IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON CLOUD - FOG - EDGE COMPUTING IN CYBER-PHYSICAL-SOCIAL SYSTEMS (CPSS)

Received January 15, 2020, accepted January 30, 2020, date of publication February 6, 2020, date of current version February 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2972032

A Cloud-Fog-Edge Closed-Loop Feedback Security Risk Prediction Method

QIANMU LI[1,2,3,4,5], YOUHUI TIAN[2], QIANG WU[6], QI CAO[7], HAIYUAN SHEN[4], AND HUAQIU LONG[1]

[1]Intelligent Manufacturing Department, Wuyi University, Jiangmen 529020, China
[2]Jiangsu Vocational Institute of Commerce, Nanjing 211168, China
[3]School of Cyber Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China
[4]Jiangsu Zhongtian Technology Company, Ltd., Nantong 226463, China
[5]Jiangsu Graduate Workstation of Nanjing University of Science and Technology, Nanjing Liancheng Technology Development Company, Ltd., Nanjing 210008, China
[6]School of Information Mechanical and Electrical Engineering, Jiangsu Open University, Nanjing 210036, China
[7]Academy of Science and Technology Strategic Consulting, Chinese Academy of Sciences, Beijing 100190, China

Corresponding authors: Qianmu Li (qianmu@njust.edu.cn) and Qi Cao (caoqi@casisd.cn)

This work was supported in part by the 2019 Industrial Internet Innovation and Development Project from the Ministry of Industry and Information Technology of China, in part by the 2018 Jiangsu Province Major Technical Research Project ''Information Security Simulation System'', Fundamental Research Funds for the Central Universities under Grant 30918012204, and in part by the Military Common Information System Equipment Pre-research Special Technology Project under Grant 315075701.

ABSTRACT In recent years, with the opening of the ''smart age'' curtain, smart devices dominated by technologies such as robots, drones, and intelligent perception have gradually moved to the center of the Intelligent CPSS stage. However, the new security risks of the Intelligent CPSS have also become increasingly prominent. Especially in recent years, in Ukraine and Venezuela's power attack incidents, a series of related attacks always occur simultaneously. This is a multi-task compound attack. This paper designs a set of Cloud-Fog-Edge closed-loop feedback security risk prediction strategies for multi-task compound attacks based on the offensive and defensive ideas of intelligent games, combining classified deep Boltzmann machines and Markov time-varying models. This strategy can be used for various types of power intelligent system terminals, and realizes security risk prediction with modularity, interoperability, open interfaces and compliance with open standards. Interoperability with other safety equipment can also be achieved through standardized interfaces to form system security protection capabilities to meet the actual needs of the industry Internet system. Experimental results show that the method is superior to the typical traditional method.

INDEX TERMS Risk Prediction, classification deep Boltzmann machine, Markov time-varying model.

I. INTRODUCTION

The Cyber-Physical-Social Systems (CPSS) includes the Cyber-Physical-Systems (CPS), and further incorporates social information and artificial system information in the virtual space. CPSS extends its research scope to social network systems, which includes systems engineering such as ubiquitous embedded environment perception, dynamic analysis of human organization behavior, network communication, and network control. CPSS enables physical systems with computing, communication, precise control, remote collaboration, and autonomous functions. The Power Intelligent Internet is a typical Intelligent CPSS.

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaokang Wang.

There is no doubt that electric power industries are important to the development of national economy. Therefore, it is extremely necessary to research on the network security of electric power industrial control systems. With the deep integration of power industrial systems and internet technology nowadays, the threat of network attacks on power industrial systems is increasing at a rapid pace. However, different from traditional power control systems, current power industrial internet involves more sophisticated techniques, in which different components are connected through the Internet and exchange the massive data and control information. In such a complicated networking environment, it is rather difficult for a power industrial internet to unify different interfaces, communication modes and equipment together. Especially, the power industrial internet is hard to defense various

security threats coming from both Internet and power industrial control systems.

The closed-loop feedback intelligent system security architecture of Power Intelligent CPSS has the following modules and functions: First, security resource awareness: Under a single system, it can accurately sense the number and status of various resources such as the platform's own security vulnerability scanning, security status monitoring, malicious behavior tracking, and security event prevention. In the case of multiple systems, the position, quantity, and status of multiple system security resources can be accurately sensed. The second is the description of the safety task: the safety task can be described in multiple layers using digital methods to form a qualitative and quantitative safety task list. Ability to break complex safety tasks into clear subtasks. The third is the safety task planning calculation: the safety task and safety resources are associated with each other to maximize the use of safety resources and minimize the feedback time. Safety tasks can be assigned in two ways: ''Person in the loop'' or fully automatic. The fourth is the feedback of the safety task evaluation: after the safety task is confirmed, the related tasks are executed in accordance with the safety task planning, and the evaluation stage evaluates the execution effect of the safety task. At the same time, the assessment results are fed back to the safety mission planning system to take further safety measures or adjust the safety action plan.

At present, security defense models for Power Intelligent CPSS are mainly static and passive. Usually, for security of systems, they will isolate intranet and extranet, and often process the attack events when invasions have occurred. However, with the advent of the big data era, the technology of isolation faces challenges because the network needs to be more open and interactive. Therefore, we need to achieve an active defense for Power Intelligent CPSSs, and improve their safety level.

Taking the power field as an example, with the continuous expansion of the demand for smart grid energy interconnection, the number of collections, storage and control on the user side and the power production site side is growing rapidly, and the power business also presents a trend of diversity and timeliness. The application of Internet of things technology, new sensor technology and machine learning technology is more and more widely, which makes the power intelligent terminal that the power grid operation depends on change to the direction of machine intelligence, perceptual intelligence and computational intelligence. This results in massive data, especially heterogeneous data. This puts forward higher requirements of computing power, storage capacity and network architecture performance for the current large data centralized processing method, which is widely used to transfer all terminal equipment data through network communication to the back-end master station system. This brings new challenges to the stability and real-time performance of business execution process, including three aspects:

1) The massive data transferred from network edge devices to cloud data center increases the load of core cloud server;

2) The delay of processing and the shortage of transmission bandwidth are increased;

3) As a result, the delay of network communication increases, and even the packet loss often occurs.

It can be seen that the existing traditional centralized business model of electric power can no longer fully and efficiently meet the business requirements of all intelligent power systems. In this context, the power edge computing model is generated to improve the scene where the centralized model has shortcomings, forming the Power Intelligent CPSS, as shown in Figure 1. AMI means Advanced Metering Infrastructure. Power Intelligent CPSS increases the ability of application task execution and data caching and analysis processing, and migrates some or all of the original centralized business model computing tasks to the converging edge computing terminal node on the edge of the network, so as to reduce the computing load of the master station system. Through edge computing, the pressure of network bandwidth can be relieved, the efficiency of data processing can be improved, the real-time response speed of business can be accelerated, and the stability and real-time reliability of business execution in intelligent power system can be guaranteed.

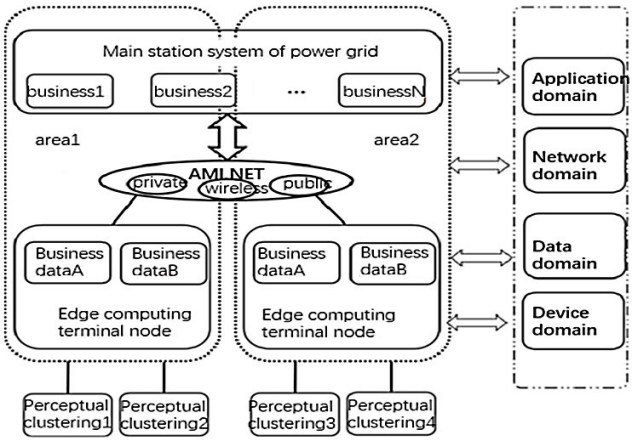

FIGURE 1. Schematic diagram of power intelligent CPSS architecture.

At present, the Power Intelligent CPSS has basically covered the power generation, transmission, transformation, distribution, power consumption, scheduling and other aspects involved in the power business.

Currently, network security prediction is a widely concerned direction in the field of network security, which is the premise and basis for preventing large-scale network attacks. For example, the network situational awareness, proposed by Tim Bass, was used to analyze the network environment, quickly obtain current states, predict future states, and finally provide appropriate responses. Thus, network security risk prediction aims to predict the next state of the network by analyzing the historical states and current states of the network. Because Power Intelligent CPSS requires strict stability and safety, many existing models are too simple to be

adapted to meet the requirements of the network. In addition, the majority of risk prediction methods are designed for public network, which also cannot directly applied to power industrial internets. Thus, it is necessary to develop new models according to the characteristics of power industrial internets.

This paper proposes a status prediction method based on Classification Deep Boltzmann Machine and Markov time-varying Model for the power industrial internet. The method improves efficiency compared with the traditional Boltzmann Machine, and helps with the decision making for active security defense of the electric power control systems.

The remainder of the paper is organized as follows. In Section II, we introduce the related researches about network security prediction. Then in Section III we introduce analysis of security risk prediction method. In Section IV, we propose our research method to predict the status of the power industrial internet. In Section V, the experimental results are presented. Finally, Section VI concludes the paper.

II. RELATED WORK

In the field of network security, network security prediction has received much attention recently, which is the premise and basis for preventing the large-scale network attacks. Currently, research in this field is still in its early stage. Although some studies on the prediction of a single intrusion event [1]–[3] have already been conducted, they cannot provide effective security risk prediction for the future trend of the entire network in general. Integrated attack is one of the main forms of network intrusion. Accurately predicting it becomes a core task of active defense research.

There are many popular data classification and prediction methods in the area of machine learning such as decision tree, SVM, Bayesian network etc. Researches have conducted in-depth research on these methods and improved these methods in different fields. P Li proposed an incremental decision tree algorithm which has a good result in handling multi label problems [4]. A large scale semantic network was proposed to compute term similarity, which is more efficient comparing with existing methods [5]. A cost-sensitive decision tree was proposed to predict the escalation risk of current defect reports for maximum net profit [6]. A novel pool-based cost-sensitive learning method was proposed, which produced a smaller total cost in the cost-sensitive total cost [7].

