Preventive Control Policy Construction in Active Distribution Network of Cyber-Physical System with Reinforcement Learning

Pengpeng Sun *, Yunwei Dong, Sen Yuan and Chong Wang

School of Computer Science and Engineering, Northwestern Polytechnical University, Xi’an 710072, China; yunweidong@nwpu.edu.cn (Y.D.); yuansen@mail.nwpu.edu.cn (S.Y.); wangchonw@mail.nwpu.edu.cn (C.W.)
* Correspondence: jiea489756@163.com; Tel.: +86-135-7248-4015

Abstract: Once an active distribution network of a cyber-physical system is in alert state, it is vulnerable to cross-domain cascading failures. It is necessary to transit the state of an active distribution network of cyber-physical system from an alert state to a normal state using a preventive control policy against cross-domain cascading failures. In fact, it is difficult to construct and analyze a preventive control policy via theoretical analysis methods or physical experimental methods. The theoretical analysis methods may not be accurate due to approximated models, and the physical experimental methods are expensive and time consuming for building prototypes. This paper presents a preventive control policy construction method based on a deep deterministic policy gradient idea (shorted as PCMD) to generate and optimize a preventive control policy with Artificial Intelligence (AI) technologies. It adopts the reinforcement learning technique to make full use of the available historical data to overcome the problems of high cost and low accuracy. Firstly, a preventive control model is designed based on the finite automaton theory, which can guide the data collection and learning policy selection. The control model considers the voltage stability, frequency stability, current overload prevention, and the control cost reduction as a feedback variable, without the specific power flow equations and differential equations. Then, after enough training, a local optimal preventive control policy can be constructed under the comparability condition among a fitted action-value function and a fitted policy function. The constructed preventive control policy contains some control actions to achieve a low cost and in accord with the principle of shortening a cross-domain cascading failures propagation sequence as far as possible. The PCMD is more flexible and closer to reality than the theoretical analysis methods and has a lower cost than the physical experimental methods. To evaluate the performance of the proposed method, an experimental case study, China Electric Power Research-Cyber-Physical System (shorted as CEPR-CPS), which comes from China Electric Power Research Institute, is carried out. The result shows that the effectiveness of preventive control policy construction with the PCMD is better than most current methods, such as the multi-agent method in terms of reducing the number of failure nodes and avoiding the state space explosion.

Keywords: active distribution network; preventive control policy; reinforcement learning; cross-domain cascading failures; cyber-physical system

1. Introduction

Active distribution network (ADN) is a typical cyber-physical system (CPS), which consists of a wide variety of distributed generators connected to a power network (PN) and a communication network (CN) [1]. An ADN at every moment of time can be abstracted as a directed graph, where equipment is abstracted as nodes, and power lines or communication connections among equipment are abstracted as edges [1], and the directed graph at different time points may be different due to changes in edges or nodes of an ADN. There are five types of nodes in the PN according to the role of nodes: a power generation node type \( V_{P1} \), a substation node type \( V_{P2} \), a distribution node type \( V_{P3} \), a load node type \( V_{P4} \),
and an external node type $V_{P5}$ (the energy from power transmission lines). Similarly, nodes in the CN are categorized into an information relay node class $V_{C1}$, a sensor node class $V_{C2}$, an actuator node class $V_{C3}$, and a control node class $V_{C4}$. Nodes in the PN are mutually interdependent with nodes in the CN. Specifically, a substation node $n_{P2} \in V_{P2}$ in the PN supplies power to nodes in the CN, a sensor node $n_{C2} \in V_{C2}$ in the CN collects the information from nodes in the PN, and an actuator node $n_{C3} \in V_{C3}$ drives the behaviors of nodes in the PN. There is a risk of the potential cross-domain cascading failures (CCF) spreading alternatively across the PN/CN due to the interdependence between the PN and CN in an ADN. The propagation process of CCF can be divided into two stages [2]. The first stage is a slowly changing process lasting for several minutes. The second stage is an avalanche-collapse process, which could lead to a large area of power outages and cause large economic losses. Hence, it is essential to prevent the propagation of CCF so as to alleviate the side effect of this event and improve the level of the safe and stable operation of an ADN.

A directed graph at every time point could be described as a state of an ADN. These states can be divided into steady states and transient states among ADN operations. A steady state means that an ADN is in stable operation, and a transient state is an intermediate state between steady states. During the propagation process of CCF, an ADN goes through a series of transient states from one steady state to another steady state. If the CCF is not prevented when the CCF is propagating, an ADN must be transferred into the next transient state, and it must stop at a final steady state, and the ADN may collapse. Hence, once the propagation process of CCF is initiated, it is better to use the preventive control to block the propagation process of CCF.

The preventive control process against CCF in an ADN is shown in Figure 1, there are five steps in each routine loop operation. They are state perception, failure diagnosis, decision, preventive control policies, and control action. The state perception gathers the real-time data (such as the voltage and current of nodes) of an ADN. The failure diagnosis detects failure symptoms and predicts failures that could be fired through some propagation paths of the CCF. The goal of decision is to select an optimal preventive control policy (PCP) to prevent potential accidents and the CCF propagation in an ADN. The preventive control policies (PCPs) are some strategies that could be adopted for decision step, and they are constructed in advance according to safety and stability requirement specifications of an ADN. The control action is the last routine step, which just carries out a sequence of operations on an ADN according to decisions and affects the state transition process of an ADN.

![Figure 1. The preventive control process against cross-domain cascading failures (CCF) in an active distribution network (ADN).](image-url)

In a process of the preventive control routine, a PCP is very important, it provides roadmaps of operations against CCF and determines the performance of the preventive
control routine. A strategy of PCPs needs to minimize the length of CCF sequences so as to block the propagation of CCF in an ADN by exerting the control actions with low cost. The selected PCP needs meet some prevention goals optimally, such as preventing voltage collapses, transient instabilities, etc. Theoretical analysis methods [3–8] and physical experimental methods [9–11] are two typical solutions for a PCP construction. The theoretical analysis method needs to obtain the approximated models of controlled plants, and the experimental method needs to build physical prototypes and owns a heavy cost. So, traditional methods for constructing PCPs always meet some challenges, such as high costs, long durations, and inaccuracies.

A data-driven approach can make full use of a lot of available historical data to overcome the shortcomings of the existing methods, generating and optimizing a PCP through AI technologies with lower cost. According to the goals of preventive controls against failures, there are many AI technologies that could be adopted for constructing PCPs. Chengxi Liu et al. propose a dynamic security assessment method and obtain a PCP based on a decision tree idea to ensure the dynamic security [12]. In order to improve the voltage stability margin of power systems, Mauricio C. Passaro et al. propose a preventive control method for rescheduling power generation based on a neural sensitivity model. The time series data samples from time domain simulations and the dynamic behavior information of the system are used, and the sensitivity is used to select the most effective set of generators to improve the security of the power system [13]. C. Fatih Kucuktezcan et al. use population optimization techniques to construct a PCP. They reduce the search space of each algorithm according to the size of an objective function and make multiple optimization algorithms run continuously, so as to improve the transient security of a power CPS [14]. Soni B P et al. use a wide area measurement system (WAMS) and phasor measurement devices to conduct real-time transient stability assessment. Specifically, the support vector machine (SVM) based on least squares is used to identify the steady state of the power systems in real time, and then, the appropriate dispatching generators are selected for preventive control to ensure transient stability [15]. Kou P et al. use the algorithm of deep deterministic policy gradient with a safety exploration layer for preventive control to ensure that the node voltages of active distribution network are within the limit [16]. However, these studies focus on the construction of PCPs for only a single failure in CPS, so these constructed PCPs cannot prevent the occurrence of cascading failures. In order to prevent the occurrence of cascading failures, researchers construct many PCPs. Rabie Belkacemi et al. use a distributed adaptive multi-agent algorithm to get a PCP. The PCP could block the propagation paths of cascading failures by dispatching the power of generators based on N-1 criterion [17]. Sina Zarrabian et al. use neural network techniques to construct a PCP to prevent the propagation of cascading failures [18], and they also use the reinforcement learning method based on Q learning to obtain a PCP to prevent cascading failures [19]. Mojtaba Khederzadeh and Arash Beiranvand use a strategy of specific thresholds to evaluate the lines with high vulnerabilities and then obtain a PCP based on a genetic algorithm. The PCP eliminates the overload of high vulnerability lines through load shedding, so as to prevent the spread of cascading failures in a CPS [20]. Dutta O et al. develop a distributed management system using adaptive critical design based on adaptive dynamic programming. The system can flexibly take preventive actions and corrective actions to deal with the thermal overload of lines in active distribution network [21]. The existing research work about data-driven methods only focuses on a single PN and is not studied from the perspective of the interdependence between PN and CN. Since the CCF propagates between the PN and CN alternatively and the existing data-driven methods for PCPs construction only collect the observation data from the PN, which are not applicable for preventing the CCF due to insufficient data. In order to prevent the propagation of CCF in an ADN, not only the measured data from the PN but also the measured data from the CN should be collected.

