

Research Article

Modeling of Failure Prediction Bayesian Network with Divide-and-Conquer Principle

Zhiqiang Cai, Weitao Si, Shubin Si, and Shudong Sun

Ministry of Education Key Laboratory of Contemporary Design and Integrated Manufacturing Technology, School of Mechatronics, Northwestern Polytechnical University, Xi'an 710072, China

Correspondence should be addressed to Shubin Si; sisb@nwpu.edu.cn

Received 30 December 2013; Revised 2 May 2014; Accepted 2 May 2014; Published 20 May 2014

Academic Editor: Balaji Raghavan

Copyright © 2014 Zhiqiang Cai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For system failure prediction, automatically modeling from historical failure dataset is one of the challenges in practical engineering fields. In this paper, an effective algorithm is proposed to build the failure prediction Bayesian network (FPBN) model with data mining technology. First, the conception of FPBN is introduced to describe the state of components and system and the cause-effect relationships among them. The types of network nodes, the directions of network edges, and the conditional probability distributions (CPDs) of nodes in FPBN are discussed in detail. According to the characteristics of nodes and edges in FPBN, a divide-and-conquer principle based algorithm (FPBN-DC) is introduced to build the best FPBN network structures of different types of nodes separately. Then, the CPDs of nodes in FPBN are calculated by the maximum likelihood estimation method based on the built network. Finally, a simulation study of a helicopter convertor model is carried out to demonstrate the application of FPBN-DC. According to the simulations results, the FPBN-DC algorithm can get better fitness value with the lower number of iterations, which verified its effectiveness and efficiency compared with traditional algorithm.

1. Introduction

With the developments of information and computer technologies, modern systems have become more complex while the relationships among systems have also become more complicated. To fulfil the high demands of system safety, operational efficiency, and life cycle cost, the key objective is to predict the system state and warn of potential failures with the help of advanced methods, which could avoid great loss before failure happens [1].

Failure prediction approaches have been divided into three types, including experience-based, condition-based, and model-based methods [2]. Li et al. [3] performed a reliability analysis with an emphasis on predicting the lifetime of diesel engine's turbocharger, in which the failure mode and the information of criticality are fully utilized. Wang and Jiang [4] evaluated the degradation of complex system performance using complex system condition monitoring information based on support vector machine (SVM). Zhang et al. [5] proposed a particle swarm optimization based SVM model with the characteristics in software reliability

prediction. Although many interesting methods have been proposed for failure prediction, the model-based method has played more important role in engineering fields for its advantages in effectiveness and efficiency.

Data mining, also referred to as knowledge discovery, is defined as the process of extracting nontrivial, implicit, previously unknown, and potentially useful information from databases [6]. With the wide application of maintenance information management systems, operation data can be collected easily. So, many scientists and engineers applied artificial intelligence or statistical methods to establish the failure prediction models. For instance, Chen et al. [7] proposed the manufacturing defect detection method using the technique of associating rule mining. Han et al. [8] used sequential association rules mining to extract the failure patterns and forecast failure sequences of Republic of Korea Air Force aircraft with various combination states of aircraft types, location, mission, and season. Dong [9] described the concepts, models, algorithms, and applications of hidden Markov models and hidden semi-Markov models in engineering asset health prognosis.

Bayesian network (BN) is a directed acyclic graph which can represent uncertain knowledge by describing relationships and influences among variables [10]. Built upon the Bayes' theorem, BN is designed to obtain posterior probabilities of unknown variables from known probabilistic relationships. Moreover, with the help of graphical diagrams consisting of nodes and edges, BN can be understood more easily than many other techniques. So, BN has got great popularity for solving system modelling problems in broad engineering fields owing to its advantages [11, 12]. Particularly for system reliability prediction, Muller et al. [13] formulated a dynamic prognosis BN model with the knowledge from functional and dynamic modelling. Langseth and Portinale [14] proposed a BN modelling framework which could translate standard fault tree to BN. Doguc and Ramirez-Marquez [15] studied a BN construction method for system reliability estimation and provided a step-by-step illustration of the method. Weber and Jouffe [16] developed the dynamic object-oriented BN by integrating system functioning and malfunctioning knowledge. Mahadevan et al. [17] applied BN to structural system reliability reassessment and validated it by analytical comparison.

Generally, it is not easy to build and quantify BN's relationship for practical cases only based on expert opinions, especially for uncertain reliability prediction problems. Because system operation data are abundant in quantity and various in characteristics, this paper introduce an expanded BN model to describe the failure prediction process for complex system under uncertainty and proposes a divide-and-conquer principle based data mining algorithm to build the corresponding model.

The rest of this paper is organized as follows. Section 2 describes the failure prediction Bayesian network (FPBN), including nodes types, edges directions, and conditional probability distributions (CPDs). In order to facilitate the FPBN modeling with failure data, a divide-and-conquer principle based modeling method is proposed in Section 3. With the helicopter convertor case, Section 4 illustrates the application of the proposed FPBN modeling method. Section 5 concludes this study and gives several possible future research topics.

