Requirements Eng (2016) 21:461–480
DOI 10.1007/s00766-015-0225-3

ORIGINAL ARTICLE

Bayesian networks for enhancement of requirements engineering: a literature review

Isabel M. del Águila[1] · José del Sagrado[2]

Received: 7 May 2014 / Accepted: 7 April 2015 / Published online: 8 May 2015

Abstract Requirements analysis is the software engineering stage that is closest to the users' world. It also involves tasks that are knowledge intensive. Thus, the use of Bayesian networks (BNs) to model this knowledge would be a valuable aid. These probabilistic models could manage the imprecision and ambiguities usually present in requirements engineering (RE). In this work, we conduct a literature review focusing on where and how BNs are applied on subareas of RE in order to identify which gaps remain uncovered and which methods might engineers employ to incorporate this intelligent technique into their own requirements processes. The scarcity of identified studies (there are only 20) suggests that not all RE areas have been properly investigated in the literature. The evidence available for adopting BNs into RE is sufficiently mature yet the methods applied are not easily translatable to other topics. Nonetheless, there are enough studies supporting the applicability of synergistic cooperation between RE and BNs. This work provides a background for understanding the current state of research encompassing RE and BNs. Functional, non-functional and -ilities requirements artifacts are enhanced by the use of BNs. These models were obtained by interacting with experts or by learning from databases. The most common criticism from the point of view of BN experts is that the models lack validation, whereas requirements engineers point to the lack of a clear application method for BNs and the lack of tools for incorporating them as built-in help functions.

Keywords Bayesian networks · Requirements engineering · Literature review

✉ Isabel M. del Águila
imaguila@ual.es

José del Sagrado
jsagrado@ual.es

[1] Departamento de Informática, Universidad de Almería, Ctra. Sacramento s/n, Edificio Científico Técnico III Matemáticas e Informática (CITE III), Planta 2, Despacho 2.19.0, 04120 Almería, Spain

[2] Departamento de Informática, Universidad de Almería, Ctra. Sacramento s/n, Edificio Científico Técnico III Matemáticas e Informática (CITE III), Planta 2, Despacho 2.16.1, 04120 Almería, Spain

1 Introduction

Software development and, especially, software requirements analysis are knowledge-intensive tasks which are mainly supported by human endeavor [2]; that is to say, their processes focus on knowledge and experience. So, when the goal is to improve process productivity and the quality of the resulting products (i.e., a software application), artificial intelligence becomes a key distinguishing feature [33, 51]. In particular, the approach based on Bayesian networks is becoming increasingly popular within the software engineering community, as these networks are capable of providing more appropriate solutions to some of the problems encountered in this field, [7, 25, 27, 59, 72].

There are several research studies in this area, but they are fragmented over different research communities and domains. Our objective is to integrate existing research by performing a literature review. This task includes the identification and articulation of the relation between requirements-related tasks and BNs, carrying out a critical evaluation. Bayesian networks are models that combine graph theory and probability, and they have been successfully applied to decision making in domains where uncertainty plays an important role. RE is one of these domains with uncertainty [38, 68]. One of the major advantages of BNs is their intuitive and graphical representation of the causal relationships between data, allowing better understanding of the domain.

Given that software engineering is a wide-ranging field [1], we have focused our attention on software requirements, the knowledge area closest to the client in a software development project. Requirements engineering plays a central role in software development because obtaining these requirements is a fuzzy step in the software development lifecycle, where a set of informal ideas (in which ambiguity and imprecision are the rule, not the exception) have to be translated into formal expressions [11, 53]. Therefore, the application of any technique that allows developers to represent and manage ambiguity and imprecision would be a valuable aid in requirements-related tasks [6].

The aim of our work is to investigate the synergistic collaboration between the requirements stage in software development and BNs. A literature review is a systematic method for identifying, evaluating and interpreting the work of researchers, scholars and practitioners in the field. Our objective is to provide a description of the kind of research activity undertaken within BNs and RE. We present this work as a systematic approach to analyze the existing literature and report the current status of this field because we have discovered that there is little evidence in this domain; that is, there are fewer studies than we initially expected. Moreover, it might help us to identify knowledge gaps in this area and define guidelines that allow researchers to apply this technique in the requirements stage, identifying those issues that have not yet been covered.

The rest of the paper is organized into five sections. Section 2 summarizes the basic BN concepts and RE keywords. Section 3 describes the method applied in this work for building our literature study—in particular, we describe the protocol, the questions raised, the search strategy applied and the main findings related to the search process. In Sect. 4, the results are discussed according to the research questions they refer to as well as key findings for practitioners. Finally, in Sect. 5, various conclusions are made.

2 Bayesian networks and requirements engineering

Defining software requirements is recognized as critical for a software project's success. An important factor in software development failure is insufficient or erroneous requirements management [24, 30]. The computer-programming community defines requirements activity as engineering in itself. Therefore, requirements engineering can be defined as the set of processes required for reaching an agreement between developers, customers and users regarding the intended functionality of a planned system; together with the acceptance criteria definition that allows stakeholders to decide whether the complete system is valid or not.

The requirements stage consists of several activities that have to be carried out in a software development project. These activities can be organized as a workflow (see Fig. 1). This workflow unifies the main methodological approaches. It starts with a feasibility study that is constructed in a way that determines the project scope and the available resources. After this, software developers execute an elicitation cycle, and then analyze, specify and validate software requirements until ending up with a valid software requirements specification. This document subsequently serves as the baseline for the software development steps [36].

Requirements are elicited from users through interviews and other techniques such as questionnaires or brainstorming. This often proves a complex task because activities requiring human communication usually imply problems in understanding, whereas the concept of requirements is to define without ambiguity what the system is expected to do. Consequently, developers are usually tasked with analyzing and refining requirements in order to obtain a valid baseline. This process can be undertaken by applying AI techniques, for example intelligent systems capable of performing text analyses to extract the requirements to be implemented.

The requirements captured are gathered in a document or its electronic equivalent, known as Software Requirements Specification (SRS) [36]. This task is performed within the requirements specification stage. When approaches to perform this were first attempted, computers were used as word processors. However, the management and maintenance of a large set of requirements using this format proved tedious and prone to error. Software tools subsequently appeared to solve this problem, providing complete environments, supported by databases, to allow effective requirements management within a software project.

Requirements validation is performed to check if the elicited and specified requirements present any inconsistencies, whether the information is complete or not, and whether there are any ambiguities in the system definition. This process (elicitation–analysis–specification–validation) is carried out over several iterations until deciding if the requirements specification has been successfully completed in order to move on to the next activity.

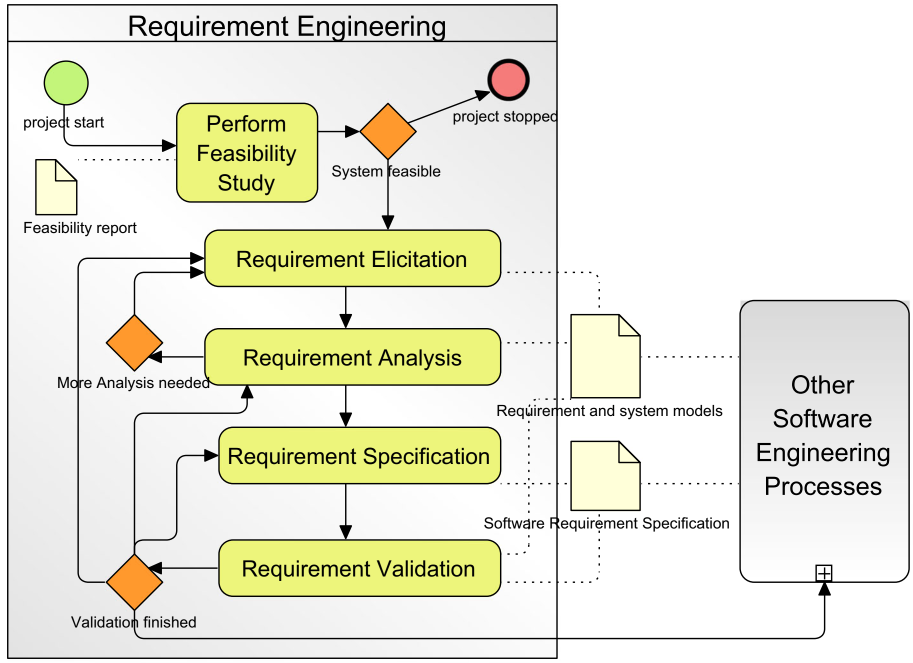


Fig. 1 Overview of a requirements engineering workflow

Table 1 Requirements engineering keywords taxonomy

The IEEE Computer Society-Keywords		
Software Engineering	Requirement / Specification	Analysis
		Elicitation methods
		Languages
		Management
		Methodologies
		Process
		Specification
		Tools
		Validation

RE processes vary widely depending on the type of system under development (e.g., an app or an operating system) and on the specific practice for the developing requirements organization (e.g., scrum or heavy-weight methodologies). On the other hand, there are many artifacts involved in these processes that have to be discovered, documented and validated, such as actors, functional and non-functional requirements and validation criteria. As a reference, we can use the keywords taxonomy proposed by the IEEE Computer Society and the list of topics for the software requirements knowledge area described in SWEBOK [1] (see Tables 1, 2).

