energies

Article

Power System Fault Diagnosis Method Based on Deep Reinforcement Learning

Zirui Wang [1,*], Ziqi Zhang [1], Xu Zhang [1], Mingxuan Du [1], Huiting Zhang [2] and Bowen Liu [1]

[1] School of Electrical and Electronic Engineering, North China Electric Power University, Beijing 102206, China
[2] State Grid Shanxi Electric Power Company Skills Training Center, Shanxi Electric Power Vocational and Technical Institute, Taiyuan 030021, China
* Correspondence: 120212201108@ncepu.edu.cn

Abstract: Intelligent power grid fault diagnosis is of great significance for speeding up fault processing and improving fault diagnosis efficiency. However, most of the current fault diagnosis methods focus on rule diagnosis, relying on expert experience and logical rules to build a diagnosis model, and lack the ability to automatically extract fault knowledge. For switch refusal events, it is difficult to determine a refusal switch without network topology. In order to realize the non-operating switch identification without network topology, this paper proposes a power grid fault diagnosis method based on deep reinforcement learning for alarm information text. Taking the single alarm information of the non-switch refusal sample as the research object, through the self-learning ability of deep reinforcement learning, it learns the topology connection relationship and action logic relationship between equipment, protection and circuit breakers contained in the alarm information, and realizes the detection of fault events. The correct prediction of the fault removal process after the occurrence, based on this, determines the refusal switch when the switch refuses to operate during the fault removal process. The calculation example results show that the proposed method can effectively diagnose the refusal switch of the switch refusal event, which is feasible and effective.

Keywords: alarm information; deep reinforcement learning; fault diagnosis; deep Q-network

check for updates

Citation: Wang, Z.; Zhang, Z.; Zhang, X.; Du, M.; Zhang, H.; Liu, B. Power System Fault Diagnosis Method Based on Deep Reinforcement Learning. *Energies* **2022**, *15*, 7639. https://doi.org/10.3390/en15207639

Academic Editor: Ahmed Abu-Siada

Received: 8 September 2022
Accepted: 13 October 2022
Published: 16 October 2022

1. Introduction

At present, with the continuous development of new power systems with new energy sources, the power grid structure is becoming more and more complex, and the power system tends to be power electronic and distribution network active. Power grid faults show more complex modes, and fast and intelligent power grid fault diagnosis has become an urgent need for current power grid dispatchment [1].

Power grid fault diagnosis is an effective means to control the development of faults by analyzing the electrical and non-electrical fault information collected by the monitoring system to determine the fault area and identify the faulty equipment. Experts and scholars in the field of electric power have conducted in-depth research on this and have proposed systems based on expert systems [2], Bayesian networks [3], fuzzy sets [4], and analytical models [5] as well as other rule-based power grid fault diagnosis methods. These methods establish a diagnostic model based on the fault occurrence mechanism and power grid topology, transform the diagnosis process into a model solving problem, and realize the correct diagnosis of most fault events. However, with the scale expansion, complex structure and intelligent system of China power grid, the information uploaded from the secondary measurement monitoring device to the energy management system (EMS) is rapidly expanding, and it is developing in the direction of a mass scale. The above diagnostic methods cannot be the direct fault diagnosis of massive alarm information, as the information still needs to rely on manual experience to screen key information first, which cannot meet the needs of rapid diagnosis.

In recent years, artificial intelligence technology represented by deep learning has become good at dealing with complex and uncertain problems and has strong advantages in feature self-learning, end-to-end, etc. [6–9], which provide new opportunities for power grid fault diagnosis ideas. Reference [10] proposes a fault diagnosis method for MTDC lines that takes into account both rapidity and accuracy in view of the characteristics of the fast rise of fault currents. A dual-branch structure convolutional neural network—Parallel Convolutional Neural Network (PCNN)—of the road and fault location branch is proposed, as is a P-CNN training method based on transfer learning. Reference [11] builds a data-driven model for fault diagnosis and fault location based on electrical quantities. However, this research is based on the PMU measurement data and cannot be carried out for alarm information. In [12,13], the k-means clustering method is applied to realize power grid diagnosis with the text mining [12] and fault coding [13] of alarm information as model inputs. Reference [14] proposes a method for the autonomous recognition of alarm events by fusing a knowledge base and deep learning using a convolutional neural network (CNN) for the local feature extraction of alarm information to achieve the intelligent recognition of alarm events. Reference [15] takes the text of power grid fault handling plans as the research object and proposes a top-down and bottom-up method for constructing a power grid fault handling knowledge graph and solves the problem of knowledge extraction in the power field. In [16], the text vectorization of alarm information is completed and a vector space model of alarm information is established. The fault types are output through the random forest (RF) model, achieving high diagnostic accuracy. Reference [17] proposes a fault diagnosis method for a power grid based on a deep pyramid convolutional neural network (DPCNN). A fault classification model and key information extraction model based on DPCNN are established for alarm information sets and a single alarm information text, which realizes fault classification and key information extraction. However, since the alarm information is a plain text data, it lacks the description of the power grid topology. For more complex fault types, such as switch refusal and protection refusal, the above fault diagnosis method still needs to combine the power grid topology to identify the refusal equipment.

Therefore, how to determine the refusal device only based on the text of the alarm information in the absence of a network topology needs to be solved urgently. If a single alarm message of a non-switch refusal sample is used as the research object, the logical relationship between equipment, protection and circuit breakers contained in each alarm message is studied, and then the text of the alarm message when the normal fault is removed can be predicted. The prediction process of alarm information is regarded as a sequence decision problem, and deep reinforcement learning [18–21] can be used to analyze data features and make decisions to solve this problem. Deep reinforcement learning combines the powerful perceptual understanding ability of deep learning and the decision-making ability of reinforcement learning to achieve a one-to-one correspondence from perception to action [22,23]. Deep learning uses information from the environment to extract features and generate a state representation of the current environment. Reinforcement learning achieves a desired goal based on the current state. Deep learning uses information from the environment to extract features and generate a state that represents the current environment. Reinforcement learning achieves a desired goal based on the current state. Currently, deep reinforcement learning has achieved remarkable results in many aspects involving power systems, such as the coordinated control of hybrid energy storage in microgrids [24], unit trip strategy in emergency situations [25], and AGC strategy research [26], etc. In turn, it shows its effectiveness in solving decision-making problems. For example, [26] explored a deep reinforcement learning algorithm for action exploration perception thinking, namely DDQN-AD, from the perspective of automatic power generation control. By taking the prediction mechanism of the neural network as the action selection mechanism of reinforcement learning and introducing the AD strategy with action exploration perception thinking and taking the regional control error and carbon emission as the comprehensive reward function, the optimal control strategy in the strong random environment is obtained. Then the random disturbance problem caused by the large-scale access of distributed energy to the power grid is solved

