A Distributed Probabilistic Model for Fault Diagnosis

Ana Li Oña García(✉), L. Enrique Sucar, and Eduardo F. Morales

Instituto Nacional de Asrofísica Óptica y Electrónica,
Sta. María Tonantzintla, Puebla, Mexico
{anali,esucar,emorales}@inaoep.mx

Abstract. Fault diagnosis in complex systems is important due to the impact it may have for reducing breakage costs or for avoiding production losses in industrial systems. Several approaches have been proposed for fault diagnosis, some of which are based on Bayesian Networks. Bayesian Networks are an adequate formalism for representing and reasoning under uncertainty conditions, however, they do not scale well for complex systems. For overcoming this limitation, researchers have proposed Multiply Sectioned Bayesian Networks. These are an extension of the Bayesian Networks for representing large domains, while ensuring the network inference in an efficient way. In this work we propose a distributed method for fault diagnosis in complex systems using Multiply Sectioned Bayesian Networks. The method was tested in the detection of multiple faults in combinational logic circuits showing comparable results with the literature in terms of accuracy, but with a significant reduction in the runtime.

Keywords: Fault diagnosis · Complex systems
Multiply Sectioned Bayesian Networks

1 Introduction

Fault detection is an important part of each engineering system, and it is often a prerequisite for commissioning, so systems must be robust to different types of faults which translates into high levels of reliability.

Research in this field has been focused on the application of different statistics techniques or Artificial Intelligence (AI), focusing efforts to timely detect abnormal behaviors from the analysis of the data of the sensed variables which translates into reduced repair costs.

A complex system is a system formed out of many components whose behavior is emergent, i.e., the behavior of the system cannot be simply inferred from the behavior of its components. A measure for assessing the complexity of such system is the amount of information necessary to describe its behavior [1]. Examples of complex systems include human economies, climate, nervous systems, and modern energy or telecommunication infrastructures.

G. R. Simari et al. (Eds.): IBERAMIA 2018, LNAI 11238, pp. 42–53, 2018.
https://doi.org/10.1007/978-3-030-03928-8_4

Several approaches have been proposed for fault detection, but not all take into account the uncertainty of real-world systems. In practice, model uncertainties and measurement noise can complicate fault detection, so the developed methods must be robust to these conditions. Bayesian Network (BN), represent an adequate formalism for the representation and reasoning under uncertainty conditions [19].

In [12], the authors proposed an algorithm for sensor validation by representing the relationships between the variables by using a Bayesian Network and the validation process is based on probabilistic propagation. The authors estimate the values of the variables, identifying the apparent fault from the expected value and the actual value of the analyzed variable. This work does not take into account, however, the complexity of the model domain and the distributed nature of its different components.

Multiply Sectioned Bayesian Networks (MSBNs), proposed by [26], are presented as an alternative for the modeling of large domain problems. MSBN identify domain partitions in smaller sub-domains that communicate with each other from shared information and where the inference process of the global network is performed efficiently.

In this work we focused on extending the work presented in [12] to a distributed approach, using the Multiply Sectioned Bayesian Network theory, for detection of multiple faults in complex systems. The proposal is tested by applying it to fault detection of combinational logic circuits.

The method was tested for 3 circuit examples, simulating the failed behavior of some of its components. The results show that, in terms of precision, our proposal has a behavior similar to the method proposed in [12], but it also results in a reduced runtime, and offers the possibility of modeling large domain problems for which the original method is intractable.

The remainder of this paper is organized as follows. First, in Sect. 2 the related work is presented. Section 3 describes the main concepts related to MSBN. In Sect. 4, we show the distributed method of fault diagnosis based on the proposal presented in [12]. In Sect. 5, we show our experimental study for fault diagnosis applied to combinational logic circuits. Finally, Sect. 6 presents some conclusions and future work.

2 Related Work

The methods of sensor validation (or more generally, the process of data validation) consist for two main stages: the detection of data faults and the correction of these failed data. The detection of defective data identifies dubious values or errors in the data, and the correction process provides methods to deal with problematic data [20]. In each category, different tools and methods exist in the scientific literature, and we will focus on the methods of fault detection as the main objective of our investigation.