2. Failure Prediction Bayesian Network

By inheriting the advantages of BN, the FPBN is introduced to describe the state of components and system and the cause-effect relationships among them for system failure prediction [18]. A FPBN is also described with $\langle \mathbf{X}, \mathbf{A}, \mathbf{P} \rangle$, where \mathbf{X} represents nodes, \mathbf{A} represents edges, and \mathbf{P} represents CPDs. However, some practical assumptions are built on nodes and edges in FPBN according to the characteristics of failure prediction tasks. So, the FPBN can perform more efficiently than traditional BN in the field of system failure prediction. A simple example is shown in Figure 1.

2.1. Types of Nodes in FPBN. In BN, all the nodes represent variables which have equal status. In FPBN, nodes reflect the state of component or system in practical engineering

systems. According to corresponding role in system failure prediction process, the node set \mathbf{X} is divided into three types of subsets as $\mathbf{X} = \mathbf{C} \cup \mathbf{M} \cup \mathbf{E}$, including failure cause subset \mathbf{C} , failure mode subset \mathbf{M} , and failure detection subset \mathbf{E} . As BN, the values of all nodes in FPBN are discrete and mutual exclusive. For binary systems, each node has two states, functioning as 0 and failure as 1. For multistate systems, there are more failure states within a node which will be represented as $\{0, 1, 2, \dots, n\}$.

2.1.1. Failure Cause Nodes. This type of nodes describes the root causes of certain failure mode. In the failure prediction process, the possible states of a failure cause node could be derived from detected information, reliability estimation, or expert initialization.

2.1.2. Failure Mode Nodes. The failure mode nodes represent the operation states of system which is the final object of failure prediction task. There will be usually only one failure mode node which is the objective of the failure prediction.

2.1.3. Failure Detection Nodes. Failure detection nodes describe the detectable states of certain sensors or lights or alarms. They are affected by the failure cause nodes or failure mode node in FPBN model.

2.2. Directions of Edges in FPBN. When dealing with the system failure prediction problem in practice, maintenance engineers usually use the failure detection information to diagnose the possible state of corresponding failure cause and integrate the failure cause states to estimate the probability of failure mode.

In traditional BN, an edge represents the relationship between any two nodes, while in FPBN, each edge a_{ij} in \mathbf{A} indicates that there are cause-effect relationships between X_i and X_j . X_i is the cause of X_j and X_j is the effect of X_i . In particular, the directions of edges between different types of node subsets are initialized. As shown in Figure 1, the failure mode nodes VL could only be affected by the failure cause nodes (HP, HV); the failure cause nodes (HP, HV) and failure mode nodes VL could be revealed by the failure detection nodes HT. But among the same node subset, such as (HP, HV), there is no restriction on the direction of edge between them. Such edges can only be determined by operation dataset or expert knowledge. The directions of edges for different node subsets in FPBN are consistent with the reasoning process of failure prediction tasks. The practical FPBN is easy to understand by maintenance engineers.

2.3. Conditional Probability Distributions in FPBN. The FPBN has the same meaning of \mathbf{P} with traditional BN. The $\mathbf{P} = \{P(X_i | \pi(X_i)), X_i \in \mathbf{X}\}$ represents the CPD of each node which expresses the intensity of relevance among X_i and its father nodes $\pi(X_i) \subseteq \{X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n\}$. For the root nodes which do not have father node, their CPDs are replaced with corresponding prior probability distributions.

Like a BN, the node is independent of other nodes in FPBN if the states of its all father nodes are known. When

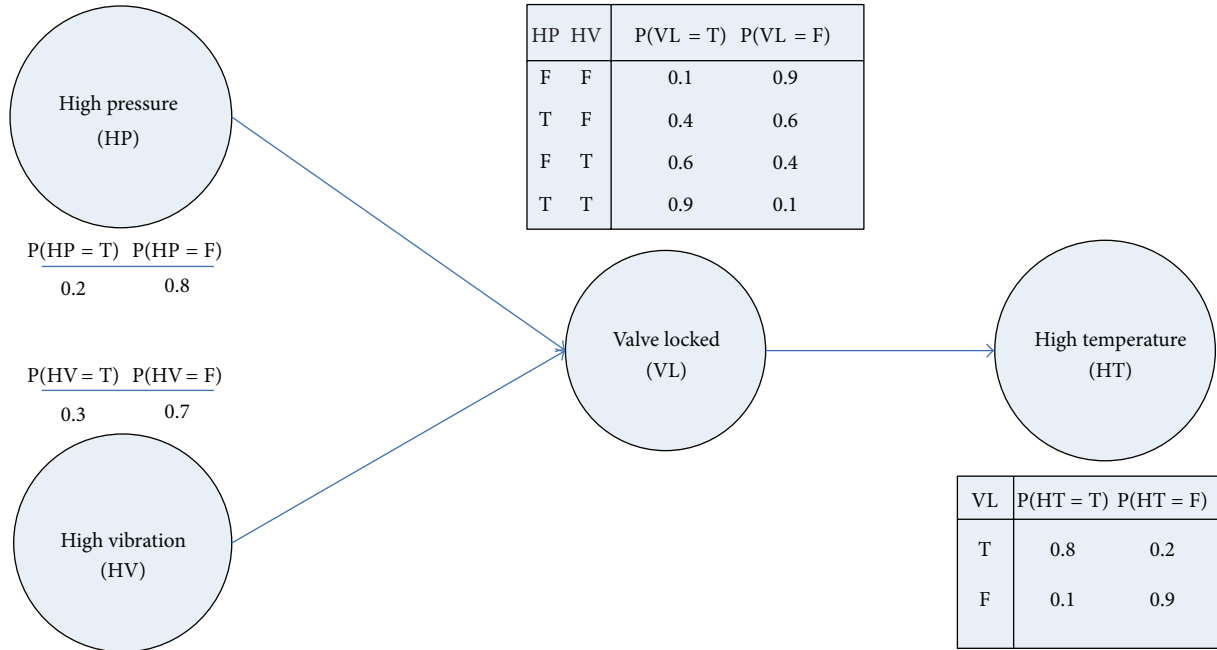


FIGURE 1: A simple FPBN model.

actual states of failure detection nodes are inputted, the FPBN model is operated with CPDs to estimate the state of failure cause nodes which will determine the state of failure mode node.

