Framework for Evidence Based-Decision-Making by using Bayesian Networks

Munigurava Reddy[1], Bullarao Domathoti[2], P. Nageswara Rao[3]
[1, 2, 3] Department of CSE
[1, 2, 3] Swetha Institute of Technology &Science, Tirupati , AP, INDIA

Abstract*- Recommendation systems in software engineering (SE) should be designed to integrate evidence into practitioners experience. Bayesian networks (BNs) provide a natural statistical framework for evidence-based decision-making by incorporating an integrated summary of the available evidence and associated uncertainty (of consequences). In this study, we follow the lead of computational biology and healthcare decision-making, and investigate the applications of BNs in SE in terms of 1) main software engineering challenges addressed, 2) techniques used to learn causal relationships among variables, 3) techniques used to infer the parameters, and 4) variable types used as BN nodes. We conduct a systematic mapping study to investigate each of these four facets and compare the current usage of BNs in SE with these two domains. Subsequently, we highlight the main limitations of the usage of BNs in SE and propose a Hybrid BN to improve evidence-based decision-making in SE. In two industrial cases, we build sample hybrid BNs and evaluate their performance. The results of our empirical analyses show that hybrid BNs are powerful frameworks that combine expert knowledge with quantitative data. As researchers in SE become more aware of the underlying dynamics of BNs, the proposed models will also advance and naturally contribute to evidence based-decision-making.*

Keywords*-* Evidence-based decision-making, Bayesian networks, Bayesian statistics, software reliability, software metrics, post-release Defects.

I. INTRODUCTION

Bayes' theorem is a simple equation that shows how a conditional probability depends on its inverse conditional probability. According to Bayes' theorem, the probability of an event A conditioned on an event B can be calculated as:

$$P(A|B)=P(B|A)P(A)/P(B)$$

Bayes' theorem expresses how a prior belief about a probability should change in the light of new evidence. For example, it can be used to update the probability of a diagnosis hypothesis given the observation of a symptom. Suppose that the prevalence of tuberculosis in a particular community is 1%, and 44% of the people in the same community suffers from shortness of breath. By considering the historical patient records, we know that 79% of the patients who had been diagnosed with tuberculosis also suffered from shortness of breath. Although this information tells nothing about the probability of having tuberculosis given that one suffers from shortness of breath, this probability can be calculated using Bayes' theorem.

BNs are graphical probabilistic models that are composed of a graphical structure and a set of parameters. The graphical structure of a BN contains nodes representing variables and directed edges representing relations between those variables. If a directed edge connects variables A and B as in $A \rightarrow B$, A is called a parent variable and B is called a child variable. Figure 1.1 shows a BN model, known as the Asia BN, which has 8 nodes and 8 edges.

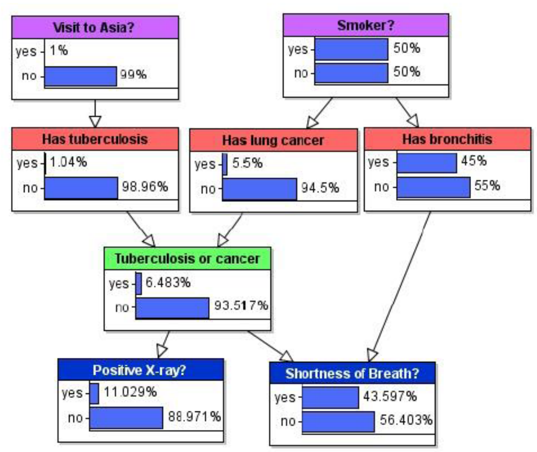


Figure 1.1 Asia BN

Variables in a BN can either be discrete or continuous. Discrete variables are defined by mutually exclusive and collectively exhaustive set of states. All of the variables in Asia BN are discrete variables that have 2 states.

Each variable in a BN has a set of parameters that defines its probabilistic relation with its parents, or its prior distribution if the variable does not have any parents. The parameters of discrete nodes are encoded by node probability tables (NPT). A NPT contains probability values for each state of the variable given every combination of the states of its parent variables. Table 2.1 shows the NPT of the ‘Has tuberculosis’ variable in the Asia BN. The NPT has 4 probability values since the variable has 1 parent, and both the variable and its parent have 2 states each.