In the state of the art related to fault detection, there are simple methods based on tests or physical or mathematical models, classifying the data in valid, invalid or missing [2,4]. These works, however, require knowledge of the observed phenomenon, what is not applicable to real problems, and where you do not have all the information of the study object.

Other studies in Artificial Intelligence have investigated the use of Artificial Neural Networks (ANNs) for sensor validation. Some of the network architectures use Multilayer Perceptrons for the estimation of some variables from other known [7,9,14,17].

In another work, [16], the authors use Multilayer Perceptrons to estimate variables, based on their values from previous epochs, for which they compare two approaches for the fault detection process. The first approach is based on the use of a set of Neural Networks for learning on-line, and the second approach is based on the use of Kalman filters [10]. The study reveals that neural online learning architectures have potential for estimation purposes in a sensor validation scheme.

Another technique proposed is Self-Organizing Maps. In these works the data are grouped into clusters of related data [3,22]. The detection of a fault is based on a measure of distance to the closest cluster.

Other works combine different Artificial Intelligence techniques. For example, in [18], the authors proposed a fault detection method using ANNs and Support Vector Machines (SVM) with Genetic Algorithms (GA).

All of these approaches that are based on the use of ANNs are limited by the need to learn the model from the complete information of the variables. It is necessary to use alternative techniques to predict the values of the variables in the presence of incomplete information or uncertain data.

Other works, such as those proposed in [8,11], introduce the use of Fuzzy Logic for the task of fault detection. The main limitation of using Fuzzy Logic is the need for expert knowledge to learn the membership functions.

Another technique used for the validation of sensor data, and which are robust to the uncertainty of the data, are probabilistic methods. The authors of [13] proposed sensor validation algorithms that combine different probabilistic methods, including Bayesian Networks. On the contrary, [21] proposed the validation of data using Sparse Bayesian Learning (SBL) and a Relevance Vector Machine (RVM), which is an SVM specialization.

Our research is an extension of the work presented in [12], for the validation of sensors in complex systems in the presence of uncertainty. The method consists of two fundamental stages: the apparent fault detection stage and the fault isolation stage.

In the fault detection stage, apparent faults are detected by comparing the current value with the predicted value through the propagation of beliefs, considering the rest of the variables as the evidence. This process is repeated for all variables, identifying those whose value differs from the value predicted as apparent faults.

In the fault isolation stage, from the apparent faults identified in the first stage and based on the property of the Markov Blanket (MB), an additional

Bayesian Network composed of two levels is created. The root nodes represent the real faults, and the nodes in the lower level represent the apparent faults.

The main advantage of this work is that it does not require fault data that can be difficult to obtain and is scarce, but it does not take into account the complexity of models in the inference process. In [6], the author demonstrated that the inference mechanism in the Bayesian Networks with multiple connections is a *NP-hard* problem, so that increasing the complexity (in terms of the connectivity of the network) of the problem modeled as a Bayesian Network increases the computational cost of probabilistic inference. That is why in the work presented in [12], the size of the domain of the problem is an important aspect, both in the probabilistic model obtained in the fault detection stage and in the Bayesian Network that is built in the fault isolation stage.

3 Multiply Sectioned Bayesian Networks

Multiply Sectioned Bayesian Networks (MSBNs) were proposed by [26] for the representation of large domain networks. An MSBN M is a set of Bayesian subnets that together defines a Bayesian Network (BN). M represents probabilistic dependence of a domain partitioned into sub-domains [25]. This technique constitutes an extension of the junction tree technique [15], where each node in the tree is formed by the clustering of a group of variables from the original network, and probabilistic inference is made on this new structure where each grouping acts as a unit for the passage of messages.

To ensure exact inference, MSBNs must satisfy the following tree conditions. The mathematical principles, as well as the definitions, are extracted from the work presented in [24].