At present, the problems of network security prediction mainly come from two aspects. On one hand, there are some prediction technologies for single intrusion attacks and compound attacks. An intrusion prediction model based on the fuzzy neural network was proposed to predict the attack intention [1]. Ghosh [2] proposed a LAN-based security prediction model that can predict potential security threats against the analysis of historical vulnerabilities. An Agent-based Bayesian network prediction model [3], based on a multivariate linear statistical model was proposed to predict the security status by statistical analysis of network user behaviors. The above work merely can be used to predict a single intrusion attack but cannot provide a prediction of the overall network security risk. On the other hand, there are some prediction technologies that consider various factors and indicators, which can affect the system security comprehensively and can integrate the network security risk assessment systems, using reasonable prediction methods to predict the security risk trends. A neural network risk prediction method based on radial basis function was proposed [8] to establish an RBF neural network model for network security and risk prediction through a large number of experiments and training. This method is not efficient when dealing the big data volume and has difficulty in the selection of basis function. A network-attack prediction technology based on SVM to predict the time series of network-attack risk assessment indicators through the support vector regression was proposed [9]. However, due to the difficulty in the selection of kernel function, this method is not feasible. Lai *et al.* [10] proposed the algorithm to use the simple weighted fusion method to calculate the current network security risk, and then use the gray theory GM(1,1) model to predict the network security risk in the future. However, this model is not applicable to the case when the change of the risk-value accumulation curve does not meet the exponential growth.

Some network security risk prediction methods have been applied to Power Intelligent CPSS systems. A Markov model with time-varying [11] breaks the assumption in traditional Markov model that the system state transition probability matrix does not change with time, where the state transition probability matrix changes with time. However, this model does not perform well on Power Intelligent CPSS systems because the network model is too simple to satisfy the actual requirements of most Power Intelligent CPSS systems. In addition, this model adopts the Principal Component Analysis (PCA) method in data sampling classification which can reduce the data dimensionality and computational burden efficiently, but to a hydraulic power generation industrial control network system, the requirements for data rigor and accuracy of results are high and PCA cannot maintains the integrity of the sample. SVM has some advantages in solving the problems with small sample, nonlinear and high dimensional features. Thus, it was used to establish a network security assessment system, which divided the network status into five types [12]. Although five types of network status is not too large, the accuracy of SVM is still not high enough to ensure an accurate prediction of network risks which is more important in the power industrial control systems. As a probabilistic knowledge representation and reasoning method, Bayesian network was used to establish a network situation assessment model [13]. Bayesian network can effective handle uncertain information and easily make inferences from incomplete, uncertain and inaccurate information [14], [15]. However, if there are many attributes or the big correlation between the attributes, there will be some problems. For example, the Bayesian network requires more data for analysis and its calculation is very complicated, especially in solving complex problems. A network situation assessment

model based on HMM [16] can dynamically evaluate the risk and improve the calculation formula of the state transition probability distribution, which enhances the adaptability of the algorithm. Although HMM shows some advantages when analyzing real systems, it also has some drawbacks, such as the assumptions of the model are too simple. This model assumes that a state depends only on the state's process and this dependency is time-independent. This assumption does not fully comply with the actual situation of a power industrial control system.

Power Intelligent CPSS is a typical cloud - fog - edge computing in cyber physical social systems (CPSS). At present, the Power Intelligent CPSS has basically covered the power generation, transmission, transformation, distribution, power consumption, scheduling and other aspects involved in the power business. The following paper presents three application scenarios based on cloud fog edge, i.e. intelligent monitoring of substation, intelligent distribution area and intelligent power consumption information collection, as follows:

1) Cloud fog edge application in the field of substation intelligent monitoring. In order to strengthen the condition information collection and remote control of all kinds of high-voltage equipment in the substation, temperature monitoring probes, partial discharge monitoring probes, infrared monitors and even mobile intelligent monitoring robots are widely deployed in the substation, as shown in Figure 2. In the case of large number of substations and large scale of Internet of things equipment in the substation, the traditional centralized transmission, monitoring and analysis business model of the main station cannot be fully applied to intelligent monitoring applications in the substation. In order to reduce the bandwidth load of the transmission network and improve the real-time transmission of the substation monitoring service data, edge computing terminal equipment is deployed on the site side of the substation to analyze and process the information aggregation of the Internet of things equipment in the station. However, if cloud fog edge cannot solve the problem of malicious attack detection and defense of edge computing terminals, it cannot solve the problem of sensitive

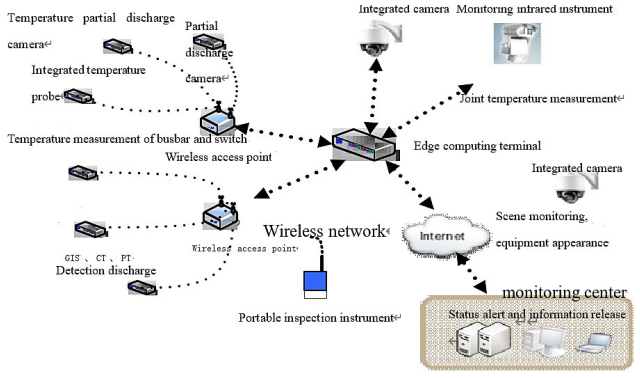

FIGURE 2. Cloud fog edge architecture of substation intelligent monitoring.

collection and monitoring information security transmission, anti-stealing and tampering in cloud fog edge, which will seriously endanger the safe and stable operation of the substation.

2) Cloud fog edge application in the field of intelligent distribution. In order to obtain a more accurate and comprehensive operation state of low-voltage distribution system, so as to implement remote telemetry and remote control for reasonable load allocation. The power distribution business has been upgraded with intelligent transformation. Intelligent devices such as intelligent circuit breakers, intelligent capacitors, multi-type environmental information sensors and phase change switches have been deployed and applied on a large scale, and a large number of measurement and control data have been generated on the low-voltage power field side. Under the requirement of 0.83ms real-time response of power grid control service, the traditional calculation mode of long-distance transmission to centralized master station system cannot meet the service needs at all. For this reason, power grid enterprises have developed a new generation of intelligent distribution and transformation terminal with edge computing function, which performs local aggregation calculation on the edge side data, and transmits part of the processed core data to the new generation of distribution automation master station system, thus forming a typical edge computing scenario supporting real-time business, as shown in Figure 3.

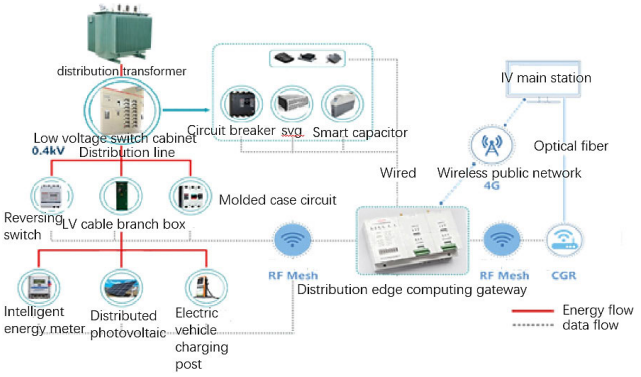

FIGURE 3. Cloud fog edge architecture of intelligent power distribution.

So, there are a lot of security risks in intelligent perception, real-time interaction, ubiquitous interconnection and full business integration.

To solve these problems, this paper proposes a Markov model with time-varying based on classification-limited Boltzmann machine, which improves the performance and reliability of data classification, and optimizes the Markov model with time-varying, and also make it higher in real-time, intelligence, objectivity accuracy and practicability to predict the risk in hydraulic power generation industrial control network.

III. ANALYSIS OF CLOUD-FOG-EDGE SECURITY RISK PREDICTION METHOD

The prediction method of network security risk is to analyze the whole state of various monitoring objects and elements in

the dynamic network environment, and predict the network state in the next stage by analyzing the historical state and current state of the network. The so-called ''state'' here refers to the whole network situation and change trend composed of terminal equipment, communication environment, user behavior in the network and many other factors. This ''state'' is often characterized by integrity, variability, complexity, uncertainty and diffusion. The source of ''state'' is network management equipment, network security equipment and network monitoring equipment. Through data fusion and data processing, data that can reflect the running state of the network can be generated and presented in various forms, so as to identify malicious behaviors of the network, and then make early warning and judgment of possible network threats, and at the same time, send out alarms or take active defense measures.

At present, the research of network security situation awareness abroad has gradually turned to practical application from the initial theoretical stage. DARPA, Rand Corporation, cert (Computer Emergency Response Team) of Carnegie Mellon University, Purdue University, and Lincoln Laboratory of MIT all carried out research on specific applications. However, due to the characteristics of the electric power industry control system, the risk prediction method for the electric power industry control system is still lacking. The existing risk prediction methods for the electric power industry control system mainly focus on the public information security theory. In the era of rapid development of smart grid and emerging of big data technology, traditional security risk prediction methods cannot fully adapt to and meet the needs of power industry control system for risk prediction perception. It is necessary to introduce data mining and artificial intelligence methods.

A. SUPPORT VECTOR MACHINE

Support vector machine (SVM) is a supervised learning method, which was proposed by Vapnik and Cortes in 1995. Through the way of increasing dimension and linearization, SVM cleverly transforms the non-linear problem in the original sample space into the linear separable problem in the high latitude space, which can well carry out pattern recognition, classification, regression analysis, etc.

