

Article

Test Strategy Optimization Based on Soft Sensing and Ensemble Belief Measurement

Wenjuan Mei ¹, Zhen Liu ^{1,*}, Lei Tang ² and Yuanzhang Su ^{1,3}

¹ School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China; meiwenjuan@std.uestc.edu.cn (W.M.); syz@uestc.edu.cn (Y.S.)

² Southwest Institute of Technical Physics, Chengdu 611731, China; ltang20001142@163.com

³ School of Foreign Language, University of Electronic Science and Technology of China, Chengdu 611731, China

* Correspondence: scdliu@uestc.edu.cn; Tel.: +86-028-6183-0316

Abstract: Resulting from the short production cycle and rapid design technology development, traditional prognostic and health management (PHM) approaches become impractical and fail to match the requirement of systems with structural and functional complexity. Among all PHM designs, testability design and maintainability design face critical difficulties. First, testability design requires much labor and knowledge preparation, and wastes the sensor recording information. Second, maintainability design suffers bad influences by improper testability design. We proposed a test strategy optimization based on soft-sensing and ensemble belief measurements to overcome these problems. Instead of serial PHM design, the proposed method constructs a closed loop between testability and maintenance to generate an adaptive fault diagnostic tree with soft-sensor nodes. The diagnostic tree generated ensures high efficiency and flexibility, taking advantage of extreme learning machine (ELM) and affinity propagation (AP). The experiment results show that our method receives the highest performance with state-of-art methods. Additionally, the proposed method enlarges the diagnostic flexibility and saves much human labor on testability design.



Citation: Mei, W.; Liu, Z.; Tang, L.; Su, Y. Test Strategy Optimization Based on Soft Sensing and Ensemble Belief Measurement. *Sensors* **2022**, *22*, 2138. <https://doi.org/10.3390/s22062138>

Academic Editor: Simon Tomažič

Received: 12 February 2022

Accepted: 4 March 2022

Published: 10 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: prognostic and health management; extreme learning machine; soft sensors

1. Introduction

With the increasing use of electric devices, prognostic and health management engineering (PHM engineering) has played an extremely significant role in product lifetime management over decades [1]. PHM engineering ensures electric devices' lifetime healthy operation and provides appropriate resource assignment for product management [2]. In recent years, the production cycle has shortened because circuit technology and system design have rapidly developed [3–6]. The system structures have become more complicated, more integrated, more intelligent, and highly intensive [7]. Additionally, the potential test procedures and fault cases grow exponentially. As a result, PHM engineering has received active demand and new challenges. Practical conditional maintenance (CM) solutions become difficult to generate on modern system applications. On another hand, CM must be flexible enough to math structure complexity and system function complexity. Hence, efficient PHM engineering solutions for modern devices become an urgent problem for academic researchers and industrial engineers.

Under the CM design, testability design and maintainability design are two essential projects to determine supportability, enhance reliability, and guarantee safety during lifetime device management [8]. Testability design analyses the system's internal structures, selects test projects and arranges test procedures, estimates the system operation condition, and locates failure modes. The key difficulty for testability design is balancing the system's high structural complexity and the solution efficiency properly. Classical testability design approaches use dynamic programming (DP) to assess optimal solutions [9,10]. However,

classic methods suffer from high time complexity when either the number of test projects or failures is larger than 12.

In recent years, the AO* method [11], a sequential testing generation method, balances the generation complexity of test solutions and detection performance for diagnostic procedures with heuristic searching and AND/OR graph topology. Thus, the AO* method became the most popular testability design technique. To match the growing electric system complexity, AO* has been improved to optimize searching mechanisms with information theory [12], evolution algorithms [13,14], dynamic design [15], etc. Additionally, advanced research has simplified searching procedures with rollout strategy [16] and bottom-up decision tree [17] to achieve practical large-system testability solutions. Despite good application results, these testability design approaches need to assign logic relationships between test procedures and failure modes. Hence, all these methods assume that dependent single-signal operations and sequential procedures can make all diagnostic decisions. However, modern devices contain highly complicated logistic relationships [18] between test procedures and potential failure modes under many scenarios. Consequently, the testability design consumes too much human power and affects the testability design efficiency to prepare prior knowledge, especially under short production cycles. Furthermore, the testability design wastes the entailed sensor recording information from test procedures as the existing methods rely on human-selected binary information.

The maintainability estimation model provides real-time operation condition diagnosis and realizes system health management along with testability design. In general, existing maintainability design approaches can be divided into physics-of-failure (PoF) approaches [19–22] and data-driven (DD) approaches [23–25]. PoF approaches use rules from physics or chemical dynamics to estimate electric system failure conditions [26]. With accelerated aging of experimental records and prior modeling knowledge, PoF approaches generate an accurate dynamic model under specific stress influences such as thermal, electrical, and humidity. However, most PoF approaches are only suitable under one stress function, and the methods face high limitations with real applications.

Unlike PoF methods, DD approaches rely on historical sensor information and build maps from sensor recording to failure modes. Hence, DD approaches become more flexible compared to PoF methods. Classical DD approaches use statistic methods such as stochastic methods, regression methods, distance estimation, and similarity estimation [27–30]. These methods require large samples to ensure unbiased estimation and model robustness, and are constrained in off-line modeling. Therefore, statistical approaches have limitations on maintainability design with few sampling records and short time constraints. In recent years, machine learning (ML) methods [31,32] have attracted research attention because of its high accuracy, strong adaptive ability, powerful robustness, and fast computation time. As a result, ML methods, such as neural networks (NN) [33], support vector machine (SVM) [34,35], k-nearest neighbors (KNN) [36], K-means clustering [37], and particle filters (PF) [38], have been used for maintenance design and tuned to successfully extract hidden rules from failure mode and recording information [39]. However, these methods require prior information selection and the preprocessing system degradation features. The performance of testability and maintainability may affect the preprocessing model input due to the complicated system structure and the complex recording information relationships.

We propose a test strategy optimization based on this paper's soft-sensing and ensemble belief measurement to overcome this weakness described above. We suggest a closed loop framework for PHM design to replace the sequential framework between testability and maintenance design. Instead of experienced knowledge, our method uses ensemble learning based on direct sensor recordings to gain the system state estimation. Additionally, we connect the testability and maintenance design by the minimal conditional criterion to optimize the test strategy. Consequently, the proposed method improves the flexibility of PHM design, saves labor, and enhances diagnostic efficiency. The contributions of our work follow.

- Our model builds diagnostic strategies without much prior knowledge and human-elected features. The diagnostic tree is constructed with ELM-based soft-sensor nodes. Instead of experienced features, ELM-based soft-sensor nodes provide basic probability assignment (BPA) directly from the sensor records. Hence, our methods cut the testability design human labor since the method needs no system mechanisms analysis.
- We build a closed loop between testability design and maintenance design. Thus, the maintenance design makes full use of testability design information and improves the testability design efficiency with the advantage of ELM-based construction modules.
- Our model divides the fault set adaptively into several fuzzy sets with affinity propagation and improves the diagnostic efficiency of single test procedures.

The experiment proves that our method has better diagnostic accuracy and lower false alarm ratios than other state-of-art diagnostic methods. Additionally, our diagnostic strategies take only a few tests with little test assignment consumption. For each fault state, the diagnostic procedures provide one efficient test sequence. Thus, the diagnostic procedure enjoys high efficiency for applications. Finally, affinity propagation enlarges the diagnostic flexibility and significantly reduces human labor used on testability design.

The rest of this paper is organized as follows. We introduce the PHM design problem and provide the general framework in the next section. Section 3 presents the details of the algorithms and Section 4 provides the experiment results and discussions. Finally, conclusions are drawn in Section 5.

2. Problem Formulation and General Framework

PHM engineering estimates the degradation processes and recognizes failure modes over the product's full lifetime. Based on experimental research and application surveys, PHM engineering provides fault analysis and maintenance advice to prevent failure occurrence. PHM engineering elements and the corresponding relationships are presented in Figure 1.

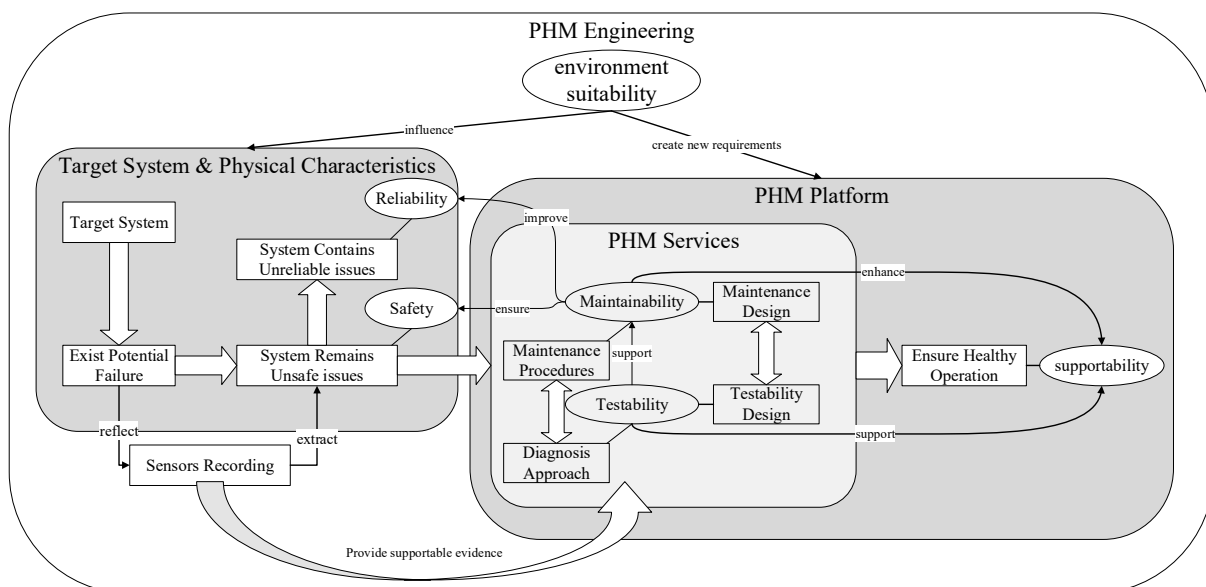


Figure 1. Element and the corresponding relationship of PHM engineering.

To study the target systems, engineers analyze potential failure modes and find unsafe and unreliable features. Thus, the target system's safety and reliability, reflected by its failure features, draw attention to system failure physical characteristics. For higher reliability and safety, the PHM platform provides maintenance procedures and diagnostic approaches for the system's supportability. The diagnostic approach depends on testability and the maintenance procedures that rely on maintainability.

Testability is the design characteristic that the health condition can be detected accurately and failure modes can be located successfully. Meanwhile, maintainability is the ability to repair and recover the system under certain conditions within a certain time. The PHM services can ensure healthy system operation and achieve the PHM engineering purpose with proper maintenance and testability design. In general, supportability is the key PHM engineering purpose influenced by testability and maintainability; thus, the PHM must meet the environmental stability needs. The physical system structure directly influences safety and reliability while system elements reflect safety and reliability critically. High testability and maintainability quality improve the system reliability and safety with good maintenance design procedures and diagnostic approaches. Hence, testability design and maintenance design are two significant PHM platform parts.