It is difficult to establish a collaborative simulation of the physical process and the computational process in an ADN, such as digital twin. However, reinforcement learning
techniques can analyze and summarize the interaction between the physical process and the computational process of an ADN by using the existing empirical data and play the role of simulation and experiment. Using reinforcement learning techniques, there are three advantages for constructing a PCP: less time, lower cost, and higher accuracy. Therefore, a PCP construction method based on deep deterministic policy gradient idea (shorted as PCMD) is proposed to prevent the propagation of CCF in an ADN, and this method needs to collect the data about voltage stability, frequency stability, current overload prevention, robustness for CCF, and control cost. However, there are some challenges during the construction process. That is, how to choose the concrete objectives and weight them into the reward function; how to adjust parameters so as to ensure that the construction process of PCMD converges and a PCP against CCF exists; how to ensure that the construction process converges to a local optimal solution; how to verify the effectiveness of the proposed PCMD.

Some contributions are concluded as follows. (1) A modeling method is proposed for the preventive control using the finite automaton theory. The preventive control model describes the effect of CCF in an ADN after the intervention of control actions and can guide the gather of required trajectory dataset and the construction of a PCP. (2) PCMD for PCP construction is presented, and it guarantees the voltage stability, the frequency stability, the current overload prevention, and the improved robustness against CCF of an ADN. The constructed PCP can generate control actions with low cost based on the principle of shortening a CCF propagation sequence as far as possible. (3) An experimental China Electric Power Research-Cyber-Physical System (CEPR-CPS) case from China Electric Power Research Institute is carried out to validate the performance of the proposed method. The result shows that the effectiveness of the PCP constructed from the PCMD is better than that with others, such as a multi-agent method in terms of reducing the number of failure nodes and avoiding the state space explosion.

The paper is organized as follows. A preventive control model is given in Section 2. In Section 3, the specific construction method is presented. Section 4 gives a case provided by China Electric Power Research Institute to validate the effectiveness of the construction method. Section 5 outlines the discussion.

2. Preventive Control Model

In order to obtain the learning dataset for PCP construction, a preventive control model is constructed to describe the steady state transition process with external interventions against CCF in an ADN. The model can describe transition relationships among the steady states after applying preventive actions to block the propagation of CCF in an ADN, and CCF sequences should be as short as possible. For example, there are three steady states (noted as $g_0$, $g_1$, and $g_2$), as shown in Figure 2. In a steady state $g_0$, substation nodes (Bus\textsubscript{1} and Bus\textsubscript{3}) supply power to all connected nodes of the CN together, and Bus\textsubscript{1} is a backup power to enable all alternative power edges when primary power Bus\textsubscript{3} fails with disconnected edges to the nodes in the CN; in this case, the steady state $g_0$ is transited to another steady state $g_1$ after nodes (PG\textsubscript{1}, Bus\textsubscript{3}, and EN\textsubscript{1}) fail. Similarly, when the power generation node PG\textsubscript{1}, the substation node Bus\textsubscript{3} and the external node EN\textsubscript{1} are repaired via the external interventions, and all disconnected edges have connected again, but they are inactive, the ADN transfers a steady state $g_2$ from state $g_1$. This transfer process can be described by a finite state machine.
Definition 1. (Preventive Control Model): A preventive control model against CCF is formally described by an extended finite automaton as a six tuple \( U = (G, S, A, F, \text{Prb}, r, g_0) \).

(1) \( G = (V, E) \) represents a finite set of directed graphs, which is a steady state under the stable operation of an ADN. The steady state \( g_i = (V_i, E_i) \in G(i \geq 0) \) represents a directed graph in an ADN at discrete time \( i \times \Delta t \), and \( \Delta t \) denotes the sample interval. \( V_i \) represents the node set in the steady state \( g_i \) of an ADN. \( g_0 = (V_0, E_0) \) is the initial steady state of an ADN, which is in normal work condition as shown in Figure 2a. A node \( n_i \in V_i \) is represented by a feature vector \( (a_{i1}, a_{i2}, \ldots, a_{INOF})^T \), NOF is the number of features of the node \( n_i \). The set \( E_i \) represents connected relations between a pair of nodes among the PN and CN under the steady state \( g_i \) of an ADN at discrete time \( i \times \Delta t \). For example, the features of a power generation node \( n_i = PG_1 \) has 6 feature attributes, which include voltage \( a_{i1} = v_i \), current \( a_{i2} = I_i \), active power \( a_{i3} = P_i \), reactive power \( a_{i4} = Q_i \), power adjustment \( a_{i5} = \Delta P_i \), frequency \( a_{i6} = f_i \).

(2) \( S \) represents a CCF set, and its element \( s_i = X_{i0} \rightarrow X_{i1} \rightarrow \ldots \rightarrow X_{in} \) includes a cross-domain cascading failures propagation sequence from an initial failure \( X_{i0} \) in a source node to a failure \( X_{in} \) in a sink node of a state \( g_i = (V_i, E_i) \). If two sequences with different orders have the same length and elements, then they are considered to be the same. The failure \( X_{ij} \) is represented by a vector \( (x_{ij1}, x_{ij2}, \ldots, x_{ijNOF})^T \), which presents the operation states of an ADN, where NOF is the number of nodes of an ADN, and \( x_{ijk} \) is the operation state of the kth node \( k \in \{1, 2, \ldots, NOF\} \) of an ADN, and when \( x_{ijk} = 0 \), it is noted that the operation state of the kth node is normal, otherwise it is faulty when \( x_{ijk} = 1 \). For example, a failure \( X_{i0} \) in a source node Bus3 of an ADN occurs in a steady state \( g_0 = (V_0, E_0) \in G \), as shown in Figure 2a, its operation states are described as \( X_{i0} = (x_{PG1}, x_{Bus3}, x_{EN1}, x_{EN1}, x_{Bus1}, x_{Bus2}, x_{PG2}, x_{SN1}, x_{IR1}, x_{CN1}, x_{IR2}, x_{AN1}, x_{CN2}, x_{IR3}, x_{SN2}, x_{AN2})^T = (0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)^T \). During the CCF propagation from the failure in a source node Bus3 to a failure \( X_{in} \) in a sink node, there are three failures occurring, and \( s_0 = X_{i0} \rightarrow X_{i1} \rightarrow X_{i3} = (0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)^T \rightarrow (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)^T \rightarrow \ldots \).
When a PCP is enabled to prevent the CCF propagation sequence $s_0$, the ADN is transferred from $g_0$ to $g_1$, as shown in Figure 2. In the next state $g_1$, there could be CCF propagation sequences from other source nodes of an ADN.

(3) $A \subseteq \mathbb{R}^n$ represents a set of control actions, which is composed of some control actions following a PCP to prevent the CCF propagation. A PCP is a mapping $(l_\perp : S \rightarrow A)$ from the CCF set to the control action set. For example, in order to prevent the instability of an ADN caused by the failure of the source node Bus$_3$, a backup power Bus$_1$ is activated to enable alternative power edges to the nodes in the CN primarily powered by Bus$_3$ and all connections from nodes (PG$_1$, Bus$_3$ and EN$_1$) to other devices are cut off, the topology of the ADN is changed, as shown in Figure 2b.

(4) $F : G \times S \rightarrow G$ represents a steady state transition function between two steady states of an ADN. For example, the CCF propagation sequence $s_0$ of a state $g_0 = \langle V_0, E_0 \rangle \in G$ in an ADN is blocked by a control action $l(s_0) \in A$, so that the ADN can quickly reach a new steady state $g_1 = \langle V_1, E_1 \rangle \in G$, it is expressed as $\langle V_0, E_0 \rangle \xrightarrow{s_0(l_\perp)} \langle V_1, E_1 \rangle$.

(5) $\Prb : F \rightarrow \mathbb{R}$ represents a state transition probability function under a certain action during the propagation process of CCF, $\mathbb{R}$ represents the set of real numbers. The propagation process of CCF has the Markov property [22], and this process is described as a Markov stochastic process. So, the state transition probability function is only related with the current state and action.

(6) $r : F \rightarrow \mathbb{R}$ is an immediate reward function after taking a certain action on a transition.

The preventive control model is a finite automaton, which can be described as a transition system. The state transition of an ADN shown in Figure 3 can be described via a finite automaton, which corresponds to Figure 2.

![Figure 3. A finite automaton preventive control model of an ADN.](image)

3. Preventive Control Policy Construction

In the preventive control model, a PCP $(l_\perp : S \rightarrow A)$ against CCF is described. Theoretical analysis methods and physical experimental methods are used to construct this mapping $(l_\perp : S \rightarrow A)$. However, the theoretical analysis methods may not be accurate due to approximated models, and the physical experimental methods are expensive and time consuming for building prototypes. A PCP construction method based on deep deterministic policy gradient idea (shorted as PCMD) is proposed to obtain the mapping $(l_\perp : S \rightarrow A)$. In the PCMD, two problems—which are the convergence and optimization conditions of the construction process—need to be solved. The specific construction method, solutions to two problems, and an algorithm implementation are explained later.

