Reliability Engineering and System Safety 94 (2009) 542–550

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

A generic method for estimating system reliability using Bayesian networks

Ozge Doguc, Jose Emmanuel Ramirez-Marquez *

Stevens Institute of Technology, Hoboken, NJ 07030, USA

ARTICLE INFO

Article history:
Received 6 August 2007
Received in revised form
26 March 2008
Accepted 14 June 2008
Available online 19 June 2008

Keywords:
System reliability
Reliability methods
Reliability analysis
Reliability modeling

ABSTRACT

This study presents a holistic method for constructing a Bayesian network (BN) model for estimating system reliability. BN is a probabilistic approach that is used to model and predict the behavior of a system based on observed stochastic events. The BN model is a directed acyclic graph (DAG) where the nodes represent system components and arcs represent relationships among them. Although recent studies on using BN for estimating system reliability have been proposed, they are based on the assumption that a pre-built BN has been designed to represent the system. In these studies, the task of building the BN is typically left to a group of specialists who are BN and domain experts. The BN experts should learn about the domain before building the BN, which is generally very time consuming and may lead to incorrect deductions. As there are no existing studies to eliminate the need for a human expert in the process of system reliability estimation, this paper introduces a method that uses historical data about the system to be modeled as a BN and provides efficient techniques for automated construction of the BN model, and hence estimation of the system reliability. In this respect $K2$, a data mining algorithm, is used for finding associations between system components, and thus building the BN model. This algorithm uses a heuristic to provide efficient and accurate results while searching for associations. Moreover, no human intervention is necessary during the process of BN construction and reliability estimation. The paper provides a step-by-step illustration of the method and evaluation of the approach with literature case examples.

1. Introduction

System reliability can be defined as the probability that a system will perform its intended function for a specified period of time under stated conditions [1]. Estimating system reliability is an important and challenging problem for system engineers. It is important because a company's reputation, customer satisfaction and system design costs can be directly related to the failures experienced by the system [2]. It is also challenging since current estimation techniques require a high level of background in system reliability analysis, and thus familiarity with the system.

Traditionally, engineers estimate reliability by understanding how the different components in a system interact to guarantee system success. Typically, based on this understanding, a graphical model (usually in the form of a fault tree, a reliability block diagram or a network graph) is proposed to represent how component interaction affects system functioning. Once the graphical model is obtained, different analysis methods [3–5] (minimal cut sets, minimal path sets, Boolean truth tables, etc.) can be used to quantitatively represent system reliability. Finally, the reliability characteristics of the components in the system are introduced into the mathematical representation in order to obtain a system-level reliability estimate.

This traditional perspective aims to provide accurate predictions about the system reliability using historical or test data. This approach is valid whenever the system success or failure behavior is well understood. However, for new complex systems, both in the design phase or already deployed, understanding component interaction may prove to be a challenging problem, which usually requires intervention of a domain expert.

To address these challenges, Bayesian networks (BNs) have been proposed as an alternative to traditional reliability estimation approaches [1,6,7]. In this respect, Barlow [8] and Almond [9] introduced some of the earliest studies for reliability estimation via BN. Barlow [8] provided a case study on spherical pressure vessels by comparing Bayesian and non-Bayesian approaches to estimate system reliability, while Almond [9] introduced the graphical-belief environment, which is used for risk prediction in large complex systems. Based on these studies, it became clear that for some systems, BN had significant advantages—in terms of efficiency in evaluating associations and simplicity in providing a system model—over traditional reliability evaluation frameworks,

Abbreviations: BN, Bayesian network; CPT, conditional probability table; $K2$, named after Kutató 2; HRP, Halden project

* Corresponding author.
E-mail addresses: jmarquez@stevens.edu, Jose.Ramirez-Marquez@stevens.edu (J.E. Ramirez-Marquez).

0951-8320/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ress.2008.06.009

Nomenclature

$p(A)$	probability of an event A
$p(H\|E)$	probability of event H given the evidence E
X_i	node in BN representing component i in the system
u	upper bound on the number of parents of a node in a BN
n	number of components in a system
F	scoring function used in the $K2$ algorithm
π_i	set of candidate parents for X_i
j	a parent in π_i
k	an instantiation of parent X_j
α_{ijk}	number of observations on component X_i, where both component X_i and parent X_j are instantiated as k
q_i	total number of unique parent set instantiations for node X_i
d_i	number of instantiations that node X_i can have
α_{ij}	sum of α_{ijk}'s for component X_i and its parent X_j
$p(X_i = 1)$	probability of node $X_i = 1$
t	number of observations on a system
A_{I}	number of incorrect associations decided by the $K2$ algorithm in a BN
A_{FP}	number of incorrect associations decided by the $K2$ algorithm in a BN
A_{FN}	number of associations in a BN that cannot be discovered by the $K2$ algorithm
A_{T}	total number of associations in a BN
ρ	error rate of the $K2$ algorithm

due to their ease of use in interaction with domain experts in the reliability field [10].

From a reliability perspective, a BN can be represented as a directed graph with (1) nodes that represent system components and (2) edges that show relationships among the nodes. To evaluate the BN, via Bayes' rule, edges within the graph are assigned with a value that shows the degree (or strength) of the relationship it represents—a detailed description of BN will be provided in Section 3. Currently, approaches for system reliability analysis via a BN [6,10–13] use specialized networks designed for a specific system. That is, the BN structure (i.e. the nodes, their associations and the strength of these associations) used for estimating system reliability must be known *a priori*. Inherently, this assumption supposes that an expert with "adequate" knowledge about the system behavior can build the BN. However, finding such an expert may not be possible at all times for every system under consideration. The number of such experts is limited, and finding one is usually difficult and costly [14]. Moreover, human intervention is always open to unintentional mistakes, which could cause discrepancy in the results.