(a) The subnets must satisfy a *hypertree* condition.

Definition 1. *Let $G = (V, E)$ be a connected graph sectioned into subgraphs $G_i = (V_i, E_i)$ such that the G_i's can be associated with a tree Ψ with the following property: Each node in Ψ is labeled by a G_i and each link between G_k and G_m is labeled by the interface $V_k \cap V_m$ such that for each i and j, $V_i \cap V_j$ is contained in each subgraph on the path between G_i and G_j in Ψ. Then Ψ is a hypertree over G. Each G_i is a hypernode and each interface is a hyperlink.*

(b) Variables shared between subnets must form a *d-sepset*.

Definition 2. *Let G be a directed graph such that a hypertree over G exists. Let x be a node that is contained in more than one subgraph and $\pi(x)$ be its parents in G. Then x is a d-sepnode if there exists one subgraph that contains $\pi(x)$. An interface I is a d-sepnode if every $x \in I$ is a d-sepnode.*

(c) The structure of an MSBN is a multiply sectioned DAG (MSDAG) with a hypertree organization.

Definition 3. *A hypertree MSDAG $G = \bigsqcup_i G_i$, where each G_i is a DAG, is a connected DAG such that (1) there exists a hypertree Ψ over G, and (2) each hyperlink in Ψ is a d-sepset.*

Under these conditions an MSBN is defined as:

Definition 4. *An MSBN M is a triplet (V, G, P). $V = \bigcup_i V_i$ is the domain where each V_i is a set of variables, called a subdomain. $G = \bigsqcup_i G_i$ (a hypertree MSDAG) is the structure where nodes of each DAG G_i are labeled by elements of V_i. Let x be a variable and $\pi(x)$ be all parents of x in G. For each x, exactly one of its occurrences (in a G_i containing $\{x\} \cup \pi(x)$) is assigned $P(x|\pi(x))$, and each occurrence in other DAGs is assigned a uniform potential. $P = \prod_i P_i$ is the joint probability distribution (jpd), where each P_i is the product of the potentials associated with nodes in G_i. A triplet $S_i = (V_i, G_i, P_i)$ is called a subnet of M. Two subnets S_i and S_j are said to be adjacent if G_i and G_j are adjacent.*

Figure 1 shows a trivial example of MSBN with three sections or subnets. The dashed nodes correspond to the variables shared between subnets.

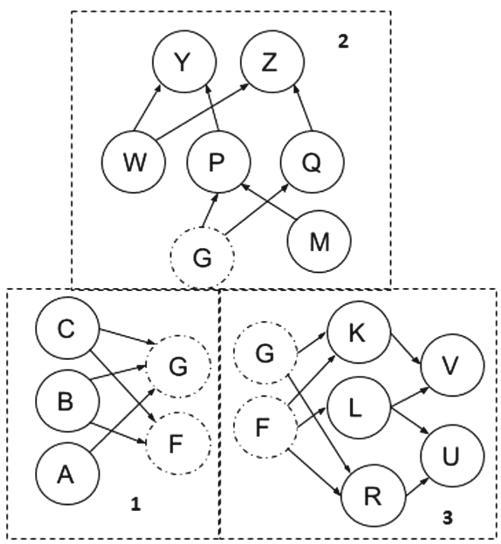

Fig. 1. A example of MSBN with three sections. Each section represents a sub-domain of the problem. The dashed nodes are the variables shared between adjacent sections.

The use of MSBN is limited to problem domains capable of breaking down into smaller subdomains. Many complex systems meet this condition so it is possible to apply this technique.

4 Distributed Fault Detection Method

In this section we briefly describe the proposed method from extending the work presented in [12] to a distributed approach with the use of the MSBN technique. To organize the work, the following subsections are divided, taking into account the two stages proposed by the authors of [12]: fault detection and fault isolation.