The basic idea of support vector machine is: first, in the case of linear separability, we find the optimal hyperplane of two kinds of samples in the original space. In the case of indivisibility of linearity, the relaxation variable is added to analyze. By using nonlinear mapping, the samples of low-dimensional input space are mapped to high-dimensional attribute space to make it linear, which makes it possible to use linear algorithm to analyze the nonlinear of samples in high-dimensional attribute space, and to find the optimal classification hyperplane in the feature space. Secondly, it uses the principle of structural risk minimization to construct the optimal classification hyperplane in the attribute space, so that the classifier can get the global optimization, and the expected risk in the whole sample space satisfies a certain upper bound with a certain probability.

The principle of data classification using SVM can be described as follows:

$$T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\} \tag{1}$$

where $x_i \in X = k^n$ is the index vector of input and its component is called index; $y_n \in y = \{-1, +1\}$ is the output and $i = 1, 2, \ldots, n$. We call the set of n samples training set. For any given new pattern x, it can be concluded that its corresponding output y is 1 or -1. This problem is transformed into finding a rule that divides the point of R^n into two parts.

As shown in Figure 4,For a set of positive and negative samples of training data, $\{x_i, y_i\}, i = 1, \cdots, l$, supposes that hyperplane $H : w \cdot x + b = 0$ can accurately separate the samples, and there are two hyperplanes H_2:$w \cdot x + b = 1$ and $H_2 : w \cdot x + b = -1$ parallel to H, so that the samples closest to hyperplane H are on H_1 and H_2. These samples are called support vectors. All the other training samples were outside of J and K.

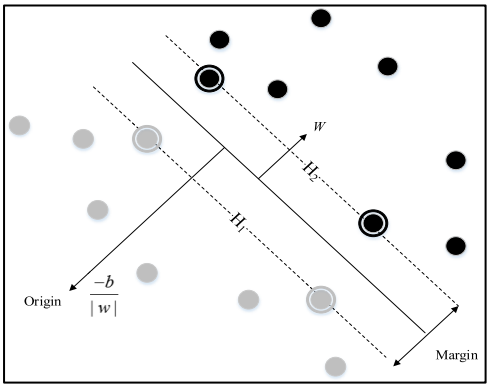

FIGURE 4. Support vector machine.

Literature [17] proposes a network security evaluation system is established by SVM. There are many indicators related to the network security evaluation system, including the network in the state of no attack and the state of attack. According to the experiment, this paper selects 12 typical indexes to establish the evaluation system: CPU occupancy, memory occupancy, port traffic, packet loss rate, available network bandwidth, average round-trip delay, transmission rate, throughput, service request rate, service response rate, error rate and response time, and other more reasonable evaluation indexes. In this paper, the security situation of the network is divided into five warning levels, and the whole network security state is marked from high to low. The process of network security situation assessment using SVM is to establish the relationship between the factors that affect the network security situation and the assessment category. When multiple network status detection values are input, the system can finally output an assessment category accurately. In this paper, the evaluation process is regarded as a multi class

classification problem. In the face of multi class problems, it is necessary to construct multi class SVM classifier. Too few classifiers will lead to inaccurate classification and coarse classification particles; if too many classifiers, it will also lead to complex calculation and slow decision-making. Since the literature divides the network security situation into five levels, it is necessary to construct five two categories of classifiers to transform the five categories of classification problems into five two categories of SVM classification problems. During training, the samples corresponding to n classes are positive, and the rest are negative. After training these five training sets, we get five classifiers: SVM1, SVM2, svm3, svm4 and svm5. Therefore, the decision function is as follows:

$$f_m(x) = sign\left(\sum_i a_i^* y_i K(x_i, x) + b^*\right) \quad m = 1 \ldots 5 \tag{2}$$

Support vector machine can rely on small samples to learn, which has the advantages of strong generalization ability, easy training, no local minimum and so on. In order to overcome the computational difficulties caused by high-dimensional space, SVM uses a variety of clever kernel functions to avoid explicit nonlinear mapping, mapping the input vector in low-dimensional space to high-dimensional space, and building a hyperplane in high-dimensional space. SVM has its own unique advantages in solving the problems of small sample, nonlinear and high-dimensional pattern recognition. However, as discussed in the previous chapters, SVM always has the problems of training difficulty, resource consumption and low efficiency in the face of large-scale training samples and solving multi classification problems. Although the network state is divided into five types, the number of classifications is not particularly large, but the accuracy of SVM classification is not high and it cannot ensure the availability of network risk prediction very well. In the power industry control system in such a system environment, the importance of this problem is particularly prominent.

B. BAYESIAN NETWORK

Bayesian network is a probability graph model. Bayesian network is developed on the basis of Bayes method. It provides a good way to express the probability of causality between variables. Bayesian network is composed of directed acyclic graph (DAG), which includes nodes and directed edges. Nodes represent random variables (including parent nodes and child nodes), and directed edges represent mutual relationships between nodes (from parent nodes to child nodes), and conditional probability is used to express relationship strength, so as to establish, express and analyze uncertainty and probability Pieces.

Definition 1 (Conditional Probability): The conditional probability of event B under the condition that event A occurs. If the basic event A, B, has $P(A) > 0$, then:

$$P(B|A) = \frac{P(AB)}{P(A)} \tag{3}$$

Definition 2 (Joint Probability): Let A and B be two events, and $P(A) > 0$, then their joint probability is:

$$P(AB) = P(B|A)P(A) \tag{4}$$

Definition 3 (Bayesian Theorem): Also known as a posterior probability formula. If the prior probability is $P(B_i)$ and the new additional information obtained from the survey is $P(A|B_i)$, where $i = 1, 2, \cdots n$, the posterior probability is:

$$P(B_i|A) = \frac{P(A|B)\,P(B_i)}{\sum_{j=1}^{n} P\left(A|B_j\right) P\left(B_j\right)} \tag{5}$$

Definition 4 (Directed Graph): It is a binary $G = G(V, E)$ represented by node set V and edge set E. if $(x, y) \in E$ indicates that there is a directed edge from node x to node y, then node x is the parent node of y and node y is the child node of x. Ancestor and successor can be defined by recursive definition of parent and child nodes. In particular, a root node is a node that has no parent.

Definition 5 (Directed Acyclic Graph (DAG)): Digraphs without loops.

Let $G = (V, E)$ represent a DAG, where V represents the set of all nodes in the graph, E represents the set of directed connecting line segments, and $X = (X_v)$, $v \in V$ is the random variable represented by a node v in the directed acyclic graph, if the joint probability of node X is expressed as:

$$P(x) = \prod_{v \in V} P(x_v|x_{pa(v)}) \tag{6}$$

Then we call X a Bayesian network relative to the directed acyclic graph G, where $pa(v)$ is called the ''cause'' or parents of the node v. As shown in Figure 5 is a simple Bayesian network.

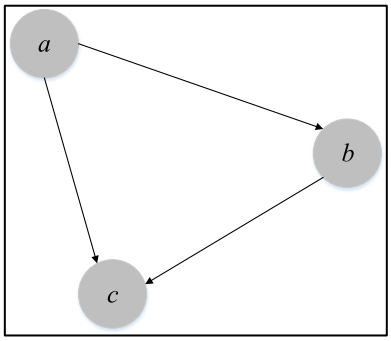

FIGURE 5. Bayesian Network.

In [13], the Bayesian network method is used to establish the network situation evaluation model. The network situation assessment indexes used in the literature are divided into three major indexes: asset index, threat index, and vulnerability index, which can also be subdivided into sub categories. Two types of nodes are established in the literature: one is the observation node, whose state can be obtained directly; the other is the hidden node, whose state must be obtained by reasoning. The structure graph G expresses the causality among the evaluation indexes, and the local probability distribution Θ expresses the strength of the relationship among the

evaluation indexes, which is expressed by conditional probability. It is known that Y_t can get X_t, and the goal of reasoning is to get the required probability through observation $y_{1:t}$. In the network security risk assessment model, the interested node is X_1, X_2, X_3, X_4, and the goal is to accurately infer the probability of the hidden node by observing the state of the variable $Y_1, \ldots, Y_m, Y_n \ldots$.

As a knowledge representation and reasoning method based on probability, Bayesian network can deal with uncertain information effectively and intuitively, and can easily make reasoning from incomplete, uncertain and inaccurate information. However, due to the characteristics of Bayesian networks, if the number of attributes is relatively large or the correlation between attributes is relatively large, there will be problems. At the same time, Bayesian networks need more data, analysis and calculation are more complex, especially in solving complex problems, this contradiction is more prominent.

C. HIDDEN MARKOV

Markov process is a kind of stochastic process. Hidden Markov model (HMM) was originally used by L. E. Baum et al. To describe Markov process with hidden unknown parameters and determine the hidden parameters associated with it from the observable parameters. HMM model has played an important role in many fields, such as natural language processing, computer vision, fault analysis, biological information processing, speech recognition and so on.

The definition of Markov model is as follows:

Set the system to have $S_1, S_2, \ldots S_n$ status, and the system can change from one status to another. Let q_t be the state of the system at time t, and the probability of the state at time t is related to the state of the system at time $1, 2 \ldots t-1$, and the probability is:

$$P(q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_{k,} \ldots) \tag{7}$$

If the state of the system at t time is only related to the state of *t-1* time, it is a discrete Markov process:

$$P\left(q_t = S_j \,|\, q_{t-1} = S_i, q_{t-2} = S_{k,} \ldots\right) = P(q_t = S_j | q_{t-1} = S_i) \tag{8}$$

If only stochastic processes independent of time t are considered:

$$P(q_t = S_j | q_{t-1} = S_i) = a_{i,j}, \quad 1 \le i, j \le N \tag{9}$$

where $a_{i,j} >= 0$, $\sum_{j=1}^{N} a_{i,j} = 1$, it is a Markov model.

The events observed in HMM are random functions of States, in which the state transition process is a hidden Markov chain, while the observable events are random functions of the hidden state transition process.

An HMM can be expressed as $\lambda = (N, M, \pi, A, B)$ and its parameter meanings are as follows:

N: number of states in the model. Set of state $S = \{s_1, s_2, \ldots, s_m\}$. Note that the states of N are $\theta_1, \cdots, \theta_N$ and the state of Markov chain is q_t, obviously $q_t = \{\theta_1, \theta_2, \ldots, \theta_m\}$.

