# Differential Diagnosis with Active Testing

Emile van Gerwen[1], Leonardo Barbini[2], and Michael Borth[3]

[1,2]*TNO-ESI, Eindhoven, The Netherlands*
*emile.vangerwen@tno.nl*
*leonardo.barbini@tno.nl*

[3]*TNO, Helmond, The Netherlands*
*michael.borth@tno.nl*

## ABSTRACT

The diagnosis of complex systems benefits greatly from a differential, multistep approach that narrows down the list of possible conditions or failures that share the same observable effects to a single root cause. We provide a suitable and practically applicable methodology for this. In extension to existing work, it covers all types of diagnostic actions, i.e., the observation of system properties, active testing and system interventions like providing a dedicated diagnostic input or forcing the system into discriminating states, but also the replacement of components. Combining all these possible steps into one probabilistic and causal reasoning framework, we I) stepwise generate the diagnostic model systematically to correctly cover the interplay of observations and diagnostic interventions, and II) provide decision support based on counterfactuals for the selection of the next diagnostic step, countering the vast number of possible actions that arise in machine diagnostic processes. We developed and successfully tried our methodology for diagnosing cyber-physical systems in the high-tech industry, but we found that it supports more processes, such as computing intervention actions for autonomous robots.

## 1. INTRODUCTION

*Machine Diagnosis*, understood as the identification of the nature and cause of a certain and unwanted phenomenon within a technical system by means of at least semi-automated analytics, is key in after-sales processes of many industries. It allows systems to partially self-diagnose, thus offering users of production equipment reduced down-times either by enabling them to fix issues themselves or by ensuring that service personnel come in with the right parts for potential hardware failures. Further, it tunes maintenance towards minimal costs and counters the shortage of experienced service engineers by raising their efficiency and by enabling less experienced personnel to perform the tasks.



One crucial design decision in the realization of a system for machine diagnosis is which information it processes. Failure effects and measurements of key performance indicators are typical candidates that can be observed and processed, but experienced service engineers consider more: they either use active testing, i.e., they set system states or provide the system with tailored inputs to discriminate between root causes or undertake costly investigative part replacements.

In the work presented here, we extend previous work for machine diagnosis (Barbini 2020) towards this level of analytical prowess and introduce automated reasoning for differential diagnosis with active testing and other interventions within a unified framework. For this, we introduce our method of diagnosis with causal Bayesian Networks in Section 2 and provide an example in Section 3 that shows that interventions are necessary in machine diagnosis and illustrates our solution for handling them. In Section 4, we extend our methodology with counterfactual reasoning to select the best diagnostic action. Section 5 illustrates two of our use cases, and Section 6 concludes with future topics.

## 2. DIFFERENTIAL MACHINE DIAGNOSIS

### 2.1. Diagnosis using Causal Belief Networks

Bayesian Belief Networks, also known as Bayes Nets (BN), are probabilistic graphical models that were shown to deliver excellent diagnostic capabilities (Heckerman, 1995, among other works). Introduced by Pearl (1986), they offer probabilistic reasoning for taking an observable event and inferring the likelihood that any one of several possible known causes was the contributing factor. In this, they form an efficient solution to compute the marginal distribution of the known causes, i.e., of a hypothesis *H*, given observations or evidence *e*, following Bayes' Theorem:

$$P(H|e) = \frac{P(e|H) * P(H)}{P(e)}$$

with P(H), P(e) the probability of observing H and e (the priors or marginal distributions), and P(H|e), P(e|H) the conditional probability of H and e, given the other one.

Defined as directed acyclic graphs (DAGs) whose nodes represent variables in the Bayesian sense and whose edges represent conditional dependencies, BNs encode the joint probability distribution over all their variables. Their advantage for the purpose of diagnosis stems in part from the versatility that those variables may be observable, latent, and even unknown parameters or hypotheses, and that, with the joint probability distribution, all arbitrary combinations of variables may be considered. Moreover, BNs always consider all causes that are possible given the observations but rank them according to their likelihood that depends on both the observations, i.e., the provided evidence, and the a priori probabilities (the priors).