4.1 Fault Detection Stage

In the fault detection stage, a MSBN is constructed, representing the relationships between the variables of the domain to be validated, and partitioned into sections that satisfy the conditions described in the previous section (partitions are created from localization principles inherent to each problem domain). For the construction of the MSBN, we use the research tool WebWeavr-IV [23]. The four main steps for its construction are summarized below:

(a) Bayesian Networks construction for each section (individual agent level). For this step the necessary parameters include:
   - a set of variables;
   - a graph that represents the relations of independence between the variables; and
   - a set of conditional probability distributions.

(b) Knowledge representation at the agent society level. In this stage the structure for communication between adjacent sections is defined. For this task we define:
   - organization of sections or agents;
   - public variables between adjacent sections; and
   - hypertree condition check.

(c) Model verification. This stage includes obtaining the junction tree associated with each section, including:
   - global acyclicity test and
   - d-sepnode test.

(d) Compilation into Linked Cluster Trees. This structure is responsible for ensuring efficient communication between the adjacent sections through message passing.

For more details related to the construction and inference process of MSBNs see [26].

In general, the apparent fault detection algorithm consists of the following steps:

```
Obtaining the MSBN that represents the domain of the problem to validate
for new evidence do
    for each section do
        for each variable to be validated (usually all) do
            Propagate the probabilities to obtain the posterior probability distribution of the variable given the new evidence. The propagation process involves communication between adjacent sections
            Compare the predicted value (maximum posterior probability) with the current value of the variable and decide if there is an error
        end for
    end for
end for
```

The output of the algorithm consists of a list S of variables with apparent faults related to the section or sections to which they belong in the case of being shared variables.

4.2 Fault Isolation Stage

In the fault isolation stage, new Bayesian Networks are built relative to the sections of the model to be validated. These new Bayesian Networks consist of two levels. The nodes in the first level represent, for all variables, the events with real faults, and the nodes in the second level represent the apparent faults in all the variables. The relationship between the two levels corresponds with the Extended Markov Blanket (EMB) for each variable.

The EMB of one variable is defined as the parents, children, and other parents of their children, including the variable. In [12] the authors showed that it is enough to find matches between the apparent faults and the Extended Markov Blanket to isolate the real faults. Considering as new evidence the apparent faults identified in the previous stage, the probabilities associated with the real faults of each variable are updated.

For the case where apparent faults are identified in shared variables, the isolation network is formed from the union of the isolation networks corresponding to each section to which the shared variable belongs. Figure 2 represents the isolation network obtained for Sect. 1 of the example described in Fig. 1.

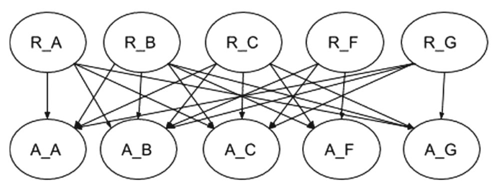

Fig. 2. Isolation network corresponding to Sect. 1 of the example of Fig. 1. The nodes (variables) at the upper level, R_i, correspond to the real faults, and the ones in the lower level, A_i, to the apparent faults.

5 Experiments and Results

In this section we tested the distributed algorithm of sensor validation for fault detection in combinational logic circuits. Initially, we will describe the main steps followed to obtain the models of MSBNs, then we will present the used evaluation metrics, and finally, we will describe the obtained results.

5.1 Integration of MSBNs in Combinational Logic Circuits

The case studies used to test our proposal correspond to combinational logic circuits formed by several components (subnets or sections) that communicate with each other. Each component is formed by a set of logical gates: AND, OR and NOT. Figure 3 depicts a simple example of a circuit partitioned in five components.

