Computer Assisted Mechanics and Engineering Sciences, **12**: 103–110, 2005.

Example of learning Bayesian networks from simulation data

Marcin Bednarski
Silesian University of Technology, Department of Fundamentals of Machine Design,
ul. Konarskiego 18A, 44-100 Gliwice, Poland

(Received March 31, 2005)

Bayesian belief networks represent and process probabilistic knowledge. This representation rigorously describes the knowledge of some domains and it is a human easy-use qualitative structure that facilitates communication between a user and a system incorporating the probabilistic model. Learning Bayesian network from data may be grouped into two modelling situations: *qualitative* learning and *quantitative* learning. The first one consists in establishing the structure of the network, whereas the second concerns determining parameters of the network (conditional probabilities). Both modelling methods were applied on exemplary data to show the possibilities and benefits of this methods. The results and conclusions are presented. It was necessary to preprocess the date first. The used method, described in detail in the paper, consists in discretization into linguistic states on the basis of evaluated signal derivative. Some remarks about adjusting the network, as a part of model identification, are also presented.

Keywords: Bayesian network, learning, diagnostic models

1. Introduction

Bayesian belief networks represent and process probabilistic knowledge. Their are composed of two major elements. The first one is concerned with graph, which is represented by nodes and links between them. This element is qualitative domain of the model. The probability distributions in the form of conditional probability tables (CPT) as a part of the quantitative domain of the model are attached to the nodes of the graph. This representation rigorously describes the knowledge of some domains and it is a human easy-use qualitative structure that facilitates communication between a user and a system incorporating the probabilistic model. These advantages cause that the Bayesian networks has become a popular representation for encoding uncertain expert knowledge in expert systems and they are used frequently in real world applications, including technical diagnosing [8], medical diagnosing [2], financial forecasting [1] and manufacturing control [19]. Bayesian belief networks have been very recently applied to diagnostics of hypothetical accidents in nuclear power plants [11].

The Bayesian network identification task can be divided into two parts according to the above mentioned elements: determination of the structure and determination of the conditional probability tables. The second one is the most important and may be realized by obtaining from different sources such as:

- results of analysis using physical or numerical models (e.g. results of Probabilistic Safety Analysis for nuclear reactors),
- results of experiments or passive observations,
- opinions of domain experts.

The expert opinion is often the only source of information on the discussed probability. It should be stressed that the quality of such data is unknown and validation of the complex Bayesian networks making use of such data is very difficult or even impossible [6]. Alternative method of model identification, which is described in this paper, is learning Bayesian Network from data. The investigations were conducted on an examplary data, which was taken from simulation of nuclear reactors accidents [5].

There is also briefly described a methodology of sensitivity analysis, which can be useful in tunning the network parameters obtained in learning process.

1.1. Bayesian networks

Bayesian networks is a *directed acyclic graph* (DAG) and is always represented visually with a set of nodes and a set of links (Fig. 1). Each node represents a specific variable (e.g. temperature, state of a patient, feature of some object etc.), which must have a finite number of mutually exclusive states (e.g. yes, no, high, low, medium increasing or decreasing etc). Each link represents a relationship between variables, and is depicted with an arrow. Practically, the Bayesian network is typically constructed using notions of cause and effect relation, where conditional probabilities represent the degree of belief in those relation. In most cases, this procedure is acceptable.