Seeing that they are among the few inference techniques that combine knowledge- and data-driven modeling (Jensen & Nielsen, 2007), there are many options to build or generate BNs.

Our own approach for the construction of BNs for the diagnosis of complex systems is based on modeling the knowledge on the system decomposition, deployment, and functional behavior using a domain specific language and the subsequent use of generative techniques to compose the needed BN from pre-build network fragments and rule-based elements such that it conforms to that knowledge. Introduced by Borth & von Hasseln (2002), we detailed and extended this approach in (Barbini & Borth, 2019) and (Barbini et al., 2020), but, so far, limited ourselves to BNs for diagnostic processes that only use a set of simultaneous observations or a sequence of observations – but not a mixed sequence of observations and interventions.

In essence, we (and others, compare, e.g., work by Ricks & Mengshoel, 2009) generated diagnostic systems that use the full power of Bayesian Belief Networks which mirror a system as it is, but did not utilize the diagnostic prowess of interventions to the system. A major reason for this restriction was that interventions alter the system's causality – and neither modelling techniques nor inference calculi were ready to handle this until quite recently.

This changed with Pearl's *do(x)* operator (2009) that represents interventions and allows to correctly predict effects of such deliberate actions in causal networks. *do(x)* operator interventions have a different meaning and diagnostic power than statistical associations (Pearl & Mackenzie, 2018): $P(Y|x) > P(Y)$ simply states that observing *x* raises the probability of *Y*, which might have other reasons including a common cause, while $P(Y|do(x))$ describes the situation after performing the action *x* that affects *Y* and eliminates the effects of other confounding factors. Within diagnosis, this translates, e.g., to the difference between observing the availability of power at a connector – with the conclusion that the functional chain to provide that power is intact – and experimentally providing power at that point, invalidating all assumptions about that chain while offering deductions from the effects observed due to the intervention.

### 2.2. Differential Causal Reasoning with Bayes Nets

Within the probabilistic graphical model, the distinction of observing *x* and *do(x)* is paramount, as the first provides evidence for the variable *x*, i.e., setting the probability of the observed state to 1, leaving the rest of the network as it was, while the second also changes the network structure to reflect the new causal flow. A differential diagnosis using a mixed sequence of observations and interventions therefore requires a new technique of stepwise inference and network augmentation. Figure 1 depicts our methodology for this.

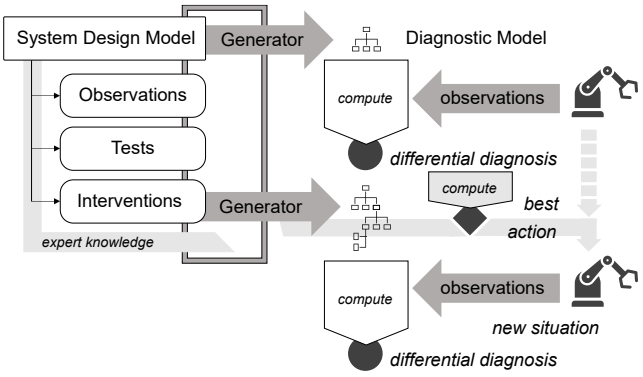


Figure 1. Differential Machine Diagnosis

As the graph shows, the diagnostic model is generated from the system design, with augmentations to the diagnostic model following subsequently for conducted actions that change the diagnosis and, in the case of interventions, the system (Section 3). The observations from the system itself allow us to compute the differential diagnosis in each step.

Using this process and given the encoded expert knowledge, we generate diagnostic models that I) accurately represent a sequence of diagnostic tests of all kinds, i.e., the observation of additional properties, the setting of a different input, system interventions, and the replacement of components, and II) correctly carry the information between steps.