Table 2.1 NPT of the ‘Has tuberculosis’ variable

		Visit to Asia?	
		Yes	No
Has tuberculosis	Yes	0.05	0.01
	No	0.95	0.99

The probability distributions of continuous variables can be defined by using statistical distributions or functions of their parent variables (see Fenton and Neil (2012a; 2012b) for a thorough introduction to modelling with discrete and continuous variables in BNs). In the following chapter, we illustrate how BNs reason by an example about the Asia BN.
To answer these questions, we set the following research objectives:

1) extend the work of Radlinski [19] and ours [20] by conducting a comprehensive mapping study on the usage of BNs in software engineering topics;
2) identify the current best practices in software engineering, i.e., usage of techniques to define causal relationships, estimate parameters, and infer the probability of the outcome;
3) highlight limitations, if any, on the usage of BNs in our field by comparing with other applications of BNs, particularly in computational biology and healthcare decision-making;
4) propose a new approach that considers current limitations and risks to improve decision-making in software engineering;
5) evaluate the performance of the new approach in two industrial case studies.

In this paper, we present our findings as follows. We present a systematic mapping on the usage of BNs in software engineering (Section 3). We summarize the state-of-the-art techniques used to build BNs, and compare them with the applications of BNs in other domains (Section 3.7). Based on our findings, we design a Hybrid BN that proposes different techniques (inspired from other domains) for building the network structure (Section 4) and evaluate its performance in two industrial cases (Section 5). Subsequently, we discuss potential threats to the validity of our systematic mapping and BN construction steps (Section 7) and conclude by summarizing the contributions of this work for practitioners and researchers (Section 8).

2.4 Condition Independence and Bayesian Networks

A BN can represent a joint probability distribution compactly in a factorised way. The graphical structure of a BN is a directed acyclic graph that encodes a set of CI assertions about its variables. Every node in a BN is independent of its non-descendants given that the state of its parents is known. Therefore, each node has a conditional probability distribution (CPD) that defines its probabilistic relation with its parents. A probability distribution P_X factorises over a BN structure G_X if P_X can be decomposed into the product of factors $P_X=(X_1,\ldots,X_n)=\prod_{i=1}^{n} P(X_i|PA_{X_i}^{G_X})$ where $X_1,\ldots,X_n$ are a set of variables, $PA_{X_i}^{G_X}$ is the set of parents of X_i in G_X.
The CIs that can be encoded in a BN can be shown by the relation between three variables.

1. If two variables, A and B, are directly connected by an edge, as shown in Figure 2.4a, a BN does not assert any CI conditions between these variables.

2. If there is a serial relation between three variables A,V and B, as shown in Figure 2.4b, then A and B becomes independent given that the state of V is known.

3. If there is a diverging relation between A,V and B, as shown in Figure 2.4c, A and B becomes independent given that the state of V is known.

4. If there is a converging relation between A, and B, as shown in Figure 2.4d, A and B are independent. However, this independence disappears if the state of V or one of its descendants is known.

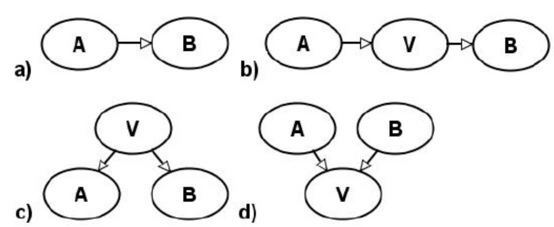


Figure 2.4 (a) Direct Connection (b) Serial Relation (c) Diverging Relation (d) Converging Relation

In general, CI assertions of a BN can be determined by d-separation (Pearl, 1988):

d-separation: A trail $X_1 \leftrightharpoons \cdots \leftrightharpoons X_n$ is a consecutive sequence of edges that can be in any direction. Let G be a BN structure, A, and V be a three disjoint sets of nodes in G. A and B are d-separated by V, $dsep_G(A;B|V)$, if and only if there is no *active* *trail* between A and B given that V is observed. An active trail requires the following conditions:

1. For every converging relation $Xi-1 \rightarrow Xi \leftarrow Xi+1$ in the trail, the node Xi or one of its descendants is a member of V.
2. The other nodes in the trail are not members of V.

If A and B are d-separated given V in the BN structure G, then A and B are conditionally independent given V in any probability distribution that factorises over the BN. A and B are called d-connected if they are not d-separated. It follows from the definition of d-separation that adding an edge to a BN increases the number of trails and therefore does not increase the number of CI conditions.

