

Article

A DRL-Based Load Shedding Strategy Considering Communication Delay for Mitigating Power Grid Cascading Failure

Yongjing Wei ¹, Anqi Tian ¹, Yingjie Jiang ¹, Wenjian Zhang ^{2,*}, Liqiang Ma ^{2,*}, Liang Ma ¹, Chao Sun ¹ and Jian Sun ²

¹ Information & Telecommunications Company, State Grid Shandong Electric Power Company, Jinan 250013, China; weiyongjing2023@163.com (Y.W.); angel_tian1992@163.com (A.T.); 18205318889@126.com (Y.J.); maliang2010sdu@126.com (L.M.); sunchaosdu@163.com (C.S.)

² The School of Information Science and Engineering, Shandong Provincial Key Laboratory of Wireless Communication Technologies, Shandong University, Qingdao 266237, China; sunjian@sdu.edu.cn

* Correspondence: zhangwenjian@mail.sdu.edu.cn (W.Z.); lqma@mail.sdu.edu.cn (L.M.)

Abstract: Successive failures in power transmission lines can cause cascading failures in the power grid, which may eventually affect large parts of the power grid and even cause the power grid system to go down. Collecting and transmitting primary equipment information and issuing load-shedding action commands in the power grid depend on the power communication network. With the help of the power communication network, we can better observe the situation of the power grid in real time and provide a guarantee for the regular working of the power grid. However, the communication network also has the problem of communication delay causing latency in load-shedding action. On the premise of preserving the key physical properties and operational characteristics of the power grid, this paper uses the IEEE 14 and 30 bus systems as examples to establish a direct current (DC) power flow simulation environment. We establish a communication network model based on the power grid topology and the corresponding communication channels. For the problem of cascading failures occurring in the power grid after transmission line failures, a load-shedding strategy using soft actor-critic (SAC) based on deep reinforcement learning (DRL) was developed to effectively mitigate cascading failures in the power grid while considering the impact of communication delay. The corresponding communication delay is obtained by calculating the shortest communication path using the Dijkstra algorithm. The simulation verifies the feasibility and effectiveness of the SAC algorithm to mitigate cascading failures. The trained network can decide on actions and give commands quickly when a specific initial failure is encountered, reducing the scale of cascading failures.

Keywords: power grid; power communication network; DC power flow model; cascading failures; DRL; load shedding strategy; SAC



Citation: Wei, Y.; Tian, A.; Jiang, Y.; Zhang, W.; Ma, L.; Ma, L.; Sun, C.; Sun, J. A DRL-Based Load Shedding Strategy Considering Communication Delay for Mitigating Power Grid Cascading Failure. *Electronics* **2023**, *12*, 3024. <https://doi.org/10.3390/electronics12143024>

Academic Editors: Karar Mahmoud, Amr Mohamed and Bala Venkatesh

Received: 16 June 2023

Revised: 5 July 2023

Accepted: 7 July 2023

Published: 10 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The power grid is a vital infrastructure for modern society, as almost all activities depend on electricity. Power grid systems must be robust enough to cope with failures that occur internally and disruptive events that arise externally. In addition, power communication network failures and human misoperation may also lead to the propagation of initial failures [1,2]. This phenomenon of continuous and uncontrolled successive failures of power grid components is known as cascading failures [3].

An important cause of major blackout accidents in smart grids is cascading failure. Severe power system cascading failures can lead to the collapse of the entire power network, triggering large-scale power outages and bringing great economic losses to society. In addition to economic losses, grid disintegration and large blackout accidents will also affect people's normal lives and endanger public safety, causing a serious social impact.

With the continuous development of the power system and the formation of the power grid, operation of the power grid is getting closer to its limit level. Thus, it is very important to study and analyze power system cascading failures, and it is also necessary to study how to mitigate a power grid's cascading failures when they occur. Mitigating cascading failures can not only effectively reduce economic losses but also avoid the inconvenience of large-scale power outages for the people.

The operation of power grid systems is faced with various security threats, and it is urgent to build a green and resilient power grid system. In recent years, people's research on new energy in the power grid has become more and more in-depth. Improving the power generation efficiency of new energy sources, such as wind and solar energy, is also one of the research hotspots [4,5].

In 2003, initial failures of transmission lines and generator sets led to massive power outages in the United States and Canada [6], costing up to USD 30 billion in economic losses. In 2005, a widespread blackout in the Moscow metropolitan area shut down all 321 substations in the Moscow power grid, paralyzing nearly half of the city's industrial production and transportation and causing economic losses of up to USD 1 billion [7]. In 2012, India experienced a massive power outage that disrupted the power supply to nearly half of the country, affecting a record 600 million people [8].

The above-mentioned large-scale outages were closely related to cascading failures in the power grid from occurrence to a large-scale spread, which shows that cascading failures are highly destructive to the power grid. Therefore, it is essential to conduct in-depth research on cascading failures and how to reduce their scale.

However, it is difficult to model the evolutionary process when cascading failures occur in a power grid system. The reason for this is that a grid contains many components and has a complex topology. The factors that cause the state of a grid to change are numerous and uncertain. In addition, operational characteristics such as the transmission line load levels in a grid can also affect the cascading state of the grid. With the advent of the Information Age, the transmission of data and operating instructions in the grid system are inextricably linked to the power communication network. When the power communication network fails to transmit information correctly or lags, it also affects cascading failures in the power grid system. The photovoltaic power generation system constructed in [5] could reduce sudden connections or cut-off load, improve system stability, and avoid cascading failures.

Recently, researchers have put great effort into modeling and understanding cascading failures in power systems. In [9], the authors showed that models based on graph theory and percolation theory do not accurately capture cascading failures. The reason for this is that when a transmission line fails, the following failed lines could be anywhere on the grid and not necessarily on an adjacent line, which is usually a typical assumption in epidemic models. There is also a discontinuous line failure propagation phenomenon in real-world power outages.

In 2015, Typhoon Rainbow hit Guangdong's power grid, and many transmission lines tripped, preventing electricity transmission from the supply side to the load side [10]. In this case, the grid operation mode calculated beforehand was no longer feasible, and the grid dispatcher needed to integrate multiple objectives (guaranteeing a power supply to important users and ensuring that the power grid is not splitting, minimizing load shedding and returning to the regular grid structure as soon as possible) to make the optimal judgment. Veteran dispatchers also tended to rely on their intuition and experience when conducting operations.

In theory, the grid dispatching problem can be solved by exhaustive enumeration to obtain the best results, but the vast action space, extremely long decision steps, excessively complex topology, and various contingencies occurring randomly make the problem size impossible to be solved by exhaustive enumeration. DRL has significant advantages in dealing with large spatial nonlinear problems, and a well-trained network can significantly reduce the decision time.

DRL combines the feature extraction capability of deep learning (DL) and the decision making capability of reinforcement learning (RL), and it is an end-to-end decision control algorithm that is widely used in dynamic decision making and real-time prediction among other uses [11].

The power communication network allows the collection of real-time information about the operation and equipment status in the power grid extremely quickly. It can automatically classify the power data to better transfer the information to the control center (CC) for further processing. The power communication network can optimize the power distribution network, make the network architecture more efficient and practical, and improve the quality of the power supply.

The proper operation of the power grid system depends on information transported via the power communication network. Interruptions or delays in the power communication network can impact the grid system's stable operation. The larger the scale of the power grid, the more information is required to be transmitted. When the transmission distance becomes more extensive, the impact of communication delays on the power grid's operation and control will become more and more significant.

In this paper, we use data from optimal DC power flow simulations to prevent the evolution of cascading failures, considering transmission line capacity constraints and communication delays, and the types of failures studied were mainly transmission line failures. We use PYPower to numerically simulate the IEEE 14 and 30 bus systems. We propose an SAC based on the DRL algorithm for mitigating cascading failures by load shedding. The use of a load shedding strategy can reduce the burden on the network, reduce the power flow in the grid, and try to avoid transmission line overloads. Then, we use the Dijkstra algorithm for route planning to find the shortest communication path to send down the load-shedding action and analyze the effect of communication delay on cascading failures. The rest of the paper is structured as follows. Section 2 details some research methods for mitigating power grid cascading failures. Section 3 describes the construction of grid and communication network models and simulation conditions. Section 4 presents the application of RL in power grid cascading failure mitigation. Section 5 gives simulation examples which prove that the proposed algorithm can effectively reduce the cascading failure scale.