These issues are particularly true in complex systems, where the number of components and interactions is large, and thus the likelihood of miscalculations can be substantial. As a result, such an assumption may not be realistic enough to be applicable. Additionally, modern systems evolve with time by adding and removing new and obsolete components [15]. Thus, the original BN model may not be accurate throughout a system life cycle, forcing human intervention to be needed right after every change in the system. Therefore, in the applications where an expert builds the BN, there is always the need for keeping that same expert as the system evolves.

To address these issues, this study introduces a holistic method for estimating system reliability. The main contribution of this paper is to relieve the need of an expert by linking BN construction from raw component and system data. In essence, the method replaces the need of an expert to find associations among the components with raw data related to component and system behavior. These data are then used to develop (via association rule mining) and evaluate (via conditional probabilities and Baye's theorem) a BN that describes the relationships and interactions of components with system success behavior. Based on the extensive literature review, this is the first study that incorporates these methods for estimating system reliability by eliminating the need for human intervention in BN construction.

The proposed method automates the process of BN construction by feeding raw system behavior data to the $K2$ algorithm (a commonly used association rule mining algorithm). This algorithm is a machine-learning algorithm that uses canonically ordered sets of variables and identifies the associations among them by using a predefined scoring function and a heuristic. The $K2$ algorithm has proven to be efficient and accurate for finding associations [16] from a dataset of historical data about the system. The $K2$ algorithm reduces the algorithmic complexity of finding associations from exponential (2^n) to quadratic (n^2) [16] with respect to the number of components (n) in the system. Therefore, the method proposed in this paper is efficient for complex systems with large datasets. Moreover, unlike previous approaches, the proposed solution is not system specific, it can be applied to systems following any kind of configuration (two terminal, k-terminal, all terminal, etc.) and behavior (binary, capacitated and multi-state) without the need for an expert to assess and identify component relationships. That is, the approach can build a BN and estimate reliability for any system when observed system failure data are available.

This paper is organized as follows: Section 2 gives a brief summary about related work in the literature. Section 3 provides information about BNs and the $K2$ algorithm, while Section 4 presents a comprehensive illustration of BN construction and system reliability estimation techniques. In Section 5 experimental analysis on the accuracy and performance of the methodology is provided. Finally, in Section 6 conclusions and future research directives are provided.

2. Literature survey

Estimation of systems reliability using BN dates back as early as 1988, when it was first defined in [8]. The idea of using BN in systems reliability has mainly gained acceptance because of the simplicity it allows to represent systems and the efficiency for obtaining component associations. The concept of BN has been discussed in several earlier studies [17–19]. More recently, BN have found applications in software reliability [20,21], fault finding systems [18] and general reliability modeling [11].

Currently, predefined BN are used for reliability estimation for specific systems. Gran and Helminen [1] provide a BN for nuclear power plants and introduce a hybrid method for estimating the reliability of the plant. The nuclear plant is composed of a software system and the plant hardware; therefore, they combined two BNs that were currently being used for corresponding systems: (1) the Halden project (HRP) [22] uses a BN for risk assessment based on disparate evidences. (2) The VTT automation [23] focuses on the reliability of software-based systems using BN. Moreover, there is another challenge: each BN uses a different modeling and simulation environment. The HRP uses HUGIN [24] and SERENE [25], which uses conditional probability table (CPT) in their assessments. However, VTT utilizes WinBUGS environment [26] which is based on continuous and discrete distributions and sampling from these distributions [1]. In another study that uses BN, Helminen and Pulkkinen [27] present a BN-based method for reliability estimation of computer-based motor protection relay. They assume the existence of a BN that models the system and introduce methods for estimating prior probabilities and assessing the system reliability accordingly.

In addition to these, Amasaki et al. [6] uses BN for software quality assessment. They modeled the phases of a software system as a BN, and by using this model they observed the faults that may occur in their system. After this step, they used the actual data and created sensitivity analysis of the BN model that they constructed. Alternatively, Boudali and Dugan [7] introduce a method for reliability assessment in dynamic systems by using temporal BN; where the system components change states at different time intervals.

Singh et al. presents their work on reliability estimation in component-based systems. They classify the component-based system reliability estimation methods into three as state-based models, path-based models and additive models. State-based models use *control graphs* [28,29] in order to represent the system architecture, which is either previously known or can be generated at run time [30]. On the other hand, path-based models consider all possible execution paths that the system can follow. Again the execution paths may either be available already [2] or should be identified during system design phase [31]. Finally, additive models [32] use component failure data focusing on growth modeling. They introduce their reliability estimation, which models the system using unified modeling language and provide use case scenarios.

Although all of the studies introduced in this section use BN for reliability estimation, they require human domain experts to evaluate the prior probabilities and understand the structure of the BN.

3. Bayesian networks and the *K*2 algorithm

In this section a general overview of BN and its construction via the *K*2 algorithm is given. In Section 3.1, BN and Bayes' rule are introduced. This is followed by a concise description of the *K*2 algorithm with a heuristic function, which is used for constructing the BN used in this study.

3.1. Bayesian networks in system reliability

One could summarize the BN as an approach that represents the interactions among the components in a system from a probabilistic perspective. This representation is performed via a directed acyclic graph, where the nodes represent the variables and the links between each pair of nodes represent the causal relationships between the variables. From a system reliability perspective, the variables of a BN are defined as the components in the system, while the links represent the interaction of the components leading to system "success" or "failure". A fundamental assumption for the construction of a BN is that in general, the strength of the interaction/influence among the graph nodes is uncertain, and thus this uncertainty is represented by assigning a probability of existence to each of the links joining the different nodes. These probabilities conform to three basic axioms:

1. $p(A)$, the probability of event A, is a number between 0 and 1.
2. $p(A)=0$ defines A as an impossible event; $p(A)=1$ defines A as a certain event.
3. $p(A \text{ or } B)=p(A)+p(B)$, provided A and B are disjoint [20].