A BN structure G asserts a set of conditional independencies (G). P can factorise on G if (G) is a subset of (P), i.e. the set of conditional independencies in P. Such G is called an I-map of P.

G is an I-map of P if and only if $(G) \subseteq I(P)$

Any CI that holds on the BN structure G must also hold on the probability distribution P, if P factorises over G. On the other hand, P can have additional CI conditions that are not reflected in G. Therefore, a probability distribution can factorise over various BN structures.

An example of this situation can be seen by the two BNs in Figure 2.5. Some probability distributions can factorise on both of these BNs even though their graphical structure is different. In the BN in Figure 2.5a, as well as in the probability distribution P that factorises over this BN, A and B are conditionally independent given that the state of C is not known. This CI is not represented in the graphical structure of the BN in Figure 2.5b. However, the CI condition can still be present in the probability distribution that factorises over this BN structure. In other words, the 31

CI between A and B can be encoded in the parameters of this BN rather than its structure. The BN on the left is preferable since an edge between A and B is unnecessary for this probability distribution, and additional edges increase the computational burden of a BN. The obvious conclusion is to choose a BN structure that encodes all of the independencies of the probability distribution in its graphical structure. Unfortunately, this is not possible in general. Symmetric variable-level CIs or some regularities in the parameters do not have a BN structure that represents all of the CIs (Pearl, 1988).

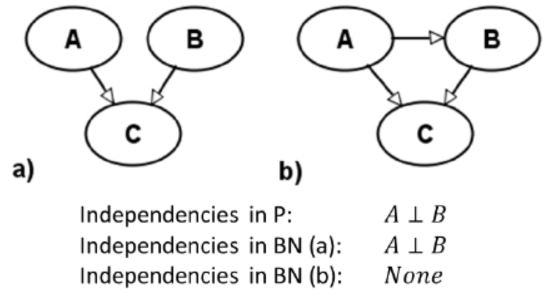


Figure 2.5 Same Probability Distribution Factorised over Two Different BN Structures

2.6.3 Hybrid Methods that Combine Knowledge and Data

Previous sections discussed several limitations of purely data and knowledge driven techniques. Methodologies that combine data and expert knowledge seek to overcome these limitations by using all available information in BN development. However, research in this area is still in early steps, and there are many challenges that need to be addressed.

Structure

Flores et al. (2011) proposes a method that integrates expert's opinion about the presence and direction of the arcs into structure learning. In this method, experts can define the type of the relations between variables and assign a prior probability representing their confidence. For example, an expert can say that he is 80% confident that there is a direct relation between two variables but he is not sure about the causal direction of this relation. The expert can also define other types of relations including direct causal connection, causal dependence, temporal order, and correlation. Afterwards, the BN structure is learned based on these expert priors using a score-based method.

Cano et al.'s method (2011) uses expert judgement during the learning process instead of using it as priors. A Bayesian score is used for the learning algorithm. The arcs that have the most uncertainty, according to the learning algorithm, are shown to experts. Afterwards, the experts make the final decision about the presence and direction of these arcs. This approach can decrease the time spent by experts since their opinion is only used for the most uncertain BN elements.

Velikova et al. (2013) uses structure learning methods as a complementary approach to evaluate and refine the BN structure built with experts. Antal et al. (2004) proposed a method for combining data and textual information from the medical literature to build BNs. They use information retrieval techniques to assist structure learning based on the textual information in medical literature.

Parameters

Bayesian learning methods can integrate expert knowledge into parameter learning by using informative priors. However, eliciting numbers for informative distributions can be difficult as experts often feel less confident in expressing quantitative statements (Druzdzel and Van Der Gaag, 2000). Therefore, using qualitative constraints, such as "value of A is greater than value of B", can be more convenient. Zhou et al. (2013a, 2013b) proposed a technique for integrating expert knowledge as constraints when learning multinomial parameters from data. Similar approaches are also proposed by Feelders and Van der Gaag (2006) for binomial parameters, by Tong and Ji (2008) for a limited amount of constraints, and by Khan et al. (2011) for diagnostic BNs.