The resulting differential diagnosis covers single fault as well as multi-fault scenarios. Intermittent faults, however, cannot be handled directly in this way (or any comparable approach), as the absence of a fault effect observation potentially leads to the diagnostic network ruling out the respective root-cause.

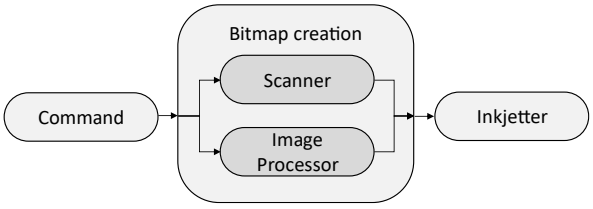


Figure 2. Schematic representation of an all-in-one printer

## 3. INTERVENTIONS

We use an idealized version of a printer as an example to illustrate our approach to handle interventions in differential diagnosis. Figure 2 shows a schema of the idealized printer.

Based on a given *Command*, either *print* or *copy*, the printer should create a *Bitmap* and produce a printed sheet with the *Inkjetter*. For the *Command copy*, the printer needs only the *Scanner* working to produce a *Bitmap*. Similarly, for the *Command print*, it needs only a working *Image Processor* to produce a *Bitmap*. The BN to diagnose the idealized printer is shown in Figure 3.[1] Each system component is modelled with inputs, outputs, and health node. The system is created by considering the outputs of a component as the inputs of the following component. For details on how to generate this BN, we refer to (Barbini et al., 2020).

### 3.1. System and component health

In the BN figures below, nodes with a green identifier in the upper right corner represent the hidden health variable of components. In this example, we assume that the *Command* never fails, so it does not have a health node. The *Scanner* is *broken* with prior probability 5%, the *Image Processor* with 1% and the *Inkjetter* with 10% respectively. The other nodes in the BN are not hidden, i.e., are nodes for which the state can be measured.

For our diagnostic task, we use the BN to infer the probability distribution of the health nodes, given evidence in the other nodes. An example of how to perform a diagnosis with the BN is shown in Figure 3. We suppose that the *Command* is set to *copy*, but we do not obtain a *Printed Sheet.* The diagnosis, as the inferred probabilities on the health nodes, is very uncertain and the goal of our differential diagnosis is to perform diagnostic tests to reduce such uncertainty.

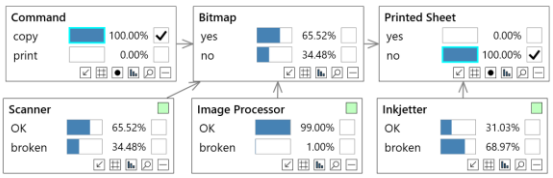


Figure 3. BN of all-in-one printer. Copying fails.

### 3.2. Diagnostics tests

There are four different types of diagnostic tests. Each type comes with specific modelling and BN interactions.

#### Type I: observe additional properties

The first type of diagnostic test, widely covered by existing work on BN for diagnosis, consists of adding additional evidence to one of the observable nodes. This is shown in Figure 4, for the observation of an existing *Bitmap*. This test is executed by probing the printer to collect the additional observation on the presence of the *Bitmap*.


Figure 4. Type I: Observe additional properties

#### Type II: set a different input

A second type of diagnostic test consists of setting a different input to the system. This is a fundamentally different type of test than the previous one. Firstly, we are not only observing, but we are intervening in the system, by doing the action of changing an input. Secondly, the result of the action could potentially lead to a different observation than in the absence of said action, making it counterfactual.

Our approach is to augment the BN when doing a test comprising an intervention. This augmentation procedure has two steps. Firstly, we duplicate all the nodes in the BN except for the *health* nodes and the *input* nodes which are not set to a different value (with *input* nodes referring to BN nodes that have no parents and are not of type *health*). Secondly, we connect the duplicated nodes as children of the original *health* nodes. This approach – and not, as is sometimes assumed, just updating the priors using the result of the inference – delivers the correct inference model, as Balke & Pearl showed (1994). Notice that with this augmentation procedure there is no need to specify new conditional probability tables, as they are the same as in the original, not augmented BN.

