Combining Knowledge Modeling and Machine Learning for Alarm Root Cause Analysis

Lisa Abele * **Maja Anic** * **Tim Gutmann** * **Jens Folmer** **
Martin Kleinsteuber * **Birgit Vogel-Heuser** **

* *Department of Electrical Engineering and Information Technology, Technische Universität München, Munich, Germany*
** *Department of Mechanical Engineering, Technische Universität München, Munich, Germany*

Abstract: Industrial alarm systems inform the operator of abnormal plant behavior and are required to guarantee safety, quality, and productivity of the plant. However, modern alarm systems often produce large amounts of false or nuisance alarms which leads to alarm floods. Operators receive far more alarms than they can handle. To reduce these alarm floods, we developped an alarm system that performs *Root Cause Analysis (RCA)* upon an alarm model constructed with Bayesian networks. In this paper, we present methods to construct Bayesian networks for RCA with a knowledge-based and a machine learning approach. Finally, we evaluated both approaches with an example of an industrial plant and propose an architecture to combine both approaches.

1. INTRODUCTION

Fault detection in industrial plants has been a very active field of research Izadi et al. (2009). Alarm systems were designed to raise an alarm when a fault is detected. Operators often receive false or nuisance alarms and thus far more alarms than they can handle. The operator is distracted and might ignore critical alarms. An example for such a critical situation is an explosion that occurred in the Texaco Milford Haven Refinery in 1994, which cost the lives of four workers Adnan et al. (2011). This accident could not be prevented, although the alarm system was intact. The operators faced an alarm flood of 275 eleven minutes prior to the explosion; a number impossible to cope with. To avoid these situations, the amount of false or nuisance alarms has to be reduced by alarm systems.

Our aim is to reduce such false or nuisance alarms with an alarm system that performs *Root Causes Analysis* upon an alarm model. *Root Cause Analysis* is based on the assumption that a problem is solved and prevented from recurrence if the root cause is identified and eliminated Rooney and Heuvel (2004). For the alarm model, we considered Bayesian networks, because they allow to model conditional dependencies and thus to identify root causes with their probability of occurrence.

To handle the problem of alarm flooding, the best solution is the radical redesign of the alarm system. But redesign is expensive and error prone due to manual changes. Besides, modern plants are a complex network of devices which are subject to frequent modifications. Consequently, the entire alarm system has to change to a dynamic one which is highly reusable and extensible.

Knowledge-based approaches offer the required properties. For this reason, we used the knowledge-based approach to build a knowledge-based Bayesian network model where rules can infer the root causes. But current knowledge-based alarm system have several drawbacks, they are language dependent, hardly scalable and the development is time and effort consuming. To cope with these issues, our aim is to combine the knowledge-based approach with the strengths of a machine learning approach to define a high-performance alarm system. A machine learning approach is used to perform supervised learning of Bayesian networks from existing alarm data.

The goal of this paper is the development of a reusable, extensible alarm system that reduces the redundancy of alarms to avoid alarm flooding and that supports the operator in his decision-making task by providing the root causes of alarms and their probabilities of occurrence.

In the following section, we first introduce the basics of alarm models, in particular Bayesian networks. In Chapter 3, the knowledge- and the machine learning approach are presented. In Chapter 4, we evaluate our results on a small case study from process automation. Chapter 5 proposes an architecture for the alarm system and concludes.

2. METHODS

Alarm flooding is an extraordinary plant state, where the rate of incoming alarms exceeds the human reception capacity; the operator is overtaxed and as a result not able to make diagnoses and take actions in required response time Izadi et al. (2009); Kondaveeti et al. (2010). Sequential structures of alarms can be modelled with causal networks to analyze how such alarm floods evolve. Causal networks are directed acyclic graphs (DAGs) that consist of a set of nodes (domain variables) and a set of directed links (relations) between the nodes. The links are of causal nature, which means they lead from cause to effect Jensen and Nielsen (2007).

Figure 1 shows an example of an alarm flood. Only the first alarms of an alarm sequence are of interest for the operator. An alarm without predecessor is called *root alarm*. These alarms are root causes for all other alarms in a network.