In this paper, a deep reinforcement learning-based grid fault diagnosis method is proposed to identify refusing equipment in the absence of network topology. First, the text environment of the alarm information is transformed into a vectorized alarm information sequence environment and then the state, action and reward of the deep reinforcement learning agent DQN model are clearly defined in order to complete the connection between the DQN agent and the alarm information sequence environment interactions; then a deep reinforcement learning agent that can accurately predict the text of alarm information can be trained. Then, taking the alarm information of non-switch refusal samples as the research object, the intelligent agent is used to learn the logical relationship between equipment, protection and circuit breakers contained in each alarm information, so as to realize the correct fault removal process after the fault event occurs. Predictions, based on this, determine the refusal switch when the switch refuses to act during the fault removal process. Finally, the effectiveness and feasibility of the method are verified based on the alarm information of the simulation system and the actual power grid.

2. Deep Reinforcement Learning

2.1. Reinforcement Learning

Different from deep learning, reinforcement learning does not need to add labels to a large number of sample data for supervised training. Feedback can be used to achieve the purpose of learning and finally find the optimal strategy that solves the problem.

The learning process of reinforcement learning can be represented by a Markov decision process (MDP), which can be represented by a quadruple (s, a, p, r), defined as the four elements of reinforcement learning. The elements are defined as follows:

1. State (s): the observation results of the environment, the state is different at different times, and the state at each time constitutes the state space S;
2. Action (a): The behavior of the agent according to the state, and the actions taken in different states can constitute the action space A;
3. State transition probability (p): the probability that the agent will transition to a specific state at the next moment after taking corresponding actions according to the current state obtained from the environment;
4. Reward (r): The reward value of the environment for the agent to enter the next state after taking a certain action in the current state.

The Markov decision process is shown in Figure 1. After the process is over, the agent obtains an action sequence, called a policy, denoted as π, and returns the cumulative reward of the policy, as shown in Equation (1).

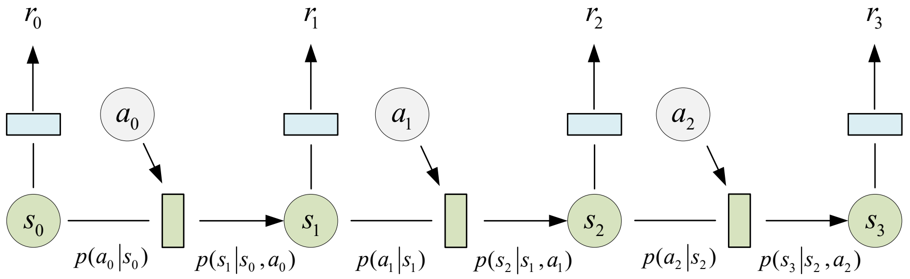


Figure 1. Markov decision process.

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, \tag{1}$$

In the formula, G_t is the reward value obtained from the environment at time t and γ is the discount factor, indicating the weight of the future reward in the cumulative reward value.

For the strategy π, the larger the cumulative reward value during the execution of the entire strategy, the better the strategy. Therefore, the state–action behavior value function $Q_\pi(s,a)$ is introduced to evaluate the value of executing action a in a specific state s, and the Bellman value of $Q_\pi(s,a)$ is calculated. The optimal strategy is obtained by the iterative solution of the equation, $Q_\pi(s,a)$, and its Bellman equation is shown in Equations (2) and (3).

$$Q(s,a) = E_\pi[G_t|s_t = s, a_t = a], \tag{2}$$

$$Q_\pi(s,a) = E_\pi[r_{t+1} + \gamma Q_\pi(s_{t+1}, a_{t+1})|s_t, a_t], \tag{3}$$

2.2. Deep Q-Network

The Deep Q-network (DQN) is a typical deep reinforcement learning method based on value function. It uses DNN to approximate the reward value function and solves the problem of the dimension disaster of iteratively solving the value function when the state space is large. The DQN algorithm uses a DNN with a weight of θ to approximately represent the current value function, then calculates the loss function according to the correct Q value provided by reinforcement learning and continuously updates the network weight θ by making the loss function $L(\theta)$ reach the minimum. $L(\theta)$ and the update of the weight θ are:

$$L(\theta) = E[(Y_i - Q(s_t, a_t;\theta))^2] = E[(r_{t+1} + \gamma\max_a Q(s_{t+1}, a_{t+1};\theta) - Q(s_t, a_t;\theta))^2] \tag{4}$$

$$\theta_{t+1} = \theta_t + \alpha(r_{t+1} + \gamma\max_a Q(s_{t+1}, a;\theta) - Q(s_t, a_t;\theta))\nabla_\theta Q(s_t, a_t;\theta), \tag{5}$$

Of these, Y_i is the optimization objective of the value function, that is, the target Q value; $Q(s_t, a_t;\theta)$ is the estimation of $Q(s_t, a_t)$; and α is the learning rate.

In order to reduce the strong correlation between the Q value output by the network and the target Y_i and improve the stability of the algorithm operation, DQN uses a separate network, that is, the target net to generate the target Q value during the training process. During the training process, the parameters θ of the current network must be updated for each step, while the parameters θ' of the target network remain unchanged. Only after the C-step iteration are the parameters of the current network copied to the target network. At this point, the loss function is:

$$L_i(\theta_i) = E[r_{t+1} + \gamma\max_a Q(s_{t+1}, a_{t+1};\theta'_i) - Q(s_t, a_t;\theta_i))^2], \tag{6}$$

In the formula, θ_i is the parameter value of the current network when running the *i*-th step and θ'_i is the parameter of the target network.

Update the current network parameter θ and target network parameter θ' by the stochastic gradient descent:

$$\nabla_\theta L(\theta) = E(r + \gamma\max Q(s_{t+1}, a_{t+1};\theta') - Q(s_t, a_t;\theta)]\nabla Q(s_t, a_t;\theta), \tag{7}$$

$$\theta_{i+1} = \theta_i + \alpha\nabla_{\theta_i} L_i(\theta_i), \tag{8}$$

$$\theta'_i = \theta_{i+C}, \tag{9}$$

In addition, under the DQN algorithm, the agent usually adopts the *ε-greedy* strategy to select the action of each step from the action set, that is, when the probability is less than ε, the action is randomly selected; otherwise the action with the highest Q value at the current moment is selected, as shown in the following formula:

$$\pi(a|s) = \begin{cases} \varepsilon/m + 1 - \varepsilon & if\ a = \text{argmax}\ Q(s,a) \\ \varepsilon/m & \text{others} \end{cases}, \tag{10}$$

Of these, m is all optional actions.