Under a reliability analysis perspective, an event A constitutes the success of a specific system component, and therefore $p(A)$ represents the probability of success for such a component. For non-trivial systems—systems not following a series, parallel or any combination of these configurations—the failure/success probability of a system is usually dependent on the failure/success of a non-evident collection of components. Strictly speaking, the probability of an event or hypothesis is conditional on the available evidence on current context. This can be made explicit by the notation $p(H|E)$, which is read as "the probability of event H given the evidence E". From a reliability perspective, this notation can be interpreted as the success probability of component H, given the evidence of component E being successful.

In a BN this dependency is represented as a directed link between two components, forming a *child* and *parent* relationship, so that the dependent component is called as the *child* of the other. Therefore, the success probability of a child node is *conditional* on the success probabilities associated with each of its parents [20]. The *conditional probabilities* of the child nodes are calculated by using the Bayes' theorem via the probability values assigned to the parent nodes. Also, the absence of a link between any two nodes of a BN indicates that these components do not interact for system failure/success, thus they are considered *independent* of each other and their probabilities are calculated separately. As will be discussed in Section 4.2 in detail, calculations for the independent nodes are skipped during the process of system reliability estimation, reducing the total amount of computational work.

BNs are known to be useful in assessing the probabilistic relationships and identifying probabilistic mappings between system components [33]. The components are assigned with individual CPT within the BN. The CPT of a given component X contains $p(X|S)$, where S is the set of X's parents. In X's CPT all of its parents are instantiated as either "Success" or "Failure"; so for m parents there are 2^m different parent set instantiations, thus 2^m entries in CPT. The BN is complete when all the conditional probabilities are calculated and represented in the model. However, the sizes of CPT can be substantially large in complex systems where components may have numerous parents [34].

To illustrate these concepts, the BN shown in Fig. 1 presents an expert's perspective on how the five components of a system interact. For this BN the child–parent relationships of the components can be observed, where on the quantitative side [14] the *degrees* of these relationships (associations) are expressed as probabilities and each node is associated with a CPT whose size grows exponentially with the number of parents.

In Fig. 1 the topmost nodes (X_1, X_2 and X_4, components 1, 2 and 4, respectively) do not have any incoming edges; therefore, they are conditionally *independent* of the rest of the components in the

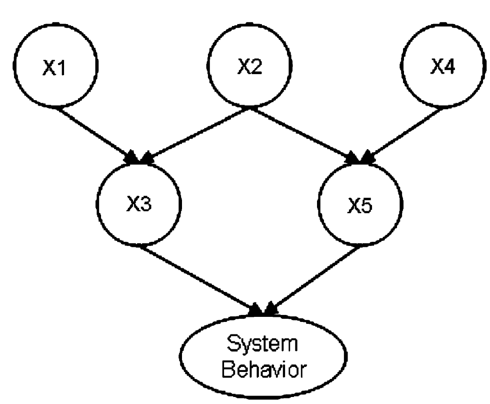

Fig. 1. A sample Bayesian network.

system. The *prior probabilities* that are assigned to these nodes should be known beforehand—with the help of a domain expert or using historical data about the system. Based on these prior probabilities, the CPT that belong to a *dependent* node, such as X_3, can be calculated using Bayes theorem as

$$p(X_3|X_1,X_2) = \frac{p(X_1,X_2|X_3)p(X_3)}{p(X_1,X_2)} \tag{1}$$

Eq. (1) shows that the probability for the node X_3 is *independent* of nodes other than X_1 and X_2 in the system. As a result of this property the total number of computations done for calculating this probability is reduced from 2^n (where n is the number of nodes in the network) to 2^m, where m is the number of parents for a node (and $m \ll n$). Similar to prior probabilities, CPT can also be computed by using historical data of the system behavior. However, an important question on how to discover the associations among the system components still remains. As an alternative to using a domain expert for this purpose, a non-supervised BN construction algorithm, $K2$ is used in this paper.

3.2. K2 Bayesian network construction algorithm

The $K2$ algorithm, for construction of a BN, was first defined in [16] as a greedy heuristic search method. This algorithm searches for the parent set for a node that has the maximum association with it. The $K2$ algorithm is composed of two main factors: a scoring function to quantify the associations and rank the parent sets according to their scores, and a heuristic to reduce the search space to find the parent set with the highest degree of association. Without the heuristic, the $K2$ algorithm would need to examine all possible parent sets, i.e. starting from the empty set, it should consider all subsets. Even with a restriction on the maximum number of parents ($u<m$), the search space would be as large as 2^u (total number of subsets of a set of size u), which requires an exponential-time search algorithm to find the most optimal parent set. With the heuristic, the $K2$ algorithm does not need to consider the whole search space; it starts with the assumption that the node has no parents and adds incrementally that parent whose addition most increases the scoring function. When addition of no single parent can increase the score, the $K2$ algorithm stops adding parents to the node. Using the heuristic reduces the size of the search space from exponential to quadratic. The pseudo-code of the $K2$ algorithm [35] is given below:

Algorithm $K2$ (T, u)
Input: A dataset T of historical observations on system S, an upper bound u for the number of parents
Output: A full BN B.

1. For each column i in dataset T
   - Create node X_i and add it to B.
   - $\pi_i = \phi$ for node X_i.
   - Calculate $f(i, \pi_i)$ using empty set ϕ
   - While the size of $\pi_i \leqslant u$
     - Let X_z be a node preceding node X_i
     - Calculate $f(i, \pi_i\ U\ \{X_z\})$ using X_z
     - If the new score is better than the previous score
       – Add X_z to π_i permanently
2. Return B

The $K2$ algorithm uses f scoring function to decide the best candidate parent set for node X_i. The f scoring function is provided as

$$f(i,\pi_i) = \prod_{j=1}^{q_i} \frac{(d_i-1)!}{(\alpha_{ij}+d_i-1)!} \prod_{k=1}^{d_i} \alpha_{ijk}! \tag{2}$$

According to the $K2$ algorithm for every node X_i, an f score is calculated for each of the candidate parent sets π_i. The scoring function iterates for each candidate parent X_j in π_i and calculates the number of observations in which both node X_i and its candidate parent X_j are in the same state k (Success or Failure), which is denoted as α_{ijk} in Eq. (2). In Eq. (2), q_i is the number of unique parent set instantiations. It can be observed that q_i grows exponentially (2^{π_i}) with the size of the parent set π_i, as its value equals to the number of subsets of the parent set. Thus, in order to reduce (or bound) the number of computations in large parent sets of size m, an upper bound u is usually defined for the number of possible parents as a parameter to the $K2$ algorithm by the systems engineer in application. Finally, d_i represents the number of states that the node X_i can be in.