3. Modeling of FPBN with Divide-and-Conquer Principle

3.1. BN Modeling Method Based on Data Mining. Since building objective BN model with expert experience is not easy, learning practical model from dataset with data mining methods has attracted considerable attention recently [19]. The BN modeling process usually consists of two parts: learning the BN structure which is represented with nodes and edges and learning the BN parameters which specify the CPDs of BN.

The key problem of learning BN structure from dataset is to find the most proper network structure which could represent the potential relationships in the dataset accurately. Because learning the BN structure from dataset is an NP-hard problem for large networks, the conditional independence tests based algorithms and the score and search based algorithms have been proposed separately to settle this challenge [20]. The former method discovers the potential conditional independence relationships of nodes from dataset with conditional independence test equation and builds BN based on such relationships [21]. In the score and search based methods, a score function is used as the criterion to represent how the candidate network structure fits the dataset while a searching algorithm is applied to find the best structure with the highest score in all candidate network structures. Some popular score functions include Copper-Herskovits function [22], Bayesian information criterion (BIC) function

[23], and minimum description length [24]. In the searching algorithms, the BN structure is usually encoded as an ordered string or a connection matrix while different operators have been designed and employed to find the one with the highest scores. Some common algorithms include genetic algorithm [25], evolutionary programming [26], ant colony optimization [27], integer linear programming [28], globally parallel learning [29], and heuristic equivalent learning [30].

Such algorithms are mainly proposed for general BN structure learning where no restriction is applied on the directions of edges. There is also another kind of BN structure learning where the sequence of all nodes is known and the latter node could only be the child node of former node. The famous K2 algorithm could deal with this kind of BN structure learning problem well with deterministic searching [22]. The FPBN is actually a new kind of BN structure learning where the sequence of all node subsets is known and the node in latter subset could only be the child node of the node in a former subset. The K2 algorithm cannot be applied to FPBN structure learning for its comprehensive node sequence restrictions. The general BN structure learning algorithms usually cost lots of time when the node number is large and the edges between nodes are complex. So, the characteristics of nodes and edges in FPBN should be considered to decrease the number of possible candidate network structures and limit the searching space.

3.2. Structure Learning of FPBN with Divide-and-Conquer Principle. In computer science, divide-and-conquer principle is an important algorithm based on multibranch recursion [31]. The divide-and-conquer algorithm breaks down a problem into two or more subproblems of the same type which are simple enough to be solved directly. The solutions

to the subproblems are then combined to give a solution to the original problem. The correctness of a divide-and-conquer algorithm can be proved by mathematical induction, and its computational cost is often determined by solving recurrence relations.

According to the directions of edges in FPBN, it is clear that (1) the father nodes of failure detection nodes may belong to failure cause nodes and failure mode nodes; (2) the father nodes of the failure mode nodes could only be failure cause nodes; (3) for a node in the failure cause node subset, every node except itself could be its father node. With these restrictions, a divide-and-conquer principle based algorithm (FPBN-DC) is introduced to learn the FPBN network structure. It will build the network structures for failure detection, failure mode, and failure cause nodes separately. The modeling process of FPBN-DC algorithm is listed as follows.

Step 1. Initialize node set $\mathbf{X} = \{X_1, X_2, X_3, \dots, X_n\}$ in FPBN and classify the nodes in \mathbf{X} into three subsets $\mathbf{C}, \mathbf{M}, \mathbf{E}$ according to the type of nodes. Three subsets are ordered in ascending which means the node in latter subset could only be the child node of node in former subset.

Step 2. Choose the score function to evaluate candidate FPBN network structure. The BIC score [23] is used as the score function, as (1). Because the BIC function of whole network could be decomposed as the sum of each node's single function, (1) can be transferred to (2):

$$V^{\text{BIC}} = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} m_{ijk} \log \frac{m_{ijk}}{m_{ij}} - \sum_{i=1}^n q_i \frac{(r_i - 1)}{2} \log m, \quad (1)$$

$$\begin{aligned} V^{\text{BIC}} &= \sum_{i=1}^n \left(\sum_{j=1}^{q_i} \sum_{k=1}^{r_i} m_{ijk} \log \frac{m_{ijk}}{m_{ij}} - \frac{q_i (r_i - 1)}{2} \log m \right) \\ &= \sum_{i=1}^n V_{X_i}^{\text{BIC}}, \end{aligned} \quad (2)$$

where n represents the number of nodes in FPBN; q_i represents the number of candidate combination states of the father nodes of the i th node; r_i represents the number of candidate states of the i th node; m_{ijk} represents the number of failure records which satisfy the request that the i th node is in the k th state and its father node set is in the j th state; m_{ij} represents the number of failure records which matches that the father node set of the i th nodes is in the j th state; m represents the whole number of all the failure data records.