2. Related Works

To clarify the occurrence and propagation mechanism of cascading failures in a power grid, researchers analyzed and studied different perspectives, such as a complex network topology and actual network characteristics, to find ways to block and reduce cascading failures. Depending on the different purposes of the modeling, the research on power system cascading failures was mainly divided into the following two ideas: (1) models with complex system theories such as power flow calculation and stability analysis as the core, including the CASCADE model [12], branching process model [13], and optimal power flow (OPF) model [14], and (2) abstract modeling of power systems and analysis of their characteristics, using complex network theories to reveal the relationship between the topology characteristics and cascading failure evolution, such as the small-world network model. The main application of this paper is the DC power flow calculation model.

The cascading failure mitigation problem could be viewed as a stochastic dynamic planning problem with unknown damage risk information. Previous studies have mainly used mathematical planning or heuristic methods to solve this problem. There are three main directions of research on cascading failure contingency control problems: (1) safety-constrained alternating current (AC) optimal power flow (SC-ACOPF) [15]; (2) optimal control [16]; and (3) traditional machine learning, such as decision trees [17] and classic RL [18].

In the early phase of power grid cascading failure analysis, most research concentrated on modeling power grid failure in a single non-interactive environment. In recent years, cascading failure studies of power grids considering the effect of power communication

networks have increased. In [19], the authors proposed a two-phase control strategy to mitigate cascading failures in a power grid by exploiting the interdependence of the power grid and the communication network.

In [20], Motter and Lai proposed the maximum load (ML) model, which has since become one of the most widely used models for studying cascading failures in complex networks. A node in a network fails and is removed from the grid when its load exceeds its maximum capacity in the ML model. At the same time, the load connected to that failed node is redistributed by some distribution method.

In [21], Cordova-Garcia proposed a load-shedding cascading control algorithm that considered the time delay of the power communication network. The authors of [22] proposed a node state influence matrix to analyze the cascading failures at the beginning of the development process. In addition, a matrix solution method based on the quadratic programming optimization model was given. Data-driven RL [23] algorithms also gained significant interest in the power systems community. A systematic RL framework was proposed to solve multilevel cascading failure problems, using the concept of “two-player games” and implementing a co-simulation platform based on DiGSILENT and MATLAB for RL [24]. The RL method based on DC-OPF was proposed to solve the cascading failure problem to some extent [25]. In [26], a new method was proposed for power system vulnerability analysis based on a double Q-learning algorithm to obtain the optimal attack results under sequential attack conditions, which provided new ideas for future research about sequential attacks on power systems.

The DRL algorithm was used for short-term voltage control of the system [27] and the determination of generator set tripping in emergency situations [28]. A deep Q-network (DQN) optimization framework which combined deep neural networks (DNNs) and RL was proposed in the single-core networks which considered mixed failure modes [29]. It provided an effective solution to reduce failure propagation and improve the robustness of SCNs.

DQN uses a greedy strategy for exploration which tends to overestimate the Q value and is not conducive to algorithm convergence. The SAC algorithm is a maximum entropy-based DRL algorithm with a stochastic strategy. Compared with the deep deterministic policy gradient (DDPG) algorithm, it can increase the randomness of exploration and the training speed to avoid obtaining local optimal solutions. Moreover, DDPG can only be applied to a continuous space. Therefore, we adopted the SAC algorithm to mitigate cascading failures.

In this paper, the SAC algorithm based on the DRL technique is used for emergency control to mitigate cascading failures when power grid failures occur. Since buses in the grid are physical entities that cannot be added or removed at will, and most transmission lines are equipped with automatic protection relay devices, the devices can trip when the power or temperature exceeds a certain threshold. Therefore, this paper mainly focuses on line failures rather than bus failures.

3. System Model Construction

3.1. Power Grid Model Construction

We used the IEEE 14 bus system in Figure 1 as an example to introduce the power grid model construction method. The system consists of N bus nodes, M transmission lines, K generators, and R loads, which can be represented by an undirected graph. The bus nodes are represented by a node collection $\mathcal{V} = \{v_i, i \in \Lambda\}$, where $\Lambda = \{1, 2, 3, \dots, N\}$. In Figure 1, $N = 14$, and $\mathcal{V} = \{v_1, v_2, \dots, v_{14}\}$.

The connectivity of the bus nodes is represented by the adjacency matrix $\mathbf{H}$. If there is a transmission line between bus v_i and bus v_j , then $\mathbf{H}(i, j) = 1$; otherwise, $\mathbf{H}(i, j) = 0$. In the constructed undirected graph, the positive and negative power flow in the transmission lines represent the direction of power flow, and a single transmission line can flow in either

the forward or reverse direction. An undirected transmission line can be considered as two directed transmission lines, and the set of directed transmission lines can be expressed as

$$E_d = \{e_{i,j} | \mathbf{H}(i,j) = 1, i \in \Lambda, j \in \Lambda\}. \quad (1)$$

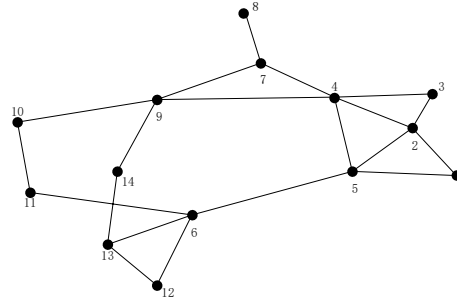


Figure 1. IEEE 14 bus system topology diagram.

Actually, $e_{i,j}$ and $e_{j,i}$ represent the same transmission line. To represent the actual grid structure, only one directed transmission line can be used to represent that physical transmission line, defining the set of physical transmission lines as follows:

$$E_u = \{e_{i,j} | \mathbf{H}(i,j) = 1, i < j, i \in \Lambda, j \in \Lambda\}. \quad (2)$$

The number of elements in the set is M . In Figure 1, we have

$$E_u = \left\{ \begin{array}{l} e_{1,2}, e_{1,5}, e_{2,3}, e_{2,4}, e_{2,5}, e_{3,4}, e_{4,5}, \\ e_{4,7}, e_{4,9}, e_{5,6}, e_{6,11}, e_{6,12}, e_{6,13}, e_{7,8}, \\ e_{7,9}, e_{9,10}, e_{9,14}, e_{10,11}, e_{12,13}, e_{13,14} \end{array} \right\}. \quad (3)$$

Thus, the modeling of the grid can be expressed by $G = (\mathcal{V}, E_u)$.

3.2. Communication Network Model Construction

We considered the power communication network from the information transmission layer, setting it to have the same topology as the power grid and to host the control and measurement agents, with the control agents having actuators that enabled remote control of power grid actions such as load shedding, circuit breaker tripping, and line disconnection. Here, we assumed that the power communication network sent appropriate signals to the grid to remotely control the grid's load-shedding action based on the DRL load shedding strategy results.

The power communication network was established with the same topology as the power grid, and the switch nodes in the power communication network had a one-to-one correspondence with the bus nodes in the power grid. We assumed that the forwarding processing delay of the switch nodes in the power communication network was the same, being fixed at τ_v . When defining the link length matrix $\mathbf{L}$, if the transmission line and the fiber optic cable link $e_{i,j} \in E_d$, then the element $d(e_{i,j})$ of row i and column j of the matrix represent the length of the line, and if $e_{i,j} \notin E_d$, then the matrix element takes the value of zero:

$$\mathbf{L}(i,j) = \begin{cases} d(e_{i,j}), & e_{i,j} \in E_d \\ 0 & , e_{i,j} \notin E_d \end{cases}, i, j \in \Lambda \quad (4)$$

The routing and policy of load-shedding commands were studied under the premise of modeling the power grid and power communication network. We defined the matrix ε to represent one of the reachable routes. If $\varepsilon(i,j) = 1$, then the reachable route contains transmission line $e_{i,j}$.

It is required to aggregate the fiber optic links and switch the nodes on a route to calculate the command transmission delay. According to the definition of route ε , the set of fiber optic links of this route can be obtained as follows:

$$E_c = \{e_{i,j} | \varepsilon(i,j) = 1, i \in \Lambda, j \in \Lambda\}. \quad (5)$$

To represent the set of switch nodes that the route passes, we took the starting nodes of all the optical links to form a node set $\{v_i | \sum_{j=1}^N \varepsilon(i,j) = 1\}$ and then added the target node to the set. The set of switch nodes can be denoted by

$$\mathcal{V}_\varepsilon = \{v_i | \sum_{j=1}^N \varepsilon(i,j) = 1\} \cup \{d^{(n)}\} \quad (6)$$

where $d^{(n)}$ is the target node.

