Enhancing Power Grid Resilience Through Weather-Aware Security Constraints: A Deep Reinforcement Learning Approach with Hybrid CNN-GRU Architecture

Yingjun Wu[a], Junyu Feng[a], Xuejie Chen[a], Yujian Ye[b,*], Zhiwei Lin[c], Jiangfan Yuan[a], Xueyan He[a], Zhengxi Yin[a], Jiayan Lu[a]

[a]*School of Electrical and Power Engineering, Hohai University, Nanjing, Jiangsu 211100, China*
[b]*School of Electrical Engineering, Southeast University, Nanjing 210096, Chin*a
[c]*School of Engineering, Cardiff University, Cardiff CF24 3AA, Wales, UK*

Abstract: Extreme weather events increasingly challenge the operational resilience of distribution systems by introducing dynamic and uncertain security limits (SLs), alongside data sparsity. Traditional model-based approaches often rely on static assumptions and require complete system modeling, making them difficult to adapt to rapidly evolving weather-induced constraints. To address these limitations, this paper proposes a model-free resilience enhancement framework based on deep reinforcement learning (DRL), integrating real-time weather-aware SL identification and adaptive dispatch. First, an ensemble Bagging-XGBoost model is developed to classify weather severity levels and determine whether static or dynamic SLs should be applied, enabling scenario-adaptive SL switching. Second, a hybrid convolutional neural network–gated recurrent unit (CNN-GRU) model, enhanced by transfer learning, is designed to accurately estimate dynamic SLs under varying weather conditions. The CNN captures spatial meteorological patterns, while the GRU models temporal evolution; transfer learning improves generalization under limited training data. Third, the dispatch problem is formulated as a constrained Markov decision process (CMDP), and solved using a primal–dual deep deterministic policy gradient (PD-DDPG) algorithm that explicitly incorporates SL constraints into the policy learning process. An attention-based meteorological data reconstruction model is further integrated to enhance the quality of input data and training efficiency. Case studies on the improved IEEE-123 test feeder demonstrate that the proposed method reduces average load loss by 23.30% and 12.10% compared to CNN-only and GRU-only baselines, respectively. Moreover, it achieves an 88.77% improvement in computational efficiency over conventional model-based resilience strategies, highlighting its robustness and applicability under limited data and high-impact weather conditions.

Keywords: Grid resilience, extreme weather, security constraints, reinforcement learning, transfer learning.

1 Introduction

1.1 Background and Motivation

Extreme weather events—such as hurricanes (e.g., Yagi, Harvey) and ice storms (e.g., Uri)—have caused increasing disruptions to power distribution systems worldwide, leading to widespread blackouts, equipment failures, and economic losses exceeding hundreds of millions of dollars [1,2]. These events highlight the limitations of conventional recovery strategies, particularly in promptly restoring critical loads (CLs) under time-varying and uncertain operating conditions [3,4].

To address these challenges, enhancing resilience—defined as the system's ability to withstand, adapt to, and recover from disturbances—has become a key focus in distribution system operation and planning. Within the broader resilience enhancement process, real-time adaptive scheduling is particularly vital, as it governs the dynamic restoration of CLs and the efficient deployment of limited resources. However, such scheduling efforts are complicated by two interrelated technical barriers: The time-varying nature of security limits (SLs)—such as voltage and power flow

[*] Corresponding author.

E-mail addresses: yeyujian@seu.edu.cn

boundaries—that respond nonlinearly to environmental conditions (e.g., temperature, wind speed, icing); and the lack of sufficient data to model or estimate these dynamic SLs under rare, high-impact disaster events.

1.2 Literature Review

Prior efforts to enhance distribution system resilience under extreme weather conditions can be categorized into three major groups: (1) model-based methods focusing on physical mechanism modeling and optimization; (2) scenario-driven strategies relying on expert-defined control logic; and (3) model-free approaches employing data-driven learning frameworks.

(1) Model-based methods

Model-based approaches typically aim to enhance resilience by explicitly representing system physics and uncertainty, often requiring detailed knowledge of electrical equipment behavior under stress conditions. Early efforts focused on pre-disaster planning and post-disaster resource coordination [4]–[7], including vegetation maintenance [5], allocation of backup lines and transformers [6], and stochastic dispatch of mobile resources [7]. These methods usually rely on forecasted outage areas and known system parameters, which may be inaccurate or unavailable during actual disaster events.

To address the weather-induced variation of security limits (SLs), researchers have explored data-driven statistical models and physics-informed mechanisms. For instance, [8] and [9] investigated the relationship between corona loss, icing, and insulator flashover through controlled laboratory testing, revealing how surface discharge thresholds vary with ambient humidity and contamination. Building on these experimental insights, [10] and [11] developed statistical regression models to estimate flashover voltages, while [12] quantified the impact of ice accretion on line ampacity.

More recently, [13] and [14] proposed mechanism-based dynamic SL models that link environmental variables to power flow or voltage constraints through analytical formulations. These studies laid the foundation for online SL adjustment. In particular, [15] introduced a weather-aware SL identification model using field observations and supervised learning, providing a practical tool for SL estimation under environmental stress.

However, the main limitations of these model-based approaches are threefold:

1) First, forecast dependency: they require accurate meteorological forecasts to update SLs in real time, which is difficult to ensure under fast-changing conditions;
2) Second, computational overhead: solving multi-period, nonlinear formulations in large networks is time-consuming and often incompatible with online decision-making;
3) Third, poor integration with dispatch: although SLs can be forecasted, most studies treat SL identification and scheduling as two decoupled stages, missing the opportunity to use SL dynamics for policy refinement.

To incorporate parameter uncertainty into resilience planning, optimization frameworks such as stochastic programming [16], mixed-integer linear programming [17], distributionally robust optimization (DRO) [18], and robust stochastic optimization [19] have been adopted. While these techniques improve robustness by modeling worst-case or probabilistic uncertainty, they typically depend on scenario enumeration or offline uncertainty sets, which are insufficient for handling the continuous and real-time evolution of weather-sensitive constraints.

(2) Scenario-driven scheduling strategies

In addition to physics-based modeling, some researchers have proposed control strategies tailored to specific disaster scenarios, emphasizing operational flexibility and local responsiveness. These strategies typically do not explicitly model system physics but instead rely on heuristic rules, rule-based reconfiguration, or offline simulations to guide actions under predefined contingencies.

For example, [20] and [21] explored intentional islanding and microgrid formation as a means to enhance local autonomy and reduce restoration latency. [22] designed real-time topology reconfiguration schemes for fast fault isolation and load balancing. [23] proposed coordinated scheduling of multi-energy systems, incorporating gas and thermal networks to support electric resilience, while [24] considered the spatiotemporal dispatch of mobile resources such as portable energy storage or diesel generators.

These scenario-driven approaches are valuable for operational planning and coordination. However, most of them:

1) Assume fixed SLs or network limits, regardless of evolving environmental conditions;
2) Depend heavily on static rules or threshold-based control, limiting generalizability across different types and scales of disasters;
3) Do not rely on data-driven adaptation, which is essential when operating under complex and nonlinear disturbances, such as heatwaves, floods, or ice storms.

Therefore, although scenario-driven methods improve operational flexibility, their limited treatment of environmental variability and real-time system feedback renders them insufficient for enhancing dynamic resilience.

(3) Model-free learning-based methods

To overcome the dependence on explicit modeling and scenario enumeration, recent research has increasingly focused on model-free, data-driven learning techniques, with reinforcement learning (RL) being a prominent example. RL formulates decision-making as a Markov decision process (MDP), allowing agents to learn optimal policies via interactions with the environment without requiring a system model [25,26].

Initial RL applications in power systems involved low-level control such as excitation regulation, frequency support, and local voltage control [26]. With advances in neural networks and the increasing availability of big data, RL has been extended to more complex tasks, such as demand response [27], electricity market bidding [28], and post-disaster restoration planning [29,30].

For instance, [30] employed tabular RL to learn restoration policies in small distribution systems; however, scalability was limited due to the high dimensionality of the state-action spaces. To address this, deep reinforcement learning (DRL) emerged as a scalable alternative. Algorithms such as deep Q-network (DQN) [31], proximal policy optimization (PPO) [32], and deep deterministic policy gradient (DDPG) [33] support continuous action spaces and policy approximation using deep neural networks.

Recent work has explored advanced DRL-based resilience strategies. [34] proposed Bayesian RL to handle epistemic uncertainty in multi-energy microgrids. [35] developed a multi-buffer DQN to enable resilient topological reconfiguration while maintaining radiality. In [36], an imitation learning framework addressed stochastic N–k outages with tie-line coordination and reactive power optimization. In [37], an actor-critic DRL approach was employed for three-phase unbalanced distribution systems, improving restoration speed without prior model assumptions.

Although these DRL strategies significantly enhance adaptive decision-making, they generally ignore the weather-dependent evolution of SLs, often assuming static or simplified network constraints throughout the learning process. This decoupling between environmental dynamics and operational limits can lead to dispatch strategies that are either too conservative or potentially infeasible under rapidly changing weather conditions.

In summary, existing literature presents valuable contributions across physical modeling, operational control, and learning-based scheduling. However, the following limitations persist:

1) Model-based and scenario-driven methods lack adaptability and fail to generalize across event types;
2) DRL methods offer promising flexibility but do not consider dynamic SLs shaped by real-time meteorological conditions;
3) To the best of our knowledge, no prior work has developed a unified framework that integrates weather-aware SL identification with DRL-based resilience dispatch.

1.3 Research Gap and Motivation

Although various methods have been developed to enhance distribution system resilience under extreme weather, most existing approaches do not consider the joint effect of real-time weather evolution and dynamic security limits (SLs). In particular, the SLs—such as voltage and power flow thresholds—are known to vary with environmental conditions, but they are often treated as static or approximated by offline estimations during the decision-making process. A comparative review (summarized in Table 1) indicates that:

1) Model-based approaches have attempted to capture weather–SL relationships using

physics-driven or data-driven formulations [8]–[15]. However, these methods generally separate SL estimation from operational scheduling. The estimated SLs are rarely integrated into real-time dispatch models, limiting the benefit of dynamic modeling under rapidly changing conditions.

2) Learning-based scheduling methods, especially those based on reinforcement learning (RL), have demonstrated improved adaptability and reduced reliance on system models [29]–[37]. Nevertheless, most RL-based methods assume fixed SLs throughout training and execution. The state representations used in these models do not reflect the time-varying nature of SLs, leading to potential misalignment between environmental changes and learned policies.
3) To the best of our knowledge, few studies have attempted to integrate real-time SL identification with resilience-oriented dispatch within a unified learning framework. Existing resilience strategies either focus on improving policy adaptability under fixed constraints or estimate dynamic SLs without embedding them into sequential decision processes. This methodological separation limits the ability to exploit environment-dependent SL dynamics to improve dispatch robustness.

In addition, two practical challenges hinder the deployment of dynamic SL-aware strategies. First, dynamic SL modeling may not always be necessary: when environmental changes are mild, static SLs may suffice. Determining when to activate dynamic modeling is therefore essential for computational efficiency. Second, training SL identification models typically requires large datasets from diverse weather conditions, which may not be available for most distribution systems. Existing studies often assume the availability of sufficient training samples [38], whereas real-world extreme weather events are rare and varied.

This study aims to address the above limitations by proposing a unified framework that integrates weather-aware SL identification with deep reinforcement learning-based resilience dispatch. The approach is designed to support adaptive scheduling decisions under varying weather severity, limited training data, and real-time constraints.

Table 1 Comparative Analysis of the Proposed Model against Alternative Resilience-Enhancing Strategies

References	About forecasting			About decision-making	
	Weather uncertainty	Consideration of weather-aware SLs	SLs Identification method based on model-free	Dealing with weather uncertainty based on dynamic scenarios	Weather uncertainty handling methods based on model-free
[8-14]	No	No	-	-	-
[15]	No	Yes	No	-	-
[16-19]	Yes	No	-	No	-
[20-24]	No	No	-	-	-
[25]	Yes	No	-	Yes	No
[26-28]	No	No	-	-	-
[29-37]	Yes	No	-	Yes	Yes
Proposed	Yes	Yes	Yes	Yes	Yes

1.4 Contributions

To address these challenges, this paper proposes a deep reinforcement learning-based resilience enhancement framework that integrates real-time weather-aware SL identification with adaptive scheduling policies. The contributions are summarized as follows:

1) A DRL-based model is formulated, where SLs are treated as weather-dependent parameters embedded in the system state space. The problem is formulated as a Constrained Markov Decision Process (CMDP) to support sequential and robust policy learning under uncertainty.
2) A Bagging-XGBoost ensemble classifier is designed to identify the severity of weather conditions and determine whether dynamic SL modeling is necessary. This improves computational efficiency and avoids unnecessary modeling during mild events.
3) A hybrid CNN–GRU architecture with transfer learning is developed to estimate weather-

aware SLs from historical meteorological data, capturing nonlinear spatiotemporal dependencies with high accuracy even under limited training data.

4) An attention-based meteorological data reconstruction module is constructed to enhance the quality of weather input features, enabling more accurate SL predictions in cases of missing or sparse data.

1.5 Paper Structure

The remainder of the paper is organized as follows. Section 2 presents the mathematical formulation and overall problem structure. Section 3 details the proposed framework for weather-aware SL identification. Section 4 presents simulation results for the improved IEEE-123 feeder and compares the proposed approach with existing methods. Section 5 concludes the paper and outlines future research directions.

2 Problem Definition and Mathematical Formulation

This section introduces a DRL-based resilience-enhancing strategy that optimizes load shedding, adjusts DGs, and reconfigures system lines by distribution system operators (DSOs) under extreme weather. These actions aim to enhance system resilience by developing strategies to mitigate grid-wide overloads.

2.1 Weather-Impacted Resilience Challenges in Power Systems

Fig. 1 presents the overall framework for the regular operation of the distribution network under extreme weather conditions. The framework comprises three main components: the DSO, the distribution network, and the input/output information necessary for regular operation. First, the DSO, as the overall coordinator of the distribution network operation, is responsible for managing and scheduling the network. Second, the distribution network, as defined in this study, includes *n* nodes connected by lines (denoted as *nn*') between adjacent nodes. The network contains load nodes (denoted as *b*), CL nodes (denoted as *d*), and DG nodes (denoted as *g*). These nodes form the following sets: Ω^{bus} , Ω^{line} , Ω^{PL} , Ω^{CL} , Ω^{DG} . The lines comprise both dispatchable and non-dispatchable lines, while the DGs include micro gas turbines, wind power, and photovoltaic power generation. Thus, the set relationships can be represented as $n \in \Omega^{\text{bus}}$, $nn' \in \Omega^{\text{line}} \left(n \in \Omega^{\text{bus}}, n' \in \Omega^{n} \right)$, $b \in \Omega^{\text{PL}} \subset \Omega^{\text{bus}}$, $d \in \Omega^{\text{CL}} \subset \Omega^{\text{PL}}$. Finally, the input information required for the regular operation of the distribution network includes historical and forecast data. The historical data includes weather data, corresponding distribution network data, and SLs, while the forecast data includes weather information and SLs. The output information necessary for regular operation consists of the grid's resilience-enhancing strategy.