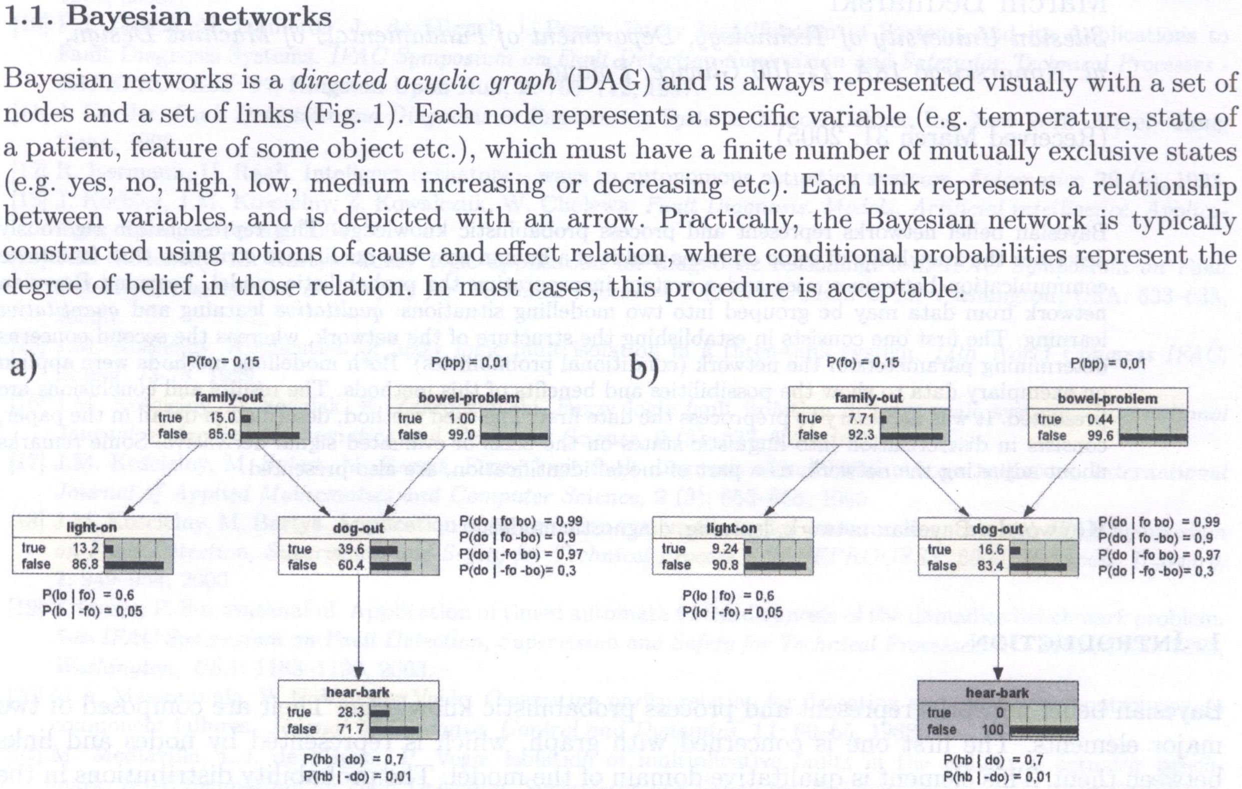

Fig. 1. An example of Bayesian network [9] a) before inference, b) after inference

The representation consists also of a set of local conditional probability distributions combined with a set of assertion of a conditional independence that allow us to construct the global joint distribution from the local distributions. The decomposition is based on the chain rule of the probability Eq. (1). The formal definition is presented below.

$$P(x_1, \ldots, x_n) = \prod_i P(x_i | pa(x_i)) \tag{1}$$

where $pa(x_i)$ is the parent set of x_i.

Definition [14]. *A Bayesian network consist of the following:*

- *A set of variables and a set of directed arcs between variables.*
- *Each variable has a finite set of mutually exclusive states.*
- *The variables together with the directed arcs form a directed acyclic graph (DAG). (A directed graph is acyclic if there is no directed path* $A_1 \to \ldots \to A_n$ *and* $A_1 = A_n$*).*

- *To each variable A with parents $B_1, ..., B_n$, there is attached the potential table $P(A|B_1, ..., B_n)$.*

Bayesian networks are used to perform inference about unknown events (by calculating *a'posteriori* probabilities) on the basis of the achieved information (for instance in the Fig. 1b $P(dog-out = false) = 0.834$ in case of $hear-bark = false$).