2.6.4 Knowledge Gap

The knowledge engineering and machine learning communities has focussed less on hybrid methodologies compared to purely knowledge or data driven approaches. Although the number of studies about hybrid methodologies has been increasing in recent years, many of these studies have addressed similar challenges. From the reviewed studies, the hybrid structure learning methods mainly focus on using knowledge to assist a data-based structure learning algorithms. The hybrid parameter learning studies mainly focus on using knowledge as constraints for parameter learning. Combination of knowledge and data also has potential benefits in other challenges of BN modelling that need to be addressed. For example, BNs that reason consistent with knowledge often contains variables that cannot be directly measured and thus not available in the dataset. Hybrid methodologies that combine knowledge and data are required to deal with this task. In the following chapter, we discuss the application of knowledge and data driven techniques for medical models. In Chapter 4, we introduce the medical case study and, by using the case study, we illustrate several modelling challenges that can be dealt with novel hybrid methodologies.

III. SYSTEMATIC MAPPING ON BNS IN SE

3.1 Research Methodology

We conduct a mapping study to achieve our research objectives #1 and #2, as defined in Section 1. Mapping study is a type of secondary study that aims to search a broader field for any kind of research in order to get an overview of the state-of-the-art or state-of-practice on a topic [25]. It is conducted if the research question for the literature review is broader, or the field is less explored. Mapping studies follow the same principled process and protocol as systematic

MISIRLI AND BENER: BAYESIAN NETWORKS FOR EVIDENCE-BASED DECISION-MAKING IN SOFTWARE ENGINEERING 535

literature reviews; however, the goals and research questions are more generic, and search strategy criteria are less stringent compared to systematic literature reviews [6]. The main objective of mapping studies is to only classify the relevant literature with respect to defined categories, as opposed to systematic literature reviews, which aggregate primary studies in terms of research outcomes and investigate whether those research outcomes are consistent or not [6], [26].

Based on the pioneering guidelines by Kitchenham et al. [5], [6] and Wohlin et al. [25], we present a systematic mapping study to better understand the usage of BNs in software engineering. We follow the review protocol defined (with examples) by Petersen et al. [26] for systematic mapping studies: "Definition of research questions", "Conducting the search for relevant papers", "Screening of papers" (inclusion/exclusion criteria), "Definition of classification scheme" (Categories), and "Data extraction and mapping process."

3.2 Research Questions

We further divide the first research question (What is the current usage of BNs in software engineering field?) into sub-questions, and answer each of them in Section 3:
RQ1a: Which topics are covered in software Engineering studies employing BNs?
RQ1b: Which techniques are used for structure learning in the construction of BNs?
RQ1c: Which techniques are used for parameter learning in the construction of BNs?
RQ1d: What type of variables are represented as nodes in BNs?

3.3 Search for Primary Studies

We define our search string as follows:
(Bayesian Network OR Bayesian Net OR Bayes Net OR Bayesian Belief Network) AND Software Engineering. We searched the following four databases since they have been widely used by other studies in the literature: IEEE Xplore Digital Library, ACM Digital Library, Science Direct, and Web of Science. We did not set any time limit to our search for answering our research questions.

3.4 Screening of Papers

Inclusion. We included all papers, published in international conferences, workshops, symposiums and journals, specifically describing the use of BNs to solve a particular problem in software engineering. Since we only selected digital publication databases, we did not work on unpublished work or presentations in this study. When several papers reported the same study (both model details and data used for validation), we only included recent publication.

Exclusion. Papers outside of the software engineering domain, only abstracts, and papers written in languages different than English (even though abstracts may be written in English) were discarded. Further, papers on the following topics were excluded:

– Papers employing Bayesian statistics (e.g., Monte
– Carlo methods to estimate regression coefficients) without building a network.
– Papers comparing several BNs that were built in previous studies.
– Papers describing the usage of Naive Bayes for a particular problem.1

3.5 Classification Scheme

Main facets are created on the basis of the research questions. Regarding RQ1a, we read the abstract, title, and keywords of all papers and classified papers based on the Knowledge Areas defined in Software Engineering Body of Knowledge (SWEBOK), version 2004 [27]. Therefore, the first facet is defined as the software engineering topic. Regarding RQ1b and RQ1c, classification for the second facet, structure learning, and the third facet, parameter learning, were derived from papers. Reading meta-data only (title, abstract, and keywords) was not sufficient to understand which techniques were employed during BN construction in a research paper. Therefore, we read those sections of the papers in which authors explained how they constructed BNs to solve their specific problems and extracted a list of approaches, algorithms, and/or techniques.