Under extreme weather conditions, within the time-space domain of **T**, the DSO dynamically adjusts the SLs and operational strategies based on input information and the current status of the distribution network. This is achieved by scheduling the resources within the network, such as controlling the on/off status of dispatchable lines, adjusting the output of DGs, and managing load shedding, in order to mitigate the adverse impacts of extreme weather.

Therefore, this study aims to examine how to efficiently schedule all internal resources of the distribution network in real-time, taking into account the dynamically changing SLs in response to weather conditions, with the objective of minimizing the negative consequences of strategy failure during extreme weather events. The related decisions in this paper are as follows:

(1) At time $t \in \mathrm{T}$, the output power of the DGs is $P_{t,g}^{\text{DG}}$;

(2) At time $t \in \mathrm{T}$, the power consumption of the load is $P_{t,b}^{\text{PL}}$;

(3) At time $t \in \mathrm{T}$, the on/off status of the dispatchable line is $o_{t,nn'}^{\text{line}}$, where $o_{t,nn'}^{\text{line}} = 0$ represents the line being disconnected, and $o_{t,nn'}^{\text{line}} = 1$ represents the line being connected.

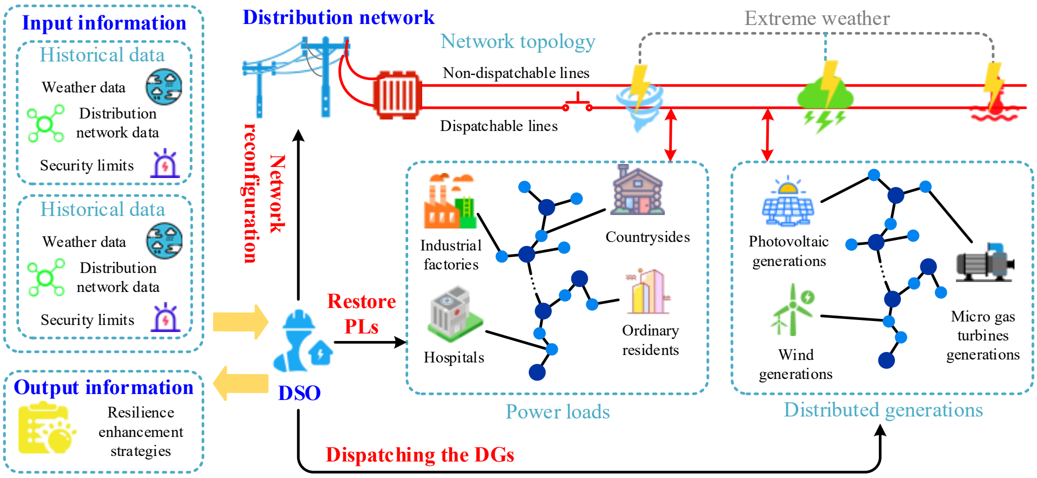


Fig. 1 Overall framework for the operation of the distribution network under extreme weather conditions

2.2 Constrained Markov Decision Process Framework

Under extreme weather conditions, the distribution system may experience severe disruptions, including damage to network topologies, power levels, DG status, and communication systems. In such harsh situations, accurately developing resilience-enhancing strategies using model-based approaches is challenging or even impossible, especially when data measurements and transmissions in the communication systems are unavailable. Hence, this paper adopts a model-free approach to enhance grid resilience to ensure the effective implementation of resilience-enhancing strategies.

The training process for resilience-enhancing strategies can be viewed as a Constrained MDP (i.e., CMDP), which can be expressed as:

$$\mathrm{CMDP}=\left\{\mathbf{S},\mathbf{A},p\left(s_{t+1}\middle|s_t,a_t\right),\mathbf{R},\mathbf{C}\middle|t,\tau\in\mathbf{T},s\in\mathbf{S},a\in\mathbf{A}\right\}\tag{1}$$

The environment f_{DSO} includes weather conditions, grid topologies, power loads (PLs), DG outputs, and distribution system SLs. It is represented explicitly as:

(1) State Space: The operating state of the distribution system is regarded as a Markov state, typically comprising the DGs state, the PLs state, the line on-off state, the voltage security limit (VSL), and the power flow security limit (PFSL) state. Nomenclature is provided in Appendix A.

$$s_t=\left\{\begin{array}{l} P^{\mathrm{DG}}_{t,g},P^{\mathrm{PL}}_{t,b},o^{\mathrm{line}}_{t,nn'},\tilde{U}^{\max}_{t,n},\tilde{P}^{\max}_{t,n}\,\Big| \\ t\in\mathbf{T},g\in\Omega^{\mathrm{DG}},b\in\Omega^{\mathrm{PL}},nn'\in\Omega^{\mathrm{line}},n\in\Omega^{\mathrm{node}}\end{array}\right\}\tag{2}$$

(2) Action Space: Control actions encompass the dispatching of DGs, PL shedding, and reconfiguring lines, represented by combinations of $P^{\mathrm{DG}}_{t,g}$, $P^{\mathrm{PL}}_{t,b}$ and $o^{\mathrm{line}}_{t,nn'}$.

(3) State Transition: After the action a_t is determined, the DSOs interact with the environment to obtain the next state s_{t+1} , i.e. $s_{t+1}=f_{\mathrm{DSO}}\left(s_t,a_t\right)$. Due to the different environments, the transition from s_t to s_{t+1} has uncertainty, i.e., there is a state transition probability $p\left(s_{t+1}\middle|s_t,a_t\right)$.

(4) Reward Space: Upon taking an action a_t and generating a new state s_{t+1} , the environment provides a reward for taking that particular action. The reward function can be represented as a normalized (Equation (4)) reward for restoring load, as shown in Equation (3). The reward for load restoration is defined by the amount of restored load for different levels of importance, as presented in Equation (5). The agent's goal is to train a policy, denoted as π , that maps a given state s_t to an action a_t in order to maximize the expected cumulative reward, as shown in Equation(6) . The relevant quantities in these equations are obtained through observations from the actual system:

$$r_t=\hat{r}^{\mathrm{CL}}_t\tag{3}$$

$$\hat{r}_t=\frac{r_t-\min\left(r_t\right)}{\max\left(r_t\right)-\min\left(r_t\right)}\tag{4}$$

$$r_t^{\mathrm{CL}} = \sum\nolimits_{t\in\mathbf{T},d\in\Omega^{\mathrm{CL}}} \varpi_d^{\mathrm{CL}} o_{t,d}^{\mathrm{CL}} P_{t,d}^{\mathrm{CL}} \tag{5}$$

$$\max_{\pi} J(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum\nolimits_{t=\tau}^{T-\tau+1} \gamma^{t-1} r_t\right] \tag{6}$$

(5) Constraint Space: The cost function of the CMDP is defined as in Equation (7). The cost function for violating power balance considers both the tolerance margin ξ, allowing the system to self-regulate within a specified range without immediately resorting to load shedding, thus ensuring the stability of the system's frequency, as shown in Equation (8). The cost function for violating voltage constraints is represented by the sum of the severities of the violations at the two most critical nodes among all over-limit nodes, as indicated in Equation (9). This choice of the two most severely over-limit nodes highlights the impact of high-risk nodes while preventing excessive interference of cost calculation by outliers from a single node. The cost function for violating flow constraints is represented by the severity of the violation at the most critical node among all over-limit nodes, as shown in Equation (10). This selection of the single most severely over-limit node enables priority triggering of dispatch adjustments for overloaded lines and reduces the risk of system collapse. In contrast, the cost function that satisfies the constraints is given in Equation (11).

$$c_t = c_{1,t} + c_{2,t} + c_{3,t} \tag{7}$$

$$c_{1,t} = \sum\nolimits_{t\in\mathbf{T},g\in\Omega^{\mathrm{DG}},b\in\Omega^{\mathrm{PL}}} \left| P_{t,g}^{\mathrm{DG}} - P_{t,b}^{\mathrm{PL}} - \xi \right| \tag{8}$$

$$c_{2,t} = \sum\nolimits_{t\in\mathbf{T},n\in\Omega^{\mathrm{VV}}} \begin{pmatrix} \max\left[0,\left(\tilde{U}_{t,n}^{\max} - U_{t,n}\right)\right] + \\ \max\left[0,\left(U_{t,n} - U_{n}^{\min}\right)\right] \end{pmatrix} \tag{9}$$

$$c_{3,t} = \sum\nolimits_{t\in\mathbf{T},nn'\in\Omega^{\mathrm{VP}}} \max\left[0,\left(\tilde{P}_{t,nn'}^{\max} - P_{t,nn'}^{\mathrm{line}}\right)\right] \tag{10}$$

$$J_{c_f}(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum\nolimits_{t=\tau}^{T-\tau+1} \gamma_c^{t-1} c_{f,t}\right] \le d_f, \forall f \in \{1,2,3\} \tag{11}$$

where d_1 denotes the constraint threshold for violating the power balance, d_2 denotes the constraint threshold for violating voltage limits, and d_3 denotes the constraint threshold for violating flow limits. γ and $\gamma_c \in [0,1]$ is the discount factor.

The mainstream methods for solving CMDP are divided into constraint policy optimization and primal-dual optimization. Although the constrained policy optimization algorithm possesses constraint-handling capabilities, it relies on complex action discretization operations in continuous action spaces. This process increases computational complexity and undermines the efficiency of real-time scheduling responses. In contrast, the primal-dual optimization, designed based on the Actor-Critic architecture, features a core advantage: its Actor network directly outputs continuous action values without requiring additional discretization steps. This characteristic enables it to accurately match the continuous adjustment requirements of scheduling variables, thereby fundamentally avoiding policy biases introduced by action discretization while ensuring the real-time performance and accuracy of decision-making. Given that the system resilience-enhancing process involves continuous decision-making and action space, this paper selects DDPG [39]. Additionally, to transform the inequality constraint problem into an unconstrained problem, a Lagrange multiplier λ is introduced, and the problem (as shown in Equations (6) and (11)) is reformulated as an unconstrained min-max optimization as shown in Equation (12). This approach is referred to as the primal-dual DDPG algorithm [40] (namely, PD-DDPG).

$$\min_{\lambda_f>0}\max_{\pi} \mathcal{L}\left(\pi,\lambda_f\right) \tag{12}$$

$$\mathcal{L}\left(\pi,\lambda_f\right) = J(\pi) - \sum\nolimits_{f=1}^{3} \lambda_f \left(J_{c_f}(\pi) - d_f\right) \tag{13}$$

The PD-DDPG algorithm typically comprises two networks, namely, the actor network and the critic network, which can be further subdivided into six deep neural networks: the main Actor network Θ_μ, the reward Critic network Θ_r, the cost Critic network Θ_c, the target Actor network $\Theta_\mu^{'}$, the target reward Critic network $\Theta_r^{'}$, and the target cost Critic network $\Theta_c^{'}$. The primary function of the actor network is to output an action that maximizes the Q-value by learning the optimal policy for decision-making. Meanwhile, the critic network evaluates the Q-value of an action within a particular state. The objective functions for the reward and cost Critic networks are as follows:

$$\begin{cases} L_r\left(\Theta_r\right) = \dfrac{1}{O}\sum_o\left[y_o - Q_r\left(s_o, a_o; \Theta_r\right)\right]^2 \\ y_o = r_o + \gamma Q_r^{'}\left(s_o^{'}, \pi^{'}\left(s_o^{'}; \Theta_\mu^{'}\right); \Theta_r^{'}\right) \end{cases} \tag{14}$$

$$\begin{cases} L_c\left(\Theta_c\right) = \dfrac{1}{O}\sum_o\left[z_o - Q_c\left(s_o, a_o; \Theta_c\right)\right]^2 \\ z_o = c_o + \gamma_c Q_c^{'}\left(s_o^{'}, \pi^{'}\left(s_o^{'}; \Theta_\mu^{'}\right); \Theta_c^{'}\right) \end{cases} \tag{15}$$

where r_o and c_o is the reward and cost value.

Additionally, the policy gradient of π is affected by the Q-values of the reward Critic Q-network Q_r and cost Critic Q-network Q_c. The Dual variable λ is updated with the dual gradient.

$$\nabla_{\Theta_\mu}\mathcal{L}\left(\Theta_\mu, \lambda_f\right) = \frac{1}{O}\sum_o \nabla_{\Theta_\mu}\left[Q_r\left(s_o, \pi\left(s_o; \Theta_\mu\right); \Theta_r\right) - \lambda_f Q_c\left(s_o, \pi\left(s_o; \Theta_\mu\right); \Theta_c\right)\right] \tag{16}$$

$$\nabla_\lambda \mathcal{L}\left(\Theta_\mu, \lambda_f\right) = \frac{1}{O}\sum_o\left[Q_c\left(s_o, \pi\left(s_o; \Theta_\mu\right)\right) - d_f\right] \tag{17}$$

During the training process, network parameters are updated via the uniform sampling of a minibatch from the experience replay buffer.

Furthermore, the target networks have the same structure as the training networks. After the actor updates, the parameters of all target networks are slowly tracked by the trained online networks at a soft update rate κ, which can be expressed as:

$$\begin{cases} \Theta_r^{'} \leftarrow \kappa\Theta_r + \left(1-\kappa\right)\Theta_r^{'} \\ \Theta_c^{'} \leftarrow \kappa\Theta_c + \left(1-\kappa\right)\Theta_c^{'} \\ \Theta_\mu^{'} \leftarrow \kappa\Theta_\mu + \left(1-\kappa\right)\Theta_\mu^{'} \end{cases} \tag{18}$$

Considering that the action space encompasses both continuous and discrete actions, the specific training process for PD-DDPG is outlined in Algorithm 1 as given in Appendix B.

Remarks: In equations (2), (9), and (10), $\tilde{\bullet}$ shall be assigned as static or dynamic SLs according to the severity of system damage caused by extreme weather. When weather conditions are extreme (with most weather factors being severe), dynamic SLs are employed to maximize the effectiveness of resilience-enhancing strategies [15]. Conversely, when weather conditions are moderately severe (with a few severe weather factors), dynamic SLs may not significantly enhance effectiveness, and static limits can be utilized to conserve computational resources. The selection of SLs enables an adaptable environment in the DRL-based model, which in turn affects its effectiveness. Therefore, an important task in resilience optimization is to determine whether to select static SLs or dynamic SLs as constraints for the optimization model, based on the severity of the weather. This process involves first recognizing an ensemble weather pattern (Section 3.2) and then determining appropriate SLs for the resilience-enhancing model based on the recognized weather pattern (Section 3.3).