4. Illustrative step-by-step example

This section provides a step-by-step explanation of the BN construction framework detailed in Section 3. Table 1 presents an example dataset that contains observations on a sample system with five components labeled X_1–X_5. Each row in Table 1 shows the state of the components at an instance of time t_i, when the observation was done. For the sake of simplicity and without loss of generality in the proposed method, component failure data follow a binary behavior. That is, for each component X_i, the value of 0 represents failure, while the value of 1 represents full functionality. Also, in Table 1, information about the overall *System Behavior* is provided in the last column.

The proposed method uses a dataset as displayed in Table 1, finds associations between columns (system components), calculates the degrees of these associations, builds the associated BN and finally uses it to estimate overall system reliability. In the next section, a detailed illustration of how the $K2$ algorithm constructs a BN using the dataset displayed in Table 1 is provided. Then in Section 4.2 the method of estimating system reliability is demonstrated using the constructed BN.

4.1. Execution of the K2 algorithm

As previously explained, output of the $K2$ algorithm is dependent on the initial linear ordering of the columns in the dataset. For this example, columns have been canonically ordered and the $K2$ algorithm starts with the first component in this ordering, X_1. As X_1 does not have any succeeding components, i.e. possible candidate parents, the $K2$ algorithm skips it and picks the second component in the dataset, which is X_2.

For X_2, there may be two alternative parent sets: the empty set ϕ or X_1. Therefore, the $K2$ algorithm computes the scoring function f using each of these alternative parent sets and compares the results. At the end of this iteration, the set of candidate parents with the highest f score will be chosen as the parent set for X_2.

Table 1
Dataset for the illustrative example

Observation	X_1	X_2	X_3	X_4	X_5	System behavior
1	1	1	0	0	0	0
2	0	1	1	1	1	1
3	1	1	0	1	1	1
4	0	0	0	0	0	0
5	1	0	1	0	0	1
6	0	0	0	0	0	0
7	1	1	0	1	1	1
8	0	1	1	1	1	1
9	1	0	1	0	0	1
10	0	0	0	0	0	0

In order to compute the f scores associated with this selection, the following values need to be calculated: α_{ijk}, α_{ij}, d_i and q_i, where i represents the index of the current node, X_2. α_{ijk} is the total number of rows in which both the parent set j and node X_i are instantiated as k (Failure = 0 or Success = 1). α_{ij} is simply the sum of α_{ijk}'s for different instantiations:

$$\alpha_{ij} = \sum_{k=1}^{d_i} \alpha_{ijk} \tag{3}$$

As discussed, d_i is the size of the input domain, which is binary for this example and therefore has a value 2. And finally, q_i is the number of subsets of the parent set for node X_2, which is 2 for X_2 (empty set and X_1).

Based on the data displayed in Table 1, for $\pi_i = \phi$, the f score is calculated as follows: because the parent set is empty, there are no unique instantiations and therefore α_{ijk}'s become α_{201} and α_{202} ($i = 2$ for X_2, $j = 0$ for the empty set and $k = 1$ and 2 representing 0 and 1 in the raw data, respectively), where they show the total number of values equal to zero and one in the X_2 column, respectively.

Using $\alpha_{201} = 5$ and $\alpha_{202} = 5$ in the formula:

$$f(X_2, \phi) = \frac{(2-1)!}{(10+2-1)!} \prod_{k=1}^{2} \alpha_{i0k}! = \frac{1!}{11!} \times 5! \times 5! = \frac{1}{2772} \tag{4}$$

The remaining candidate parent set is $\{X_1\}$, and the f score for this set is calculated as follows, using the values from Table 2.

Based on these values, the f scoring function yields:

$$f(X_2, \{X_1\}) = \frac{(2-1)!}{(5+2-1)!} \prod_{k=1}^{2} \alpha_{21k}! \times \frac{(2-1)!}{(5+2-1)!} \prod_{k=1}^{2} \alpha_{22k}! = \frac{1}{3600} \tag{5}$$

At the end of this iteration the values 1/2772 and 1/3600 are compared with the former, representing the score of the empty set $\{\phi\}$, picked as the parent. So, the $K2$ algorithm decides that X_2 has no parents. Hence, the data does not support any association between X_1 and X_2).

In the next iterations of the $K2$ algorithm, the number of possible candidate parent sets to be considered and thus the amount of computations for f score calculation increases. Skipping the details, f scores of the candidate parent sets for the X_3 component are given in Table 3.

Because the $K2$ algorithm iterates on the components according to the initial linear ordering, components X_4 and X_5 are not taken into account as candidate parents for X_3. The $K2$ algorithm selects the set $\{X_1, X_2\}$ as the parent set of X_3, because it has the highest f score. The number of computations grows with the order of the component; therefore, the rest of the computation done in the next steps of the $K2$ algorithm is skipped here. When the $K2$ algorithm finishes with the last column (System Behavior in Table 1), it outputs the BN structure displayed in Fig. 1.

4.2. Estimating system reliability using Bayesian networks

The next step of the proposed method estimates system reliability using the BN that was constructed by the $K2$ algorithm.

