream of branch 109 via (12), and adjusts generators' outputs with the highest sensitivities on the overloading branch, the active power of G70 increased from 0 MW to 5 MW, G24 decreased from 38.2 MW to 31.2 MW, and G72 decreased from 99.3 MW to 94.3 MW, as shown in Fig. 10(e) and (f) (due to spatial limitations, G72 are not presented in Fig. 10). As a result, GESIL eliminates the overloading of branch 109 at 09:20 and avoids the cascaded circuit trips.

Fig. 10(c) and (d) show the loading ratios of branches 107 and 100 (the other branches 108, 26, 30, and 35 exhibit similar performance as discussed later). As can be seen, SAC and TGES schemes apply with no effect in optimizing the grid loading ratio; as a result, the overloading of branch 107 at 117% is observed immediately after the fault in branch 106. At 08:40, branch 107 is disconnected, aggravating overloadings in the branches 100 and 108 at 116% and 105%, respectively. At 09:00, branches 100 and 108 are cascadedly disconnected, further causing the overloadings of the branches 26, 30, and 35, eventually leading to their disconnection at 09:20 (due to spatial limitations, detailed results are not presented in Fig. 10). Finally, at 09:25, the chain of these disconnections occurs in the overall grid, resulting, in an N-14 fault and the non-convergence of power flow (physically a power blackout).

5.2.3. Slack generation regulation margin optimization

Fig. 11 illustrates the active power outputs from the slack generator located at node 69, which operates within the active power range of 0 to 805 MW. As observed, the active power output from the slack generator under TGES has the largest deviation from the neutral point of 402.5 MW. Under SAC, the active power output from the slack generator experiences dramatic fluctuation. In particular, there is a sharp active power drop of 300 MW at 08:00 (attributed to the instability of the trained SAC agent), implying a high risk of system power imbalance. In comparison, the proposed GESIL maintains the active power output from the slack generator continuously around the neutral point of 402.5 MW, effectively maximizing its upper and lower reserve margins via (16)–(21) for governing the grid overall power balance.

5.3. Discussion

This paper verifies the superiority of GESIL in terms of algorithm performance and grid operation optimization based on the IEEE 118 system in Sections 4.1 and 4.2. GESIL is also capable of handling increasing grid size and complexity without significantly raising computational demands. It demonstrates strong generality and applicability, with its core formulas (12)–(24) incorporating a graph theory-based power grid model, model-driven pure rule-based GES, and GESIL-based implementation scheme, all of which are not constrained by the size

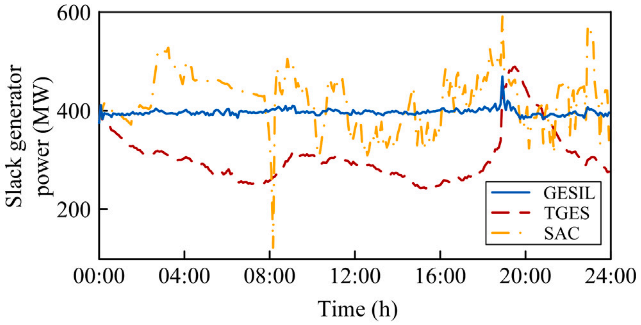


Fig. 11. Comparison of slack generation regulation margin optimization.

of the power grid or the number of nodes. In the future, methods like transfer learning [46] can be utilized to further enhance the applicability of GESIL to larger-scale power systems.

6. Conclusion

This paper presents a novel real-time power system dispatch scheme using grid expert strategy-based imitation learning (GESIL), which has the following features: (1) GESIL constructs a grid model through graph theory, which enables the real-time acquisition of correlation information for grid nodes, generators, and branches; (2) GESIL designs a grid expert strategy (GES) based on this model, effectively optimizing grid operation and facilitating power balance control; (3) GESIL effectively fuses imitation learning with the GES, finalizing an essentially model-data-driven real-time dispatch scheme.

By comparing the performances of GESIL with those of a model-driven Traditional Grid Expert strategy (TGES) and a data-driven Soft Actor–Critic (SAC), the results verify that GESIL can effectively reduce grid loading ratio, average power flow distributions amongst all the regulated circuits, and balance system power imbalance in real-time, which are not achieved by TGES. Additionally, GESIL exhibits a computational efficiency enhancement of approximately 17 times compared to TGES. Also, GESIL can perform online learning based on GES, improving the efficiency of agent action space exploration and enhancing generalization capability, which are not completely addressed by SAC. Moreover, GESIL's training speed is improved by 14.5 times compared to SAC. In future, the proposed GESIL can act a real-time power dispatch solution to secure the system operations under high penetration of renewable energy. This dispatching method will be further researched for making efficient decisions in the face of cyber attacks.