At the same time, DQN introduces an experience replay mechanism, which stores the experience of each step in the experience pool and forms a replay experience memory sequence. When training DNN, the corresponding number is randomly selected from the experience pool according to a certain batch size each time. The historical samples are used for training and the parameters of the neural network are updated, which improves the update efficiency of the neural network and reduces the correlation between the sample data.

3. Power Grid Fault Diagnosis Model Based on DQN

The process of alarm information text prediction can be regarded as a sequence decision problem, so the deep reinforcement learning method can be used for prediction. In this paper, the deep Q network (DQN) model is selected as the fault diagnosis model.

3.1. Alarm Information Text Processing

Since the alarm information text is completely based on the natural language text environment, the alarm information text should be digitized first and each alarm information should be converted into a digital vector representing its semantics, taking the alarm information samples shown in Table 1 as an example (the time information of each alarm information is not necessarily related, so it has been de-sequentially processed).

Table 1. Alarm information example.

Device	Protection	Protection/Switch State
Yandang station main transformer fault recorder	Recorder starts	Action
Yandang station 220 kV fault recorder	Recorder starts	Action
Yandang station 110 kV fault recorder	Recorder starts	Action
Yandang station 10 kV Yan 957 Line	Protect	Action
Yandang station 10 kV Yan 957 Line	Overcurrent I stage	Action
Yandang station 10 kV Yan 957 Line 957 Switch	0	Switch general outlet tripping action
Yandang station 10 kV Yan 957 Line 957 Switch	0	Open
Yandang station 10 kV Yan 957 Line	Protect	Reset
Yandang station 10 kV Yan 957 Line	Overcurrent I stage	Reset
Yandang station 10 kV Yan 957 Line	Recloser	Action
Yandang station 10 kV Yan 957 Line 957 Switch	0	Close
Yandang station 10 kV Yan 957 Line 957 Switch	0	Spring not charged action
Yandang station 10 kV Yan 957 Line	Recloser	Reset
Yandang station main transformer fault recorder	Recorder starts	Reset
Yandang station 220 kV fault recorder	Recorder starts	Reset
Yandang station 110 kV fault recorder	Recorder starts	Reset
Yandang station 10 kV Yan 957 Line 957 Switch	0	Switch general outlet tripping reset
Yandang station 10 kV Yan 957 Line 957 Switch	0	Spring not charged reset.

It can be seen that each alarm information is determined by device, protection and protection/switch state. The state composition represents a piece of alarm information as a quadruple m vectorized by device, protection, protection state and switch state.

$$m = (d, p, ps, bs), \tag{11}$$

In the formula, d is the device number, p is the protection number, ps is the protection state and bs is the switch state. The settings of each parameter are as follows:

1. d is the number of the device in the device set and the device set consists of the devices included in all training samples. For example, only for the alarm information samples shown in Table 1, $d \in \{0, 1, 2, 3, 4\}$, d = 0 can be set to mean "Yandang station main transformer fault recorder", d = 1 means "Yandang station 220 kV fault recorder", d = 2 means "Yandang station 110 kV fault recorder ", d = 3 means "Yandang station 10 kV Yan 957 Line " and d = 4 means "Yandang station 10 kV Yan 957 Line 957 Switch";

2. p is the number of the protection in the protection set, and the protection set consists of protections contained in all training samples. For example, only for the alarm information samples shown in Table 1, $p \in \{0, 1, 2, 3, 4\}$, where p = 0 means the recorder starts, p = 1 means protect, p = 2 means overcurrent I stage, p = 3 means recloser and p = 4 means no protect;
3. $ps \in \{0, 1, 2\}$, ps = 0 means no protection signal, ps = 1 means protection action and ps = 2 means protection reset;
4. $bs \in \{0, 1, 2, 3, 4, 5, 6\}$, bs = 0 means no switch signal, bs = 1 means the switch general outlet tripping action, bs = 2 means the switch general outlet tripping reset, bs = 3 means the switch is open, bs = 4 means the switch is closed, bs = 5 means the spring is not charged action and bs = 6 means the spring is not charged to reset.

According to the setting of the above parameters, the entire fault occurrence process can be represented by the alarm information sequence $M = \{m_1, m_2, \cdots, m_n\}$. The vectorized alarm information vector sequence is shown in Formula (12):

$$\begin{aligned} M = \{m_1, m_2, \cdots, m_{13}\} = \{&(0,0,1,0),(1,0,1,0),(2,0,1,0),(3,1,1,0),(3,2,1,0),\\ &(4,4,0,1),(4,0,0,2),(3,1,2,0),(3,2,2,0),(3,3,1,0),(4,4,0,4),(4,4,0,5),\\ &(3,3,2,0),(0,0,2,0),(1,0,2,0),(2,0,2,0),(4,4,0,2),(4,4,0,6)\} \end{aligned} \tag{12}$$

3.2. Design of Fault Diagnosis Model Based on DQN

After converting the alarm information text environment into a vectorized alarm information sequence environment, it is necessary to clearly define the state, action and reward of the DQN model of the deep reinforcement learning agent to complete the communication between the DQN agent and the alarm information sequence environment interaction, and then train it to obtain a deep reinforcement learning agent that can accurately predict the text of the alarm information.

In this paper, the text prediction of alarm information is carried out using "multiple input and single output", that is, the first n alarm information is input to predict the n + 1 alarm information. At this time, the definitions of status, action and reward are as follows:

1. State: The state is the current input obtained by the agent from the environment. For the text prediction of alarm information, the environment is the sequence of alarm information vectors corresponding to the text. Since DQN uses multiple inputs to predict, the state is set to the quadruple sequence corresponding to n alarm information, and its initial state s_0 is:

$$s_0 = (m_1, m_2, \cdots, m_n), \tag{13}$$

When the input s_0 predicts the n + 1th alarm information m_{n+1}, the first $n - 1$ alarm information and the predicted n + 1th alarm information quadruple are combined as the next state, namely:

$$s_1 = (m_2, \cdots, m_n, m_{n+1}), \tag{14}$$

and so on in order to obtain the next state after each action. Assuming that an alarm information sample contains one piece of alarm information, the state set corresponding to the sample is:

$$S = \{s_0, s_1, \cdots, s_{l-n}\}, \tag{15}$$

2. Action: After the agent obtains the current state information from the environment, it needs to select an action from the action space according to a certain strategy to predict the next alarm information. Since a single alarm message consists of device number, protection number, protection state or switch state, the action is designed as a combination of the above four elements, and the action space is the different combinations of device number, protection number, protection state and switch state. The agent achieves state transition by selecting actions from the action space.

$$A = \{(d, p, ps, bs) | d \in (1, t), p \in (1, k), ps \in (0, 2), bs \in (0, 5)\}, \tag{16}$$

Of these, t is the number of devices in the device set, and k is the number of protections in the protection set.