Moreover, accurate recognition of the severity of weather conditions is essential for selecting the appropriate SLs. Since the ensemble weather pattern is represented by a combination of multiple weather factors, and different weather factors influence the SLs in distinct ways [41], it is necessary to reconstruct the meteorological data to accurately identify the weather state used for selecting SLs (Section 3.1).

Therefore, based on the two aforementioned issues, the framework for enhancing resilience while considering weather-aware security constraints is illustrated in Fig. 2. The strategy framework comprises two main parts: the weather-aware SLs identification module and the DRL-based strategies generation module. The SL's identification method is divided into three parts: first, meteorological data are reconstructed using an attention mechanism (Section 3.1); second, the Bagging-XGBoost method is employed for weather pattern recognition (Section 3.2). If the weather is recognized as severe (output=1), the weather-aware SLs determination model is activated; otherwise, the resilience strategies generation method is directly applied. Finally, the hybrid CNN-GRU architecture and KT method are used to decide the dynamic SLs (Section 3.3). The SLs output by the SLs identification module are input into the resilience strategies generation module. Then, the main Actor network selects actions, stores samples in the buffer, and samples minibatches to send to target networks. It updates reward and cost Critic networks via gradients, updates the main Actor network, and performs soft parameter updates, ultimately producing the resilience-enhancing strategies.

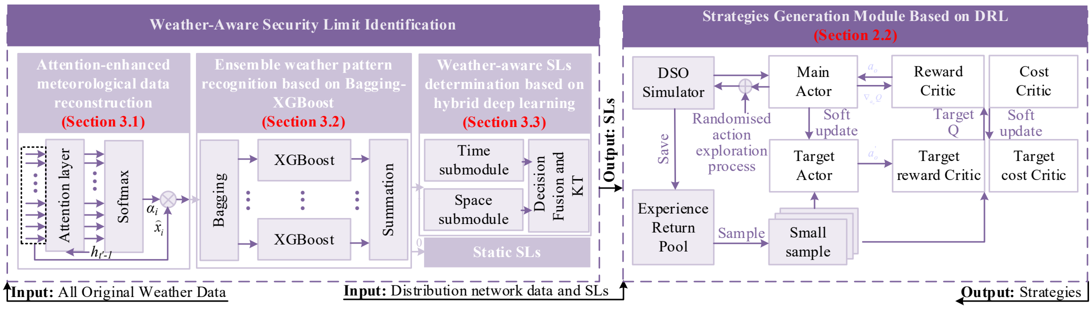


Fig. 2 The framework of resilience enhancement considering weather-aware security constraints

3 Weather-Aware Security Limit Identification Framework

This section proposes a weather-aware SLs identification model based on meteorological data reconstruction and ensemble weather pattern recognition, which consists of three parts: an attention-enhanced meteorological data reconstruction model, an ensemble weather pattern recognition model based on Bagging-XGBoost, and a hybrid deep learning model for dynamic SLs determination.

3.1 Attention-Enhanced Meteorological Data Reconstruction

The attention mechanism is employed to capture the most important features and suppress redundant information, extracting meteorological data that has a significant impact on SLs. This subsection presents the meteorological data for the model described in the preceding section. The model employs an attention mechanism to assign weights to the input data, filtering out less influential weather quantities and focusing on vital weather aspects to avoid distraction.

Let the *i*-th weather sequence be denoted as $\widehat{\mathrm{x}}_i^{\mathrm{w}} = \left(\widehat{x}_{i,1}^{\mathrm{w}}, \widehat{x}_{i,2}^{\mathrm{w}}, \cdots, \widehat{x}_{i,T}^{\mathrm{w}}\right)$, and the attention weight can be expressed as:

$$\varepsilon_t^i = \mathbf{V}_\varepsilon \tanh\left(\mathbf{W}_\varepsilon \mathrm{h}_t + \mathbf{U}_\varepsilon \widehat{\mathrm{x}}_i^{\mathrm{w}}\right) \tag{19}$$

By applying the softmax function to ε_t^i, the sum of attention weights is constrained between 0 and 1:

$$\alpha_{1,t}^i = \frac{\exp\left(\varepsilon_t^i\right)}{\sum_{j=1}^{J} \exp\left(\varepsilon_t^j\right)} \tag{20}$$

By combining the input variables with the attention weights, the final reconstructed weather is given by:

$$\mathbf{x}^{\mathrm{w}} = \left(\alpha_1 \widehat{x}_1^{\mathrm{w}}, \alpha_2 \widehat{x}_2^{\mathrm{w}}, \cdots, \alpha_{\mathrm{I}} \widehat{x}_{\mathrm{I}}^{\mathrm{w}}\right) \tag{21}$$

3.2 Ensemble Weather Pattern Recognition Based on Bagging-XGBoost

In some mildly severe weather conditions, changes in dynamic SLs are negligible and have little impact on resilience-enhancing strategies, whereas in extremely severe weather conditions, changes in dynamic SLs significantly affect these strategies. Therefore, in this subsection, a Bagging-XGBoost-based ensemble model for weather pattern recognition is developed to identify the severity of extreme weather events, thereby improving the efficiency of strategy formulation. It is worth noting that, although this study focuses on ice disaster scenarios, the model's logic for determining SL types (static or dynamic) can implicitly account for the impacts of different extreme weather types. This is because the influence of any extreme weather type on the power system is ultimately reflected through the five key meteorological factors (relative humidity, wind speed, ambient temperature, freezing rain rate, and ice thickness) that serve as core inputs to the Bagging-XGBoost model. The severity assessed by the model integrates the comprehensive effects of these meteorological factors, which in turn encapsulate the system-related risks brought by various extreme weather types. Feeding the reconstructed weather factors from Section 3.1 into the Bagging-XGBoost model thus helps determine the weather severity (a synthesis of weather type impacts and intensity), which enables DSOs to choose the appropriate type of SLs: when the model's output is 0, static SLs are applied for mildly severe weather; when the output is 1, dynamic SLs are applied.

Identifying the severity of extreme weather falls under classification and regression tasks commonly addressed using XGBoost. The implementation process of XGBoost can be described as follows:

(1) Constructing Objective Function: The objective function of XGBoost minimizes the weights assigned to the weather samples, which can be expressed as:

$$\begin{cases} obj = -\sum_{v_1}^{\mathrm{V}_1}\left[\zeta_{v_1}\log\left(\hat{\zeta}_{v_1}^{(q)}\right)+\left(1-\zeta_{v_1}\right)\log\left(1-\hat{\zeta}_{v_1}^{(q)}\right)\right] \\ \qquad +\sum_{v_2=1}^{\mathrm{V}_2}\Omega\left(f_{v_2}\right) \\ \Omega\left(f_{v_2}\right)=\vartheta_1\mathrm{V}_{2,v_3}+0.5\vartheta_2\left\|w\right\|^2 \end{cases} \tag{22}$$

(2) Taylor Expansion of Objective Function: Directly solving Equation (22) requires handling nonlinear expressions. To address this, in each iteration, the objective function undergoes a quadratic Taylor expansion intended to simplify the solution process for complex objective functions. This expansion approximates the original function as a quadratic function of the "difference between the current predicted value and the output of the new tree." Compared to the first-order Taylor expansion, which only retains linear terms, this second-order expansion additionally incorporates the second derivative, enabling more accurate fitting of the curvature changes of the original objective function and reducing approximation errors, which can be expressed as:

$$L\left(\zeta_{v_1},\hat{\zeta}_{v_1}^{(q-1)}+f_q\left(x_{v_1}^{\mathrm{w}}\right)\right)\approx L\left(\zeta_{v_1},\hat{\zeta}_{v_1}^{(q-1)}\right)+h_{v_1}f_q\left(x_{v_1}^{\mathrm{w}}\right)+g_{v_1}f_q^2\left(x_{v_1}^{\mathrm{w}}\right) \tag{23}$$

(3) Greedy Addition of Decision Trees: All possible split points are traversed to reduce the objective function, and the feature and split point with the maximum split gain are selected. The gain can be expressed as:

$$\mathrm{G}^{\mathrm{gain}}=\frac{1}{2}\left[\frac{\left(G^{\mathrm{L}}\right)^2}{H^{\mathrm{L}}+\vartheta_2}+\frac{\left(G^{\mathrm{R}}\right)^2}{H^{\mathrm{R}}+\vartheta_2}-\frac{\left(G^{\mathrm{L}}+G^{\mathrm{R}}\right)G_{\mathrm{L}}^2}{H^{\mathrm{L}}+H^{\mathrm{R}}+\vartheta_2}\right]-\vartheta_1 \tag{24}$$

(4) Logistic Regression: Set a threshold value and compare it with the determination probability of adding up all the trees. When the cumulative value exceeds the threshold, the output is 1; otherwise, the output is 0.

The Bagging algorithm is introduced and combined with the XGBoost, resulting in Bagging-XGBoost to mitigate the risk of overfitting of XGBoost, as shown in Fig. 3. The process of Bagging-XGBoost can be expressed as follows:

(1) Data Sampling: Multiple subsets of data, each of the same size as the original set, are created from the reconstructed weather training set using bootstrapping, known as a "bootstrap sample."

(2) Model Training: The XGBoost model is trained on each subset of data.

(3) Ensemble Weather Pattern Recognition: For each result, the individual sub-model results are combined using a mean voting strategy and compared to a threshold.

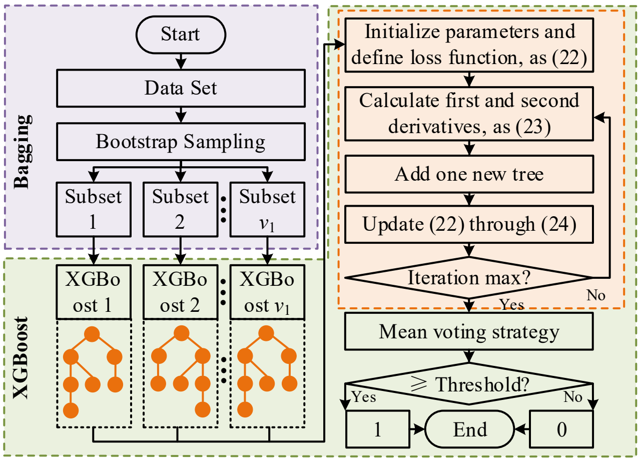


Fig. 3 Schematic diagram of Bagging-XGBoost algorithm

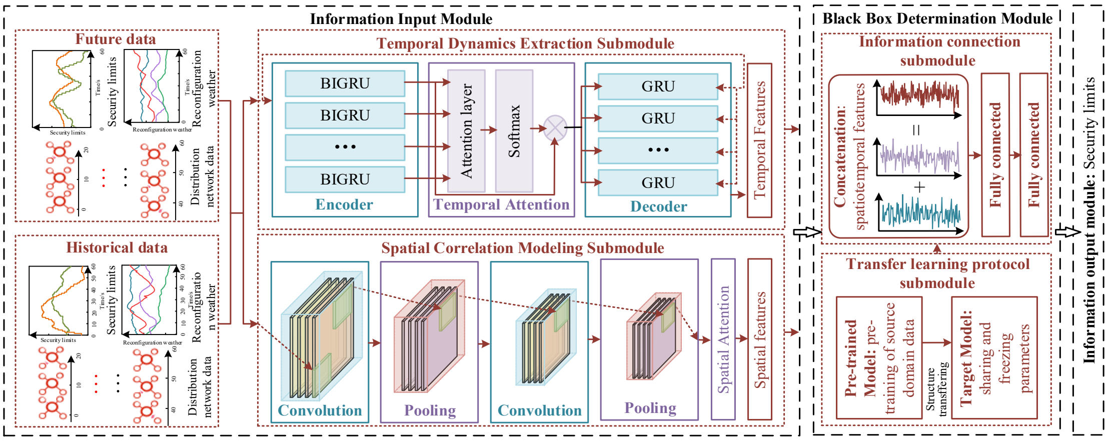
The training process for Bagging-XGBoost is outlined in Algorithm 2 in Appendix B.

3.3 Hybrid Deep Learning for Dynamic Security Limits Determination

This subsection proposes a weather-aware SLs determination model based on a hybrid deep learning, a hybrid CNN-GRU architecture, and a KT algorithm, as depicted in Fig. 4. This model can be abstractly represented as a combination of three modules: information input, black-box determination, and information output. When determining SLs, two critical factors need to be considered: 1) the spatiotemporal features of the extreme weather and SLs input information in the information input module, and 2) the determination accuracy of the black-box determination module when the data density of the input information is sparse (i.e., a system with less extreme weather). The former must be considered because the SLs of nodes and lines at different locations are affected differently by the weather, even in the same weather conditions, due to their varying setup positions (e.g., angles). The latter must be considered because systems are less likely to experience extreme weather and may lack operational data to achieve satisfactory accuracy.

For 1), the model incorporates a temporal dynamics extraction submodule and a spatial correlation modeling submodule into the information input module. These two submodules are referred to together as the hybrid CNN-GRU architecture. The temporal dynamics extraction submodule extracts temporal features using a GRU based on the input historical SLs, reconstructed future weather, and historical weather data. The spatial correlation modeling submodule extracts spatial features based on these data using a CNN.

For 2), the black-box determination module, in addition to applying the information connection submodule, also utilizes KT to determine SLs, thereby enhancing determination accuracy in cases of low data density. The detailed formulation will be presented in the following subsections.



Information Input Module
Future data
Security limits
Reconfiguration weather
Distribution network data
Historical data
Security limits
Reconfiguration weather
Distribution network data
Temporal Dynamics Extraction Submodule
BIGRU
BIGRU
•••
BIGRU
Encoder
Attention layer
Softmax
Temporal Attention
GRU
GRU
•••
GRU
Decoder
Temporal Features
Spatial Correlation Modeling Submodule
Convolution
Pooling
Convolution
Pooling
Spatial Attention
Spatial features
Black Box Determination Module
Information connection submodule
Concatenation: spatiotemporal features
=
+
Fully connected
Fully connected
Transfer learning protocol submodule
Pre-trained Model: pre-training of source domain data
Structure transferring
Target Model: sharing and freezing parameters
Information output module: Security limits

Fig. 4 Weather-aware SLs determination model based on hybrid deep learning

(1) Information Input Module

The information input module contains the temporal dynamics extraction submodule and the spatial correlation modeling submodule.