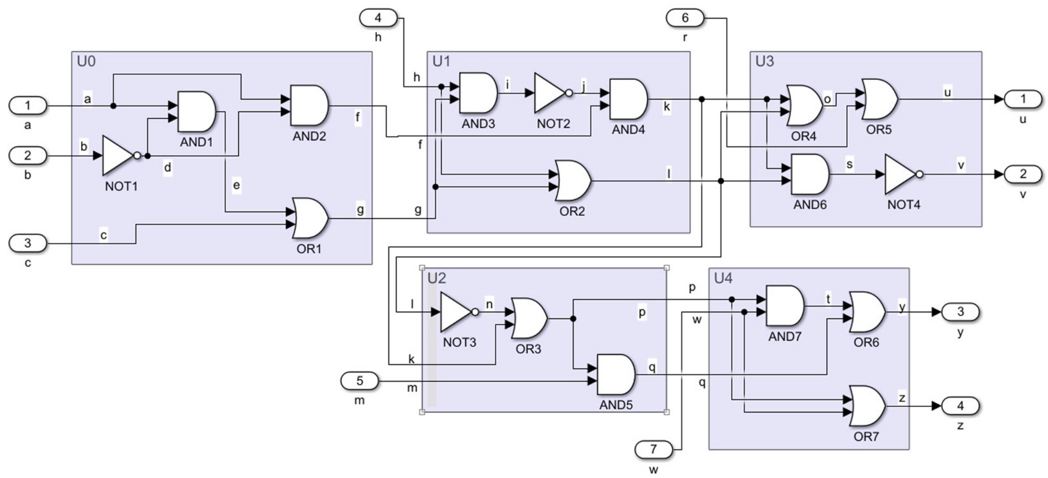

Fig. 3. Combinational logic circuit partitioned into 5 components.

For the work with the MSBNs we use the WEBWEAVR-IV toolkit [23], both for the creation of Bayesian networks at the local level, and for communication between the adjacent sections globally.

Each component is modeled as a Bayesian Network where each node represents the input and output variables of each logic gate within the circuit. For the parameters learning of the BN, the normal behavior of the circuit was simulated with the incomplete information of the variables represented by the BNs. Figure 4, shows a BN from component U0 in Fig. 3.

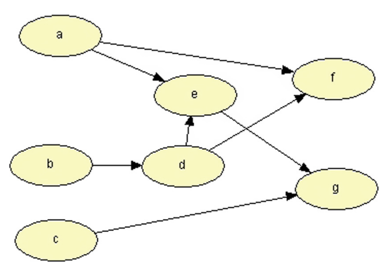

Fig. 4. Bayesian Network that represents the relationship between the variables of component U0 of Fig. 3.

After defining the sections at the local level, the communication structure is defined at the section level, establishing the variables shared between adjacent sections and verifying the hypertree condition. The model is also verified with the global aciclicity test and d-sepset tests. The last step in the construction of the model is the inference in all the MSBN that consists of two fundamental steps: the inference at global level using the junction tree technique and the inference to guarantee the global consistency from the construction of the Linked Cluster

Trees between adjacent sections for the passage of messages. All these functions are implemented in the WEBWEAVR-IV toolkit.

To model the unsuccessful behavior of the circuit, three types of failures were modeled: some component of the circuit stuck to 0, stuck to 1, or negate the output of a component. The components that fail were randomly selected, as well as the type of fault that occurs.

5.2 Evaluation Metrics

After performing the isolation stage, the output of our system will be the variables involved in the fault detection ordered by the probability of the occurrence of a real failure in each one of those variables. This can be seen as a problem of information retrieval where it is desirable that, in the first positions of this ordered list, the variables with real faults are found which would be the relevant variables.

To test the behavior of our research, we used two evaluation metrics proposed in the information retrieval work: P@5 and MAP [5]. P@5 is the precision at the 5-th position in the ranking of results. Mean Average Precision (MAP) is the Mean of the Average Precision scores for a group of queries, and average precision is the average of the precision scores at the rank locations of each relevant variable. This metric takes into account the order in which the variables are returned and is defined as:

$$MAP = AVG\left(\frac{\sum_{i=1}^{n} P(i) * rel(i)}{|relevant\_variables|}\right) \tag{1}$$

where n is the number of retrieved variables, $P(i)$ is the precision of the first i variables, and $rel(i)$ is a binary function indicating if variable at i-position is relevant or not.

5.3 Results

For the experiments we tested with 3 different examples of logic circuits. To each example, 20 cases of failures were made, 50% simple faults and 50% multiple faults (two and three simultaneous failures). Each test case corresponds to the abnormal behavior of one or more components of the circuit and consists of 100 instances.