3. Reward: When the agent selects an action according to the strategy, its state transitions. At this point, the action needs to be evaluated through the reward function to get the feedback of the environment on the action. In the text prediction process, the reward function is set by comparing the prediction result with the actual alarm information, so as to minimize the difference and maximize the prediction accuracy.

For the predicted alarm information quadruplet, it is necessary to judge whether each element is the same as the element in the actual quadruple, count the number of different elements, and the design reward values of different sizes accordingly. The reward function is shown in the following formula.

$$reward = \begin{cases} 2 & error = 0 \\ \frac{1}{error} & error \neq 0 \end{cases}, \tag{17}$$

Of these, *error* is the number of different elements in the predicted quadruple and the actual quadruple, and *error* = 0 or 1 or 2 or 3 or 4.

According to the above definitions of states, actions and rewards, the complete interaction process between the DQN agent and a single alarm information text environments can be obtained, as shown in Figure 2. The alarm information vector sequence in the figure is $M = \{m_1, m_2, m_3, \cdots, m_n\}$, each alarm information is represented by a color, input three alarm information for prediction, and $m'_4, m'_5, \cdots, m'_{n-1}, m'_n$ represent the corresponding prediction result. If the prediction result does not match the actual alarm information, it will be distinguished by different colors.

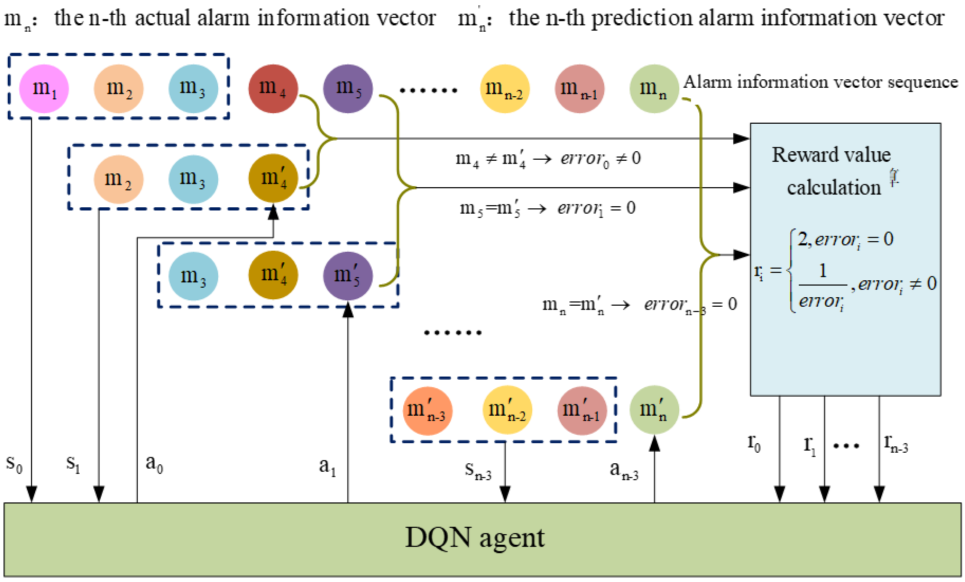

m_n: the n-th actual alarm information vector
m'_n: the n-th prediction alarm information vector
m1 m2 m3 m4 m5 ...... mn-2 mn-1 mn
Alarm information vector sequence
m2 m3 m'4
m3 m'4 m'5
m'n-3 m'n-2 m'n-1 m'n
m4 ≠ m'4 → error0 ≠ 0
m5=m'5 → error1 = 0
mn=m'n → errorn-3 = 0
......
Reward value calculation
ri = {2, errori = 0; 1/errori, errori ≠ 0
s0 s1 a0 a1 sn-3 an-3
r0 r1 ... rn-3
DQN agent

Figure 2. Interaction between DQN agent and alarm information.

3.3. Fault Diagnosis Process Based on the DQN

Based on the above design of the DQN model, this paper trains the DQN prediction model for the set of non-switch refusal alarm information of different fault types to achieve the accurate prediction of the removal process of various faults and identify the refusal switch of the switch refusal sample accordingly. Figure 3 is the flow chart of using the DQN model to identify the refusal switch of the switch refusal fault sample, which is divided into two parts: agent training and testing.

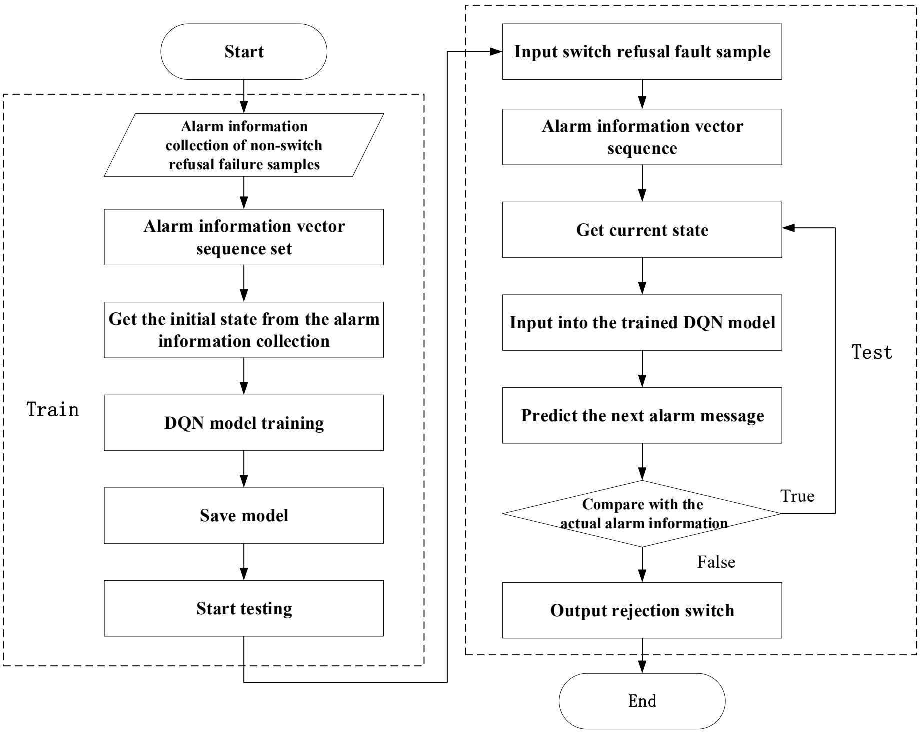

Start
Alarm information collection of non-switch refusal failure samples
Alarm information vector sequence set
Get the initial state from the alarm information collection
Train
DQN model training
Save model
Start testing
Input switch refusal fault sample
Alarm information vector sequence
Get current state
Input into the trained DQN model
Test
Predict the next alarm message
Compare with the actual alarm information
True
False
Output rejection switch
End

Figure 3. Flowchart of grid fault diagnosis based on DQN.