A Bayesian network handles inferences from incomplete data entries, providing the decision maker with a way to conduct analyses and propose solutions for a complex process, as is the case with the RE process. A BN is a probabilistic graphical model that represents a set of random variables and their conditional dependencies via a directed acyclic graph. Because a BN is a complete model for the variables and their relations, it can be used for answering probabilistic queries about them.

Formally, a BN [37, 42, 58] is a pair $(G, \mathbf{P})$ where

- $G = (\mathbf{V}, \mathbf{E})$ is a directed acyclic graph whose set of nodes $\mathbf{V} = \{\mathbf{X_1}, \mathbf{X_2}, \ldots, \mathbf{X_n}\}$ represents the system variables and whose set of arcs $\mathbf{E}$ represents direct dependence relations between the variables, and
- $\mathbf{P}$ is a set of conditional probability distributions containing a conditional probability distribution $P(X_i|pa(X_i))$ for each variable X_i given its set of parents $pa(X_i)$ in the graph.

The joint probability distribution over $\mathbf{V}$ can be recovered from this set $\mathbf{P}$ of conditional probability distributions applying the chain rule as:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid pa(X_i)). \quad (1)$$

Probabilistic reasoning in BNs consists of computing the posterior probability distributions of certain variables of interest $\mathbf{V_I}$ after being given some observed variables $\mathbf{V_E}$ (this set of findings is called *evidence*), $P(\mathbf{V_I}|\mathbf{V_E})$. This process is performed via a flow of information through the network in any direction. If we give a *causal interpretation* to the links in the network (i.e., for an arc $X_i \rightarrow X_j$ we say that X_j is a *cause* of X_i and X_j is the *effect* of X_i), we can perform several types of reasoning [15, 43]:

Table 2 Requirements engineering processes according to SWEBOK

SWEBOK
Software requirements
Software requirements fundamentals
Definition of a software requirement
Product and process requirements
Functional and non-functional requirements
Emergent properties
Quantifiable requirements
System requirements and software requirements
Specification
Requirements process
Process models
Process actors
Process support and management
Process quality and improvement
Requirements elicitation
Requirements sources
Elicitation techniques
Requirements analysis
Requirements classification
Conceptual modeling
Architectural design and requirements allocation
Requirements negotiation
Requirements specification
The system definition document
System requirements specification
Software requirements specification
Requirements validation
Requirements review
Prototyping
Model validation
Acceptance test
Practical consideration
Iterative nature of the requirements
Process change management
Requirements attributes
Requirements tracing
Measuring requirements

- *Diagnostic* reasoning: the evidence flows in the opposite direction to the arcs, from effects to cause (i.e., some effects receive evidence and some of their causes are the variables of interest).
- *Predictive* reasoning: the evidence flows in the direction of the arcs, from cause to effect (i.e., some causes receive evidence and their effects are the variables of interests).
- *Intercausal* reasoning: the evidence flows in all directions. This type of reasoning involves reasoning about mutual causes of a common effect (i.e., the effect and some of the causes receive evidence and some of the causes are the variables of interests).

Sometimes reasoning does not match any of these types. This situation occurs when diagnostic and predictive reasoning are used simultaneously, i.e., given a certain variable of interest, some of its causes and effects receive evidence simultaneously.

However, the successful application of a BN to a specific domain is not easy. The process of obtaining the graph and the probabilities of a BN can be carried out either *manually*, from expert knowledge on the domain, or *automatically*, from databases. In the first case, the elicitation of probabilities constitutes a bottleneck in the development of BNs [21] and a very difficult task those who are untrained. In the second case, mistakes can appear due to noise, discretization methods or database errors. The use of Bayesian networks in software engineering is not new [26, 27, 46, 13], for example they have been applied to maintenance [13], defect prediction [26] and the implementation of a software project [46].

In contrast to other software engineering activities, RE is inherently difficult. Requirements are originally described in the user world, i.e., in the problem domain, and therefore have to be translated into the solution domain. In other words, “the requirement problem space is less constrained than the software solution space—in fact, it is the requirements definition that helps to delimit the solution space“ [11]. As requirements are prone to change and, usually, generate conflicts between stakeholders, they become a source of instability and uncertainty during the project development. In this environment, defined by the intrinsic nature of the requirements, BNs can be used to represent imprecise knowledge and reasoning under uncertainty.

An example of BN application in an RE task is the work of del Sagrado & del Aguila [17]. This specifically focused on the identification and assessment of risky requirements. BNs are built from data collected on projects such as the NASA-Independent Verification and Validation Facility’s MDP, a repository providing access to software metrics and the associated error data at the function/method level for NASA software development projects. In these datasets, there are three risk levels that make up the different classes used for requirements classification. The constructed BNs define causal relations between several individual indicators obtained from the SRS as well as the risk of each requirement in each dataset. Table 3 includes the indicators (i.e., metrics) for individual requirements that have been included as variables in the BNs.

Figure 2 shows a BN for one of the datasets studied in [17]. The number of conditions (*conditional*) in a requirement has a direct influence on the number of actions

Table 3 BN variables [17]

Variable	Description
Action	Number of actions the requirement needs to be capable of performing
Conditional	If the requirement is addressing more than one condition. (i.e., if, when, in the event of)
Continuance	Phrases such as "the following:" after an imperative and preceding the definition of a lower level requirement specification. The extent to which continuances are used is an indication of whether requirements have been organized and structured
Imperative	Those words and phrases that command something to be provided. "Shall" normally dictates the provision of a functional capability. "Must" or "must not" normally establish performance requirements or constraints. "Will" normally indicates that something will be provided from outside the capability being specified
Option	Those words that give the developer latitude in the implementation of the specification that contains them
Source	Represents the number of sources the requirement will interface with or receive data from
Weak phrase	Clauses that are apt to cause uncertainty and leave room for multiple interpretations
Risk level	Risk level of a given requirement

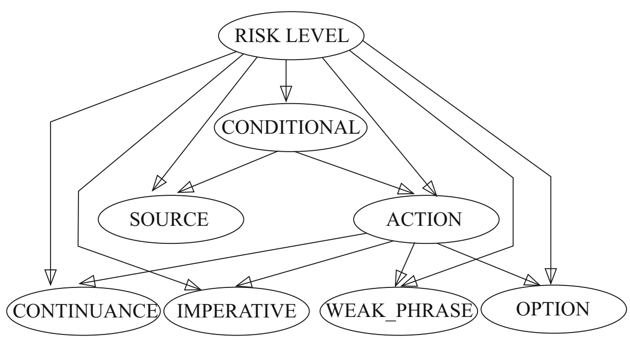


Fig. 2 A Bayesian network for a requirement risk level forecast

and interfaces (*sources*) of that requirement. In addition if a requirement has to be able to develop a certain number of actions, then it should be organized and structured (*continuance*), as well as providing some functional capability and/or be subject to restrictions (*imperative*). Moreover, if a requirement presents optionality, then it should cover all options and translate into more actions. Finally, imprecision (*weak phrase*) is associated with not having a clear understanding of the actions. The aim is to forecast the risk level of a given requirement characterized by the values returned from the metrics applied.

In this case, the aim is to forecast the risk level of a given requirement characterized by the values returned from the metrics applied. Therefore, once the BN is constructed, it will be used to compute the most probable value for this requirement's risk level by carrying out an inference process giving its metric values. Now consider the situation in which the requirement under study needs to be able to perform an action—its description contains one imperative term and it receives data from two different sources. In order to assess the risk level of this requirement, the above information (evidence) is propagated through the BN in an inference process. The BN responds with a risk level of 2, which is the value obtaining the highest probability (i.e., 92.6 %), (see [17] for details).

Despite difficulties in applying the BN, there are several studies that explore their use in different areas of RE. Even though software engineers and academic researchers collaborate to tackle this subject from different angles, a definitive approach has not been achieved so far. Until now, no literature review has been carried out to bring together the works published on this subject. A systematic analysis of the literature could open up a new range of possibilities generating new types of knowledge within a specific domain and describing all of the research that has been undertaken. Practitioners have insufficient evidence to confirm the suitability, the limits, the quality, the cost or the associated risk of using BNs in the RE domain. Our objective is to identify, evaluate and interpret the relevant research to help define a set of good practices that can be followed by practitioners in order to successfully apply BNs to RE activities.