Figure 5 shows an example of this augmentation procedure. The starting point is the diagnosis BN in Figure 3. The duplicated nodes have a *1* at the end of their names, e.g., *Command1*. We now observe a *Printed Sheet1* for the printer input *Command1 print*. As the augmented BN's reasoning also includes the evidence coming from the situation when copying failed, it infers that the only possible diagnostic solution is that the *Scanner* is *broken* and the *Image Processor* and *InkJetter* are *OK*.

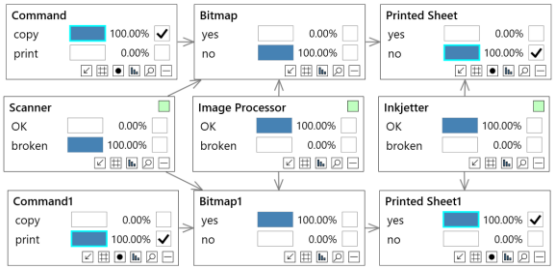


Figure 5. Type II: set a different input

[1] We modelled all the BNs with the Bayes Server software [Bayes Server].

### Type III: replace a component

The third type of diagnostic test is the replacement of a component. This is also an active type of test, and thus of a similar nature as the previous one. When doing differential diagnosis with this type of test we also need to augment the BN, but – differently than with Type II tests – we duplicate its health node when we replace a component.

Figure 6 shows this procedure on the idealized printer for the case that we replace the *InkJetter* after copying fails. For the replaced component we assumed that *InkJetter1* is known to be functioning (this is not necessary and might be relaxed). The *input* node *Command* is not set to a different value when the printer has a new *Inkjetter*, and is thus not duplicated, making it a common parent for both the *Bitmap* and *Bitmap1* node.

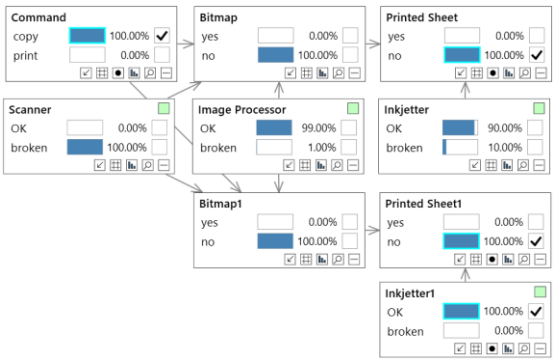


Figure 6. Type III: replace a component

The observation related to the action of replacing the *InkJetter* is that there is still not a *PrintedSeet1*. The BN infers that a *broken Scanner* explains the situation, allowing even a switch back to the original *InkJetter*, as the diagnosis established that the failure had to progress via the *Bitmap*, a statement that could not have derived from another probe.

### Type IV: intervene within the system

The fourth type of test consists of forcing some of the physical quantities to a given value. In practical situations this corresponds to performing an action in the system, like connecting an external power supply to a part, and then observing the results in other parts, for example whether these are now correctly functioning. This test type also entails an intervention on the system and modeling it requires both an augmentation of the BN and the *do(x)* operator introduced above.

Figure 7 shows an example of type IV testing: After copying fails, we try to print, which is a Type II test, but then we still do not get a *Printed Sheet1*. At this point, we intervene in the system and manually provide a *Bitmap2* to the *InkJetter*. The light gray arrows pointing into *Bitmap2* node indicate the obstruction of information flow.

The BN is consequently augmented with the duplication procedure described above, but in this type of test, we insert evidence with the *do(x)* operator on the duplicated node on which we force its physical quantity to a given value. In the case of Figure 7, this is the *Bitmap2* node, and the use of *do(x)* to set the evidence is shown with a red check mark.