Since testability and maintenance are important, various techniques provide practical testability design and maintenance design. To our best knowledge, existing methods build a sequential approach to generate the PHM service. As Figure 2 shows, the traditional framework arranges the test procedures to assess the sensor recording and create maintenance design with the assigned sensor records. The framework is applicable for many large systems. However, the testability design uses human experience binary information such as information flow chart, dependence matrix and AND/OR graph. Therefore, the testability design suffers from extended time and the financial burden with modern complex systems, ignores the coupling effects between test procedures, and wastes valuable sensor recording information. As a result, the maintenance design receives poor performance and low efficiency because of poor information usage and low knowledge transmission efficiency from testability design and maintenance design.

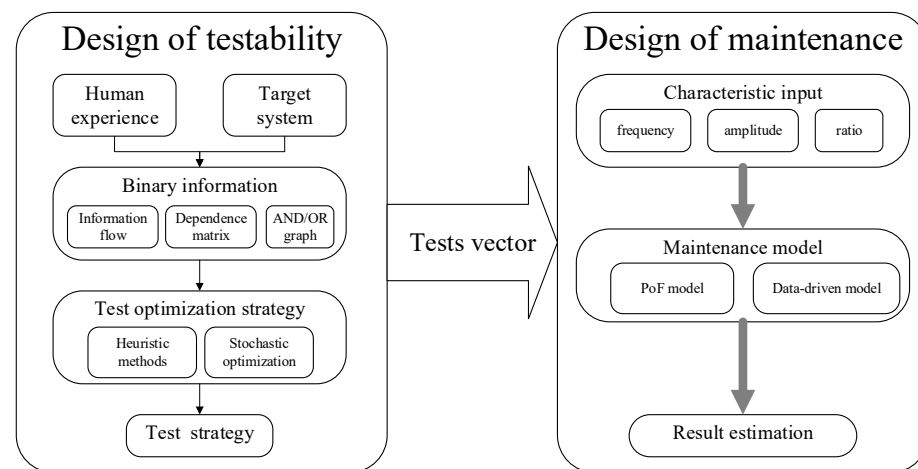


Figure 2. Traditional framework of testability and maintenance design.

We introduce a closed-loop test strategy optimization method on soft-sensor information and ensemble learning to overcome the weakness. Similar to traditional PHM design approaches, the proposed method aims to generate a fault diagnostic tree and cut the fault set with testability sensor information and maintenance signal processing. In contrast, our fault tree grows with direct sensor recording information directly along with the processing module and extends with basic probability assignments. On the other hand, the PHM design process contains a cooperative closed loop between testability and maintenance design to improve information usage and transmission efficiency during fault detection.

The general framework of the proposed method is presented in Figure 3. For maintenance design, the soft sensor integrates the recording information from assigned sensors and the processing signal, such as statistical method, support machine, and learning machine to extract suitable test features. Considering the requirement of fast diagnosis and detection, we use extreme learning machine (ELM) [34], a noniterative single-layer learning

machine, to generate accurate fast processing modules of soft-sensing nodes. The details are introduced in the next section.

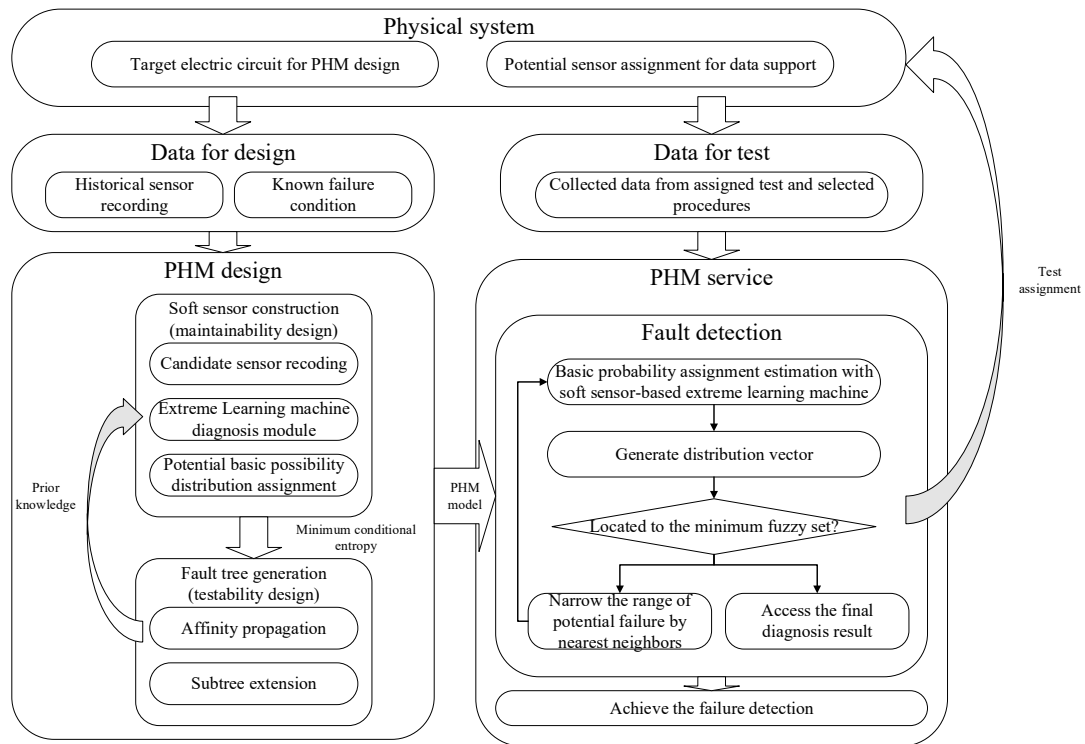


Figure 3. General framework of test strategy optimization based on soft sensing and ensemble belief measurement.

Suppose there exists N possible sensors with the PHM system design; therefore, the potential test procedures set is denoted as $T_{potential} = \{t_1, t_2, \dots, t_i, \dots, t_{N-1}, t_N\}$, where t_i means the test procedures with i -th sensors. As each sensor contains a vector of information $x_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,K}\}$, $i = 1, 2, 3, \dots, N$ with M existing samples from the targeted system, the training information is regarded as:

$$X_{train} = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,N} \\ x_{2,1} & x_{2,2} & \dots & x_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{M,1} & x_{M,2} & \dots & x_{M,N} \end{bmatrix} \quad (1)$$

where x_{ij} is the recording information vector from the j -th sensor for i -th sample, $j = 1, 2, 3, \dots, N$, and $i = 1, 2, 3, \dots, M$.

With the sampling information, the potential fault tree soft-sensor node structure is denoted as following:

$$Node = \{S_{father}, S_{son}, T_{node}, X_{node}, Y_{node}, ELM_{node}, m_{exemplar}\} \quad (2)$$

where the T_{node} is the assigned test procedures from previous selected soft-sensor nodes and the potential selected test procedure, and X_{node} is the training sample sensor recording information to detect under the node, as follows:

$$X_{node} = [x_{1,T_{node}}, x_{2,T_{node}}, \dots, x_{M_{node},T_{node}}]^T \quad (3)$$

$$x_{i,T_{node}} = \{x_{i,j} | t_j \in T_{node}\} \quad (4)$$

where M_{mode} is the number of the sampling data. \mathbf{Y}_{node} provides the corresponding sample failure conditions. Suppose the whole set of failure mode is $\mathcal{S}_{node} = \{s_1, s_2, \dots, s_K\}$, where K is the number of fault modes considered, then \mathbf{Y}_{node} is determined with the actual condition of training samples as follows:

$$\mathbf{Y}_{node} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{M_{node}}]^T \quad (5)$$

$$\mathbf{y}_i = [y_{i,1}, y_{i,2}, \dots, y_{i,K}] \quad (6)$$

$$y_{i,j} = \begin{cases} 1 & ss_i = s_j \\ 0 & ss_i \neq s_j \end{cases} \quad (7)$$

where ss_i is the actual i -th sample fault mode. Additionally, \mathcal{S}_{father} is the fault mode set of the soft-sensor node, represented as follows:

$$\mathcal{S}_{father} = \{s_j | \exists \mathbf{x}_i \in \mathbf{X}_{node} \text{ and } s_j \in \mathcal{S}, ss_i = s_j\} \quad (8)$$

With \mathbf{X}_{node} and \mathbf{Y}_{node} , the ELM serves as the soft-sensor signal processing part and aims to provide the fuzzy set of fault states as much as possible. To achieve the goal, ELM builds a map $f: R^{1 \times N} \rightarrow R^{1 \times K}$ from the training sample signals to estimate the fault states and provides the training sample prediction \mathbf{M}_{node} as follows:

$$\mathbf{M}_{node} = [f(\mathbf{x}_{1,T_{node}} | ELM_{node}), f(\mathbf{x}_{2,T_{node}} | ELM_{node}), \dots, f(\mathbf{x}_{M_{node},T_{node}} | ELM_{node})]^T \quad (9)$$

With \mathbf{M}_{node} , ELM estimates the training sample failure modes. To determine the processing part performance, \mathbf{Y}_{node} is used as the expected output marks for the fault detection process. From the view of detection process, two indexes, fault detection rate (FDR) and false alarm rate (FAR), play essential roles in evaluating the accuracy.