Then, we inductively formed higher levels of classifications covering each of these approaches in the list. Our final classification is described in the next section with examples from the literature.

Regarding RQ1d, the fourth facet is defined as the type of variables represented as nodes in BNs. This facet was also investigated by reading the whole paper. The classification for this facet is actually trivial because there are mainly three types that can be used to represent the nodes in a BN:

1) Categorical (also known as Discrete), 2) Continuous, and 3) Both.

Based on these four facets, we have classified all primary studies and answered the research questions in the next section. Prior to classification, both authors of this study agreed on the mapping protocol, and the terminology related to four research questions. During the classification process, one author made the classification for each of the four aspects by reading the papers. In case of doubtful assignments, the guidelines in [6] were followed: Both authors read the paper and discussed the doubtful assignment until they reached an agreement.

3.6 Answers to RQ1

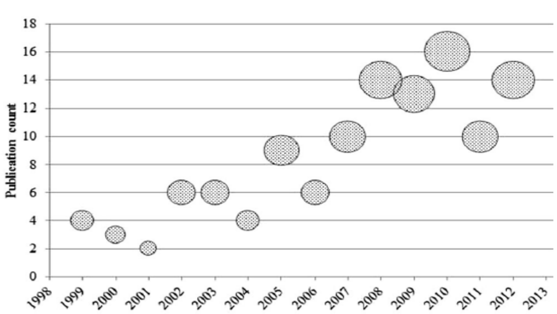
The number of publications returned by our search is depicted in Table 1. In this table, Inc represents the number of included publications, Exc represents the number of excluded publications (based on our exclusion criteria), whereas Dup./Repl. represents the number of duplicated/replicated publications. The cases where duplicate records were retrieved from many databases, or same papers are published in more than one venue, were excluded from our study. After the first screening (reading the abstract, title, and keywords), we selected 145 papers using BNs to contribute to software engineering research. To answer RQ1b, 1. Naive Bayes is a special type of BN in which all input variables are assumed as independent, and have a direct relationship with the output variable. Furthermore, all variables follow a Gaussian distribution whose parameters (mean, standard deviation) are estimated from data.

TABLE 1: Screening Process

Database	Results of the search string	Potentially relevant studies (1^{st} selection)			Primary studies (2^{nd} selection)	
		Exc.	Dup./ Repl.	Inc.	Exc.	Inc.
IEEE Xplore	508	415	0	93	20	73
ACM Digital Library	217	189	4	24	3	21
Science Direct	487	462	0	25	4	21
Web of Science	8	0	5	3	1	2
Total	1220	1066	9	145	28	117

RQ1c, and RQ1d, we also read some sections in these papers where the details of BNs were given. After this second screening, we excluded 28 papers based on our exclusion criteria, and classified 117 papers in this study. These 117 primary studies were published in 53 different international conferences, symposiums and workshops, and 13 journals. Full list of journals and conferences with the distribution of primary studies is also provided as a supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TSE.2014.232 1179. Fig. 2 depicts a growing trend over years (in terms of the number and percentage of publications),which may also indicate a growing interest in BNs as a means for solving different challenges in software engineering. Radlinski [19] also experienced a similar growing trend in the publications using BNs for software effort estimation (65 percent of publications in 2008-2010). However, we cannot draw a concrete conclusion based on these trends due to several reasons. First, when we observe Fig. 2 in terms of the percentage values, it is clear that 2010 was the best year with 14 percent (17 publications), and no smooth increase has occurred in terms of the studies employing BNs since 2008. Second, to report a convincing evidence about the growing interest on BNs in software engineering, we have to compare these trends with the applications of other Artificial Intelligence (AI) techniques.

There have been few studies summarizing the applications of AI on different software engineering topics, such as natural language processing in software requirements analysis [28], genetic algorithms (GAs) in software architecture, test automation and project planning [28], [29], and case based reasoning (CBR) on software coding, reuse and predictions [30]. Shepperd [30] states that CBR is a relatively recent technology formulized in 1980s that many exciting opportunities can be found for employing it in our field; yet it has open challenges such as adaptation of rules and collaboration with human experts. GAs, on the other hand, are found very popular in the context of project planning and design [28], [29]. Though there is no comparison between the growth rate of GAs and other AI techniques in these topics, it is emphasized in [29] that the use of BNs is likely to be more appealing in practice due to its transparency to practitioners. The trend in Fig. 2 generally supports this claim.