To search the best network structure with the highest V^{BIC} , it is clear that every part of $V_{X_i}^{\text{BIC}}$ should get its score as high as possible. Since FPBN is an extension from BN, it still has to satisfy the request that there is no loop in the network structure. So, the key point is to decompose V^{BIC} logically where there is no loop in the corresponding network structure.

Theorem 1. *If there is no loop inside the interior structure of each of node subsets $\mathbf{C}, \mathbf{M}, \mathbf{E}$ in FPBN, then no loop exists in the whole FPBN model.*

Proof. If there is a loop in the FPBN model with no loop inside three subsets $\mathbf{C}, \mathbf{M}, \mathbf{E}$, then there must be at least one edge point from one of the subsets $\mathbf{C}, \mathbf{M}, \mathbf{E}$ with a second edge pointing back to this subset. But according to the FPBN edge' directions, the edge could only point to the latter subset and cannot point back to the former subset. So, such a loop in the FPBN model does not exist. \square

Lemma 2. *The maximal BIC score of the FPBN model could be broken up to the sum of the maximal BIC score of the three subsets, as $\max(V^{\text{BIC}}) = \max(\sum_{i=1}^n V_{X_i}^{\text{BIC}}) = \max(V_{\mathbf{C}}^{\text{BIC}}) + \max(V_{\mathbf{M}}^{\text{BIC}}) + \max(V_{\mathbf{E}}^{\text{BIC}})$.*

Proof. According to Theorem 1, there will be no loop between the subsets. When every subset has the highest score with no loop inside the subset, the whole FPBN structure satisfies the limitation of no loop and the score of the structure is the highest. \square

With Lemma 2, the FPBN structure searching problem for the highest score is divided into three small scale structure searching problems.

Step 3. Select a node E_i which belongs to subset \mathbf{E} randomly and remove it from \mathbf{X} . Its candidate father node set is $\pi(E_i)_a \subseteq (\mathbf{C} \cup \mathbf{M})$.

Theorem 3. *Adding a father node to the node which belongs to failure detection subset \mathbf{E} will not form a loop inside the subset.*

Proof. Because $\pi(E_i)_a \subseteq (\mathbf{C} \cup \mathbf{M})$, adding a father node to any node E_i belonging to subset \mathbf{E} will not connect any two nodes inside the subset with an edge. This means that there will be no relationship between failure detection nodes. \square

Lemma 4. *The maximal BIC score of subset \mathbf{E} could be broken up to the sum of the maximal BIC score of each node inside the subset, as $\max(V_{\mathbf{E}}^{\text{BIC}}) = \max(\sum_{E_i \in \mathbf{E}} V_{E_i}^{\text{BIC}}) = \sum_{E_i \in \mathbf{E}} \max(V_{E_i}^{\text{BIC}})$.*

Proof. According to Theorem 3, there will not be any loop between the nodes inside the subset \mathbf{E} . When each node $(E_i, i = 1 \dots)$ has the highest score, the subset structure satisfies the limitation of no loop and the score of the subset $V_{\mathbf{E}}^{\text{BIC}}$ is the highest. \square

Lemma 4 could reduce the searching complexity by just calculating the highest score of every single node inside the subset \mathbf{E} .

Step 4. With the selected node E_i , for every node in candidate father node set $\pi(E_i)_a$, compute the updated structure score $V_{E_i}^{\text{BIC}}$ supposing that the node is added to the actual father node set $\pi(E_i)$ of E_i . According to Theorem 3, it does not need to verify the structure when a node is added to node E_i 's father node set.

Step 5. Select the node in the set $\pi(E_i)_a$ that leads to the highest score of $V_{E_i}^{\text{BIC}}$ and name this score as $V_{E_i\text{-new}}^{\text{BIC}}$. If the score $V_{E_i\text{-new}}^{\text{BIC}}$ is higher than the old structure score $V_{E_i\text{-old}}^{\text{BIC}}$, move this node from the set $\pi(E_i)_a$ to the actual father node set $\pi(E_i)$ of E_i and update the score as $V_{E_i\text{-old}}^{\text{BIC}} = V_{E_i\text{-new}}^{\text{BIC}}$. Turn to Step 4 and search other father nodes of E_i in the rest candidate father node set $\pi(E_i)_a$. If $(V_{E_i\text{-new}}^{\text{BIC}} < V_{E_i\text{-old}}^{\text{BIC}})$, it means there is no possible father node in set $\pi(E_i)_a$ which could lead to a higher score. Turn to Step 6.

Step 6. Check whether there is still a node in set \mathbf{X} which belongs to failure detection subset \mathbf{E} . If yes, turn to Step 3 and search its maximal score. If no, it means the highest score of every node in subset \mathbf{E} has been found. Turn to Step 7 to search the maximal score of the failure mode node.

Step 7. According to the FPBN description, there will be usually only one failure mode node M_i in subset \mathbf{M} . Select the node M_i and remove it from \mathbf{X} . Its candidate father node set is $\pi(M_i)_a \subseteq \{\mathbf{C}\}$.