FDR is defined as the ratio between the failure mode probability that is successfully detected with the ELM and the total failure modes probability. Here, we assume the historical samples are subject to the general failure probability distribution of the real applications. Thus, the statistic characteristics of training samples reflect the total failure mode probability and the training sample detection performance depicts the detection probability. From above, FDR for the node is presented as follows:

$$FDR_{node} = \frac{\sum_{i=1}^{M_{node}} \sum_{j=1}^K P(f_j(\mathbf{x}_{1,T_{node}} | ELM_{node}) \geq 1 - \varepsilon, y_{i,j} = 1)}{\sum_{i=1}^{M_{node}} \sum_{j=1}^K y_{i,j}} \quad (10)$$

where ε is the detection margin. From the generation process, $f_j(\mathbf{x}_{1,T_{node}} | ELM_{node})$ and $y_{i,j}$ are independent of each other. Additionally, since ELM provides a continuous probability estimation, the loss function with respect to FDR_{node} is computed as follows:

$$L_{FDR_{node}} = \frac{\sum_{i=1}^{M_{node}} \sum_{j=1}^K (y_{i,j} - f_j(\mathbf{x}_{1,T_{node}} | ELM_{node})) P(y_{i,j} = 1)}{\sum_{i=1}^{M_{node}} \sum_{j=1}^K y_{i,j}} \quad (11)$$

Along with FDR, FAR presents the ratio between false-alarm failure mode probability and the total failure detection probability. Taking consideration of $f_j(\mathbf{x}_{1,T_{node}} | ELM_{node})$, FAR is computed as follows:

$$FAR_{node} = \frac{\sum_{i=1}^{M_{node}} \sum_{j=1}^K P(f_j(\mathbf{x}_{1,T_{node}} | ELM_{node}) \geq 1 - \varepsilon, y_{i,j} = 0)}{\sum_{i=1}^{M_{node}} \sum_{j=1}^K P(f_j(\mathbf{x}_{1,T_{node}} | ELM_{node}) \geq 1 - \varepsilon)} \quad (12)$$

Here, it is assumed that the model has enough accuracy so that the failure estimation and the total failure conditions have approximate values. Thus, $\sum_{i=1}^{M_{node}} \sum_{j=1}^K P(f_j(\mathbf{x}_{1,T_{node}} | ELM_{node}))$

$\geq 1 - \varepsilon$) is able to be approximately equal to $\sum_{i=1}^{M_{node}} \sum_{j=1}^K y_{i,j}$. Therefore, the loss function of the FAR is denoted as:

$$L_{node} = \frac{\sum_{i=1}^{M_{node}} \sum_j^K (y_{i,j} - f_j(x_{1,T_{node}} | ELM_{node}))^2 P(y_{i,j}=1)}{\sum_{i=1}^{M_{node}} \sum_j^K y_{i,j}} + \frac{\sum_{i=1}^{M_{node}} \sum_j^K (y_{i,j} - f_j(x_{1,T_{node}} | ELM_{node}))^2 P(y_{i,j}=0)}{\sum_{i=1}^{M_{node}} \sum_j^K y_{i,j}} \quad (13)$$

$$= \frac{\sum_{i=1}^{M_{node}} \sum_j^K (y_{i,j} - f_j(x_{1,T_{node}} | ELM_{node}))^2}{\sum_{i=1}^{M_{node}} \sum_j^K y_{i,j}}$$

As $\sum_{i=1}^{M_{node}} \sum_j^K y_{i,j}$ is independent from the soft-sensor construction, the task of ELM is to minimize $\sum_{i=1}^{M_{node}} \sum_j^K (y_{i,j} - f_j(x_{1,T_{node}} | ELM_{node}))^2$, which is the difference between M_{node} and Y_{node} , expressed as follows:

$$ELM = argmin \left\{ \sum_{i=1}^{M_{node}} \|y_i - f(x_{1,T_{node}} | ELM_{node})\|^2 \right\} \quad (14)$$

Since the maintenance design of the proposed method directly relies on the recording sensor signals, the physical system knowledge is largely preserved and the information usage is highly enhanced.

Based on soft-sensor node construction, testability design process adds the soft-sensor node with best performance and builds a fault tree, taking consideration of potential soft-sensor node under the minimum conditional entropy criterion. Hence, the assigned soft-sensor nodes decrease the diagnostic uncertainty and improve the detection efficiency. Besides, affinity propagation (AP) is adapted to separate the fuzzy set of the fault modes and generate subnodes for the diagnostic model with the exemplar probability estimation $m_{exampor}$ and basic probability assignment BPA_{node} . The subnodes are denoted with the subset of failure modes $S_{son} = [S_{son,1}, S_{son,2}, \dots, S_{son,K_{node}}]$ and satisfies the condition that $\bigcup_{i=1}^{K_{node}} S_{son,i} = S_{father}$.

After adding the soft-sensor nodes and extending the subset of fault modes, the information of assigned nodes T_{node} is regarded as prior knowledge, serving as feedback from testability design to maintenance design, and extending the fault tree until reaching the minimum fault condition set.

With the cooperation between testability design and maintenance design, a PHM model based on soft-sensor information is generated and the sensors for PHM maintenance are assigned based on the selected test set of PHM model, as follows.

$$T_{PHM} = \{t_i : \exists T_{node}, t_i \in T_{node}\} \quad (15)$$

To locate the fault condition when starting with the maximum set of failure modes, the corresponding sensor recording is collected and used to compute the potential basic probability assignment. Then, the subset of potential modes is determined based on existing samples with nearest neighbor strategy and the detection process is continued until finding the minimum failure sets and obtaining the failure detection. For each test sample denoted as $case_i$, the PHM detection procedures generate a test sequence corresponding to the assigned sensor recording and the estimation from the signal processing by a branch of assigned soft-sensor nodes, as follows:

$$S_{node} = \left\{ Node_j : \exists m_{exemplar,k} \in m_{exemplar}, \left\| f(x_{case_i, T_{node,j}} | ELM_{node,j}) - m_{exemplar,k} \right\| = d_{node} \right\} \quad (16)$$

where d_{Node_j} is the minimum distance from the failure condition estimation vector $f(x_{case_i, T_{node,j}} | ELM_{node,j})$ to all exemplars of soft-sensor nodes with the same father node

of $Node_j$. The detection process ends when the procedures reach the terminal node of S_{node} denoted as $Node_{terminal,case}$. Then, the estimated failure condition vector is computed as:

$$\hat{y}_{case,j} = \begin{cases} 1 & s_j \in S_{node_{terminal,case},father} \\ 0 & s_j \notin S_{node_{terminal,case},father} \end{cases} \quad (17)$$

According to the definitions of FDR, FAR, and detection accuracy, the test performance indexes of the detection procedures are computed as follows:

$$FDR_{test} = \sum_{X_{case}} \sum_{j=1}^K P(\hat{y}_{case,j} = 1 | y_{case,j} = 1) \quad (18)$$

$$FDR_{test} = \sum_{X_{case}} \sum_{j=1}^K P(\hat{y}_{case,j} = 0 | y_{case,j} = 1) \quad (19)$$

$$Accuracy_{test} = \sum_{X_{case}} \sum_{j=1}^K P(\hat{y}_{case,j} = 1 | y_{case,j} = 1) + P(\hat{y}_{case,j} = 0 | y_{case,j} = 0) \quad (20)$$

From these, the test strategy optimization aims to select T_{PHM} from $T_{potential}$ and generate the diagnostic tree with soft sensors and AP. For each case, AP determines the next procedure and the corresponding soft sensors with previous estimations. Thus, for each case, the diagnostic tree provides an adaptive test sequence and leads to the final evaluation on the terminal node. Similar to Equation (4), the objective function is the combination of FAR loss and FDR loss, as follows:

$$L_{tree} = \frac{\sum_{i=1}^M \sum_{j=1}^K (y_{i,j} - F_j(x_{i,T_{seq,i}} | T_{seq,i} \in T_{PHM}, ELM_{node} \in tree))^2}{\sum_{i=1}^M \sum_{j=1}^K y_{i,j}} \quad (21)$$

where $T_{seq,i}$ is the required test procedure with respect to the diagnostic tree, and $F_j(x_{i,T_{seq,i}} | T_{seq,i} \in T_{PHM}, ELM_{node} \in tree)$ is the state estimation from the terminal node with respect to the test procedure of the i -th case.

3. Test Strategy Optimization Based on Soft Sensing and Ensemble Belief Measurement

3.1. Construct Soft-Sensor Node with Extreme Learning Machine

As mentioned in the previous section, each soft sensor contains the recording information from the assigned sensors, the artificial intelligence signal processing modules, and probability estimation parameter for the isolation of fault states. During the construct process, maintenance design produces the soft-sensor node with candidate test procedures and candidate soft-sensor nodes with high performance, and generates the fault tree. For each candidate node, the sensor recording input is created as follows:

$$X_{candidate, node} = \{x_{s_i, T_i^*} : x_i \in X_{candidate, node}\} \quad (22)$$

$$x_{s_i, T_i^*} = \{x_{s_i, t} | t \in T_i^*\} \quad (23)$$

$$T_i^* = \{T_{sequence}^*, t^*\} \text{ where } \{t^*\} \cap T_{sequence}^* = \emptyset \quad (24)$$

where $T_{sequence}^*$ integrates the previous test information of before the candidate node and makes full use of sensor recording knowledge.

At the same time, we use ELM to generate artificial intelligence signal processing modules for fast training and high generation ability. Shown in Figure 4, ELM is a noniterative three-layer neural network and contains parameters of a fully connected hidden layer and a linear-combined output layer with an activation function, as follows:

$$ELM_{candidate, node} = \{(W_{node}, b_{node}), \beta_{node}, f_h(\cdot)\} \quad (25)$$

$$\mathbf{W}_{node} = \begin{bmatrix} w_{1,1} & \dots & w_{1,N_T} \\ w_{2,1} & \dots & w_{2,N_T} \\ \vdots & \ddots & \vdots \\ w_{L,1} & \dots & w_{L,N_T} \end{bmatrix} \quad (26)$$

$$\mathbf{b}_{node} = [b_{node,1}, b_{node,2}, \dots, b_{node,L}] \quad (27)$$

$$\boldsymbol{\beta}_{node} = \begin{bmatrix} \beta_{1,1} & \dots & \beta_{1,N_T} \\ \beta_{2,1} & \dots & \beta_{2,N_T} \\ \vdots & \ddots & \vdots \\ \beta_{L,1} & \dots & \beta_{L,N_T} \end{bmatrix} \quad (28)$$

where \mathbf{W}_{node} is denoted as the weights matrix of the hidden layer while \mathbf{b}_{node} is the hidden layer bias, and L is the number of hidden nodes. $\boldsymbol{\beta}_{node}$ is the output layer weight and $f_h(\cdot)$ determines the activation function from the sensor input and hidden output. Here, the sigmoid function is taken as the activation functions for all soft-sensor nodes. Relative to $\mathbf{X}_{candidate\ node}$, the hidden outputs of training samples are produced as follows:

$$\mathbf{H}_{node} = \begin{bmatrix} f_h(w_1 \mathbf{x}_{1,T^*} + b_{node,1}) & \dots & f_h(w_L \mathbf{x}_{1,T^*} + b_{node,L}) \\ f_h(w_1 \mathbf{x}_{2,T^*} + b_{node,1}) & \dots & f_h(w_L \mathbf{x}_{2,T^*} + b_{node,L}) \\ \vdots & \ddots & \vdots \\ f_h(w_1 \mathbf{x}_{M_{node},T^*} + b_{node,1}) & \dots & f_h(w_L \mathbf{x}_{M_{node},T^*} + b_{node,L}) \end{bmatrix} \quad (29)$$

As ELM is a noniterative learning machine, \mathbf{W}_{node} and \mathbf{b}_{node} can be assigned randomly with respect to arbitrary probability distribution, and the output of the model is computed as a linear combination of the hidden output with trained $\boldsymbol{\beta}_{node}$ as follows:

$$\dot{\mathbf{Y}}_{candidate\ node} = \boldsymbol{\beta}_{node} \mathbf{H}_{node} \quad (30)$$

To estimate the failure situation as accurately as possible, $\dot{\mathbf{Y}}_{candidate\ node}$ is supposed to be consistent with actual failure states \mathbf{Y}_{node} defined based on Equations (5)–(7). According to Equation (14), the loss function of the candidate node is computed as follows:

$$Loss_{candidate\ node} = \left\| \mathbf{Y}_{node} - \dot{\mathbf{Y}}_{candidate\ node} \right\|^2 = \left\| \mathbf{Y}_{node} - \boldsymbol{\beta}_{node} \mathbf{H}_{node} \right\|^2 \quad (31)$$

Taking differential of $Loss_{candidate\ node}$ with respect to $\boldsymbol{\beta}_{node}$, the trained output weight is accessed as follows:

$$\boldsymbol{\beta}_{node} = \left(\mathbf{H}_{node} \mathbf{H}_{node}^T + \lambda I \right)^{-1} \mathbf{H}_{node} \mathbf{Y}_{node} \quad (32)$$

Based on the proper assignment of ELM parameters, the soft-sensor nodes gain knowledge from the training samples and obtain accurate condition estimation with the test samples.