CRediT authorship contribution statement

Siyang Xu: Writing – original draft, Validation, Methodology. **Jiebei Zhu:** Writing – review & editing, Conceptualization. **Bingsen Li:** Conceptualization. **Lujie Yu:** Data curation. **Xueke Zhu:** Investigation. **Hongjie Jia:** Formal analysis. **Chi Yung Chung:** Formal analysis. **Campbell D. Booth:** Visualization. **Vladimir Terzija:** Visualization.

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.

Data availability

All data used have been cited and described in the manuscript.

Table 3
Description of GESIL training parameters.

	GESIL
Encoder output	128 with 10 encoders
Implicit layer	2048,2048,2048,1024,1024,512,512,256,256
Output dimension	53
Learning rate in BC	1e−4
Batch size in BC	256
Learning rate in DAgger	5e−5
Batch size in DAgger	64

Table 4
Hyperparameters setting of the SAC algorithm.

	SAC
Actor network structure	1280,2048,2048,1024,512,256
Critic network structure	1333,2048,1024,256
Discount factor γ	0.98
Whether to use PER	Yes
Explore noise benchmark values	0.2
Reward scale	0.2
Batch size	64

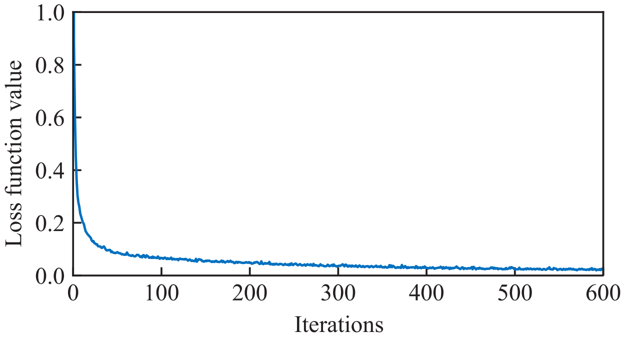


Fig. A.1. Loss function value in BC algorithm training stage.

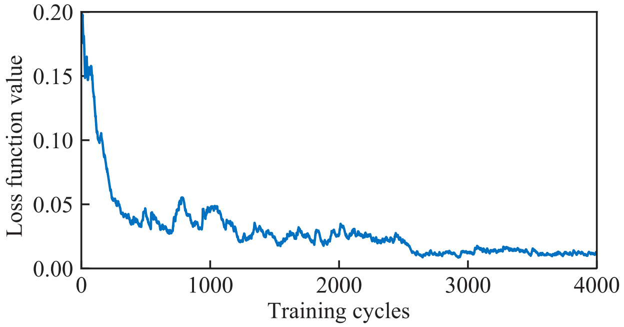


Fig. A.2. Loss function value in DAgger algorithm training stage.

Appendix

A.1. GESIL training parameters

See Table 3.

A.2. GESIL's loss function values in the training phase

See Figs. A.1 and A.2.

A.3. SAC hyperparameters

See Table 4.

References

[1] Hassanzadeh ME, Nayeripour M, Hasanvand S, Sepehrzad R. Hierarchical optimal allocation of BESS using APT-FPSO based on stochastic programming model considering voltage sensitivity and eigenvalues analyses. Int J Electr Power Energy Syst 2023;153:109291.

[2] Heffron R, McCauley D, Zarazua G. Balancing the energy trilemma through the energy justice metric. Appl Energy 2018;229:1191–201.

[3] Xie M, Xiong J, Ke S, Liu M. Two-stage compensation algorithm for dynamic economic dispatching considering copula correlation of multiwind farms generation. IEEE Trans Sustain Energy 2016;8(2):763–71.

[4] Liu F, Bie Z, Wang X. Day-ahead dispatch of integrated electricity and natural gas system considering reserve scheduling and renewable uncertainties. IEEE Trans Sustain Energy 2018;10(2):646–58.

[5] Hlalele G, Naidoo M, Bansal C, Zhang J. Multi-objective stochastic economic dispatch with maximal renewable penetration under renewable obligation. Appl Energy 2020;270:115120.