3 Methods

There are three main stages in making a systematic review: the planning, the reviewing itself and the reporting of results [23, 39]. Each stage includes a set of tasks, or processes, that the reviewers have to perform. Table 4 collates the task descriptions and the results that have to be built into each stage. The details are described in the following subsections.

3.1 Protocol followed

The protocol has been designed over several discussion meetings by the two authors and, it addresses decisions regarding the research questions, data source, search strategy, inclusion criteria, study selection, data extraction, quality assessment and data synthesis.

Table 4 Adaptation of the literature review stages

Processes/tasks	Results
Planning	
Formulating the review questions Translating a relevant problem or an information need into an answerable question	Questions, RQ
Formulating the search strategy Defining the information sources, search words and inclusion/exclusion criteria	Sources
Formalizing a protocol This is needed to write a detailed protocol document, and to train all reviewers thus ensuring consistent execution	Search words Selection criteria Quality questions Review Protocol
Conducting	
Searching for the literature The reviewers have to specify and describe the details of the literature search, and as well as explain search comprehensiveness of the search was assured, using the appropriate search chains	Included studies
Selection of primary studies Reviewers need to apply the correct inclusion/exclusion criteria. This step requires reviewers to be explicit about which studies were considered for review, and which ones discounted	Studies scores
Quality appraisal A critical appraisal of each article's suitability, identifying them for their rigor and relevance. All articles included need to be scored for their quality level	Forms with data
Data extraction Reviewers need to systematically extract the applicable information from each study	Forms with subjective information
Reporting	
Synthesis and summary of study results This involves combining the data extracted from the studies using appropriate techniques, whether quantitative, qualitative or both	Report
Writing the review The review process needs to be reported in sufficient detail for the review results to be independently reproducible	

We follow the P-I-C-O (Population, Intervention, Comparison, Outcome) framework adopted as standard [41, 57]. Although this approach was developed around medicine and was therefore designed for clinical studies, it can be adapted for any research context. Table 5 collates the definition of each P-I-C-O element and its adaptation to this review. The research questions (RQ) have been defined to provide a wide overview of the research area. Study selection criteria allow us to assess the relevance of a paper according to the research questions. In this review, we search for evidence of how BNs assist RE. So we focus on RE areas and RE products or artifacts. There are papers related to software architecture (even derived from requirements) [74] or risks in general (i.e., without a specific kind of risk) [27] that will be out of the scope of this review because, even though assisting in a software development project, these are not RE specific.

Quality questions are a way of weighting the importance of individual studies. Researchers use these questions as a means to measure each paper against the others. Kitchenham [41] proposes certain category elements to measure a study: design, conduct, analysis and conclusion. Design and conduct quality questions are related to how the paper is built (e.g., clarity, proposal, context or justification). Conclusion questions are about the value added or the applicability of the results obtained. The analysis category deals with the study domain (e.g., How is the BN built? Which RE knowledge area is covered?).

Other issues related to the protocol are the decision adopted when a dis-agreement occurs and what to do when a study is not peer-reviewed or is inaccessible. Disagreements will be solved through discussion meetings and, if they persist, we consult another expert in our group (although, for this review, we did not have to do this). In the case of a paper being inaccessible, we excluded it after failing to get hold of it by various methods (i.e., emailing the authors).

3.2 Research questions

Our objective is to identify experiences and practices for applying BNs to RE. In order to achieve this goal, we formulated four research questions (RQ):

- RQ1. What are the knowledge areas of RE to which BNs have been applied?
  The aim of this question is to identify if all the RE knowledge areas have had BNs applied, or if studies have only been concentrated on some RE specific tasks or processes. We want to know the scope of BN application to RE.

Table 5 Adaptation of the P-I-C-O framework

	Description	Adaptation to this work
Population	What is the problem that you have come for and need an answer to?	RE knowledge areas
Intervention	What information do you want to capture?	BNs and their reasoning
	What is it that you wish to observe?	
Comparison	What is the main alternative? (If appropriate)	Enhancement of RE artifacts and processes
	Has your review captured all components thus far?	
Outcome	What are you trying to accomplish, measure, improve, effect?	How to build a BN in the RE domain?
	What is it that you will be measuring, observing, assessing?	What are BN usage gaps in RE?

- RQ2. Which reasons do researchers give for combining RE and BNs?
  The answer to this question allows us to identify why BNs are a good approach in solving RE problems.
- RQ3. Is there any method that shows how to adopt BNs in RE processes?
  This question is concerned with identifying the steps that have to be undertaken to develop the qualitative and quantitative part of the BNs in RE areas.
- RQ4. What are the advantages and disadvantages of using BNs in RE?
  This question allows us to evaluate the effectiveness that BN application improvements have on RE.

3.3 Search strategy

This review is primarily focused on finding published papers in journals and conference proceedings from 1990 to 2013. The search strategy consisted of applying three issues related to the review: sources where the search is going to be performed, keywords that define the search query and the criteria for including or excluding studies from the review.

3.3.1 Sources

- Elsevier SciVerse is a set of databases for peer-reviewed literature containing abstracts, citations and full-text journal articles along with book chapters.
- Thomson Reuters Web of Knowledge is a research platform for information in the sciences, social sciences, arts and humanities.
- The IEEE Xplore digital library is a resource for discovery and access to scientific and technical content published by the IEEE (Institute of Electrical and Electronics Engineers) and the Institution of Engineering and Technology.
- SpringerLink is an online platform that provides full-text journals and books published by Springer-Verlag and other editors, such as Urban and Vogel, Steinkop, Birkhauser and Kluwer. It includes prestigious multidisciplinary magazines and books concentrated on areas of science and medicine.
- The ACM Digital Library (DL) is a collection of full-text articles and bibliographic records covering the fields of computing and information technology. The full-text database includes the complete collection of ACM publications, including journals, conference proceedings, magazines, newsletters and multimedia titles.
- Google Scholar is a freely accessible web search engine that indexes full-text scholarly literature across an array of publishing formats and disciplines.

3.3.2 Search terms

The search string had to comprise the P-I-C-O elements [57] integrated by *and* connector. For our study this translated into the following search string:

(”software requirements” OR ”requirements engineering” OR ”requirements elicitation” OR ”requirements analysis” OR ”requirements model” OR ”requirements specification” OR ”requirements validation” OR ”requirements management” OR ”requirements trace” OR ”requirements process” OR ”requirements tools” OR ”functional requirements” OR ”no functional requirements” OR ”requirements negotiation” OR ”emergent properties” OR ”requirements measuring” OR) AND (Bayesian Network or Bayesian belief network)

These terms were adjusted to accommodate the different database search engines since each uses its own syntax for defining the search string to query the database. Once the search string was defined, the first stage consisted of performing a database search, which initially covered the title, abstract and keywords. However, in the case of IEEE Explore and SpringerLink, the search was carried out on full text in order to enlarge the search space. With Google Scholar, the situation was completely different because it returns millions of irrelevant records, nevertheless, it returned none if the search was restricted to title and abstract. Thus, its results were used as a guide for opportunistic searches, as is done in other systematic reviews [18].

Once the search strings were defined and executed on the selected search engines, we performed syntactic filtering. A syntactic filter removes duplicates and excludes forewords, references to workshops and full conference proceedings books. In order to complete these searches, we had to manually execute a second search based on references and authors found in primary studies.

As a result of the above search process, we obtained a large number of potentially eligible records. The study selection process should be carried out in such a way that risks, errors and biases are minimized. Each eligible record needs to be assessed against a set of given criteria before inclusion into the review.

3.3.3 Inclusion and exclusion criteria

Once the studies were located, we extracted two sets of features based on the research questions: one for the inclusion criteria and the other for the exclusion criteria:

A study was included in the review if:

- it used Bayesian networks in any Requirements Engineering process and
- it was written in English.

On the other hand, a study was excluded from the review if:

- it was not specific to Requirements Engineering or any of its areas,
- it appeared in references or related works, as background knowledge,
- it corresponded to an editorial, preface, summary or whole proceedings book.

We carried out the review following the workflow shown in Fig. 3, after applying the search process. As a result, we retrieved 614 studies. Only 67 of them met the selection criteria. This candidate list was extended with four extra papers gathered from a deep review of the references lists of the candidate papers. Due to duplicity and unavailability, only 50 studies were retrieved (when exist several versions of the same work, we considered only the most extended one). Each study in the candidate list was reviewed and evaluated separately by both authors. Its relevance, with respect to the review, was measured applying quality assessment criteria. Those publications that failed to achieve a sufficient score were eliminated from the review. As a consequence, 29 papers remained for review. Table 6 shows the distribution of these studies, based on the databases from which they were extracted. The following section describes the quality assessment details and the assessment process.

3.4 Study suitability assessment

This section describes how to refine the results of the search process by applying a quality question, as a guide for interpreting the findings, instead of as a weighting schema of the collected data.