Bayesian networks (BNs) enhance causal networks with probabilities. This feature makes them an excellent tool for reasoning

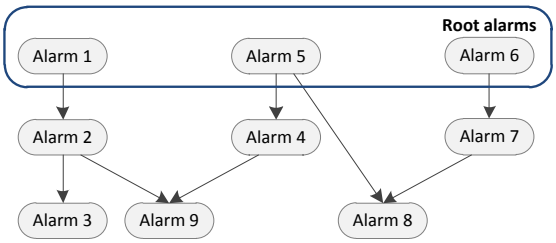

Fig. 1. Example of an alarm flood: The arcs depict the dependencies between the alarms. Alarm 1, 5 and 6 are *root alarms*. All other alarms are consequences of these three alarms.

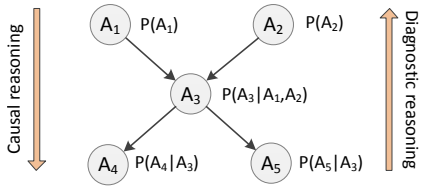

Fig. 2. A simple BN with two root nodes A_1 and A_2 and three conditioned nodes A_3, A_4 and A_5.

under uncertainty and offers our system the possibility to identify root alarms with their probability of occurrence.

A BN $\mathcal{B} = (\mathcal{G}, \mathcal{P})$ graphically shows the relationship between random variables of a probability distribution $\mathcal{P}$ as specified by Kjaerulff and Madsen (2008). The representation of the network structure is a DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of vertices and $\mathcal{E}$ the set of edges of the graph structure. Each vertex $v_i \in \mathcal{V}$ corresponds to a random variable X_{v_i} in $\mathcal{X}_\mathcal{V}$ for all i = 1, . . . , n. Each edge is a directed link between two vertices, which represents a conditional dependency of these vertices.

The joint distribution $\mathcal{P}$ over the entire network can be factorized as

$$P(\mathcal{X}_\mathcal{V}) = \prod_{i=1}^{n} P(\mathsf{X}_{v_i} | \mathsf{X}_{pa(v_i)}),$$

where $\mathsf{X}_{pa(v_i)}$ denotes a set of parent variables of the variable X_{v_i}. This is the *chain rule for BNs* or *Markov assumption*. Figure 2 depicts a BN of five vertices.

In general, Bayesian networks can be inferred or learned. To infer a BN, we defined a knowledge-based approach which requires a set of facts and axioms and reasoning mechanism. To learn a BN, we implemented a structure learning approach which requires alarm data with relevant statistical structure.

2.1 Infering Bayesian Networks

The knowledge describing an industrial plant, that is to be modeled, is provided by system experts, the operator, and system documentation in form of data sheets, operator handbooks, or piping and instrumentation diagrams (P&IDs). It is referred to as expert knowledge. The expert knowledge is in a raw state. It is scattered, unsorted, and unstructured, and needs to be processed which is the main task of the knowledge engineer. The knowledge engineer can apply two useful and complementary analysis techniques from the field of Reliability Engineering to collect and structure the expert knowledge: *Failure Mode and Effect Analysis (FMEA)* and *Fault Tree Analysis (FTA)* Pradhan et al. (2007).

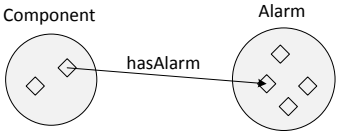

Fig. 3. Graphical representation of the three main elements of an *OWL*-ontology: individuals (diamonds), properties (links), and classes (circles).

A knowledge-based alarm system that assists the operator in decision-making requires three main components: a knowledge base containing the structured expert knowledge and rules, an inference engine which processes the rules, and a user interface to enable the interaction with the operator. These components must satisfy a number of basic requirements which are summarized in the following.

The knowledge base should

- be extensible,
- appropriately represent uncertain knowledge

The inference engine, should automatically and rapidly draw inferences from a knowledge base.

The user interface should be intuitive to use and flexible concerning new knowledge.

Main tasks of a knowledge engineer is to collect, structure, and model the expert knowledge to create a knowledge base. These tasks require a close communication between the knowledge engineer and the expert and a free access to system documentation.

The expert, for his part, has to check the final model for plausibility. Hence, both the process of modeling and the process of evaluation rely on the expert's knowledge, experience, and opinion.

Based on these requirements, we evaluated appropriate tools.

Ontologies Ontologies are semantic data models that represent the knowledge about a domain through classes, individuals, and relations. They can be interpreted as more-structured and detailed dictionaries that specify the domain vocabulary and its proper usage Gruber (1993).