The DQN training process is shown in Figure 4. First, samples of different fault types, such as line faults, bus faults and transformer faults, are converted into corresponding alarm information vector sequences. The DQN model obtains the state s from it, takes action a and outputs the predicted next alarm information m' and the reward r and the next state s' will be fed back and at the same time, it will be put (s, a, r, s') into the experience playback pool. After n-step prediction, judge the prediction accuracy rate of n alarm information. If the accuracy rate is less than 50%, continue to train the first n alarm information. If it is greater than 50%, obtain a new state from the alarm information set, and so on. The training of the DQN model is carried out by randomly sampling samples of a certain batch size from the experience pool. After multiple training optimizations, the trained model is saved for use in online fault diagnosis.

During the test, select a switch refusal fault sample, also perform vectorization processing on it, obtain the initial state and input it into the saved DQN model, and compare it with the actual alarm information text for each step of prediction, then output the refusal through the comparison switch.

The specific steps of grid fault diagnosis based on DQN are as follows:

1. Prepare historical alarm information samples of line, bus and transformer failures without the switch and refusal to act, perform de-sequencing processing and vectorization processing on the samples, and express the alarm information text as an alarm information vector sequence;
2. Obtain the initial state of the alarm information text and input it into the DQN, and train the DQN agent through continuous interaction with the environment of the alarm information vector sequence;
3. Set the number of training rounds (episodes), and repeat Step 2 until the loss function is the smallest;
4. After the DQN training is completed, vectorize the tested switch refusal fault samples to obtain the initial state and input it into the DQN model for prediction;
5. For each prediction step, compare the predicted result with the actual alarm information, until the comparison result is different, end the prediction and output the switch that refuses to act.

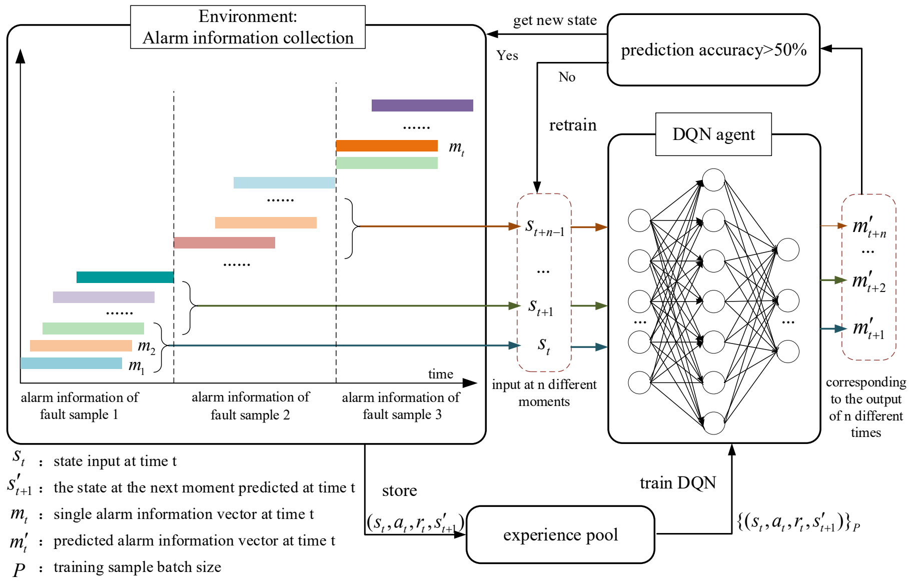

Environment:
Alarm information collection
get new state
Yes
No
prediction accuracy>50%
retrain
DQN agent
m_t
m_2
m_1
......
time
alarm information of fault sample 1
alarm information of fault sample 2
alarm information of fault sample 3
s_{t+n-1}
...
s_{t+1}
s_t
input at n different moments
m'_{t+n}
m'_{t+2}
m'_{t+1}
corresponding to the output of n different times
s_t : state input at time t
s'_{t+1} : the state at the next moment predicted at time t
m_t : single alarm information vector at time t
m'_t : predicted alarm information vector at time t
P : training sample batch size
store
$(s_t, a_t, r_t, s'_{t+1})$
experience pool
train DQN
$\{(s_t, a_t, r_t, s'_{t+1})\}_P$

Figure 4. DQN training process.

4. Example Verification

4.1. Sample Data

In order to reflect the prediction ability of the DQN model for different power grid data, this paper adopts the TS2000 fault samples and the alarm information set of the actual power grid to train the model. The composition of the two types of training samples is as follows:

1. The TS2000 fault samples are non-switch refusal samples with simple faults on lines, busbars and transformers. Since the deep reinforcement learning agent can continuously repeat training by setting multiple episodes, in order to reduce the training time and speed up the convergence speed, you can use the A small number of fault samples are used for training, and the composition of the training samples is shown in Table 2;
2. The actual alarm information set is the alarm information within a period of time randomly intercepted according to different time window sizes from the historical monitoring alarm information. As shown in Table 3, the alarm information sets within 10 min, 20 min, 30 min, 50 min and 1 h were selected as training samples from a certain moment of historical monitoring and only the first 10 min in the alarm information set A fault event occurs within the time period, and the alarm information in the rest of the time period is non-fault information.

Table 2. TS2000 simulation system training sample composition.

	Line Fault	**Busbar Fault**	**Transformer Fault**
Number of samples	10	4	10

Table 3. Sample composition of actual alarm information set.