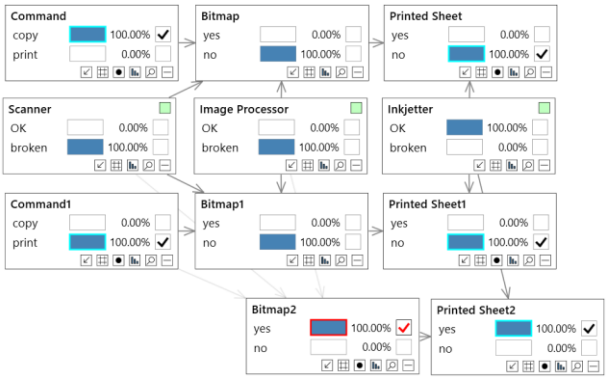


Figure 7. Type IV: intervene within the system

In our example, we finally observe a *Printed Sheet2* after providing *Bitmap2*. The augmented BN infers that the only possible solution to the diagnostic problem is that both the *Scanner* and the *Image Processor* are *broken* and the *InkJetter* is *OK*.[2]

## 4. Decision Support

A major part of our system for machine diagnosis is to recommend the next best action to service engineers, as this increases their efficiency and reduces system down-time. To do so, we assess the value of possible diagnostics tests, i.e., we determine how much we would learn from them using counter factual 'what-if' reasoning. A test can be of any of the types outlined in the previous section. The result of each "what-if" scenario is the information gain or equivalently the reduction of uncertainty, for which we use an entropy measure (Oladyshkin & Nowak, 2019).

Ideally, we would consider all possible tests sequences and then pick the one that is expected to lead to the diagnostic solution with minimal efforts. However, considering that for large systems there are hundreds of possible tests at any point, this approach seems infeasible. Instead, encouraged by de Kleer and others (1992) stating that "one step lookahead is pretty good", we take the myopic approach in which we look only at the currently available tests, and pick the test based on the expected entropy within the BN after doing that single test.

[2] Without the *do(x)* operator, this evidence on *Printed Sheet2* could not be set. It would have a null probability of finding as we tried both printing and copying and both failed. The *do(x)* operator deletes the dependence of *Bitmap2* on its parent nodes and as a result one can have a *Bitmap2* even if both the *Scanner* and the *Image Processor* are *broken*.

Intuitively, the entropy over the component health states is a measure of the uncertainty we have about the health of these components. The diagnostic process is then doing tests to reduce the entropy as much as possible.

If we denote the set of health nodes in the Bayesian network with $\Omega$, conditional entropy in model $M$ with $H_M(\cdot \mid \cdot)$, and $T_i = (A_i, O_i)$ a possible test that comprises of an action $A_i$ and an observation $O_i$, we calculate the recommended test $T^*$ as

$$T^* = \underset{T_i}{\mathbf{argmax}}\left( H_M(\Omega) - H_{M \leftarrow A_i}(\Omega \mid O_i)\right) \qquad (1)$$

where M is the model representing the current situation and $M \leftarrow A_i$ the hypothetical situation if we would perform action $A_i$ on model $M$ as described in section 3. Notice that in the same $T_i$ the action and the observation can be performed in different system's components.

Calculating $T^*$ thus requires generating networks for every action $A_i$.

Once the recommended test is executed, we generate the corresponding network, add the evidence that reflects the outcome of the test, and calculate the posterior probabilities of the components' health, thereby presenting the new failure hypotheses. This procedure is repeated until a solution to the diagnostic problem is found.