To give an example, we define a class *Component* and a class *Alarm* and concrete class objects, called indivduals connected with the directed link *hasAlarm* as shown in Figure 3. Ontologies create a common understanding of the terms and relations applied in a certain domain upon which knowledge sharing is possible. Moreover, an ontology is extensible and makes knowledge reusable Noy and McGuinness (2001).

Ontologies in OWL store axioms which allows inference mechanisms. However, ontologies lack a direct way to express uncertain knowledge. Therefore, we inserted rules to act along the knowledge stored in the ontology to derive new facts.

Rule engine To built a Bayesian network based on the expert knowledge, we used *Prolog* as inference engine. *Prolog* stands for *programming in logic* (Bratko, 2012, p. 20 ff). Facts and rules define relations between objects and form the knowledge base of every *Prolog* program.

Bratko developed an interpreter for BNs, a *Prolog* program that computes conditional probabilities in BNs (Bratko, 2012,

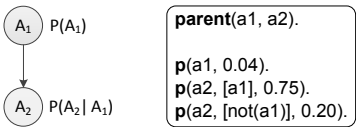

Fig. 4. Simple BN example and the corresponding *Prolog* code. The *Prolog* code separately comprises the qualitative and quantitative part of the BN.

Table 1. Examplary dataset for structural Bayesian Learning

case	A_1	A_2	...
1	1	low	...
2	2	medium	...
3	?	high	...
4	2	?	...
5	3	high	...
⋮	⋮	⋮	

p. 378 ff). The basic module of this interpreter is the Bayes formula. We extended this interpreter by adding inference rules to determine the root causes of alarms and their corresponding probabilities of occurrence.

This interpreter is referred to as extended Bratko interpreter. Our interpreter requires a knowledge-based model (e.g. an ontology) that contains the structure and the parameters of a BN as shown in Figure 4.

2.2 Learning Bayesian Networks

This section focuses on an approach to learn a Bayesian model with relevant statistical alarm data. In contrast to the previous approach, no additional expert knowledge is required for the learning task.

Our interest concerns the structure of the alarm sequences to identify the required *root alarms*. Therefore we evaluated several algorithms for structure learning on Bayesian networks.

Structure Learning A dataset $\mathcal{D}$ for structure learning of Bayesian networks should have a schema as shown in Table 1 and contains categorical data. The rows of this dataset are called cases. In such a case, a value is assigned to each variable $\mathsf{A}_i \in \mathcal{X}_V$. A_i represents a single alarm in the plant.

Several requirements were identified and used to evaulate the structure learning algorithms. (1) The alarm dataset must satisfy several assumptions according to Spirtes et al. (2000):

- The set of observed variables is causally sufficient.
- Every case has the same causal relations among variables.
- There are no dependencies between cases.
- The underlying probability distribution $\mathcal{P}$ is faithful to an acyclic directed graph of the causal structure.

These assumptions may be violated, e.g. if the dataset does not have sufficient cases.

(2) The number of variables have to be in the adequate range. The article Robinson (1977) proposes a recursive equation

$$f(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} f(n-i), \text{ with } f(1) = 1 \quad (1)$$

which calculates the number of possible DAGs as a function of the number of nodes in the BN. The increasing number of variables in a network leads to a super-exponential growth in $f(n)$. This clarifies the difficulties of finding the fitting structure for a larger network.

(3) This requirement concerns the conditional independence and dependence relations (CIDs), which a BN embodies. Some learning algorithms revert this procedure and use CIDs to recover the structure. But the fact that A is dependent of B and C, written as $\mathsf{A} \not\perp\!\!\!\perp \mathsf{C}|\mathsf{B}$, can be derived from three different structures, see Figure 5. Some of these structures are *Markov equivalent* (Koski and Noble, 2009, p. 71f) and were learned with identical data sets. An unambiguous assignment gets thus impossible.