M: the number of possible observations for each state. Observation symbol set $V = \{v_1, v_2, \ldots, v_m\}$.

T: length value of observation sequence, including observation sequence $O = \{o_1, o_2, \ldots, o_m\}$. Let the observed value at any time be o_t, where $o_t \in \{v_1, v_2, \ldots, v_m\}$.

π: initial state probability $\pi = (\pi_1, \cdots, \pi_N)$, then

$$\pi_i = P(q_1 = \theta_i) \tag{10}$$

A: is the time independent state transition probability matrix $(a_{ij})_{N*N}$, where

$$a_{ij} = P\left(q_{t+1} = \theta_j \,|q_t = \theta_i\right) \quad 1 \le i, j \le N \tag{11}$$

B: probability matrix $(b_{ik})_{N*N}$ of observed value, where

$$b_{jk} = P\left(o_t = V_k \,|q_t = \theta_j\right) \quad 1 \le j \le N, \ 1 \le k \le M \tag{12}$$

The process of HMM is as follows:

(1) According to the probability of initial state distribution π, set the initial state and make $n = 1$.

(2) According to B, the probability distribution S_i $(n = 1)$ of output in b_{m1} $(n = 1)$ state is obtained.

(3) According to A, the next state is obtained from the probability distribution of n state transition from S_i state at $n = n + 1$ time to S_j state, and $n = n + 1$ is collocated.

(4) If $n < N$, go back to step 2, otherwise end.

Although hidden Markov model shows great value in the analysis of real system, it also has some disadvantages. One of the biggest disadvantages is that the assumption of the model is too simplified. Because the model assumes a state only depends on the state in its process, and this kind of dependence is time independent. This assumption is not completely in line with the actual situation of the electric power industrial control system, because the electric power industrial control system requires high accuracy of prediction, and this time independent assumption is too simple to be directly applied to the special scene of electric power industrial control.

The advantages and disadvantages of the above methods are summarized as follows:

Through the discussion of the above common methods and the comparison of practical application, in order to better serve the Power Intelligent CPSS, the Boltzmann machine and Markov time-varying model are selected.

IV. METHODOLOGY

A Markov time-varying model (MTM) was used to predict the probability of real-time network security risks [11]. However, their model is relatively simple and cannot be applied to power industrial control systems. First, their network model is so simple that cannot reflect the actual situation of power industrial control systems. Second, their classification model for the network state is also weak. Although the computation complexity of their method is reduced by the cohesion and Principal Components Analysis, the accuracy cannot be guaranteed when applied to the industrial control

TABLE 1. Method comparision.

Method name	Method Summary	Advantage	Disadvantage
Support vector machine	A method of supervised learning	Learning with small samples, strong generalization ability, easy training, no local minima	In the face of large-scale training samples and multi classification problems, training is difficult, resource consumption is too large and efficiency is too high.
bayesian network	A probability pattern model	It can deal with uncertain information effectively and intuitively, and can easily make reasoning from incomplete, uncertain and inaccurate information	When the number of attributes is large or the correlation between attributes is large, the accuracy is low, and the analysis and calculation are complex.
Hidden Markov	Markov processes with implicit unknown parameters	The result of process state prediction is good	The model assumes that a state only depends on the state in its process, and this dependence is time independent

systems of power generation, where outliers in application scenarios cannot be simply ignored. Therefore, this paper still follows the train of thought of ''time-varying'' theory and proposed a novel network model to make it more in line with the application requirements of power industrial control systems. Furthermore, this paper introduces the Classification Deep Boltzmann machine (ClassRBM). Thus, a Markov model with time-varying based on the classification Deep Boltzmann machine (ClassRBM & MTM) is proposed for hydraulic power generation industrial control systems.

A. BASIC STRUCTURE OF A DEEP BOLTZMANN MACHINE

The Boltzmann Machine (BM) was proposed by Hinton and Sejnowski as a generative stochastic neural network [18]. The neurons in the network are random, and the output of a neuron has only two states (active and inactive states), represented by binary zeros and ones. The value of the state is determined according to the law of probability and statistics. The Boltzmann machine is a feedback neural network composed of random connections of random neurons, which is symmetrically connected and has no self-feedback. It contains a visible layer and a hidden layer as shown in Figure 8. The main purpose of the Boltzmann machine is to generate a neural network that models the probability distribution of the input data. In general, it is impossible to get a complete model unless the number of hidden layers in the index of the visible unit. Although the Boltzmann machine has a very strong unsupervised learning ability of learning complicated rules from data, the training (learning) time is rather long. Moreover, it is difficult to obtain both the distribution of the Boltzmann machine and random samples which obeying the distribution. A Deep Boltzmann machine could solve these difficulties.

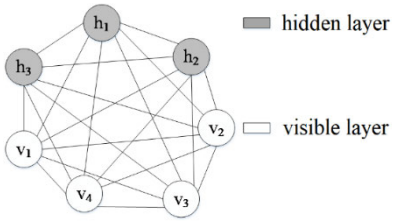

FIGURE 6. Structure of the Boltzmann machine.

The architecture of the Deep Boltzmann Machine (RBM) is very powerful and is based on the principle of the Boltzmann machine. Paul Smolensky proposed the concept of the Deep Boltzmann machine [19], which limits the original inter-layer connections of the Boltzmann machine. Different nodes of the same network layer are not connected with each other, and only the nodes between the hidden layer and the visible layer are connected. Thus, the structure is easier to obtain the probability distribution function of RBM. The Deep Boltzmann machine reduces the complexity of the Boltzmann machine, though the structure of the entire network is still based on the undirected graph. However, different from the Boltzmann machine, RBM removes the connections between the hidden layers so that the status of the neurons in hidden layers is conditionally independent.

The Deep Boltzmann machine can be regarded as an undirected graph model [20], [21]. As shown in Figure 7, v is the visible layer, which represents the observation data, h is the hidden layer, which can be regarded as some feature detectors, and W is the connection weight between the two layers. The hidden units and visible units in the Deep Boltzmann machine can obey any exponential family distribution. For example, they can be a Softmax, a Gauss, or a Poisson distribution. For simplicity, we assume that all visible and hidden units are binary variables. That is, for any i, j, we have $v_i \in \{0, 1\}$ and $h_j \in \{0, 1\}$.

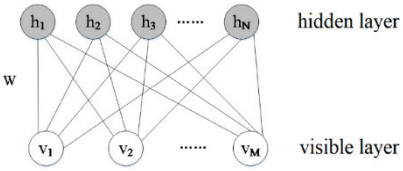

FIGURE 7. Structure of Deep Boltzmann machine.

If a Deep Boltzmann machine has n visible units and m hidden units, then vectors v and h are used to indicate the states of the visible and hidden units respectively, where v_i represents the state of the i^{th} visible unit and h_j represents the state of the j^{th} hidden unit. Then, for a given set of states, the energy of a Deep Boltzmann machine as a system is defined as:

$$E(v, h\,|\theta) = -\sum_{i=1}^{n} a_i v_i - \sum_{j=1}^{m} b_j h_j - \sum_{i=1}^{n}\sum_{j=1}^{m} v_i W_{ij} h_j \tag{13}$$

where $\theta = \{W_{ij}, a_i, b_j\}$ is the parameters, W_{ij} represents the connection weights between the visible unit i and the hidden unit j, a_i represents the bias of the visible unit, and b_j represents the offset of the hidden unit j. When the parameters are determined,, we can obtain the joint probability distribution of (v, h) based on the energy function as follows:

$$P(v, h|\theta) = \frac{e^{-E(v,h|\theta)}}{Z(\theta)} \tag{14}$$

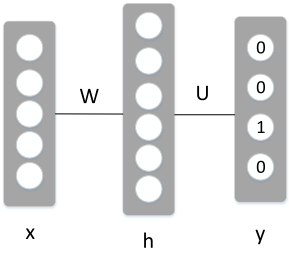

FIGURE 8. Classification Deep Boltzmann machine.

Here, $Z(\theta)$ is a normalization factor (also known as the Partition Function) defined as:

$$Z(\theta) = \sum_{v,h} e^{-E(v,h|\theta)} \tag{15}$$

For a practical problem, we most concern about the distribution $p(v|\theta)$ of the observed data v defined by the Deep Boltzmann machine, i.e. the marginal distribution of the joint probability distribution $p(v, h|\theta)$, also known as the Likelihood Function.

$$p(v|\theta) = \frac{1}{Z(\theta)} \sum_{v,h} e^{-E(v,h|\theta)} \tag{16}$$

To determine this distribution, the normalization factor $Z(\theta)$ needs to be calculated, which requires $O(2^{m+n})$ calculations. Therefore, even if the parameters W_{ij}, a_i, and b_j of the model can be obtained through training, we still cannot effectively calculate the distribution determined by these parameters. However, according to the special structure of the Deep Boltzmann machine, when the states of the visible units are given, the activation states of the hidden units are conditionally independent. At this point, we can obtain the activation probability of the j^{th} hidden as follows:

$$P(h_j = 1|v, \theta) = \sigma(b_j + \sum_i v_i W_{ij}) \tag{17}$$

$\sigma(x) = \frac{1}{1+exp(-x)}$ is a sigmoid activation function.