1) Temporal Dynamics Extraction: GRU and BIGRU with update and reset gates

Given the abrupt nature of extreme weather changes and the real-time requirements for strategy formulation, the algorithm must effectively address these challenges. Since the GRU not only accelerates training but also retains the ability to respond to changes in input sequences rapidly, this subsection constructs a GRU-based temporal dynamics extraction module consisting of a BIGRU-based encoder submodule, a temporal attention submodule, and a GRU-based decoder submodule, as illustrated in Fig. 5. The encoder transforms the input data into hidden states, the temporal attention module generates the context vector by attending to the hidden states, and the decoder decodes the context vector to obtain the temporal features.

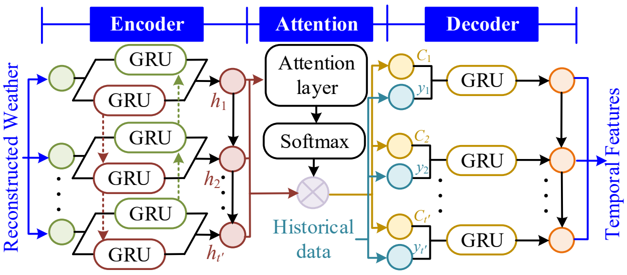


Fig. 5 The structure of the temporal dynamics extraction module

Since the encoder needs to capture all relevant information about the input sequence, BIGRU can simultaneously process the input sequence from both directions, ensuring comprehensive mining and integration of the information. Therefore, the encoder module utilizes BIGRU with the following expression:

$$\mathbf{h}_t = \left[\mathbf{h}_t^{\mathrm{b}};\mathbf{h}_t^{\mathrm{f}}\right] \tag{25}$$

$$\mathbf{h}_t^{\mathrm{f/b}} = \left(1-\mathbf{z}_t^{\mathrm{f/b}}\right)\odot \mathbf{h}_{t-1}^{\mathrm{f/b}} + \mathbf{z}_t^{\mathrm{f/b}} \odot \bar{\mathbf{h}}_t^{\mathrm{f/b}} \tag{26}$$

$$\bar{\mathbf{h}}_t^{\mathrm{f/b}} = \tanh\left(\mathbf{W}_{\tilde{\mathrm{h}}}^{\mathrm{f/b}}\left[\mathbf{r}_t^{\mathrm{f/b}} \odot \mathbf{h}_{t-1}^{\mathrm{f/b}};\mathbf{x}_t\right]+\mathbf{b}_{\tilde{\mathrm{h}}}^{\mathrm{f/b}}\right) \tag{27}$$

$$\mathbf{z}_t^{\mathrm{f/b}} = \sigma\left(\mathbf{W}_{\mathrm{z}}^{\mathrm{f/b}}\left[\mathbf{h}_{t-1}^{\mathrm{f/b}};\mathbf{x}_t\right]+\mathbf{b}_{\mathrm{z}}^{\mathrm{f/b}}\right) \tag{28}$$

$$\mathbf{r}_t^{\mathrm{f/b}} = \sigma\left(\mathbf{W}_{r}^{\mathrm{f/b}}\left[\mathbf{h}_{t-1}^{\mathrm{f/b}};\mathbf{x}_t\right]+\mathbf{b}_{r}^{\mathrm{f/b}}\right) \tag{29}$$

A temporal attention module is essential after the encoder to improve the model's ability to capture information when dealing with long data series, such as weather, which can be expressed as:

$$\mathbf{p}_t^j = \mathbf{V}_p \tanh\left(\mathbf{W}_p\mathbf{s}_{t-1} + \mathbf{U}_p\mathbf{h}_t^j\right), \tag{30}$$

$$\beta_{2,t,j} = \frac{\exp\left(\mathbf{p}_{j,t}\right)}{\sum_{j=1}^{J}\exp\left(\mathbf{p}_{t,j}\right)}, \tag{31}$$

$$\mathbf{c}_t = \sum_{j=1}^{J}\beta_{2,t,j}\mathbf{h}_{t,j} \,. \tag{32}$$

Since the decoder outputs temporal features using the mapping function φ_1, its output depends only on the previously generated information and the contextual information of the temporal attention. Therefore, it is sufficient to use the GRU. The expression for GRU is similar to BIGRU, and GRU only requires the forward propagation process, which is given by (25)-(29).

2) Spatial Correlation Modeling: Graph CNN

Given that local variations in extreme weather have a more direct impact on the power grid and that strategy formulation requires sufficient real-time responsiveness, the algorithm must be capable of effectively addressing these demands. Since CNNs are not only advantageous in processing local information but also efficient enough for this task, this subsection constructs a CNN-based spatial correlation modeling module containing convolutional, pooling layers, and a spatial attention layer. The convolutional submodule processes the input weather data through filter convolution and activation functions, and its output is sent to the pooling layer. The pooling submodule merges the output neuron clusters from the current layer into the next layer of neurons, thereby outputting spatial features with reduced dimensionality. The spatial attention submodule extracts the more important spatial features.

The convolutional submodule can be expressed as:

$$\mathbf{d}_{k,k'}^{l} = \sigma\left(\sum_{m=1}^{M} \mathbf{W}_{m,k'}^{l} \cdot \mathbf{x}_{k+m-1,k'}^{0} + \mathbf{b}_{k'}^{l}\right). \tag{33}$$

The pooling submodule selects the maximum pooling method:

$$\mathbf{MP}_{k,k'}^{l} = \max \mathbf{d}_{k\times e_1+e_2,k'}^{l-1} \,. \tag{34}$$

The spatial attention submodule is similar to the temporal attention module, and the expression can be referred to as Equations (30)-(32). First, calculate spatial attention scores. Take the spatial feature matrix from the CNN pooling layer as input. For each spatial position and each spatial feature dimension, compute a score that reflects the feature's importance for SL estimation. The calculation involves linear transformations of the previous spatial position's hidden state and the current position's feature, then uses the hyperbolic tangent activation function to map the result to a reasonable range. Second, normalize spatial attention weights. A normalized weight closer to 1 indicates that the spatial position has a more significant impact on SL estimation, while a weight closer to 0 indicates less impact. Third, generate spatially weighted feature outputs. Multiply each normalized weight by the original spatial feature of the corresponding position. This amplifies features of high-weight positions and suppresses those of low-weight positions. The resulting features are the sub-module's final output, which is later combined with temporal features to determine dynamic SL.

$$\mathbf{p}_{k}^{i} = \mathbf{V}_{p} \tanh\left(\mathbf{W}_{p}\mathbf{s}_{k-1} + \mathbf{U}_{p}\mathbf{h}_{k}^{i}\right) \tag{35}$$

$$\beta_{3,k,i} = \frac{\exp\left(\mathbf{p}_{i,k}\right)}{\sum_{i=1}^{I} \exp\left(\mathbf{p}_{k,i}\right)} \tag{36}$$

$$\mathbf{c}_{k} = \sum_{i=1}^{I} \beta_{3,k,i} \mathbf{h}_{k,i} \tag{37}$$

(2) Black-Box Determination Module

The black-box determination module contains the information connection submodule and the transfer learning protocol submodule.

1) Information Connection: Hybrid CNN-GRU architecture-based feature aggregation

This subsection constructs the information connection submodule, comprising a concatenation layer and two fully connected layers. The concatenation layer integrates spatial and temporal features into spatiotemporal features. Finally, the fully connected layer maps spatiotemporal features into decided SLs using a mapping function.

The objective function of the black-box determination module is expressed by the mean squared error between the true and the decision-derived SLs and resilience indicators, as follows:

$$L = \frac{1}{2\theta_1} \sum \left| \breve{y}^{SL} - y^{SL} \right|^2 + \frac{1}{2\theta_2} \sum \left| \breve{y}^{str} - y^{str} \right|^2 \,. \tag{38}$$

The SLs determination module, based solely on a hybrid CNN-GRU architecture, employs four evaluation metrics to assess the accuracy of various SLs determination methodologies: root mean square error (RMSE), percent bias (PBIAS), and the coefficient of determination (R^2). Lower RMSE, PBIAS values closer to zero, and R^2 values closer to one indicate superior model performance and vice versa [38]. The workflow of the SLs determination module, based solely on a hybrid CNN-GRU architecture, is provided in Algorithm 3 of Appendix B.

2) Transfer Learning Protocol: Dynamic ensemble strategy-guided knowledge transferring algorithm

In this subsection, the transfer learning protocol, i.e., the KT algorithm, is utilized in the black-box determination process of SLs to address the issue of insufficient training data, such as historical extreme weather data measurements. Areas with frequent extreme weather occurrences are designated as a source domain, while areas with infrequent extreme weather are designated as a target domain. The KT process (Fig. 6) involves four steps:

① Pre-training of source domain data: The model is pre-trained using SLs from lines or nodes frequently impacted by extreme weather, denoted as the pre-trained (PT) model, $\mathbb{Z}$.

② TL based on PT source domain knowledge: The parameters of the PT model are transferred and frozen into the corresponding layer of another determination model for systems with infrequent extreme weather. Subsequently, the parameters of the unfixed layer are fine-tuned via back-propagation. This model is denoted as the TL model, $\mathbb{S}$.

③ Online learning based on knowledge of the target domain: The model is trained using data from a distribution system with infrequent weather extremes and is continuously updated as the number of weather extremes encountered increases. This model is denoted as the transfer learning with fixed weight coefficients (OTL-F), $\mathbb{F}$.

④ Online transfer learning based on dynamic ensemble strategy: An adaptive integration strategy combines transfer and online determination models based on their online performance. The resulting model is denoted as the OTL model, $\mathbb{SF}$, which can be expressed as:

$$\mathbb{SF} = \omega \cdot \mathbb{S} + \upsilon \cdot \mathbb{F} \tag{39}$$

The dynamic update formulas for ω and υ are as:

$$\begin{cases} \omega_{n+1} = \dfrac{\omega_n \eta_1}{\omega_n \eta_1 + \upsilon_n \eta_2} \\ \upsilon_{n+1} = \dfrac{\upsilon_n \eta_2}{\omega_n \eta_1 + \upsilon_n \eta_2} \end{cases}, \tag{40}$$

$$\begin{cases} \eta_1 = \dfrac{L\left(\breve{\mathcal{Y}}_n^{\Omega,\mathbb{F}}, \mathcal{Y}_n^{\Omega}\right)}{L\left(\breve{\mathcal{Y}}_n^{\Omega,\mathbb{F}}, \mathcal{Y}_n^{\Omega}\right)} \\ \eta_2 = \dfrac{L\left(\breve{\mathcal{Y}}_n^{\Omega,\mathbb{S}}, \mathcal{Y}_n^{\Omega}\right)}{L\left(\breve{\mathcal{Y}}_n^{\Omega,\mathbb{S}}, \mathcal{Y}_n^{\Omega}\right)} \end{cases}. \tag{41}$$

where, L is the mean absolute error, and $\mathcal{Y}_n^{\Omega} = \left\{ \mathrm{y}_{a,b}^{\Omega} \middle| a = 1, \cdots, n; b = 1, \cdots, N \right\}$ represents the output labels of the N-th SLs taken from the target domain.

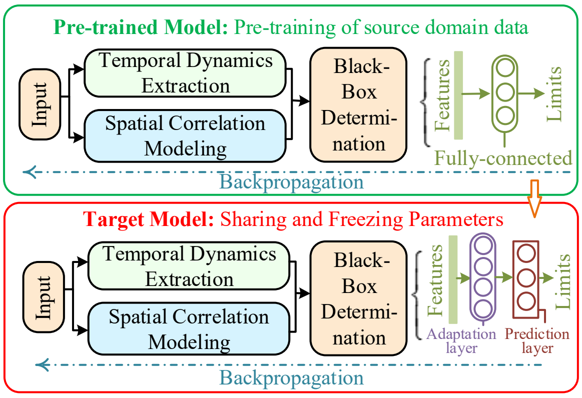

Pre-trained Model: Pre-training of source domain data
Input
Temporal Dynamics Extraction
Spatial Correlation Modeling
Black-Box Determi-nation
Features
Limits
Fully-connected
Backpropagation
Target Model: Sharing and Freezing Parameters
Input
Temporal Dynamics Extraction
Spatial Correlation Modeling
Black-Box Determi-nation
Features
Limits
Adaptation layer
Prediction layer
Backpropagation

Fig. 6 The structure of the KT process

The workflow of the SLs determination module, based on hybrid deep learning, is provided in Algorithm 4 of Appendix B.

4 Case Studies

In this section, the performance of the proposed method is tested in a modified IEEE 123-node distribution system.

4.1 Benchmark Setup

In this paper, the IEEE-123 node system is modified by adding CLs, DGs, and dispatchable lines, resulting in the modified IEEE 123-node system, as illustrated in Fig. 7. The data for the CLs are presented in Table 2, while the data for the DGs are provided in Table 3.

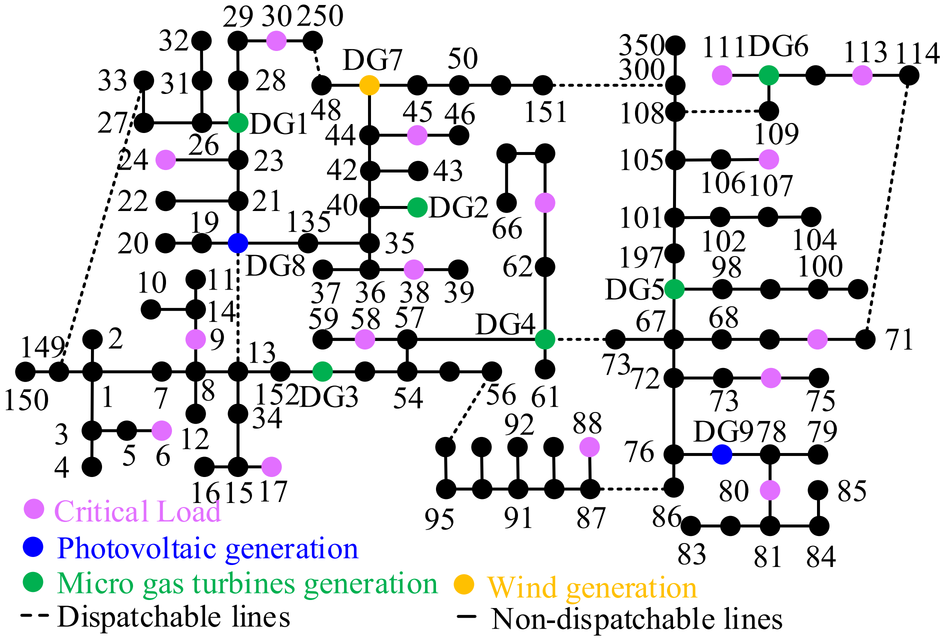


Fig. 7 The modified IEEE-123 node test system

Table 2 CLs parameters

CL	Node	Load(kW)	CL	Node	Load(kW)
1	6	100	9	63	59
2	9	68	10	70	195
3	17	83	11	74	73
4	24	153	12	80	174
5	30	160	13	88	159
6	38	217	14	107	160
7	45	86	15	111	249
8	58	37	16	113	45

Table 3 DGs parameters

DG	Node	Maximum generation(kW)	Minimum generation(kW)
1	25	600	60
2	41	580	58
3	52	500	50
4	60	400	40
5	97	600	60
6	110	370	37
7	47	730	73
8	18	340	34
9	77	390	39

Firstly, the study period is set to be 15:00-18:00, coinciding with an ice storm weather event. DSOs dispatch every 15 minutes, and based on the significance of factors and engineering practicality, the following five weather factors are adopted. The weather data represent the average of the forecasts obtained from the weather station for the corresponding period [42], as shown in Table 4.