Table 2
Values of variables in f scoring function

X_1	X_2	α_{ijk}	α_{ij}
0	0	3	5
0	1	2	
1	0	2	5
1	1	3	

Table 3
f Scores for all possible candidate parent sets

Parent set	f Score
ϕ	1/2772
$\{X_1\}$	1/3600
$\{X_2\}$	1/3600
$\{X_1, X_2\}$	1/144

Table 4
CPT of the X_3 node

Parents		Probability	
X_1	X_2	$X_3 = 0$	$X_3 = 1$
0	0	1	0
0	1	0	1
1	0	0	1
1	1	1	0

Besides the associations that were discovered in the previous step, the inference rules described in Section 3.1 should be used to calculate the *conditional probabilities*. Although the $K2$ algorithm is helpful for computing the conditional probabilities between components in its intermediate steps, it does not output them explicitly. Therefore, the conditional probabilities should be calculated alone. These probabilities are essential for calculating the overall reliability of the system, as they represent the degrees of association between components of a system.

The conditional probabilities are calculated and stored in CPT in a BN. Each component with a non-empty parent set in the network is associated with a CPT. (The ones with no parents are *independent* of others and associated with *prior probabilities* as explained in Section 3.1.) Each CPT has entries as many as 2^u where u is the number of parents of that component in the network. In the BN example given in Section 4.1, the CPT of component X_3 is provided in Table 4.

The probability values displayed in the CPT are calculated by using the raw data in Table 1 and can be expressed as the probability of an instantiation of the parent set. Therefore the probability, X_3 being 0 given the parent instantiations as $X_1 = 0$ and $X_2 = 0$, is 1, as 3 out of 10 observations parents are instantiated as 0 and 0, and for all of these cases X_3 is instantiated as 0. In this example, the probability values in the CPT are either 1 or 0, showing a very strong association between X_3 and its parent set $\{X_1, X_2\}$. However, in normal applications the values in the CPT can vary between 0 and 1.

In the next step, with the help of CPT and the prior probabilities that X_1 and X_2 have, the success probability value for X_3 can be calculated. According to the BN structure (in Fig. 1) components X_1 and X_2 are independent of others; therefore, their success probabilities can be directly inferred from the observation dataset in Table 1. From Table 1 it can be evaluated that $p(X_1 = 1) = 0.5$ and $p(X_2 = 1) = 0.5$. The rest of the calculation is shown in Table 5.

The probability of success for component X_3 is 0.5 as both components X_1 and X_2 have the same degree of association on X_3. Extending the computations for the other components in the network, success probabilities for the rest of the components in the sample system can be evaluated, such that $p(X_4 = 1) = 0.4$ and $p(X_5 = 1) = 0.6$.

In the last step, the system reliability can be calculated by using these probability values and the CPT of the "System Behavior" node in the BN structure given in Fig. 1. Using the table of observations provided in Table 1, the CPT for the "System Behavior" node can be constructed as given in Table 6.

By implementing Bayes' rule, the overall system success probability can be computed as follows:

$$p(\text{System Behaviour} = 1) = 0.5 \times 0.4 \times 0 + 0.5 \times 0.6 \times 1 + 0.5 \times 0.4 \times 1 + 0.5 \times 0.6 \times 1 = 0.8$$

The success probability of the *System Behavior* node is 0.8 or 80%; which is the reliability of the sample system used in this section. The reader must recall that this reliability value is calculated based on only 10 observations on the sample system. With more observations available, the *K*2 algorithm can provide more accurate results on the degrees of associations between the system components and calculate more precise values in the CPT of the nodes, which will increase the accuracy of the calculated system reliability in turn. The proposed method for estimating system reliability using observations dataset is superior to previously defined methods due to its unsupervised nature; almost all steps of the required computations can be done without any human intervention.

5. Experimental analysis

In this section experimental performance analysis of the proposed method for system reliability estimation is provided. In order to give a better perception of analysis, performances of the two phases of the proposed method are examined separately. First, performance and correctness of the *K2* BN construction algorithm are analyzed using historical data (obtained via Monte Carlo simulation) for BNs given in Fig. 2.

BNs displayed in Fig. 2 represent different systems with various components. In order to make experimental analysis, separate datasets—similar to Table 1—are used for each example BN. If a component is not functioning during an observation it is represented as 0 and if it is working properly it is represented as 1 in the dataset. Therefore the datasets contain *binary data*, and the size of a dataset is $n \times t$ where n is the number of components in the system and t is the number of observations done to collect data.

As it was explained in Section 4.1, the *K*2 algorithm uses this binary data as input. Therefore, running time of this algorithm is highly dependent on the size of the input dataset, and thus the number of nodes (n) and the number of observations (t). Each of the case BN shown in Fig. 2 has different numbers of nodes and the performance of the *K*2 algorithm on each BN is analyzed using different input datasets. Fig. 3 shows the experimental results on the performance of the algorithm.

It can be observed from Fig. 3 that the running time of the algorithm is quadratic ($O(n^2)$) with the number of nodes and linear with the number of observations. This is an expected result as implied in Section 3.2; the *K*2 algorithm reduces the time complexity of finding associations from exponential (2^n) to quadratic (n^2). This would bring the conclusion that for even substantially large systems ($n > 100$) the *K*2 algorithm will be efficient to use. On the other hand, for the systems with around 20

Table 5
Probability of success calculation for node X_3

$p(X_3 = 1) = \Sigma\, p(X_1, X_2, X_3 = 1)$
$p(X_1 = 0, X_2 = 0, X_3 = 1)+p(X_1 = 0, X_2 = 1, X_3 = 1)+p(X_1 = 1, X_2 = 0, X_3 = 1)+p(X_1 = 1, X_2 = 1, X_3 = 1)$
$= 0.5 \times 0.5 \times 0+0.5 \times 0.5 \times 1+0.5 \times 0.5 \times 1+0.5 \times 0.5 \times 0 = 0.5$