Since the structure of the Deep Boltzmann machine is symmetric, when the states of the hidden units are given, the activation state of each visible unit is also conditionally independent, that is, the activation probability of the i-th visible unit is:

$$P(v_i = 1|h, \theta) = \sigma(a_i + \sum_j W_{ij} h_j) \tag{18}$$

B. CLASSIFICATION DEEP BOLTZMANN MACHINE

Larochelle [22], [23] proposed the Classification Deep Boltzmann Machine (ClassRBM). When using a Deep Boltzmann machine to solve classification tasks, the most common approach is to treat the Deep Boltzmann machine as a Feature Detector and use the observation data (ignoring the class label) to train the Deep Boltzmann machine. Once the Deep Boltzmann machine has been trained, both the activation probabilities of the hidden units based upon the original training data and the original class labels can be used as a new training set, then we can train the classifier by other common classification algorithms. Since the Deep Boltzmann machine is trained in an unsupervised learning manner, the learned features are not entirely suitable for the classification tasks. The Deep Boltzmann machine can be directly used to solve supervised learning tasks. The main idea of classification Deep Boltzmann machine is to use the hidden units which contains the binary random variables to fit the joint distribution of inputs and class labels. In short, we can treat inputs and class labels as visible units, and train the model. The process of training a classification Deep Boltzmann machine is to generate a three-layer Deep Boltzmann machine network which contains a visual layer, a hidden layer and an output layer, then we use the training data to train the model. Afterwards, the conditional probability distribution between inputs and class labels is established, which can be used to classify the inputs directly.

Compared with Deep Boltzmann machine, the energy function of Classification Deep Boltzmann machine has one more output layer which needs the corresponding weight and bias terms. The classification of Deep Boltzmann machine simplifies the classification process because no additional train the classifier is required, ensuring that the features learned by the algorithm are discriminative. Finally, the training can be performed in an online manner, and the discriminative performance of the learned feature representation can be monitored in real time. Classification Deep Boltzmann machine is a joint distribution model [16]. Let $x = (x_1, \cdots, x_D)$ be the input and $y \in \{1, \cdots, C\}$ be the target class. A hidden layer and uses a binary random unit $h = (h_1, \cdots, h_H)$. All above are integrated in an energy function:

$$E(y, x, h) = -h^T W_x - b^T x - c^T h - d^T e_y - h^T U e_y \tag{19}$$

where $\Theta = (W, b, c, d, U)$ are parameters. W represents the weight matrix of x and h. U denotes the weight matrix of e_y and h. Furthermore, b, c, d are the biases of x, e_y and h respectively. And $e_y = (1_{i=y})_{i=1}^{C}$, it represents an output of y on C. From this energy function, we can obtain the probability values of y, x and h:

$$p(y, x, h) = \frac{exp(-E(y, x, h))}{Z} \tag{20}$$

where Z is a normalized constant (also known as a partition function) that ensures an effective probability distribution for the above formula. Suppose the elements of x are binary, but the elements are directly given when they are expanded

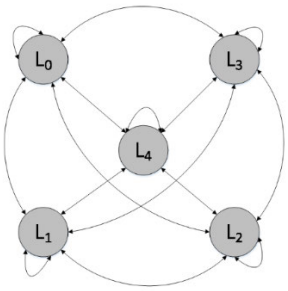

FIGURE 9. Network security risk state transfer.

to actual values on bounded or unbounded intervals. Figure 10 shows the structure of classification Deep Boltzmann machine.

It is usually rather difficult to calculate $p(y, x, h)$ or $p(y, x)$. However, for classification Deep Boltzmann machines, Gibbs sampling can be used [22]. That is, alternating between the current value of a given visible layer and the sampled value of the hidden layer, and vice versa. All the necessary conditional distributions are very simple. When making adjustments on the visible layer, we have:

$$p(h|y, x) = \prod_j p(h_j|y, x) \tag{21}$$

and

$$p(h_j = 1|y, x) = sigm(c_j + U_{jy} + \sum_i W_{ji}x_i) \tag{22}$$

$sigm(a) = 1/(1 + exp(-a))$ is Sigmoid function. When we make adjustments on the hidden layer, we can get the following equation:

$$p(x|h) = \prod_i p(x_i|h) \tag{23}$$

and,

$$p(x_i = 1|h) = sigm(b_i + \sum_j W_{ji}h_j) \tag{24}$$

$$p(y|h) = \frac{exp(d_y + \sum_j U_{jy}h_j)}{\sum_{y*} exp(d_{y*} + \sum_j U_{jy*}h_j)} \tag{25}$$

Based on the above equations, we may accurately calculate $p(y|x)$, thus it is possible to process classification. In fact, we can find that:

$$\begin{aligned}
&\sum_{h_1\in\{0,1\}} \cdots \sum_{h_H\in\{0,1\}} \\
&\quad exp(h^T W_x + b^T x + c^T h + d^T e_y + h^T U e_y) \\
&= exp(d_y) \sum_{h_1\in\{0,1\}} exp(h_1(c_1 + U_{1y} + \sum_i W_{1i}x_i)) \cdots \\
&\quad \sum_{h_H\in\{0,1\}} exp(h_H(c_H + U_{Hy} + \sum_i W_{Hi}x_i)) \\
&= exp(d_y) \left(1 + exp(c_1 + U_{1y} + \sum_i W_{1i}x_i)\right) \cdots \\
&\quad \left(1 + exp(c_H + U_{Hy} + \sum_i W_{Hi}x_i)\right) \\
&= exp(d_y + \sum_j log(1 + exp(c_j + U_{jy} + \sum_i W_{ji}x_i))) \\
&= exp(d_y + \sum_j softplus(c_j + U_{jy} + \sum_i W_{ji}x_i))
\end{aligned} \tag{26}$$

where $softplus(a) = log(1 + exp(a))$. Then we can get that (27), as shown at the bottom of the next page, where $F(y, x)$ is called the free energy function. When calculating the conditional distribution at time $O(HD + HC)$, we can calculate the conditional parameter $c_j + \sum_i W_{ji}x_i$ firstly, and then this conditional parameter will be used again when generating a program calculating function $softplus(c_j + U_{jy*} + \sum_i W_{ji}x_i)$ for all y^* classes.

C. CLASSIFICATION DEEP BOLTZMANN MACHINE BASED ON HYBRID TRAINING

In order to train a classification Deep Boltzmann machine to solve the particular classification problems, we can simply define a target $D_{train} = \{(x_t, y_t)\}$ to minimize all the examples in the training set.

Suppose that we have a model with joint probability $p(x, y)$, the objective function of classification Deep Boltzmann machine training is still defined by maximum likelihood estimation. The input layer is defined as x, the hidden layer is h, and the output layer is y. Then the generated model can be defined by the joint probability distribution

FIGURE 10. Data set event_warning.

maximum likelihood function of x and y:

$$L_{gen}(D_{train}) = -\sum_{t=1}^{|D_{train}|} log\ p(y_t, x_t) \tag{28}$$

Since we are only interested in how to obtain good predictive results for input, we can appropriately ignore the part of unsupervised learning in the generated model and focus on the part of supervised learning.

Based on this premise, we can get the definition of the discriminated model as follows:

$$L_{disc}(D_{train}) = -\sum_{t=1}^{|D_{train}|} log\ p(y_t|x_t) \tag{29}$$

It is obvious that the equation $L_{gen}(D_{train}) = \sum_{t=1}^{|D_{train}|} log\ p(y_t|x_t)$ is to model the posterior conditional distribution, and the discriminated model can be used to train the Deep Boltzmann machine without the generated model. Hybrid training is to combine the generation model and the discriminated model for training. The contribution of the two models can be determined by setting the weights, so:

$$\begin{aligned} L_{hybrid}(D_{train}) &= L_{disc}(D_{train}) + \alpha L_{gen}(D_{train}) \\ &= -(1+\alpha)\sum_{t=1}^{|D_{train}|} log\ p(y_t|x_t) \\ &\quad -\alpha\sum_{t=1}^{|D_{train}|} log\ p(x_t) \end{aligned} \tag{30}$$

Classification can be achieved by adjusting α (that is, balancing choices between bias and variance).

D. MARKOV TIME-VARYING MODEL

The Markov time-varying model (MTM) is defined as follows [11]:

Definition 6: The Markov chain is a sequence of random variables $X_1, X_2, X_3 \cdots$. The range of random variables is called the state space. If the conditional probability distribution of X_{t+k} for past states is only a function of X_t, then:

$$\begin{aligned} &P\left(X_{t+k} = i_{t+k} \,|X_1 = i_1, X_2 = i_2, \cdots, X_t = i_t\right) \\ &\qquad = P\left(X_{t+k} = i_{t+k} \,|X_t = i_t\right) \end{aligned} \tag{31}$$

Here $X_{t+k} = i_{t+k}$ means that the process is in the i_{t+k} state at time $t+k$. The above identity can be seen as a Markov property. That is, the probability distribution of the system state $X_{t+k} = i_{t+k}$ at the time $t+k$ is only related to the state at time t, and is independent of the state before time t.

Definition 7: The Markov chain model can be expressed as (S, P, π), where

(1) S is a non-empty state set consisting of all possible states of the system, namely the state space of the system. For example, the state space of the network is $S = \{1, 2, \cdots, n\}$ where 1 represents the initial state and n represents the final state.

(2) $P = \left[p_{ij}(t, t+k)\right]_{m\times n}$ is the state transition probability matrix of the system, $p_{ij}(t, t+k) = P\{X_{t+k} = j\,|X_t = i\}, i, j \in S$ represents the probability that the system is in state i at time t and is in state j after the k-step state transition. Since the chain starts from any state at time t and goes to another time $t+k$, it must be transferred to one of the state spaces, so for any $i \in S$, we can get $\sum_{j=1}^{n} p_{ij}(t, t+k) = 1, 0 \le p_{ij}(t) \le 1, i, j \in S$.

(3) $\pi = (\pi_1, \pi_2, \cdots, \pi_n)$ is the initial probability distribution of the system. π_i is the probability that the initial moment of the system is in state i, which satisfies $\sum_{i=1}^{n} \pi_i = 1$.

For $p_{ij}(t, t+k) = P\{X_{t+k} = j\,|X_t = i\}, i, j \in S$, when $k = 1, p_{ij}(t, t+k) = p_{ij}(1)$ is called a one-step state transition probability at time t, and $\boldsymbol{P}$ is a one-step state transition probability matrix.