The control command delivery delay τ_ε for route ε is the sum of the transmission delay of the fiber optical link and the processing and forwarding delay of the intermediate nodes. The intermediate nodes of the route are the nodes after removing the start and target nodes from the switch nodes through which the route passes. The number of switch nodes passing through the route is $|\mathcal{V}_\varepsilon|$, the number of intermediate nodes is $|\mathcal{V}_\varepsilon| - 2$, and the processing and forwarding delay of all intermediate nodes is

$$\tau_1 = (|\mathcal{V}_\varepsilon| - 2)\tau_v. \quad (7)$$

From the link length matrix and the set of control command distribution route links, the control command distribution route length is

$$L_\varepsilon = \sum l(e_{i,j}), e_{i,j} \in E_c. \quad (8)$$

Then, the propagation delay of the control command data in the transmission line is

$$\tau_2 = \frac{1}{c/n_r} L_\varepsilon \quad (9)$$

where c/n_r is the control command transmission speed, c is the speed of light, and n_r is the refractive index of the fiber optic cable. The control command delivery delay τ_ε is $\tau_1 + \tau_2$.

Let the information collection delay be τ_3 , which is the maximum value of the delay when the CC collects all the instantaneous data of the network and the same as the command transmission delay of the farthest node from the CC:

$$\tau_3 = (\tau_1 + \tau_2)_{\max}. \quad (10)$$

The delay in generating and issuing load-shedding control commands from the CC is τ_4 .

Therefore, the total delay for the control command to be generated and take effect is

$$\tau = \tau_1 + \tau_2 + \tau_3 + \tau_4. \quad (11)$$

This time delay is the power communication delay. The Dijkstra algorithm [30] is used to calculate the shortest route between the control node and the action application node as well as the longest way to collect the node data.

3.3. Constraints

The upper limit of the transmission line capacity with a head node i and tail node j is

$$CL_{ij} = (1 + \beta)C_{ij} \quad (12)$$

where C_{ij} is the initial power flow of the transmission line with a head node i and tail node j . Here, β denotes the safety margin of the line. The safety margin in the actual

power grid is generally not large for economic reasons, and thus $\beta = 0.5$ is used in the subsequent simulations.

3.3.1. DC Power Flow Constraint

The matrix representation of the DC power flow model is as follows:

$$\mathbf{P} = \mathbf{B}\boldsymbol{\varphi} \quad (13)$$

where $\mathbf{B}$ is the nodal admittance matrix. In the DC model, the power flow f_{ij} depends on the reactance x_{ij} and the voltage angle $\boldsymbol{\varphi}$ (i.e., $\varphi_i - \varphi_j = x_{ij}f_{ij}$). The P_i in the active power vector $\mathbf{P} = [P_1, \dots, P_i, \dots, P_N]^T$ corresponds to $\sum_{j \in V(i)} f_{ij} = P_i, \forall i, j \in \mathcal{V}$, and $(i, j) \in E_d, V(i)$, which is the set of neighborhoods of node i .

3.3.2. Basic Kirchhoff's Law and Ohm's Law Constraints

The matrix representation of the constraints are as follows:

$$\begin{aligned} \mathbf{X}\mathbf{f} &= \mathbf{g} \\ \mathbf{f} &= \mathbf{W}\mathbf{X}^T\boldsymbol{\varphi} \end{aligned} \quad (14)$$

where $\mathbf{f}$ represents the power flow in the transmission line, the connection matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_M]$ is an N -row and M -column matrix, the m th ($m = [1, 2, \dots, M]$) column vector $\mathbf{x}_m$ corresponds to the transmission line m , and the $+1$ and -1 in $\mathbf{x}_m$ denote the head and tail nodes of the transmission line, respectively. The values on the remaining nodes are zero, while $\mathbf{g}$ is the supply and demand vector. The node is connected to the generator if $g_i > 0$. The node is connected to the load if $g_i < 0$. If $g_i = 0$, then the node is connected to neither the generator nor the load. The diagonal element in $\mathbf{W}$ denotes the magnetization or weight of the corresponding line [31].

3.3.3. Supply and Demand Balance Constraint

The power generated by the generator is balanced with the energy consumed by the load.

$$\mathbf{1}^T \mathbf{p} = 0. \quad (15)$$

3.4. Cascading Failure Evolution Process

Removing a transmission line changes the network's topology. When a transmission line fails, the network begins a series of cascading iterations. First, subnet detection is performed to determine whether the subnet contains generator nodes, and if there are no generator nodes, then all unserved nodes are removed. The power flow distribution is calculated for the subnet containing generator nodes. The nodes and transmission lines that exceed their capacity will be removed. The process is repeated until all surviving nodes and transmission lines work stably, at which point the cascading failure process is completed. Figure 2 shows the power grid cascading failures occurrence flow chart.

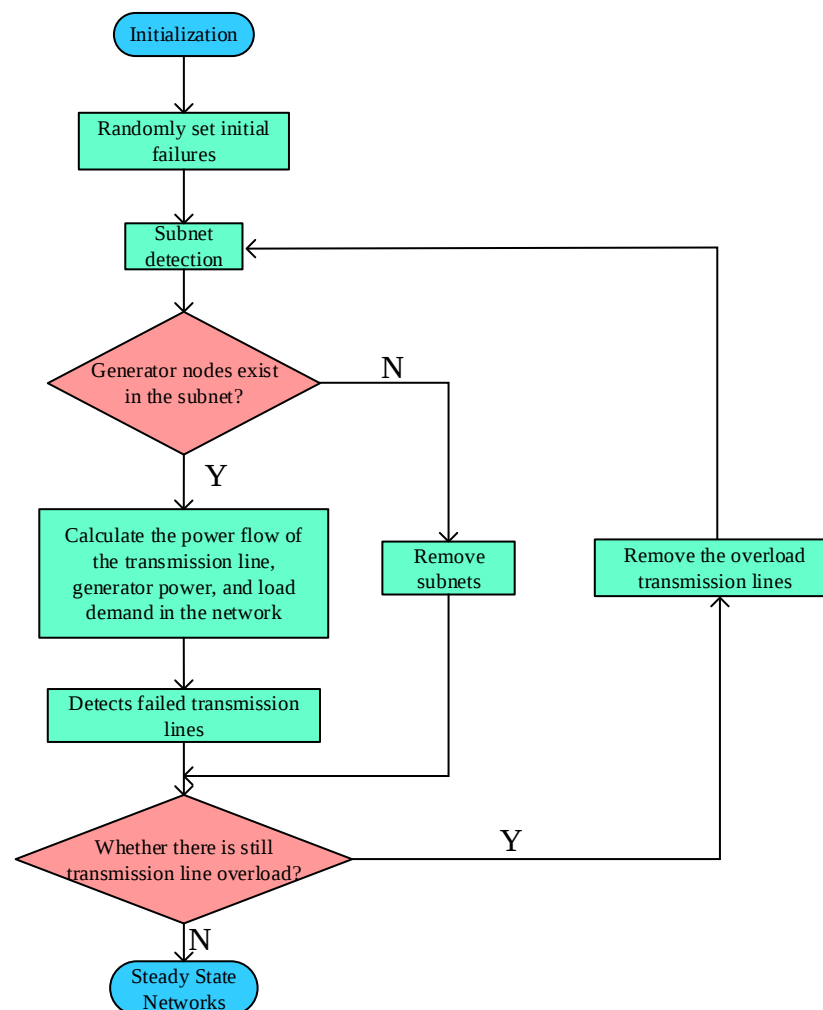


Figure 2. Network cascade failures occurrence flow chart.

3.5. Simulation Tool PYPOWER

The power system analysis toolkit PYPOWER can be understood as a Python version of MATPOWER, with similar functions to MATPOWER.

MATPOWER is an open-source Matlab-based power system simulation package widely used for research and education on AC and DC currents and OPF simulation. Many sample power flow and OPF cases are included, ranging from trivial four-bus examples to real-life cases with several thousand buses.

The simulation data mainly include the baseMVA, bus, branch, gen, and gencost, where baseMVA is a scalar and the rest are matrices. The simulations in this paper mainly used the general power flow calculation function in PYPOWER and especially considered the DC power flow. The primary data of interest for the simulation included the active part of the bus and branch matrices, and the partial structures of the two matrices are shown in Tables 1 and 2.

Table 1. Partial parameters of bus matrix.

Bus_i	Type	Pd	Qd	Gs	Bs	Area	Vm	Va	baseKV
1	1	97.6	44.2	0	0	1	1.0364	−13.537	311

From left to right, the parameters in Table 1 indicate the bus number, bus type, the active power of the bus injected load, reactive power of the bus injected load, bus parallel

conductance, bus parallel conductance, grid section number, the initial amplitude of the bus voltage, the initial phase angle of the bus voltage, and the reference voltage of the bus.