Table 6
CPT of the "System Behavior" node

Parents		Probability	
X_3	X_5	System Behavior = 0	System Behavior = 1
0	0	1	0
0	1	0	1
1	0	0	1
1	1	0	1

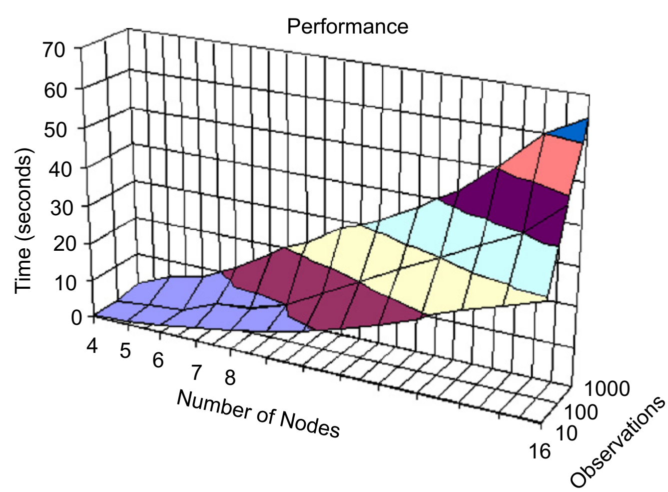

Fig. 3. Running time of the *K*2 algorithm.

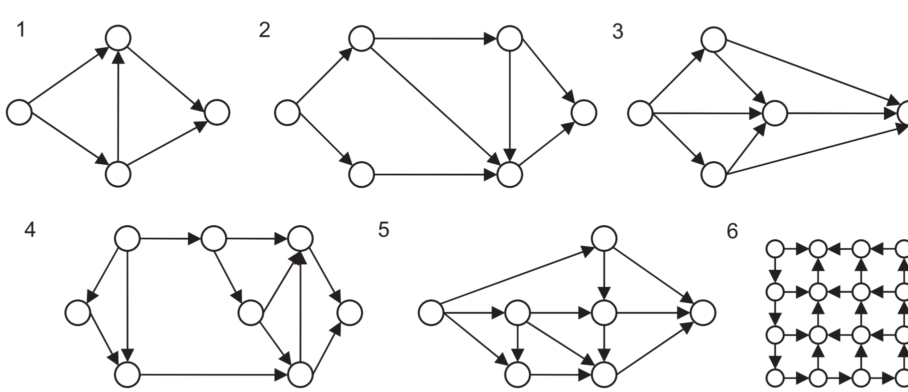

Fig. 2. Case BN tested on the *K*2 algorithm.

components the associations between components can be discovered in less than 5 min.

The number of observations that are used for discovering the associations between system components is also an important measure for both the efficiency of the $K2$ algorithm and the correctness of the constructed BN. Accuracy of the reliability estimation is highly dependent on the correctness of the resulting BN model. Errors in the $K2$ algorithm would lead to incorrect assignments of associations in the BN, which will end up with inaccuracies in the reliability values. Once the BN is correctly constructed, estimating the system reliability is simple and straightforward as discussed in Section 4.2; therefore, the main focus of this section is correctness of the BN constructed by the $K2$ algorithm. The $K2$ algorithm can be expected to find out associations more accurately when more observations are used as input. Using the constructed BN, error rate (ρ) of the $K2$ algorithm can be calculated as

$$\rho = \frac{A_{\mathrm{M}}}{A_{\mathrm{T}}} = \frac{A_{\mathrm{FP}} + A_{\mathrm{FN}}}{A_{\mathrm{T}}} \tag{6}$$

A *false positive* (A_{FP}) is an association decided by the $K2$ algorithm; however, does not exist in the actual BN given in Fig. 2. Conversely, a *false negative* (A_{FN}) is an existing association in the actual BN that is missed by the $K2$ algorithm. While calculating the accuracy (where accuracy $=1-\rho$) of the $K2$ algorithm using the constructed BN, both should be taken into account. In this study, correctness of the constructed BN models are evaluated by using datasets with 10, 100 and 1000 observations for different case networks. General analysis results on the correctness and accuracy are provided in Fig. 4.

According to Fig. 4, regardless of the size of the constructed BN, accuracy increases as more observations are used. This is an expected result, because the associations among components are decided by using the observations and these associations can be figured out more precisely with more observations available. However, as shown in Fig. 3 running time of the $K2$ algorithm increases as more observations are used. Trade-off between efficiency and accuracy of the $K2$ algorithm for case BN #6 is provided in Fig. 5. It can be observed in Fig. 5 that the error rate of the algorithm drops linearly as the number of observations is increased exponentially from 10 to 1000. However, the running time of the algorithm also increased by 9 times. These two curves

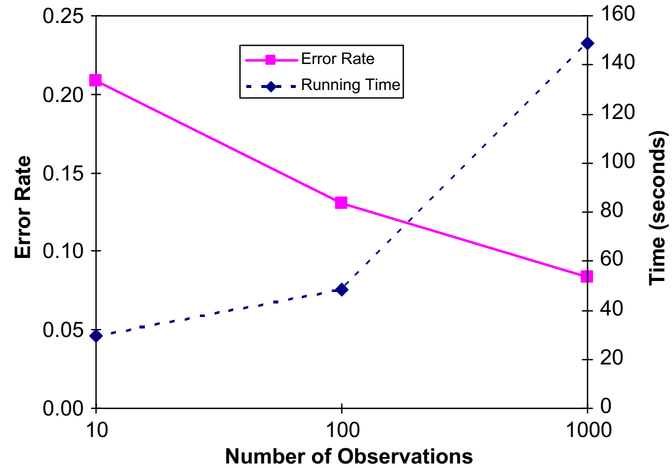

Fig. 5. Performance vs. accuracy curve for the $K2$ algorithm for case BN #6.

intersect at around 300, which is the optimum number of observations to be used for case BN #6.