Theorem 8: Let $\{X_t, t = 1, 2, \cdots, n\}$ be a Markov chain, then for any moment u, v, we can get equation (32):

$$P_{ij}(u+v) = \sum_{k=1}^{n} P_{ik}(u) P_{kj}(v), \quad i, j \in S \tag{32}$$

Write the above formula into a matrix form, we can obtain the following equation:

$$\mathbf{P}_{u+v} = \mathbf{P}_u \mathbf{P}_v \tag{33}$$

Corollary 9: The k-step transition probability matrix is the k^{th} power of the one-step transition probability matrix. Which is to say $\mathbf{P}_k = \mathbf{P} \cdot \mathbf{P}_{k-1} = \mathbf{P}^k$.

Proof: Using $\mathbf{P}_k = \mathbf{P} \cdot \mathbf{P}_{k-1} = \mathbf{P}^k$, let $u = 1, v = k-1$, we can get a recursive relationship:

$$\mathbf{P}_k = \mathbf{P}\mathbf{P}_{k-1} = \mathbf{P}\mathbf{P}\mathbf{P}_{k-2} = \ldots = \mathbf{P}^k \tag{34}$$

Let row vector $\pi(k) = ((\pi_1(k), \pi_2(k), \ldots, \pi_n(k))$, where $\pi_j(k)$ represents the probability that the event is in the state j at the k time after the k state transition under the condition that the initial state ($k = 0$) is known. And we can get equation (35) below:

$$\sum_{j=1}^{n} \pi_j(k) = 1 \tag{35}$$

According to the no aftereffect of the Markov process, the Bayesian conditional probability formula and inference,

$$\begin{aligned} p(y|x) &= \frac{\sum_{h_1\in\{0,1\}} \cdots \sum_{h_H\in\{0,1\}} \exp(h^T W x + b^T x + c^T h + d^T e_y + h^T U e_y)}{\sum_{y*\in\{1,\cdots,C\}} \sum_{h_1\in\{0,1\}} \sum_{h_H\in\{0,1\}} \exp(h^T W x + b^T x + c^T h + d^T e_{y*} + h^T U e_{y*})} \\ &= \frac{exp(d_y + \sum_j softplus(c_j + U_{jy} + \sum_i W_{ji} x_i))}{\sum_{y*\in\{1,\cdots,C\}} exp(d_{y*} + \sum_j softplus(c_j + U_{jy*} + \sum_i W_{ji} x_i))} \\ &= \frac{exp(-F(y,x))}{\sum_{y*\in\{1,\cdots C\}} exp(-F(y,x))} \end{aligned} \tag{27}$$

we can find that:

$$\begin{cases} \pi\ (1) = \pi(0)\mathbf{P}_1 = \pi(0)\mathbf{P} \\ \pi\ (3) = \pi(0)\mathbf{P}_2 = \pi(0)\mathbf{P}^2 \\ \cdots \\ \pi(k) = \pi(0)\mathbf{P}_k = \pi(0)\mathbf{P}^k \end{cases} \tag{36}$$

In the formula(36), the initial state probability vector $\pi(0) = (\pi_1(0), \pi_2(0), \ldots, \pi_n(0))$. $\mathbf{P}_i(i = 1, 2, \ldots, k)$ represents the i-step state transition probability matrix of the system.

It can be seen from the above inferences that the traditional Markov prediction model is based on the assumption that the state transition probability matrix of the system does not change with time. However, in many practical problems, especially in network attack environments, the security state of the network is not fixed at different moments, and the transition probability of the state is constantly changing. The Markov model with time-varying improves the accuracy of prediction by continuously updating the state transition probability matrix. From formula (36), we can get equation (37):

$$\begin{cases} \pi\ (1) = \pi(0)\mathbf{P}_{(0)} \\ \pi\ (3) = \pi\ (1)\mathbf{P}_{(1)} \\ \pi\ (4) = \pi\ (3)\mathbf{P}_{(3)} \\ \cdots \\ \pi(k) = \pi(k-1)\mathbf{P}_{(k-1)} \end{cases} \tag{37}$$

where $\mathbf{P}_{(i)}$ represents the state transition probability matrix at time i.

It can be seen from the equations that if the initial state of an event at the 0^{th} time has been known, which means that $\pi\ (0)$ has been known. Then, using the recurrence formula(37), we can find the probability of the event in various possible states at the k^{th} moment after k state transition, i.e. $\pi\ (k)$. Thus, the state probability prediction of the event at the k^{th} time is obtained. Therefore, it is important for the prediction to determine the state transition probability matrix $\mathbf{P}_{(i)}$ at different moments.

E. UPDATE METHOD BASED ON CLASSRBM&MTM

The security risks of the network mainly come from malicious attacks on the network. The attack behavior generally consists of three phases: the information collection phase, the attack progress phase, and the attack completion phase. According to the different stages of the attack, the network security risks are divided into five risk levels: no security risk L_0(that is, the network is in a normal state), low micro security risk L_1 (the network is in the scanned state), micro security risk L_2 (the vulnerability after the network scanning and may be exploited), a more serious security risk L_3 (the network is in an attacked state), and serious security risk L_4 (the network has been severely attacked and has been compromised). These different security risk levels constitute the state space in the Markov time-varying prediction model, i.e. $S = \{L_0, L_1, L_2, L_3, L_4\}$, then the state transition of the network security risk is shown in Figure 9.

The state transition probability matrix of the network security risk can be determined as follows:

$$\mathbf{P} = \begin{pmatrix} p_{L_0L_0}, p_{L_0L_1}, p_{L_0L_2}, p_{L_0L_3}, p_{L_0L_4} \\ p_{L_1L_0}, p_{L_1L_1}, p_{L_1L_2}, p_{L_1L_3}, p_{L_1L_4} \\ p_{L_2L_0}, p_{L_2L_1}, p_{L_2L_2}, p_{L_2L_3}, p_{L_2L_4} \\ p_{L_3L_0}, p_{L_3L_1}, p_{L_3L_2}, p_{L_3L_3}, p_{L_3L_4} \\ p_{L_4L_0}, p_{L_4L_1}, p_{L_4L_2}, p_{L_4L_3}, p_{L_4L_4} \end{pmatrix} \tag{38}$$

Calculating the state transition probability matrix $\boldsymbol{P}$ is to calculate the state transition probability from each state to any other state. The state transition probability is generally calculated using an assumption that the frequency approximately equals to the probability. That is,

$$p_{ij} = \frac{n_{ij}}{\sum_j n_{ij}} \tag{39}$$

where n_{ij} represents the number of samples transferred from state i to state j.

A status update algorithm based on ClassRBM is proposed and its calculation process of updating the state transition probability matrix P is as follows:

Algorithm 1 Status Update Algorithm Based on ClassRBM

Input: Sample Data
Output: status update transition matrix
Step 1: Historical sample data Initialization
Step 2: train Model 1 $\sim$ 5, P(sample status = status | Data), status = {L_0, L_1,L_2, L_3, L_4}
Step 3: *new_sample* $\rightarrow$ *Step*2, calculate risk transfer times
Step 4: update status transfer probability;
Step5: calculate the probability of network security risk status

V. EXPERIMENTS

A. DATA SETS

In order to verify the applicability of the classification Deep Boltzmann machine in power industrial control systems, we performed an experiment on the "event_warning" data set in the national grid technology project. The data set was provided by the project partner and it contains the alarm data of the Power Intelligent CPSS, as shown in Figure 10.

First, according to the type of warning, the degree of harm, the description of different attacks by MIT Lincoln Laboratory Intrusion Detection Experiment and the different stages of attacks, the network security risk levels corresponding to attacks are divided into five classes. We compared with a variety of classification methods in the experiment. The error rates of these methods are shown in Figure 11. The experimental results show that the error rate of the classification Deep Boltzmann machine based on the hybrid training is the lowest among the four methods. It proves that the classification Deep Boltzmann machine has better adaptability in the complex environment of Intelligent CPSS with big data.

To verify the validity of the Markov time-varying model for real-time risk probability prediction, this paper uses the

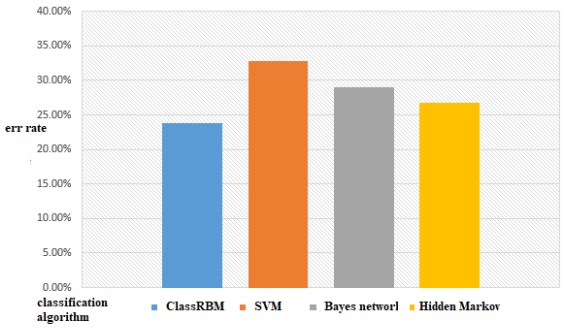

FIGURE 11. Comparison of classification error rates.

published KDD CUP 99 data set for simulation experiments. describes the different attacks. According to the different descriptions of attacks in the MIT Lincoln Labs Intrusion Detection Experiment and different phases of attacks, the network security risk levels corresponding to attacks are classified as shown in Table 2.

TABLE 2. Attack classification.

L0	L1	L2	L3	L4
normal	Portsweep ,insidesniffer ,ipsweep, satan	back, nmap , land, Neptune, pod, smurf	teardrop,rootkit , phf, back, teardrop, buffer_overflow, imap, multihop	guess_passwd,ftp_write, spy,warezclient, warezmaster,perl, loadmodule

Since the KDD data set is too large, only one attack of each attack stage is selected for experiment. The attacks selected in this article are *normal*, *satan*, *back*, *root kit*, and *guess_passwd*. According to the classification of 38 attacks in Table 2, the experiment extracts 2% data in *normal* (1946 records), all data in *satan* (1589 records), all data in *back* (2203 records), all data in *root kit* (10 records) and all data in guess passwd (53 records) as training data. There are 41 qualitative and quantitative attribute features in the KDD CUP 1999 data set, including 8 discrete attribute variables and 33 continuous attribute variables. The training data is divided into five categories, represented by C_1, C_2, C_3, C_4, and C_5, representing five safety risk states L_0, L_1, L_2, L_3, and L_4, respectively. The class analysis results are showed in Table 3.

