Bayesian Structural Learning for an Improved Diagnosis of Cyber-Physical Systems

Nicolas Olivain[1,2], Philipp Tiefenbacher[2], and Jens Kohl[2]

[1] Mines Paristech - Paris Sciences & Lettres, Paris, France,
nicolas.olivain@mines-paristech.fr
[2] BMW Group, Development Powertrain, Munich, Germany,
{nicolas.olivain, philipp.tiefenbacher, jens.kohl}@bmw.com

Abstract. The diagnosis of cyber-physical systems (CPS) is based on a representation of functional and faulty behaviour which is combined with system observations taken at runtime to detect faulty behaviour and reason for its root cause. In this paper we propose a scalable algorithm for an automated learning of a structured diagnosis model which -although having a reduced size- offers equal performance to comparable algorithms while giving better interpretability. This allows tackling challenges of diagnosing CPS: automatically learning a diagnosis model even with hugely imbalanced data, reducing the state-explosion problem when searching for a failure's root cause, and an easy interpretability of the results. Our approach differs from existing methods in two aspects: firstly, we aim to learn a holistic global representation which is then transformed to a smaller, label-specific representation. Secondly, we focus on providing a highly interpretable model for an easy verification of the model and to facilitate repairs. We evaluated our approach on data sets relevant for our problem domain. The evaluation shows that the algorithm overcomes the mentioned problems while returning a comparable performance.

Keywords: Bayesian networks, Cyber-physical systems, Diagnosis, Genetic Algorithms, Root Cause Analysis, Structural learning.

1 Introduction

1.1 Diagnosis of cyber-physical systems

In cyber-physical systems (CPS, defined in e.g. [40]) mechanical, electrical and electronic components are combined with and controlled by software components to execute defined tasks. They are widely used in consumer electronics, Internet-of-Things, plants, airplanes or automotive vehicles. Since the tasks assigned to a CPS can be safety-relevant and harmful to its environment, possible faulty behaviour has to be detected, the fault's root cause has to be inferred and failures have to be prevented or mitigated (in this paper we use the terminology as introduced by [2]). Furthermore, details about the fault's root cause are necessary for a system remedy via repair. These are the central tasks of diagnosis.

The diagnosis of CPS consists of several challenges regarding *building or learning the diagnosis' model* given the CPS' *interconnections and dependencies* and *imbalanced data* from different *data sources*, running the diagnosis model under *hard real-time constraints* delivering *fast inference of the root cause(s)* for failure mitigation or prevention and a *focused repair*.

Building the diagnosis model for a CPS is a complex and especially time-consuming process due to lots of functional dependencies and interconnections between the components of a CPS as well as with other interacting CPS. Additionally, CPS components can have fault dependencies while having no functional dependencies (e.g. blocking a shared communication device thus causing a time out). Thus, a lot of these dependencies are 'learned' by trial-and-error.

Since building a diagnosis model manually requires huge efforts, using supervised learning algorithms for *learning a diagnosis model* seems promising. However, using such approaches is aggravated since the available data is vastly imbalanced. Given that most CPS operate according to their specification, supervised learning approaches tend to regard the few faulty CPS as rather noise or outliers. Additionally, the available observations of a faulty CPS are imbalanced as well. While CPS typically sample hundreds or more of different observations of their environment and internals with rates of 1–1000 Hertz or more, their purely mechanical parts cannot be sampled due limited insight.

Hence, to increase the amount of available data for mechanical parts other observations along their whole life-cycle have to be collected, such as the production data of their individual parts, their assembly into a CPS and -if available- customer feedback. Since these data points belong to different *data sources*, they need to be encoded as observations.

For safety-relevant domains, the diagnosis of CPS has to adhere to *(hard) real-time constraints* meaning faulty behaviour has to be detected (almost) instantly followed by searching for diagnosis candidates fitting the model and the observations. For complex CPS this forms an exponential state space aggravating the need for a *fast inference of the root cause*.

Since most CPS provide only limited direct access or insight, a *focused repair* to remedy the CPS depends on concise diagnosis results. Concise diagnosis results however differ: while domain experts need extensive information about the root cause - especially during development of a system - service people or end-users prefer understandable results to support a focused repair.