From above, considering the candidate node with sensor recording inputs $\mathbf{X}_{candidate, node}$, the previous test sequence $T_{sequence}^*$ and candidate test point t^* , the procedure to generate the ELM_{node} follows:

Step 1: Assign the candidate node with Equation (2), where $\mathbf{S}_{father} = \{s_i | s_{candidate\ node} = s_i, s_i \in \mathbf{S}\}$. Meanwhile, T_{node} is generated as Equation (24), \mathbf{Y}_{node} is assigned with Equations (5)–(7);

Step 2: Initialize ELM parameters ($\mathbf{W}_{node}, \mathbf{b}_{node}$) randomly in $[-1, 1]$;

Step 3: Calculate the hidden output with respect to $\mathbf{X}_{candidate, node}$ as in Equation (29);

Step 4: Train the output weights $\boldsymbol{\beta}_{node}$ with Equation (32);

Step 5: Obtain the estimation of candidate set $\dot{\mathbf{Y}}_{candidate\ node}$ with Equation (30).

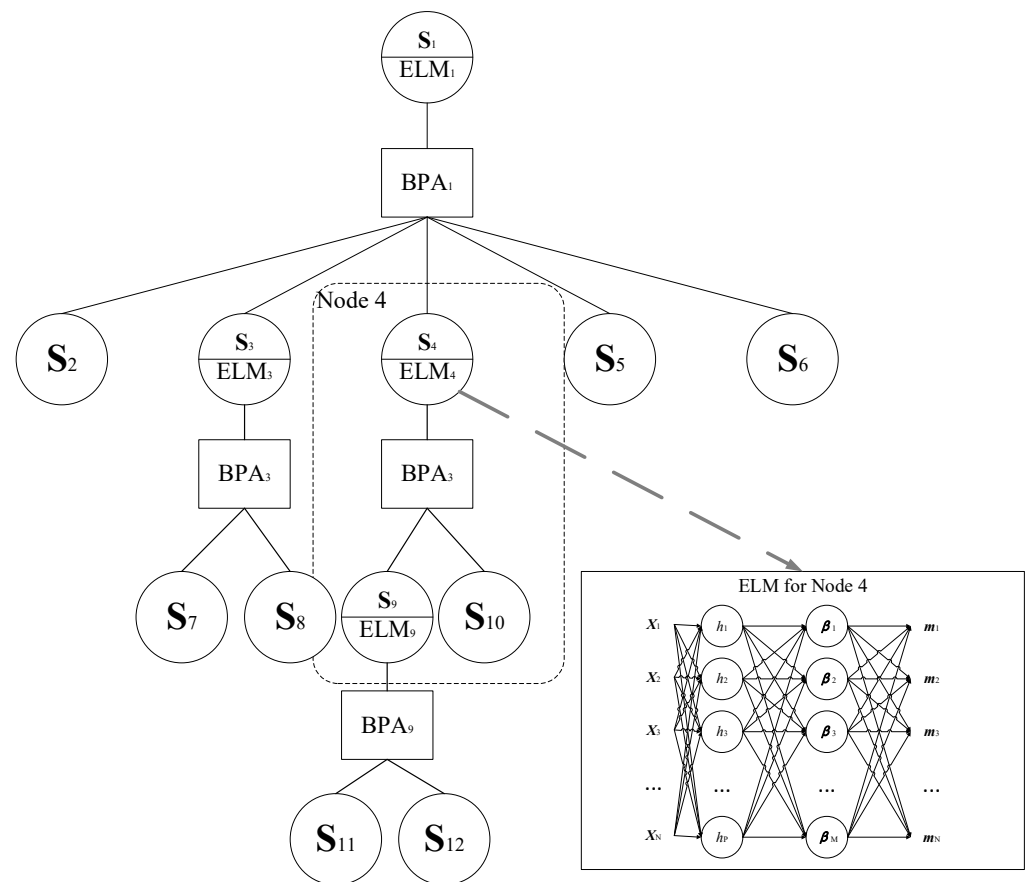


Figure 4. Fault tree of test strategy optimization based on soft sensing and ensemble belief measurement.

3.2. Separate the Fault Set Based on Affinity Propagation

With ELM-based soft-sensor nodes, the condition of trained samples and test samples can be estimated with high efficiency. Meanwhile, owing to the individual sensor recording knowledge limitation, the ELM condition evaluation has a vague part with unrelated failure modes. Thus, the fault set of corresponding nodes S_{father} is divided into several fuzzy sets $S_{son} = [S_{son,1}, S_{son,2}, \dots, S_{son,K_{node}}]$ based on the fault state evaluation value $\dot{Y}_{candidate\ node}$. When constructing traditional diagnostic tree and fault analysis processes, the failure mode subset is divided by comparing the fault state evaluation and reference value of the failure mode or the failure mode calibration value. However, these strategies are only applicable to systems with small structures or systems with known mechanisms and a historical sample may ignore the diversity and validity. Hence, in the proposed method, we introduce a new dividing strategy based on affinity propagation (AP) to cut the fuzzy set and samples with evaluation similarity measurement between pairs of data points. With AP, S_{son} it is constituted based on the similarity between condition estimates and the fault tree generation flexibility is enhanced.

Instead of assigning engineering-experience reference information, AP generates clusters based on all training set evaluation values $\dot{Y}_{candidate\ node} = \{y_1, y_2, \dots, y_{M_{node}}\}$. The clustering method treats each training sample as one data edge point and transmits two real-valued messages: the responsibility value $r(y_i, y_j)$ and the availability value $a(y_i, y_j)$, to realize communication between edge nodes until a good set of exemplars and corresponding clusters emerges. The responsibility value $r(y_i, y_j)$ indicates the accumulated evidence for how well suited a historical data point y_j is to serve as the exemplar for the historical data point y_i . In addition, the availability value $a(y_i, y_j)$ represents how appropriate it would be for a historical data point y_i to choose a historical data point y_j

as its exemplar. For initialization, the similarity between the historical points $(\mathbf{y}_i, \mathbf{y}_j)$ is calculated based on Euclidean distance, as follows:

$$d(\mathbf{y}_i, \mathbf{y}_j) = -\|\mathbf{y}_i - \mathbf{y}_j\|^2 \quad (33)$$

where $d(\mathbf{y}_i, \mathbf{y}_j)$ reflects how well the historical data point \mathbf{y}_j is suited to be the exemplar for historical data point \mathbf{y}_i . AP aims to provide a clustering solution that satisfies the historical data points with larger values of distance estimation, which are more likely to serve as exemplars. To achieve this purpose, the clustering method recursively conducts the following updating process, sending the responsibility message from each data point to the corresponding data point.

First, the responsibility value $r(\mathbf{y}_i, \mathbf{y}_j)$ is computed based on the following data driven approach:

$$r(\mathbf{y}_i, \mathbf{y}_j) = d(\mathbf{y}_i, \mathbf{y}_j) - \max_{k, s.t. k \neq j} \{a(\mathbf{y}_i, \mathbf{y}_k) + d(\mathbf{y}_i, \mathbf{y}_k)\} \quad (34)$$

As the availability value $a(\mathbf{y}_i, \mathbf{y}_j)$ is set to 0, the responsibility value $r(\mathbf{y}_i, \mathbf{y}_j)$ is initialized as the input similarity $d(\mathbf{y}_i, \mathbf{y}_j)$ minus the largest similarity value between \mathbf{y}_i and other exemplars. Hence, the updating process does not consider how many other points favor each candidate exemplar. In later process, if some point is efficient to assign with other exemplars, the corresponding availability value will drop to less than 0 with the updating of $a(\mathbf{y}_i, \mathbf{y}_j)$. Then, negative $a(\mathbf{y}_i, \mathbf{y}_j)$ will decrease the effective value of the similarities value $d(\mathbf{y}_i, \mathbf{y}_j)$ by Equation (34) and removes the corresponding candidate exemplars from competition. Especially, the self-responsibility value $r(\mathbf{y}_i, \mathbf{y}_j)$ is set to the input preference that the training data point \mathbf{y}_i becomes one of the exemplars of clusters and reflects accumulated evidence that \mathbf{y}_i is an exemplar based on its input preference tempered by how ill-suited it is for assignment to another exemplar.

After calculating the responsibility value, the availability value is updated to gather evidence from the training data point as to whether each candidate exemplar makes a good exemplar, as follows:

$$a(\mathbf{y}_i, \mathbf{y}_j) = \min\{0, d(\mathbf{y}_i, \mathbf{y}_j) + \max_{k, s.t. k \neq j} \{a(\mathbf{y}_i, \mathbf{y}_k) + d(\mathbf{y}_i, \mathbf{y}_k)\}\} \quad (35)$$

In addition, the self-availability value $a(\mathbf{y}_i, \mathbf{y}_j)$ is updated as follows:

$$a(\mathbf{y}_i, \mathbf{y}_j) = \sum_{k, s.t. k \neq j, k \neq i} \max\{0, r(\mathbf{y}_i, \mathbf{y}_k)\} \quad (36)$$

where $a(\mathbf{y}_i, \mathbf{y}_j)$ reflects accumulated evidence that the training data point \mathbf{y}_i becomes an exemplar. For data point \mathbf{y}_i , the data point \mathbf{y}_j that maximizes $a(\mathbf{y}_i, \mathbf{y}_j) + r(\mathbf{y}_i, \mathbf{y}_j)$ is chosen to be the exemplar. Additionally, if $i = j$, then it is necessary to identify the data point \mathbf{y}_j as the exemplar and assign its estimation value \mathbf{y}_j as the exemplar value $\mathbf{m}_{exemplar}$. The set of all the $\mathbf{m}_{exemplar}$ is denoted as $\mathbf{M}_{exemplar}$. Based on each exemplar, one subset of \mathbf{S}_{father} satisfies.

$$\begin{aligned} \forall \mathbf{m}_{exemplar} \in \mathbf{M}_{exemplar}, \exists \mathbf{S}_{son,k} \subset \mathbf{S}_{son,k} \text{ s.t. } \mathbf{S}_{son,k} \\ = \{s_i : \exists \mathbf{x}_j \in \mathbf{X}_{node}, a(\mathbf{y}_i, \mathbf{m}_{exemplar}) + r(\mathbf{y}_i, \mathbf{m}_{exemplar})\} \\ = \operatorname{argmax} \{a(\mathbf{y}_i, \mathbf{y}_j) + r(\mathbf{y}_i, \mathbf{y}_j)\}. \end{aligned} \quad (37)$$

From above, the AP process with respect to the candidate node is conducted as follows:

- Step 1: For $\dot{y}_i, \dot{y}_j \in \dot{Y}_{candidate\ node}$, initialize the responsibility value $r(\dot{y}_i, \dot{y}_j)$ as the Euclidean distance $d(\dot{y}_i, \dot{y}_j)$ with Equation (33), and set the availability value $a(\dot{y}_i, \dot{y}_j)$ to zero;
- Step 2: Update the responsibility value $r(\dot{y}_i, \dot{y}_j)$ with Equation (34);
- Step 3: Update the availability value $a(\dot{y}_i, \dot{y}_j)$ with Equations (35) and (36);
- Step 4: If $r(\dot{y}_i, \dot{y}_j)$ and $a(\dot{y}_i, \dot{y}_j)$ become stable, conduct Step 5, otherwise return to Step 2;
- Step 5: For each sample $\dot{y}_i \in \dot{Y}_{candidate\ node}$, assign the sample data corresponding to $a(\dot{y}_i, \dot{y}_j) + r(\dot{y}_i, \dot{y}_j)$ as exemplar node $m_{exemplar}$ and then generate the exemplar set $M_{exemplar}$;
- Step 6: Separate S_{father} with Equation (37).