2. Identification of Bayesian network-based models

2.1. Structure

The first thing to take into consideration during the identification of Bayesian Network-based models is that its purpose is to give estimates of the certainties (probabilities) for events that are not known (observable or only observable at an unacceptable cost) [13]. Therefore, the first task in the model identification is to recognize these events. They are called *hypothesis events*. The identified events are then grouped into the sets of mutually exclusive events to form *hypothesis variables*. The next step is to identify the types of achievable information that may introduce something about the hypothesis variables. These types of information are grouped into *information variables* in accordance with the rule that the one variable contains only mutually exclusive states. Then, a typical piece of information is a statement that a certain variable is in a particular state, but softer statement are allowed too [14]. Sometimes there is a need to introduce some variables, which are not either the hypothesis variable or the information variable. Such variables are called *hidden variables* and they are used to maintain the conditional independence or make the model identification process easer (modelling tricks).

Having identified the variables for the model, the next thing will be to establish the relations between variables as a links and their directions. In most cases the directed links represents the causality but it is not a rule, because sometimes causal relation is not obvious or it is not correct for the sake of modelling process.

2.2. Determining the conditional probabilities

The Bayesian network parameters, as the conditional probabilities are the crucial elements of the bayesian network-based models and they play an important role in the inferring process. Determining these parameters is the most difficult and controversial problem. In diagnostic expert systems these parameters are very often based on totally subjective estimates of the certain of an event and they are usually assessed by experts from a given domain or bases on passive observations of the identical or similar systems. In many cases the knowledge of experts is inaccurate, contradictory and it is sometimes only based on intuition. Alternative approach is learning bayesian network parameters from data, that can be obtained from both physical or numerical models and databases.

2.3. Model identification from data – learning Bayesian network

Learning Bayesian network from data may be grouped into two modelling situations. The first is called *qualitative* learning and consists in establishing the structure using the database of cases. The second one concerns determining parameters of the network (conditional probabilities) and it is called *quantitative* learning. There is a lot of programs, which is useful for such learning, but a large majority of them are limited only to quantitative learning (e.g. Netica [20] for earlier built structure (e.g. according to Sec. 2.1). Some programs enable full identification of model from data i.e. qualitative and quantitative learning (e.g. Belief Network PowerConstructor [7]). Learning Bayesian network is widely described in many papers. As a starting reference in Bayesian network and learning matter it can be recommended the important book of Finn V. Jensen [13]. Many interesting overviews in this filed also can be found (e.g. [3, 10, 12, 15, 16] etc.).

2.4. Sensitivity analysis

The parameters assigned on knowledge of the experts or determined in the process of learning, may be imprecise. Such a situation is inevitable. Most of the time a very high precision in the parameters is not required, but on the other hand there may be a limited number of parameters with a significant impact on the result. These parameters are called weak points of the model (network). The use of sensitivity analysis makes it possible to find a list of such points (parameters) of the network. Such a list may be useful in the next development steps to tune the network by make more precise assessments of a few identified probabilities.

In case of learning bayesian models it is possible to take advantage of one-way sensitivity analysis, which can be performed in at least two totally different manners. The first simple method consists of varying the considered parameter by small values. Observed simultaneous changes in the selected network output allow calculation of sensitivities [6]. The second more sophisticated method is based on parametric modelling of relations between the considered parameter and the output. The possibility of such parametric modelling was introduced in [21]. The method consist in evaluation of the coefficients of the function representing the relation between the output value (probability of the hypothesis node value) and one selected parameter (conditional probability). Such approach allow determination of the derivative (sensitivity coefficient) for the given value of the input parameter. Unfortunately, the results of such sensitivity analysis depend strongly on a context or case, i.e. current states of all observable nodes, which results in large expenditure on the calculation. It is possible to reduce it by application of so-called sensitivity set. The sensitivity set is a set of all those, and only those, nodes whose variation of conditional and a'priori probabilities may affect the network output. The practical application of this method is presented in [4].