Finally, the CPS' diagnosis model needs to be maintained over the whole product lifecycle which typically spans several years.

1.2 Overview of our contribution

The contribution of this paper is a scalable methodology able to cope with the detailed challenges. We use a Bayesian Network to automatically learn a feature ensemble explaining the functional and fault relationships and dependencies of the CPS. This learned representation is optimized by using a genetic algorithm, thus returning a reduced diagnosis model with equal accuracy as other comparable algorithms while having a better interpretability. A huge benefit of our

approach is that it works with imbalanced data -as quite common in our domain- by using information-based criteria as an evaluation metric both when building the network and in the genetic algorithm's fitness function.

The increased interpretability is especially helpful for domain experts in the development phases and for non-domain experts during repairs. Furthermore, the Markov-separated model is easily maintainable by domain experts. They can integrate further knowledge from different data sources as Boolean formulas into the model and transfer separated parts of the model to similar CPS or domains without many changes.

Finally, having a reduced model helps avoiding the state explosion problem for the inference. This allows us to deploy the model on CPS with limited computing resources.

1.3 Outline

In this section we described our problem domain, detailed its main challenges and outline our contribution. In the next section, Sec. 2, we discuss related work. Sec. 3 details our contribution and its benefits to our problem domain. In Sec. 4 we show an evaluation of our methodology on two different data sets, a medical dataset and one for an automotive component. Finally, Sec. 5 concludes this paper and shows possible future work.

2 Discussion of related Work

Several model-based methodologies have been used for a representation of a CPS' functional behaviour [9,11,16,29]. However, these approaches cover software and electric/ electronic parts of CPS, but not mechanical parts and behaviour. [17] introduces one of the first approaches to enable a holistic view of a CPS.

Regarding representing a CPS' faulty behaviour and its diagnosis, two different approaches stand out. First of all, *expert-based diagnosis systems*, as first defined in [10], use rules to encode the system's (faulty) behaviour. While they are easy to build and to understand, they rely on expert domain knowledge and are time-consuming to build and difficult to update and maintain. *Model-based diagnosis* has been used for CPS for a long time. In contrast to expert-based systems, the diagnosis model is based on a formal specification of the system's behaviour. This model is combined with observations of the system's behaviour at runtime to discover deviations from the specified behaviour. Both model and observations can be encoded with boolean/ first-principle logic [14, 41], as a discrete event system [43] or with differential equations [26, 27]. A drawback of model-based systems is that the determination and isolation of a root cause leads to an exponential state space. [3, 4] showed how the diagnosis' inference can be transformed into a satisfiability problem and then efficiently be solved via a Satisfiability (SAT-)solver (e.g. [36]), [31] applied this to the automotive domain, a prime domain for CPS. Even though model-based systems are widely

used, their models are difficult to build and understand even for domain experts, and have a rather unmanageable complexity.

Approaches from *representation Learning* are on the rise for explaining system behaviour. Restricted Boltzmann Machines (RBM) [22, 33] are widely used for learning feature representations with newest works focusing on the explainability of such RBM. [1] proposed a RBM consisting of additional explainability units in the visible layer for recommender systems. They defined a joint distribution over visible and hidden units and a conditional distribution on explainability scores in what they called a conditional RBM. In the diagnosis field, [46] extracted features for motor fault diagnosis using stacked RBM [48], i.e. Deep Belief Networks. [35] enhances RBM for failure diagnosis with an additional regularisation term to maintain features relevant for the health of a system. In these works, however, RBM are only used for identifying the most relevant features; explainability and structure within the feature space are not addressed.

A *Bayesian network* is a directed acyclic graph representing causal relationship between variables using the Bayes rule [5]. Bayesian networks were first defined by [30, 37] and used for inference in tree networks. [38] extended the approach by including continuous random variables, [21] combined statistical data and encoded knowledge. Bayesian Networks have been used in industry domains since the 1980s for diagnosing systems [7, 20, 30]. The advantages of Bayesian networks are that built models and the connections of its elements can be visualised (e.g. graphs) and thus are easy to understand. Additionally, the approach offers to incorporate uncertainty and compare different solutions. However, Bayesian Networks face some challenges. They have high computational costs since when updating a variable, its whole branch including all its dependent variables have to be updated as well. This update can extend up to the whole net, if the network has no separation. Hence, a careful construction of the network and its dependent variables to limit its complexity is essential. This can be done by domain experts in a time-consuming process or algorithmically. [44] showed how to apply the learning problem of Bayesian networks to thousands of variables without prior expert knowledge.