3.3. Generate the Fault Diagnostic Tree under Minimum Conditional Criterion

Based on soft-sensor construction and subset division, the fault states of the target system can be located by cutting the set of potential failure sets with the function of sequences of soft-sensor nodes. In this section, we introduce how to generate the fault tree using a potential soft-sensor node under heuristic strategy based on the minimum conditional criterion. For the assigned failure set S_{father} , the contains numbers of potential sensor nodes corresponding to the candidate procedures. To choose the soft sensor for fault tree construction, the condition entropy $H(Y_{node} | \dot{Y}_{node}, ELM_{candidate})$ is introduced as follows:

$$H(Y_{node} | \dot{Y}_{node}, ELM_{candidate}) = - \sum_{y \in Y_{node}} p(y, \dot{y} | ELM_{candidate}) \log(p(y, \dot{y} | ELM_{candidate})) \quad (38)$$

Since the soft-sensor model is data-driven, the estimation value of the conditional entropy is computed as follows:

$$H(Y_{node} | \dot{Y}_{node}, ELM_{candidate}) = \sum_{y \in Y_{node}} \log p(y | \dot{y}, ELM_{candidate}) \quad (39)$$

Assuming the data information $(Y_{node}, \dot{Y}_{node})$ is subject to Gaussian distribution, then $\dot{H}(Y_{node} | \dot{Y}_{node}, ELM_{candidate})$ is simplified as follows:

$$\dot{H}(Y_{node} | \dot{Y}_{node}, ELM_{candidate}) = \sum_{y \in Y_{node}} \left\| Y_{node} - \dot{Y}_{node} \right\|^2 \quad (40)$$

For all candidate soft-sensor nodes, the node with the lowest conditional entropy is selected to build the fault diagnostic tree. From above, the process to construct the fault diagnostic tree follows:

- Step 1. Initialize the root node $Node_{root}$ of the decision tree. Assign the father fault set S_{father} as the total fault set S_{total} and set the data set X_{node} as the whole training data set. Take all the test procedures as the potential test set $T_{potential}$ for $Node_{root}$. Set $T_{sequence}^*$ to \emptyset .
- Step 2. Generate the potential soft-sensor node $Node_{candi, t_{potential}}$ for each test procedure $t_{potential} \in T_{potential}$ based on Equations (24), (28) and (31).
- Step 3. Evaluate with Equation (39) the condition entropy $\dot{H}(Y_{node} | \dot{Y}_{node}, ELM_{candidate})$ for the entire potential soft-sensor node. Select the corresponding soft-sensor node $Node_{candi, t_{potential}}$ with the lowest condition entropy as the target node to the diagnostic tree, and update the $T_{sequence}^*$ as $\{T_{sequence}^*, t_{opt}\}$. Additionally, remove t_{opt} from $T_{potential}$.
- Step 4. Apply AP to separate the fault set S_{father} based on \dot{Y}_{node} and assign the exemplar reference $M_{exemplar}$ with Equation (36).

- Step 5. For each subset S_i in S_{son} , construct the extending node $Node_{S_i}$. For each extending node $Node_{S_i}$, the father set is assigned as S_i and the data set is constructed based on AP result. $T_{sequence}^*$ is initialized as $\{T_{sequence}^*, t_{opt}\}$.
- Step 6. Generate the subset node by repeating Steps 2 to 5 until reaching the minimal subset of failure mode. The construction process is completed when all the subnodes of the failure tree are constructed.

With the minimum conditional criterion, the fault diagnostic tree is generated with data-driven mechanisms and requires few engineering experiences. Thus, the generating process is applicable to complex systems with insufficient knowledge about structures, functions, and mechanisms.

With the diagnostic tree generated, the diagnostic process of the target system is implemented as follows:

- Step 1. Initialize the target node $Node_{target}$ with the root node $Node_{root}$ of the diagnostic tree. Assign the potential fault set $S_{potential}$ to the total fault set S_{total} . Set the target sensor recording x_{target} as \emptyset and set the test sequence $T_{sequence}$ to \emptyset .
- Step 2. Conduct the new test procedure t^* in the $Node_{target}$ and obtain the sensor recording x_{target,t^*} . Add t^* into $T_{sequence}$ and merge x_{target,t^*} into x_{target} . Compute the estimation of the target system y_{target} with ELM_{node} in $Node_{target}$ by Equations (29) and (30).
- Step 3. Find the optimal $m_{exemplar}$ in $M_{exemplar}$ of $Node_{target}$ with smallest distance. Locate the subfault set S^* as the updated $S_{potential}$ with Equation (37).
- Step 4. Search the soft-sensor node in the diagnostic tree with $S_{father} = S_{potential}$ and assign the corresponding node as the new target node $Node_{target}$.
- Step 5. Continue the diagnostic procedure by steps 2 to 4 until reaching the minimal subset of the failure set. After the diagnosis is finished, achieve the final estimation by using Equation (17).

4. Experiment

In this section, we use the analog circuit in [40] to evaluate the detection performance of the proposed method with state-of-art methods. As Figure 5 shows, the target system contains four second-order filters and one adding device. The detail of the system is presented in Table 1. The tolerance of $R_1, R_2, R_3, R_4, R_5, R_6, R_7,$ and R_8 is $\pm 10\%$ while the tolerance of $R_9, R_{10},$ and R_{11} is $\pm 1\%$. For capacitances, the tolerance is set to $\pm 5\%$. Under healthy operation, the transmission gain of $Av_1, Av_2, Av_3,$ and Av_4 is within a range of $\pm 1\%$.

Here, the failures caused by different changes of amplifiers are taken into failure detection. The failure modes are defined based on the range of transmission gain for $Av_1, Av_2, Av_3,$ and Av_4 , as shown in Table 2. Since 80% of failures in real applications have a single-failure mode, we only consider failure detection of single failure modes. For example, the failure condition of Av_1 is divided into five phases with different ranges of transmission gain while the transmission gain of $Av_2, Av_3,$ and Av_4 , are collected with four different frequencies (10 Hz, 100 Hz, 10 kHz, and 100 kHz) of input signals. The details are shown in Table 3. These voltage outputs from $Av_1, Av_2, Av_3,$ and Av_4 are regarded as the potential test points for failure detection. In total, there are 16 candidate test points and 17 potential fault states that consider the health state.

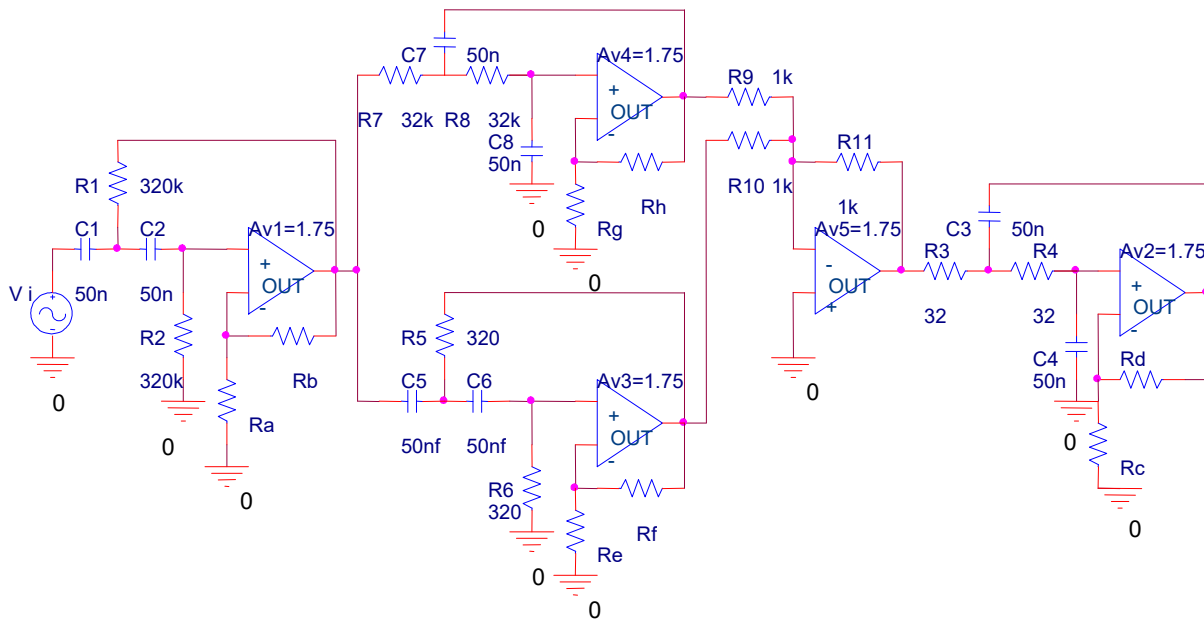


Figure 5. Target analog circuit.

Table 1. Details for the components of the circuit.

Components	Nominal Value	Tolerance	Subsystem
R ₁	320 kΩ	10%	High-Pass Filter 1 F ₁ = 10 Hz
R ₂	320 kΩ	10%	
C ₁	50 nF	5%	
C ₂	50 nF	5%	
Av ₁	1.75	1%	Low-Pass Filter 1 F ₂ = 100 kHz
R ₃	32 Ω	10%	
R ₄	32 Ω	10%	
C ₃	50 nF	5%	
C ₄	50 nF	5%	High-Pass Filter 2 F ₃ = 10 kHz
Av ₂	1.75	1%	
R ₅	320 Ω	10%	
R ₆	320 Ω	10%	
C ₅	50 nF	5%	Low-Pass Filter 2 F ₄ = 100 Hz
C ₆	50 nF	5%	
Av ₃	1.75	1%	
R ₇	32 kΩ	10%	
R ₈	32 kΩ	10%	Adder
C ₇	50 nF	5%	
C ₈	50 nF	5%	
Av ₄	1.75	1%	
R ₉	1 kΩ	1%	
R ₁₀	1 kΩ	1%	
R ₁₁	1 kΩ	1%	

Table 2. Denotation of fault states.