The parameters in Table 2 from left to right indicate the branch starting node number, branch ending node number, branch resistance, branch reactance, branch electric power, branch long-term allowable power, branch short-term allowable power, branch emergency allowable power, branch variation ratio, branch phase angle, and branch operating state.

Table 2. Partial parameters of branch matrix.

fbus	tbus	r	x	b	rateA	rateB	rateC	ratio	angle	status
1	2	0.3	0.4	0.6	60	60	60	0	0	1

4. Deep Reinforcement Learning

RL can interact with the environment and learn specific knowledge so that it can solve related problems. Interacting with the environment is similar to the evolution process in a power grid system. To apply RL to a practical power grid problem, it is first necessary to map the physical quantities of the power grid to the components of the RL framework (i.e., the agent, state, action, reward, policy, and Q-function).

For cascading failure problems in practical power grid systems, the state space and action space can become massive due to many bus nodes in the power grid system, and the traditional RL represented by Q-learning is no longer applicable. DNNs have obvious advantages in dealing with nonlinear large space problems. Therefore, DNNs are used to approximate the policy function and Q-function, which are combined with RL as DRL.

4.1. The Structure of RL

1. Agent: This is used to transmit data information and intervention operations in the power grid.
2. State s_t : This represents the observation value at time step t in the power grid environment. It includes the generator's active power P_g , load active power P_d , and voltage V_d , line state $\mathbf{o}$, and power flow $\mathbf{f}$.
3. Action a_t : This denotes the output value at time step t according to the state s_t and the policy network π . In our simulation environment, the action reduced the power value of the load. The reduction value was 20% of the initial power [32].
4. Policy π : This represents the probability that the agent takes action a_t in state s_t .
5. Reward r_t : The reward r_t is used as a performance metric to evaluate how good the action a_t is in the given state s_t .
6. Q-function: This indicates the expected return of taking action a_t under state s_t .

During the interaction with the power grid environment, the agent searches for an ideal policy that maximizes the long-term cumulative discount reward. The state-action Q-function describes the expectation of the cumulative discount reward and is expressed as follows:

$$Q_{\pi}(s_t, a_t) = \mathbb{E}_{\pi} \left[\sum_{\tau=0}^{\infty} \gamma^{\tau} r_{t+\tau+1} \mid s_t, a_t \right] \quad (16)$$

where $\gamma \in (0, 1]$ denotes the discount factor.

We used an actor network $\mu(\theta^{\pi} \mid s)$ to approximate the policy function and a critic network $Q(\theta^Q \mid s, a)$ to approximate the state-action Q-function. The actor network is represented by $\mu(\theta^{\pi} \mid s)$, with the state s as the input and action a as the output. The action a is the input to the critic network $Q(\theta^Q \mid s, a)$ along with the state s , and the output is the Q value of the state-action pair, while θ^Q and θ^{π} denote the critic network and actor network parameters.

4.2. SAC

Since the load-shedding action in the power grid system is discrete, we adopted the maximum entropy-based SAC [33,34] method to settle the cascading failure problems. In contrast to the deterministic policy, the entropy of the accumulative reward and the strategy is maximized, instead of simply maximizing the cumulative discount reward. The desired actor network is defined as follows:

$$\mu(\theta^{\pi*} | s_t) = \arg \max_{\mu} \mathbb{E}_{\mu} \left[\sum_{\tau=0}^{\infty} \gamma^{\tau} (r_{t+\tau+1} + \alpha \mathcal{H}(\mu(\theta^{\pi} | s_t))) \right] \quad (17)$$

where α is a coefficient that controls the importance of entropy and $\mathcal{H}(\mu(\theta^{\pi} | s_t))$ is the entropy of the actor network $\mu(\theta^{\pi} | s_t)$, denoted by

$$\mathcal{H}(\mu(\theta^{\pi} | s)) = \mathbb{E}_{\mu} [-\log(\mu(\theta^{\pi} | s))]. \quad (18)$$

In the SAC algorithm, the Q-function is rewritten as

$$Q(\theta^Q | s, a) = \mathbb{E}_{\mu} \left[\sum_{\tau=0}^{\infty} \gamma^{\tau} (r_{t+\tau+1} + \alpha \mathcal{H}(\mu(\theta^{\pi} | s_t))) \right]. \quad (19)$$

The SAC algorithm consists of five networks: an actor network $\mu(\theta^{\pi} | s)$, two critic networks $Q(\theta_1^Q | s, a)$ and $Q(\theta_2^Q | s, a)$, and two target critic networks $Q(\theta_{1,\text{target}}^Q | s, a)$ and $Q(\theta_{2,\text{target}}^Q | s, a)$. In the learning process, the smaller value of the two critic networks is used as the Q value in each step of the SAC algorithm to avoid overestimating the value. Each neural network contains a four-layer network structure (i.e., an input layer, two fully connected layers, and an output layer).

At time step t , the action a_t is obtained from the actor network. Based on the state s_t , the corresponding reward r_t is calculated, and then the state is updated to s_{t+1} , while the quaternion data (s_t, a_t, r_t, s_{t+1}) are stored in an experience replay buffer Rb of a size D . Due to the time delay of the power communication network, the state s_t and load-shedding action a_t are not available and effective in real time. According to the action a_t selected by the CC in the power communication network, the corresponding time delay τ^{a_t} is calculated. The action a_t will take effect only after τ^{a_t} time has elapsed. The number of delay steps $T_{RL}^{a_t}$ in RL is expressed as follows:

$$T_{RL}^{a_t} = \left\lfloor \frac{\tau^{a_t}}{\tau_{RL}} \right\rfloor \quad (20)$$

where $\lfloor \cdot \rfloor$ is rounded down and τ_{RL} is the time delay corresponding to each time step. Figure 3 depicts the mechanism by which the actions take effect during training.

When the data in the experience replay buffer is full, the newly deposited quaternion data replaces the initial quaternion data. At each time step t , the quaternion data of size B are randomly selected for a network update. The target critic network is used to compute

$$y_{T_i^t} = r_{T_i^t} + \gamma (\min_{j=1,2} Q(\theta_j^Q | s_{T_i^t+1}, a_{T_i^t+1}) - \alpha \mathcal{H}(\mu(\theta^{\pi} | s_{T_i^t}))). \quad (21)$$

The loss functions of the two critic networks are represented by

$$LF_t(\theta_j^Q) = \frac{1}{B} \sum_{i=1}^B (y_{T_i^t} - Q(\theta_j^Q | s_{T_i^t}, a_{T_i^t}))^2, j = 1, 2. \quad (22)$$

The loss function of the actor network is expressed as follows:

$$J_t(\theta^{\pi}) = \frac{1}{B} \sum_{i=1}^B (\alpha \log \mu(\theta^{\pi} | s_{T_i^t}, a_{T_i^t}') - \min_{j=1,2} Q(\theta_j^Q | s_{T_i^t}, a_{T_i^t}')) \quad (23)$$

where $a'_{\mathcal{I}_i^t}$ indicates the state $s_{\mathcal{I}_i^t}$ of the reparameterized sampling action. The loss function of the coefficient α of entropy is expressed as follows:

$$\mathcal{J}_t(\alpha) = -\frac{1}{B} \sum_{i=1}^B \alpha (\log \mu(\theta^\pi | s_{\mathcal{I}_i^t}) + \mathcal{H}') \quad (24)$$

where $\mathcal{H}'$ is defined as the objective entropy. Gradients are used to update the two critic networks, actor network, and α :

$$\theta_j^Q \leftarrow \theta_j^Q - l_Q \nabla_{\theta_j^Q} L_t(\theta_j^Q), j = 1, 2 \quad (25)$$

$$\theta^\pi \leftarrow \theta^\pi - l_\pi \nabla_{\theta^\pi} J(\theta^\pi) \quad (26)$$

$$\alpha \leftarrow \alpha - l_\alpha \nabla_\alpha \mathcal{J}(\alpha) \quad (27)$$

where l_Q , l_π , and l_α denote the learning rates of the corresponding network parameters and take on a value between 0 and 1. The two target critic networks are updated by a soft update:

$$\theta_{j,\text{target}}^Q \leftarrow \eta \theta_j^Q - (1 - \eta) \theta_{j,\text{target}}^Q, j = 1, 2 \quad (28)$$

where η denotes the learning rate of the target critic networks.