	10 min	**20 min**	**30 min**	**50 min**	**1 h**
Number of alarm information	46	65	90	198	252

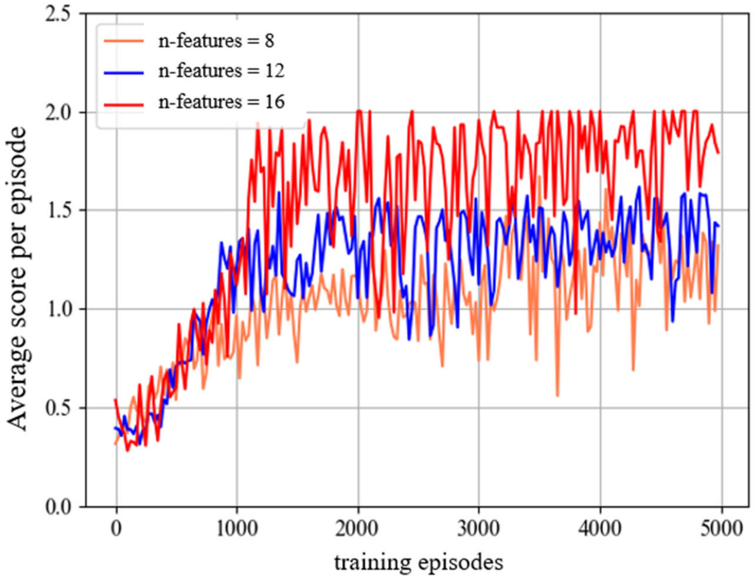
4.2. Evaluation Index

1. Average reward: The agent trains multiple episodes, executes a certain number of steps in each episode, records all the reward values obtained and averages them to obtain the average reward value for each round;

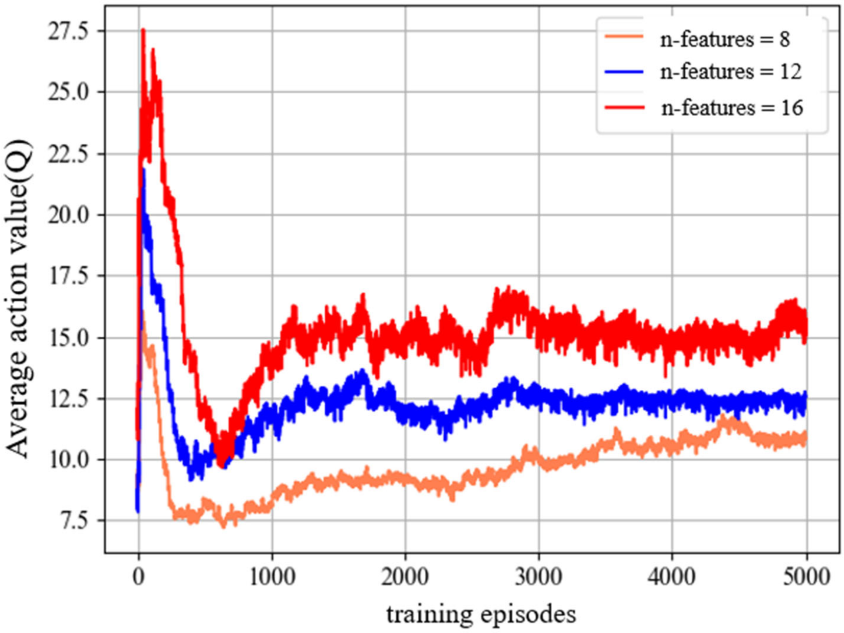
2. Average Q value: After determining the set of state–action pairs, the agent tracks the maximum predicted Q value corresponding to these states at each step in each training round and takes the average to obtain the average Q value of each round.

4.3. Algorithm Parameter Settings

The DQN model predicts in the way of "multiple input and single output". The size of its state dimension (n_features) changes with the number of input alarm information. In this paper, 2, 3 and 4 alarm information are input, respectively, judging the average reward value and average Q value under different state dimensions (n_features = 8, 12, 16) and the experimental results are shown in Figure 5.



(**a**)



(**b**)

Figure 5. Average reward and average Q value under different state dimensions: (**a**) Average reward; (**b**) Average Q value.

It can be seen that the average reward value and average Q value increase with the increase of the state dimension. Since the purpose of the alarm information prediction in this paper is to compare with the rejection samples, the amount of input alarm information should not be too large, so the state dimension is determined to be 16, the rest of the algorithm parameters have been trained and tested many times and the parameter values are determined according to the prediction accuracy, as shown in Table 4.

Table 4. DQN parameter table.

DQN Parameters	
learning_rate	0.01
Gamma	0.9
replace_target_iter	200
greedy_increment	0.001
ε	0.95
step	5
episode	5000
batch_size	128

4.4. Analysis of Results

According to the selected alarm information samples and parameters, the DQN model is trained. As shown in Tables 5 and 6, the prediction results of the models for the simulated fault samples and the actual fault samples, respectively. Among them, the key information refers to the alarm information of equipment protection and circuit breaker action.

Table 5. Sample diagnosis results.

	Line Fault	**Busbar Fault**	**Transformer Fault**
Total number of alarm information	44	145	56
Number of key information	16	56	21
Predict the correct number of key information	16	46	20
Key information prediction accuracy	100%	82.1%	95.2%
Overall prediction accuracy	93.2%	58.3%	88.5%

Table 6. Sample diagnosis results.

	10 min	**20 min**	**30 min**	**50 min**	**1 h**
Number of alarm information	46	65	90	198	252
Predict the correct number of alarm information	40	50	65	132	120
Prediction accuracy	86.9%	76.9%	72.2%	66.7%	47.6%

It can be seen from Table 5 above that, in terms of overall alarm information prediction, the prediction accuracy of line fault samples is the highest, followed by transformer faults, and the lowest accuracy of busbar faults. This is because there are fewer electrical devices involved in the alarm message text when the line is faulty, its logical relationship is relatively simple and the DQN model can easily learn the relationship. In the power grid topology, transformers and busbars are closely connected with other devices. When the transformer or busbar fails, the protection and circuit breakers of other devices (except the faulty device) will also act accordingly. The logic and relationship between the devices are more complex and it is difficult for the DQN model to accurately learn all the alarm information, but it can achieve a good prediction effect in the prediction of key information.

It can be seen from Table 6, with the continuous expansion of the alarm information time window, the number of alarm information continues to increase and the number of model training rounds also increases, but the prediction accuracy decreases with the increase of the number of alarm information. The reasons are as follows:

1. The more alarm information, the more devices and protections, the more complex the topology connection and action logic relationship, the more difficult the model learning and the lower the prediction accuracy;
2. There are few fault events in the actual alarm information set. For example, the alarm information set used for training in this paper contains only one fault event, and the proportion of fault alarm information is small, most of which is non-fault alarm information. The logical relationship with the circuit breaker action is not clear, and the model is not easy to learn.

In order to better evaluate the text prediction ability of the DQN model, this paper evaluates the model according to the sample complexity and selects three types of samples: simple fault, switch refusal fault and developmental fault and input them into the model for training and testing. The average reward and sample prediction accuracy are used as evaluation metrics. The average reward during training and the average prediction accuracy of test samples are shown in Figures 6 and 7, respectively.