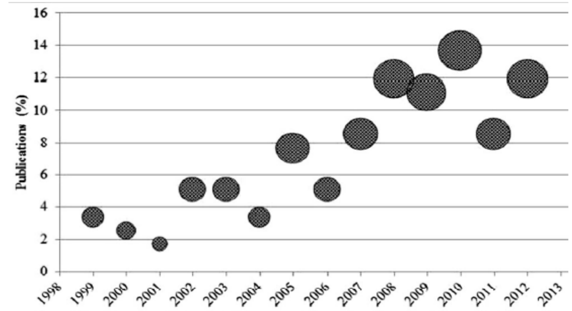


Fig. 2. Bubble plot for publications by year.

IV. RESULTS

Prediction performances of BN models are presented in Table 11. Bold cells indicate significant improvements in terms of performance measures (Mann-Whitney U-test, p _ value 0:05).

Table 11 shows that a model built through statistical techniques produces a better prediction performance than an expert-based model in both companies. In Company A, we managed to predict bi-weekly releases with 39 percent MRE on average using Model #2, whereas 57 percent of the releases were predicted with less than 30 percent MRE. Compared to model #1, Model #2 also fits the data better, i.e., significantly lower DIC. In Company B, Model #2 also achieves better performance in terms of MdMRE (9 percent decrease), Pred(25), and Pred(30) (11 percent increase in both) compared to Model #1. This shows that models built based on quantitative data may reveal other types of relationships between variables that experts could not cover.

Hybrid models, on the other hand, operate differently in both companies: In Company A, Model #3 achieved significantly better performance than Model #2, by approximately 60 percent improvements in terms of MMRE and MdMRE, and by 75 percent improvement in terms of Pred(25). Therefore, incorporating qualitative evidence into quantitative data through statistical techniques helped practitioners in Company A, to predict post-release defects. Because Gibbs sampling can successfully handle inference in BNs with a mixture of continuous and categorical variables, it is not necessary to apply any discretization technique (based on expert judgement, using a dynamic approach, e.g., [48], [54]) in order to employ the inference algorithm.

TABLE 2: Software Engineering Topics Covered in Primary Studies

KA	#Publications
Software quality	54 (46.15 %)
Software engineering management	31 (26.5 %)
Software design	7 (5.98 %)
Software testing	7 (5.98 %)
Software requirements	5 (4.27 %)
Software construction	5 (4.27 %)
Software maintenance	3 (2.56 %)
Software engineering tools and methods	2 (1.71 %)
Related disciplines	2 (1.71 %)
Software engineering process	1 (0.85 %)
Software configuration management	0

In company B, we observe that hybrid approach (Model #3) does not improve the prediction performance of Model #2, even though we increase the information content by adding anew subnet through surveys with practitioners. There may be more than one reasons for this. One explanation is related to the quality of data collected through surveys. The survey is taken from [55] such that each question corresponds to a node in the requirements specification subnet. These questions might not be appropriate to cover fundamental specification activities in Company B, and thus, the corresponding subnet does not provide valuable information to the whole BN. Another explanation might be related to the software

TABLE 11: Prediction Performance of the BN Models

	Model no.	MMRE	MdMRE	Pred(25)	Pred(30)	DIC
Company A	#1	0.47	0.33	0.39	0.46	417.0
	#2	0.39	0.27	0.48	0.57	384.7
	#3	0.16	0.11	0.84	0.84	278.1
Company B	#1	0.66	0.39	0.33	0.40	326.91
	#2	0.69	0.30	0.44	0.51	280.08
	#3	0.69	0.32	0.42	0.47	284.42

development practices, and the characteristics of the development teamin Company B. Software companies like Company B, with relatively small and cohesive teams that consist of highly experienced staff, have almost no staff turnover. Local data produced by senior developers has been collected consistently through measurement programs, and supporting tools.

Hence, the local data may actually represent expert knowledge effectively in such organizations. Thus, expert knowledge collected through surveys may not add new evidence into the model. In such cases, quantitative data adequately represents software process implementation, and can be preferred over expert judgment during BN construction. However, in companies like A, a mixed data collection approach should be considered to represent both expert knowledge and quantitative data and build hybrid systems.