3. Examples of learning Bayesian network

Model identification from data will be presented for data obtained from simulation model (MAAP4 – Modular Accident Analysis Program [18]) of nuclear reactor (Boiling Water Reactor). For the purpose of learning Bayesian network two programs: Netica [20] (*quantitative* learning) and Belief Network PowerConstructor [7] (*qualitative* and *quantitative* learning) were applied. The learning examples was prepared on the basis of available data [5]. The data contain results of simulation for major accidents. The accident scenarios were considered as an element of Cartesian product of the following elementary states: water-pipe failure (small, medium, large), steam-pipe failure (small, medium, large), leakage between drywell and wetwell (small, medium, large), failure of safety systems (operate, non-operate). The following signals were the output from the simulation model: pressure in reactor containment, temperature in reactor containment, hydrogen concentration in reactor containment, temperature of water in wetwell, pressure in wetwell, water level in wetwell, pressure in reactor vessel, water level in reactor vessel.

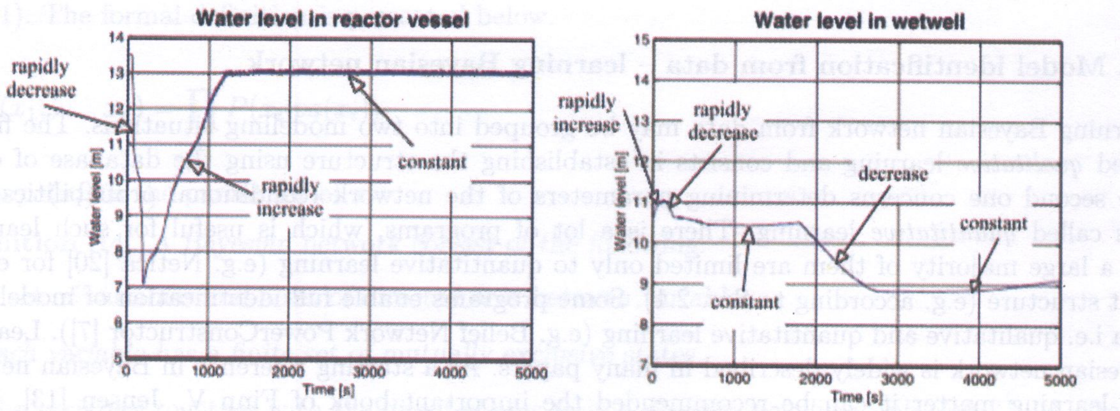

Fig. 2. Preparation of the learning data

It was necessary to discretizing the data (output signals) (Fig. 2) to learn Bayesian networks. It was performed through breaking up the total range of the signals derivative into a number of subranges. The appropriate numbers showing where one subrange ends and the next begins were assumed (Fig. 3). Each subrange corresponds to one state of the discrete version of the variable. All was performed in Matlab enviroment [17]. The Table 1 shows selected learning examples for different failure.