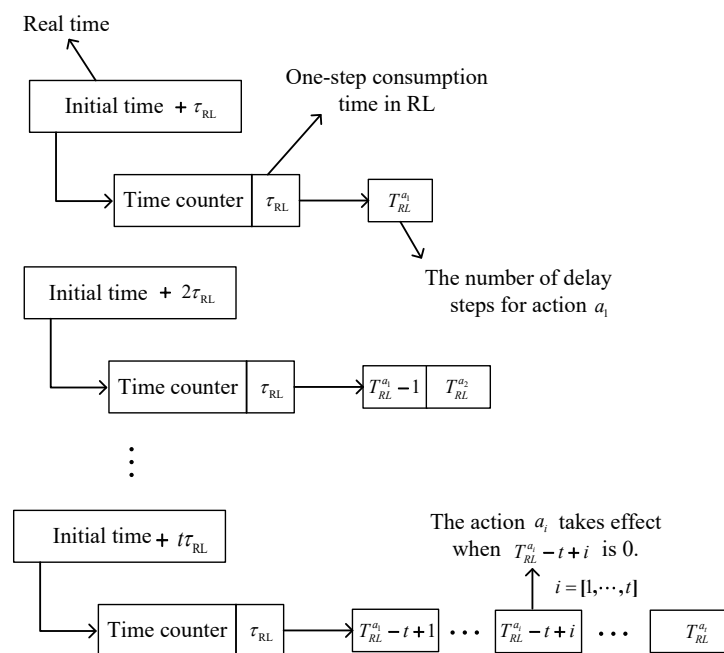


Figure 3. Schematic diagram of the actions taking effect at step t .

The structure of the SAC algorithm is shown in Figure 4, and the algorithm details of SAC are shown in Algorithm 1. At each episode, the interaction process between the SAC algorithm and the power grid environment is shown in Figure 5.

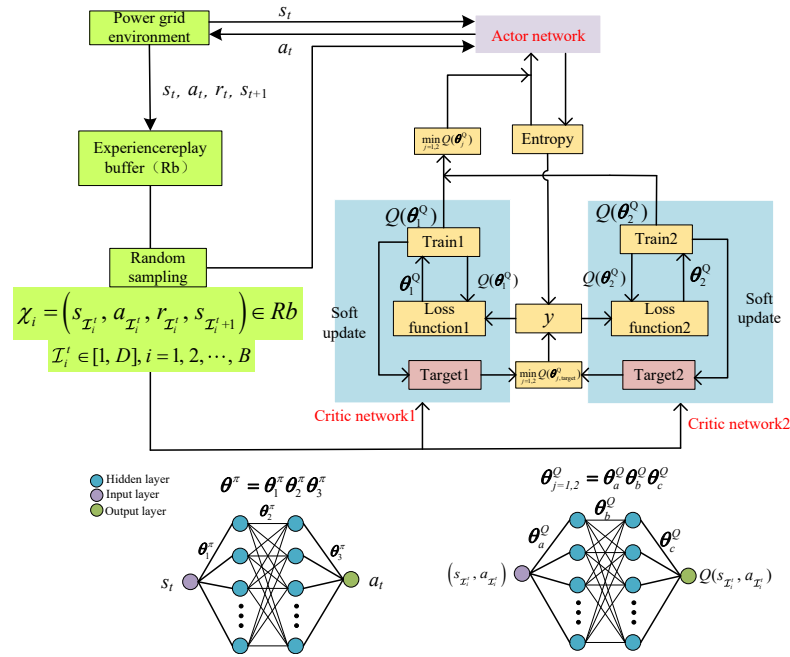


Figure 4. The structure of the SAC algorithm.

Algorithm 1 SAC algorithm.

Input: $\mathbf{P}_g, \mathbf{P}_d, \mathbf{V}_d, \mathbf{o}, \mathbf{f}, l_\pi, l_Q, l_\alpha, \eta, D$ and B .

Output: The load-shedding action for mitigating power grid cascading failures.

Initialize: Get the initial topology and data information in the power grid, set the initial failures, initialize the neural network parameters $\theta_j^Q, \theta_{j,\text{target}}^Q, \theta^\pi, j = 1, 2$, and initialize the experience replay buffer Rb .

for $episode = 1, 2, \dots, U$ **do**

 Initialize the power grid environment and get state s_1 .

for $t = 1, 2, \dots, T$ **do**

 Obtain action $a_t = \mu(\theta^\pi | s_t)$ based state s_t .

 Calculate the action a_t time delay τ^{a_t} , and action a_t takes effect depending on the time delay τ^{a_t} ,

 calculate reward r_t and obtain new state s_{t+1} .

 Place quaternion $\{s_t, a_t, r_t, s_{t+1}\}$ into Rb .

 Random sample minibatch of size B from Rb .

 Update two critic train networks with (21) (22) (25).

 Update the actor network with (23) (26).

 Update the coefficient α of entropy with (24) (27).

 Update the target networks according to (28).

end for

end for

First, we obtained the data information of the power grid system in which the transmission line failure occurred, from which we found the state s_t , including the active generator power $\mathbf{P}_g$, active load power $\mathbf{P}_d$, load voltage $\mathbf{V}_d$, line state $\mathbf{o}$, and power flow $\mathbf{f}$. According to the state s_t , the action a_t was obtained from the actor network, and the load-shedding action was performed at the corresponding load according to the power communication network delay. Then, the active load power $\mathbf{P}_d$ was updated, and the updated data were sent to PYPOWER to simulate the evolution of the grid. The power flow was obtained after the evolution finished, and we updated the line state according to Equation (29).

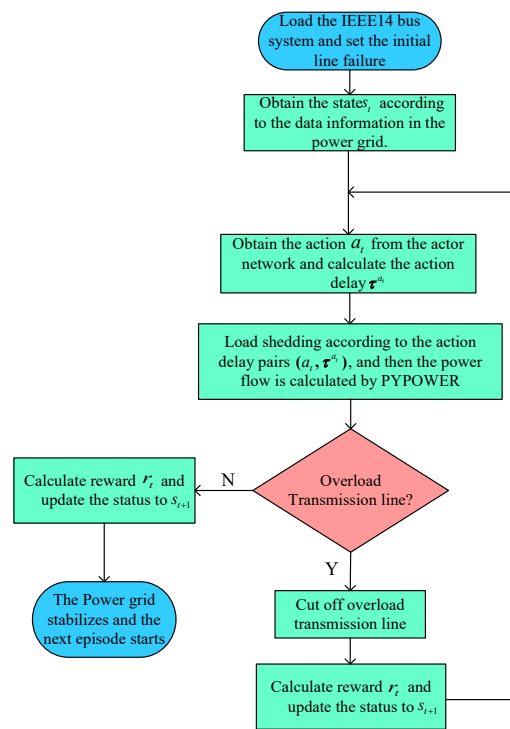


Figure 5. The interaction process between the SAC algorithm and the power grid environment in an episode.

$$o_{i,j} = \begin{cases} 0 & \text{if } f_{i,j} > C_{i,j} \\ 1 & \text{else} \end{cases} \quad (29)$$

where $f_{i,j}$ indicates the power flow of the transmission line $e_{i,j}$ and $o_{i,j}$ denotes the state of the transmission line $e_{i,j}$, where a one indicates a connected one and a zero indicates a disconnected one. The system becomes stable if no new overloaded line appears in the grid. Then, the interaction process of the current episode with the power grid environment ends, and a new interaction process of the next episode starts. If there are still overloaded lines, then the iterative process continues until the power grid reaches a stable state or the maximum number of iterative steps T .

5. Numerical Results and Evaluations

We used the IEEE 14 and 30 bus systems as simulation cases. This consisted of $N = 14$ bus nodes, $M = 20$ transmission lines, $K = 5$ generators, and $R = 11$ loads in the IEEE 14 bus system. We assumed that the distance of all transmission lines was 20 km and the communication CC was located at bus node 6. Choosing node 6 as the CC was a result of random selection. Selecting other nodes was feasible. This would change the transmission delay, and the simulation effect would differ. The node processing forwarding delay τ_v was 0.022 ms. Table 3 describes the total time delay of the load-shedding action on each load when the CC was at nodes 6 and 10. The IEEE 30 bus system environment is not further described.

All layers in the actor network and the fully connected layer in the critic networks used the ReLU activation function, and the softmax function was used in the output layers of the critic networks. The optimizers of all networks were Adam, and the two fully connected layers of all networks contained 256 and 64 neurons. The GPU and CPU in our computer used for the simulations were an RTX3050Ti and i5-11400H. The specific simulation parameters are shown in Table 4.

Table 3. Total delay of each load number (LN) when the CC is at different nodes.