3.4.1 Quality assessment

The studies were evaluated using eight questions derived from those proposed in [22, 40, 69] and following the framework described in Sect. 3.1 [41]:

- QA1—Is there an adequate description of the proposed contribution, method or approach?
- QA2—Are the aims and objectives clearly reported, including a justification for why the study was undertaken?
- QA3—Is it described how the BN was built?
- QA4—Is the qualitative part of the BN properly described?
- QA5—Is the quantitative part of the BN properly described?
- QA6—Is the RE knowledge area enhanced by the proposal clearly indicated?
- QA7—Is it described how to use the BN?
- QA8—Is the BN used alone or is it embedded in a management tool?

Each question can be answered: “yes”, “partly” or “no” and each answer is scored with 1, 0.5 and 0, respectively. QA1 and QA2 belong to the design and quality conduct questions category; these have a “yes” value when the objectives and proposal of the paper are clearly included within them; “no” when they are not defined and “partly” if they are only implicit. QA3, QA4, QA5 and QA6 are analysis questions as they focus on the research. If we want to replicate the application of BN to any other RE knowledge area, we have to know explicitly (“yes”), or at least roughly (“partly”), how all the elements of the BN have been elicited. Finally, QA7 and QA8 help in the assessment of the added value obtained from applying the

Fig. 3 Search workflow

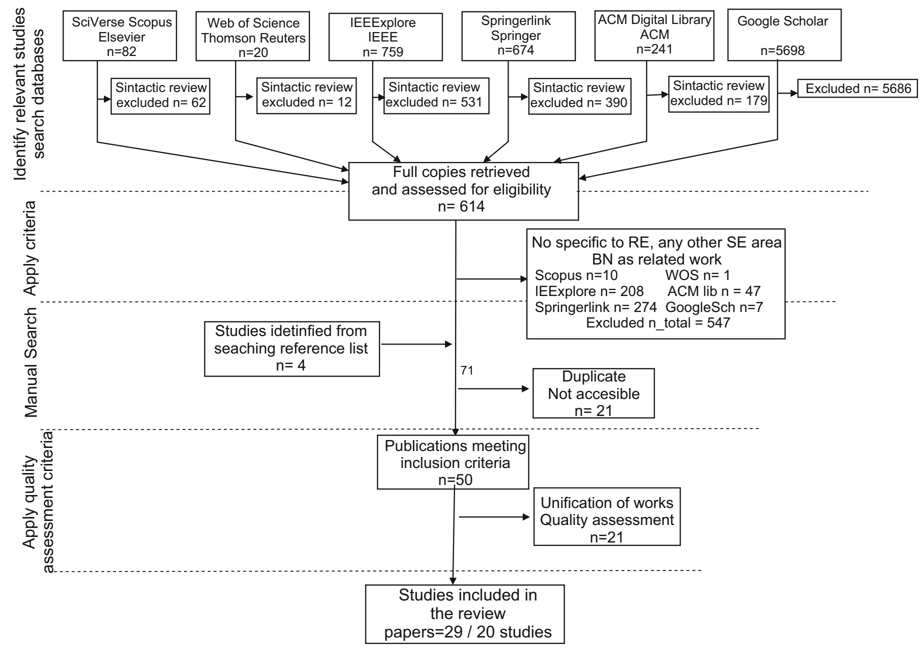
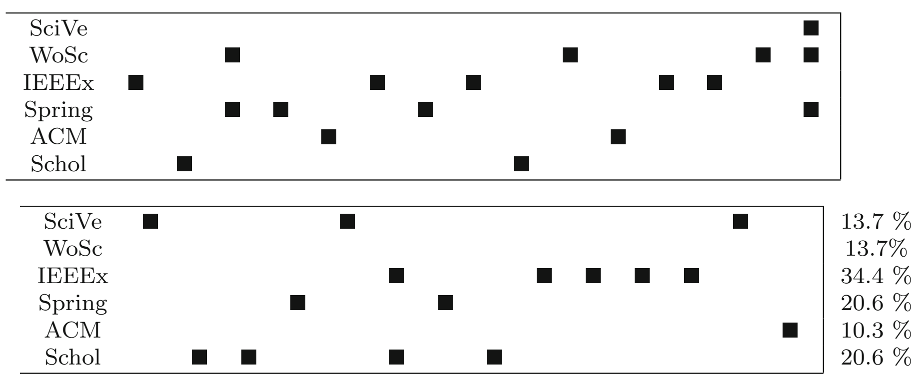


Table 6 Studies by search engine, each symbol represents one paper



BN. Each study has an associated score that is calculated by summing up the scores obtained in each QA question. Both authors individually assessed each study, and any disagreements were discussed and solved assigning a consensus score (see Table 8).

3.4.2 *Data extraction*

The extraction process consisted of identifying the required data to answer the research questions. We created a set of forms to collect information from the studies under review. One of these forms collated general information about the context of the study (i.e., qualitative and quantitative objective data). Another form collated subjective information related to the impressions and conclusions reached by the reviewers after reading the study. All the gathered information aimed at providing answers to the research questions and identifying the RE knowledge areas to which the BNs had been applied (Table 8). These data also collated the problems and limitations reported by the authors and a summary of the results of the study, how the BN had been built (e.g., manually, automatically or a mixed approach labeled semiautomatically) and the type of reasoning (e.g., diagnostic, predictive or intercausal). The data extracted from each study are shown in Tables 7, 9, and 10.

3.5 Synthesis of findings

The search process, previously described, identified 29 articles, published from 1999 to 2013, that dealt with the application of BNs to RE. Out of these, 6 (6/29 = 20.6 %) papers were published in journals; 17 (17/29 = 58.6 %) papers appeared in conference proceedings, symposia or workshops; 5 (5/29 = 17.2 %) came from book chapters and one (1/29 = 3.5 %) was a technical report. Subsequently, those papers that described similar works were clustered together into 20 studies—this was the case with studies S2, S6, S7, S8, S10 and S16. Studies S2, S6 and S10 include works that show different points of view on the same research results. Studies S7 and S16 gathered new versions of a previous work, such as papers [29] and [19] respectively. We should highlight S8, which unifies many works [28, 31, 32, 64, 65]. This study represents in-depth research work that has been developed over several years. There are many studies that use NASA's IV&V Program; this was established in 1993 as part of an agency-wide strategy to provide the highest achievable levels of safety and cost-effectiveness for mission critical software (S7, S15, S16 and S19).

Table 7 Studies included in the review

Id	Ref	Date	Publication site	Type	Title
S1	[50]	2010	INCoS	Conference	A Bayesian Based Method for Agile Software Development Release Planning and Project Health Monitoring
S2	[16]	2009	ACIR Book Series	Book chapter	A Bayesian Network for Predicting the Need for a Requirements Review
	[17]	2011	LNCS vol 7023, CAEPIA	Book chapter	Architecture for the Use of Synergies between Knowledge Engineering and Requirements Engineering
S3	[61]	2011	LNCS vol 6786	Book chapter	A Framework for Integrated Software Quality Prediction Using Bayesian Nets
S4	[8]	1999	UAI	Conference	An Application of Uncertain Reasoning to Requirements Engineering
S5	[63]	2014	System and Soft	Journal	An Evaluation Model for Dependability of Internet-scale Software on Basis of Bayesian Networks
S6	[66]	2009	LNCS vol 5854, WISM	Book chapter	An Expert System Based Approach to Modeling and Selecting Requirement Engineering Techniques
	[67]	2009	ICMLA	Conference	Requirement Engineering Techniques Selection and Modeling—An Expert System-Based Approach
S7	[29]	2006	ISSRE	Workshop	Analysis of Milestone Readiness Levels During the Software Requirements Development Phase
	[56]	2005	NASA/IEEE	Workshop	Bayesian Networks applied to Software IV&V
S8	[28]	1999	TOCHI	Journal	An Impact Analysis Method for Safety- Critical User Interface Design
	[64]	1999	RE Symposium	Conference	Human Errors and System Requirements
	[65]	1999	RE IEEE Conf.	Conference	Validating Functional System Requirements with Scenarios
	[31]	2005	TSE	Journal	Scenario-Based Assessment of Nonfunctional Requirements
	[32]	2005	RE Journal	Journal	The system reliability analyzer tool
S9	[75]	2012	2012 ISRA IEEE	Symposium	Application of Active Learning Strategy and Formalization Method in Requirement Analysis
S10	[44]	2000	SEKE	Conference	Breaking the Knowledge Bottleneck for Bayesian Networks Using Language (UML) Artifacts
	[45]	2000	SEKE	Conference	Development of Bayesian Networks from Unified Modeling Language Artifacts
S11	[60]	2002	PROFES	Conference	Evaluating Evolutionary Software Systems
S12	[55]	2011	IST	Journal	Exploring a Bayesian and linear approach to requirements traceability
S13	[54]	1998	IEEE SMC	Conference	Framework For Hardware/Software Partitioning Utilizing Bayesian Belief Networks
S14	[9]	2006	LNCS vol 4166, SAFECOMP	Book chapter	Gaining Confidence in the Software Development Process Using Expert Systems
S15	[49]	2006	West Virginia Univ.	Tech.report	Improving IV&V Techniques Through the Analysis of Project Anomalies
S16	[20]	2005	SEW	Workshop	Is My Software Good Enough to Release? A Probabilistic Assessment Methodology
	[19]	2003	RAMS	Conference	Modeling the "good enough to release" decision using IV&V preference structures and Bayesian belief networks
S17	[73]	2010	COMPSACW	Workshop	Optimizing Requirements Elicitation with an i* and Bayesian Network Integrated Modelling Approach
S18	[47]	2011	SSIRI	Conference	Probabilistic Risk Assessment for Security Requirements: A Preliminary Study
S19	[14]	2011	IJSEKE	Journal	Requirement risk level forecast using Bayesian networks classifiers
S20	[35]	2006	IUI	Conference	Who's asking for help a Bayesian approach to intelligent assistance