Table 4 Reconstructed weather factor

Period	Relative humidity (%)	Wind speed (*m*/*s*)	Ambient temperature (℃)	Freezing rain rate (*mm*/*h*)	Ice thickness (*mm*)
T_1: 15:00-15:15	76	3.1	-7	4.673	0.387
T_2: 15:15-15:30	72	3.2	-9	3.697	0.698
T_3: 15:30-15:45	87	7.9	-11	14.659	2.417
T_4: 15:45-16:00	91	5.1	-13	19.985	4.244
T_5: 16:00-16:15	90	10	-15	18.785	6.772
T_6: 16:15-16:30	88	9.2	-16	16.895	8.937
T_7: 16:30-16:45	83	9.5	-17	13.957	10.800
T_8: 16:45-17:00	87	7.5	-18	15.853	12.589
T_9: 17:00-17:15	78	4.6	-20	10.478	13.541
T_{10}:17:15-17:30	67	2.1	-21	5.866	13.988
T_{11}:17:30-17:45	83	7.2	-22	12.578	15.399
T_{12}:17:45-18:00	95	9.1	-23	23.523	18.300

The network parameters of DRL [44,45] are presented in Table I in Appendix C, while the parameters of the SLs determination model [46] are detailed in Table II in Appendix C.

4.2 Resilience Enhancement Performance

This subsection establishes four approaches for comparative analysis to assess the effectiveness of the proposed resilience-enhancing strategies. The first approach utilizes the model-based approach proposed in [15]. The last three use the proposed model-free approach, the difference being whether meteorological data reconstruction and ensemble weather pattern recognition are taken into account or not. They have the same SL inputs, which are calculated based on the SL determination model proposed in this paper.

- **M1:** Model-based resilience-enhancing strategies [15] considering meteorological data reconstruction and ensemble weather pattern recognition;
- **M2**: Model-free resilience-enhancing strategies considering meteorological data reconstruction without ensemble weather pattern recognition;
- **M3**: Model-free resilience-enhancing strategies considering ensemble weather pattern recognition without meteorological data reconstruction;
- **M4**: Model-free resilience-enhancing strategies considering meteorological data reconstruction and ensemble weather pattern recognition (the method proposed in this paper).

The effectiveness of the model-free approach, the ensemble weather pattern recognition module, and the meteorological data reconstruction module is evaluated in the following sections.

4.2.1 Model-free DRL vs. Model-based method

Table 5 provides a comparison of resilience enhancement strategies under four distinct methods, using the mild disaster period (T_1) and the severe disaster period (T_{12}) as representative cases. Compared to the model-based method (M1), the model-free methods (M2, M3, and M4) not only exhibit notable differences in line scheduling but also demonstrate a substantial increase in the DGs' generation.

Furthermore, Fig. 8(a) presents the percentage of non-exceedance and computation time for the model-free method (M4) and the model-based method (M1). M4 shows a significant increase in the percentage of non-exceedance across most periods when compared to M1, with only a slight decline observed in a few intervals. Simultaneously, the computation time for M4 is consistently lower throughout all periods, with a reduction of 88.77% compared to M1.

These results suggest that, while traditional model-based strategies can enhance resilience, they fall short in both effectiveness and real-time performance relative to the model-free approach. This underscores the necessity and efficacy of adopting model-free strategies.

4.2.2 Ensemble Weather Pattern Recognition Accuracy

Table 5 presents a comparison between the method incorporating ensemble weather pattern recognition (M4) and the method without ensemble weather pattern recognition (M2), revealing no significant differences in terms of both line scheduling and DGs' generation.

Fig. 8 (b) further illustrates the percentage of non-exceedance and computational time for M4 and M2. In the periods corresponding to non-mild ice disasters (T_3-T_{12}), the percentage of non-exceedance remains unchanged mainly between the two methods, with a slight increase of 45.30% in computational time for M4; the average calculation time increases by 14.07 seconds. However, during the mild ice disaster periods (T_1 and T_2), while the percentage of non-exceedance shows no notable difference, the computational time is significantly reduced, with the average calculation time decreasing by 66.30 seconds, representing an 87.95% drop. The reduction in computational time during T_1 and T_2 is notably greater than the increase observed during T_3-T_{12}. It indicates that the ensemble weather pattern recognition module sees an increase in computational time under non-mild ice disasters, while its effect of reducing computational time is more significant under mild ice disasters.

Considering the significantly higher occurrence probability of mild ice disasters in practical scenarios, the reduction in computational time during these events outweighs the minor increase observed during non-mild events. Consequently, although the ensemble weather pattern recognition module marginally increases the computational time for non-mild disasters, it offers a substantial reduction in computational time during mild ice disasters. These findings underscore the necessity and effectiveness of the ensemble weather pattern recognition module.

4.2.3 Meteorological Data Reconstruction Efficacy

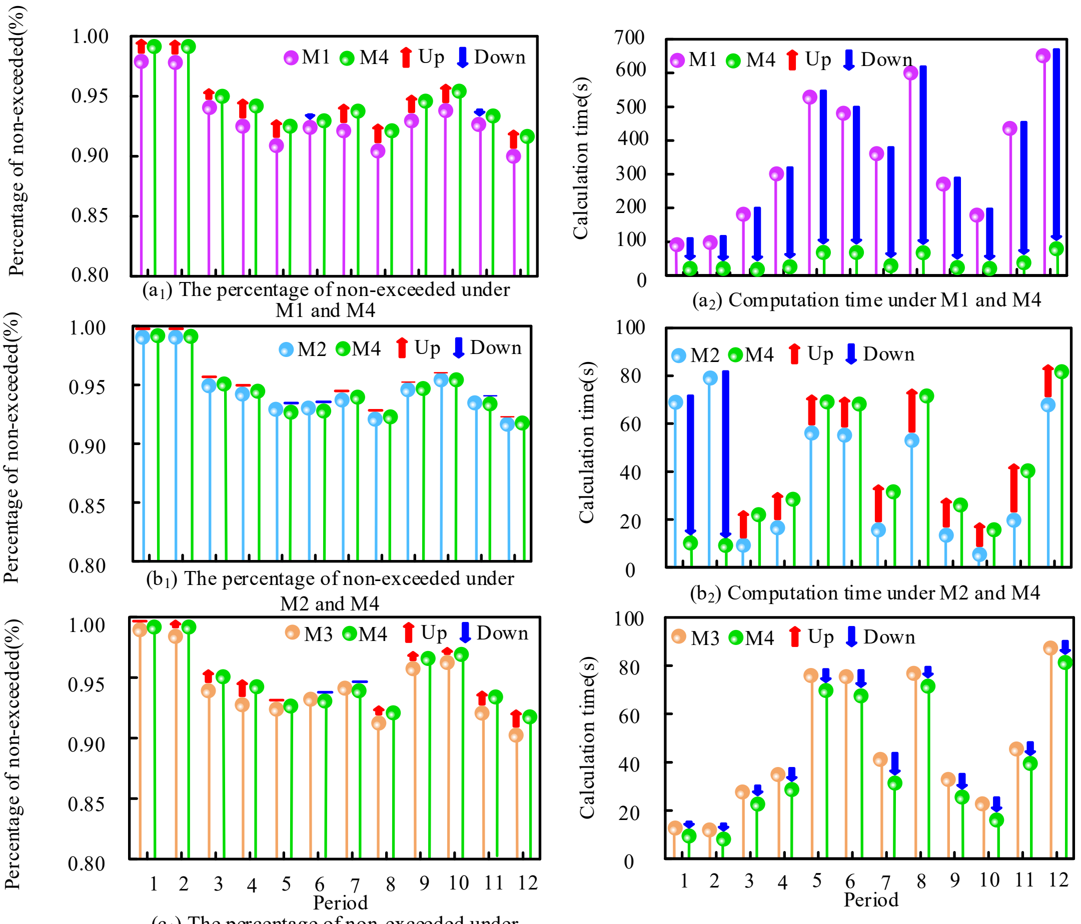
Table 5 demonstrates that the method incorporating meteorological data reconstruction (M4) shows no significant differences in line scheduling compared to the method without meteorological data reconstruction (M3). However, there is a slight but non-negligible increase in DGs' generation under M4.

Furthermore, Fig. 8(c) compares the percentage of non-exceedance and computation time for M4 and M3. M4 shows a slight or negligible increase in the rate of non-exceedance, while its computation time is reduced by 3.67%, corresponding to a decrease of 1.49 seconds compared to M3.

These findings indicate that the meteorological data reconstruction module, while maintaining the effectiveness of the final strategy, contributes to a reduction in computation time. This further validates the necessity and efficacy of incorporating meteorological data reconstruction.

Table 5 Resilience-enhancing strategies under four different approaches at T_1 and T_{12}

Period	Approach	Dispatched lines	The outputs of DGs								
			DG_1	DG_2	DG_3	DG_4	DG_5	DG_6	DG_7	DG_8	DG_9
T_1	M1	56-96	379	319	319	329	457	327	420	287	319
	M2	86-87	389	369	346	378	479	348	520	310	329
	M3	86-87	403	318	320	398	438	369	512	318	324
	M4	86-87	417	348	333	389	466	365	515	326	326
T_{12}	M1	71-114,108-109	186	197	384	187	176	295	175	276	41
	M2	71-114	256	237	442	227	219	335	236	316	62
	M3	71-114	236	231	454	257	246	262	215	346	47
	M4	71-114	265	276	438	259	236	279	253	346	53



(a$_1$) The percentage of non-exceeded under M1 and M4

(a$_2$) Computation time under M1 and M4

(b$_1$) The percentage of non-exceeded under M2 and M4

(b$_2$) Computation time under M2 and M4

(c$_1$) The percentage of non-exceeded under M3 and M4

(c$_2$) Computation time under M3 and M4

Fig. 8 The percentage of non-exceeded and computation time under different models: (a$_1$) and (a$_2$) for M1 and M4; (b$_1$) and (b$_2$) for M2 and M4, and (c$_1$) and (c$_2$) for M3 and M4

4.3 Security Limit Determination Performance

This subsection establishes three approaches for comparative analyses to assess the effectiveness of the proposed SLs determination:

- **M5**: An SL determination method considering the hybrid CNN-GRU architecture without KT;
- **M6**: An SL determination method considering KT without the hybrid CNN-GRU architecture;
- **M7**: An SL determination method considering the hybrid CNN-GRU architecture and KT.

In the 12 periods considered, the first two periods are classified by the ensemble weather pattern recognition module as using static SLs, as verified in Fig. 8. In these periods, the CLs loss, DGs' output, and associated costs are identical across all three methods. Consequently, a detailed analysis of T_1 and T_2 is not provided in this section.

The effectiveness of KT and the hybrid CNN-GRU architecture is evaluated in the following sections.

(1) Knowledge Transferring Impact

Fig. 9 presents the frequency distribution of the VSL and PFSL evaluation indicators under different methods, specifically during periods T_3, T_7, and T_{11}. Compared to M5, M7 demonstrates a reduction in the average voltage RMSE and PBIAS of 0.015 and 0.338, respectively, along with a decrease in current RMSE and PBIAS of 0.135 and 0.344. Additionally, the average voltage R^2 and current R^2 increase by 0.147 V and 0.133 A, respectively. These results indicate that the application of KT significantly enhances the determination accuracy of dynamic SLs.

Furthermore, Fig. 10 shows that during non-light ice disaster periods (T_3-T_{12}). M7 exhibits a significantly lower percentage of critical load (CL) loss and associated costs compared to M5—specifically, the average load loss of M7 is 23.30 % lower than that of M5, with the corresponding costs decreasing by 23.37%—accompanied by higher distributed generation (DG) outputs and costs for M7. This suggests that SLs derived through KT can train more effective strategies, ensuring that a greater proportion of CLs remain unaffected by ice disasters.

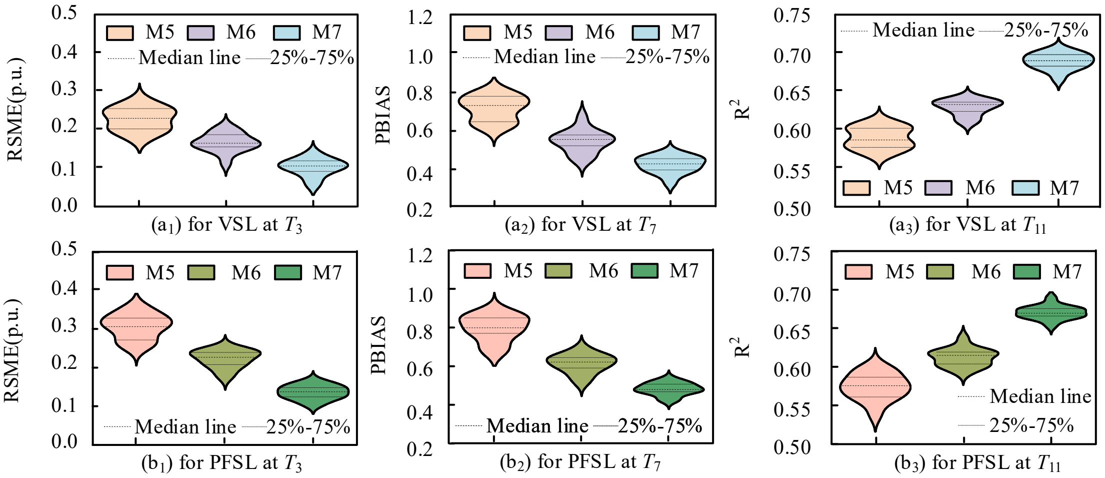
Collectively, these findings confirm the effectiveness of KT in determining SL and formulating strategies.