3.1. Construction Method

The PCMD is divided into three steps, which are data collection, failure diagnosis, and preventive control policy fitting. In the first step of PCMD, it needs to collect data including the voltage of nodes, frequency, the current of nodes, and the robustness for CCF, and it needs also to consider the control cost reduction as a feedback variable. In the second step of PCMD, it needs to detect failure symptoms and predict a CCF propagation sequence from the occurrence of an initial failure. In the third step of PCMD, it uses the deep neural network to maintain a parameterized PCP $l(s, w)$ pursuing the maximum of the infinite horizon discounted expected cumulative reward. Based on the model defined in definition 1, the infinite horizon discounted expected cumulative reward from an initial state $g_0$ is $E \left[ \sum_{i=0}^{\infty} \gamma^i r(F_i) \right]$, $E$ is the expectation about different trajectories, $F_i$ is a steady state transition function in a trajectory, and $\gamma$ is the discount factor. In addition, the PCMD needs also
deep neural networks to maintain a parameterized target PCP $l'(s, w_l')$ and parameterized action-value functions $Q(s_i, l', w_Q)$ and $Q'(s_i, l', w_Q')$ for constructing $l(s, w_l)$ steadily. Where $s \in S$ is a CCF propagation sequence, and $s_i$ is a CCF propagation sequence in a steady state $g_i$. $w_l, w_l', w_Q,$ and $w_Q'$ are the parameters of parameterized functions.

According to Definition 1, many possible expected steady states could be reached from the current steady state when a PCP is attempted to block a CCF propagation sequence. The goal of the constructed PCP is to be selected to reach an optimal steady state from the power bound of a power generator node $n$. $ζ$ denotes the proportion of failure nodes during the propagation of CCF in an ADN. $W$ denotes the weight of the control cost objective, and $R_P$ (a discrete quantity) denotes the proportion of failure nodes during the propagation of CCF in an ADN.

Formula (1) describes the control cost objective and the robustness (against CCF) objective. Instead of using the Pareto idea, the commonly used weighting method [19] is used to integrate multiple objectives into a single objective function.

During the process of blocking a CCF propagation sequence, the safety requirements and the basic physical constraints should be obeyed. Formula (2) describes inequality constraints (including continuous quantities and discrete quantities) corresponding to safety requirements. Formula (3) describes the equality constraint of power flow equations in the PN.

$\min \left( \left( \left| V_{P1} \cup V_{P5} \right| \right) \sum_{i=1}^{\left| V_{P1} \cup V_{P5} \right|} W_{Gi} \Delta P_{Gi} \right) + W_R \left( 1 - R_P \right)$

where $V_{P1}$ denotes the set of power generation nodes (distributed generators), and $V_{P5}$ denotes the set of external nodes (the energy from power transmission lines). $W_{Gi}$ denotes the weight of the control cost objective, and $W_R$ denotes the weight of the robustness (against CCF) objective. $ΔP_{Gi}$ is defined in Definition 1. $R_P$ (a discrete quantity) denotes the proportion of failure nodes during the propagation of CCF in an ADN.

Formula (1) describes the control cost objective and the robustness (against CCF) objective. Instead of using the Pareto idea, the commonly used weighting method [19] is used to integrate multiple objectives into a single objective function.

During the process of blocking a CCF propagation sequence, the safety requirements and the basic physical constraints should be obeyed. Formula (2) describes inequality constraints (including continuous quantities and discrete quantities) corresponding to safety requirements. Formula (3) describes the equality constraint of power flow equations in the PN.

$\left\{ \begin{array}{l} P_{Gi, min} \leq P_{Gi} \leq P_{Gi, max}, Gi \in V_{P1} \\ Q_{Gi, min} \leq Q_{Gi} \leq Q_{Gi, max}, Gi \in V_{P1} \\ I_{Pi} \leq I_{Pi, max}, Pi \in V_{P} \\ v_{i, min} \leq v_i \leq v_{i, max}, i \in V_{P} \\ \Delta P_{Gi, min} \leq \Delta P_{Gi} \leq \Delta P_{Gi, max}, Gi \in V_{P1} \\ 0 \leq R_{Pf} \leq 1 \end{array} \right.$

\begin{cases} P_{Gi} - P_{Li} - \sum_{j=1}^{\text{Num}_g} |v_j||v_j||Y_{ij}| \cos(\alpha_{ij} - \zeta_i + \zeta_j) = 0 & i = 1, \ldots, \text{NON}_P \\ Q_{Gi} - Q_{Li} + \sum_{j=1}^{\text{Num}_g} |v_j||v_j||Y_{ij}| \sin(\alpha_{ij} - \zeta_i + \zeta_j) = 0 & i = 1, \ldots, \text{NON}_P \end{cases}

where $P_{Gi}$ and $Q_{Gi}$ denote the active power and the reactive power of a power generator node $n_{Gi}$, respectively, $P_{Li}$ and $Q_{Li}$ denote the active power and the reactive power of a load node $n_{Li}$, respectively, $v_i$ denotes the voltage of a node $n_i$ in the PN, $I_{Pi}$ denotes the current of a node $n_i$ in the PN, $f_{ref}$ denotes the frequency of an ADN, and $P_{Gi}, Q_{Gi}, P_{Li}, Q_{Li}, v_i, I_{Pi},$ and $f_{ref}$ are all defined in Definition 1. To be safe, it is necessary to limit the value ranges of variables. Correspondingly, $P_{Gi, min}$ and $P_{Gi, max}$ denote the lower and the upper active power bound of a power generator node $n_{Gi}$, respectively. Similarly, $Q_{Gi, min}$ and $Q_{Gi, max}$ denote the lower and the upper reactive power bound of a power generator node $n_{Gi}$, respectively. $P_{Li, min}$ and $P_{Li, max}$ denote the lower and the upper active power bound of a load node $n_{Li}$, respectively. $Q_{Li, min}$ and $Q_{Li, max}$ denote the lower and the upper reactive power bound of a load node $n_{Li}$, respectively. $v_{i, min}$ and $v_{i, max}$ denote the lower and the upper voltage bound of a node $n_i$ in the PN, $I_{Pi, max}$ denotes the upper current bound of a node $n_i$ in the PN, $f_{ref, min}$ and $f_{ref, max}$ denote the frequency limits of an ADN. \text{NON}_P$ is the number of nodes in the PN. $\zeta_i$ denotes the voltage phase angle of a node $n_i$ in the PN. $Y_{ij}$ and $\alpha_{ij}$ denote the
admittance and admittance phase angle between a node \( n_i \) and a node \( n_j \), respectively, in the PN.

In order to get the fitted PCP, the objective function and inequality constraints shown in Formulas (1) and (2) are integrated into the infinite horizon discounted expected cumulative reward, and then the immediate reward \( r \) of a transition should be given. The immediate reward \( r \) is composed of a reward \( r_1 \) reflecting voltage stability, a reward \( r_2 \) for preventing the CCF caused by the current overload, a reward \( r_3 \) reflecting the frequency stability, a reward \( r_4 \) reflecting the control cost, and a reward \( r_5 \) reflecting robustness against CCF. The reward \( r_1 \) is shown in Formula (4).

\[
  r_1 = \sum_{i \in V_p} w_{ni} r_{ni_1}
\]  

(4)

where \( r_1 \) is the weighted sum of \( r_{ni_1} \), and the definition of \( r_{ni_1} \) is shown in Formula (5). \( w_{ni} \) is the weight (proportional to the importance of a node) of a node \( n_i \). \( V_p \) is the set of nodes in the PN. \( r_{1,ni_1} \) (greater than zero) and \( r_{1,ni_2} \) (less than zero), respectively, denote the specific immediate reward value when the voltage value of the node \( n_i \) is in different ranges, \( \parallel \cdot \parallel \) denotes the norm, \( v_{i,normal} \) is the reference voltage value of a node \( n_i \), \( v_{ni} = \min\{||v_{i,normal} - v_{i,min}||,||v_{i,max} - v_{i,normal}||\} \) is the voltage threshold of a node \( n_i \).

\[
  r_{ni_1} = \begin{cases} 
    r_{1,ni_1} & \parallel v_i - v_{i,normal} \parallel \leq v_{ni} \\
    r_{1,ni_2} & \parallel v_i - v_{i,normal} \parallel > v_{ni}
  \end{cases}
\]  

(5)

The reward \( r_2 \) is shown in Formula (6).

\[
  r_2 = \sum_{i \in E_p} w_{ni} r_{ni_2}
\]  

(6)

where \( r_2 \) is the weighted sum of \( r_{ni_2} \), and the definition of \( r_{ni_1} \) is shown in Formula (7). \( E_p \) is the set of edges in the PN. \( r_{2,ni_1} \) (greater than zero) and \( r_{2,ni_2} \) (less than zero), respectively, represent the specific reward value when the current value of the node \( n_i \) is in different ranges, \( i_{normal} \) is the reference current value of a node \( n_i \) in an ADN, \( i_{ni} = i_{normal} \) is the current threshold value of the node \( n_i \).

\[
  r_{ni_2} = \begin{cases} 
    r_{2,ni_1} & \parallel i_i - i_{normal} \parallel \leq i_{ni} \\
    r_{2,ni_2} & \parallel i_i - i_{normal} \parallel > i_{ni}
  \end{cases}
\]  

(7)

The reward \( r_3 \) is shown in Formula (8).

\[
  r_3 = \begin{cases} 
    r_{3,ref_1} & \parallel f_{ref} - f_{ref,normal} \parallel \leq c_{ref} \\
    r_{3,ref_2} & \parallel f_{ref} - f_{ref,normal} \parallel > c_{ref}
  \end{cases}
\]  

(8)

where \( r_{3,ref_1} \) (greater than zero) and \( r_{3,ref_2} \) (less than zero), respectively, represent the specific reward value when the frequency value is in different ranges, \( f_{ref,normal} \) is the reference frequency value, \( c_{ref} = \min\{||f_{ref,min} - f_{ref,normal}||,||f_{ref,max} - f_{ref,normal}||\} \) is the frequency threshold value of an ADN.