Delay/ms CC \ LN	LN										
	1	2	3	4	5	6	7	8	9	10	11
Node 6	0.712	0.809	0.831	0.712	0.593	0.593	0.831	0.949	0.831	0.712	0.593
Node 10	0.949	0.831	0.831	0.712	0.831	0.712	0.712	0.831	0.593	0.593	0.593

Table 4. Simulation parameters.

Parameters	Value	Parameters	Value
U	1200	Soft update learning rate η	0.001
T	100	α learning rate l_α	0.001
Batch size B	32	Discount factor	0.96
Experience replay buffer D	10,000	Fiber refractive index nr	1.45
Actor network learning rate l_π	0.001	Critic network learning rate l_Q	0.001

For the rewards in the simulation process, we defined the following sub-rewards: load-shedding sub-reward r^1 , line disconnection sub-reward r^2 , line usage sub-reward r^3 , and residual load sub-reward r^4 :

$$r^1 = -10 \frac{n_1}{R} \quad (30a)$$

$$r^2 = -10 \frac{n_2}{M} \quad (30b)$$

$$r^3 = \sum r_{ij, \text{line_usage}} \quad (30c)$$

$$r^4 = 5 \frac{\|\mathbf{P}_d\|_1}{\|\mathbf{P}_{d, \text{ini}}\|_1} \quad (30d)$$

where n_1 denotes the total number of removing loads and n_2 denotes the total number of failure lines, while $r_{ij, \text{line_usage}}$ denotes the utilization reward of line $e_{i,j}$, defined as

$$r_{ij, \text{line_usage}} = \begin{cases} \cos\left(\frac{(b_{i,j}-0.8)\pi}{0.2}\right) - 1 & \text{if } b_{i,j} > 0.8 \\ 0 & \text{else} \end{cases} \quad (31)$$

where $b_{i,j}$ denotes the utilization of the line $e_{i,j}$. This is expressed as follows:

$$b_{i,j} = \sum \frac{f_{i,j}}{CL_{i,j}}, \quad i < j, i \in \Lambda, j \in \Lambda \quad (32)$$

We calculated the average reward with a sliding average of the rewards obtained for different episodes:

$$r_{ave} = \frac{\sum_{i=1}^{epi} r_i}{epi}, epi = 1, 2, \dots, U. \quad (33)$$

In a power grid and power communication network, the time delay of the power communication network will impact the collection of data and the distribution of actions in the power grid. In response to failures in the power grid, the load-shedding action made by the CC will lag for some time before it takes effect. Figure 6 depicts the reward varying with episodes under the influence of a delay in the power communication network. As can be seen in Figure 6, the delay reduced the average reward obtained by the SAC algorithm. At the beginning of the training process, as the agent was in the exploration phase, it took actions that may have aggravated the cascading failures, thus reducing the average reward.

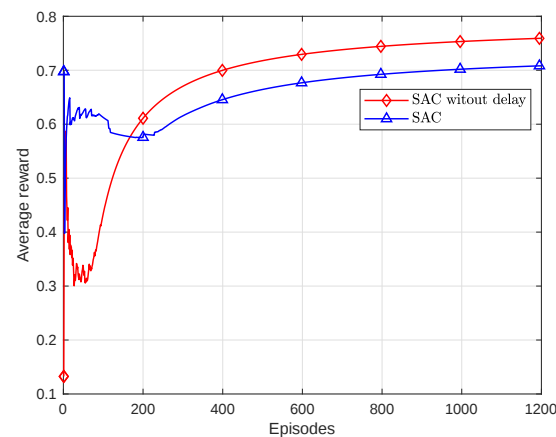


Figure 6. Average rewards varying with increasing episodes.

When using the SAC algorithm without considering the time delay, the actions taken by the agent take effect immediately in the exploration phase. In contrast, when considering the time delay, the actions that may aggravate the cascading failures at the current moment will take effect with a delay. By the time the actions take effect, they may be transformed into actions that mitigate the cascading failures for the current power grid, which in turn increases the average reward.

Table 5 depicts the network cascading failure scale for the IEEE 14 and 30 bus systems after the system is stabilized by free evolution of the power grid. F in Table 5 indicates the scale of the cascading failures. The failure scale is the ratio of the number of transmission lines that fail in the system to the total number of transmission lines. Figures 7 and 8 show the failure scale of the IEEE 14 and 30 bus systems with the SAC algorithm for the load-shedding action with or without considering the time delay compared to the failure scale without taking load-shedding action.

Table 5. Failure scale caused by partial transmission line failure of IEEE 14 and 30 bus systems.

Line Number	$e_{i,j}(14)$	$F(14)$	$e_{i,j}(30)$	$F(30)$
1	$e_{4,5}$	11	$e_{3,4}$	41
2	$e_{1,2}$	10	$e_{8,28}$	25
3	$e_{4,9}$	6	$e_{1,3}$	16
4	$e_{9,14}$	6	$e_{12,16}$	16

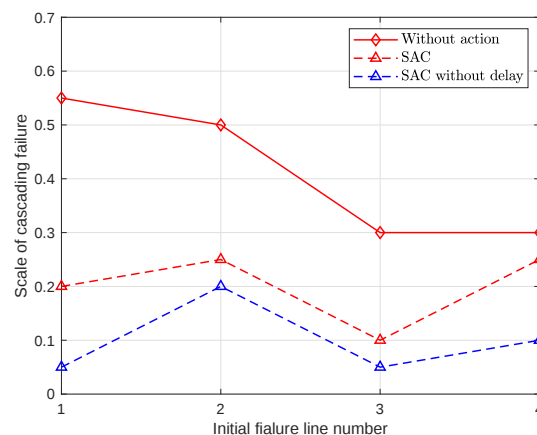


Figure 7. Relationship between initial failure line and cascading failure scale of IEEE 14 bus system under different algorithms.

In Figure 7, the even-numbered lines have a larger failure scale than the odd-numbered lines with the SAC algorithm. The cause of this phenomenon is accidental. Every numbered line has a different connection relationship limited power value. There was no regularity for the scale of failure caused by the failure of the odd- or even-numbered lines because we numbered the line numbers randomly. This phenomenon occurred coincidentally.

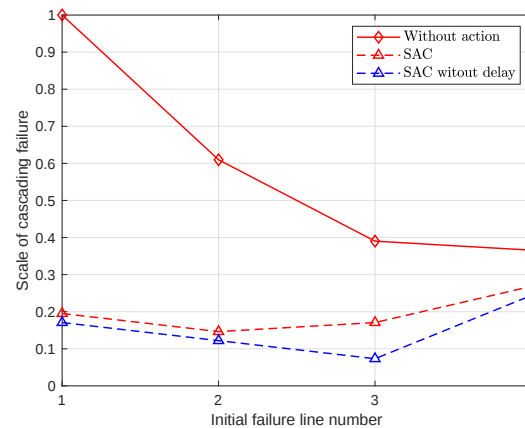


Figure 8. Relationship between initial failure line and cascading failure scale of IEEE 30 bus system under different algorithms.

As seen in Figures 7 and 8, for four different line failures, the SAC algorithm can reduce the cascading failure scale from 0.55, 0.5, 0.3, and 0.3 as well as 1, 0.61, 0.39, and 0.39 to 0.2, 0.25, 0.1, and 0.25 as well as 0.2, 0.15, 0.17, and 0.27, respectively. With the network trained by the SAC algorithm, the CC can quickly make the corresponding load-shedding action according to the failure, and the time from the initial failure state to the stable state was 42.86 ms. The less time it takes for the power grid system to return to the stable state, the smaller the scale of the resulting cascading failure.

From Figures 6–8, we can conclude that the load shedding strategy based on the SAC algorithm can obviously reduce the cascading failure scale of a power grid. The action loses its timeliness due to the impact of the communication network delay. It is slightly less effective at mitigating cascading failures than not considering the communication delay.

The percentage of remaining load under different initial line failures after load shedding can be seen in Figure 9. The remaining load for some numbered loads was zero. This was due to the load disconnected from the network, which was caused by line disconnection during the evolution of the grid. Since the network structure in the IEEE 14 bus system is relatively simple, the grid with SAC reached a stable state after 2–3 steps of load shedding. Therefore, the load shedding of SAC in Figure 9 is relatively small. The SAC algorithm can retain most of the load compared with the free evolution without action.

Figure 10 depicts the transmission line utilization of the IEEE 14 bus system with the SAC algorithm after setting different initial failure lines. As can be seen in Figure 10, most of the line utilization in the IEEE 14 bus system was below 85% with the SAC algorithm, avoiding highly loaded transmission lines, which can significantly alleviate the cascading failure of the power grid.