V. CONCLUSION

In this study, we focus on evidence-based decision-making in software engineering, and its close links with Bayesian decision making. A BN is naturally a good framework for incorporating an integrated summary of the evidence and associated uncertainty. More specifically, BNs have the ability of holding different types of evidence, i.e., observations from real data, statistical distributions, assumptions, and expert judgment, in a single hybrid model. The construction of BNs is flexible with various exact/approximate inference algorithms, and structure learning techniques. Hence, BNs are very popular in some domains, such as computational biology and healthcare. In this research, we aim to observe how BNs are treated in software engineering research, whether the state-of-the-art techniques in our field is different from other disciplines, and how the usage of BNs in software engineering can be improved or expanded. To accomplish this, we first conduct a systematic mapping study on the application of BNs in software engineering and aggregate existing literature with respect to the following four main facets: topics, structure learning, parameter learning, and variable types.

Our systematic mapping on the applications of BNs in software engineering shows that BNs are not well exploited in terms of their capabilities, and used as black box tools by a majority of researchers. Many studies in this field use expert judgment to identify cause-effect relationships of a BN and prefer to transform continuous data into categorical values for an easier inference. Other domains, such as computational biology and healthcare, on the other hand, have utilized BNs by adopting various structure and parameter learning techniques depending on the problem at hand. As researchers practice more on BNs, they managed to build useful and effective models that aid evidence based decision-making in healthcare.

Similar to computational biology and healthcare, we need to make decisions under uncertainty using multiple data sources. Thus, we propose a Hybrid BN, which utilizes techniques that are widely used in other disciplines, such as dependence analysis for structure learning and Gibbs sampling for inference, on a combination of qualitative and quantitative evidence. We have conducted an empirical study with two industrial projects to evaluate the hybrid model in the context of software reliability prediction. Our analysis shows that BNs learned from a mixed data with different learning techniques give practitioners more flexibility, so that the models become less dependent on experts. Furthermore, hybrid models aggregate different types of evidence, infer associated uncertainty, and improve decision- making in software engineering.

REFERENCES

[1] B. Kitchenham, P. Brereton, D. Budgen, M. Turner, J. Bailey, and S.Linkman, "Systematic literature reviews in software engineering a systematic literature review," Inform. Softw. Technol., vol. 51, pp. 7–15, 2009.

[2] B. A. Kitchenham, D. Budgen, and O. P. Brereton, "Using mapping studies as the basis for further research—a participantobserver case study," Inform. Softw. Technol., vol. 53, no. 6, pp. 638–651, Jun. 2011.

[3] D. Heckerman, "A tutorial on learning with Bayesian networks," Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR- 95–06, 1995.

[4] R. Schutt. (2012, Oct.). 10 important data science ideas [Online]. Available: http://columbiadatascience.com/2012/10/15/10-important-data-science-ideas/

[5] B. Turhan and A. Bener, "On combining the scattered knowledge: Putting the bricks together," in Proc. Int. Conf. Artif. Intell. Softw. Eng., 2012, pp. 1–3.

[6] D. Wilkinson, "Bayesian methods in bioinformatics and computational systems biology," Briefings Bioinformatics., vol. 8, no. 2, pp.109–116, 2007.

[7] R. Winkler, "Why Bayesian analysis hasn't caught on in healthcare decision making," Int. J. Technol. Assessment Health Care, vol.17, no. 1, pp. 56–66, 2001.

[8] R. Daly, Q. Shen, and S. Aitken, "Learning Bayesian networks:Approaches and issues," Knowl. Eng. Rev., vol. 26, pp. 99–157,2011.

[9] M. Shaw, "What makes good research in software engineering?" Int. J. Softw. Tools Technol. Transfer, vol. 4, pp. 1–7, 2002.

[10] N. E. Fenton, M. Neil, and D. Marquez, "Using Bayesian networks to predict software defects and reliability," J. Risk Rel., vol. 222, pp. 701–712, 2008.

[11] N. Fenton, W. Marsh, M. Neil, P. Cates, S. Forey, and M. Tailor, "Making resource decisions for software projects," in Proc. 26th Int. Conf. Softw. Eng., 2004, pp. 397–406.

[12] M. Fineman, N. Fenton, and L. Radlinski, "Modelling project trade-off using Bayesian networks," in Proc. Int. Conf. Comput. Intell. Softw. Eng., 2009, pp. 1–4.