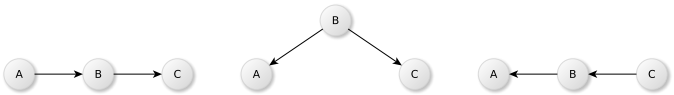

Fig. 5. DAGs of the same Markov equivalence class (Koski and Noble (2009))

For learning the causal structure of alarm sequences, we evaluated if *constraint-based* (cb) and *score-based* (sb) algorithms meet our requirements. The cb algorithms rely on the conditional independences and dependences (CIDs) induced from the data, the sb algorithms perform a search through the space of possible DAGs and return the DAGs with the highest score Jensen and Nielsen (2007). In an ideal setting, i.e. infinite data and fully observed data, both approaches will recover the same DAG Bouckaert (1995), but within a usual setting, i.e. with a finite dataset, they might differ. Considering Equation 1 we found that the drawback of the sb algorithm is its large search space in BNs with multiple nodes. Heuristic search algorithms could help here but their search might not return the original network and gets stucked in a local optimum Steck (2001), such that the found graph is not at all related to the real causal structure.

An issue with the cb approach is the robustness of the algorithm Margaritis (2003). Robustness means that a single error in an early phase of the algorithm will entail many errors in the final graph. But these algorithms are deterministic, have a well-defined stopping criterion Dash and Druzdzel (2003), and rebuild the network by searching for CIDs in the data and find the causal relationships within the data. We decided thus to better rely on the alarm structure learned by the cb algorithm. In a next step, we compared several vendor-independent tools for cb structure learning. All tools are freely available for research purposes.

Tools for Structural Learning We considered three tools: SMILE/GeNIe [1] , the Tetrad Project [2] , and the Weka suite [3] .

Table 2 summarizes the results of our comparison. Criteria were the quality of reconstruction of the network structure, the ergonomy of the GUI, and the features provided to simplify the time-consuming preparation of the dataset. Additional features, such as libraries and software licenses, were considered as further criteria. These features are important to integrate the tools in our alarm system.

[1] http://genie.sis.pitt.edu/
[2] http://www.phil.cmu.edu/projects/tetrad/
[3] http://www.cs.waikato.ac.nz/ml/weka/

Table 2. Comparison of vendor-independent BN learning tools.

	Tetrad	**Weka**	**GeNIe**
Reconstruction	very good	errors in large networks	good
GUI	intuitive	intuitive, but problems with displaying larger networks	intuitive
Functional range	many cb algorithms	1 cb and several sb algorithms	1 cb and several sb algorithms
Requirements	JRE $\geq$ 1.6	JRE $\geq$ 1.6	MS Windows
Libraries	Java	Java	C++
License	GNU GPL	GNU GPL	Proprietary, Freeware

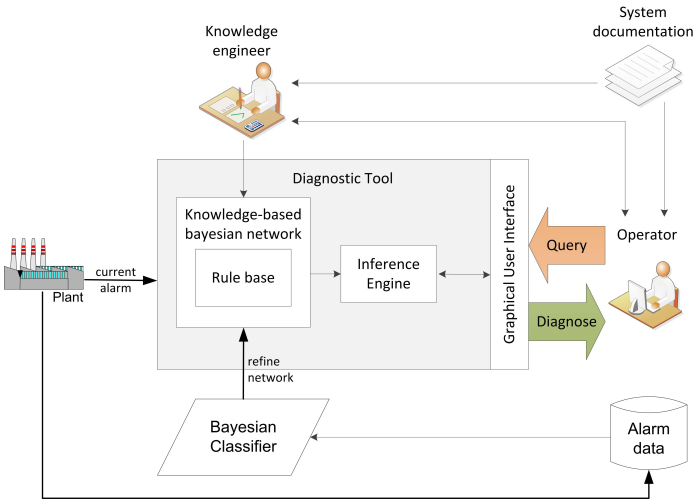

Fig. 6. System architecture of the combined diagnosis system (arrows represent the data flow). Both, expert knowledge and data, are used to build a BN representing the alarm dependencies.

Based on the results of the evaluation, we identified Tetrad as best tool. Tetrad offers a variety of algorithms and showed the best results in reconstructing the original alarm structure.

3. EVALUATION

In this section, we demonstrate in an example how to combine the knowledge and the machine learning approach in one application.

In a first step, the knowledge-engineer uses the input of the operator and system documentation to define a knowledge-based bayesian network with its plant specific rule base. The operator can query the diagnosis system to get the root cause of the current alarm which is inferred by an inference engine. When a sufficient amount of alarm data is gathered, this data is processed by a bayesian classifier. The results are then used to refine the bayesian network. The architecture of the diagnosis system is represented in figure 6.