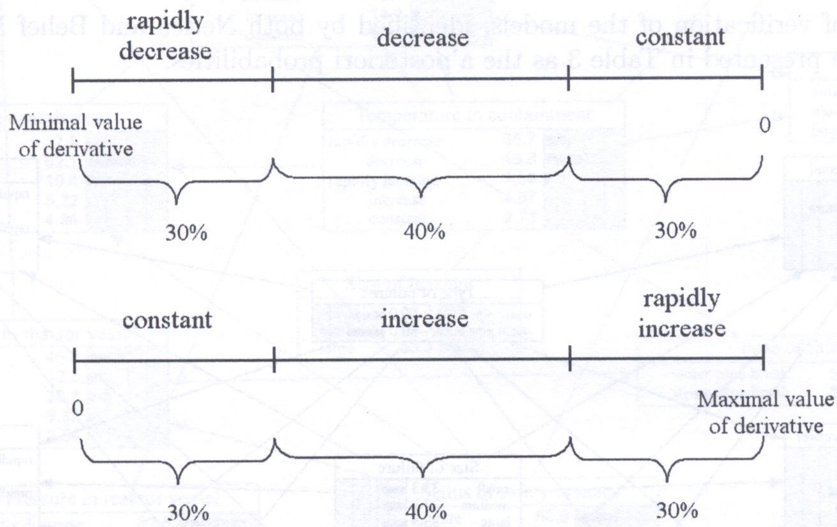

Fig. 3. Discretization of the data on the basis of the derivative value

Table 1. Selected learning examples

Failure				Consider features							
Type of failure	Size of failure	Size of leakage	Status of safety systems	Pressure in reactor vessel	Water level in reactor v.	Pressure in containment	Temp. in cont.	H_2 concent.	Pressure in wetwell	Temp. in wetwell	Water level in wetwell
Water pipe break	Large	Small	Operate	rapidly decrease	increase	increase	increase	constant	increase	rapidly increase	rapidly increase
Water pipe break	Medium	Small	Operate	decrease	decrease	decrease	decrease	constant	decrease	increase	decrease
Water pipe break	Medium	Small	Non-operate	rapidly decrease	decrease	increase	increase	rapidly decrease	increase	rapidly increase	increase
Water pipe break	Large	Small	Operate	decrease	rapidly decrease	decrease	decrease	constant	increase	increase	decrease
Steam pipe break	Small	Small	Operate	decrease	constant	constant	increase	constant	increase	increase	constant
Steam pipe break	Medium	Medium	Non-operate	decrease	decrease	increase	increase	decrease	increase	increase	increase
Water pipe break	Medium	Large	Operate	rapidly decrease	decrease	decrease	decrease	constant	increase	increase	decrease
Water pipe break	Small	Large	Non-operate	rapidly decrease	decrease	constant	increase	decrease	increase	rapidly increase	increase
Steam pipe break	Small	Large	Operate	decrease	constant	decrease	rapidly decrease	constant	constant	decrease	decrease

3.1. Results

In case of Netica program, which allows only to quantitative learning (determining conditional probabilities), the structure presented in Fig. 4 was assumed. The failures were established as the hypothesis variables (type of failure, size of failure, size of leakage and status of safety systems). The information variables were assumed as the symptoms of the failure (pressure in the reactor vessel, water level in the reactor vessel, pressure in the containment, temperature in the containment, H_2

concentration in the wetwell, temperature in the wetwell, water level in the wetwell) The links between nodes were modelled as cause-effect relations i.e. from failures to symptoms.

In case of Belief Network PowerConstructor, the structure obtained as a result of qualitative learning is presented in Fig. 5.

From set of 288 learning examples, 4 testing examples were chosen randomly in order to verify obtained models. The values of the symptoms for them are presented in Table 2.

The results of verification of the models, identified by both Netica and Belief Network PowerConstructor, are presented in Table 3 as the a'posteriori probabilities.