Theorem 5. *Adding a father node to the node which belongs to failure mode subset \mathbf{M} will not form a loop inside the subset.*

Proof. Because there is only one node in the failure mode subset and its candidate father nodes belong to failure cause subset, adding a father node to the node M_i belonging to subset \mathbf{M} will not connect the node itself with an edge. This means that there will be no loop in the subset \mathbf{M} . \square

Step 8. With the selected node M_i , for every node in its candidate father node set $\pi(M_i)_a$, compute the updated structure score $V_{M_i}^{\text{BIC}}$ supposing that the node is added to the actual father node set $\pi(M_i)$ of M_i . According to Theorem 5, it does not need to verify the structure when a node is added to node M_i 's father node set.

Step 9. Select the node in set $\pi(M_i)_a$ which leads to the highest score of $V_{M_i}^{\text{BIC}}$ and name this score as $V_{M_i\text{-new}}^{\text{BIC}}$. If the score $V_{M_i\text{-new}}^{\text{BIC}}$ is higher than the old structure score $V_{M_i\text{-old}}^{\text{BIC}}$, move this node from the set $\pi(M_i)_a$ to the actual father node set $\pi(M_i)$ of M_i and update the score as $V_{M_i\text{-old}}^{\text{BIC}} = V_{M_i\text{-new}}^{\text{BIC}}$. Turn to Step 8 and search other father nodes of M_i in the rest candidate father node set $\pi(M_i)_a$. If $(V_{M_i\text{-new}}^{\text{BIC}} < V_{M_i\text{-old}}^{\text{BIC}})$, it means there is no possible father node in set $\pi(M_i)_a$ which could lead to a higher score. Turn to Step 10.

Step 10. The father nodes of a failure cause node C_i in subset \mathbf{C} could be any other nodes in \mathbf{C} , as $\pi(C_i)_a \subseteq \{\mathbf{C} - C_i\}$. So, the problem transfers to how to learn a general BN structure inside the subset \mathbf{C} with the highest score of $V_{\mathbf{C}}^{\text{BIC}}$. An immune algorithm based structure learning method for BN (BN-IA) [32] is applied to deal with this problem.

Step 11. All maximal scores of three subsets have been calculated and the score of $\max(V^{\text{BIC}})$ is the sum of them.

Using FPBN-DC algorithm, it is clear that the original n nodes of FPBN structure learning problem are broken down into three BN structure learning problems with a fewer number of nodes. Then these 3 smaller scale searching problems can be solved with general BN structure learning methods easily.

3.3. Parameter Learning of FPBN. For the BN, the parameter learning is to find the \mathbf{P} to maximize objective likelihood function $P(\mathbf{P} \mid \mathbf{X}, \mathbf{A}, \mathbf{D})$ when the best network structure $\langle \mathbf{X}, \mathbf{A} \rangle$ is learned from the dataset \mathbf{D} . The calculation of \mathbf{P} is a parameter estimation problem in statistics field and is usually solved by the maximum likelihood estimation (MLE) method [33].

In the MLE based BN parameter learning method, the CPDs of nodes $\{X_1, X_2, \dots, X_n\}$ are $\mathbf{P} = \{P_{ijk} = P(X_i = k \mid \pi(X_i) = j), 1 \leq i \leq n, 1 \leq j \leq q_i, 1 \leq k \leq r_i\}$. By counting the state distributions of each node under every state combination of all its father nodes from dataset, the MLE method can find the best probability distributions \mathbf{P}^* for all nodes. Each parameter p_{ijk}^* in \mathbf{P}^* is calculated as (3) and its practical meaning is shown in (4). When all p_{ijk}^* in \mathbf{P} reach p_{ijk}^* , the objective likelihood function $P(\mathbf{P}^* \mid \mathbf{X}, \mathbf{A}, \mathbf{D})$ will have the largest value:

$$p_{ijk}^* = \begin{cases} \frac{m_{ijk}}{\sum_{k=1}^{r_i} m_{ijk}}, & \sum_{k=1}^{r_i} m_{ijk} > 0 \\ \frac{1}{r_i}, & \text{else,} \end{cases} \quad (3)$$

$$p_{ijk}^* = (\text{the number of records fit } X_i = k \text{ and } \pi(X_i) = j \text{ in dataset } \mathbf{D}) \times (\text{the number of records fit } \pi(X_i) = j \text{ in dataset } \mathbf{D})^{-1}. \quad (4)$$

Because the CPDs in FPBN are the same with CPDs in BN, the parameter learning of FPBN also used the effective MLE method.

4. Simulation Study

4.1. Simulation Dataset. For the simulation study, we introduced a practical helicopter convertor FPBN model [34] as the original model. The nodes in the helicopter convertor FPBN model belong to three subsets. The failure cause subset includes "Power part," "Voltage adjustor," "Transformation filter," "Output filter," and "Fan." The failure mode node is "No output" and the failure detection nodes are "Voltage output," "Filter output," and "Fan sound." The details of these nodes are shown in Table 1.

The cause-effect relationships among nodes are shown in Figure 2. The node "Power part" affects node "Voltage adjustor" while the nodes "Voltage adjustor," "Transformation filter," and "Output filter" result in the convertor failure of "No output." The node "Voltage output" is an outer representation of "Power part" and the node "Filter output"

TABLE 1: Nodes in the helicopter convertor FPBN model.