We defined an appropriate tool chain for the alarm system which is described in the following subsections. Our decisions are based on an evaluation of the alarm system which was conducted with a pressure tank system as shown in figure 7. The purpose of this system is to store a pressurized fluid in tank B. An electric powered pump transports the pressurized fluid to the tank. Switch E is responsible for starting the engine by activating relay D and then relay C. The pressure tank system is equipped with two security mechanisms: time relay F and

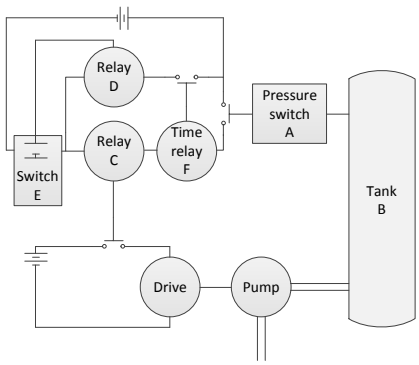

Fig. 7. Setup of the pressure tank system Castillo et al. (1997). The system stores a pressurized fluid in tank B. It consists of eight components including two security mechanisms: time relay F and pressure switch A.

pressure switch A. The system works properly if the pump works for periods less than one minute; thus, time relay F interrupts the current after 60 seconds. If the pressure in the tank exceeds a certain threshold, pressure switch A also interrupts the current. We focused on the failures of the components A, B, C, D, E, and F.

3.1 Evaluation of the Knowledge-Based Approach

Table 3. Conditional probability tables of the pressure tank system Castillo et al. (1997).

D	F	$p(g\|D,F)$
d	f	1
d	$\bar{\text{f}}$	1
$\bar{\text{d}}$	f	1
$\bar{\text{d}}$	$\bar{\text{f}}$	0

E	G	$p(h\|E,G)$
e	g	1
e	$\bar{\text{g}}$	1
$\bar{\text{e}}$	g	1
$\bar{\text{e}}$	$\bar{\text{g}}$	0

A	H	$p(i\|A,H)$
a	h	1
a	$\bar{\text{h}}$	0
$\bar{\text{a}}$	h	0
$\bar{\text{a}}$	$\bar{\text{h}}$	0

C	I	$p(j\|C,I)$
c	i	1
c	$\bar{\text{i}}$	1
$\bar{\text{c}}$	i	1
$\bar{\text{c}}$	$\bar{\text{i}}$	0

B	J	$p(k\|B,J)$
b	j	1
b	$\bar{\text{j}}$	1
$\bar{\text{b}}$	j	1
$\bar{\text{b}}$	$\bar{\text{j}}$	0

We derived an alarm tree from expert knowledge in the system description. The total failure K of the pressure tank system represents the top-event for the succeeding *Fault Tree Analysis (FTA)*. The combination of alarms that lead to alarm k is logically expressed by the equation

$$k = b \vee c \vee (a \wedge e) \vee (a \wedge d) \vee (a \wedge f).$$

To simplify matters, intermediate alarms (g, h, i, and j) are added to the graph, resulting in the alarm tree of Figure 8 where

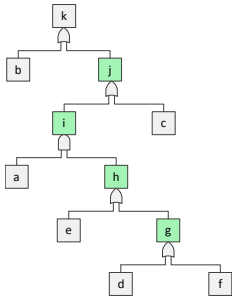

Fig. 8. Extended alarm tree of the pressure tank system Castillo et al. (1997).

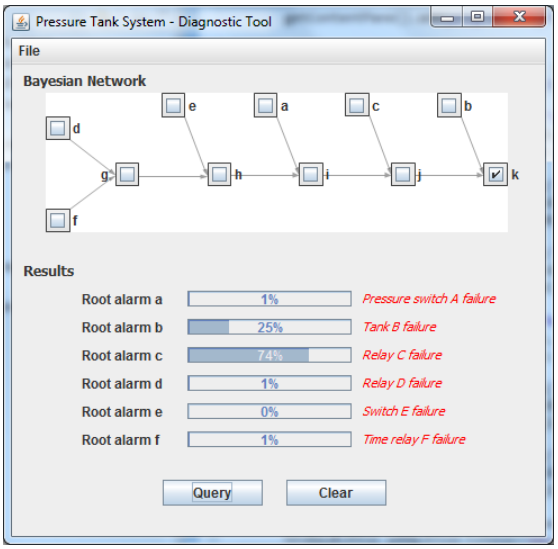

Fig. 9. Simple graphical user interface.