A common approach for algorithmic optimization are *Genetic algorithms* (GA). They are inspired by natural evolution. Each individual of a given population is ranked by a defined fitness-function selecting only the best ones for further reproduction in a subsequent generation until convergence towards a defined optimum. Genetic Algorithms were principally defined in [18] and popularised by [24, 25]. GA have been used for a long time for structural learning of Bayesian networks [13, 32] or other supervised learning algorithms [15, 28, 34]. Here custom fitness-functions are used to rate and optimise a learned topology until reaching an optimal graph, i. e. one maximising a defined fitness function.

3 Contribution

In this section we detail our two-fold algorithm: the first part builds a Bayesian Network to learn a representation of the CPS and its causal relationships. In the

second part, we use a genetic algorithm which explores the network to select the best nodes and vertices for explaining a failure.

3.1 Data Preprocessing and preparation

First, we convert all non-categorical features of the dataset into discrete ordinal features as we only consider discrete probability distributions for the Bayesian Network. Hence, we use a simple binning based on quantiles with continuous values mapped to the index of their corresponding quantile with the quantiles amount a definable hyper-parameter. Non-numerical values are encoded with Boolean formulas and ordinal numbers.

3.2 Structural learning

We build a Bayesian Network to learn causal relationships between variables before selection. Instead of having each variable depending on all others, we reduce the set of the variables that can influence a given variable to a parent subset. Learning the optimal Bayesian Network for a given dataset is NP-hard due to state explosion. Thus we need to estimate the structure using a method scalable to consequent datasets.

We use the BIC^* [44] as an approximation for the Bayesian Information Criteria (BIC, [45]) to measure the relevance of a set of candidate features S being the parent set of a given variable X_i. These criteria serve as metrics to evaluate the likelihood of a set of vertices. We can compute a list of candidate parent sets for each feature X_i from our data set. Given that we already have BIC values for S_1 and S_2, we can use already computed scores to approximate the BIC^* value for a candidate $S = S_1 \cup S_2$. Thus, BIC^* is used for evaluating the large number of possible candidates.

To generate the lists of candidates for each node, we use the independence selection algorithm of [44]. The parent set is stored in two lists: one consisting of all parents which have been explored and a second list for the unexplored parent set. Our algorithm returns a list of explored potential parents for a node X_i with their score.

The algorithm can be parallelised allowing an evaluation of several different candidates simultaneously. Once given the list of parent set candidates for all the variables of the dataset, i.e. all the network's nodes, we can build the network.

In the next step, parents for each node are picked given the constraints of a directed acyclic graph (DAG). Picking parents for a node sequentially introduces an anchoring bias, i.e. while the first nodes can select from all parents, later nodes are limited in their selection. To avoid this, we propose a new selection policy taking into account the order of the nodes in relation to the investigated label nodes, i.e. faults. This policy is based on the idea that the root cause of faults, i.e. parents close to the label, are the main goal of creating our graph. In contrast, [44] maximizes the graph according to BIC while we are creating the graph around the label nodes focusing on the best explanation of a given label node.

Nodes that are connected directly to the label pick first by order of relation. For instance, a second order linked variable will be picked only after all the first order variables have picked their parent sets. Variables that are separated from the label pick their parent set in a random order afterwards.

Algorithm 1 Parent Set Selection

Input: Parent set candidates for each node and specific *label*
Output: Parent sets for each node
Initialize *parents* as empty
Initialize *open* with *label*
while *open* is not empty **do**
  Pop the first element of *open* as X
  while Candidate list for X is not empty **do**
    Pop the best candidate C for X from candidate list
    if X is not a descendant of C **then**
      Accept C as X parent set, store it in *parents*
      Add C nodes not having a parent set yet to *open*
      Break
    end if
  end while
  if no candidate was selected **then**
    set X parents as $\emptyset$
  end if
end while
Apply the same procedure for the remaining nodes, regardless of their order
return *parents*

Alg. 1 details the learning procedure of a DAG with each node X having an associated parent set so that the graph remains acyclic. We then can use this general architecture to extract a smaller topology that determines potential root causes of our label nodes.