The reward \( r_4 \) is shown in Formula (9).

\[
  r_4 = \begin{cases} 
    r_{4,c_1} & a' \leq c_{pc} \\
    r_{4,c_2} & a' > c_{pc}
  \end{cases}
\]  

(9)

where \( a \in R^n = A \) is the control action, \( r_{4,c_1} \) (greater than zero) and \( r_{4,c_2} \) (less than zero), respectively, denote the specific reward value when the control action is in different ranges, and \( c_{pc} \) is the control action threshold value.

The reward \( r_5 \) is the sum of \( r_{5,1} \) and \( r_{5,2} \), as shown in Formula (10).

\[
  r_5 = r_{5,1} + r_{5,2} = r_{nf1}^i_{failure}(t) + \sum_{i=0}^{\frac{t}{\Delta t}} r_{nf2}^i_{failure}(i \ast \Delta t)
\]  

(10)
where \( r_{5,1} \) denotes the number of failure nodes at the current time \( t \), \( r_{5,2} \) denotes the cumulative number of failure nodes from the start time 0 to the current time \( t \). \( r_{nf1} \) and \( r_{nf2} \) denote the weights of \( r_{5,1} \) and \( r_{5,2} \), respectively, and they are both greater than zero, and \( n_{failure}(t + \Delta t) \) \( (i \in \{0,1,2, \ldots, t/\Delta t\}) \) denotes the number of failure nodes in an ADN at discrete time \( t/\Delta t \).

The values of the rewards \( r_1, r_2, r_3, \) and \( r_4 \) are the larger the better, while the value of the reward \( r_5 \) is the smaller the better. Thus, the total immediate reward \( r \) is shown in Formula (11).

\[
r = \sum_{j=1}^{4} r_j - r_5 \tag{11}
\]

The convergence and optimization conditions of the construction process in the PCMD need to be guaranteed, and they are considered in the following sections.

### 3.2. Convergence

In the PCMD, it is necessary to ensure that the construction process converges and the PCP against CCF generated from the construction process exists. These requirements are guaranteed by the two step parameter’s adjustments. In the first step, the convergence of the construction process could be guaranteed by adjusting the parameters in fitted functions \( (l(s,w_l), l'(s,w_l), Q(s,l(s,w_l),w_Q), \) and \( Q'(s,l'(s,w_l),w_Q')) \). In the second step, the existence of the PCP against CCF could be guaranteed by adjusting additionally the parameters in the reward \( r \).

The parameters fitted functions need to be adjusted, including the learning rate \( \delta \), the initial weights of fitted functions \( (l(s,w_l), l'(s,w_l), Q(s,l(s,w_l),w_Q), \) and \( Q'(s,l'(s,w_l),w_Q')) \), the parameter \( \alpha \), and other parameters of fitted functions based on deep neural networks.

**Theorem 1.** Given the equation \( w = \phi(w) \), if its derivative \( \phi'(w) \) is continuous on a closed interval \([w_1, w_2]\), when \( w \in [w_1, w_2] \), \( \phi(w) \in [w_1, w_2] \), there exists a constant \( 0 < z < 1 \), such that when \( w \in [w_1, w_2] \), \( ||\phi'(w)|| \leq z < 1 \), the following results hold.

1. \( \phi(w) \) has a unique fixed point \( w_\perp \) on the closed interval \([w_1, w_2]\).
2. For any initial value \( w_0 \in [w_1, w_2] \), the iterative scheme \( w_{k+1} = \phi(w_k) \ (k = 0, 1, 2, \ldots) \) converges, and \( \lim_{k \to \infty} w_k = w_\perp \).
3. For the sequence \( \{w_k\}_{k=0}^{\infty} \), there exists an asymptotic error estimation shown in Formula (12).

\[
\lim_{k \to \infty} \frac{w_k - w_\perp}{w_{k+1} - w_\perp} = \phi'(w_\perp) \tag{12}
\]

The process of fitting \( l_\perp (s) \) is actually a fixed-point iterative scheme, and the weight \( w_l \) of the fitted PCP \( l(s,w_l) \) is obtained by solving the equation \( w_l = \phi(w_l) \). The iterative function \( \phi \) is defined as shown in Equation (13).

\[
\phi(w_l) = w_l + \delta \left( \frac{1}{N_s} \sum_{i=1}^{N_s} \left( \nabla_s Q(s_i, l(s_i, w_l), w_Q) \right) \left( \nabla_{w_l} l(s_i, w_l) \right) \right) \tag{13}
\]

where \( N_s \) is the size of a trajectory data sample from a data pool, \( s_i \) is a CCF sequence in a steady state \( g_i \).

If the adjustments of the learning rate \( \delta \) and the initial weight of the fitted function \( l(s,w_l) \) meet the preconditions of Theorem 1, then the weight sequence of the \( l(s,w_l) \) converges to the solution \( w_{l(1)} \) of the equation \( w_l = \phi(w_l) \).

The following analysis then explains how the initial weights of the fitted functions \( (l'(s,w_l), Q(s,l(s,w_l),w_Q) \) and \( Q'(s,l'(s,w_l),w_Q')) \) are adjusted. For example, due to the
weights update \( \omega^{(k+1)}_Q = \alpha \omega^{(a)}_Q + (1 - \alpha) \omega^{(k)}_Q \) at step \( k \), there is an error equation shown in Equation (14).

\[
\omega^{(k+1)}_Q - \omega^{(\perp)}_Q = \alpha \left( \omega^{(u)}_Q - \omega^{(\perp)}_Q \right) + (1 - \alpha) \left( \omega^{(k)}_Q - \omega^{(\perp)}_Q \right)
\]

(14)

where \( \omega^{(k)}_Q \) denotes the weight of the \( Q'(s,l'(s,w'),w_Q) \) at step \( k \). \( \omega^{(u)}_Q \) denotes the weight of the \( Q(s,l(s,w),w_Q) \) at step \( u \), and \( u \geq k \). \( \omega^{(\perp)}_Q \) denotes the true value of the weights \( w_Q \) and \( w_{Q'} \).

An inequation is obtained by taking the norm of the elements in Equation (14), as shown in Formula (15).

\[
\left\| \omega^{(k+1)}_Q - \omega^{(\perp)}_Q \right\| \leq \alpha \left\| \omega^{(u)}_Q - \omega^{(\perp)}_Q \right\| + \left\| \omega^{(k)}_Q - \omega^{(\perp)}_Q \right\|
\]

(15)

where \( 0 < \alpha < 1 \).

The true action-value function \( Q_\perp(s_i, a_i) \) of a steady state \( g_i \) has the relation shown in Formula (16).

\[
Q_\perp(s_i, a_i) = E[r(F_i) + \gamma Q_\perp(s_{i+1}, a_{i+1})]
\]

(16)

Thus, the objective function value \( y_i \) of the \( Q(s,l(s,w),w_Q) \) in a steady state \( g_i \) is derived from Formula (16), as shown in Formula (17).

\[
y_i = r(F_i) + \gamma Q'(s_{i+1}, l'(s_i, w'), w_{Q'})
\]

(17)

where \( s_{i+1} \) is a CCF sequence of a steady state \( g_{i+1} \).

From Formula (17), a loss function \( L_Q \) for updating the weight \( w_Q \) of the \( Q(s,l(s,w),w_Q) \) is shown in Equation (18).

\[
L_Q = E \left[ (Q(s_i, l(s_i, w_i), w_Q) - y_i)^2 \right] \approx \frac{1}{N_S} \sum_{i=1}^{N_S} (Q(s_i, l(s_i, w_i), w_Q) - y_i)^2
\]

(18)

According to Formulas (17) and (18), the weights of the \( Q(s,l(s,w),w_Q) \) converge to the weights of the \( Q'(s,l'(s,w'),w_{Q'}) \), and when \( u \gg k \),

\[
\left\| \omega^{(k+1)}_Q - \omega^{(\perp)}_Q \right\| \leq \eta_Q \left\| \omega^{(k)}_Q - \omega^{(\perp)}_Q \right\| , \quad 0 < \eta_Q \leq 1.
\]

So, the weight sequence of the \( Q'(s,l'(s,w'),w_{Q'}) \) converges to the true value \( \omega^{(\perp)}_Q \). Likewise, the weight sequence of the \( Q(s,l(s,w),w_Q) \) converges to the true value \( \omega^{(\perp)}_Q \) and

\[
\left\| \omega^{(k+1)}_Q - \omega^{(\perp)}_Q \right\| \leq \eta_Q \left\| \omega^{(k)}_Q - \omega^{(\perp)}_Q \right\| , \quad 0 < \eta_Q \leq 1.
\]

An inequation is obtained by taking the norm of the elements in Equation (14) and repeating the substitutions, as shown in Formula (19).

\[
\left\| \omega^{(k+1)}_Q - \omega^{(\perp)}_Q \right\| \leq \alpha \eta_Q^{k} \left\| \omega^{(0)}_Q - \omega^{(\perp)}_Q \right\| + (1 - \alpha)\alpha \eta_Q^{k-1} \left\| \omega^{(0)}_Q - \omega^{(\perp)}_Q \right\| + \ldots + (1 - \alpha)\sum_{i=0}^{k-1} \alpha \eta_Q^{i} \left\| \omega^{(0)}_Q - \omega^{(\perp)}_Q \right\|
\]

(19)

where \( \omega^{(0)}_Q \) denotes the initial weight of the \( Q(s,l(s,w),w_Q) \), and \( \omega^{(0)}_{Q'} \) denotes the initial weight of the \( Q'(s,l'(s,w'),w_{Q'}) \).