TABLE 3. Analysis results for classes.

category	#samples in class	#original samples	error rate
C1	1923	1946	1.18%
C2	1606	1589	1.07%
C3	2211	2203	0.36%
C4	9	10	10.00%
C5	52	53	1.89%

Due to the use of the classification Deep Boltzmann machine based on the hybrid training method, the classification accuracy is improved, and the problem of the traditional Deep Boltzmann machine was also solved. The proposed method improved the efficiency, where α is 0.01 and H is 1500.

Compared with the traditional methods, the feature extraction and classification methods proposed in this paper are almost the same in training time, which indicates that the Deep Boltzmann machine greatly improves the efficiency of the Boltzmann machine. Moreover, the accuracy of the Deep Boltzmann machine is higher than the traditional classification method. At the same time, the classification Deep Boltzmann machine based on the hybrid training method does not need to use other classification tools because it has the ability to classify and simplifies the process of classification, which makes this method more intelligent and practical.

This paper counted the times of each state transition in the training data, the results are shown in Table 4.

TABLE 4. Status transfer count.

	L0	L1	L2	L3	L4	Total
L0	1936	9	1	0	0	1946
L1	8	1577	4	0	0	1589
L2	4	3	2196	0	0	2203
L3	2	0	0	7	1	10
L4	1	0	0	0	52	53

B. NETWORK STATUS PREDICTION

Initially, the network is in a normal state, so its initial state probability can be considered as $(0) = (1, 0, 0, 0, 0)$. Multiplying $\pi(0)$ by equation (32), we can get $\pi\ (1) = \pi(0)\cdot P$, $P = (0.99486, 0.00462, 0.00051, 0, 0)$ as the current state probability of the network. It means the probability of the network in state L_0 is 0.99486, in state L_1 is 0.00462, in state L_2 is 0.00051, in state L_3 is 0 and in state L_4 is 0. It can be seen that the network is most likely in a state without security risk, then is low micro security risk state and micro security risk state, and the result is consistent with the actual situation.

Using the transition probability of each state calculated by equation (32), the initial risk state transition probability matrix is obtained as follows:

$$\mathbf{P} = \begin{pmatrix} 0.99486 & 0.00462 & 0.00051 & 0 & 0 \\ 0.00503 & 0.99245 & 0.00252 & 0 & 0 \\ 0.00182 & 0.00136 & 0.99682 & 0 & 0 \\ 0.2 & 0 & 0 & 0.7 & 0.1 \\ 0.01887 & 0 & 0 & 0 & 0.9811 \end{pmatrix} \quad (40)$$

The experiment extracted the data set consisting of *normal, satan, back, root kit and guess_passwd* from the remaining 90% of the data in kddcup.data.txt as test data, and tested them in 5 groups.

Time T_1: input the first set of experimental test samples, which consists of 1000 normal sample data;

Time T_2: Next, input the second set of experimental test samples, consisting of 1000 satan sample data;

Time T_3: Then input the third set of experimental test samples, consisting of 1000 back sample data;

Time T_4: Then input the fourth set of experimental test samples, consisting of 10 root kit sample data;

Time T_5: Finally, input the fifth set of experimental test samples, which consists of 50 guess_passwd sample data.

When a new test sample arrives, the state transition probability matrix update algorithm is processing, and the classification Deep Boltzmann machine is used to classify the class, then recalculates the state transition probability matrix by the risk state transition probability matrix calculation method. Finally, we use equation (32) to predict the probability of future networks at various security risk levels. Table 5 to Table 9 show the statistics of the number of state transitions for the 5 groups of test samples, respectively.

TABLE 5. Statistics of the number of normal sample state transitions.

	L0	L1	L2	L3	L4	Total
L0	965	1	0	0	0	966
L1	2	2	0	0	0	4
L2	0	0	0	0	0	0
L3	0	0	0	0	0	0
L4	0	0	0	0	0	0

TABLE 6. Statistical results of satan sample state transition times.

	L0	L1	L2	L3	L4	Total
L0	0	1	0	0	0	1
L1	0	990	1	0	0	991
L2	0	0	0	0	0	0
L3	0	0	0	0	0	0
L4	0	0	0	0	0	0

TABLE 7. Statistical results of back sample state transition times.

	L0	L1	L2	L3	L4	Total
L0	0	0	0	0	0	0
L1	0	0	1	0	0	1
L2	0	0	980	2	1	983
L3	0	0	0	0	0	0
L4	0	0	0	0	0	0

TABLE 8. Statistical results of rootkit sample state transition times.

	L0	L1	L2	L3	L4	Total
L0	0	0	0	0	0	0
L1	0	0	0	0	0	0
L2	0	0	0	1	0	1
L3	0	0	1	8	0	9
L4	0	0	0	0	0	0

TABLE 9. Statistics results of guess_passwd sample state transition times.

	L0	L1	L2	L3	L4	Total
L0	0	0	0	0	0	0
L1	0	0	0	0	0	0
L2	0	0	0	0	0	0
L3	0	0	0	0	1	1
L4	0	0	0	1	46	47

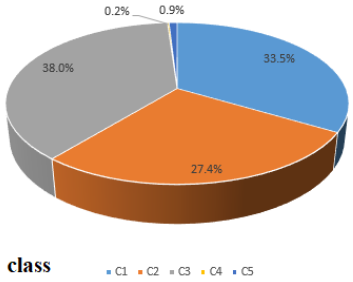

FIGURE 12. Distribution of various class of data.

The new state transition probability matrices of the 5 groups can be obtained respectively.

$$\mathbf{P}_{(1)} = \begin{pmatrix} 0.99896 & 0.00104 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{41}$$

$$\mathbf{P}_{(2)} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0.99899 & 0.00101 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{42}$$

$$\mathbf{P}_{(3)} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0.99695 & 0.00203 & 0.00102 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{43}$$

$$\mathbf{P}_{(4)} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0.11111 & 0.88889 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{44}$$

$$\mathbf{P}_{(5)} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0.02128 & 0.97872 \end{pmatrix} \tag{45}$$

Table 10 gives the prediction results of the Markov model with time-varying.

The prediction results of the time-varying Markov model based on the classification Deep Boltzmann machine are shown in Figure 13.

It can be seen from the experimental results that the type of attack predicted by the Markov time-varying model based on the classification Deep Boltzmann machine is faster and more accurate. The input data shows that the test data of the network at each moment has the same type, so the risk status of the network at each moment should be L_0, L_1, L_2, L_3 and L_4 in turn. Since the state transition probability matrix of the Markov time-varying model based on the classification Deep Boltzmann machine is updated in real time with the addition of samples, the network state probability obtained

TABLE 10. Markov time-varying model prediction results.

Last moment state probability	Updated state transition probability matrix	Risk probability prediction				
		L_0	L_1	L_2	L_3	L_4
(1,0,0,0,0)	$\begin{pmatrix} 0.99486 & 0.00462 & 0.00051 & 0 & 0 \\ 0.00503 & 0.99245 & 0.00252 & 0 & 0 \\ 0.00182 & 0.00136 & 0.99682 & 0 & 0 \\ 0.2 & 0 & 0 & 0.7 & 0.1 \\ 0.01887 & 0 & 0 & 0 & 0.9811 \end{pmatrix}$	0.99486	0.00462	0.00051	0	0
(0.99486,0.00462, 0.00051,0,0)	$\begin{pmatrix} 0.99896 & 0.00104 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$	0.99614	0.00334	0	0	0
(0.99614, 0.00334,0,0,0)	$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0.99899 & 0.00101 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$	0	0.99948	0	0	0
(0,0.99948,0,0,0)	$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0.99695 & 0.00203 & 0.00102 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$	0	0	0.99948	0	0
(0,0,0.99948,0,0)	$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0.11111 & 0.88889 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$	0	0	0	0.99948	0
(0,0,0,0.99948,0)	$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0.02128 & 0.97872 \end{pmatrix}$	0	0	0	0	0.99948

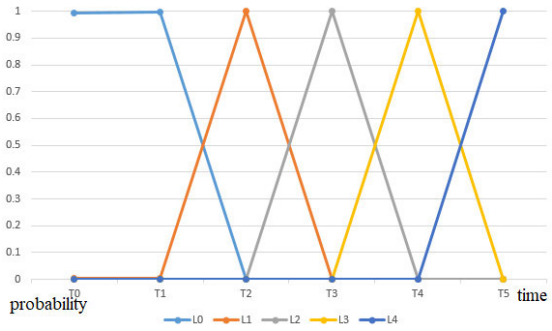

FIGURE 13. Risk probability prediction of Markov time-varying model based on classification deep Boltzmann machine.

each time is constantly updated and changed. This makes the prediction of risk more accurate and objective. The simulation experiments show that when using the Markov time-varying model based on the classification Deep Boltzmann machine, the network has the highest risk probability in the L_0 state at time T_1; has the highest risk probability in the L_1 state at time T_2; has the highest risk probability in the L_2 state at time T_3; has the highest risk probability in the L_3 state at time T_4 and has the highest risk probability in the L_4 state at time T_5, which coincides with the test data. At the same time, since the model uses the technique of classification Deep Boltzmann machine to classify the data, the classification method of the original Markov time-varying model is changed, and the classification speed and accuracy are improved. This model has a significant improvement in the complex network environment of power generation industrial control systems.

VI. CONCLUSION

Facing the complexity of Power Intelligent CPSS and their network environments, this paper proposes a Cloud-Fog-Edge closed-loop feedback security risk prediction model, a Markov model with time varying based on classification Deep Boltzmann machine, for Power Intelligent CPSS systems to predict the network security risk accurately in real time. This model uses a classification Deep Boltzmann machine based on hybrid training to extract the feature and

classify the data smartly. It not only adapts the working environment of power industrial control systems, but also solves the defects of the original Markov model with time-varying. Consequently, it outperforms the traditional classification algorithms and inherits the advantages of the Markov model with time-varying. Moreover, the probability of the security risk state of the network at a future time is predicted by updating the state transition probability matrix in this model. In a word, this model has high real-time, intellectuality, practicability, objectivity and accuracy in the field of network risk prediction.