3.3 Root Cause Analysis

Given the learned graph G, our goal is now to select a feature ensemble E maximizing our ability to explain a specific label. Ideally, given all the values of the variable in the ensemble, we want to be able to determine the label class.

We denote the usual entropy of a discrete random variable X as $H(X)$, its conditional variant for discrete random variables X and Y is defined as $H(X|Y)$. The discrete probabilities derived here can be estimated empirically using the maximum likelihood estimator.

We also define the mutual information between two discrete random variable X and Y as $I(X,Y) = H(X) - H(X|Y)$ representing the amount of information shared between X and Y using properties from $H(X|Y)$. This conditional entropy is equal to $H(X)$ if X and Y are independent. It is equal to 0 if the

knowledge of Y gives us the whole knowledge of X, meaning that X can be derived in a deterministic manner using the value of Y. This metric is often used as a distance in clustering problems and thus normalized in a symmetrical fashion [49].

In our case, since we are interested asymetrical parent/child relationships, we apply the *Uncertainty Coefficient* defined as $U(X,Y) = \frac{I(X,Y)}{H(X)}$ [39]. This metric represents how much of X can be predicted given Y, with 0 in case of independent X and Y and 1 for a deterministic relationship. We can now generalize this definition for $U(X,E)$ to compare how good the different sets E can explain our label node.

In a Bayesian Network for a given node, anything outside its Markov blanket is independent and hence not relevant [38]. However, since our graph is generated with neither expert nor previous knowledge, we assume that some residual information can still to be grasped within higher order dependencies.

From the definition of $U(X,E)$, we can see that this value increases for larger sets E with more states. Hence, the defined genetic algorithm would tend to increase the amount of selected features regardless of their importance potentially leading to an overfitting. To address this, we use a $L2$-regularization term on the number of states from E. This term introduces two hyper-parameters depending on the dataset, the characteristic state number τ denoting the expected number of states and the intensity of the regularization C (typically C 10^{-3}) for balancing the regularisation term. We then define the fitness function F as:

$$F(X_0,E) = U(X_0,E) - C\max(0,(\frac{|E|-\tau}{\tau})^2).$$

Thus we search the subset E maximising $F(X_0,E)$ by exploring available graph topologies, starting from the label node and going up its parents. We use a genetic algorithm to explore the possible topologies and retain the best ones. This method is easily scalable to wider graphs, as the number of possibilities scales exponentially, and assesses gains from one generation to another.

This algorithm allows us to test a wide range of different extracted graphs depending on hyper-parameter tuning. It can be adapted using different breeding methods such as exhaustive search (given a narrow network or high computational power) or random mutation.

Hence, we aim for first learning relationships of the whole CPS and then select the relevant ones for a given incident. The fitness function F in Alg. 2 measures 'separatibility' between classes given the parents. When a feature allows to separate a significant amount of positive samples from the other overwhelming majority of negative ones, the U term for the fitness will increase greatly. This approach does not rely on a balance of samples thus making it suitable for identifying potential root causes. It is also worth mentioning that U is the non-symmetrized version of symmetrical uncertainty which has been shown as relevant for high-dimensional feature selection [42]. Our approach first learns a global representation of the data, meaning that all vertices will be considered and weighted regardless of their affiliation with the label. When the reduced

Algorithm 2 Root Cause Extraction

Input: DAG G, *label*, *max_gen*, K, *patience*, *plateau*
Output: A reduced DAG from *label*'s parents
  Initialize *generation* as empty
  Initialize *stagnation* and *gen_number* with 1
  for all combination E of *label*'s parents in G **do**
    Add $F(label, E)$ to *generation*
  end for
  Select the K fittest individuals as *best*
  while *stagnation* $<$ *patience* and *gen_number* $<$ *max_gen* **do**
    Set *generation* as empty
    Add *best* individuals from previous generation to *generation*
    Breed *best* individuals and add them to *generation*
    Select new K fittest individuals from *generation*
    if difference between top individuals $<$ *plateau* **then**
      Increment *stagnation*
    end if
    Increment *gen_number*
  end while
  return *generation* fittest individual

model is extracted, this ensures that its edges are relevant according to the BIC-criteria as they were picked by a global model in the first place against many others. Thus, we ensure the relevance of our final vertices.