Equation (1) does not consider the costs such as time spent and monetary replacement costs of various tests but of course these must be incorporated. Intuitively, a test that reduces uncertainty the most per dollar spent is the most valuable. Assuming a linear relation between the uncertainty reduction and cost, Eq. (1) then becomes

$$T^* = \underset{T_i}{\mathbf{argmax}}\left( \frac{H_M(\Omega) - H_{M \leftarrow A_i}(\Omega \mid O_i)}{C(A_i) + C(O_i)}\right) \qquad (2)$$

where $C(A_i)$ is the cost of doing action $A_i$ and $C(O_i)$ is the cost of observation $O_i$.

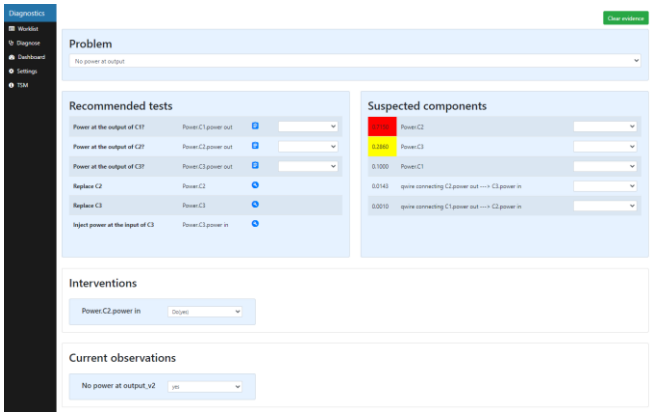


Figure 8. Prototype Diagnosis Tool with Decision Support

## 5. USE CASES

We applied the described methodology to two use cases, targeting support service engineers in one, and in the other autonomous decision making in surveillance robots.

*Production Printing*

Canon Production Printing, part of Canon Inc., designs and manufactures large professional printers. In the professional markets, like book on demand printing, minimal downtime is key to reduce total costs of ownership, but the avoidance of unscheduled downtimes is of equal importance. Especially for the first of these objectives, service engineers must be able to quickly diagnose misbehaving printers – a task that is complicated by the high system complexity that resulted from increasing demands on print quality as well as on production throughput. To assist these service engineers, we pioneered and established our methodology on a subsystem of about 50 potentially failing components and qualitatively assessed the recommendations given by our diagnostic system in different scenarios.

The results exceeded expectations and by now led to Canon Production Printing incorporating the method in their service tooling, for which a prototype is shown in Figure 8. Factors in this success were stringent diagnostic conclusions that stem from our diagnostic basis in system modelling, and the push towards the most likely root cause that follows from the a priori data on component failure likelihoods but also from the system's ability to propose novel diagnostic strategies that minimize efforts. We saw our model ruling out many potential faults with quick active tests that were seemingly unrelated to the current issue, but surprisingly greatly reduced the uncertainty on the system's state.

*Adaptive Behavior in Robot Dogs*

The ability to compute which intervention would best enhance an active investigation also allowed us to improve adaptive behaviors of autonomous robot dogs (TNO, 2022).

One of the use cases for these robot dogs is the inspection of industrial sites, e.g., to check for leaking pipes. Equipped with gas sensors and both visual and acoustic cameras, these robots move in noisy environments that are likely affected by wind, smog, and fog. It is thus not sensible to simply follow prescribed procedures for the inspections: the robots must instead actively seek to improve their performance, especially their detection and perception capabilities. We reached such adaptive behavior with an online diagnosis system that reasons about causes for low performance and then computes the most promising intervention, like moving in the line of the wind or out of a zone with high noise.

Even though the system descriptions focus on performance next to the functionality for the robots, and the possible interventions included behaviors, we followed the same methodology in both studies. This gives us confidence that our approach is applicable to many scenarios and domains.

## 6. CONCLUSION AND FUTURE WORK

Being able to deal with intervening diagnostic tests is crucial for a real-world diagnostic support tool. Based on the long-standing theory of counter factual reasoning, we can now deal with these interventions in practice. Preliminary results on small use cases indicate this method indeed enables a structured way of deciding among different possible actions and warrants scaling up to large systems.