Fault Index	Av ₁ Value Range	Av ₂ Value Range	Av ₃ Value Range	Av ₄ Value Range
S ₀ (normal)	(1.70,1.80)	(1.70,1.80)	(1.70,1.80)	(1.70,1.80)
S ₁	(1.60,1.70)	(1.70,1.80)	(1.70,1.80)	(1.70,1.80)
S ₂	(1.80,1.90)	(1.70,1.80)	(1.70,1.80)	(1.70,1.80)
S ₃	(1.50,1.60)	(1.70,1.80)	(1.70,1.80)	(1.70,1.80)
S ₄	(1.90,2.00)	(1.70,1.80)	(1.70,1.80)	(1.70,1.80)
S ₅	(1.70,1.80)	(1.60,1.70)	(1.70,1.80)	(1.70,1.80)
S ₆	(1.70,1.80)	(1.80,1.90)	(1.70,1.80)	(1.70,1.80)
S ₇	(1.70,1.80)	(1.50,1.60)	(1.70,1.80)	(1.70,1.80)
S ₈	(1.70,1.80)	(1.90,2.00)	(1.70,1.80)	(1.70,1.80)
S ₉	(1.70,1.80)	(1.70,1.80)	(1.60,1.70)	(1.70,1.80)
S ₁₀	(1.70,1.80)	(1.70,1.80)	(1.80,1.90)	(1.70,1.80)
S ₁₁	(1.70,1.80)	(1.70,1.80)	(1.50,1.60)	(1.70,1.80)
S ₁₂	(1.70,1.80)	(1.70,1.80)	(1.90,2.00)	(1.70,1.80)
S ₁₃	(1.70,1.80)	(1.70,1.80)	(1.70,1.80)	(1.60,1.70)
S ₁₄	(1.70,1.80)	(1.70,1.80)	(1.70,1.80)	(1.80,1.90)
S ₁₅	(1.70,1.80)	(1.70,1.80)	(1.70,1.80)	(1.50,1.60)
S ₁₆	(1.70,1.80)	(1.70,1.80)	(1.70,1.80)	(1.90,2.00)

We apply Monte Carlo simulation to generate 20 samples for each failure mode by using Pspice software to access data.

According to the traditional PHM framework shown in Figure 3, the optimization process requires binary fault marks based on human experience and design detection circuits for each fault state. Since there are four fault states for each second-order filter, the detection is a large burden on circuit design. Testability design may also fail to relate the relationship between binary estimations of different test procedures. On the other hand, the maintenance design suffers low estimation efficiency as the testability does not consider the detailed information of sensor recordings.

Unlike the sequential framework, the proposed method considers the direct sensor information under testability and maintenance design. With cooperative procedures in Figure 4, the proposed method generates the candidate soft-sensor nodes for maintenance. At the same time, the testability design uses the minimal conditional criterion to generate the optimized diagnostic strategy. The minimal conditional criterion enhances the flexibility of testability design by considering the soft-sensor estimation in maintenance phases. On the other hand, the maintenance performance with full sensor recording increases information usage efficiency. Instead of human-experience processing, the proposed method saves many costs during the PHM design. To evaluate diagnostic performance, we compared our method with the hidden Markov method (HMM), support learning machine (SVM), and radial basis function (RBF) by using all recordings of 16 test points as input information. To estimate the feature extraction performance and learning machine function, we also took HMM and SVM with principal component analysis (PCA) and extreme learning machine (ELM) into comparison. For each method, we used 70% of the samples in each fault state as the training samples to construct the model and the other 30% as the sensor information of target systems. We assigned 100 kernels or hidden nodes for our soft-sensor nodes for the SVM, RBF, and ELM models. Each method was conducted 30 times to obtain the average performance.

Table 3 shows the FAR, FDR, and accuracy of each method. HMM has high diagnostic accuracy without feature extraction, especially for S_0 . This is because HMM has a higher statistical analysis ability than SVM and RBF. However, HMM gives poorer FAR than the other methods. Unlike HMM and RBF, SVM and ELM have lower FAR and become more sensitive to false-negative samples by the advantage of the learning machine. Comparing HMM, SVM, PCA-HMM, and PCA-SVM, PCA improves HMM diagnosis for S_5 and S_8 and proves that proper feature extractions can benefit from the diagnostic performance. Our method has the lowest FAR, the highest FDR, and the highest accuracy compared with the other methods. Based on the same sensor recordings or even less information, the generated strategy provides an accurate location for all test samples for all 16 fault states. Hence, with ensemble learning based on soft sensors, the functions of sensor recording are largely improved.

Figure 6 shows the diagnostic tree of our method. All fault conditions are separated and recognized with 13 individual testing sequences with the tree structure. Each testing sequence takes less than 5 test procedures and the whole diagnostic tree contains only 9 testing points out of 16 potential test points. In other words, the fault state of the target systems is located within 2–5 test procedures instead of collecting all 16 sensor recordings. From above, the diagnostic tree has higher efficiency and lower cost testing consumptions than diagnostic strategies that require full-tests built by SVM, HMM, and ELM, as well as the constrained diagnostic tests strategies that require PCA-based methods. Additionally, unlike traditional diagnostic trees with binary structures, our method separates the fuzzy set with the evaluation results from constructing and dividing modules. As a result, our testability design extends the diagnostic flexibility and improves the diagnostic accuracies of each fault mode.

Based on soft-sensor construction, the potential test accuracies of the diagnostic tree root node are compared based on different test procedures. As shown in Figure 7, different test procedures differ on diagnostic accuracies, especially for S_0 – S_5 . Therefore, the minimum conditional criterion-based fault tree constructions can efficiently select proper soft-sensor nodes into the diagnostic tree and ensure that the diagnostic FAR and FDR and their accuracies receive high improvement under the constructing process.

The affinity propagation result from the diagnostic-tree-root node is presented in Figure 8. As the root nodes evaluate all samples, the affinity results are generated with fault mode evaluations of all 17 fault conditions (from S_0 to S_{16}). Here, we depict the affinity propagation result from three fault states, which occur at the same space, such as Av_1 failures (S_0 , S_1 , and S_2), Av_2 failures (S_5 , S_7 , and S_{14}), and Av_4 failures (S_{12} , S_{13} , and S_{15}). Additionally, we present the affinity propagation results with three failure modes in different places: S_2 in Av_1 , S_7 in Av_2 , and S_{14} in Av_4 .

From Figure 8, the soft-sensor nodes provide distinguishable BPA evaluations for each data point. The affinity propagation generates sample clusters from different dimensions adaptively with the data point similarity measurements. Most data points are clustered with topological closer clusters; others may differ on other dimensions. Compared with traditional clustering strategies for diagnosis, affinity propagation provides practical, automatic fault-state divisions and saves much human labor on engineering applications.

Finally, the sequence testing performance for fault states is shown in Figure 9. Here, we present the test sequence of S_0 , S_3 , S_8 , and S_{15} . These four test sequences achieve their best diagnostic accuracy within three to five test procedures, and the test efficiency is much higher than traditional maintenance methods. Test accuracy grows increasingly for all test sequences as test nodes are added into the test procedures. Especially for the test sequence of S_0 , the diagnostic accuracy is less than 60% in the first test procedures as the corresponding fuzzy state set contains many members. However, the accuracy grows fast as more nodes are added into the sequences and the potential states become smaller. Thus, the ensemble function of soft-sensor nodes improved the diagnostic performance with high efficiency.

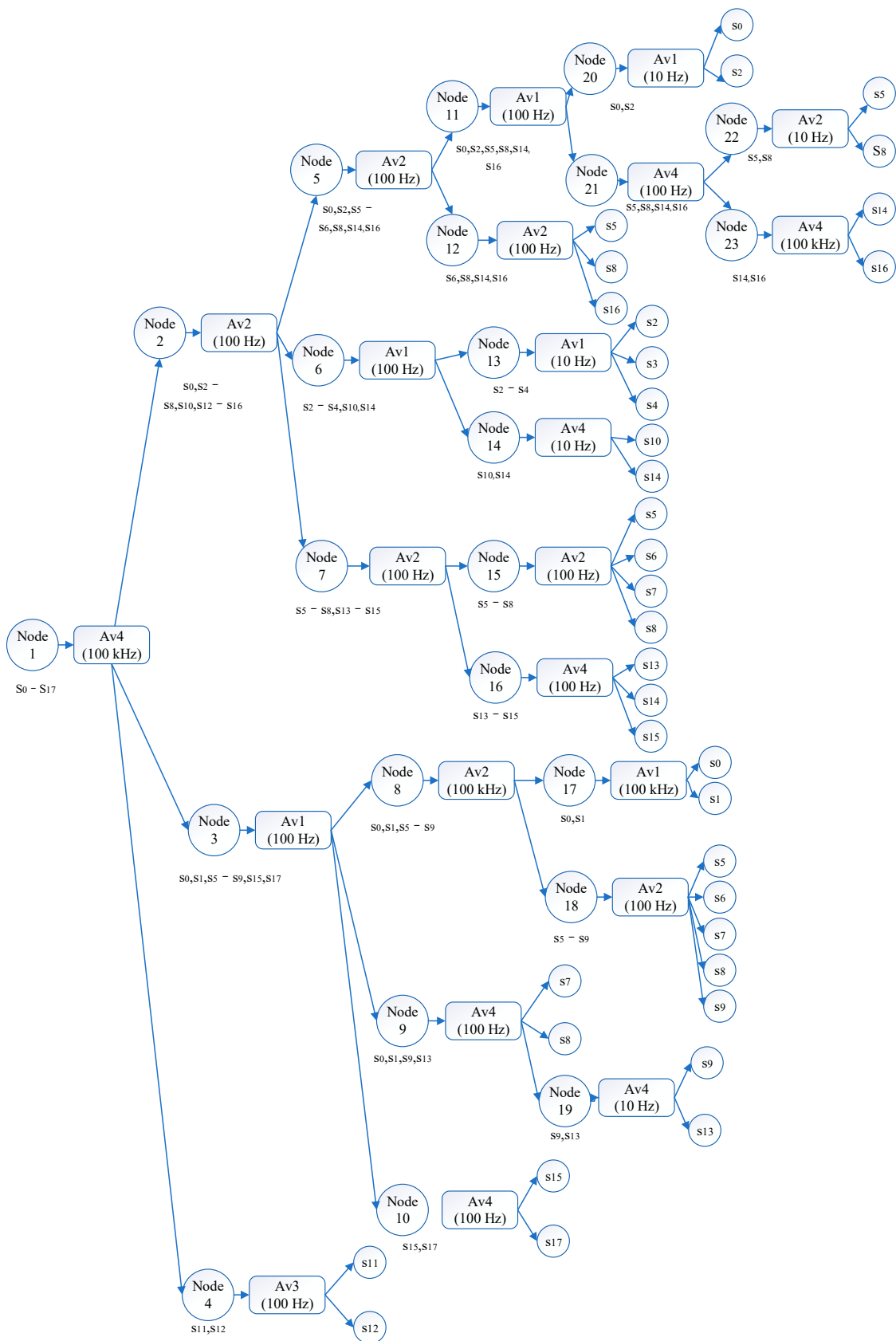


Figure 6. Analog circuit diagnostic tree.