4 Evaluation

We evaluated our approach on two use cases. The first use case is the diagnosis of an automotive component, the second use case is medical diagnosis of humans. We chose medical diagnosis since our main challenges imbalanced data and interpretability also hold for the medical domain. For both use cases we have imbalanced data with hundreds or thousands of features of which only a small amount indicates a given defect.

The main target for our evaluation was receiving a comparable accuracy and an improved interpretability of the diagnosis' results. We mentioned that many feature-based selection techniques instead of identyfing these features could rather classify them as noise and additionally lack interpretability for domain experts.

The evaluation shows that the algorithm overcomes the mentioned problems while returning a comparable accuracy.

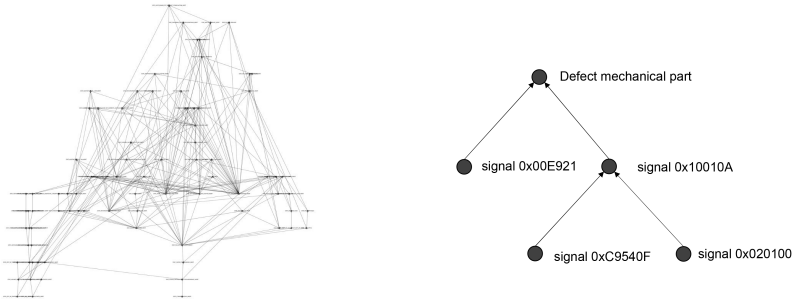
4.1 Automotive Use Case

Our first use case is the root cause analysis of a failure of a mechanical component of the powertrain occuring during its development. In Sec. 1.1 we mentioned

that measurement data for mechanical parts is scarce. Hence, we need to link all available data for an analysis. This leads to complex failure patterns which motivated our approach of a more explainable representation. The exact data for this use case is not publishable due confidentiality.

We built two classes, one with vehicles having the defect (the label) and the other with similar vehicles not having this defect. As mentioned, the data is heavily imbalanced, since specific defects only came up in very few vehicles.

From the initial 3000+ available features, we built a Bayesian Network numbering around 1800 nodes. Using the genetic algorithm, we reduced it down to 4 features for a specific failure, which we then verified with domain experts. To ease understanding, Fig. 1 shows only an excerpt of the learnt Bayesian Network and short type information for the features in the reduced network.



(a) Excerpt of the learnt Bayesian Network (68 nodes, 99 vertices)

(b) Reduced Bayesian Network (5 nodes, 4 vertices)

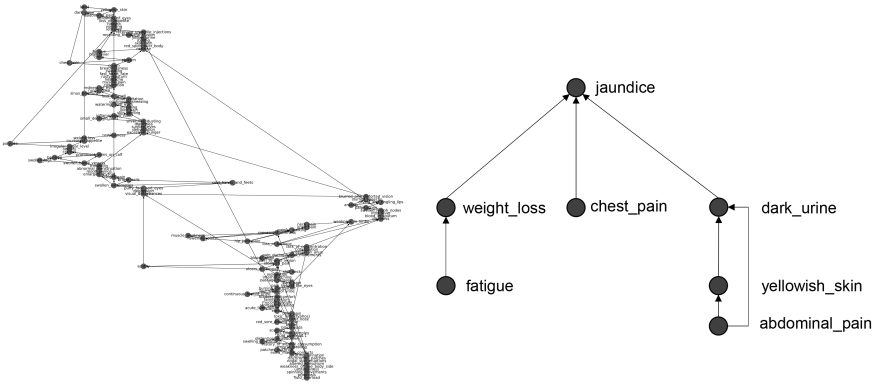
Fig. 1: Learnt representation for the automotive use case

4.2 Medical Diagnosis Use Case

As basis for our second use case we chose a medical data set on Kaggle [47]. The original dataset contains 41 different pathologies as target classes linked to 132 different symptoms. We generated 500 patients for each pathology. Our test dataset uses a probabilistic approach with each symptom having an appearance probability depending on the type of pathology simulated.