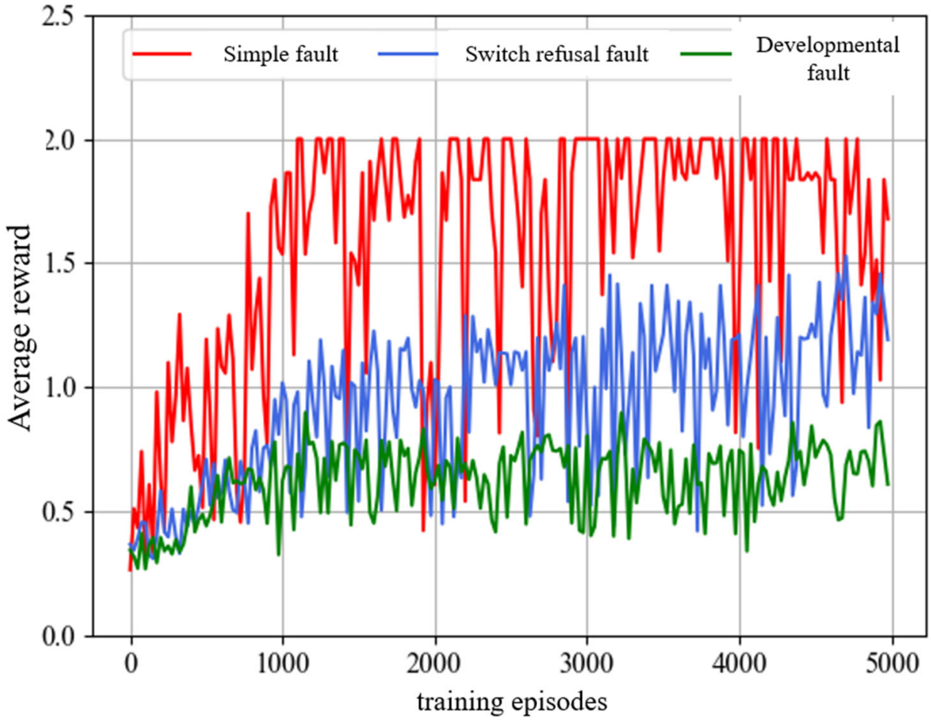


Figure 6. Average reward.

It can be seen that the model prediction accuracy is greatly affected by the sample complexity. The higher the sample complexity, the more complex the logical relationship between equipment, protection and circuit breakers, the more difficult the model is to learn and the lower the prediction accuracy.

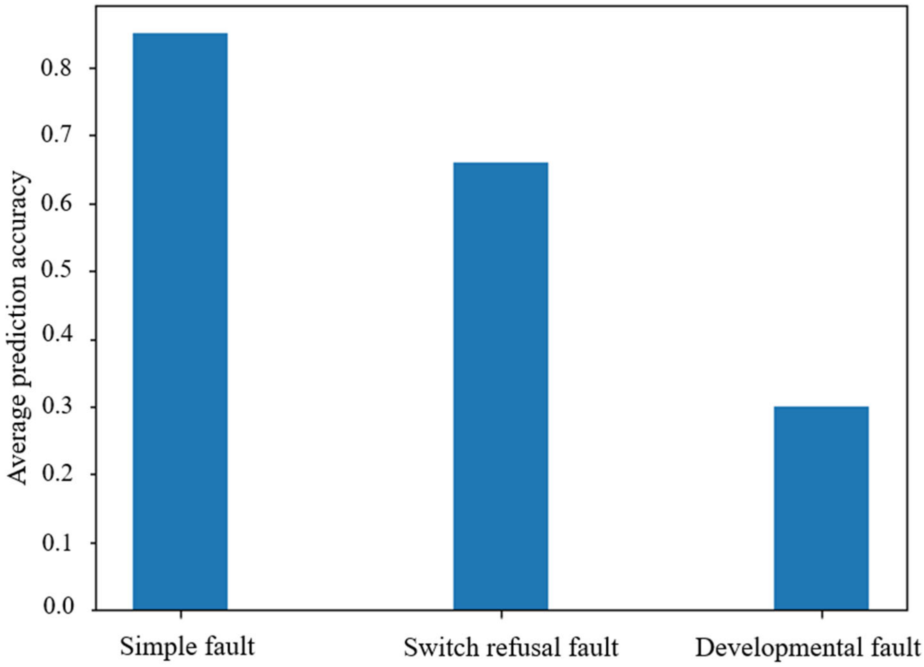


Figure 7. Average prediction accuracy.

4.5. Case Analysis

This section selects the switch refusal samples generated by the TS2000 simulation system for testing. In the TS2000 simulation system, Yandang Station 10 kV Yan 957 line is set to have an AB phase-to-phase short-circuit instantaneous fault and the 957 switch refuses to move.

Using the above fault diagnosis method, the first four alarm messages are used as input to diagnose the fault case. The results are shown in Table 7. It can be seen that when the DQN model predicts the alarm information numbered 6, the prediction result is different from the actual one. The alarm information is different, so it is recognized that the "Yandang Station 10 kV Yan 957 Line 957 switch" refuses to move.

Table 7. Diagnosis results.

Number	Prediction Result	Actual Alarm Information	Diagnosis Results
5	10 kV Yan 957 line overcurrent section I	10 kV Yan 957 line overcurrent section I	None
6	10 kV Yan 957 line 957 switch general outlet tripping action	10 kV Yan 957 line 957 switch general outlet tripping action	None
7	10 kV Yan 957 line 957 switch open	10 kV Yan 957 line protection	10 kV Yan 957 Line 957 switch refuses to move
...	...	...	...

5. Conclusions

In view of the intelligent demand of power grid fault diagnosis, this paper proposes a power grid fault diagnosis method based on deep reinforcement learning, which realizes fault diagnosis based on power grid alarm information, which is of great significance for improving the level of power grid intelligence:

1. This paper proposes a DQN-based power grid fault diagnosis method. Aiming at the problem that it is difficult to determine the refusal switch under the no network topology, a deep reinforcement learning fault diagnosis method oriented to alarm

information is designed and a fault diagnosis model based on DQN is established. Through the learning and prediction of the implicit logical relationship between equipment, protection and circuit breaker actions by the reinforcement learning agent, the normal fault removal process when the fault event occurs is obtained, and then compared with the fault removal process of the switch refusal sample, which identifies the refusal switch. The experimental results show that the method can learn the logical relationship between the equipment, protection and circuit breaker actions contained in the alarm information without analyzing the network topology structure and then identify the faulty equipment, which is feasible and effective;

2. The fault diagnosis method based on DQN proposed in this paper is greatly affected by the complexity of the samples, the prediction accuracy of complex fault samples is low and the refusal switch of complex faults may not be correctly identified. Therefore, the follow-up work should further improve the model. It can improve the diagnosis model, improve the fault tolerance rate of the model for complex fault samples, and then can correctly diagnose the refusal switch of complex fault samples.