REFERENCES

[1] G. Zhang and J. Sun, "A novel network intrusion attempts prediction model based on fuzzy neural network," in *Computational Science* (Lecture Notes in Computer Science), vol. 3991. Berlin, Germany: Springer, 2006, pp. 419–426.

[2] S. Bhattacharya and S. Ghosh, "Security threat prediction in a local area network using statistical model," in *Proc. IEEE Int. Parallel Distrib. Process. Symp.*, Mar. 2007, pp. 1–8.

[3] X. Qin and W. Lee, "Attack plan recognition and prediction using causal networks," in *Proc. 20th Annu. Comput. Security Appl. Conf.*, Apr. 2005, pp. 370–379.

[4] P. Li, X. Wu, X. Hu, and H. Wang, "An incremental decision tree for mining multilabel data," *Appl. Artif. Intell.*, vol. 29, no. 10, pp. 992–1014, Nov. 2015.

[5] P. Li, H. Wang, K. Q. Zhu, Z. Wang, X. Hu, and X. Wu, "A large probabilistic semantic network based approach to compute term similarity," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 10, pp. 2604–2617, Oct. 2015.

[6] V. S. Sheng, B. Gu, W. Fang, and J. Wu, "Cost-sensitive learning for defect escalation," *Knowl.-Based Syst.*, vol. 66, pp. 146–155, Aug. 2014.

[7] V. S. Sheng, "Studying active learning in the cost-sensitive framework," in *Proc. 45th Hawaii Int. Conf. Syst. Sci.*, Jan. 2012, pp. 1097–1106.

[8] W. Ren, J. X. Hao, and S. TanFeng, "RBFNN-based prediction of networks security situation," *Comput. Eng. Appl.*, vol. 42, no. 31, pp. 136–138, 2006.

[9] Z. xiang, H. C. Zhen, and L. S. Hang, "Research on network attack situation forecast technique based on support vector machine," *Comput. Eng.*, vol. 11, pp. 10–12, Nov. 2011, doi: 10.1016/j.cageo.2006.02.011.

[10] L. Jibao, W. Huiqiang, and Z. Liang, "Study of network security situation awareness model based on simple additive weight and grey theory," in *Proc. Int. Conf. Comput. Intell. Secur.*, vol. 2, Nov. 2006, pp. 1545–1548.

[11] L. Gang, L. Qian-Mu, and L. Feng-Yu, "A time-varying Markov model and its application to network real-time risk probability prediction," *Acta Armamentarii*, vol. 33, no. 2, pp. 163–169, 2012.

[12] L. Chen and T. Liu, "Using support vector machines to achieve the rating of the network security posture," *Comput. Program. Skills Maintenance*, vol. 2011, no. 20, pp. 152–153, 2011.

[13] T. SiSi, "Research on information security risk assessment method based on discrete dynamic Bayesian network," Northeastern Univ., Boston, MA, USA, Tech. Rep. 20130601, 2013.

[14] L. Jiang, C. Li, and S. Wang, "Cost-sensitive Bayesian network classifiers," *Pattern Recognit. Lett.*, vol. 45, pp. 211–216, Aug. 2014.

[15] L. Jiang, S. Wang, C. Li, and L. Zhang, "Structure extended multinomial naive Bayes," *Inf. Sci.*, vol. 329, pp. 346–356, Feb. 2016.

[16] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition,"Proceedings of the IEEE, 1989, vol. 77, no. 2, pp. 257–286.

[17] C. Liuwei, "Design of situation assessment algorithm for network security situation assessment system," Harbin Eng. Univ., Harbin, China, Tech. Rep. 20101213, 2011.

[18] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines," *Cognit. Sci.*, vol. 9, no. 1, pp. 147–169, 1985.

[19] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1. Cambridge, MA, USA: MIT Press, 1986, pp. 194–281.

[20] X. Xu, Q. Liu, X. Zhang, J. Zhang, L. Qi, and W. Dou, "A blockchain-powered crowdsourcing method with privacy preservation in mobile environment," *IEEE Trans. Comput. Soc. Syst.*, vol. 6, no. 6, pp. 1407–1419, Dec. 2019, doi: 10.1109/tcss.2019.2909137.

[21] M. Welling, M. Rosen-Zvi, and G. Hinton, "Exponential family harmoniums with an application to information retrieval," *Zvi*, vol. 17, no. 4, pp. 123–128, 2005.

[22] X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, and W. Dou, "BeCome: Blockchain-enabled computation offloading for IoT in mobile edge computing," *IEEE Trans. Ind. Informat.*, to be published, doi: 10.1109/tii.2019.2936869.

[23] H. Larochelle, M. Mandel, and R. Pascanu, "Learning algorithms for the classification deep Boltzmann machine," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 643–669, 2012.

[24] X. Xu, Y. Xue, L. Qi, Y. Yuan, X. Zhang, T. Umer, and S. Wan, "An edge computing-enabled computation offloading method with privacy preservation for Internet of connected vehicles," *Future Gener. Comput. Syst.*, vol. 96, pp. 89–100, Jul. 2019.

[25] X. Wang, L. T. Yang, X. Xie, J. Jin, and M. J. Deen, "A cloud-edge computing framework for cyber-physical-social services," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 80–85, Nov. 2017.

[26] X. Wang, L. T. Yang, L. Kuang, X. Liu, Q. Zhang, and M. J. Deen, "A tensor-based big-data-driven routing recommendation approach for heterogeneous networks," *IEEE Netw.*, vol. 33, no. 1, pp. 64–69, Jan. 2019.

[27] X. Wang, L. T. Yang, H. Li, M. Lin, J. Han, and B. O. Apduhan, "NQA: A nested anti-collision algorithm for RFID systems," *ACM Trans. Embed. Comput. Syst.*, vol. 18, no. 4, pp. 1–21, Jul. 2019.

[28] X. Wang, L. T. Yang, Y. Wang, X. Liu, Q. Zhang, and M. J. Deen, "A distributed tensor-train decomposition method for cyber-physical-social services," *ACM Trans. Cyber-Phys. Syst.*, vol. 3, no. 4, pp. 1–15, Oct. 2019, doi: 10.1145/3323926.

[29] L. Qi, Q. He, F. Chen, W. Dou, S. Wan, X. Zhang, and X. Xu, "Finding all you need: Web APIs recommendation in Web of things through keywords search," *IEEE Trans. Comput. Soc. Syst.*, vol. 6, no. 5, pp. 1063–1072, Oct. 2019.

[30] H. Liu, H. Kou, C. Yan, and L. Qi, "Link prediction in paper citation network to construct paper correlation graph," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, Oct. 2019, Art. no. 233, doi: 10.1186/s13638-019-1561-7.

[31] L. Qi, X. Zhang, W. Dou, C. Hu, C. Yang, and J. Chen, "A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment," *Future Gener. Comput. Syst.*, vol. 88, pp. 636–643, Nov. 2018.

[32] L. Qi, X. Zhang, W. Dou and Q. Ni, "A distributed locality-sensitive hashing-based approach for cloud service recommendation from multi-source data," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2616–2624, Nov. 2017.

[33] W. Gong, L. Qi, and Y. Xu, "Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment," in *Wireless Communications and Mobile Computing*, vol. 2018. London, U.K.: Hindawi, Apr. 2018, pp. 1–8, doi: 10.1155/2018/3075849.

[34] L. Qi, Y. Chen, Y. Yuan, S. Fu, X. Zhang, and X. Xu, "A QoS-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems," *World Wide Web*. 2019, doi: 10.1007/s11280-019-00684-y.

[35] X. Xu, C. He, Z. Xu, L. Qi, S. Wan, and M. Z. A. Bhuiyan, "Joint optimization of offloading utility and privacy for edge computing enabled IoT," *IEEE Internet Things J.*, to be published, doi: 10.1109/jiot.2019.2944007.

[36] X. Xu, Y. Chen, X. Zhang, Q. Liu, X. Liu, and L. Qi, "A blockchain-based computation offloading method for edge computing in 5G networks," *Softw., Pract. Exper.*, to be published, doi: 10.1002/spe.2749.

QIANMU LI received the B.Sc. and Ph.D. degrees from the Nanjing University of Science and Technology, China, in 2001 and 2005, respectively. He is currently a Full Professor with the School of Cyber Science and Engineering, Nanjing University of Science and Technology, China. His research interests include information security and data mining. He received the China Network and Information Security Outstanding Talent Award, in 2016, and the Education Ministry Science and Technology Awards, in 2012.

YOUHUI TIAN received the master's degree from the Heilongjiang University of Science and Technology. He is currently a Senior Engineer with the Jiangsu Vocational Institute of Commerce. His research area is security communication and information systems.

QIANG WU received the B.Sc. degree in industrial automation major from Chongqing University. He is currently a Senior Engineer with Jiangsu Open University. He is also the Vice President of the School of Information Mechanical and Electrical Engineering. His research area is communication and information systems.

QI CAO was born in Lu'an, Anhui, in October 1984. She is currently an Engineer with the Institute of Science and Technology Strategic Consulting, Chinese Academy of Sciences. Her research area is communication and information systems.

HAIYUAN SHEN was born in 1984. He received the master's degree in 2009. He became a Software Engineer in 2009. He also serves as the Director of Zhongtian Software. He works with the Jiangsu Zhongtian Technology Company, Ltd. His research interests include service computing and cloud computing.

HUAQIU LONG received the B.Sc. degree from the Intelligent Manufacturing Department, Wuyi University, in 2019. He is currently pursuing the part-time master's degree. He is currently with the Cyber Security Laboratory, Wuyi University. He is also a teacher in the university laboratory. His research interests include information security, computing system management, and data mining.