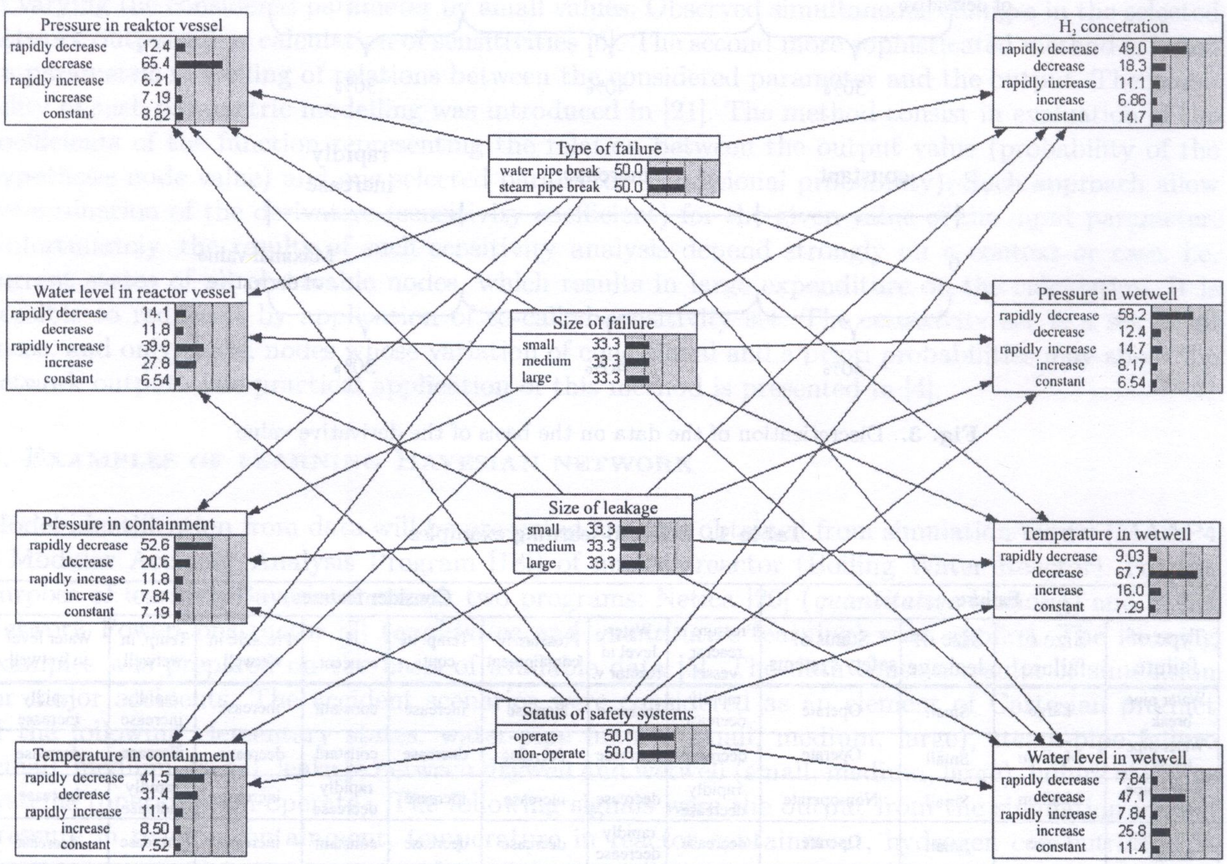

Fig. 4. The structure of Bayesian network for Netica quantitative learning

Table 2. The testing examples

Consider features							
Pressure in react. vessel	Water level in react. vessel	Pressure in containment	Temperature in containment	H_2concentartion	Pressure in wetwell	Temperature in wetwell	Water level in wetwell
decrease	constant	constant	increase	constant	increase	increase	constant
decrease	decrease	increase	decrease	decrease	increase	increase	increase
decrease	decrease	increase	decrease	rapidly decrease	increase	increase	increase
decrease	decrease	decrease	decrease	constant	increase	increase	increase