Table 8 Quality assessment of the studies

	QA1	QA2	QA3	QA4	QA5	QA6	QA7	QA8	Total (*)
S1	1	1	0.5	1	0.5	0	1	0	5
S2	1	1	1	1	0.5	1	1	0.5	7
S3	1	0.5	1	1	1	0.5	1	0.5	6.5
S4	1	1	0.5	0.5	0	1	0.5	0.5	5
S5	1	0.5	1	1	0.5	0.5	0	0	4.5
S6	1	1	1	1	0.5	1	0	0	5.5
S7	1	1	1	1	1	1	1	0	7
S8	1	1	0.5	1	1	1	1	1	7.5
S9	1	0.5	0.5	0.5	0.5	0	0.5	0	3.5
S10	1	1	1	1	0.5	1	1	0.5	7
S11	1	0.5	1	0.5	0.5	1	0	0	4.5
S12	1	1	1	1	1	1	1	1	8
S13	1	1	0.5	1	0	0.5	0	0	4
S14	1	1	0.5	1	0.5	0.5	1	0	5.5
S15	1	0.5	1	1	0.5	0.5	1	1	6.5
S16	1	1	0.5	1	0.5	1	1	0.5	6.5
S17	1	0.5	1	1	0	1	0	0	4.5
S18	1	1	1	1	0.5	1	1	0	6.5
S19	1	1	1	1	1	1	0	0	6
S20	1	1	1	0.5	1	0.5	0	0	5

(*) Quality level: $[0, 3.5)$ Bad , $[3.5, 4.5]$ Border line, $(4.5, 6.5]$ Acceptable, $[6.5, 8]$ Good

Regarding the quality of the selected studies, we can check in Table 8 that the studies with a enough level of quality are 15 out of 20 taking into account that in the study selection strategy, we required the quality score of a study to exceed or be equal to 3.5, so five studies are in the border line of quality. Table 9 shows a summary of data extracted from the studies that identifies the RE area, type of requirements, the process applied to build the BN, the type of reasoning used and whether the BN was validated or not.

4 Results and discussion

Once the data had been extracted, the studies were analyzed to seek answers to the research questions. Table 11 focuses on the answers of each question as summary of the discussion.

4.1 RQ1: What are the RE knowledge areas to which BNs have been applied?

The process followed in the requirements stage is articulated by the execution of several activities and has been defined by different authors with few variations in their requirements process [1, 3, 70] (see Tables 1, 2). These cyclic activities (*elicitation*, *analysis*, *specification* and *validation*) are carried out through several iterations in order to complete the requirements definition, before moving toward the next development task. In addition, these knowledge areas are covered under the umbrella of *requirements management tasks'*, which are transversal to the process (i.e., risk assessment, change control, metrics and requirements tracing).

The studies in this review focus on the application of BNs to one of these RE activities (see second column in Table 9). Visually, the Fig. 4 shows the studies distribution according to the RE knowledge areas from Table 2. The highest values are related to requirements validation, because in this category, we have included both *quality assurance* studies and those only related to *validation*.

Moreover, the concept of requirement, in its broadest sense, must be understood as a logical unit of behavior, and specified by including functional and quality aspects—we also use the concept of feature [62]. But a requirement has different dimensions. That is to say, the meaning of "requirement" is narrowed down with defining labels: system, hardware, software, user, client, functional, non-functional, performance, etc. This gives us the opportunity to make a classification of the studies according to the sort of requirements on which the study is focused. Figure 4 shows the distribution of the studies according to the most extended requirements taxonomy: functional or non-functional. *Functional requirements*, expressed by case use or

Table 9 Summary of data extracted from the studies

	RE area	Req category	↑	↓	↕	How is built? quail. | quant.	Map.	Val.
S1	Req process	Functional			✔	man | man		
	Quality assurance							
S2	Req specification	Functional	✔	✔	✔	man | man		
S3	Quality assurance	-ilities		–		man | man		•
S4	Req analysis	Functional		✔		man | auto	•	
	Req manag. Tracing							
S5	Req analysis. Allocation	Functional		–		man | auto	•	
	Quality assurance	-ilities						
S6	Req process	–		–		man | auto		•
S7	Req specification	Functional		✔		semiauto	•	
	Req manag. Metrics							
S8	Req elicitation	-ilities	✔	✔	✔	man | semi		•
	Req validation	Functional						
S9	Req elicitation	Functional		✔		man | man		
S10	Req validation	-ilities		✔		man | man	•	
S11	Req validation	-ilities		✔		man | man		
S12	Req manag. Tracing	Functional		✔		auto | auto		•
	Req manag. Changes ctrl							
S13	Req analysis. Allocation	Functional		✔		man | man		
S14	Quality assurance	–		✔		man | man		
S15	Quality assurance	–		–		semi | auto		•
S16	Req validation	-ilities		✔		man | man		•
	Quality assurance							
S17	Req elicitation	Functional	✔	✔	✔	semiauto	•	
	Req analysis							
S18	Req management	-ilities		✔		man | man		
S19	Req manag. Risk	Functional		✔		auto | auto		•
S20	Req elicitation	Functional		✔		man | auto		

↓ Predictive reasoning, ↑ Diagnostic reasoning, ↕ Intercausal reasoning, Map. mapping approach, Val. Validation executed

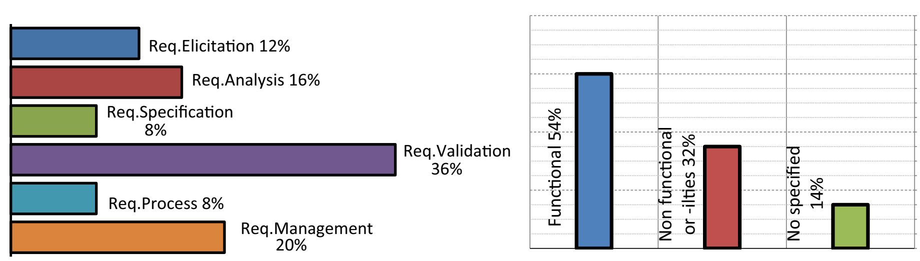


Fig. 4 Distribution of studies according to the RE area and the kinds of requirements

using textual descriptions, were the focus of 54 % of the studies: S1, S2, S4, S5, S7, S8, S9, S12, S13, S17, S19 and S20. *Non-functional requirements*, often called qualities or -ilities, are addressed in 32 % of the selected studies S3, S5, S8, S10, S11, S16 and S18. Finally, three studies (S6, S14, S15) do not clearly specify the kind of requirements because they relate to the overall *requirements process*, helping in the selection of the best RE technique for each project selecting between agile, traditional or product line, S6; or because they relate to RE products quality assessment S14, S15.

Some studies do not exclusively belong to RE, because they deal with several software engineering areas or topic, but did specifically include a reference to this development stage S1, S14 and S15. Additionally, we found studies that proposed general approaches for applying BNs in order to enhance the final results of the software development processes S14 and [17] in S2.

Now, let us take a look at the second level of RE-related topics (see Table 2). These areas are highlighted in the text. There are a group of papers related to *tracing* and *change management* S4, S10 and S12. S4 examines the use of BNs in tackling the problem of translating user requirements into system requirements S4. The S12 study determines the relevance of traceability links between entities for a scenario where a software developer can work on multiple code artifacts to achieve different system features or use cases. The trace validation established between use cases and other UML artifacts is treated in S10.

Inside the Requirement analysis level, there are two works related to *architectural design and requirement allocation* S5 and S13. The first defines a methodology for functional partitioning into hardware and software, in accordance with the technology S13; the second defines a model that analyzes the structure of Internet-scale software and establishes a dependability evaluation system for this kind of software S5.