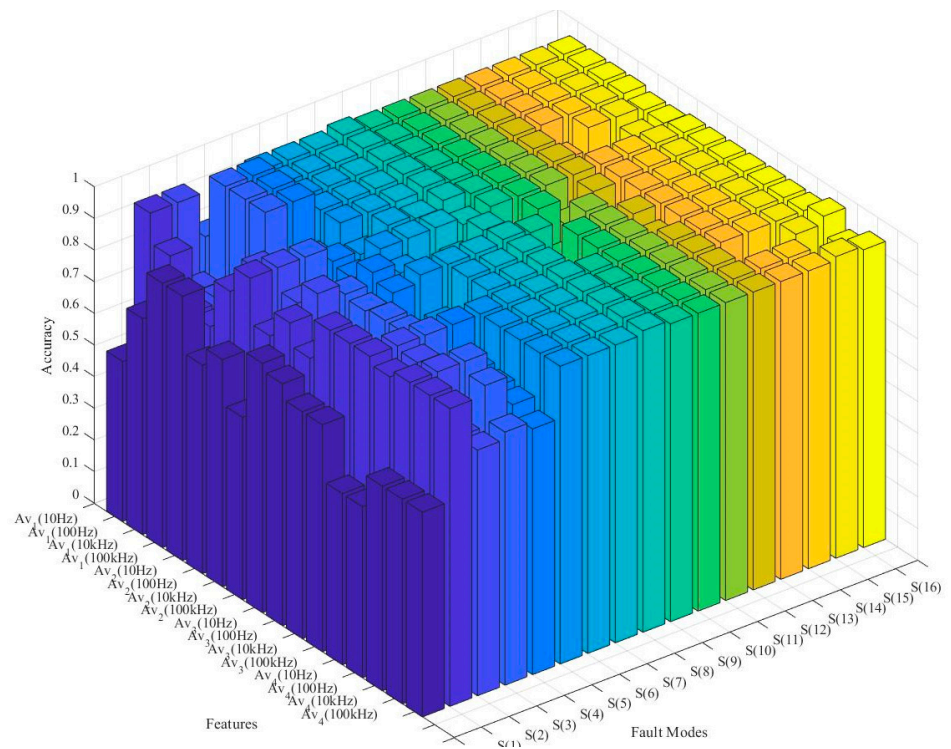


Figure 7. Analog circuit diagnostic tree potential ELM model accuracy comparison.

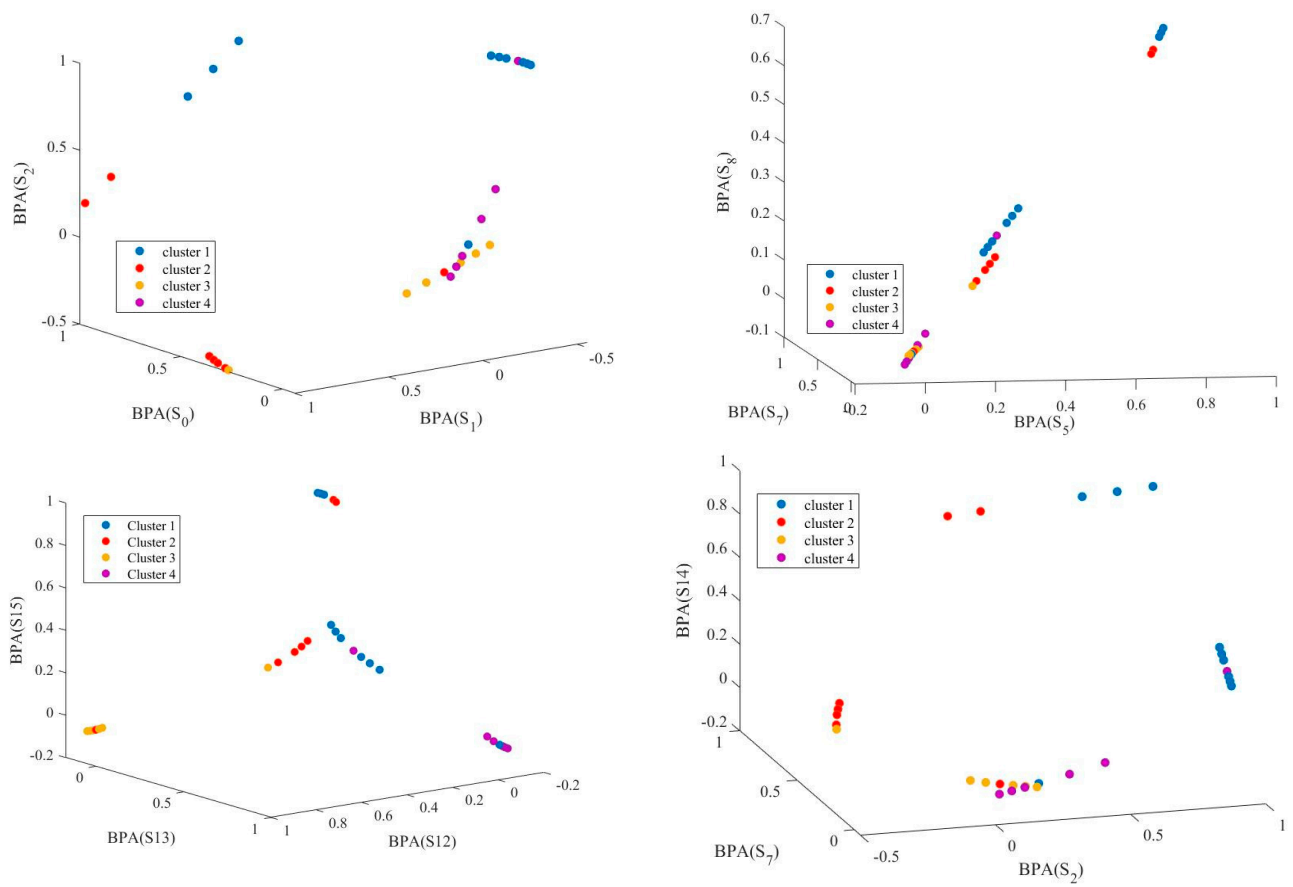


Figure 8. Potential ELM model accuracy comparison.

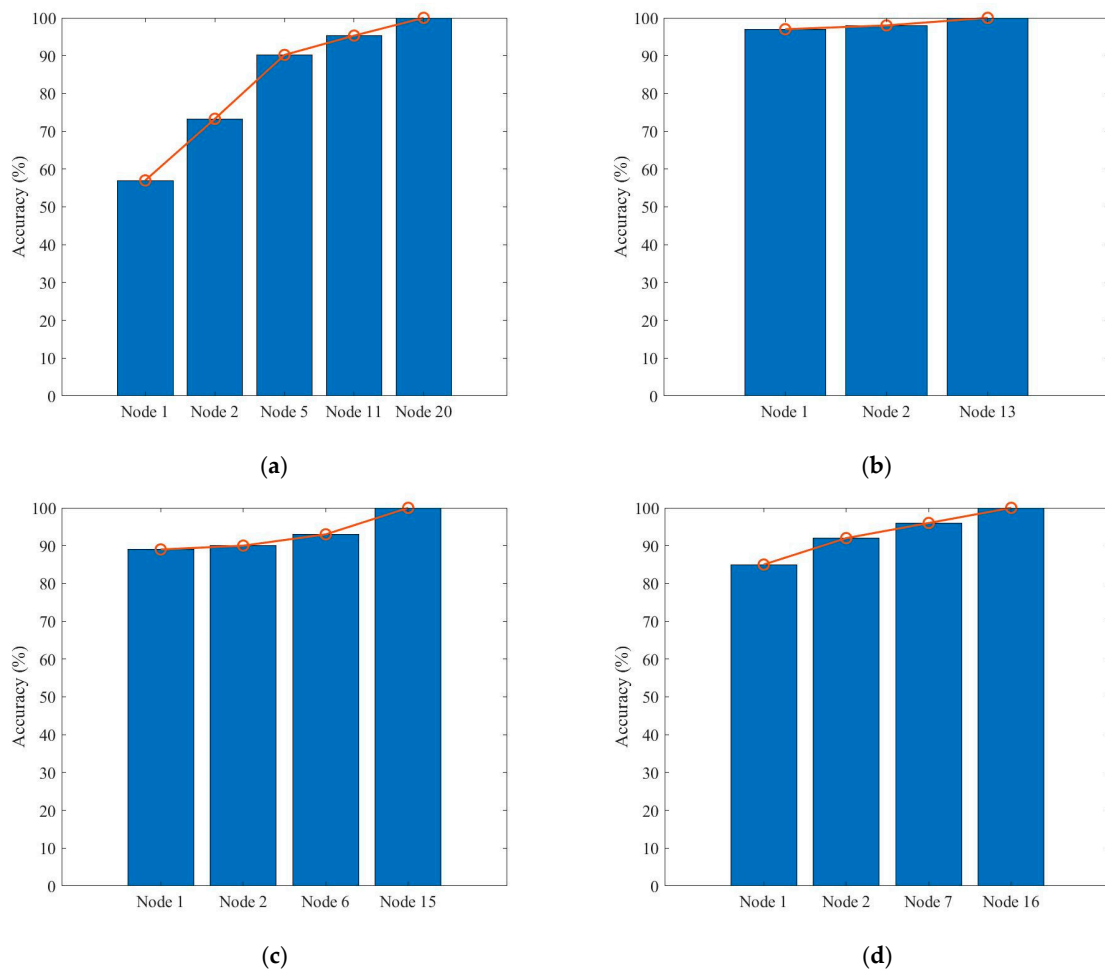


Figure 9. Test sequence accuracy comparison: (a) S_0 test sequence, (b) S_3 test sequence, (c) S_8 test sequence, and (d) S_{15} test sequence.

From above, our method has better diagnostic accuracies and lower FARs compared with other state-of-art diagnostic methods. Additionally, our diagnostic strategies take only 9 out of 16 tests points and save much test assignment consumption. For each fault state, the diagnostic procedures provide 1 test sequence within 5 test procedures. Thus, the diagnostic procedure enjoys high efficiency for applications. Finally, the affinity propagation enlarges the diagnostic flexibility and saves much human labor on testability design.

5. Conclusions

Along with a short production cycle and rapid development of design technology, existing PHM techniques have become impractical and fail to match the structural and functional complexity. Prior knowledge preparation costs too much in human labor and binary decision-making strategies waste the entailed sensor recording, especially for large complicated systems.