Subset	ID	Variable	States	States description
Failure cause	1	Power part	0, 1, 2	Normal, failure 1, failure 2
	2	Voltage adjustor	0, 1	Normal, failure
	3	Transformation filter	0, 1	Normal, failure
	4	Output filter	0, 1	Normal, failure
	5	Fan	0, 1	Normal, failure
Failure mode	6	No output	0, 1	Normal, no output
	7	Voltage output	0, 1	Output normal, output abnormal
Failure detection	8	Filter output	0, 1	Output, no output
	9	Fan sound	0, 1	Normal, no sound

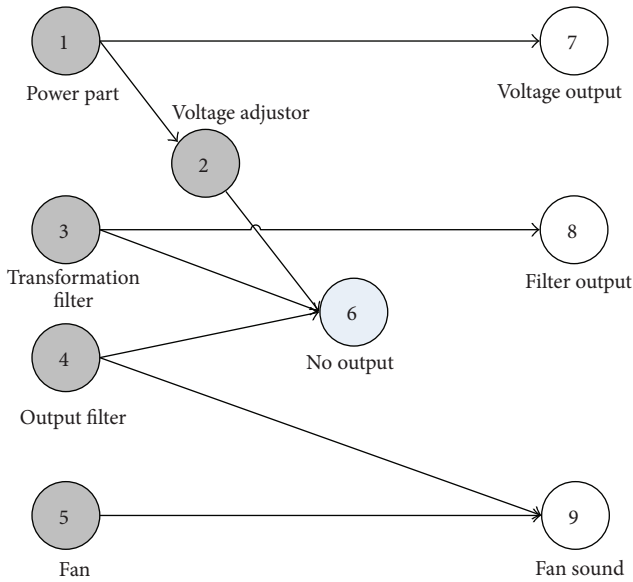


FIGURE 2: Network structure of a helicopter convertor FPBN model.

is also an outer representation of “Transformation filter.” The node “Fan sound” is the result of both “Output filter” and “Fan.”

According to this practical model, 3000, 5000, and 7000 operation records are generated from it separately with a random sampling method. Each record represents the corresponding states of all variables in helicopter convertor at a time. These different scales of failure record datasets are named as dataset 1, dataset 2, and dataset 3 to demonstrate the application of FPBN-DC and verify its performance independently.

4.2. Simulation Results. To verify the effectiveness and efficiency of the proposed FPBN-DC algorithm, the BN-IA algorithm, which ignores the assumption of node types and edge directions in FPBN, is also introduced to learn the network structure from datasets.

First of all, we discuss the coding scale of network structure which determines the searching space of each algorithm. For both FPBN-DC and BN-IA, an adjacency matrix is used to describe the network structure, as shown

TABLE 2: Coding scale in the helicopter convertor FPBN model.

	FPBN-DC	BN-IA
Number of nodes	5	9
Max number of father nodes	4	8
Number of bits	20	72
Possible value of each bit	0, 1, 2, 3, 4, 5	0, 1, 2, 3, 4, 5, 6, 7, 8, 9

in Table 2. In BN-IA, the number of bits for structure code is 72 because there are 9 nodes in the model and each node needs 8 bits to represent its father nodes. In FPBN-DC, the structure code has only 20 bits because the structure learning scale is reduced according to Lemma 2. Actually, we just need to learn the network structure of nodes in failure cause subset. Generally, the algorithm searching time is mainly consumed in the score calculation process for the score and search based algorithms. Because BIC score has to go through the dataset ($q_i \times r_i$) times to count the m_{ijk} and m_{ij} according to (1), the searching time deeply depends on the parameter of q_i and r_i . Since q_i represents the number of all candidate combination states of the father nodes of the i th node, it relates to the max number of a node’s father nodes directly. It is obvious that the FPBN-DC algorithm has smaller searching space and less searching time.

Then, two algorithms learn the 3 generated datasets 10 times separately with same parameters. The highest and average fitness values (an equivalent of the BIC score) of each algorithm for every dataset are listed in Table 3. The convergence iterations of the corresponding fitness values are also listed in it. Since the fitness value represents the similarity between the network structure and corresponding dataset, the highest fitness of 10 runs is the main criterion for the effectiveness of algorithms. The corresponding convergence iteration of the highest fitness is a reasonable criterion for algorithm efficiency.

According to the highest fitness (in bold type) of each algorithm and each dataset during the 10 runs in Table 3, the FPBN-DC and BN-IA algorithms reach the same highest fitness values for all the 3 datasets. However, the corresponding iterations needed to reach the highest fitness in FPBN-DC are much less than the iterations in BN-IA. Furthermore, the average fitness values of 10 runs in 3 datasets equal the highest fitness values in 10 runs for FPBN-DC. This means

TABLE 3: Fitness values and corresponding iterations of each algorithm for every dataset.