Author Contributions: Conceptualization, X.Z. and H.Z.; Methodology, Z.W.; Software, B.L.; Validation, Z.W., Z.Z. and M.D.; Writing—original draft preparation, Z.W. and H.Z.; Writing—review and editing, Z.W.; Visualization, Z.W.; Supervision, X.Z.; funding acquisition, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 51877079.

Data Availability Statement: Not applicable.

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

References

1. Wang, S.P.; Zhao, D.M. A Hierarchical Power Grid Fault Diagnosis Method Using Multi-Source Information. *IEEE Trans. Smart Grid* **2020**, *11*, 2067–2079. [CrossRef]
2. Fukui, C.; Kawakami, J. An Expert System for Fault Section Estimation Using Information from Protective Relays and Circuit Breakers. *IEEE Trans. Power Deliv.* **1986**, *6*, 83–90. [CrossRef]
3. Yan, W.; Lanqin, G. Bayesian Network Based Fault Section Estimation in Power Systems. In Proceedings of the TENCON 2006—2006 IEEE Region 10 Conference, Hong Kong, China, 14–17 November 2006, pp. 1–4. [CrossRef]
4. Chang, C.S.; Chen, J.M.; Liew, A.C.; Srinivasan, D.; Wen, F.S. Power system fault diagnosis using fuzzy sets for uncertainties processing. In Proceedings of the International Conference on Intelligent Systems Applications to Power Systems, Orlando, FL, USA, 28 January 1996–2 February 1996. [CrossRef]
5. Wen, F.S.; Ledwich, G.; Liao, Z.W.; He, X.; Liang, J. An analytic model for fault diagnosis in power systems considering malfuctions of protective relays and circuit breakers. *IEEE Trans. Power Deliv.* **2010**, *25*, 1393–1401.
6. Pourbabaee, B.; Roshtkhari, M.J.; Khorasani, K. Deep convolutional neural networks and learning ECG features for screening paroxysmal atrial fibrillation patients. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *48*, 2095–2104. [CrossRef]
7. Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Hassabis, D. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* **2018**, *362*, 1140–1144. [CrossRef] [PubMed]
8. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Hassabis, D. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489. [CrossRef] [PubMed]
9. Zhang, D.; Han, X.; Deng, C. Review on the research and practice of deep learning and reinforcement learning in smart grids. *CSEE J. Power Energy Syst.* **2018**, *4*, 362–370. [CrossRef]
10. Wang, H.; Yang, S.D.; Zhou, B.W. Fault Diagnosis of Multi-terminal HVDC Transmission Line Based on Parallel Convolutional Neural Network. *Autom. Electr. Power Syst.* **2020**, *44*, 84–92.
11. Gou, B.; Xu, Y.; Xia, Y.; Wilson, G.; Liu, S. An intelligent time-adaptive data-driven method for sensor fault diagnosis in induction motor drive system. *IEEE Trans. Ind. Electron.* **2018**, *66*, 9817–9827. [CrossRef]
12. Wang, C.; Jiang, Q.; Tang, Y.; Zhu, B.; Xiang, Z.; Tang, J. Fault diagnosis of power dispatching based on alarm signal text mining. *Electr. Power Autom. Equip.* **2019**, *39*, 126–132.
13. Zhao, J.; Wei, Y.; Liu, J.; Wei, S.; Wang, Z.; Ke, Y.; Deng, X. Power Grid Fault Diagnosis Based on Fault Information Coding and Fusion Method. In Proceedings of the 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2), Beijing, China, 20–22 October 2018; pp. 1–6.
14. Sun, G.; Shen, P.; Zhao, Y.; Zhu, H.; Ding, X. Intelligent recognition of power grid monitoring alarm event combining knowledge base and deep learning. *Electr. Power Autom. Equip.* **2020**, *40*, 40–47.

15. Guo, R.; Yang, Q.; Liu, S.H.L.; Wei, Y.X.; Huang, X.H. Construction and Application of Power Grid Fault Handing Knowledge Graph. *Power Syst. Technol.* **2021**, *45*, 2092–2100.
16. Huang, H.; Lv, Y.; Dong, R.; Xue, L.; Shen, Z.; Liu, H.; Hu, E. Power Grid Fault Diagnosis Based on Random Forest. In Proceedings of the 2021 IEEE 5th Conference on Energy Internet and Energy System Integration (EI2), Taiyuan, China, 22–24 October 2021; pp. 3143–3148.
17. Zhang, X.; Zhang, H.; Zhang, D. Power grid fault diagnosis based on a deep pyramid convolutional neural network. *CSEE J. Power Energy Syst. to be published*. 2022.
18. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with Deep Reinforcement Learning. *arXiv* **2013**, arXiv:1312.5602.
19. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. A brief survey of deep reinforcement learning. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. [CrossRef]
20. Liu, Q.; Zhai, J.W.; Zhang, Z.C.; Zhong, S.; Xu, J. A survey on deep reinforcement learning. *Chin. J. Comput.* **2018**, *41*, 1–27.
21. Hou, J.; Li, H.; Hu, J.; Zhao, C.; Guo, Y.; Li, S.; Pan, Q. A review of the applications and hotspots of reinforcement learning. In Proceedings of the 2017 IEEE International Conference on Unmanned Systems (ICUS), Beijing, China, 27–29 October 2017; pp. 506–511.
22. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
23. Hansen, S. Using deep Q-learning to control optimization hyperparame ters. *arXiv* **2016**, arXiv:1602.04062.
24. Zhang, Z.D.; Qiu, C.M.; Zhang, D.X.; Xu, S.W.; He, X. A coordinated control method for hybrid energy storage system in microgrid based on deep reinforcement learning. *Power Syst. Prot. Control* **2019**, *43*, 1914–1921.
25. Liu, W.; Zhang, D.; Wang, X.; Hou, J.X.; Liu, L.P. A decision making strategy for generating unit tripping under emergency circumstances. *Proc. CSEE* **2018**, *38*, 109–119.
26. Xi, L.; Lei, X.I.; Lu, Y.U.; Yimu, F.U. Automatic generation control based on deep reinforcement learning with exploration awareness. *Proc. CSEE* **2019**, *39*, 4150–4162.