the root causes stay the same. The BN represents the joint probability distribution $P(A, B, C, D, E, F, G, H, I, J, K)$. A particular value in the joint distribution is $P(a, b, c, d, e, f, g, h, i, j, k)$. Applying the chain rule for BN, we get the factorization

$$\begin{aligned} P(a,b,c,d,e,f,g,h,i,j,k) = {} & P(a)P(b)P(c)P(d)P(e)P(f) \\ & P(g|d,f)P(h|e,g)P(i|a,h) \\ & P(j|c,i)P(k|b,j) \end{aligned}$$

with the prior probabilities

$$\begin{aligned} P(a) = 0.002,\ P(b) = 0.001,\ P(c) = 0.003, \\ P(d) = 0.010,\ P(e) = 0.001,\ P(f) = 0.010. \end{aligned}$$

The conditional probability tables are shown in Table 3.

We expressed the BN structure and its parameters using *Prolog* facts with our extended Bratko interpreter. In addition, we created a simple graphical user interface for the pressure tank system shown in Figure 9.

Alarm systems strongly rely on experience of the plant operators. As the knowledge base combines several knowledge sources, it allows accurate and reliable decision-support. However, pursuing a knowledge-based approach for a diagnostic tool has two main weaknesses which are described in the following.

Expert Knowledge as Limiting Factor The two basic prerequisites for modeling, a close communication with the operator and a free access to system documentation, are not a matter of course. Owing to secrecy policies, operating companies are not always willing to give profound plant information to third parties. Even if the knowledge engineer succeeds to establish a close communication with the operator, we have to be aware that the knowledge of operators is limited.

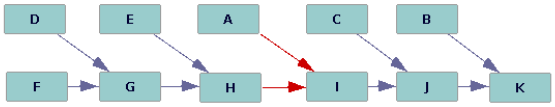

Fig. 10. Learned BN of the pressure tank system. The BN was computed with Tetrad, using a datafile of 10.000 cases. The red marked arcs were not found.

For instance, some causal relations between alarms might be unknown to the operator, like in case of rare (but important) alarms. Plus, the expert might have difficulties to insert probabilities of occurrence of particular combinations of alarms. These circumstances affect the accuracy of modeling.

Time and Effort for Modeling The knowledge engineer has to collect, structure, and model the expert knowledge. All these tasks are very time and effort consuming and increase with the complexity of the industrial plant to be modeled. Furthermore, the models are language dependent and have to be translated to other languages if needed.

3.2 Evaluation of the Machine Learning Approach

With the machine learning approach we can benefit from several advantages: BNs are faster to built by learning, the approach is scalable and language independent.

To demonstrate the machine learning approach, we used a BN of the pressure tank system with the given structure (see Figure **??**) and probabilities (see Table 3) and simulated different cases. We analyzed the resulting data set with Tetrad. Figure 10 shows the resulting network for a dataset with 10.000 cases. Except the two arrows H $\rightarrow$ I and A $\rightarrow$ I, the structure was recovered correctly.

Usage of Expert Knowledge for Construction of Bayesian Networks The result, shown in Figure 10, demonstrates that expert knowledge is still needed after the learning process to adjust the BN. These corrections are necessary, since structural learning algorithms might be erroneous as stated in Section 2.2.1.

Data Preparation For preprocessing and analyzing the raw alarm data, our system requires specific knowledge about the plant. Especially, we have to define the values an alarm variable can take: *Present* and *Absent* or—more detailed and grouped for a functional section— e.g. for heater *ok*, *temperature high*, and *temperature low*. Article Morales-Menéndez et al. (2007) presents an example for a similar application: The causal structure for a cutting tool diagnosis is sought so that the condition of the cutting tool can be classified based on the measured process variables.

Resulting advices Tests with different BNs show that the computation time and the probability of incorrect connections between variables increases with the amount of variables. Therefore, it is critical to construct a network based on several hundreds of variables. As a consequence, we advice to split up

the dataset to group the alarms according to their location of occurrence in different plant sections. In cosequence, several BNs are generated. These BNs can then be used in an object-oriented approach, similar to Weidl and Madsen (2003). Furthermore, the computational costs of reasoning in sub-BNs are smaller than inferring in a network of multiple variables.