Runs	Dataset 1		Dataset 2		Dataset 3	
	Iteration	Fitness	Iteration	Fitness	Iteration	Fitness
FPBN-DC						
1	14	3.828	12	2.475	8	1.636
2	14	3.828	7	2.475	5	1.636
3	11	3.828	6	2.475	93	1.636
4	9	3.828	15	2.475	6	1.636
5	8	3.828	7	2.475	15	1.636
6	8	3.828	16	2.475	6	1.636
7	17	3.828	15	2.475	10	1.636
8	10	3.828	9	2.475	9	1.636
9	12	3.828	12	2.475	15	1.636
10	13	3.828	6	2.475	8	1.636
Average	11.6	3.828	10.5	2.475	17.5	1.636
BN-IA						
1	79	3.828	99	2.475	79	1.636
2	38	3.828	58	2.475	93	1.636
3	80	3.79	42	2.473	64	1.636
4	92	3.828	30	2.475	99	1.635
5	56	3.827	59	2.47	70	1.634
6	95	3.824	85	2.474	82	1.626
7	97	3.828	58	2.475	75	1.636
8	86	3.828	83	2.47	89	1.634
9	43	3.828	88	2.472	47	1.634
10	60	3.828	69	2.472	56	1.636
Average	72.6	3.8237	67.1	2.4731	75.4	1.6343

that the FPBN-DC can get the best network structure in every searching run. For BN-IA, the average fitness values of 10 runs in 3 datasets are less than the highest fitness values, which represent the stochastic error of this algorithm.

The objective of the FPBN structure learning is to search the best network which could represent the dataset comprehensively. So, comparing the best network of each algorithm with the original one is another useful criterion for algorithm effectiveness. The edge differences between the best network of each algorithm in 10 runs and original FPBN in Figure 2 are listed in Table 4, where A represents the number of added edges, L represents the number of lacked edges, and R represents the number of reverse edges. The best network structures learned by FPBN-DC and BN-IA algorithm are exactly the same as the original one in 3 datasets. This result also shows the ability of FPBN-DC in data retrieval.

Finally, according to the comparison results, the FPBN-DC algorithm can get higher fitness value with a lower number of iterations, which verified its effectiveness and efficiency compared with BN-IA algorithm.

5. Conclusion

The paper proposed an effective algorithm to build the FPBN model from system operation dataset. The types of network

TABLE 4: Edge differences between the learned FPBN structures with the original one.

	Dataset 1			Dataset 2			Dataset 3		
	A	L	R	A	L	R	A	L	R
FPBN-DC	0	0	0	0	0	0	0	0	0
BN-IA	0	0	0	0	0	0	0	0	0

nodes, the directions of network edges, and the CPDs of nodes in FPBN are discussed in detail at first. Then, the FPBN-DC algorithm is introduced into the FPBN modeling process to learn the network structure of failure detection, failure mode, and failure cause nodes separately according to their assumptions on edge directions. Finally, the simulation study of a helicopter convertor FPBN model is carried out. The proposed FPBN-DC and the BN-IA algorithms learn the same 3 generated datasets 10 times separately with same parameters. Taking the advantages of divide-and-conquer principle, the FPBN-DC has a smaller coding scale than BN-IA which means smaller searching space and less searching time. The comparison results also show that FPBN-DC can get higher fitness value with the lower number of iterations. The learned network structures by FPBN-DC in 3 datasets are exactly the same as the original one which also verified its effectiveness. For the future research, with the application of sensors in practical engineering system, we plan to introduce the real-time detection node into FPBN model which may provide a more precise failure prediction.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors gratefully acknowledge the financial support for this research from the National Natural Science Foundation of China (Grant nos. 71101116 and 71271170), the Aeronautical Science Foundations of China (Grant no. 2011ZC53027), and the Basic Research Foundation of NPU (Grant no. JC20120228).

References

- [1] T. H. Xu, T. Tang, H. F. Wang, and T. M. Yuan, "Risk-based predictive maintenance for safety-critical systems by using probabilistic inference," *Mathematical Problems in Engineering*, vol. 2013, Article ID 947104, 9 pages, 2013.
- [2] M. Roemer, C. Byington, G. Kacprzyński, and G. Vachtsevanos, "An overview of selected prognostic technologies with application to engine health management," in *Proceedings of the ASME Turbo Expo: Power for Land, Sea, and Air (ET '06)*, pp. 707–715, Barcelona, Spain, May 2006.
- [3] Y. F. Li, H. Z. Huang, H. L. Zhang, N. C. Xiao, and Y. Liu, "Fuzzy sets method of reliability prediction and its application to a turbocharger of diesel engines," *Advances in Mechanical Engineering*, vol. 2013, Article ID 216192, 7 pages, 2013.