(2) Study of Hybrid CNN-GRU Architecture

In Fig. 9, a comparison between M6 and M7 reveals that M7 achieves reductions in average voltage RMSE and PBIAS, as well as power flow RMSE and PBIAS, by 0.007, 0.181, 0.073, and 0.194, respectively. Furthermore, average voltage R^2 and power flow R^2 increase by 0.084 and 0.070, respectively. These improvements indicate that the hybrid CNN-GRU architecture enhances the determination accuracy of dynamic SLs by effectively extracting both temporal and spatial features.

As illustrated in Fig. 10, during periods T_3-T_{12}, M7 demonstrates a significantly lower percentage of critical load (CL) loss and associated costs compared to M6—specifically, the average load loss of M7 is 12.10% lower than that of M6, with the corresponding costs decreasing by 12.11%—while also achieving relatively higher distributed generation (DG) outputs and costs. This suggests that integrating the hybrid CNN-GRU architecture further mitigates the adverse impacts of ice storms on the power grid.

Together, these results substantiate the effectiveness of the hybrid CNN-GRU architecture in enhancing SLs determination and strategies development.



(a$_1$) for VSL at T_3 (a$_2$) for VSL at T_7 (a$_3$) for VSL at T_{11}

(b$_1$) for PFSL at T_3 (b$_2$) for PFSL at T_7 (b$_3$) for PFSL at T_{11}

Fig. 9 The frequency distribution of evaluation indicators under different approaches: (a) for VSL and (b) for PFSL: (•$_1$) at T_3; (•$_2$) at T_7, and (•$_3$) at T_{11}

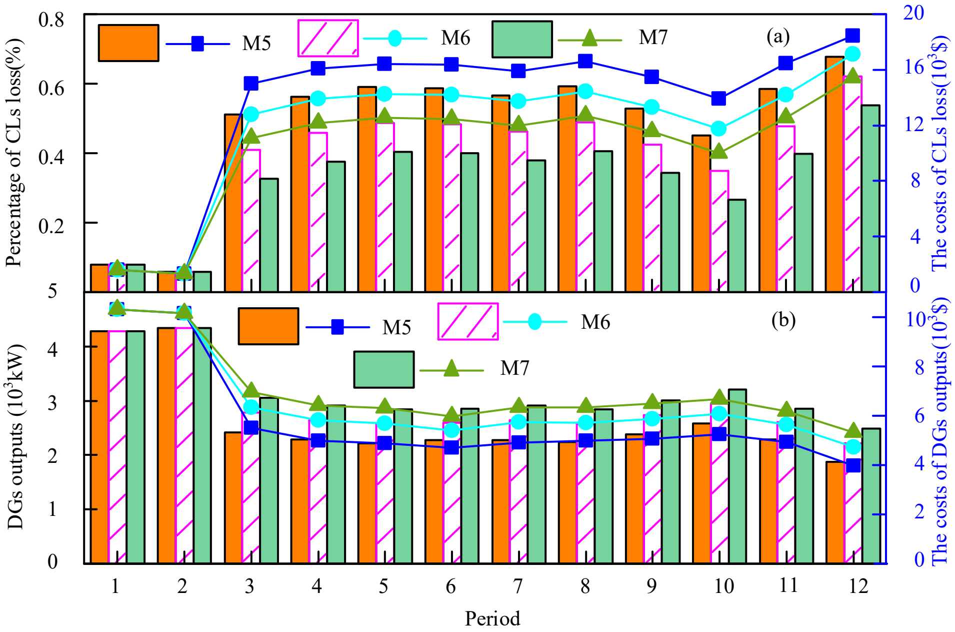


Fig. 10 The percentage of CLs loss, DGs outputs, and their costs

4.4 Sensitivity Analysis

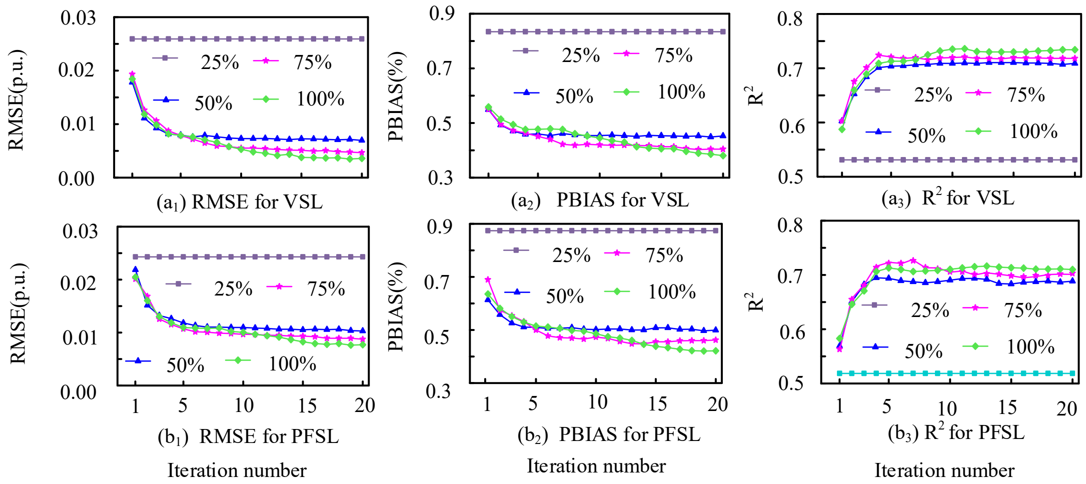
Many factors can directly or indirectly affect the accuracy of SL determinations. Therefore, evaluations are conducted to assess the impact of specific factors on the proposed method through three illustrative cases.

(1) Knowledge Transferring Completeness

In Section 3.3, the proposed KT contains four models, each contributing to different degrees of completeness in the KT model. The completeness of the PT model alone is 25%. Combining the PT and TL models yields 50%, and including the PT, TL, and OTL-F models achieves 75%. Incorporating all four models results in 100% completeness. For the 75% model, both ω and υ are set to 0.5 while ω and υ are adjusted to 0.8 and 0.2, respectively, for the 100% model.

Fig. 11 illustrates the metrics for four different completeness levels of the KT models. Initially, at 25% completeness, the metrics remain static since the PT model is no longer involved in online training after offline training. As completeness increases to 50%, all metrics outperform those of the 25% model, gradually improving as training iterations proceed. This improvement is attributed to the prior knowledge from the source domain and the ability to fine-tune the PT model parameters in the TL model. However, after reaching a certain iteration threshold, the metrics stabilize, possibly due to disparities in target domain knowledge, limiting further model enhancements.

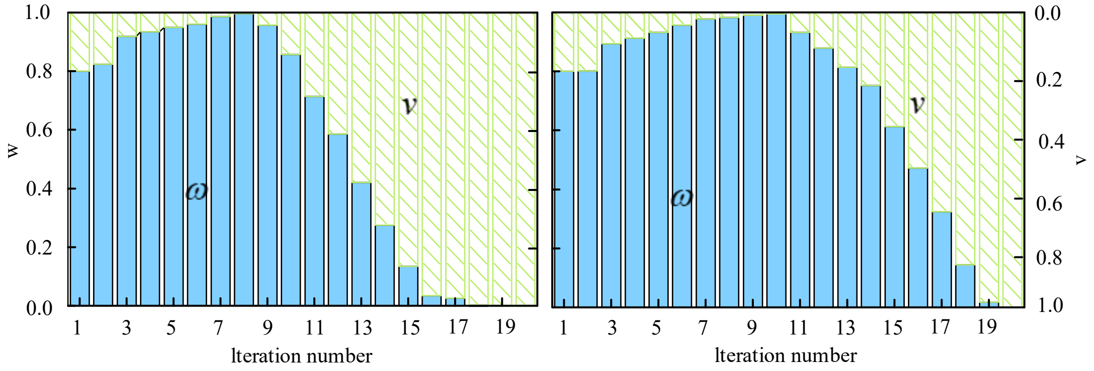
Meanwhile, at 75% completeness, accuracy surpasses that of the 25% and 50% models with a sufficiently high number of iterations, owing to the accelerated convergence facilitated by online learning. Ultimately, at 100% completeness, optimal accuracy is achieved through numerous online iterations. ω and υ for the 100% model are depicted in Fig. 12. Initially, as the number of iterations is small, ω increases while υ decreases due to the superior performance of the 50% model. Consequently, SL determination accuracy steadily improves as the KT model achieves completeness. This once again validates the KT model.



(a₁) RMSE for VSL (a₂) PBIAS for VSL (a₃) R^2 for VSL

(b₁) RMSE for PFSL (b₂) PBIAS for PFSL (b₃) R^2 for PFSL

Fig. 11 The three metrics under different completeness KT models: (a) for VSL; (b) for PFSL; (•$_1$) for RMSE; (•$_2$) for PBIAS, and (•$_3$) for R^2



(a) for VSL (b) for PFSL

Fig. 12 Weights of KT of 100% model: (a) for VSL and (b) for PFSL

(2) Meteorological Variability

The meteorological elements are reconstructed according to Section 3 to establish various weather scenarios, where the severity of the disasters increases progressively from M8 to M10.

- **M8**: Humidity, wind speed, freezing rain rate, and ice thickness are all at low levels, and the temperature is -10°C;
- **M9**: Humidity, wind speed, freezing rain rate, and ice thickness are all at moderate levels, and the temperature is -15°C;
- **M10**: Humidity, wind speed, freezing rain rate, and ice cover thickness are all at high levels, and the temperature is -20°C.

Fig. 13 presents the frequency distribution of PBIAS and R^2 metrics for different scenarios. Observing the median lines in the figure, it can be seen that as the severity of the ice disaster increases, the median and overall distribution of the voltage and power flow evaluation metrics approach optimal values (PBIAS=0, R^2=1). For voltage, the PBIAS reduces from 0.44 in M8 to 0.40 in M10, while the R^2 increases from 0.73 in M8 to 0.75 in M10. For power flow, the PBIAS decreases from 0.47 in M8 to 0.43 in M10, and the R^2 rises from 0.72 in M8 to 0.74 in M10, indicating a gradual improvement in determination accuracy. The reason behind this lies in the more pronounced characteristics of extreme weather as disaster severity intensifies. This phenomenon enables more accurate reconstruction of meteorological data and more reliable ensemble weather pattern recognition, which in turn makes the identified SLs closer to actual values. Therefore, it can be inferred that SL determinations are more accurate under highly severe ice disasters than under minor ice disasters.

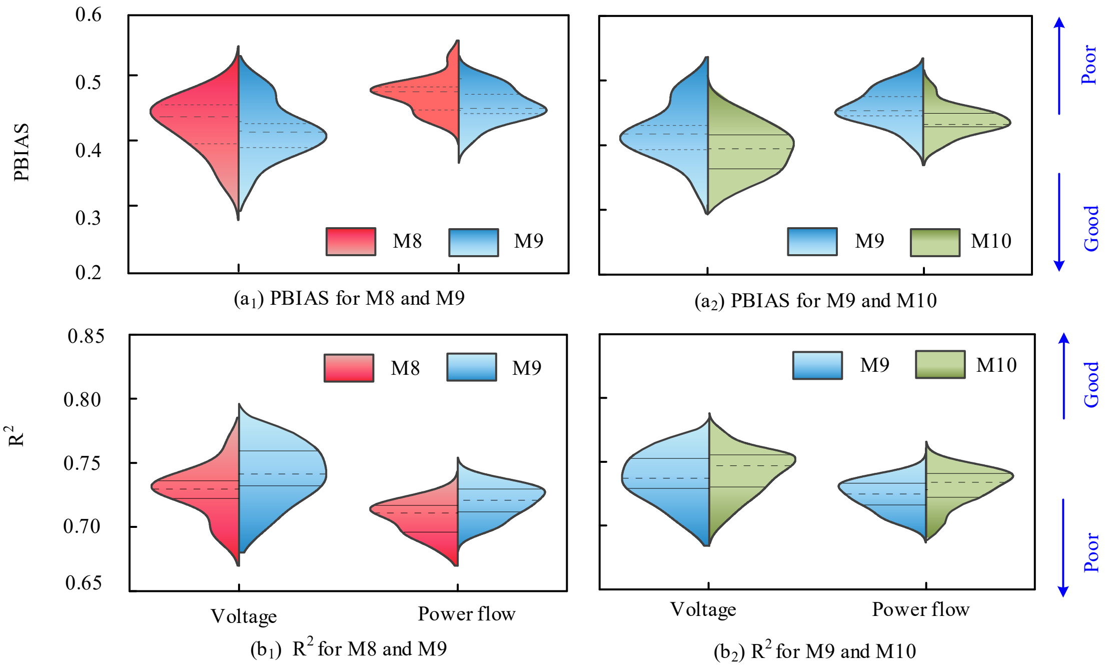


Fig. 13 The frequency distribution for different scenarios: (a) for PBIAS; (b) for R^2; (•$_1$) for M8 and M9; (•$_2$) for M9 and M10

(3) Weather Forecast Accuracy

This subsection examines ten levels of weather forecast accuracy, ranging from 80% to 98%, using the ice disaster data from T12 as input. Fig. 14 illustrates the average cumulative rewards of the resilience-enhancing strategies for different weather forecast accuracies. As the accuracy of the weather forecast decreases, the number of iterations gradually increases. When the number of iterations is too large, there is a high probability that it will hinder the timely execution of resilience-enhancing strategies in real-time during disasters. Fig. 15 presents the values of the three metrics for various weather forecast accuracies, using ten nodes and ten lines as examples. There is no significant discrepancy among the metrics when the accuracy ranges from 96% to 98%. However, when the determination accuracies fall between 80% and 94%, RMSE and PBIAS show significant increases, while R^2 notably decreases. This trend indicates a gradual decrease in the decision-making accuracy of SLs, potentially rendering resilience-enhancing strategies ineffective. Consequently, the optimal weather forecast accuracy interval for this study is 96%-98%.