Table 1 shows a summary of the test examples. The results are shown independently of the sectioning performed on each example.

To evaluate the effectiveness of our proposal we will compare it with the work presented in [12], which we will call baseline.

Table 2 shows the results of the P@5 and MAP for the three test cases. As shown in the results for example 1 and 2, the results are the same, which makes sense because the main difference of our proposal with the baseline is the representation and the way of making the inference, which translates into reduced runtime as the complexity of the problem to be modeled increases. For

Table 1. Summary of the logic gates and the variables of each test example.

Example	OR gates	AND gates	NOT gates	Variables
1	8	6	4	25
2	20	24	8	97
3	96	75	44	391

Table 2. Comparison of our proposal vs. [12] (baseline) in P@5 and MAP for the three combinational logic circuits of Table 1.

Example	1		2		3	
	P@5	MAP	P@5	MAP	P@5	MAP
Baseline	0.9417	0.5788	0.8917	0.6113	-	-
Our proposal	0.9417	0.5788	0.8917	0.6113	0.8802	0.5771

example 3, given the complexity of the problem, it is not possible to obtain a solution for the case of the work presented in [12].

Table 3 shows the results of the two metrics used for the detection of simple and multiple faults. The best results, in terms of P@5, are obtained for the detection of simple faults, where in most cases the variable with fault is returned within the first 5 positions of variables with the highest probability of having a real fault.

Table 3. Comparison between simple and multiple faults in terms of P@5 and MAP.

Example	1		2		3	
	P@5	MAP	P@5	MAP	P@5	MAP
Simple faults	1	0.575	0.9	0.5583	1	0.5409
Multiple faults	0.8833	0.5826	0.864	0.6643	0.7262	0.5931

Table 4 shows the comparison in terms of execution time of the work presented in [12] vs. our proposal. The times are indicated in minutes. This analysis includes the learning of the parameters, the fault detection stage and the fault isolation stage. The time reduction of the proposed algorithm is considerable, and as the complexity of the problem increases, the difference becomes even more evident. In Example 3, given the size of the problem, it is not possible to obtain a solution with the baseline method.

Table 4. Comparison between the average execution time of [12] (baseline) vs. our proposal. The times are indicated in minutes.

Example	1	2	3
Baseline	0.1173	23.717	-
Our proposal	0.0325	9.07	4368.9

6 Conclusions

We proposed a distributed extension of the work presented in [12], for the multiple faults detection in complex systems. For this, we use MSBNs, that is a technique for representing large domains that it is possible to partition into smaller sub-domains. The proposed method was tested for the detection of faults in combinational logic circuits. Based on the experiments, we can conclude that the proposed method maintains the effectiveness in terms of accuracy with respect to the original work while significantly reducing the execution time which makes it possible to deal with larger domain models.

As future work, we will apply this approach to other domains, in particular for diagnosis of wind turbines, which include discrete and continuous variables.

Acknowledgement. This work was sponsored by CEMIE-Eolico (CONACYT and SENER) and INAOE. The first author gratefully acknowledges CONACyT for her master scholarship 611489.

References

1. Bar-Yam, Y.: Dynamics of Complex Systems, vol. 213. Addison-Wesley, Reading (1997)
2. Bertrand-Krajewski, J.L., Winkler, S., Saracevic, E., Torres, A., Schaar, H.: Comparison of and uncertainties in raw sewage cod measurements by laboratory techniques and field UV-visible spectrometry. Water Sci. Technol. **56**(11), 17–25 (2007)
3. Böhme, T., Cox, C., Valentin, N., Denoeux, T.: Comparison of autoassociative neural networks and kohonen maps for signal failure detection and reconstruction. Intell. Eng. Syst. Through Artif. Neural Netw. **9**, 637–644 (1991)
4. Branisavljević, N., Kapelan, Z., Prodanović, D.: Improved real-time data anomaly detection using context classification. J. Hydroinformatics **13**(3), 307–323 (2011)
5. Chowdhury, G.G.: Introduction to Modern Information Retrieval. Facet Publishing, London (2010)
6. Cooper, G.F.: The computational complexity of probabilistic inference using Bayesian belief networks. Artif. Intell. **42**(2–3), 393–405 (1990)
7. Eryurek, E., Upadhyaya, B.: Sensor validation for power plants using adaptive backpropagation neural network. IEEE Trans. Nuclear Sci. **37**(2), 1040–1047 (1990)
8. Goebel, K., Agogino, A.: An architecture for fuzzy sensor validation and fusion for vehicle following in automated highways. In: Proceedings of the 29th International Symposium on Automotive Technology and Automation (1996)