Elicitation-related activities are difficult to execute so some studies focus on *elicitation techniques* S8, S9, S17 and S20. S17 proposes how to improve requirements models elicited from user goals analysis expressed in a graphical language (i.e., i*). Requirements elicitation for user interfaces is dealt with in S20 and specially in the study S8, which develops probabilistic models that predict human and machine reliabilities using given input variables representing the scenario and ranges of environmental conditions.

The *iterative nature of requirements* is discussed in S9 and [16] in S2. These studies evaluate the accuracy of the elicited and specified requirements in order to assist developers in deciding when to start the next stage of the development process. Other management issues, such as risks, are the focus of S18 and S19; these propose probabilistic models for risk assessment related to security requirements S18, and for risk assessment automation S19.

Many studies (see Table 11) are concerned *review requirements to assure the quality* of the obtained products: checking if the product meets the release criteria [20] in S16; providing a probabilistic assessment of the software's overall "good enough to release" status based on processes assessment and product evidence [19] in S16; assessing the readiness of the maturity of requirement specification by combining evidence from diverse sources to evaluate each use case S7; evaluating non-functional requirements S11 or building predictive models regarding the software quality focus on -ilities S3.

We can conclude that there are several RE areas which are not covered such as requirement negotiation or acceptance tests. These areas lie within essential complex communication processes where decisions have to be made, and where support, such as BNs, could be of help. However, the BN technique needs a specific problem to focus on and RE decision making is not sufficiently mature [2]. The studies included in this review are bespoke developments because there are many dimensions to RE decision making. As yet, no closed set of RE decision problems is available that could be addressed using BNs. Table 11 shows how each study covers a specific RE area (marked with a dot) and also the type of requirement treated.

4.2 RQ2: What are the reasons researchers give to combine RE and BNs?

Decision making in RE is considered a knowledge-intensive activity [2, 6]. The aim of applying BNs to any RE area is to support decision making in order to obtain a global benefit for the entire software development process by means of improving the RE stage. BNs have been successfully used in software engineering, especially in risk management [27]. Nonetheless, requirements are the basis for the rest of the software development and can be considered the bricks gluing together different stages in the software project. Thus, any support that manages uncertainty and incompleteness in the RE stage of the decision process is a research challenge.

Experts are quite good at making judgments at the observable level (the lower level) but are less able at a higher level (i.e., the entire software level). This is because making a decision at a higher level requires a more complex process. Decision-makers need to integrate, or synthesize, a range of information coming from lower levels, where usually many variables are involved S13. A BN can provide the theoretical foundation to incorporate accumulated evidence such as this in a unique inference process [28] in S8 and S17. So, for example, researchers obtained predictors that indicate the healthiness of a software project S1, the goodness of the software requirements specification [16] in S2, the software quality S3, software safety integrity S14, if the software is good to release S16, the risk of requirements S18, S19, or an early quality prediction (especially in market-driven development) S11. The predictor core in all these works is a BN. The lower level data are introduced as evidence and the propagation result of this evidence is used by requirements engineers to make decisions. Moreover, BNs are well-suited for capturing vague and uncertain knowledge and are also a good means (i.e., theoretical model) to unify data with different natures (i.e., qualitative as expertise or quantitative as metrics) and different sources, by means of incorporating this data as evidence S6, S8, S12 and S20.

Quantitative measurements are essential in all sciences; there is a continuous effort by computer science practitioners and theoreticians to bring similar approaches to software development. A software metric is a measure of a certain property, of a piece of software or of its specifications. For this reason, many studies in this review are based on requirements metrics as a means of incorporating data into the decision-making process S7, S11 and S19.

Automatically learning the qualitative and quantitative parts of a BN is a challenge which is undertaken when data are available. The various datasets freely distributed by the Predictor Models In Software Engineering (PROMISE) projects (http://promisedata.org) and by the National Aeronautics and Space Administrator (NASA) IV & V Facility—Metrics Data Program (MDP) repository help facilitate the use of learning processes to acquire the knowledge necessary for embedding in a BN. Several of the studies in this work use these datasets for a deep application of many BN algorithms: S7, S15, S16 and S19. The availability of databases promotes a synergy between RE and BNs that is positive for researchers of both disciplines since their research goals are complementary.

There are two main insights resulting from the analysis of these studies into the adoption of BNs in RE. The first is when researchers start with a specific problem that needs RE support and the availability of experts allows the application of BNs. This approach relies on close collaboration between RE and BN experts. RE experts are in charge of formulating the problem and adapting it to the probabilistic method used; the BN experts have to extract the knowledge and select the best way to build the network. Table 11 shows the areas of decision making highlighted by each study. The second insight into constructing a BN capable of assisting in RE tasks is to take the benefits of automatic learning processes from metrics databases. Nonetheless, even though for design and implementation, there are many repositories (e.g., GitHub, Sourceforge) that can be mined, or metrics databases that can be accessed directly (e.g., [34]) at the requirements level, these kinds of resources are not easily available.

4.3 RQ3: Is there any method that shows how to adopt BNs into RE processes?

Several authors have defined the basic steps that have to be performed when constructing a BN [10, 12, 52]. We can unify these steps as follows:

- *Variable identification* (VI). How a given domain is going to be represented as a BN (i.e., what are the variables that are going to be included)?
- *Qualitative Structuring* (QS). The topology of the network captures relations between variables. Which relations are captured by the network topology?
- *Quantitative Elicitation* (QE). The strength of the relations between variables has to be quantified by specifying the conditional probabilities.
- *Validation & Testing* (VT). The suitability of the BN model, along with the job it is designed for, has to be checked.

In all the revised studies, the authors constructed their BN models in the RE domain applying different approaches. The steps for building the BN can be carried out manually or automatically. The columns "how is it built?" and "Map" (i.e., mapping) in Table 9 collate which method was used in each study. Nevertheless, all the studies included in this review have more or less followed, the previously defined steps, with only a few papers including the specific way in which they tackled each one [16] in S2, S3 and S6.

The selection of the variables and the relations between them (i.e., VI and QS steps) have been carried out using three different paths: mapping requirement artifacts as variables in the networks, defining relevant variables from experts (i.e., requirements engineers) or applying machine learning techniques in databases. The first approach is found in S4, S7, S10 and S17 in which the following were mapped as variables: S4 maps user requirements and system requirements, S10 maps use cases and UML classes or S17 maps user goals expressed in i* and use cases. Some of these mappings need to be adjusted because a BN does not allow cyclic relations S4. This approach exploits the inherent relations created during requirements activities, which provides a significant opportunity for creating BNs from requirements artifacts without any additional user intervention. Machine learning is used to execute VI and QS from databases S12, S15 and S19, and the variables and the relations obtained must be validated by the experts. The qualitative part of the BN (i.e., $G = (\mathbf{V}, \mathbf{E})$) can be elicited from domain experts. Several domain experts must be consulted, and the knowledge engineer finds out what the valid structures are that represent the knowledge domain. Due to the human nature of this communication-intensive task, the authors usually need to use linguistic terms S1, or, otherwise, they use a negotiated VI and QS starting point (e.g., standards on software engineering) [16] in S2 and S3.

The quantitative estimation (QE) step acquires numbers and data necessary to estimate the conditional probabilities for the BN model (i.e., $\boldsymbol{P}$). Of course, the acquisition of numbers is not exempt from many real-world problems [21]. The task of estimating $P(X_i|pa(X_i))$ becomes very difficult, not only because of the high number of variables but also because of the various causes, $pa(X_i)$. It is necessary to relate pieces of knowledge where the only connection they have is a common consequence X_i. Despite noise or error susceptibility in databases, from the stakeholder's point of view, the machine learning approach is the easiest method to obtain the BN probability distribution S4, S5, S6, S7, S12 and S19. The other studies all apply a manual QE method, which is not clearly described (the QA5 value in Table 8 has the lowest rates). Also, some authors argue that a learning process would be needed to enhance their results S1.

The purpose of VT is to answer questions such as: Does the structure reflect the problem's fundamental independence relations? What is the level of predictive accuracy acquired? And, is the network sensitive to changes? These questions help in validating the network and understanding how the network can be used in the RE field. It is important not to forget to measure usability and performance in order to know whether the BN meets customer use criteria. Not all the studies explicitly talk about this step (see the last column in Table 9) S3, S6, [31] in S8, S12, S15 and S19. Some authors suggest that BN models need a deeper validation S12. The methods applied in the validation are: simulation S3, classification accuracy S6 and S19, comparing model results with the experts' reported values S12, learning validation methods [31] in S8, S11 and cross-validation S19.