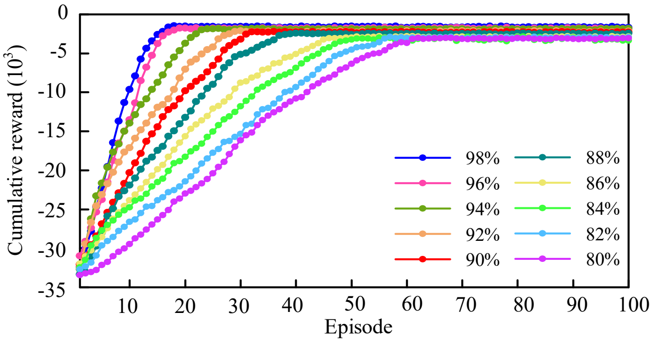
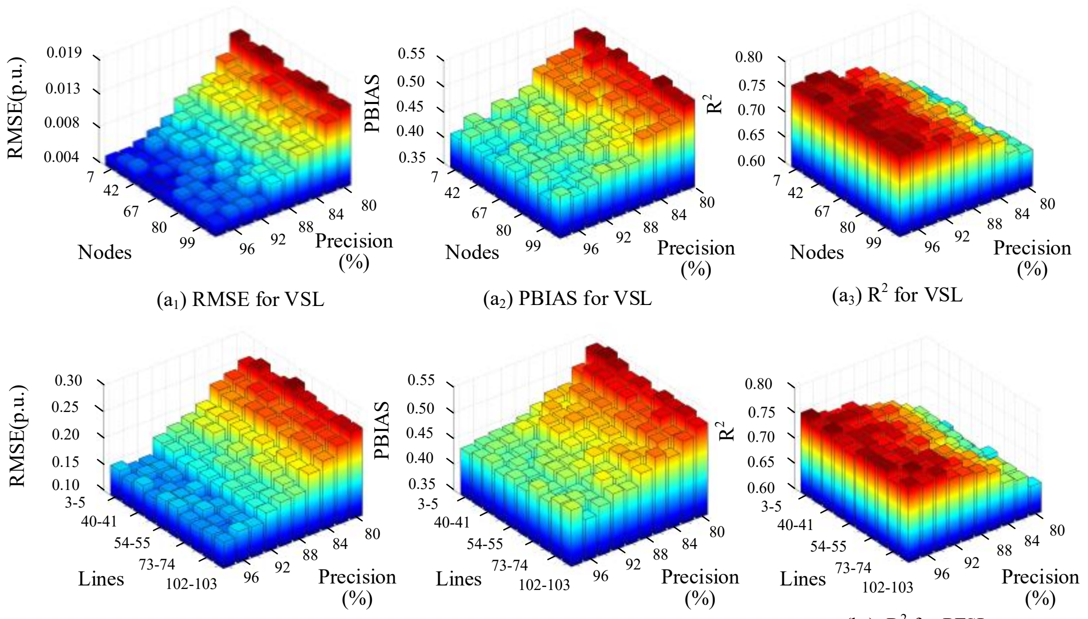


Fig. 14 The average cumulative rewards of the resilience-enhancing strategies for different weather forecast accuracies



(a_1) RMSE for VSL (a_2) PBIAS for VSL (a_3) R^2 for VSL

(b_1) RMSE for PFSL (b_2) PBIAS for PFSL (b_3) R^2 for PFSL

Fig. 15 The three metrics for different weather determination accuracies: (a) for VSL; (b) for PFSL; (•$_1$) for RMSE; (•$_2$) for PBIAS, and (•$_3$) for R^2

5 Conclusions

This paper presents an innovative model-free approach to enhance the resilience of distribution systems by incorporating the variability of SLs in response to fluctuating weather conditions. The proposed methodology integrates three primary components: a meteorological data reconstruction model, an ensemble weather pattern recognition model, and a dynamic SLs determination model. Initially, an attention-enhanced meteorological data reconstruction model is introduced to streamline the training process by distilling input weather data. Furthermore, scenarios in which weather-aware SLs are identified, and a Bagging-XGBoost-based ensemble model for weather pattern recognition is proposed to replace static SLs with dynamic counterparts during periods of extreme weather. A dynamic SLs determination model is then developed to address the multidimensional characteristics of input data, recognizing the interaction between spatial and temporal factors through a hybrid CNN-GRU architecture. To further extend the model's applicability to systems less susceptible to extreme weather events, a KT approach is employed to decide dynamic SLs in such contexts.

The effectiveness of the proposed model is validated through case studies on a modified IEEE 123-node distribution test system. The simulation results clearly demonstrate the advantages of the proposed model in terms of computation time and the percentage of non-exceeded SLs compared to alternative approaches that either disregard weather-aware SLs or utilize model-based methods. Moreover, the decision-making performance of the proposed SLs determination method, as evidenced by key metrics and CLs loss ratios, surpasses that of existing techniques.

While the method has been rigorously validated for enhancing resilience and SLs determination under ice disaster conditions, its application to other extreme weather scenarios remains underexplored. Future work will focus on developing models tailored to a broader range of extreme weather conditions and assessing the effectiveness of the proposed method in these diverse contexts.

6 Acknowledgements

This work was supported by the National Natural Science Foundation of China (52477087, 52477083, 52207082) and the Fundamental Research Funds for the Central Universities (B230201048).

References

[1] Zhou D, Pan X, Guo J, Sun X, Wang C, Zheng J. Resilience quantification of offshore wind farm cluster under the joint influence of typhoon and its secondary disasters. Applied Energy 2025;383:125323. https://doi.org/10.1016/j.apenergy.2025.125323.

[2] Hu C, Li Y, Hou Y. Risk-informed resilience planning of transmission systems against ice storms. Applied Energy 2025;392:125801. https://doi.org/10.1016/j.apenergy.2025.125801.

[3] Dehghani NL, Jeddi AB, Shafieezadeh A. Intelligent hurricane resilience enhancement of power distribution systems via deep reinforcement learning. Applied Energy 2021;285:116355. https://doi.org/10.1016/j.apenergy.2020.116355.

[4] Zhou J, Zhang H, Cheng H, Zhang S, Wang Z, Zhang X. Resilience-Oriented Transmission Expansion Planning under Hurricane Impact Considering Vulnerable Line Identification and Hardening. Protection and Control of Modern Power Systems 2025;10:98–113. https://doi.org/10.23919/PCMP.2024.000063.

[5] Li X, Du X, Jiang T, Zhang R, Chen H. Coordinating multi-energy to improve urban integrated energy system resilience against extreme weather events. Applied Energy 2022;309:118455. https://doi.org/10.1016/j.apenergy.2021.118455.

[6] Arab A, Khodaei A, Khator SK, Ding K, Emesih VA, Han Z. Stochastic Pre-hurricane Restoration Planning for Electric Power Systems Infrastructure. IEEE Trans Smart Grid 2015;6:1046–54. https://doi.org/10.1109/TSG.2015.2388736.

[7] Sharifi A, Yamagata Y. Principles and criteria for assessing urban energy resilience: A literature review. Renewable and Sustainable Energy Reviews 2016;60:1654–77. https://doi.org/10.1016/j.rser.2016.03.028.

[8] Bian X, Chen L, Yu D, MacAlpine JMK, Wang L, Guan Z, et al. Influence of aged conductor surface conditions on AC corona-generated audible noise with a corona cage. IEEE Transactions on Dielectrics and Electrical Insulation 2012;19:2037–43. https://doi.org/10.1109/TDEI.2012.6396963.

[9] Yang L, Hao Y, Li L, Zhao Y. Comparison of pollution flashover performance of porcelain long rod, disc type, and composite UHVDC insulators at high altitudes. IEEE Transactions on Dielectrics and Electrical Insulation 2012;19:1053–9. https://doi.org/10.1109/TDEI.2012.6215112.

[10] Yin F, Farzaneh M, Jiang X. Corona investigation of an energized conductor under various weather conditions. IEEE Transactions on Dielectrics and Electrical Insulation 2017;24:462–70. https://doi.org/10.1109/TDEI.2016.006302.

[11] Sun J, Hu Y, Jiang X, Wu Y, Li T, Zhao Y, et al. Discharge development process and flashover performance of snow-accreted insulators under natural environment. International Journal of Electrical Power & Energy Systems 2023;146:108678. https://doi.org/10.1016/j.ijepes.2022.108678.

[12] Murphy S, Niebur D. Solving the Overhead Transmission Conductor Heat Balance Equation Using the Newton Raphson Algorithm. IEEE Transactions on Power Delivery 2021;36:3743–51. https://doi.org/10.1109/TPWRD.2020.3048831.

[13] de Santos H, Sanz-Bobi MÁ. A Cumulative Pollution Index for the Estimation of the Leakage Current on Insulator Strings. IEEE Transactions on Power Delivery 2020;35:2438–46. https://doi.org/10.1109/TPWRD.2020.2968556.

[14] Jiang J-A, Liang Y-T, Chen C-P, Zheng X-Y, Chuang C-L, Wang C-H. On Dispatching Line Ampacities of Power Grids Using Weather-Based Conductor Temperature Forecasts. IEEE Transactions on Smart Grid 2018;9:406–15. https://doi.org/10.1109/TSG.2016.2553964.

[15] Wu Y, Chen X, Chen R, Chen K, Yao W, Mbonyineza R. Resilience-Enhancing Model Considering Weather-Sensitive Constraints. IEEE Transactions on Power Delivery 2024;39:1209–21. https://doi.org/10.1109/TPWRD.2024.3359613.

[16] Saravi VS, Kalantar M, Anvari-Moghaddam A. Resilience-constrained expansion planning of integrated power–gas–heat distribution networks. Applied Energy 2022;323:119315. https://doi.org/10.1016/j.apenergy.2022.119315.

[17] Zhang Q, Wang Z, Ma S, Arif A. Stochastic pre-event preparation for enhancing resilience of distribution systems. Renewable and Sustainable Energy Reviews 2021;152:111636. https://doi.org/10.1016/j.rser.2021.111636.

[18] Babaei S, Jiang R, Zhao C. Distributionally Robust Distribution Network Configuration Under

Random Contingency. IEEE Transactions on Power Systems 2020;35:3332–41. https://doi.org/10.1109/TPWRS.2020.2973596.

[19] Dehghani NL, Shafieezadeh A. Multi-Stage Resilience Management of Smart Power Distribution Systems: A Stochastic Robust Optimization Model. IEEE Transactions on Smart Grid 2022;13:3452–67. https://doi.org/10.1109/TSG.2022.3170533.

[20] Villanueva-Rosario JA, Santos-García F, Aybar-Mejía ME, Mendoza-Araya P, Molina-García A. Coordinated ancillary services, market participation and communication of multi-microgrids: A review. Applied Energy 2022;308:118332. https://doi.org/10.1016/j.apenergy.2021.118332.

[21] Han H, Xu Y, Wu C, Jiang X, Cao S, Zang H, et al. Nash Equilibrium-Based Two-Stage Cooperative Operation Strategy for Multi-Microgrids Considering Uncertainty. Protection and Control of Modern Power Systems 2024;9:42–57. https://doi.org/10.23919/PCMP.2024.000295.

[22] Zhang W, Zhang C, Li J, Zhu L, Cao S, Huang W, et al. Multi-Resource Collaborative Service Restoration of a Distribution Network with Decentralized Hierarchical Droop Control. Protection and Control of Modern Power Systems 2024;9:19–37. https://doi.org/10.23919/PCMP.2023.000530.

[23] Zhang W, Zhang C, Li J, Zhu L, Cao S, Huang W, et al. Multi-Resource Collaborative Service Restoration of a Distribution Network with Decentralized Hierarchical Droop Control. Protection and Control of Modern Power Systems 2024;9:19–37. https://doi.org/10.23919/PCMP.2023.000530.

[24] Zhang X, Dong Z, Huangfu F, Ye Y, Strbac G, Kang C. Strategic dispatch of electric buses for resilience enhancement of urban energy systems. Applied Energy 2024;361:122897. https://doi.org/10.1016/j.apenergy.2024.122897.

[25] Wang C, Ju P, Lei S, Wang Z, Wu F, Hou Y. Markov Decision Process-Based Resilience Enhancement for Distribution Systems: An Approximate Dynamic Programming Approach. IEEE Transactions on Smart Grid 2020;11:2498–510. https://doi.org/10.1109/TSG.2019.2956740.

[26] Sun X, Qiu J. Two-Stage Volt/Var Control in Active Distribution Networks With Multi-Agent Deep Reinforcement Learning Method. IEEE Transactions on Smart Grid 2021;12:2903–12. https://doi.org/10.1109/TSG.2021.3052998.

[27] Wang B, Li Y, Ming W, Wang S. Deep Reinforcement Learning Method for Demand Response Management of Interruptible Load. IEEE Transactions on Smart Grid 2020;11:3146–55. https://doi.org/10.1109/TSG.2020.2967430.

[28] Zhu Z, Chan KW, Bu S, Or SW, Xia S. Analysis of strategic interactions among distributed virtual alliances in electricity and carbon emission auction markets using risk-averse multi-agent reinforcement learning. Renewable and Sustainable Energy Reviews 2023;183:113466. https://doi.org/10.1016/j.rser.2023.113466.

[29] Zhao J, Li F, Sun H, Zhang Q, Shuai H. Self-Attention Generative Adversarial Network Enhanced Learning Method for Resilient Defense of Networked Microgrids Against Sequential Events. IEEE Transactions on Power Systems 2023;38:4369–80. https://doi.org/10.1109/TPWRS.2022.3215510.

[30] Zhang X, Knueven B, Zamzam A, Reynolds M, Jones W. Primal-Dual Differentiable Programming for Distribution System Critical Load Restoration. 2023 IEEE Power & Energy Society General Meeting (PESGM), 2023, p. 1–5. https://doi.org/10.1109/PESGM52003.2023.10252454.

[31] Momen H, Jadid S. A novel microgrid formation strategy for resilience enhancement considering energy storage systems based on deep reinforcement learning. Journal of Energy Storage 2024;100:113565. https://doi.org/10.1016/j.est.2024.113565.

[32] Zhou Z, Wu Z, Jin T. Deep reinforcement learning framework for resilience enhancement of distribution systems under extreme weather events. International Journal of Electrical Power & Energy Systems 2021;128:106676. https://doi.org/10.1016/j.ijepes.2020.106676.

[33] Jiang C, Lin Z, Liu C, Chen F, Shao Z. MADDPG-Based Active Distribution Network Dynamic Reconfiguration with Renewable Energy. Protection and Control of Modern Power Systems 2024;9:143–55. https://doi.org/10.23919/PCMP.2023.000283.

[34] Zhang T, Sun M, Qiu D, Zhang X, Strbac G, Kang C. A Bayesian Deep Reinforcement Learning-Based Resilient Control for Multi-Energy Micro-Gird. IEEE Transactions on Power

Systems 2023;38:5057–72. https://doi.org/10.1109/TPWRS.2023.3233992.

[35] Zhao J, Li F, Mukherjee S, Sticht C. Deep Reinforcement Learning-Based Model-Free On-Line Dynamic Multi-Microgrid Formation to Enhance Resilience. IEEE Transactions on Smart Grid 2022;13:2557–67. https://doi.org/10.1109/TSG.2022.3160387.

[36] Zhang Y, Qiu F, Hong T, Wang Z, Li F. Hybrid Imitation Learning for Real-Time Service Restoration in Resilient Distribution Systems. IEEE Transactions on Industrial Informatics 2022;18:2089–99. https://doi.org/10.1109/TII.2021.3078110.

[37] Huang Y, Li G, Chen C, Bian Y, Qian T, Bie Z. Resilient Distribution Networks by Microgrid Formation Using Deep Reinforcement Learning. IEEE Transactions on Smart Grid 2022;13:4918–30. https://doi.org/10.1109/TSG.2022.3179593.