Using the case BN that are shown in Fig. 2, the corresponding reliabilities can be computed by following the methodology described in Section 4.2 in detail. The reliability estimation results as well as CPU times for BN construction and reliability estimation are presented in Table 7.

5.1. Reliability estimation results

In this section, the automation of the reliability analysis is implemented on the system presented in Fig. 6 and the results are compared against the actual values presented in Ramirez–Marquez and Jiang [36]. For this problem a dataset including information of 1000 instances of system behavior was obtained. This dataset was implemented in the $K2$ algorithm and a BN was obtained and evaluated with results presented in Table 8.

The results presented in Table 8 illustrate that based on the data provided the proposed approach was able to identify a reliability value within 1% of the actual system reliability. That is, the traditional problem of describing the interaction of system components graphically, so that system reliability can be obtained can be accurately replaced by the approach presented.

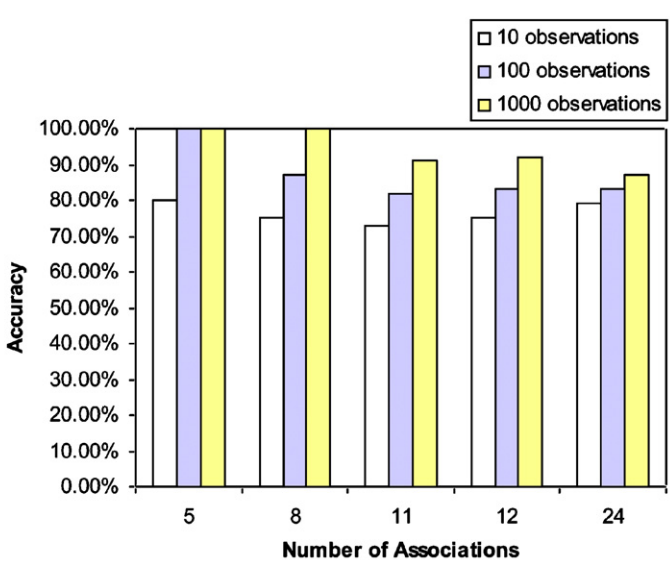

Fig. 4. Accuracy of results in BN construction.

6. Conclusions

Estimating system reliability using the BN model is a very popular practice and widely studied recently. There are numerous methods in the literature defined for estimating system reliability, which are mainly focused on doing it for specific systems, such as nuclear plants. However, none of these studies dealt with the problem of constructing the BN without the need of a human expert. This is the first study that introduces a methodology for efficient construction of BN models and estimating system reliability, eliminating the human expert requirement. The proposed method uses historical data about the system to be modeled and constructs the BN model without the need for human intervention. The $K2$ algorithm is used for this purpose, which is a popular and efficient association rule mining method.

Next it was shown that the system reliability can efficiently be estimated by using the BN model. According to the experimental results, reducing the running time of finding associations from $O(2^n)$ to $O(n^2)$, the proposed methodology can work efficiently

Table 7
Compilation of results for case BN

Network	Number of nodes	Number of associations	*K*2 algorithm CPU time (s)	Accuracy of the constructed BN (w/1000 observations) (%)	Reliability estimation CPU time (s)	Reliability
1	4	5	1.839	100.00	0.465	0.5609
2	6	8	10.471	100.00	0.981	0.3765
3	5	8	5.567	100.00	0.830	0.6103
4	8	11	37.898	90.91	1.004	0.3605
5	7	12	26.012	91.66	0.991	0.5569
6	16	24	148.762	87.50	1.587	0.5914

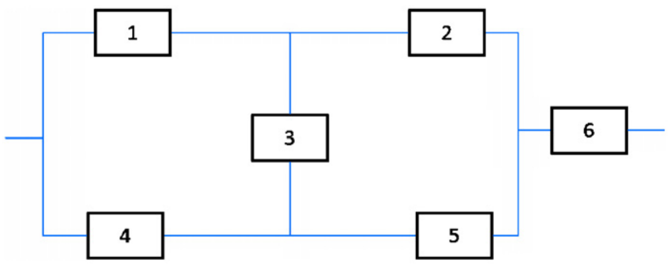

Fig. 6. Case two-terminal network.

Table 8
Comparison of system reliability results

Component	Nominal reliability	
	Case 1	Case 2
1	0.9	0.85
2	0.8	0.8
3	0.9	0.95
4	0.93	0.9
5	0.83	0.875
6	0.85	0.85
System reliability via BN	0.80725	0.80975
System reliability (Ramirez–Marquez and Jiang)	0.813388	0.815113

even with substantially large systems. Moreover, the BN models constructed by the *K*2 algorithm are shown to be accurate, especially when more historical data about the system are available. As expected, the experimental results show that when 1000 historical observations on the system are available, the constructed BN are more than 90% accurate. The accuracy of the *K*2 algorithm can further be improved when the already existing associations between system components are taken into account. However, this improvement requires the intervention of a human expert and thus will be left for future work.

Accuracy of the constructed BN is highly influential on the correctness of the system reliability values, as incorrect associations in the BN would lead to biased calculations while estimating system reliability. In conclusion, the methodology introduced in this study will help systems engineers as it minimizes human interaction and provides efficient ways of automatically building a BN model and estimating system reliability.