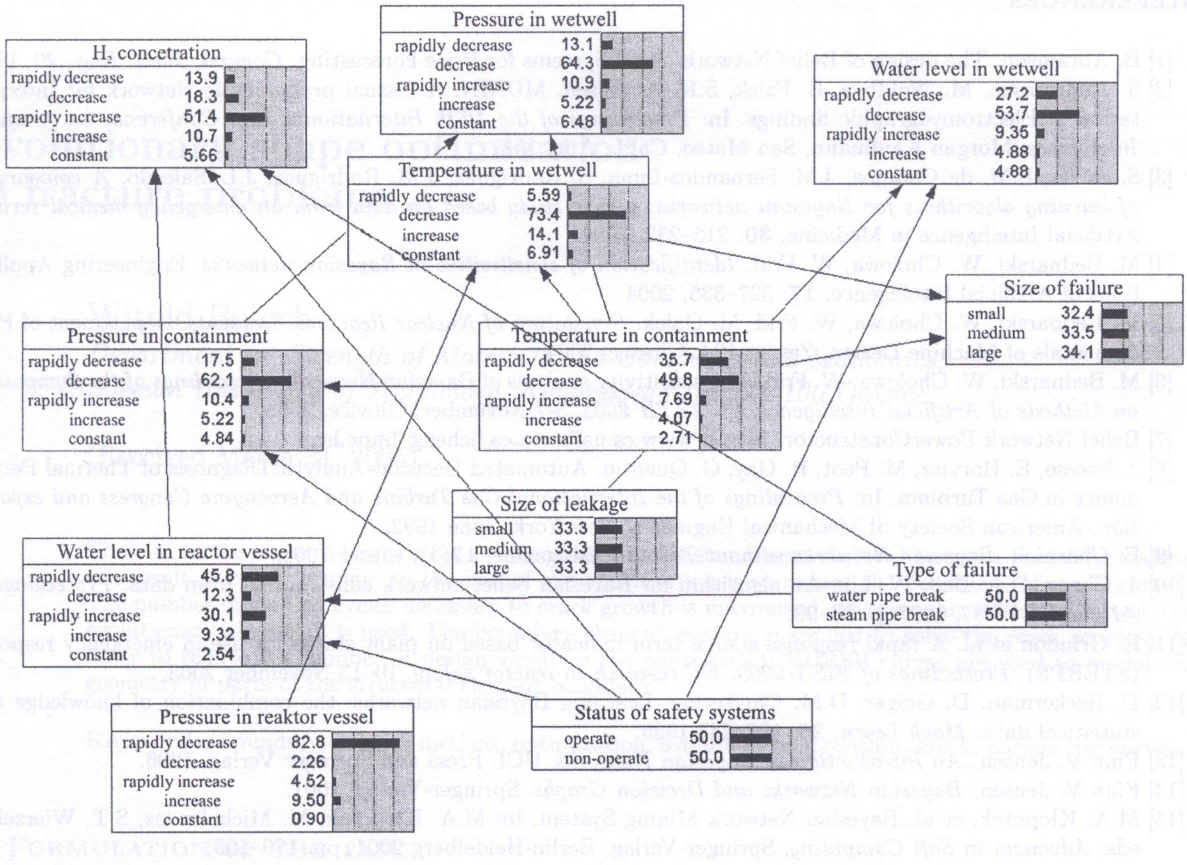

Fig. 5. The structure of Bayesian network as a result from Belief Network PowerConstructor

Table 3. The results of the models verification

Failure											
Type of failure	Probability Netica	Probability BN PC	Size of failure	Probability Netica	Probability BN PC	Size of leakage	Probability Netica	Probability BN PC	Status of safety systems	Probability Netica	Probability BN PC
Steam pipe break	**0.99**	**0.99**	small	**0.82**	**0.48**	small	**0.58**	**0.33**	operate	**0.91**	**0.71**
Steam pipe break	**0.75**	**0.62**	medium	**0.63**	**0.43**	small	**0.52**	**0.40**	non-operate	**0.93**	**0.98**
Water pipe break	**0.84**	**0.93**	medium	**0.75**	**0.43**	small	**0.53**	**0.40**	non-operate	**0.92**	**0.98**
Water pipe break	**0.73**	**0.88**	medium	**0.57**	**0.43**	small	**0.46**	**0.40**	operate	**0.94**	**0.99**

3.2. Conclusions

The obtained results are satisfactory. The errors (small values of probability) may be caused by small number of learning examples and/or reduced accuracy affected by earlier discretization and change of the quantitative values to the qualitative values.

It should be pointed out that the testing set is small and it is selected directly from the learning data. Obtained networks are probably not precise enough and final models should be more carefully tested and probably adjust. It can be done for instance with the use of sensitivity analysis.

On the basis of these results one may also conclude that accuracy of model, obtained in qualitative and quantitative learning process (learning the structures and parameters – Belief Network PowerConstructor), is considerable smaller. It comes from the fact that the qualitative learning is new and complex matter and existing methods are experimental and still evolving.