According to Formula (19), the weight of the fitted function \( Q'(s,l'(s,w'),w_{Q'}) \) tends to the true value \( \omega^{(\perp)}_Q \) as \( u \) and \( k \) all tend to infinity. Because of the parameters \( \alpha < 1 \) and \( \eta_Q \leq 1 \), the initial weight \( \omega^{(0)}_Q \) of the \( Q(s,l(s,w),w_Q) \) can be selected randomly. The initial weight \( \omega^{(0)}_Q \) of the fitted function \( Q(s,l(s,w),w_Q) \) can be selected randomly due to Formulas (16) and (17). By exploring the similar analysis, the weight sequence of the \( l'(s,w') \) converges to the true value \( \omega^{(\perp)}_Q \) and the initial weight \( \omega^{(0)}_Q \) of the \( l'(s,w') \) can also be selected randomly. As for the adjustment of the parameter \( \alpha \), the value of the parameter \( \alpha < 1 \) can be set as high as possible in order to speed up the convergence of the fitted
functions $Q'(s, l'(s, w_r), w_Q')$ and $l'(s, w_r)$. However, because of $\eta_Q \leq 1$ and $\eta_{Q'} \leq 1$, the value of the parameter $a < 1$ should not be set too high in order to improve the computational stability of the fitted function $Q(s, l(s, w), w_Q)$.

In addition, in order to prevent the output of the fitted functions ($l(s, w_2), l'(s, w_r), Q(s, l(s, w), w_Q)$, and $Q'(s, l'(s, w), w_Q')$) from being saturated, the data should be standardized (per unit is used in this paper), the number of neural network layers should be appropriately reduced, the batch normalization layer should be raised, and the activation function layers should be placed behind them. After applying the above parameters adjustments, the convergence of the construction process could be guaranteed, and a PCP could be generated.

In order to guarantee the existence of the PCP against CCF in an ADN, the parameters in the reward $r$ should be adjusted in addition to the parameters adjustments in fitted functions. The idea of the parameters adjustment in the reward $r$ is to reduce the propagation width and depth of CCF. The basic idea of the parameter adjustments based on reducing the propagation width of CCF in an ADN is to ensure that the number of failure nodes at the same time is as small as possible, and the requirements of the voltage stability, the frequency stability, the current overload prevention, and the control cost reduction are given, respectively. The gradient theorem of the deterministic control policy shown in Theorem 2 ensures the existence of the gradient of the deterministic control policy. In addition, once the condition of the compatibility between the fitted function $Q(s, l(s, w), w_Q)$ and the fitted PCP $l(s, w_2)$ against CCF is satisfied, shown in Theorem 3, a local optimum PCP $l(s, w_2)$ against CCF can be constructed through the construction process proposed in this paper.

3.3. Satisfiability of Suboptimal Solution

If the PCMD has been guaranteed to be convergent, it is necessary to ensure this process should converge to the optimal solution. The following analysis shows that if the compatibility condition is satisfied, the generated PCP against CCF in an ADN can converge to a local optimal solution.

The performance function for evaluating the PCP against CCF generated in the PCMD is defined as the expected fitted function $E(Q(s, l(s, w), w_Q) l(s, w))$, which is shown in Equation (20).

$$J(l(s, w)) = E(Q(s, l(s, w), w_Q))$$

The gradient theorem of the deterministic control policy and the condition of compatibility between the fitted function $Q(s, l(s, w), w_Q)$ and the fitted PCP $l(s, w_2)$ against CCF are given, respectively. The gradient theorem of the deterministic control policy shown in Theorem 2 ensures the existence of the gradient of the deterministic control policy. In addition, once the condition of the compatibility between the fitted function $Q(s, l(s, w), w_Q)$ and the fitted PCP $l(s, w_2)$ against CCF is satisfied, shown in Theorem 3, a local optimum PCP $l(s, w_2)$ against CCF can be constructed through the construction process proposed in this paper.

**Theorem 2. (Gradient Theorem of Deterministic Control Policy):** It is assumed that the environment (the controlled object) satisfies the condition of a Markov decision process (MDP), and the fitted action-value function $Q(s, a)$ and the fitted PCP $l(s, w)$ are both continuously differentiable. It is also assumed that the gradient $\nabla_a Q(s, a)$ of the function $Q(s, a)$ with respect to the control action $a$ and the gradient $\nabla_w l(s, w)$ of the function $l(s, w)$ with respect to the parameter $w_i$ exist. According to Formula (20), then the gradient of the performance function $J(l(s, w))$ exists, and it is shown in Equation (21). (The proof of Theorem 2 is given in Appendix A).
\[ \nabla_w E(Q(s, l(s, w_l))) = E(\nabla_a Q(s, a = l(s, w_l)) \nabla_w l(s, w_l)) \] (21)

In the PCMD, there are two fitted PCPs \( l(s, w_l) \) and \( l'(s, w_{l'}) \), and there are two fitted action-value functions \( Q(s, l(s, w_l)) \) and \( Q'(s, l'(s, w_{l'})) \). It is assumed that the gradients \( \nabla_w Q(s, l(s, w_l)) \), \( \nabla_w l(s, w_l) \), \( \nabla_w l'(s, w_{l'}) \), and \( \nabla_{w_{l'}} Q'(s, l'(s, w_{l'})) \) exist. Then, according to Theorem 2, the gradient of the performance function for the fitted PCP \( l(s, w_l) \) is shown in Equation (22).

\[ \nabla_w J(l(s, w_l)) = E(\nabla_l Q(s, l(s, w_l)) \nabla_w l(s, w_l)) \] (22)

Correspondingly, the gradient of the performance function for the fitted target PCP \( l'(s, w_{l'}) \) is shown in Equation (23).

\[ \nabla_w J(l'(s, w_{l'})) = E(\nabla_{l'} Q'(s, l'(s, w_{l'})) \nabla_w l'(s, w_{l'})) \] (23)

**Theorem 3.** (Compatibility Theorem of Fitted Function \( Q(s, a, w_Q) \) and Fitted PCP \( l(s, w_l) \)): In a learning process, the \( Q(s, a, w_Q) \) is a fitted function about function \( Q_{\perp} (s, a) \) of control action \( a \). If the action-value function \( Q_{\perp} (s, a) \) and the fitted function PCP \( l(s, w_l) \) are both continuously differentiable, their gradients \( \nabla_w Q(s, a, w_Q), \nabla_a Q_{\perp} (s, a) \), and \( \nabla_w l(s, w_l) \) exist, and the second gradient \( \nabla_{w_{l'}} \nabla_a Q_{\perp} (s, a, w_Q) \) also exists, and the compatibility condition shown in Equation (24) is satisfied, and the parameter \( w_Q \) can minimize the function shown in Formula (25), then there is a local optimal PCP, which is noted as \( l(s, w_l) \) shown in Equation (26). (The proof of Theorem 3 is given in Appendix B).

\[ \nabla_a Q(s, a = l(s, w_l), w_Q) = \nabla_w l(s, w_l)^T w_Q \] (24)

\[ E\left[ (\nabla_a Q(s, a = l(s, w_l), w_Q) - \nabla_a Q_{\perp} (s, a = l(s, w_l)))^2 \right] \] (25)

\[ E[\nabla_a Q(s, a = l(s, w_l), w_Q) \nabla_w l(s, w_l)] = E[\nabla_a Q_{\perp} (s, a = l(s, w_l)) \nabla_w l(s, w_l)] \] (26)

where \( \nabla_w l(s, w_l)^T \) is the transpose of \( \nabla_w l(s, w_l) \).

According to Theorem 1, if the parameters adjustments conditions are satisfied, then the construction process for a PCP against CCF converges. According to Theorem 2, the existence of the gradient of the performance function \( J(l(s, w_l)) \) is guaranteed. According to Theorem 3, if the condition of the compatibility between the fitted function \( Q(s, l(s, w_l), w_Q) \) and the fitted PCP \( l(s, w_l) \) is satisfied, then the alternative gradient formed by the fitted function \( Q(s, l(s, w_l), w_Q) \) and the fitted PCP \( l(s, w_l) \) is equal to the gradient of the performance function \( J(l(s, w_l)) \), and a local optimum PCP \( l(s, w_l) \) can be constructed through the PCMD.