From this data set we chose the *Jaundice* disease as label and generated a dataset of 20'500 patients with diverse symptoms. This dataset is heavily imbalanced (less than 2% positive samples) and contains 132 different symptoms or features. Figure 2 shows the learnt models. This representation contains 7 out of 132 features. More importantly, all 6 features are among *Jaundice* symptoms.

We now compare our approach to the common classifiers such as Decision Tree [8], Random Forest [23] and Gradient Tree Boosting [12]. Additionally, we



(a) Fully learnt Bayesian Network (133 nodes, 192 vertices)

(b) Reduced Bayesian Network (7 nodes, 7 vertices)

Fig. 2: Learnt representations for the medical data

fed the nodes of the reduced Baysian Network in a constrained decision tree with a maximum depth of 5. By using the constrained decision tree with its depth limit we can check on the accuracy whether we chose the right features.

Table 1 shows the classifiers' results for the positive class, its precision, recall and F1-score. Precision represents the amount of relevant positive items among the predicted positive items. Recall, or sensitivity, on the other hand, depicts how many among the relevant positive items are predicted positive by our model. F1-score mixes these two indicators for an overall performance estimation and is typically used to select the prefered classification method. For diagnosis use cases, precision or specifity is very interesting, although in our use cases the imbalanced ratio makes it hard to discriminate on the latter.

Model	Precision	Sensitivity	Specificity	F1-score
Decision Tree	0.67	0.69	0.99	0.68
Random Forest	0.78	0.68	0.99	0.73
XGBoost	0.80	0.77	0.99	0.78
Red. Bayesian Network	0.80	0.65	0.99	0.72
Constrained Decision Tree	0.80	0.77	0.99	0.78

Table 1: Comparison of Results for positive class

Regarding precision, all of the algorithms performed quite similar except decision tree which had a significantly lower value. The reduced Bayesian network seems comparable to a random forest in terms of performance, but uses a lot less

features and is easier to interprete. The benefit of the Bayesian network and its derived constrained decision tree is the improved interpretability compared to Random Forest and XGBoost. While these models use 100 different estimators of various depth, the reduced graph of the Bayesian network in contrast consists of 6 (of which only 3 are actually used for Jaundice prediction due to Markov separation) out of the initial 132 features. Furthermore, we can derive a confidence metric for the network by infering probabilities of the label given specific symptoms. Given a positive label and/ or other observed symptoms, we can find the most likely value for the missing symptoms.

Overall, we showed how a combination of a reduced Bayes model and a constrained decision tree gives valuable and reliable insights for root cause analysis, even without prior knowledge of the system.

5 Conclusion and Future Work

We presented a scalable algorithm capable of learning a structured, optimised diagnosis model for cyber-physical systems with comparable accuracy to standard approaches but higher interpretability.

With its optimised structure and its separated structure, the learnt diagnosis model can be easily understood and provides concise, consistent and accurate diagnosis results.

Additionally, the learned model can be easily integrated into the workflow of diagnosis engineers providing benefits such as an improved model maintenance or by easily integrating further knowledge to the model in form of boolean formulas. Moreover, they can easily add new defects to the existing model by computing the defect's parents using the described methodologies and then search for the defect's root cause(s). This allows for continuous, iterative developement, incremental model upgrades and thus an improved maintainability.

Given these points, we are convinced that our approach helps domain experts and outweighs the inherent mentioned disadvantages of Bayesian networks. Additionally, we can use the learned, reduced model for other methodologies such as supervised learning algorithms or 'traditional' model-based diagnosis approaches.

Regarding future work, we are planning to extend our approach for diagnosing multiple defects respectively a multi-label classification. Multiple defects are defects occuring at the same time (multiple defects happening sequentially are covered by our approach as our algorithm would then learn the prior defects as indicator of the latter).

Finally, since we can annotate the model with repair measures, their costs and repair time as well as chance of success, we can create a real-options model [6] for optimising repair measures [19]: for example, changing a specific part of a CPS can be cheaper than changing the whole CPS but requires more time and has a different probability of success.