9. Guo, T.H., Nurre, J.: Sensor failure detection and recovery by neural networks. In: Seattle International Joint Conference on Neural Networks, IJCNN 1991, vol. 1, pp. 221–226. IEEE (1991)
10. Haykin, S.S., et al.: Kalman Filtering and Neural Networks. Wiley, Hoboken (2001)
11. Holbert, K.E., Heger, A.S., Alang-Rashid, N.K.: Redundant sensor validation by using fuzzy logic. Nuclear Sci. Eng. **118**(1), 54–64 (1994)
12. Ibargüengoytia, P.H., Vadera, S., Sucar, L.E.: A probabilistic model for information and sensor validation. Comput. J. **49**(1), 113–126 (2005)
13. Ibarguengoytia, P., et al.: Any time probabilistic sensor validation. Ph.D. thesis, University of Salford, UK (1997)
14. Khadem, M., Alexandro, F., Colley, R.: Sensor validation in power plants using neural networks. In: Neural Network Computing for the Electric Power Industry, pp. 51–54 (1993)
15. Lauritzen, S.L., Spiegelhalter, D.J.: Local computations with probabilities on graphical structures and their application to expert systems. J. Roy. Stat. Soc. Ser. B (Methodol.) **50**, 157–224 (1988)
16. Napolitano, M.R., Windon, D.A., Casanova, J.L., Innocenti, M., Silvestri, G.: Kalman filters and neural-network schemes for sensor validation in flight control systems. IEEE Trans. Control Syst. Technol. **6**(5), 596–611 (1998)
17. Rajakarunakaran, S., Venkumar, P., Devaraj, D., Rao, K.S.P.: Artificial neural network approach for fault detection in rotary system. Appl. Soft Comput. **8**(1), 740–748 (2008)
18. Samanta, B.: Gear fault detection using artificial neural networks and support vector machines with genetic algorithms. Mech. Syst. Sig. Process. **18**(3), 625–644 (2004)
19. Sucar, L.E.: Probabilistic Graphical Models - Principles and Applications. Advances in Computer Vision and Pattern Recognition. Springer, Heidelberg (2015). https://doi.org/10.1007/978-1-4471-6699-3
20. Sun, S., et al.: Literature review for data validation methods. Sci. Technol. **47**(2), 95–102 (2011)
21. Tipping, M.E.: Sparse Bayesian learning and the relevance vector machine. J. Mach. Learn. Res. **1**(Jun), 211–244 (2001)
22. Valentin, N., et al.: A neural network-based software sensor for coagulation control in a water treatment plant. Intell. Data Anal. **5**(1), 23–39 (2001)
23. Xiang, Y.: Webweavr-iv research toolkit (2006)
24. Xiang, Y.: Comparison of multiagent inference methods in multiply sectioned Bayesian networks. Int. J. Approx. Reason. **33**(3), 235–254 (2003)
25. Xiang, Y., Jensen, F.V., Chen, X.: Inference in multiply sectioned Bayesian networks: methods and performance comparison. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **36**(3), 546–558 (2005)
26. Xiang, Y., Poole, D., Beddoes, M.P.: Multiply sectioned Bayesian networks and junction forests for large knowledge-based systems. Comput. Intell. **9**(2), 171–220 (1993)