4. CONCLUSION

In this paper, we separately investigated a knowledge- and machine learning approach to address the problem of alarm flooding. Main idea was to build an alarm model as Bayesian Network which is used to perform *Alarm Root Cause Analysis* to reduce the alarm flood. The knowledge-based approach ideally provides accurate decision-support for the operator of an industrial plant. However, the necessary expert knowledge is limited. Furthermore, modeling and evaluation are highly time- and effort-consuming. The machine learning approach enables fast modeling and accurate parametrization of alarm dependencies. But the preparation of the alarm data and the evaluation of the computed alarm structure require additional expert knowledge.

To overcome these drawbacks, we recommend a combined procedure based on knowledge and data, shown in Figure **??**. The BN is built using the expert knowledge and alarm data. An ontology is well-suited to store and maintain the expert knowledge. On the one hand, we can derive the BN from the ontology by applying the probabilistic framework *BayesOWL* Ding et al. (2006). On the other hand, we can recover the structure of the alarm system from the prepared data by applying a constraint-based algorithm. The automatic combination of these two approaches towards a BN is left for future work.

REFERENCES

Adnan, N.A., Izadi, I., and Chen, T. (2011). On expected detection delays for alarm dystems with deadbands and delay-timers. *Journal of Process Control*, 21(9), 1318–1331.

Bouckaert, R. (1995). *Bayesian Belief Networks: From Inference to Construction*. Ph.D. thesis, Department of Computer Science, Utrecht university, The Netherlands.

Bratko, I. (2012). *Prolog Programming for Artificial Intelligence*. Pearson Education Limited, 4 edition.

Castillo, E., Gutiérrez, J.M., and Hadi, A.S. (1997). *Expert Systems and Probabilistic Network Models*. Springer-Verlag.

Dash, D. and Druzdzel, M.J. (2003). Robust independence testing for constraint-based learning of causal structure. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, UAI'03, 167–174. Morgan Kaufmann Publishers.

Ding, Z., Peng, Y., and Pan, R. (2006). BayesOWL: Uncertainty modeling in semantic web ontologies. *Soft Computing in Ontologies and Semantic Web*, 3–29.

Gruber, T. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2), 199–220.

Izadi, I., Shah, S., Shook, D., and Chen, T. (2009). An introduction to alarm analysis and design. In *Fault Detection, Supervision and Safety of Technical Processes*, 645–650.

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

Kjaerulff, U.B. and Madsen, A.L. (2008). *Bayesian Networks and Influence Diagrams - A Guide to Construction and Analysis*. Springer.

Kondaveeti, S., Izadi, I., Shah, S., and Black, T. (2010). Graphical Representation of Industrial Alarm Data. In *Proceedings of the 11th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems*.

Koski, T. and Noble, J.M. (2009). *Bayesian Networks: An Introduction*. John Wiley & Sons.

Margaritis, D. (2003). *Learning Bayesian Network Model Structure from Data*. Ph.D. thesis, School of Computer Science - Carnegie Mellon University.

Morales-Menéndez, R., Vallejo, A., Garza, L.E., Cantú, F., and Nebot, J.V.A. (2007). AI approaches for cutting tool diagnosis in machining processes. In *Proceedings of the 25th IASTED International Multi-Conference: artificial intelligence and applications*, 186–191. ACTA Press.

Noy, N. and McGuinness, D. (2001). Ontology development 101: A guide to creating your first ontology.

Pradhan, S., Singh, R., Kachru, K., and Narasimhamurthy, S. (2007). A bayesian network based approach for root-cause-analysis in manufacturing process. In *International Conference on Computational Intelligence and Security*, 10–14. IEEE.

Robinson, R. (1977). Counting unlabeled acyclic digraphs. In *Combinatorial Mathematics V*, volume 622 of *Lecture Notes in Mathematics*, 28–43. Springer.

Rooney, J.J. and Heuvel, L.N.V. (2004). Root cause analysis for beginners. *Quality Progress*, 45–46.

Spirtes, P., Glymour, C., and Scheines, R. (2000). *Causation, prediction, and search*, volume 81. The MIT Press.

Steck, H. (2001). *Constraint-Based Structural Learning in Bayesian Networks using Finite Data Sets*. Ph.D. thesis, Department of Informatics, Technical University Munich.

Weidl, G. and Madsen, A.L. (2003). Object oriented bayesian networks for industrial process operation. In *In Proceedings Workshop on Bayesian modelling, Uncertainty in AI*.