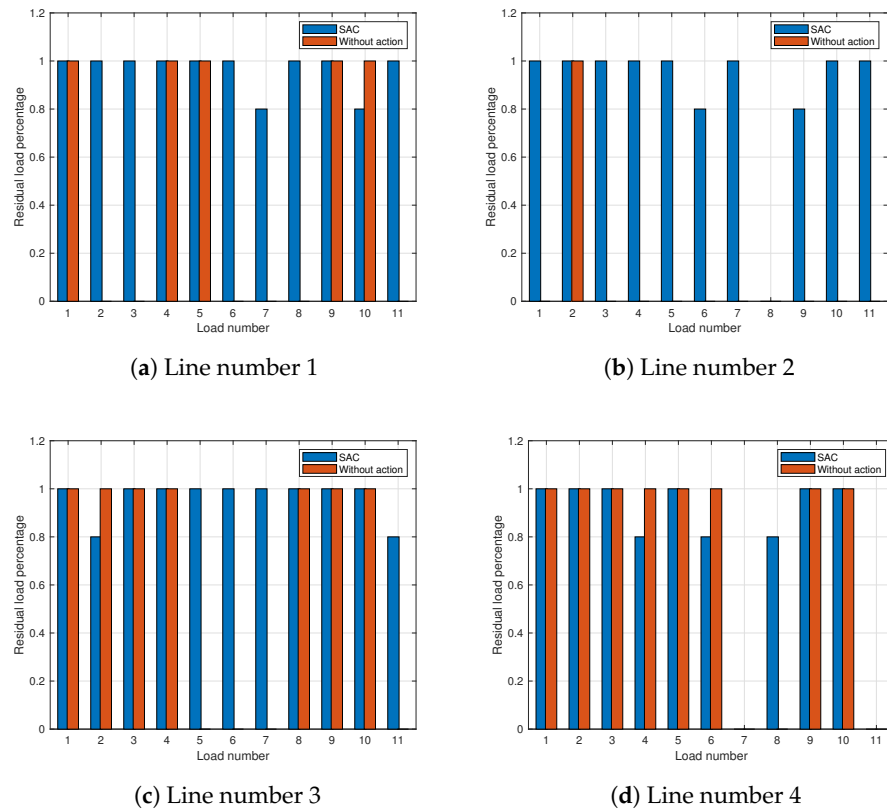


Figure 9. Residual load of different initial failure lines after system stabilization using the SAC algorithm in the IEEE 14 bus system.

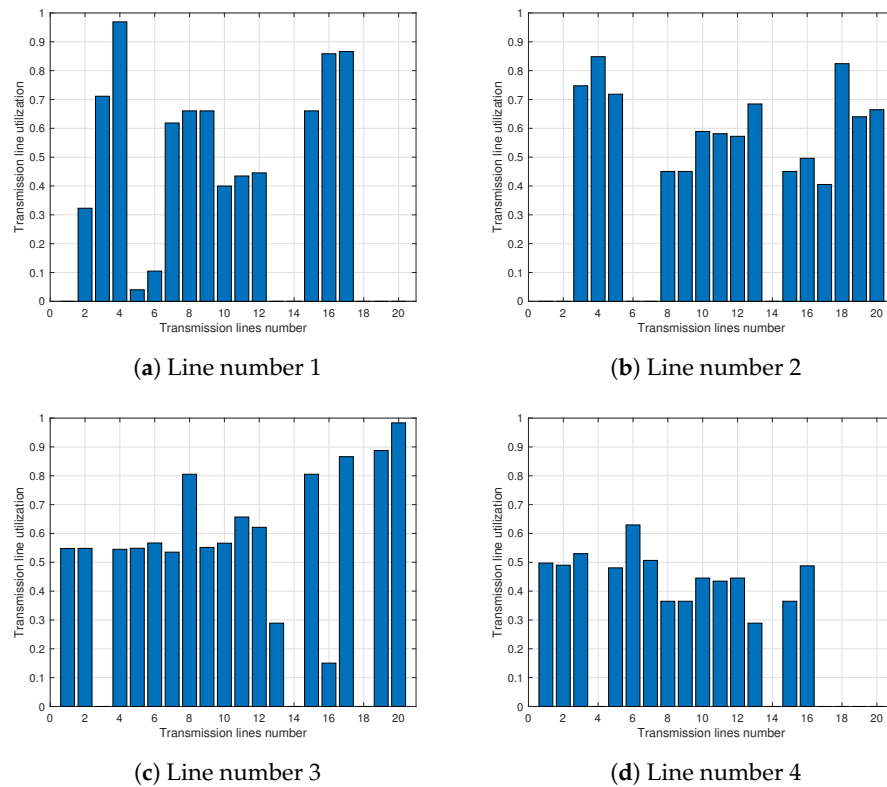


Figure 10. Line utilization of different initial failure lines after system stabilization using the SAC algorithm in the IEEE 14 bus system.

6. Conclusions

In this paper, we addressed the problem of grid cascading failures triggered by a single transmission line failure and proposed a DRL-based SAC algorithm to mitigate grid cascading failures, considering communication delay by load shedding. The simulation environment was established with IEEE 14 and 30 bus systems, and the simulation process considered the latency of a load-shedding action due to communication delay and verified the feasibility and effectiveness of the SAC algorithm for mitigating cascading failures. The trained network can decide on actions and give commands quickly when a specific initial failure is encountered, reducing the scale of cascading failures. However, affected by the information transmission delay of the power communication network, there was a noticeable gap in the effectiveness of cascading failure mitigation compared with when there was no delay. In future research, we will consider different control actions, such as changing line connection relationships and node voltages, as well as simulations for different failure types in the power grid and power communication network, such as node failure, switch failure, and communication route blockage, to verify the effectiveness of the algorithm for various scenarios.

Author Contributions: Conceptualization, Y.W. and W.Z.; Methodology, Y.W., A.T., Y.J. and L.M. (Liqiang Ma); Software, A.T., Y.J. and L.M. (Liqiang Ma); Validation, L.M. (Liqiang Ma); Formal analysis, Y.W. and W.Z.; Investigation, A.T.; Resources, Y.J., W.Z. and L.M. (Liang Ma); Data curation, L.M. (Liang Ma) and C.S.; Writing—original draft, Y.W. and W.Z.; Writing—review & editing, A.T., L.M. (Liqiang Ma) and J.S.; Visualization, L.M. (Liang Ma) and C.S.; Supervision, J.S.; Project administration, A.T.; Funding acquisition, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science and Technology project of the State Grid Corporation of China (Research on Dispatching Fusion Communication Oriented to Power Communication Network and Its Cooperative Control with Power Network Operation, 52060022001B).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Shuvro, R.A.; Wangt, Z.; Das, P.; Naeini, M.R.; Hayat, M.M. Modeling cascading-failures in power grids including communication and human operator impacts. In Proceedings of the 2017 IEEE Green Energy and Smart Systems Conference (IGESSC), Long Beach, CA, USA, 6–7 November 2017; pp. 1–6. [\[CrossRef\]](#)
- Wang, Z.; Rahnamay-Naeini, M.; Abreu, J.M.; Shuvro, R.A.; Das, P.; Mammoli, A.A.; Ghani, N.; Hayat, M.M. Impacts of Operators' Behavior on Reliability of Power Grids During Cascading Failures. *IEEE Trans. Power Syst.* **2018**, *33*, 6013–6024. [\[CrossRef\]](#)
- Dobson, I.; Carreras, B.; Newman, D. Branching Process Models for the Exponentially Increasing Portions of Cascading Failure Blackouts. In Proceedings of the 38th Annual Hawaii International Conference on System Sciences, Big Island, HI, USA, 3–6 January 2005; p. 64a. [\[CrossRef\]](#)
- Saleh, B.; Yousef, A.M.; Abo-Elyousr, F.K.; Mohamed, M.; Abdelwahab, S.A.M.; Elnozahy, A. Performance Analysis of Maximum Power Point Tracking for Two Techniques with Direct Control of Photovoltaic Grid -Connected Systems. *Energy Sources Part A Recover. Util. Environ. Eff.* **2021**, *44*, 413–434. [\[CrossRef\]](#)
- Eid, M.A.E.; Abdelwahab, S.A.M.; Ibrahim, H.A.; Alaboudy, A.H.K. Improving the Resiliency of a PV Standalone System Under Variable Solar Radiation and Load Profile. In Proceedings of the 2018 Twentieth International Middle East Power Systems Conference (MEPCON), Cairo, Egypt, 18–20 December 2018; pp. 570–576. [\[CrossRef\]](#)
- Anderson, R. *Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations*; Technical Report; U.S.-Canada Power System Outage Task Force: Washington, DC, USA, 2004.
- Makarov, Y.; Reshetov, V.; Stroeve, A.; Voropai, I. Blackout Prevention in the United States, Europe, and Russia. *Proc. IEEE* **2005**, *93*, 1942–1955. [\[CrossRef\]](#)
- Lai, L.L.; Zhang, H.T.; Lai, C.S.; Xu, F.Y.; Mishra, S. Investigation on July 2012 Indian blackout. In Proceedings of the 2013 International Conference on Machine Learning and Cybernetics, Tianjin, China, 14–17 July 2013; Volume 1, pp. 92–97. [\[CrossRef\]](#)
- Bernstein, A.; Bienstock, D.; Hay, D.; Uzunoglu, M.; Zussman, G. Power grid vulnerability to geographically correlated failures—Analysis and control implications. In Proceedings of the IEEE INFOCOM 2014—IEEE Conference on Computer Communications, Toronto, ON, Canada, 27 April–2 May 2014; pp. 2634–2642. [\[CrossRef\]](#)
- Lin, Z.; Wen, F.; Wang, H.; Lin, G.; Mo, T.; Ye, X. CRITIC-Based Node Importance Evaluation in Skeleton-Network Reconfiguration of Power Grids. *IEEE Trans. Circuits Syst. II Express Briefs* **2018**, *65*, 206–210. [\[CrossRef\]](#)