References

1. Abdollahi, B., Nasraoui, O.: Explainable Restricted Boltzmann Machines for Collaborative Filtering (2016)
2. Avizienis, A., Laprie, J.C., Randell, B., Landwehr, C.: Basic concepts and taxonomy of dependable and secure computing. IEEE transactions on dependable and secure computing **1**(1), 11–33 (2004)
3. Bauer, A.: Simplifying diagnosis using LSAT: a propositional approach to reasoning from first principles. In: International Conference on Integration of Artificial Intelligence and Operations Research Techniques in Constraint Programming. pp. 49–63. Springer (2005)
4. Bauer, A., Leucker, M., Schallhart, C.: Model-based runtime analysis of distributed reactive systems. In: Proceedings of the Australian Software Engineering Conference. pp. 243–252 (2006)
5. Bayes, T.: An essay toward solving a problem in the doctrine of chances. Philosophical Transactions of the Royal Society of London **53**, 370–418 (1763)
6. Black, F., Scholes, M.: The pricing of options and corporate liabilities. Journal of political economy **81**(3), 637–654 (1973)
7. Breese, J., Heckerman, D.: Decision-theoretic troubleshooting: a framework for repair and experiment. In: Proceedings of the 12th international conference on Uncertainty in artificial intelligence. pp. 124–132 (1996)
8. Breiman, L., Friedman, J., Stone, C., Olshen, R.: Classification and regression trees. Wadsworth (1984)
9. Broy, M., Stølen, K.: Specification and development of interactive systems: Focus on streams, interfaces, and refinement. Springer Science & Business Media (2012)
10. Buchanan, B., Shortliffe, E. (eds.): Rule-Based Expert Systems - The MYCIN Experiments of the Stanford Heuristic Programming Project, Addison-Wesley Series in Artificial Intelligence, vol. 1. Addison-Wesley (1984)
11. Buck, J., Ha, S., Lee, E., Messerschmitt, D.: Ptolemy: A framework for simulating and prototyping heterogeneous systems. International Journal of Computer Simulation: Simulation Software Development **4**, 155–182 (1994)
12. Chen, T., Guestrin, C.: XGBboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 785–794 (2016)
13. Contaldi, C., Vafaee, F., Nelson, P.: Bayesian network hybrid learning using an elite-guided genetic algorithm. Artificial Intelligence Review **52**(1), 245–272 (2019)
14. De Kleer, J., Williams, B.: Diagnosing multiple faults. Artificial intelligence **32**(1), 97–130 (1987)
15. Demiriz, A., Bennett, K., Embrechts, M.: Semi-supervised clustering using genetic algorithms. Artificial neural networks in engineering pp. 809–814 (1999)
16. Derler, P., Lee, E., Vincentelli, A.: Modeling cyber–physical systems. Proceedings of the IEEE **100**(1), 13–28 (2011)
17. Drave, I., Rumpe, B., Wortmann, A., Berroth, J., Hoepfner, G., Jacobs, G., Spuetz, K., Zerwas, T., Guist, C., Kohl, J.: Modeling mechanical functional architectures in SysML. In: Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems. pp. 79–89 (2020)
18. Fraser, A., Burnell, D.: Computer models in Genetics. Computer models in genetics (1970)
19. Haddad, G., Sandborn, P., Pecht, M.: Using real options to manage condition-based maintenance enabled by prognostics and health management. In: 2011 IEEE Conference on Prognostics and Health Management. pp. 1–7 (2011)