Nevertheless, the decision support as outlined in section 4 can be refined in several ways.

Firstly, Eq. (1) inspired by existing service manual procedures, assumes a fixed relation between an action $A_i$ and observation $O_i$. In theory, however, we could do any action $A_i$ and the observe some $O_j$, leading to

$$T^* = \underset{T_{ij}}{\mathbf{argmax}}\left(\frac{H_M(\Omega) - H_{M \leftarrow A_i}(\Omega|O_j)}{C(A_i) + C(O_j)}\right) \qquad (3)$$

However, many of the pairs $(A_i, O_j)$ will not make sense in a practical situation so the computational complexity might not be worth the additional information gain.

Secondly, the method assumes that costs are independent while typically they depend on the system state $M$. A canonical example is that many actions and observations might be expensive because they require opening the system, but once opened, all of these become cheap. The greedy one step look ahead might therefore not be optimal. The most challenging part here is to find the sweet spot between planning optimality and ease of cost structure specification.

Finally, we might relax the linear dependency between entropy reduction and cost and allow for a more finetuned relation, leading to a nominator $f(H_M(\Omega) - H_{M \leftarrow A_i}(\Omega|O_i)))$ in Eq. (2) where $f$ is to be optimized.

All these extensions are currently being researched or planned as future work.

## ACKNOWLEDGEMENT

The research is carried out as part of the Carefree project under the responsibility of TNO-ESI with Canon Production Printing as the carrying industrial partner. The Carefree research is supported by the Netherlands Organisation for Applied Scientific Research TNO as part of the Appl.AI program.

## REFERENCES

Balke A, Pearl J, (1994) Probabilistic evaluation of counterfactual queries. AAAI-94 proceedings

Bayes Server, www.bayesserver.com, retrieved 24-03-2023.

Barbini, L., & Borth, M. (2019). Probabilistic Health and Mission Readiness Assessment at System-Level. Annual Conference of the PHM Society, 11(1).

Barbini, L., Bratosin, C., & van Gerwen, E. (2020). Model-based diagnosis in complex industrial systems. PHM Society European Conference, 5(1), 8.

Borth, M., & von Hasseln, H. (2002). Systematic generation of Bayesian networks from systems specifications. In Intelligent Information Processing: IFIP 17th World Computer Congress on Intelligent Information Processing August 25–30, 2002, Montréal, Québec, Canada 1 (pp. 155-166). Springer US.

de Kleer, J., Raiman O., Shirley, M. (1992) One Step Lookahead is Pretty Good, In: Readings in Model-Based Diagnosis, Hamscher, W., de Kleer, J. and Console, L., Morgan Kaufmann.

Heckerman, D., Mamdani, A. & Wellman, M.P. (1995). Real-world applications of Bayesian networks. *Communications of the ACM 38.3*: 24-26.

Jensen, F.V. & Nielsen, T.D. (2007). Bayesian Networks and Decision Graphs, Springer Verlag.

Oladyshkin, S. & Nowak, W. (2019). The Connection between Bayesian Inference and Information Theory for Model Selection, Information Gain and Experimental Design. *Entropy*, vol. 21, no. 11, Art. no. 11, Nov. 2019, doi: 10.3390/e21111081.

Pearl, J. (1986). Fusion, propagation, and structuring in belief networks. *Artificial intelligence*, 29(3), 241-288.

Pearl, J. (2009). Causality. Cambridge University Press.

Pearl, J. & Mackenzie, D. (2018). The Book of Why: The New Science of Cause and Effect. Basic Books.

Ricks, B. W., & Mengshoel, O. J. (2009). Methods for probabilistic fault diagnosis: An electrical power system case study. Annual Conference of the Prognostics and Health Management Society.

TNO (2022). Situational awareness in robot dogs. www.tno.nl/en/digital/artifical-intelligence/safe-autonomous-systems/situational-awareness-robot-dogs/