11. Huang, Q.; Huang, R.; Hao, W.; Tan, J.; Fan, R.; Huang, Z. Adaptive Power System Emergency Control using Deep Reinforcement Learning. *IEEE Trans. Smart Grid* **2020**, *11*, 2933191. [\[CrossRef\]](#)
12. Qu, Y.; Gao, M.; Chen, Y.; Jing, F.; Zhang, S.; Zhang, L. An analysis of the invulnerability for communication networks base on Cascading failure model. In Proceedings of the 2020 International Conference on Robots Intelligent System (ICRIS), Sanya, China, 7–8 November 2020; pp. 154–157. [\[CrossRef\]](#)
13. Qi, J.; Ju, W.; Sun, K. Estimating the Propagation of Interdependent Cascading Outages With Multi-Type Branching Processes. *IEEE Trans. Power Syst.* **2017**, *32*, 1212–1223. [\[CrossRef\]](#)
14. Li, P.; Sheng, W.; Duan, Q. Optimal Power Flow Calculation Method for AC/DC Hybrid Distribution Network Based on Power Router. In Proceedings of the 2021 6th Asia Conference on Power and Electrical Engineering (ACPEE), Chongqing, China, 8–11 April 2021; pp. 1694–1699. [\[CrossRef\]](#)
15. Misra, S.; Roald, L.; Vuffray, M.; Chertkov, M. Fast and Robust Determination of Power System Emergency Control Actions. *arXiv* **2017**, arXiv:1707.07105.
16. Li, Z.; Yao, G.; Geng, G.; Jiang, Q. An Efficient Optimal Control Method for Open-Loop Transient Stability Emergency Control. *IEEE Trans. Power Syst.* **2017**, *32*, 2704–2713. [\[CrossRef\]](#)
17. Li, Y.; Song, X.; Zhao, S.; Gao, F. A Line-Fault Cause Analysis Method for Distribution Network Based on Decision-Making Tree and Machine Learning. In Proceedings of the 2020 5th Asia Conference on Power and Electrical Engineering (ACPEE), Chengdu, China, 4–7 June 2020; pp. 1–5. [\[CrossRef\]](#)
18. Glavic, M.; Fonteneau, R.; Ernst, D. Reinforcement Learning for Electric Power System Decision and Control: Past Considerations and Perspectives. *IFAC-PapersOnLine* **2017**, *50*, 6918–6927. [\[CrossRef\]](#)
19. Parandehgheibi, M.; Modiano, E.; Hay, D. Mitigating cascading failures in interdependent power grids and communication networks. In Proceedings of the 2014 IEEE International Conference on Smart Grid Communications (SmartGridComm), Venice, Italy, 3–6 November 2014; pp. 242–247. [\[CrossRef\]](#)
20. Motter, A.E.; Lai, Y.C. Cascade-based attacks on complex networks. *Phys. Rev. E Stat. Nonlin Soft Matter Phys.* **2003**, *66*, 065102. [\[CrossRef\]](#) [\[PubMed\]](#)
21. Cordova-Garcia, J.; Wang, X.; Xie, D.; Zhao, Y.; Zuo, L. Control of Communications-Dependent Cascading Failures in Power Grids. *IEEE Trans. Smart Grid* **2019**, *10*, 5021–5031. [\[CrossRef\]](#)
22. Yu, C.; Zhang, L.; Sun, S.; Sun, Y.; Yin, X.; Wang, C. Cascading Failure Propagation in Cyber Physical Power Systems under Extreme Weather Events. In Proceedings of the 2022 4th International Conference on Smart Power & Internet Energy Systems (SPIES), Beijing, China, 27–30 October 2022; pp. 1486–1490. [\[CrossRef\]](#)
23. Kumar, A.; Fu, J.; Tucker, G.; Levine, S. Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction. *arXiv* **2019**, arXiv:1906.00949.
24. Zhu, Y. Power Grid Cascading Failure Mitigation by Reinforcement Learning. In Proceedings of the ICML 2021 Workshop on Tackling Climate Change with Machine Learning, Virtual, 18–24 July 2021.
25. Zhu, Y.; Liu, C. Mitigating Multi-Stage Cascading Failure by Reinforcement Learning. In Proceedings of the 2019 IEEE Innovative Smart Grid Technologies—Asia (ISGT Asia), Chengdu, China, 21–24 May 2019; pp. 3724–3728. [\[CrossRef\]](#)
26. Li, X.; Hu, W.; Hou, P.; Shang, T.; Gao, X.; Li, D. Optimal Attack Strategy of Power Grid based on Double Q-learning Algorithm. In Proceedings of the 2021 IEEE 5th Conference on Energy Internet and Energy System Integration (EI2), Taiyuan, China, 22–24 October 2021; pp. 2460–2465. [\[CrossRef\]](#)
27. Zhang, J.; Lu, C.; Si, J.; Song, J.; Su, Y. Deep Reinforcement Learning for Short-term Voltage Control by Dynamic Load Shedding in China Southern Power Grid. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8. [\[CrossRef\]](#)
28. Liu, W.; Zhang, D.; Wang, X.; Hou, J.; Liu, L. A Decision Making Strategy for Generating Unit Tripping Under Emergency Circumstances Based on Deep Reinforcement Learning. *Zhongguo Dianji Gongcheng Xuebao/Proc. Chin. Soc. Electr. Eng.* **2018**, *38*, 109–119. [\[CrossRef\]](#)
29. Zhang, L.; Zhou, J.; Ma, Y.; Shen, L. Sequential Topology Attack of Supply Chain Networks Based on Reinforcement Learning. In Proceedings of the 2022 International Conference on Cyber-Physical Social Intelligence (ICCSI), Nanjing, China, 18–21 November 2022; pp. 744–749. [\[CrossRef\]](#)
30. Kumar, V.; Jangir, S.; Patanvariya, D.G. Traffic Load Balancing in SDN Using Round-Robin and Dijkstra Based Methodology. In Proceedings of the 2022 International Conference for Advancement in Technology (ICONAT), Goa, India, 21–22 January 2022; pp. 1–4. [\[CrossRef\]](#)
31. Ba, Q.; Savla, K. A dynamic programming approach to optimal load shedding control of cascading failure in DC power networks. In Proceedings of the 55th IEEE Conference on Decision and Control, CDC 2016, Las Vegas, NV, USA, 12–14 December 2016; pp. 3648–3653. [\[CrossRef\]](#)
32. Kuiava, R.; Bogodorova, T.; Fernandes, T.C.C.; Ramos, R.A. A Study on the Relation between the Maximum Loadability Point and Undervoltage Load Shedding Schemes. In Proceedings of the 2020 IEEE Power & Energy Society General Meeting (PESGM), Montreal, QC, Canada, 2–6 August 2020; pp. 1–5. [\[CrossRef\]](#)

33. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; Volume 80, pp. 1861–1870.
34. Ding, F.; Ma, G.; Chen, Z.; Gao, J.; Li, P. Averaged Soft Actor-Critic for Deep Reinforcement Learning. In Proceedings of the 2017 International Conference on Computing, Networking and Communications (ICNC), Silicon Valley, CA, USA, 26–29 January 2021; pp. 1488–1493. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.