We propose a test strategy optimization based on soft sensing and ensemble belief measurement to overcome these weaknesses. The proposed method constructs a closed loop between testability design and maintenance design, generating an efficient fault diagnostic tree with ELM-based soft-sensor nodes. Unlike traditional diagnostic approaches, our diagnostic tree adaptively separates the fault sets by affinity propagation, and the soft-sensor nodes are assigned with the minimum conditional criterion. Thus, our methods can achieve high efficiency and flexibility for diagnostic processes.

The experiment results prove that our methods have minimum FAR and maximum accuracies on fault diagnosis among state-of-art methods. Additionally, our methods

require fewer test procedures and increase the test efficiency compared with other methods. Because the construction processes are based on ELM and AP, the PHM design saves much human labor and becomes more flexible compared to traditional PHM approaches. Hence, the proposed method has good performance on test strategy design. However, the proposed method uses an offline construction technique for the diagnostic tree. As a result, the diagnostic performance only depends on the assigned fault set, and the recordings of online operations do not work on the PHM design. Therefore, the online updating of the diagnostic strategy should be further investigated.

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

Funding: This work was supported by the National Natural Science Foundation of China under Grant No. U1830133(NSFC) and the Project of Sichuan Youth Science and Technology Innovation Team, China (Grant No. 2020 JDTD0008).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available on request from the authors.

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

References

- Braunfelds, J.; Senkans, U.; Skels, P.; Janeliukstis, R.; Salgals, T.; Redka, D.; Lyashuk, I.; Porins, J.; Spolitis, S.; Haritonovs, V.; et al. FBG-based sensing for structural health monitoring of road infrastructure. *J. Sens.* **2021**, *2021*, 8850368. [\[CrossRef\]](#)
- Suryasarman, V.M.; Biswas, S.; Sahu, A. Automation of test program synthesis for processor post-silicon validation. *J. Electron. Test.* **2018**, *34*, 83–103. [\[CrossRef\]](#)
- Liu, H.; Chen, X.; Duan, C.; Wang, Y. Failure mode and effect analysis using multi-criteria decision-making methods: A systematic literature review. *Comput. Indust. Eng.* **2019**, *135*, 881–897. [\[CrossRef\]](#)
- Yin, T.; Huang, D.; Fu, S. Intrinsic determinants and harmonic measure of flight safety. *IEEE Trans. Ind. Electron.* **2019**, *67*, 881–897. [\[CrossRef\]](#)
- Zhu, Z.; Pan, Y.; Zhou, Q.; Lu, C. Event-Triggered adaptive fuzzy control for stochastic nonlinear systems with unmeasured states and unknown Backlash-like hysteresis. *IEEE Trans. Fuzzy Syst.* **2020**, *19*, 1273–1283. [\[CrossRef\]](#)
- Roman, R.C.; Precup, R.E.; Petriu, E.M. Hybrid data-driven fuzzy active disturbance rejection control for tower crane systems. *Eur. J. Control* **2021**, *58*, 373–387. [\[CrossRef\]](#)
- Li, S.; Yu, G.; Tang, D.; Li, M.; Han, H. Vibration diagnosis and treatment for a crubber system connected to a reciprocating compressor. *J. Sens.* **2020**, *2020*, 6697295. [\[CrossRef\]](#)
- Boumen, R.R.; Ruan, S.; De Jong, I.S.M.; Van De Mortel-Fronczak, J.A.; Rooda, J.K.; Pattipati, K.R. Hierarchical testing sequencing for complex systems. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2009**, *39*, 640–649. [\[CrossRef\]](#)
- Garey, M.R. Optimal binary identification procedures. *SIAM J. Appl. Math.* **1972**, *23*, 173–186. [\[CrossRef\]](#)
- Unler, A.; Murat, A. A discrete particle optimization method for feature selection in binary classification problem. *Eur. J. Oper. Res.* **2010**, *206*, 528–539. [\[CrossRef\]](#)
- Pattipati, K.R.; Alexandrid, M.G. Application of heuristic search and information theory to sequential fault diagnosis. *IEEE Trans. Syst. Man Cybern.* **1990**, *20*, 872–887. [\[CrossRef\]](#)
- Xu, S.; Bi, W.; Zhang, A.; Mao, Z. Optimization of flight test tasks allocation and sequencing using genetic algorithm. *Appl. Soft Comput.* **2022**, *115*, 108241. [\[CrossRef\]](#)
- Sun, L.; Zhang, X.; Qian, Y.; Xu, J.; Zhang, S. Feature selection using neighborhood entropy-based uncertainty measures for gene expression data classification. *Inf. Sci.* **2019**, *502*, 18–41. [\[CrossRef\]](#)
- Zhang, S.; Song, L.; Zhang, W.; Hu, Z.; Yang, Y. Optimal sequential diagnostic strategy generation considering test placement cost for multi-mode systems. *Sensors* **2015**, *15*, 25592–25606. [\[CrossRef\]](#) [\[PubMed\]](#)
- Yang, C.; Liu, Z.; Cong, H. Fault diagnosis of analog filter circuit based on genetic algorithm. *IEEE Access* **2019**, *7*, 54969–54980. [\[CrossRef\]](#)
- Fang, T.U.; Pattipati, K.R. Rollout strategies for sequential fault diagnosis. *IEEE Trans. Syst. Man Cybern.* **2003**, *33*, 86–99. [\[CrossRef\]](#)

17. Kundakcioly, O.E.; Unluyurt, T. Bottom-up construction of minimum cost AND/OR trees for sequential fault diagnosis. *IEEE Trans. Syst. Man Cybern.* **2007**, *37*, 621–629. [[CrossRef](#)]
18. Roy, S.; Rashid, A.U.; Abbasi, A.; Murphree, R.C.; Hossain, M.M.; Faruque, A.; Metreveli, A.; Zetterling, C.-M.; Fraley, J.; Sparkman, B.; et al. Silicon carbide bipolar analog circuits for extreme temperature signal conditioning. *IEEE Trans. Electron. Dev.* **2019**, *66*, 3764–3770. [[CrossRef](#)]
19. Yao, F.; Ma, J.; Tang, S.H. IGBT module bonding damage mechanism, evolution law and condition monitoring. *Chin. J. Sci. Instrum.* **2019**, *40*, 88–99.
20. Shi, Y.; Li, Q.; Meng, X.; Zhang, T.; Shi, J. On time-series InSAR by SA-SVR algorithm: Prediction and analysis of mining subsidence. *J. Sens.* **2020**, *2020*, 8860225. [[CrossRef](#)]
21. Liang, Y.; Yang, S.; Wen, X.; Yin, H.; Huang, L. Research on defect quantification technology in pulsed eddy current non-destructive testing. *Chin. J. Sci. Instrum.* **2018**, *39*, 70–78.
22. Samie, M.; Perinpanayagam, S.; Alghassi, A.; Motlagh, A.M.; Kapetanios, E. Developing prognostic models using duality principles for DC-DC converters. *IEEE Trans. Power Electron.* **2015**, *30*, 2872–2884. [[CrossRef](#)]
23. Seixas, A.; Cesar, R.; Flesch, C.A. Data-driven soft sensor for the estimation of sound power levels of refrigeration compressors through vibration measurements. *IEEE Trans. Ind. Electron.* **2020**, *67*, 7065–7120.
24. Senguta, E.; Jain, N.; Garg, D.; Choudhury, T. A review of payment card fraud detection methods using artificial intelligence. In Proceedings of the International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS), Belagavi, India, 21–23 December 2018; pp. 494–499.
25. Ampatzidis, Y.; Partel, V.; Meyering, B.; Albercht, U. Citrus rootstock evaluation utilizing UAV-based remote sensing and artificial intelligence. *Comput. Electron. Agric.* **2019**, *164*, 104900. [[CrossRef](#)]
26. Yue, D.; Han, Q. Guest editorial special issue on new trends in energy internet: Artificial intelligence-based control, network security and management. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 1551–1553. [[CrossRef](#)]
27. Gupta, P.; Batra, S.S. Sparse short-term time series forecasting models via minimum model complexity. *Neurocomputing* **2017**, *243*, 1–11. [[CrossRef](#)]
28. Jia, L.; Tao, R.; Wang, Y.; Wada, K. Forward prediction solution for adaptive noisy FIR filtering. *Sci. China Inf. Sci.* **2001**, *52*, 10071014. [[CrossRef](#)]
29. Zhang, J.S.; Xiao, X.C. A reduced parameter second-order volterra filter with application to nonlinear adaptive prediction of chaotic time series. *Acta Phys. Sin.* **2001**, *50*, 1248–1254. [[CrossRef](#)]
30. Liu, C.; Dong, J. Simultaneous fault detection and containment control design for multi-agent systems with multi-leaders. *J. Frankl. Inst.* **2020**, *357*, 9063–9082. [[CrossRef](#)]
31. Yin, L.; He, Y.; Dong, X.; Lu, Z. Multi-step prediction of volterra neural network for traffic flow based on chaos algorithm. In Proceedings of the 3rd International Conference on Information Computing and Applications, Chengde, China, 14–16 September 2012; pp. 232–241.
32. Ge, Z. Mixture Bayesian regularization of PCR model and soft-sensing application. *IEEE Trans. Ind. Electron.* **2015**, *62*, 4336–4343. [[CrossRef](#)]
33. Geng, H.; Liang, Y.; Liu, Y.; Alsaadi, F.E. Bias estimation for asynchronous multi-rate multi-sensor fusion with unknown inputs. *Inf. Fusion* **2018**, *39*, 15–139. [[CrossRef](#)]
34. Hong, W.-C. Chaotic particle swarm optimization algorithm in a support vector regression electric load forecasting model. *Energ. Convers. Manag.* **2009**, *50*, 105–117. [[CrossRef](#)]
35. Liang, Z.; Zhang, L. Intuitionistic fuzzy twin support vector machines with the insensitive pinball loss. *Appl. Soft Comput.* **2022**, *15*, 108231. [[CrossRef](#)]
36. Cai, H.S.; Feng, J.S.; Yang, Q.B.; Li, W.Z.; Li, X. A virtual metrology method with prediction uncertainty based on Gaussian process for chemical mechanical planarization. *Comput. Ind.* **2020**, *119*, 103228. [[CrossRef](#)]
37. Singh, J.; Darpe, A.K.; Singh, S.P. Bearing remaining useful life estimation using an adaptive data-driven model based on health state change point identification and K-means clusterin. *Meas. Sci. Technol.* **2020**, *31*, 085601. [[CrossRef](#)]
38. Deng, Y.F.; Du, S.C.; Jai, S.Y.; Zhao, C.; Xie, Z.Y. Prognostic study of ball screws by ensemble data-driven particle filter. *J. Manuf. Syst.* **2020**, *56*, 359–372. [[CrossRef](#)]
39. Huang, Y.-W.; Lai, D.-H. Hidden node optimization for extreme learning machine. *AASRI Procedia* **2012**, *3*, 375–380. [[CrossRef](#)]
40. Catelani, M.; Fort, A. Soft sensor detection and isolation in analog circuits: Some results and a comparison between a fuzzy approach and radial basis function networks. *IEEE Trans. Instrum. Meas.* **2002**, *51*, 196–202. [[CrossRef](#)]