### 3.4. Algorithm Implementation

The reinforcement learning algorithms based on the stochastic control policy gradient can deal with the problems with the discrete control action space, the continuous control action space, the continuous state space, and the discrete state space. Because the stochastic control policy gives the corresponding probability for each value in the control action space, it is computation intensive and it could be practically infeasible in large-scale networks. Hence, the deterministic control policy is produced. Deep deterministic policy gradient (DDPG) is a deep deterministic control policy reinforcement learning algorithm [23]. It is different from the random control policy gradient method mentioned above. It integrates the deep neural networks and the deterministic control policy [24], and it updates the parameters of the deep neural networks by using the gradient descent of the deterministic control policy. DDPG uses the deterministic control policy gradient to deal with the continuous action space, and it is also applicable to both the continuous state space and the discrete state space.
Q equation is used to implement the construction process for the PCP against CCF in an ADN through the offline interactive learning, as shown in Algorithm 1. Algorithm 1 adopts the deep neural network to fit a PCP (l(s, w_l)). The weight w_l in l(s, w_l) is learned by the PMCD algorithm.

Algorithm 1 PMCD Algorithm

Input: the state g_t, the immediate reward r_t at discrete time t * Δt.
///Initialization
1 w_Q←rand, w_l←rand, RB←N_1, w_Q′←w_Q, w_l←w_l;
///Training Process
2 for episode = 1 to N_2 do
3 N_1←randN;
4 Receive the state g_1;
5 Detect failure symptoms;
6 Predict a CCF propagation sequence s_t from the occurrence of an initial failure;
7 for t = 1, N_4 do
8 a_t←l(s_t, w_l) + N_2;
9 Perform the action a_t and observe the immediate reward r_t and the next state g_{t+1};
10 Detect failure symptoms;
11 Predict a CCF propagation sequence s_{t+1} from the occurrence of an initial failure;
12 Put the data (s_t, a_t, r_t, s_{t+1}) into the replay buffer RB;
13 Take randomly the mini-batch N_5 samples (s_t, a_t, r_t, s_{t+1}) from the replay buffer;
14 if g_{t+1} is the final state do
15 y_t = g_{t+1};
16 else
17 y_t = r_t + γQ'(s_{t+1}, a_{t+1}) + N_6;
18 w_Q ← argmin 1/N_4 \sum_{i=1}^{N_4} (y_{t,i} - Q(s_i, a_i, w_l))^2;
19 w_l ← w_l + δ \left( \frac{1}{N_4} \sum_{i=1}^{N_4} (Q(s_i, a_i, l(s_i, w_l), w_l)) - \frac{1}{N_4} \sum_{i=1}^{N_4} (Q(s_i, a_i, l(s_i, w)) - \nabla w_l l(s_i, w_l)) \right);
20 w_Q′ ← w_Q + (1-α)w_Q;
21 w_l ← w_l + (1-α)w_l;
Output: A preventive control policy l(s).

Where Rand denotes the random number, RB denotes the size of the replay buffer, and randN denotes the selected random process. N_1, N_2, N_4 and N_5 denote the parameters in Algorithm 1, and N_3 denotes a random process. α and δ are described in Section 3.2.

The execution order of PCMD algorithm for an ADN is as follows. For each time step Δt under each episode, firstly, the algorithm detects failure symptoms and predict a CCF propagation sequence s_t from the occurrence of an initial failure. Then, the fitted PCP l(s, w_l) selects a control action a_t at discrete time t * Δt according to the current CCF propagation sequence s_t, and the ADN transits from a current state g_t to another state g_{t+1} under the influence of the CCF propagation sequence s_t, and then, the data (s_t, a_t, r_t, s_{t+1}) are put into a replay buffer. At last, the weights of the Q(s, l(s, w_l), w_Q), Q'(s, l'(s, w'), w_Q'), l(s, w_l), and l'(s, w'_l) are updated.

The playback buffer mechanism used in Algorithm 1 is first proposed in the paper [25]. The playback buffer mechanism makes the correlated data independent and the noise cancel each other, which makes the learning process of Algorithm 1 converge faster. If there is no playback buffer mechanism, Algorithm 1 could make a gradient descent in the same direction for a period of time. Under the same step size, calculating the gradient directly may make the learning process not converge. The playback buffer mechanism is to select some samples randomly from a memory pool, update the weight of the fitted Q(s, l(s, w_l), w_Q) by the temporal-difference learning, then, use this weight information to estimate the gradients of the Q(s, l(s, w_l), w_Q) and the l(s, w_l), and then update the weight of the fitted PCP. The introduction of the random noise in Algorithm 1 ensures the execution of the exploration process.
According to Section 3.2, the convergence of Algorithm 1 can be guaranteed via parameters adjustments, and a PCP against CCF could be obtained. In addition, the introduction of the parameter $\alpha$ in Algorithm 1 ensures the stability of weights updating in the $l'(s,w_l)$ and the $Q'(s,l'(s,w_l),w_Q)$ and thus ensures the stability of the whole learning process. According to Section 3.3, once the compatibility condition is satisfied, the generated PCP against CCF in an ADN can converge to the local optimal solution. Thus, after interacting with the offline trajectory dataset of an ADN iteratively, a fitted PCP can be learned by Algorithm 1.

4. Case Study

An experimental CEPR-CPS is an ADN case, and it is designed from the actual system provided by China Electric Power Research Institute. The topology and node numbering of PN and CN in CEPR-CPS are shown in Figure 4. Node numbers in the PN are from 1 to 80, and node numbers in the CN are from 81 to 109. Nodes 1–4, 23, 60, 75–80 (abstracted from the contact switch) are distribution nodes. Nodes 17, 18, 24, 25, 43, 44, 56 are substation nodes. Nodes 66–74 are power generation nodes (66, 68, and 70: battery energy storage nodes, 67 and 73: photovoltaic power generation nodes, 69 and 74: wind power generation nodes, 71: a hydraulic turbine node, 72: a micro-gas turbine node). Nodes 63–65 are external nodes. The rest in the PN are load nodes. A node 81 is the control node, nodes 82–87 are information relay nodes, and nodes 88–109 are sensor/actuator nodes. There are 109 nodes in the CEPR-CPS. There are 80 nodes and 107 edges in the PN, and there are 29 nodes and 28 edges in the CN. The directed independence edges between the PN and CN in the CEPR-CPS are shown in Table 1. A directed independence edge is from a node to its successor node. Nodes of PN supply power to nodes of CN. Accordingly, nodes of CN collect the information of nodes in the PN and thus control the actions of them. For example, the node 43 of PN supplies power to nodes 104, 106, and 107 of CN (as shown by the dotted orange line in Figure 4) and the node 99 of CN collects the information of the node 16 in the PN (as shown by the dotted blue line in Figure 4).

![Figure 4. Topology and node numbering of PN and CN in the CEPR-CPS.](image)
Table 1. The directed independence edges between the power network (PN) and communication network (CN) in the China Electric Power Research-Cyber-Physical System (CEPR-CPS).

| Node | Successor Node | Node | Successor Node |
|------|----------------|------|----------------|
| 17   | 84, 95, 96, 97, 98, 99, 109 | 95   | 12, 63, 64     |
| 18   | 100, 101, 102 | 96   | 13, 65         |
| 24   | 81, 83, 85    | 97   | 14, 66, 67, 68 |
| 25   | 103           | 98   | 15, 69, 70, 71, 72 |
| 43   | 104, 106, 107 | 99   | 16             |
| 44   | 105           | 100  | 23             |
| 56   | 82, 86, 108   | 101  | 21, 22, 24     |
| 88   | 1, 2, 3, 4, 5 | 102  | 25, 30, 31, 32 |
| 89   | 6, 17, 18, 19, 20 | 103 | 33, 34, 35, 36 |
| 90   | 7, 26, 27, 28, 29 | 104 | 40, 41, 42, 44 |
| 91   | 9, 46, 47, 48, 50 | 105 | 43, 51         |
| 92   | 9, 46, 47, 48, 50 | 106 | 45, 56, 58, 59 |
| 93   | 10, 52, 53, 54, 55 | 107 | 49             |
| 94   | 11, 60, 61, 62 | 108  | 57, 73         |
|      |                | 109  | 72, 74         |

The CEPR-CPS is used to verify the effectiveness of the PCP against CCF generated from the PCMD compared with the one from the multi-agent method.

4.1. Trajectory Data Collection

In order to construct a PCP against CCF in the CEPR-CPS using the PCMD, it needs to collect the offline trajectory data \( \{i \in \text{Tra} | (g_{i\Delta t}, a_{i\Delta t}, r_{i\Delta t}, g_{(i+1)\Delta t})\} \) from the CEPR-CPS. Where \( \text{Tra} = \{0, 1, 2, \ldots, Tr/\Delta t\} \), \( \Delta t \) denotes the duration of a trajectory. According to Definition 1, the data are collected in the CEPR-CPS at discrete time \( i^{\star} \Delta t (i \geq 0) \), and the data collected include the state data \( g_{i\Delta t} \), the control action data \( a_{i\Delta t} \), and the reward data \( r_{i\Delta t} \) of CEPR-CPS.