In conclusion, it is worth noting that most studies construct the qualitative part of the BN manually taking standards, metrics or RE artifacts as the starting point. Next, researchers try to identify causal relations between them for the problem at hand. The quantitative part is based mostly on trust in experts although data are used when available. These conclusions are summarized in Table 11 defining four categories for learning probabilities: mapping, from expert, from data, both.

4.4 RQ4: What are the advantages and disadvantages of using BNs in RE?

The RE stage is complicated due to its workflow, which is dynamic, difficult and uncertain. The application of any AI technique to this workflow must run a parallel path in order to facilitate the independent evolution of the AI technique itself and the RE stage [17] in S2. The advantages of using BNs in RE are their ability to reason under uncertainty along with the combination of a graphical representation based on sound mathematics (i.e., probability theory). However, the construction of BNs is not exempt from problems because, if it is carried out manually (using experts), then probabilities elicitation will be a bottleneck, and if carried out automatically (from data), it will not be easy to find good RE data sources that are free from errors.

Each of the studies in this review provides a specific contribution to enhance a particular RE process or artifact. All of them are listed in Table 10 from the point of view of requirements engineers. However, none of them compare their RE solution with an execution that does not use RE. Regarding the RE area, the studies in this review address the disadvantages of using BNs in different ways. In requirements elicitation (S8, S9, S17 and S20), the manual process for BN construction predominates, as this is primarily a human task. This tendency is also present in requirements analysis (S4, S5, S13, S17 and S20), but with the difference that the quantitative part of the BN is built automatically with data gathered from software metrics. Requirements specification (S2 and S7) needs detailed and continuous communication between the project team and the customer up until software completion. This is the reason why, preferably, the BN is constructed manually. Requirements validation (S1, S3, S5, S8, S10, S11, S14, S15 and S16) is basically a task ensuring that the requirement descriptions meet the needs (and the descriptions) given by the customer or any other identified stakeholders, so as to detect problems with the expression of requirements. Related studies conform to this, and in order to automate this task, they build a BN manually. Researchers do not demonstrate any clear preference between manual or automatic ways of building the graphical BN structure for requirements management (S4, S7, S12, S18 and S19) despite the availability of datasets from the NASA IV&V MDP repository, although they rely on the automatic extraction of probability distribution from the data.

One of the main problems detected is that BNs should be embedded in a software requirement management tool in order to be useful to practitioners. This will reduce the workload of requirements managers. However, only a few studies point out that BN models have been implemented (see quality measure QA8). Indeed, only 25 % of the studies have a tool that implements the BN S4, S8, S12 and S15. A deep analysis and evaluation of the tool from a user's point of view is only included in [32] in S8 and S12. Moreover, BNs should seamlessly integrate with the RE stage.

BNs can mix expert opinion and data to build models, and they show explicit dependence relations between variables. The graphical structure (the qualitative part) of a BN allows a representation of the information contained in the model, which is accessible and can easily be interpreted by others. The probability distributions (the quantitative part) of a BN can be obtained subjectively from domain experts (i.e., expert opinions), objectively deriving them from available data or using a combination of both. Furthermore, a BN has the ability to propagate evidence throughout the entire network (probabilistic reasoning), propagating the impact of the available information onto the other model variables. So, we can use them to explore the impact that different configurations have on the rest of variables of the

Table 10 Studies contributions

Id	Contributions
S1 [50]	▹ A model of agile release planning
S2 [16, 17]	▹ REQUISITE, this BN assess goodness of the software requirement specification (SRS)
	▹ An architecture to integrate REQUISITE in a CARE tool
S3 [61]	▹ BaNISoQ, Bayesian network for Integrated Software Quality prediction
S4 [8]	▹ SRW (System Requirement Web) as a concept to represent relations between requirement (implemented as a BN)
	▹ BOSH architecture
S5 [63]	▹ A method for building a mapping BN to evaluate Internet-scale software dependability
S6 [66, 67]	▹ A BN-based expert system for selecting the RE technique (agile, traditional or product line) in future projects.
S7 [29, 56]	▹ A BN model for assessing the readiness of requirements artifacts (i.e., use cases)
S8 [28, 64, 65, 31, 32]	▹ A BN model for estimating human errors in user interface elicitation
	▹ An automatic scenario-based testing tool (SRA, System Requirement Analyzer) focused on non-functional requirements
S9 [75]	▹ An active learning strategy using BN to improve RE
S10 [44, 45]	▹ A BN model in a prototype (BOSH) for verifying the transformation (relations) between use cases and UML classes
S11 [60]	▹ Prometheus (Probabilistic method for early evaluation of non-functional requirements)
S12 [55]	▹ CRI (Continuum of Relevance Index) a extension of Eclipse to monitor trace relations
	▹ A Bayesian technique to model relevance of use cases, develop an artifact associated with requirement traceability links
S13 [54]	▹ A BN-based methodology for functional partitioning into hardware and software classes
S14 [9]	▹ A BN model to predict software safety integrity based on the type of reasoning present in safety standards
S15 [49]	▹ QNET to assess the quality of requirements
S16 [20, 19]	▹ GETR methodology, provides an overall probabilistic assessment of acceptability in software quality assessment
S17 [73]	▹ A method to derive a BN model from i* model. The BN model helps requirement engineers optimize i* models.
S18 [47]	▹ A probabilistic model for risk assessment in security requirements
S19 [14]	▹ A BN model for identifying risky requirements based on requirements metrics
S20 [35]	▹ A method for user modeling assisting in requirement elicitation

models (i.e., given that we want to obtain certain value on a value of interest, we can explore what are the values that the others variables have to adopt). Once the BN model is built, it can be used alone of embedded in a tool (see the third rows in RQ4 in Table 11). However, despite of the advantages (i.e., BNs are based on probability, are not a black box approach, can mix expert opinion and data), the use of BNs in RE is not free of drawbacks. Exploring alternatives is at the heart of requirement and different alternatives produce different results. The RE tasks variability, lack of accuracy and measures lead to limited scope applicability and partial conclusions.

4.5 Limitations

The results obtained are subject to the limitations inherent in a systematic approach to analyze the literature which requires rigor and effort. The rigor is based on the study's teamwork, bias control and systematically defined research process. It is worth discussing the results validity in order to provide a comprehensive understanding of limitations and extent that the work has. In the literature [4, 71], we can find different ways of classifying aspects of validity as well as validity threats. We are going to follow the four types of threats proposed [4], even where their origins are empirical software engineering studies, aspects of validity can be translated to our work.

Construct validity reflects if the measures and scales used have properly captured the concepts they need to represent. The search protocol is defined to prevent the eventual omission of main papers as it includes both automatic and manual searches [5, 48, 69]. Indeed, as mentioned in [5, 22], software engineering keywords are not standardized; even more so in Requirements Engineering and, likewise, with Bayesian networks. Hence, even carefully defining consistent search keywords lead us to launch multiple searches, the results of which must be unified before applying quality criteria—what is called the syntactic filtering stage (see Fig. 3). During the execution of the search workflow, we detected that works under a wider scope (such as software engineering) include topics and proposals related to BNs in requirements stages. However, they have been left out of our review because it is restricted to RE areas.

Table 11 Summary of research questions

		S1 [50]	**S2** [16,17]	**S3** [61]	**S4** [8]	**S5** [63]	**S6** [66,67]	**S7** [29,56]	**S8** [28,64, 65, 31,32]	**S9** [75]	**S10** [44,45]	**S11** [60]	**S12** [55]	**S13** [54]	**S14** [9]	**S15** [49]	**S16** [20,19]	**S17** [73]	**S18** [47]	**S19** [13]	**S20** [35]
RQ1 (*)	**RE activity**																				
	Elicitation								●	●								●			●
	Analysis				●	●								●				●			
	Specification		●					●													
	Validation	●		●		●			●		●	●			●	●	●				
	Process	●					●														
	Management				●			●					●						●	●	
	Type of requirement																				
	Functional	●	●			●		●		●			●	●				●		●	●
	Non func/-ilities			●		●			●		●	●					●		●		
RQ2 (*)	**Decision making about**																				
	Next release	●																			
	SRS		●																		
	Quality			●					●			●			●	●	●		●	●	
	RE artifacts				●	●		●			●		●								
	Methodology						●			●			●	●			●	●			●
RQ3 (**)	Variable id	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	Qual.struct	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	Quant.elicit.	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	●	●	○	○
	Val. & Test(*)			●			●		●				●			●	●			●	
RQ4 (***)	Mix expertise & data	○	○	●	○	●	●	●	●	○	○	○	●	○	○	●	○	●	●	○	●
	Usability	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	Integration	○	○	○	●	○	○	○	●	○	○	○	●	○	○	●	○	○	○	○	○
Cited by		7	1	5	4	1	2	9	150	-	1	8	6	4	5	-	2	-	3	1	31

(*) **RQ1, RQ2, Validation and test** ● Yes
(**) **How the BN is built** ○ from expert, ○ from data, ● both sources, ○ mapping RE artifacts
(***) **Mix expertise & data** ○ expertise, ○ data ● both sources
(***) **Usability** ○ metrics, ○ subjective ○ scenarios
(***) **Used alone or embedded in a tool** ○ alone ● embedded

Internal validity regards the rigor of the protocol followed. We have ensured that we have met the four quality criteria included in [39, 40] given that we have: explicitly defined the inclusion criteria; searched six digital libraries and included additional search strategies; assessed the quality/validity of the included studies by means of the explicit definition of some quality criteria; and finally, described and presented data extracted from each study (see the above sections).