- [4] H. X. Wang and Y. Jiang, "Performance reliability prediction of complex system based on the condition monitoring information," *Mathematical Problems in Engineering*, vol. 2013, Article ID 836517, 7 pages, 2013.
- [5] X. N. Zhang, J. F. Yang, S. L. Du, and S. D. Huang, "A new method on software reliability prediction," *Mathematical Problems in Engineering*, vol. 2013, Article ID 385372, 8 pages, 2013.
- [6] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, San Francisco, Calif, USA, 2005.
- [7] W. C. Chen, S. S. Tseng, and C. Y. Wang, "A novel manufacturing defect detection method using association rule mining techniques," *Expert Systems with Applications*, vol. 29, no. 4, pp. 807–815, 2005.
- [8] H. K. Han, H. S. Kim, and S. Y. Sohn, "Sequential association rules for forecasting failure patterns of aircrafts in Korean airforce," *Expert Systems with Applications*, vol. 36, no. 2, pp. 1129–1133, 2009.
- [9] M. Dong, "A tutorial on nonlinear time-series data mining in engineering asset health and reliability prediction: concepts, models, and algorithms," *Mathematical Problems in Engineering*, vol. 2010, Article ID 175936, 22 pages, 2010.
- [10] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Francisco, Calif, USA, 1st edition, 1988.
- [11] F. Zong, H. G. Xu, and H. Y. Zhang, "Prediction for traffic accident severity: comparing the Bayesian network and regression models," *Mathematical Problems in Engineering*, vol. 2013, Article ID 475194, 9 pages, 2013.
- [12] Y. Yan and B. Suo, "Risks analysis of logistics financial business based on evidential Bayesian network," *Mathematical Problems in Engineering*, vol. 2013, Article ID 785218, 8 pages, 2013.
- [13] A. Muller, M. Suhner, and B. Iung, "Formalisation of a new prognosis model for supporting proactive maintenance implementation on industrial system," *Reliability Engineering and System Safety*, vol. 93, no. 2, pp. 234–253, 2008.
- [14] H. Langseth and L. Portinale, "Bayesian networks in reliability," *Reliability Engineering and System Safety*, vol. 92, no. 1, pp. 92–108, 2007.
- [15] O. Doguc and J. E. Ramirez-Marquez, "A generic method for estimating system reliability using Bayesian networks," *Reliability Engineering and System Safety*, vol. 94, no. 2, pp. 542–550, 2009.
- [16] P. Weber and L. Jouffe, "Complex system reliability modelling with dynamic object oriented Bayesian networks (DOOBN)," *Reliability Engineering and System Safety*, vol. 91, no. 2, pp. 149–162, 2006.
- [17] S. Mahadevan, R. X. Zhang, and N. Smith, "Bayesian networks for system reliability reassessment," *Structural Safety*, vol. 23, no. 3, pp. 231–251, 2001.
- [18] Z. Q. Cai, S. D. Sun, S. B. Si, and N. Wang, "Research of failure prediction Bayesian network model," in *Proceedings of the IEEE 16th International Conference on Industrial Engineering and Engineering Management*, pp. 2021–2025, Beijing, China, October 2009.
- [19] L. M. Wang, "Extraction of belief knowledge from a relational database for quantitative Bayesian network inference," *Mathematical Problems in Engineering*, vol. 2013, Article ID 297121, 10 pages, 2013.
- [20] D. Chickering, D. Heckerman, and C. Meek, "Large-sample learning of Bayesian networks is NP-hard," *Journal of Machine Learning Research*, vol. 5, pp. 1287–1330, 2004.
- [21] J. Cheng, R. Greiner, J. Kelly, D. Bell, and W. R. Liu, "Learning Bayesian networks from data: an information-theory based approach," *Artificial Intelligence*, vol. 137, no. 1-2, pp. 43–90, 2002.
- [22] G. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, no. 4, pp. 309–347, 1992.
- [23] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.
- [24] W. Lam and F. Bacchus, "Learning Bayesian belief networks: an approach based on the MDL principle," *Computational Intelligence*, vol. 10, no. 3, pp. 269–293, 1994.
- [25] P. Larranaga, H. Karshenas, C. Bielza, and R. Santana, "A review on evolutionary algorithms in Bayesian network learning and inference tasks," *Information Sciences*, vol. 233, pp. 109–125, 2013.
- [26] M. L. Wong, W. Lam, and K. Leung, "Using evolutionary programming and minimum description length principle for data mining of Bayesian networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 2, pp. 174–178, 1999.
- [27] C. Wang, S. Liu, and M. Zhu, "Bayesian network learning algorithm based on unconstrained optimization and ant colony optimization," *Journal of Systems Engineering and Electronics*, vol. 23, no. 5, pp. 784–790, 2012.
- [28] M. Studeny and D. Haws, "Learning Bayesian network structure: towards the essential graph by integer linear programming tools," *International Journal of Approximate Reasoning*, vol. 55, no. 4, pp. 1043–1071, 2014.
- [29] O. Nikolova, J. Zola, and S. Aluru, "Parallel globally optimal structure learning of Bayesian networks," *Journal of Parallel and Distributed Computing*, vol. 73, no. 8, pp. 1039–1048, 2013.
- [30] Y. H. Zhang, W. S. Zhang, and Y. Xie, "Improved heuristic equivalent search algorithm based on maximal information coefficient for Bayesian network structure learning," *Neurocomputing*, vol. 117, pp. 186–195, 2013.
- [31] F. Hu and G. Y. Wang, "Knowledge reduction based on divide and conquer method in rough set theory," *Mathematical Problems in Engineering*, vol. 2012, Article ID 864652, 24 pages, 2012.
- [32] Z. Q. Cai, *Research on failure prediction methods for complex equipments [Ph.D. thesis]*, Northwestern Polytechnical University, Xian, China, 2010 (Chinese).
- [33] D. J. Spiegelhalter and S. L. Lauritzen, "Sequential updating of conditional probabilities on directed graphical structures," *Networks*, vol. 20, no. 5, pp. 579–605, 1990.
- [34] Z. Q. Cai, S. D. Sun, S. B. Si, and N. Wang, "Learning failure prediction Bayesian networks based on genetic algorithms," in *Proceedings of the 17th ISSAT International Conference on Reliability and Quality in Design*, pp. 260–264, Vancouver, Canada, August 2011.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