[38] Li Y, Song L, Zhang S, Kraus L, Adcox T, Willardson R, et al. A TCN-Based Hybrid Forecasting Framework for Hours-Ahead Utility-Scale PV Forecasting. IEEE Transactions on Smart Grid 2023;14:4073–85. https://doi.org/10.1109/TSG.2023.3236992.

[39] Li Y, Wang R, Li Y, Zhang M, Long C. Wind power forecasting considering data privacy protection: A federated deep reinforcement learning approach. Applied Energy 2023;329:120291. https://doi.org/10.1016/j.apenergy.2022.120291.

[40] Wang Z, Zhang Y, Zhou H, Li J, Wang D, You X. Performance Analysis and Optimization for Distributed RIS-Assisted mmWave Massive MIMO With Multi-Antenna Users and Hardware Impairments. IEEE Trans Commun 2024;72:4661–76. https://doi.org/10.1109/TCOMM.2024.3376768.

[41] Niu Y, Wang J, Zhang Z, Luo T, Liu J. De-Trend First, Attend Next: A Mid-Term PV forecasting system with attention mechanism and encoder–decoder structure. Applied Energy 2024;353:122169. https://doi.org/10.1016/j.apenergy.2023.122169.

[42] Wu Y, Lin Z, Liu C, Huang T, Chen Y, Ru Y, et al. Resilience enhancement for urban distribution network via risk-based emergency response plan amendment for ice disasters. International Journal of Electrical Power & Energy Systems 2022;141:108183. https://doi.org/10.1016/j.ijepes.2022.108183.

[43] Xie H, Tang L, Zhu H, Cheng X, Bie Z. Robustness assessment and enhancement of deep reinforcement learning-enabled load restoration for distribution systems. Reliability Engineering & System Safety 2023;237:109340. https://doi.org/10.1016/j.ress.2023.109340.

[44] Hua M, Zhang C, Zhang F, Li Z, Yu X, Xu H, et al. Energy management of multi-mode plug-in hybrid electric vehicle using multi-agent deep reinforcement learning. Applied Energy 2023;348:121526. https://doi.org/10.1016/j.apenergy.2023.121526.

[45] Li K, Li Z, Huang C, Ai Q. Online transfer learning-based residential demand response potential forecasting for load aggregator. Applied Energy 2024;358:122631. https://doi.org/10.1016/j.apenergy.2024.122631.

[46] Jia K, Gao Z, Ma R, Chai H, Sun S. An Adaptive Optimization Algorithm in LSTM for SOC Estimation Based on Improved Borges Derivative. IEEE Transactions on Industrial Informatics 2024;20:1907–19. https://doi.org/10.1109/TII.2023.3280340.

Appendix A

Nomenclature		DG	Distributed generation
		PL	Power load
MDP	Markov decision process	CL	Critical load
SL	Security limit	PT	Pre-trained
SC	Security constraint	TL	Transfer learning
VSL/PFSL	Voltage/power flow SL	OTL	Online transfer learning
RL	Reinforcement learning	OTL-F	OTL with fixed weight coefficients
DRL	Deep reinforcement learning	RMSE	Root mean square error
DQN	Deep Q learning	PBIAS	Percent bias
PPO	Proximal policy optimization	***Parameters***	
DDPG	Deep deterministic policy gradient	ξ	A tolerable slack margin
Bagging	Bootstrap aggregating	ϖ	Weights of different reward functions
XGBoost	Extreme gradient boosting	κ	Relevant constants during soft update
GRU	Gated recurrent unit	O	The number of sub-policy sets from the replay buffer
BIGRU	Bi-directional GRU	ρ^{Θ}	The distribution of states generated by the behavioural strategy Θ
CNN	Convolutional neural network	γ , γ_c	Discount factor
KT	Knowledge transferring	ϑ	Regularization factor
DSO	Distribution system operator		
w	Weights of weather in the Bagging-XGBoost model	$\mathbf{c}$	Context vector
G^{gain}	Gain value	$\mathbf{d}$	Output of the convolutional layer
H , h_{v_1}	The first-order derivative of the loss function	$\mathbf{MP}$	Pooling layer matrix
G , g_{v_1}	The second-order derivative of the loss function	$\hat{\mathbf{x}}^{\mathrm{w}}$, $\mathbf{x}^{\mathrm{w}}$	Unreconstructed and reconstructed meteorological data
e_1 , e_2	Step size and pooling operation size	***Variables***	
θ_1 , θ_2	The number of output SLs and strategies data	t , τ	Current moment and time steps
η_1 , η_2	The penalty factor	s , a , r	State, action, and reward
		c	Cost function
$p(\cdot)$	State transition probability	r^{CL}	Incentives to restore CLs
$J(\cdot)$	Performance functions for policies	n , b , d g	All nodes, PL nodes, CL nodes, and DG nodes
$L(\cdot)$	Loss function	P , U	Active power and voltage
$\mathcal{L}(\cdot)$	Lagrange function transformed from a CMDP model	o^{line} , o^{CL}	Line on-off state, where 0 represents off, 1 represents on, and CLs restoration state, where 0 represents unrecovered, 1 represents recovered.
$Q(\cdot)$	Value function	Θ , π	Network parameters, action strategy
$\Omega(\cdot)$	Regularization function	i , j	Number of reconstructed weather factors and decoders
$[\cdot;\cdot]$	Connection of two quantities	ζ , $\hat{\zeta}^{(q)}$	The actual weather probability and the weather probability accrued after q-th iterations
$\sigma(\cdot)$	Activation function	v_1 , v_2 , v_3	Number of samples, trees, and leaf nodes
φ	Mapping function	q	Number of iterations in the Bagging-XGBoost model
Index and set of DGs		k , k'	Spatial location index
$\mathbf{S}$, $\mathbf{A}$, $\mathbf{R}$, $\mathbf{C}$, $\mathbf{T}$	DRL state, action, reward, constraint, and time-space	m , l	Number of filters and convolution layers

Ω^{DG} , Ω^{PL} ,	The set of DGs, PLs, lines, and all nodes	y^{SL} , y^{str}	The value of resilience index and SLs of the output
Ω^{line} , Ω^{node}	The set of lines, all nodes	ϖ , υ	Weights of KT
Ω^{VV} , Ω^{VP}	The set of nodes and lines violating VSL and PFSL		
$\boldsymbol{\varepsilon}$, $\mathbf{p}$	Weight of attention	$\bullet^{\mathrm{min}}$, $\bullet^{\mathrm{max}}$	The minimum and maximum limits of the corresponding quantity $\bullet$
$\boldsymbol{\alpha}$, $\boldsymbol{\beta}$	The sum of attention weights	$\tilde{\bullet}$, $\hat{\bullet}$	Correlation quantity of SLs, normalized quantity
$\mathbf{z}$, $\mathbf{r}$	The set of update and reset gates	$\bullet^{\mathrm{L}}$, $\bullet^{\mathrm{R}}$	The left and right subtrees of the corresponding quantity $\bullet$
$\mathbf{W}$, $\mathbf{b}$	Weights and bias matrices in the SLs determination model	$\bullet^{\mathrm{f}}$, $\bullet^{\mathrm{b}}$	Forward and reverse of correlation quantity $\bullet$
$\mathbf{h}$, $\bar{\mathbf{h}}$	Current and candidate hidden layer matrices in an encoder	$\breve{\bullet}$	The decision value relative to true value $\bullet$
$\mathbf{s}$	Hidden state in a decoder		
$\mathbf{V}$, $\mathbf{W}$, $\mathbf{U}$	Learning parameter matrix		

Appendix B

Algorithm 1: PD-DDPG Algorithm

Input: Historical data (weather data, corresponding distribution network data, and SLs) and forecast data (reconstructed weather data and SLs)

```
Initialize experience replay buffer, initialize network parameters
For each iteration period (episode=1 to maximum iterations), do
    Initialize a random action exploration process and global state
    For each time step $t=1$ to $T$, do
        Main Actor network selects discrete and continuous actions from the action space
        Use a Softmax layer to output the probability distribution of discrete actions
        Add action exploration for continuous actions
        Combine discrete and continuous actions to form the complete action $a_t$
        Main Actor network receives sample $(s_t, a_t, r_t, c_t, s_{t+1})$ and stores it in the experience replay buffer
        Main Actor network randomly samples $\left(s_o, a_o, r_o, c_o, s_o^{'}\right)$ a mini-batch from the experience replay buffer $D$ and sends it to the target Actor, and reward and cost Critic network
        Target Actor network computes the strategy $a_o^{'} = \mu^{'}\left(s_o^{'};\Theta_\mu^{'}\right)$ and sends it to the target reward and cost Critic network
        Target reward and cost Critic network calculates the temporal difference target $y_o = r_o + \gamma Q_r^{'}\left(s_o^{'},\left(s_o^{'};\Theta_\mu^{'}\right);\Theta_r^{'}\right)$, $z_o = c_o + \gamma_c Q_c^{'}\left(s_o^{'},\left(s_o^{'};\Theta_\mu^{'}\right);\Theta_c^{'}\right)$ and sends it to the reward and cost Critic network
        The reward and cost Critic network computes the gradient $\nabla_{\theta_r} L_r(\Theta_r)$ and $\nabla_{\theta_c} L_c(\Theta_c)$ based on Equation (14) and (15), then updates the $\Theta_r$ and $\Theta_c$
        Main Actor network calculates the gradient $\nabla_{\Theta_\mu}\mathcal{L}\left(\pi,\lambda_f\right)$ and $\nabla_\lambda \mathcal{L}\left(\pi,\lambda_f\right)$ based on Equation (16) and (17) to updates $\Theta_\mu$ and $\lambda_f$
        Perform parameter soft update according to $\begin{cases} \Theta_r^{'} \leftarrow \kappa\Theta_r + (1-\kappa)\Theta_r^{'} \\ \Theta_c^{'} \leftarrow \kappa\Theta_c + (1-\kappa)\Theta_c^{'} \\ \Theta_\mu^{'} \leftarrow \kappa\Theta_\mu + (1-\kappa)\Theta_\mu^{'} \end{cases}$
    End
End
```

Output: the resilience-enhancing strategy

Algorithm 2: Bagging-XGBoost Algorithm

Input: Reconstructed meteorological data

```
Initialize parameters
For each iteration cycle, episode1=1 to N, do
    For each iteration cycle, episode2=1 to N, do
        Randomly sample one sample with replay from the reconstructed weather dataset
    End
    Accumulate outputs to obtain a data subset
End
For each iteration cycle, episode1=1 to $V_1$, do
    For each iteration cycle, episode2=1 to the maximum iteration number, do
        For the subset $v_1$, calculate the loss function using equation (22)
        Calculate the first and second derivatives using equation (23)
        Find the optimal splitting point using equation (24)
        Add a tree at this point
        Update the loss function
    End
    Accumulate outputs and calculate the average
    End
If  the mean ≥ threshold
    |   Output: Dynamic limit
or
    |   Output: Static limit
End
```

Algorithm 3： SLs determination module based only on hybrid CNN-GRU architecture

Input: Historical data (reconstructed weather data, corresponding distribution network data, and SLs) and future data (reconstructed weather data and SLs)

```
Initialize parameters for GRU, attention mechanism, and CNN, along with the experience replay buffer
For each training epoch $episode_1$=1 to maximum training value, do
    For each $batch_1$=1 to maximum training batches, do
        Sample a small batch from the experience replay buffer
        Compute the forward reset gate, update gate, candidate hidden state, and current hidden state using equations (29) to (26)
        Compute the backward parameters using equations (29) to (26)
        Concatenate forward and backward hidden states using equation (25)
        Compute the attention score using equation (30)
        Calculate the attention weight using equation (31)
        Compute the weighted features using equation (32)
        Compute the relevant parameters for the decoder using equations (26) to (29)
        Output the temporal features
        Extract spatial features through convolution layer 1 using equation (33)
        Reduce dimensions using pooling layer 1 according to equation (34)
        Extract deeper spatial features through convolution layer 2 using equation (33)
        Further reduce dimensions using pooling layer 2 based on equation (34)
        Flatten the spatial features into a 1D vector
        Compute the weighted spatial features using equations (30) to (32)
        Output the spatial features
        Combine the spatial and temporal features through a concatenation layer and a fully connected layer
        Compute the loss function using equation (38)
        Update the parameters of GRU, attention mechanism, and CNN using the Adam optimizer [46]
    End
End
```

Output: Decision on the SLs

Algorithm 4： SLs determination module based on hybrid deep learning

Input: Historical data (reconstructed weather data, corresponding distribution network data, and SLs) and future data (reconstructed weather data and SLs)

```
Extract sample space $\mathcal{X}_n^\Omega \times \mathcal{Y}_n^\Omega$ from lines or nodes that frequently experience extreme weather events
Initialize the model $\mathbb{Z}$, $\mathbb{S}$, $\mathbb{F}$ and weights $\omega_1$ and $\upsilon_1$
Train the model $\mathbb{Z}$ using Algorithm 3.1 within the sample space
For $n$=1 to the maximum number of samples
    Retrieve the label data, $\mathcal{Y}_n^\Omega = \left\{ \mathrm{x}_{n,b}^\Omega, \mathrm{y}_{n,b}^\Omega \middle| b = 1, \cdots, N \right\}$, of the target line or node for the $n$-th extreme weather event
    Update the sample space
    Retrain the model $\mathbb{F}$ and fine-tune $\mathbb{S}$ using Algorithm 3.1 on the updated sample space
    Calculate the penalty factor based on Equation (41)
    Calculate the weight using Equation (40)
    Obtain the model $\mathbb{SF}$ output using Equation (39)
End
```

Output: Decision on the SLs

Appendix C

All experiments are conducted on a hardware platform equipped with an NVIDIA GTX 1050 graphics card and 16 GB of memory.

Table I The parameters of DRL

Parameter	Value
Layer type	dense
Hidden layer structure	64-128-64
Batch size	128
Number of episodes	100
Optimizer	[0,1] normalization
Actor learning rate	0.0001
Critic learning rate	0.0001
Discount factor	0.75
Stochastic forward passes	50

Table II The parameters of the SLs determination model

Type	Hyper-parameters
Convolutional 1(C1)	Filter:8 Kernel size:3
Pooling 1(P1)	Kernel size:2
Convolutional 2(C2)	Filter:16 Kernel size:3
Pooling 2(P2)	Kernel size:2
Flatten	None
GRU 1(G1)	Cell size:32
GRU 2(G2)	Cell size:16
Flatten	None
Concatenate	None
Self-Attention Mechanism	None
Flatten	None
Fully connected 1(D1)	Neuron number:16
Fully connected 2(D2)	Neuron number:12