20. Heckerman, D.: Probabilistic similarity networks. Networks **20**(5), 607–636 (1990)
21. Heckerman, D., Geiger, D., Chickering, D.: Learning Bayesian networks: The combination of knowledge and statistical data. Machine learning **20**(3), 197–243 (1995)
22. Hinton, G., Sejnowski, T.: Learning and relearning in Boltzmann machines. In: Parallel Distributed Processing, vol. 1, pp. 282–317. MIT Press (1986)
23. Ho, T.K.: Random decision forests. In: Proceedings of 3rd international conference on document analysis and recognition. vol. 1, pp. 278–282. IEEE (1995)
24. Holland, J.: Adaptation in natural and artificial systems. University of Michigan Press (1975)
25. Holland, J.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. MIT Press (1992)
26. Isermann, R.: Process fault detection based on modeling and estimation methods — a survey. automatica **20**(4), 387–404 (1984)
27. Isermann, R.: Model-based fault-detection and diagnosis–status and applications. Annual Reviews in control **29**(1), 71–85 (2005)
28. Janikow, C.: A knowledge-intensive genetic algorithm for supervised learning. In: Genetic Algorithms for Machine Learning, pp. 33–72. Springer (1993)
29. Jensen, J., Chang, D., Lee, E.: A Model-Based Design Methodology for Cyber-Physical Systems. In: 7th International Wireless Communications and Mobile Computing Conference (IWCMC). pp. 1666 – 1671 (July 2011)
30. Kim, J., Pearl, J.: A computational model for causal and diagnostic reasoning in inference systems. In: Proceedings of the 8th international joint conference on Artificial intelligence. vol. 1, pp. 190–193 (1983)
31. Kohl, J., Bauer, A.: Role-Based Diagnosis for Distributed Vehicle Functions. In: 21st International Workshop on the Principles of Diagnosis (DX) (2010)
32. Larranaga, P., Poza, M., Yurramendi, Y., Murga, R., Kuijpers, C.: Structure learning of Bayesian networks by genetic algorithms: A performance analysis of control parameters. IEEE Transactions on pattern analysis and machine intelligence **18**(9), 912–926 (1996)
33. Le Roux, N., Bengio, Y.: Representational power of restricted Boltzmann machines and deep belief networks. Neural computation **20**(6), 1631–1649 (2008)
34. Leung, F., Lam, H.K., Ling, S.H., Tam, P.: Tuning of the structure and parameters of a neural network using an improved genetic algorithm. IEEE Transactions on Neural networks **14**(1), 79–88 (2003)
35. Liao, L., Jin, W., Pavel, R.: Enhanced restricted Boltzmann machine with prognosability regularization for prognostics and health assessment. IEEE Transactions on Industrial Electronics **63**(11), 7076–7083 (2016)
36. Moskewicz, M., Madigan, C., Zhao, Y., Zhang, L., Malik, S.: Chaff: Engineering an efficient SAT solver. In: Proceedings of the 38th annual Design Automation Conference. pp. 530–535 (2001)
37. Pearl, J.: Reverend Bayes on inference engines: a distributed hierarchical approach. In: Proceedings of the 2nd AAAI Conference on Artificial Intelligence. pp. 133–136. AAAI Press (1982)
38. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann (1988)
39. Press, W., Flannery, B., Teukolsky, S., Vetterling, W.: Numerical Recipes: the Art of Scientific Computing. Cambridge University Press, 3rd edn. (1992)
40. Rajkumar, R., Lee, I., Sha, L., Stankovic, J.: Cyber-physical systems: the next computing revolution. In: Design automation conference. pp. 731–736. IEEE (2010)

41. Reiter, R.: A theory of diagnosis from first principles. Artificial intelligence **32**(1), 57–95 (1987)
42. Ruiz, R., Riquelme, J., Aguilar-Ruiz, J., Garcia Torres, M.: Fast feature selection aimed at high–dimensional data via hybrid-sequential-ranked searches. Expert Systems with Applications **39**, 11094–11102 (09 2012)
43. Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., Teneketzis, D.: Diagnosability of discrete-event systems. IEEE Transactions on automatic control **40**(9), 1555–1575 (1995)
44. Scanagatta, M., Campos, C.d., Corani, G., Zaffalon, M.: Learning Bayesian Networks with thousands of variables. Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS 2015) pp. 1855 – 1863 (01 2015)
45. Schwarz, G.: Estimating the dimension of a model. The Annals of statistics **6**(2), 461–464 (1978)
46. Shao, S., Sun, W., Wang, P., Gao, R., Yan, R.: Learning features from vibration signals for induction motor fault diagnosis. In: International Symposium on Flexible Automation (ISFA). pp. 71–76 (2016)
47. Singh, R.: Symptom Checker with machine learning dataset. `https://www.kaggle.com/rabisingh/symptom-checker` (2020)
48. Smolensky, P.: Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. In: Rumelhart, D., McClelland, J.L., Group, P.R. (eds.) Information Processing in Dynamical Systems: Foundations of Harmony Theory, pp. 194–281. MIT Press (1986)
49. Steuer, R., Daub, C., Selbig, J., Kurths, J.: Measuring distances between variables by mutual information. In: Innovations in Classification, Data Science, and Information Systems. pp. 81–90. Springer Berlin Heidelberg (2005)