The state data \( g_{i\Delta t} \) of CEPR-CPS include the three-phase voltage and the three-phase current values (complex number, per unit) of the 80 nodes, the rotor mechanical angle (deg) and the rotor speed (per unit) of a hydraulic turbine node, the active and the reactive power values (complex number, per unit) of two wind power generation nodes, and the three-phase voltage values and the three-phase current values (complex number, per unit) from inverters of one photovoltaic node, one micro-gas turbine node, and three energy storage nodes.

According to different operating conditions, leading conditions and subsequent constraints, the control action data \( a_{i\Delta t} \) of CEPR-CPS include the phase voltage of the controllable voltage source of energy storage nodes, the reactive power of wind power generation nodes, the three-phase voltage of the controllable voltage source of micro-gas turbine nodes and photovoltaic nodes, and the active power of a hydraulic turbine node.

The reward data \( r_{i\Delta t} \) include a reward \( r_1 \) reflecting the voltage stability, a reward \( r_2 \) for preventing the CCF caused by the current overload, a reward \( r_3 \) reflecting the frequency stability, a reward \( r_4 \) reflecting the control cost, and a reward \( r_5 \) reflecting robustness against CCF.

After the state data \( g_{i\Delta t} \), the action data \( a_{i\Delta t} \) and the reward data \( r_{i\Delta t} \) are obtained at the discrete time \( i^{\star} \Delta t (i \geq 0) \), then the transition data \( (g_{i\Delta t}, a_{i\Delta t}, r_{i\Delta t}, g_{(i+1)\Delta t}) \) can be constructed. After obtaining the transition data in a successively dependent manner, trajectory data can be obtained. Then, multiple different trajectory data form an offline trajectory dataset \( \{(i \in \text{Tra} | (g_{i\Delta t}, a_{i\Delta t}, r_{i\Delta t}, g_{(i+1)\Delta t}))\} \).

4.2. Learning Process

After the trajectory data collection step, the PCMD can interact with the offline trajectory dataset of the CEPR-CPS iteratively. Once parameters in the PCMD are adjusted and the condition of the compatibility between the fitted action-value function \( Q(s, \{s, a, w\}, aw_Q) \)
and the fitted PCP $l(s,w)$ against CCF is satisfied, then the weights of the fitted action-value functions and a local optimal preventive control policy are learned. The parameters used in the PCMD for the CEPR-CPS is shown in Table 2.

**Table 2. Parameters used in the PCMD for the CEPR-CPS.**

| Case | Parameters |
|------|------------|
| CEPR-CPS | stopTrainV $\times 10^{17}$ | cl_RB | $V_{ni}$ | $r_{1_{ni},1}$ | $N_{3_{var}}$ | $T_s$ | $\gamma$ | $\alpha$ |
| 20.0 | False | 0.2 | 10.0 | 4.5 | $1.0 \times 10^{-6}$ | 0.96 | $5.0 \times 10^{-5}$ |
| $r_{1_{ni,2}}$ | 50 | $w_{ni}$ | $I_{ni}$ | $r_{2_{ni,1}}$ | $N_{3,decR}$ | $T_f$ | $\delta$ | $N_2$ |
| $c_{ref}$ | $r_{3,ref,1}$ | $r_{3,ref,2}$ | $c_{pc}$ | $r_{4,c,1}$ | $r_{4,c,2}$ | $r_{nf1}$ |
| $N_1$ | $N_5$ | $N_4$ | $gr_{Th}$ | $r_{nf2}$ | $isD$ |
| 30.0 | $10^{6}$ | | 1024.0 | 3.0 | 1.0 | 10.0 |

Where $T_s$ is the sampling period (seconds), $T_f = Tr$ is the simulation time (seconds), $N_{3_{var}}$ is the variance of the random process, $N_{3,decR}$ is the decay rate of the variance of a random process, $gr_{Th}$ denotes the gradient threshold, $stopTrainV$ denotes the stopping training threshold, $cl_RB$ denotes whether to clear the replay buffer, and $isD$ denotes the stopping threshold of every simulation subprocess. The state is 504 dimensional in total, and the control action is 29 dimensional corresponding to the outputs of distributed generators. The CCF propagation sequence input layer of the fitted $Q$ function is regular and fully connected, including 600 neuron nodes. Its CCF propagation sequence output layer is $relu$, it is fully connected and contains 400 neuron nodes. The action input layer of the fitted $Q$ function is fully connected and contains 400 neuron nodes, the merging layer of the CCF propagation sequence output and the action output of the fitted $Q$ function is $relu$, and it is fully connected and contains only one neuron node. The output layer of the fitted PCP $l$ is regularized and fully connected, including 300 neuron nodes. The output layer of the fitted PCP $l$ is $tanh$, and it is fully connected, containing 29 neuron nodes. However, there is no hidden layer.

**4.3. Results and Analysis**

It focuses on a scenario of the open circuit failure of a node as the initial failure, that is, the initial failure is the physical equipment failure in the CEPR-CPS. The multi-agent method [11] is selected as a control method. Comparison diagram of CCF initiated from a node failure in the PN and in the CN are shown in Figures 5 and 6, respectively.

According to Figures 5 and 6, the PCMD proposed in this paper can block the occurrence of CCF in the CEPR-CPS compared with the existing multi-agent method. Meanwhile, the conclusion that the convergence of PCMD could be guaranteed by the parameters adjustments is verified. The conclusion that a local optimal PCP can be constructed under the comparability condition among a fitted $Q$ function and a fitted policy function is also verified. The PCMD is better than the multi-agent method in terms of reducing the number of failure nodes and avoiding the state space explosion problem. So, an existing problem of the multi-agent method is the curse of dimensionality (feature dimension), and the multi-agent method is not suitable for large-scale networks due to its poor scalability. The PCMD overcome the curse of dimensionality and can be extended to large-scale networks.

The PCMD is compared with the other two kinds of methods: theoretical analysis methods and physical experimental methods in terms of time, cost, reliability, and accuracy. The specific comparison results are shown in Table 3. It can be shown that the PCMD is better than theoretical analysis methods and physical experimental methods in terms of time and cost.
Figure 5. Comparison diagram of CCF initiated from a node failure in the PN.

Figure 6. Comparison diagram of CCF initiated from a node failure in the CN.
Table 3. Comparison between different methods.

| Property  | PCMD | Theoretical Analysis Methods | Physical Experimental Methods |
|-----------|------|------------------------------|-----------------------------|
| time      | short| long                         | long                        |
| cost      | low  | high                         | high                        |
| reliability | medium | high                         | high                        |
| accuracy  | high | high                         | medium                      |

5. Discussion

In this paper, a preventive control model based on the finite automaton theory is designed, which is a six-tuple to describe preventive actions for blocking the propagation of cross-domain cascading failures in an active distribution network of cyber-physical system, and cross-domain cascading failures sequences should be as short as possible. This model is helpful for guiding the trajectory data collection and learning policy selection.

Then, this paper proposes a methodology (named as PCMD) for constructing a preventive control policy to block the propagation of cross-domain cascading failures in an active distribution network of cyber-physical system. This methodology is based on the deep deterministic policy gradient idea to train the deep neural networks with the trajectory data samples originated from simulations and does not need to consider the specific power flow equations. In addition, the parameter adjustments are proposed to guarantee the convergence of the construction process for generating a preventive control policy against cross-domain cascading failures. The gradient theorem of deterministic control policy and the compatibility condition of the fitted state-action function and the fitted preventive control policy have been given to ensure the suboptimality of the generated preventive control policy.

Finally, the proposed PCMD has been verified in the CEPR-CPS provided by the China Electric Power Research Institute. It is shown that the PCMD is better than the multi-agent method in terms of reducing the number of failure nodes and avoiding the state space explosion problem. So, the scalability of the multi-agent method is poor and not suitable for large-scale networks due to the curse of dimensionality. The space complexity of PCMD is very high, and the PCMD has a defect in storing a lot of data. In addition, the proposed PCMD is compared with the theoretical analysis methods and the physical experimental methods in terms of time, cost, reliability, and accuracy. It is shown that the PCMD is better than theoretical analysis methods and physical experimental methods except in terms of reliability.

In future works, there is also a point we should consider. The propagation process of cross-domain cascading failures is time-dependent, and it also takes time for a preventive control policy to work. Therefore, how to reduce the deviation between the time when the preventive control policy works and the time of failure occurrence, that is, the control actions applied by the preventive control policy, really work before a next failure occurs in a cross-domain cascading failures propagation sequence so as to achieve an accurate preventive control effect.

Author Contributions: Conceptualization, P.S. and Y.D.; methodology, P.S.; software, P.S. and S.Y.; validation, P.S., Y.D.; formal analysis, P.S.; investigation, P.S. and Y.D.; resources, Y.D.; data curation, P.S., S.Y. and C.W.; writing—original draft preparation, P.S.; writing—review and editing, Y.D.; visualization, P.S.; supervision, Y.D.; project administration, Y.D.; funding acquisition, Y.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key Research and Development Program of China (Basic Research Class)—Basic Theories and Methods of Analysis and Control of the Cyber–Physical Systems for Power Grid under Grant No. 2017YFB0903000.