[6] Wang Z, Shen C, Liu F, Wu X, Liu C-C, Gao F. Chance-constrained economic dispatch with non-Gaussian correlated wind power uncertainty. IEEE Trans Power Syst 2017;32(6):4880–93.

[7] Li P, Wu Q, Yang M, Li Z, Hatziargyriou ND. Distributed distributionally robust dispatch for integrated transmission-distribution systems. IEEE Trans Power Syst 2020;36(2):1193–205.

[8] Chen Y, Guo Q, Sun H, Li Z, Wu W, Li Z. A distributionally robust optimization model for unit commitment based on Kullback–Leibler divergence. IEEE Trans Power Syst 2018;33(5):5147–60.

[9] Zare A, Chung CY, Zhan J, Faried SO. A distributionally robust chance-constrained MILP model for multistage distribution system planning with uncertain renewables and loads. IEEE Trans Power Syst 2018;33(5):5248–62.

[10] Li Y, Han M, Shahidehpour M, Li J, Long C. Data-driven distributionally robust scheduling of community integrated energy systems with uncertain renewable generations considering integrated demand response. Appl Energy 2023;335:120749.

[11] Sepehrzad R, Ghafourian J, Hedayatnia A, Al-Durrad A, Khooban MH. Experimental and developed DC microgrid energy management integrated with battery energy storage based on multiple dynamic matrix model predictive control. J Energy Storage 2023;74:109282.

[12] Papavasiliou A, Mou Y, Cambier L, Scieur D. Application of stochastic dual dynamic programming to the real-time dispatch of storage under renewable supply uncertainty. IEEE Trans Sustain Energy 2017;9(2):547–58.

[13] Tang Y, Dvijotham K, Low S. Real-time optimal power flow. IEEE Trans Smart Grid 2017;8(6):2963–73.

[14] Sepehrzad R, Hedayatnia A, Amohadi M, Ghafourian J. A al-durra and a anvari-moghaddam, two-stage experimental intelligent dynamic energy management of microgrid in smart cities based on demand response programs and energy storage system participation. Int J Electr Power Energy Syst 2024;155:109613.

[15] Zhou Y, Zhang B, Xu C, Lan T, Diao R, Shi D, et al. A data-driven method for fast AC optimal power flow solutions via deep reinforcement learning. J Mod Power Syst Clean Energy 2020;8(6):1128–39.

[16] Bui H, Hussain A, Kim M. Double deep Q-learning-based distributed operation of battery energy storage system considering uncertainties. IEEE Trans Smart Grid 2019;11(1):457–69.

[17] Sepehrzad R, Langeroudi ASG, Khodadadi A, Adinehpour S, Durra AAl, Moghaddam AA. An applied deep reinforcement learning approach to control active networked microgrids in smart cities with multi-level participation of battery energy storage system and electric vehicles. Sustainable Cities Soc 2024;107:105352.

[18] Xu J, Gao H, Wang R, et al. Real-time operation optimization in active distribution networks based on multi-agent deep reinforcement learning. J Mod Power Syst Clean Energy 2023.

[19] Zhang H, Yue D, Dou C, et al. Resilient optimal defensive strategy of TSK fuzzy-model-based microgrids' system via a novel reinforcement learning approach. IEEE Trans Neural Netw Learn Syst 2021;34(4):1921–31.

[20] Liu Z, Liu Y, Xu H, Liao S, Zhu K, Jiang X. Dynamic economic dispatch of power system based on DDPG algorithm. Energy Rep 2022;8:1122–9.

[21] Zhou Y, Ma Z, Zhang J, Zou S. Data-driven stochastic energy management of multi energy system using deep reinforcement learning. Energy 2022;261:125187.

[22] Wang S, Duan J, Shi D, Xu C, Li H, Diao R, et al. A data-driven multi-agent autonomous voltage control framework using deep reinforcement learning. IEEE Trans Power Syst 2020;35(6):4644–54.

[23] Zhu Z, Chan KW, Xia S, Bu S. Optimal bi-level bidding and dispatching strategy between active distribution network and virtual alliances using distributed robust multi-agent deep reinforcement learning. IEEE Trans Smart Grid 2022;13(4):2833–43.