References

[1] B. Abramson. The design of Belief Network-based Systems for Price Forecasting. *Comput. Elect. Eng.*, 20, 1994.

[2] S. Andreassen, M. Woldbye, B. Falck, S.K. Andersen. MUNIN: A Casual probabilistic Network for Interpretation of Electromyographic findings. In: *Proceedings of the 10-th International Join Conference on Artificial Inteligence*, Morgan Kaufmann, San Mateo, Calif., Aug. 1987.

[3] S. Acida, L.M. de Campos, J.M. Fernandez-Luna, S. Rodriguez, J.M. Rodriguez, J.L. Salcedo: *A comparison of learning algorithms for Bayesian networks: a case study based on data from an emergency medical service.* Artificial Intelligence in Medicine, **30**: 215–232, 2004

[4] M. Bednarski, W. Cholewa, W. Frid. *Identification of sensitivities in Bayesian networks.* Engineering Applications of Artificial Intelligence, **17**: 327–335, 2004

[5] M. Bednarski, W. Cholewa, W. Frid, M. Galek. *Simulation of Nuclear Reactors Accidents.* Department of Fundamentals of Machine Design, Zeszyt 124, Gliwice, 2004.

[6] M. Bednarski, W. Cholewa, W. Frid. The sensitivity analysis of Bayesian Networks. *Procedings of the Symposium on Methods of Artificial Intelligence AI-METH 2003,* 5-7 November, Gliwice, 2003.

[7] Belief Network PowerConstructor. http://www.cs.ualberta.ca/jcheng/bnpc.htm.

[8] J. Breese, E. Horvitz, M. Peot, R. Gay, G. Quentin. Automated Decision-Analytic Diagnosis of Thermal Performance in Gas Turbines. In: *Proceedings of the International Gas Turbine and Aeroengine Congress and exposition,* American Society of Mechanical Engineers, New York, June 1992.

[9] E. Charniak. *Bayesian Networks without Tears.* AI Magazine, **12**(4), winter 1991.

[10] J. Cheng, D.A. Bell, W. Liu. An algorithm for Bayesian belief network construction from data. *Proceedings of AI and STAT'97,* 1997. p. 83–90.

[11] E. Grindon et al. A rapid response source term indicator based on plant status for use in emergency response (STERPS). *Proceedings of FISA-2003, EU research in reactor safety,* 10–13 November 2003.

[12] D. Heckerman, D. Geiger, D.M. Chickering. Learning Bayesian networks: the combination of knowledge and statistical data. *Mach Learn,* **20**: 197–243 1995.

[13] Finn V. Jensen. *An Introduction to Bayesian Networks.* UCL Press and Springer Verlag, 1996.

[14] Finn V. Jensen. *Bayesian Networks and Decision Graphs.* Springer-Verlag, 2001.

[15] M.A. Kłopotek, et al. Bayesian Network Mining System. In: M.A. Kłopotek, M. Michalewicz, S.T. Wierzchoń eds. *Advances in Soft Computing,* Springer-Verlag, Berlin-Heidelberg 2001, pp. 179–193,

[16] M.A. Kłopotek, S.T. Wierzchoń. Bayesian Nets for Recommender Systems. In: M.A. Kłopotek, S.T. Wierzchoń, K. Trojanowski, eds. *Advances in Soft Computing,* Springer-Verlag, Berlin-Heidelberg 2003, pp. 87–96.

[17] The MathWorks, http://www.mathworks.com/.

[18] Modular Accident Analysis Program MAAP4: www.maap4.com.

[19] F. Nadi, A. Agogino, D. Hodges. Use of Influence Diagrams and Neural Networks in Modeling Semiconductor Manufacturing Processes. *IEEE Trans. on Semicond. Mfg.*, 4, 1991.

[20] Norsys Software Corp. http://www.norsys.com.

[21] Veerle M.H. Coupe, Linda C. van der Gaag. *Properties of sensitivity analysis of Bayesian belief networks.* Annals of Mathematics and Artificial Intelligence, **36**: 323-356, 2002.