Conflicts of Interest: The authors declare that they have no conflicts of interest.
Appendix A. A Proof of Theorem 2

Proof. The gradient of Equation (20) with respect to the parameter $w_i$ is calculated as shown in Equation (A1).

$$\nabla_{w_i} l(s, w_i) = \nabla_{w_i} E(Q(s, l(s, w_i))) = E(\nabla_{w_l} Q(s, l(s, w_i)))$$ (A1)

By the gradient properties of a composite function and $a = l(s, w_i)$, Formula (A2) can be derived.

$$\nabla_{w_l} Q(s, l(s, w_i)) = \nabla_a Q(s, a = l(s, w_i)) \nabla_{w_l} l(s, w_i)$$ (A2)

After substituting Formula (A2) into Formula (A1), then Formula (A3) is obtained.

$$\nabla_{w_l} l(s, w_i) = E(\nabla_a Q(s, a = l(s, w_i)) \nabla_{w_l} l(s, w_i))$$ (A3)

Where $\nabla_{w_l} l(s, w_i)$ is a Jacobian matrix, if $l(s, w_i)$ is an m-dimensional vector function, and the weight $w_i$ is M-dimensional, then the Jacobian matrix $\nabla_{w_l} l(s, w_i)$ is represented shown in Formula (A4).

$$\nabla_{w_l} l(s, w_i) = \begin{bmatrix}
\frac{\partial l_1(s, w_i)}{\partial w_1} & \frac{\partial l_1(s, w_i)}{\partial w_2} & \cdots & \frac{\partial l_1(s, w_i)}{\partial w_M} \\
\cdots & \cdots & \cdots & \cdots \\
\frac{\partial l_m(s, w_i)}{\partial w_1} & \frac{\partial l_m(s, w_i)}{\partial w_2} & \cdots & \frac{\partial l_m(s, w_i)}{\partial w_M}
\end{bmatrix}$$ (A4)

□

Appendix B. A Proof of Theorem 3

Proof. The minimization function shown in (25) means that the condition is satisfied shown in Equation (A5).

$$\nabla_{w_Q} E \left[ (\nabla_a Q(s, a = l(s, w_i), w_Q) - \nabla_a Q \perp (s, a = l(s, w_i)))^2 \right] = 0$$ (A5)

Then, Formula (A6) can be derived from Formula (A5).

$$E \left[ 2(\nabla_a Q(s, a = l(s, w_i), w_Q) - \nabla_a Q \perp (s, a = l(s, w_i))) \nabla_{w_Q} \nabla_a Q(s, a = l(s, w_i), w_Q) \right] = 0$$ (A6)

Thus, Formula (A7) is derived from Formula (24) and Formula (A6).

$$E \left[ (\nabla_a Q(s, a = l(s, w_i), w_Q) - \nabla_a Q \perp (s, a = l(s, w_i))) \nabla_{w_l} l(s, w_i) \right] = 0$$ (A7)

At last, Formula (26) is derived from Formula (A7). □

References
1. Sun, P.; Dong, Y.-W.; Wang, C.; Lv, C.; War, K.Y.; Sun, D.; Wang, L. War Cyber-Physical Active Distribution Networks Robustness Evaluation against Cross-Domain Cascading Failures. Appl. Sci. 2019, 9, 5021. [CrossRef]
2. Voropai, N.; Kurbatsky, V.G.; Tomin, N.; Panasetsky, A.D. Preventive and emergency control of intelligent power systems. In Proceedings of the 2012 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe), Berlin, Germany, 14–17 October 2012.
3. Rabiee, A.; Soroudi, A.; Keane, A. Risk-Averse Preventive Voltage Control of AC/DC Power Systems Including Wind Power Generation. IEEE Trans. Sustain. Energy 2015, 6, 1494–1505. [CrossRef]
4. Tan, Y.; Li, Y.; Cao, Y.; Tan, Y.; Jiang, L.; Keune, B. Comprehensive decision-making method considering voltage risk for preventive and corrective control of power system. IET Gener. Transm. Distrib. 2016, 10, 1544–1552.
5. Xu, Z.; Julius, A.A.; Chow, J.H. Robust testing of cascading failure mitigations based on power dispatch and quick-start storage. IEEE Syst. J. 2017, 12, 3063–3074. [CrossRef]
6. Li, S.; Tan, Y.; Li, C.; Cao, Y.; Jiang, L. A Fast Sensitivity-Based Preventive Control Selection Method for Online Voltage Stability Assessment. *IEEE Trans. Power Syst.* 2018, 33, 4189–4196. [CrossRef]

7. Dong, Y.; Xie, X.; Shi, W.; Zhou, B.; Jiang, Q. Demand-Response-Based Distributed Preventive Control to Improve Short-Term Voltage Stability. *IEEE Trans. Smart Grid* 2018, 9, 4785–4795. [CrossRef]

8. Khazali, A.; Rezaei, N.; Ahmadi, A.; Hredzak, B. Information Gap Decision Theory Based Preventive/Corrective Voltage Control for Smart Power Systems with High Wind Penetration. *IEEE Trans. Ind. Inform.* 2018, 14, 4385–4394. [CrossRef]

9. Alburguetti, L.M.; Grillo, A.P.; Ramos, R.A. Preventive Control for Voltage Stability Enhancement Using Reactive Power from Wind Power Plants. In Proceedings of the Power and Energy Society General Meeting, Atlanta, GA, USA, 4–8 August 2019.

10. Xypolytou, E.; Zseby, T.; Fabini, J.; Gawlik, W. Detection and mitigation of cascading failures in interconnected power systems. In Proceedings of the IEEE PES Innovative Smart Grid Technologies Conference, Torino, Italy, 26–29 September 2017.

11. Babalola, A.A.; Belkacemi, R.; Zarrabian, S. Real-Time Cascading Failures Prevention for Multiple Contingencies in Smart Grids Through a Multi-Agent System. *IEEE Trans. Smart Grid* 2018, 9, 373–385. [CrossRef]

12. Liu, C.; Sun, K.; Rather, Z.H.; Chen, A.; Bak, C.L.; Thøgersen, P.; Lund, P. A Systematic Approach for Dynamic Security Assessment and the Corresponding Preventive Control Scheme Based on Decision Trees. *IEEE Trans. Power Syst.* 2014, 29, 717–730. [CrossRef]

13. Passaro, M.C.; da Silva, A.P.A.; Lima, A.C.S. Preventive Control Stability via Neural Network Sensitivity. *IEEE Trans. Power Syst.* 2014, 29, 2846–2853. [CrossRef]

14. Kucuktezcan, C.F.; Genc, V.M.; Erol, O.K. An optimization method for preventive control using differential evolution with consecutive search space reduction. In Proceedings of the IEEE PES Innovative Smart Grid Technologies Conference, Ljubljana, Slovenia, 9–12 October 2016.

15. Soni, B.P.; Saxena, A.; Gupta, V.; Surana, S.L. Transient stability-oriented assessment and application of preventive control action for power system. *J. Eng.*, 2019, 2019, 5345–5350. [CrossRef]

16. Kou, P.; Liang, D.; Wang, C.; Wu, Z.; Gao, L. Safe deep reinforcement learning-based constrained optimal control scheme for active distribution networks. *Appl. Energy* 2020, 264, 114772. [CrossRef]

17. Belkacemi, R.; Babalola, A.; Zarrabian, S. Experimental implementation of Multi-Agent System Algorithm to prevent Cascading Failure after N-1-1 contingency in smart grid systems. In Proceedings of the 2015 IEEE Power & Energy Society General Meeting, Denver, CO, USA, 26–30 July 2015.

18. Zarrabian, S.; Belkacemi, R.; Babalola, A.A. Intelligent mitigation of blackout in real-time microgrids: Neural Network Approach. In Proceedings of the Power and Energy Conference at Illinois, Urbana, IL, USA, 19–20 February 2016.

19. Zarrabian, S.; Belkacemi, R.; Babalola, A.A. Reinforcement learning approach for congestion management and cascading failure prevention with experimental application. *Electr. Power Syst. Res.* 2016, 141, 179–190. [CrossRef]

20. Khederzadeh, M.; Beiranvand, A. Identification and Prevention of Cascading Failures in Autonomous Microgrid. *IEEE Trans. Syst. J.* 2018, 12, 308–315. [CrossRef]

21. Dutta, O.; Mohamed, A. Reducing the Risk of Cascading Failure in Active Distribution Networks Using Adaptive Critic Design. *IET Gener. Transm. Distrib.* 2020, 14, 2592–2601. [CrossRef]

22. Rahnamaynaini, M.; Hayat, M.M. Cascading Failures in Interdependent Infrastructures: An Interdependent Markov-Chain Approach. *IEEE Trans. Smart Grid* 2016, 7, 1997–2006. [CrossRef]

23. Lillicrap, T.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* 2016, arXiv:1509.02971.

24. Silver, D.; Lever, G.; Heess, N. Deterministic Policy Gradient Algorithms. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014.

25. 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.