References

[1] Gran BA, Helminen A. A Bayesian belief network for reliability assessment. Safecomp 2001 2001;2187:35–45.
[2] Krishnamurthy S, Mathur AP. On the estimation of reliability of a software system using reliabilities of its components. In: Proceedings of the ninth international symposium on software reliability engineering (ISSRE '97). NM: Albuquerque; 1997. p. 146.
[3] Coyle T, Arno RG, Hale PS. Application of the minimal cut set reliability analysis methodology to the gold book standard network. In: Proceedings of the industrial and commercial power systems technical conference; 2002. p. 82–93.
[4] Fant K, Brandt S. Null convention logic, a complete and consistent logic for asynchronous digital circuit synthesis. In: Proceedings of the international conference on application specific systems, architectures, and processors (ASAP '96); 1996. p. 261–73.
[5] Gopal C, Kuolung H, Nader A. A new approach to system reliability. IEEE Trans Reliab 2001;50(1):75–84.
[6] Amasaki S, Takagi Y, Mizuno O, Kikuno T. A Bayesian belief network for assessing the likelihood of fault content. In: Proceedings of the 14th international symposium on software reliability engineering; 2003. p. 215–26.
[7] Boudali H, Dugan JB. A continuous-time Bayesian network reliability modeling, and analysis framework. IEEE Trans Reliab 2006;55(1):86–97.
[8] Barlow RE. Using influence diagrams. Accelerated Life Testing and Experts' Opinions in Reliability 1988:145–50.
[9] Almond R.G. An extended example for testing graphical belief, Technical Report No. 6, 1992.
[10] Sigurdsson JH, Walls LA, Quigley JL. Bayesian belief nets for managing expert judgment and modeling reliability. Qual Reliab Eng Int 2001;17:181–90.
[11] Bobbio A, Portinale L, Minichino M, Ciancamerla E. Improving the analysis of dependable systems by mapping fault trees into Bayesian networks. Reliab Eng Syst Saf 2001;71(3):249–60.
[12] Gelman A, Carlin JB, Stern HS, Rubin DB. Bayesian data analysis. London: Chapman & Hall; 1995. p. 1–526.
[13] Littlewood B, Popov P, Strigini L. Assessment of the reliability of fault-tolerant software: a bayesian approach. In: Proceedings of the 19th international conference on computer safety, reliability and security (SAFECOMP 2000). Berlin: Springer; 2000.
[14] Lagnseth H, Portinale L. Bayesian networks in reliability. Tech Rep 2005.
[15] Herald T, Ramirez-Marquez JE. System element obsolescence replacement optimization via life cycle cost forecasting. NJ: Hoboken; 2007 [Technical report, Stevens Institute of Technology].
[16] Cooper GF, Herskovits E. A Bayesian method for the induction of probabilistic networks from data. Mach Learn 1992;9(4):309–47.
[17] Cowell RG, Dawid AP, Lauritzen SL, Spiegelhalter DJ. Probabilistic networks and expert systems. New York: Springer; 1999.
[18] Jensen FV. Bayesian networks and decision graphs. New York: Springer; 2001.
[19] Pearl J. Probabilistic reasoning in intelligent systems. San Francisco, CA: Morgan Kaufmann; 1988.
[20] Fenton N, Krause P, Neil M. Software measurement: uncertainty and causal modeling. IEEE Software 2002;10(4):116–22.
[21] Gran BA, Dahll G, Eisinger S, Lund EJ, Norstrøm JG, Strocka P, Ystanes BJ. Estimating dependability of programmable systems using Bbns. In: Proceedings of the Safecomp 2000. Berlin: Springer; 2000. p. 309–20.
[22] Dahll G, Gran BA. The use of Bayesian belief nets in safety assessment of software based systems. Special Issues Int J Intelligent Inf Syst 2000;24(2): 205–29.
[23] Helminen A. Reliability estimation of software-based digital systems using Bayesian networks. Technical Report, Helsinki University of Technology Espoo, 2000. p. 1–50.
[24] Hugin Expert, (2007), Aalborg, Denmark 〈http://www.hugin.dk〉 (2008).
[25] Serene-Safety and Risk Evaluation Using Bayesian Nets, (2006), 〈http://www.hugin.dk/serene/〉 (2008).
[26] Spiegelhalter D, Thomas A, Best N, Gilks W. Bugs 0.5 Bayesian inference using Gibbs sampling manual (Version Ii). MRC Biostatistic Unit 1996;1:1–59.
[27] Helminen A, Pulkkinen U. quantitative reliability estimation of a computer-based motor protection relay using Bayesian networks. Safecomp 2003 2003;2788:92–102.
[28] Ghokale S, Lyu M, Trivedi K. Reliability simulation of component based software systems. In: Proceedings of the international symposium on software reliability engineering (ISSRE'98); 1998.
[29] Ghokale S, Wong E, Trivedi K, Horgan JR. An analytical approach to architecture based software reliability prediction. In: Proceedings of the symposium on application specific systems and software engineering technology (ASSET '98). TX: Dallas; 1998.

[30] Singh H, Cortellessa V, Cukic B, Gunel E, Bharadwaj B. A Bayesian approach to reliability prediction and assessment of component based systems. In: Proceedings of the 12th international symposium on software reliability engineering (ISSRE'01); 2001. p. 12.
[31] Yacoub S, Cukic B, Ammar H. Scenario-based reliability analysis of component-based software. In: Proceedings of the 10th international symposium on software reliability engineering (ISSRE '99); 1999. p. 22–31.
[32] Xie M, Wohlin C. An additive reliability model for the analysis of modular software failure data. In: Proceedings of the sixth international symposium on software reliability engineering (ISSRE '95); 1996. p. 188–93.
[33] Shijun Q, Agogino AM, Song S, Wu J, Sitarama S. A fusion of Bayesian and fuzzy analysis for print faults diagnosis. In: Proceedings of the 16th conference on computers and their applications; 2001. p. 229–32.
[34] Detcher R, Mateescu R. Mixtures of deterministic-probabilistic networks and their and/or search space. In: Proceedings of the 20th conference on uncertainty in artificial intelligence; 2004. p. 120–9.
[35] Ruiz C. Illustration of the K2 algorithm for learning Bayes net structures. In: Lecture notes, Worchester Polytechnic Institute, 2005.
[36] Ramirez-Marquez JE, Jiang W. Confidence bounds for the reliability of binary capacitated two-terminal networks. Reliab Eng Syst Saf 2006;91(7): 905–14.