[24] Mu C, Shi Y, Xu N, Wang X, Tang Z, Jia H, et al. Multi-objective interval optimization dispatch of microgrid via deep reinforcement learning. IEEE Trans Smart Grid 2024;15(3):2957–70.

[25] Khodadadi A, Adinehpour S, Sepehrzad R, Durra AA, Moghaddam AA. Data-driven hierarchical energy management in multi-integrated energy systems considering integrated demand response programs and energy storage system participation based on MADRL approach. Sustainable Cities Soc 2024;105264.

[26] Hu D, Ye Z, Gao Y, Ye Z, Peng Y, Yu N. Multi-agent deep reinforcement learning for voltage control with coordinated active and reactive power optimization. IEEE Trans Smart Grid 2022;13(6):4873–86.

[27] Paine TL, Gulcehre C, Shahriari B, Denil M, Hoffman M, Soyer H, et al. Making efficient use of demonstrations to solve hard exploration problems. In: Proc int conf learn represent. 2019.

[28] Zheng B, Verma S, Zhou J, Tsang IW, Chen F. Imitation learning: Progress, taxonomies and challenges. IEEE Trans Neural Netw Learn Syst 2024;35(5):6322–37.

[29] Osa T, Pajarinen J, Neumann G, Bagnell JA, Abbeel P, Peters J. An algorithmic perspective on imitation learning. Found Trends Robot 2018;7(1–2):1–179.

[30] Hussein A, Gaber MM, Elyan E, Jayne C. Imitation learning: A survey of learning methods. ACM Comput Surv 2017;50(2):1–35.

[31] Guo W, Tian W, Ye Y, Xu L, Wu K. Cloud resource scheduling with deep reinforcement learning and imitation learning. IEEE Internet Things J 2020;8(5):3576–86.

[32] Modarresi MS, Xie L, Campi MC, Garatti S, Carè A, Thatte AA, et al. Scenario-based economic dispatch with tunable risk levels in high-renewable power systems. IEEE Trans Power Syst 2018;34(6):5103–14.

[33] Zou D, Gong D, Ouyang H. The dynamic economic mmission dispatch of the combined heat and power system integrated with a wind farm and a photovoltaic plant. Appl Energy 2023;351:121890.

[34] Qin J, Wan Y, Yu X, Kang Y. A Newton method-based distributed algorithm for multi-area economic dispatch. IEEE Trans Power Syst 2019;35(2):986–96.

[35] Gao S, Xiang C, Yu M, Tan KT, Lee TH. Online optimal power scheduling of a microgrid via imitation learning. IEEE Trans Smart Grid 2021;13(2):861–76.

[36] Osa T, Pajarinen J, Neumann G, Bagnell JA, Abbeel P, Peters J. An algorithmic perspective on imitation learning. Found Trends Robot 2018;7(1–2):1–179.

[37] Haarnoja T, Zhou A, Hartikainen K, Tucker G, Ha S, Tan J, et al. Soft actor-critic algorithms and applications. 2018, arXiv preprint arXiv:1812.05905.

[38] Ho J, Gupta J, Ermon S. Model-free imitation learning with policy optimization. In: Proc int conf mach learn. 2016, p. 2760–9.

[39] Dubey SR, Singh SK, Chaudhuri BB. Activation functions in deep learning: A comprehensive survey and benchmark. 2022, arXiv preprint arXiv:2109.14545v2.

[40] Christie Rich. Power Systems Test Case Archive. [Online]. Available: http://labs.ece.uw.edu/pstca/pf118/pgtca118bus.htm.

[41] Che L, Liu X, Shuai Z, Zhao J. The impact of ramp-induced data attacks on power system operational security. IEEE Trans Ind Informat 2019;15(9):5064–75.

[42] Todays California ISO website. [Online]. Available: http://www.caiso.com/TodaysOutlook/Pages/default.aspx.

[43] Lincoln Richard. Pypower. 2017, Available: https://github.com/rwl/PYPOWER.

[44] Zimmerman RD, Murillo-Sánchez CE, Thomas RJ. MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. IEEE Trans Power Syst 2010;26(1):12–9.

[45] Zhao B, Shi Y, Dong X, et al. Short-term operation scheduling in renewable-powered microgrids: A duality-based approach. IEEE Trans Sustain Energy 2013;5(1):209–17.

[46] Ali AH, Yaseen MG, Aljanabi M, Abed SA. Transfer learning: A new promising techniques. Mesop J Big Data 2023;29–30.