Conclusion validity is concerned with the relation between approach and outcome, ensuring that there is a correct relation between them. For our study, this is a major threat to validity because there are fewer studies than we initially expected—thus we had to restrict this work to an analysis of the literature to report on the current status of this field. The low number of identified studies suggests that the adoption of BNs in RE remains a research challenge. However, there are enough studies that support the applicability of the synergistic cooperation between RE and BNs.

External validity deals with the generalization of observed results and the key findings for practitioners and researchers. All the cases treated in these studies are bespoke solutions which are hard to generalize. Due to the different dimensions within RE, it is very difficult to make generalizations.

5 Conclusions

In this paper, we studied the applicability of BNs to Requirements Engineering. In particular, we carried out a systematic analysis that helps us use primary research papers to find out what we know and what we do not know about the enhancement of RE using BNs. Our work analyzed and discussed 20 studies that brought together 29 research papers.

There are few studies which link these two domains. Indeed, RE areas exist that have not even been covered yet, such as requirements negotiation and requirements classification. Other RE areas have only been addressed in one or two papers adopting the same point of view. RE areas are usually complex communication processes, because collaboration is needed between various stakeholders and requirement engineers and where the people involved have to make decisions. But RE decision making is not sufficiently mature, and no closed set of decision problems in RE are available that could be tackled with BNs.

In this work, we have identified improvements that should be addressed, such as those related to the BN-building process. Usually, experts in RE are not also experts in BNs because of this, the papers proposing the use of a BN to assist their activities should also include use cases illustrating the way in which RE experts should use the BN as well as explain how to extrapolate their results. Unfortunately, most of the studies do not include the way to use the BN that has been constructed, nor what the benefits are that requirement practitioners might obtain by their use.

Conversely, from the BN point of view, we can conclude that researchers do not follow a standard approach in constructing a BN. Despite the basic VI, QS, QE and VT steps being executed, each study applies its own approach: inter-views with experts, mapping RE artifacts, learning from databases, etc. The machine learning approach is the easiest method to obtain the BN structure and probability distributions, nonetheless, on a requirement level, there are no easily available repositories to be mined. Furthermore, our review has identified a number of common major challenges, such as how to deal with networks validation, or how to embed the models obtained in computer-aided software engineering tools (CARE tools). In fact, if we had tools that allowed the practical implementation of BNs into real-world software development projects, the validation of the models would be easier and faster.

In conclusion, to obtain successful results in future research works, we need to have closer collaboration between the roles of RE experts and BN experts. RE experts are in charge of formulating the problem and adapting it to the probabilistic method used, whereas BN experts have to extract the knowledge and select the best way to build the network. Nonetheless, the applicability of BN to RE has a limited scope due to RE tasks variability, lack of metrics and accuracy.

Acknowledgments This research has been funded by the Spanish Ministry of Economy and Competitiveness under project TIN2013-46638-C3-1-P.

References

1. Abran A, Moore J, Bourque P, Dupuis R, Tripp L (2004) Guide to the software engineering body of knowledge. IEEE Computer Society, Los Alamitos
2. Alenljung B, Persson A (2008) Portraying the practice of decision-making in requirements engineering: a case of large scale bespoke development. Requir Eng 13(4):257–279. doi:10.1007/s00766-008-0068-2
3. Alexander I, Beus-Dukic L (2009) Discovering requirements. How to specify products and services. Wiley, New York
4. Ali S, Briand LC, Hemmati H, Panesar-Walawege RK (2010) A systematic review of the application and empirical investigation of search-based test case generation. IEEE Trans Softw Eng 36(6):742–762. doi:10.1109/TSE.2009.52
5. Alves V, Niu N, Alves C, Valena G (2010) Requirements engineering for software product lines: a systematic literature review. Inf Softw Technol 52(8):806–820. doi:10.1016/j.infsof.2010.03.014

6. Aurum A, Wohlin C (2003) The fundamental nature of requirements engineering activities as a decision-making process. Inf Softw Technol 45(14):945–954. doi:10.1016/S0950-5849(03)00096-X
7. Bai CG (2005) Bayesian network based software reliability prediction with an operational profile. J Syst Softw 77(2):103–112. doi:10.1016/j.jss.2004.11.034
8. Barry PS, Laskey KB (1999) An application of uncertain reasoning to requirements engineering. In: Proceedings of the 15th conference on uncertainty in artificial intelligence, pp 41–48. Morgan Kaufman
9. Brito M, May J (2006) Gaining confidence in the software development process using expert systems. In: Garski J (ed) Computer safety, reliability, and security, lecture notes in computer science, vol 4166. Springer, Berlin, pp 113–126. doi:10.1007/11875567_9
10. Buntine W (1996) A guide to the literature on learning probabilistic networks from data. IEEE Trans Knowl Data Eng 8(2):195–210. doi:10.1109/69.494161
11. Cheng BHC, Atlee JM (2007) Research directions in requirements engineering. In: Briand A, Wolf LC (eds) FOSE, pp 285–303 . doi:10.1145/1253532.1254725
12. Cooper G, Herskovits E (1992) A Bayesian method for the induction of probabilistic networks from data. Mach Learn 9(4):309–347. doi:10.1007/BF00994110
13. de Melo AC, Sanchez AJ (2008) Software maintenance project delays prediction using Bayesian networks. Expert Syst Appl 34(2):908–919. doi:10.1016/j.eswa.2006.10.040
14. del Águila IM, del Sagrado J (2011) Requirement risk level forecast using Bayesian networks classifiers. Int J Softw Eng Knowl Eng 21(2):167–190. doi:10.1142/S0218194011005219
15. del Águila IM, del Sagrado J (2012) Metamodelling of Bayesian networks for decision-support system development. In: Proceedings of 8th workshop on knowledge engineering and software engineering (KESE8 2012)
16. del Sagrado J, del Aguila IM (2010) A Bayesian network for predicting the need for a requirements review. In: Meziane F, Vadera S (eds) Artificial Intelligence Applications for Improved Software Engineering Development: New Prospects. Information Science Reference, Hershey, pp 106–128
17. del Sagrado J, del Águila IM, Orellana FJ (2011) Architecture for the use of synergies between knowledge engineering and requirements engineering. In: Lozano JA, Gámez JA, Moreno JA (eds) CAEPIA, lecture notes in computer science, vol 7023. Springer, New York, pp 213–222. doi:10.1007/978-3-642-25274-7_22
18. Dieste O, Juristo N (2011) Systematic review and aggregation of empirical studies on elicitation techniques. IEEE Trans Softw Eng 37(2):283–304. doi:10.1109/TSE.2010.33
19. Donohue S, Dugan J (2003) Modeling the "good enough to release" decision using V & IV preference structures and Bayesian belief networks. In: Annual reliability and maintainability symposium, 2003, pp 568–573. doi:10.1109/RAMS.2003.1182051
20. Donohue S, Dugan J (2005) Is my software "good enough" to release? A probabilistic assessment. In: 29th annual IEEE/NASA Software engineering workshop, 2005, pp 5–13. doi:10.1109/SEW.2005.30
21. Druzdzel MJ, Flynn RR (2000) Decision support systems. In: Kent A (ed) Encyclopedia of library and information science, vol 67. Marcel Dekker Inc, New York, pp 120–133
22. Dybå T, Dingsøyr T (2008) Empirical studies of agile software development: a systematic review. Inf Softw Technol 50(9–10):833–859. doi:10.1016/j.infsof.2008.01.006
23. Dybå T, Kitchenham B, Jorgensen M (2005) Evidence-based software engineering for practitioners. IEEE Softw 22(1):58–65. doi:10.1109/MS.2005.6
24. El Emam K, Koru A (2008) A replicated survey of it software project failures. IEEE Softw 25(5):84–90. doi:10.1109/MS.2008.107
25. Fenton N, Krause P, Neil M (2002) Software measurement: uncertainty and causal modeling. IEEE Softw 19(4):116–122. doi:10.1109/MS.2002.1020298
26. Fenton N, Neil M, Marsh W, Hearty P, Marquez D, Krause P, Mishra R (2007) Predicting software defects in varying development lifecycles using Bayesian nets. Inf Softw Technol 49(1):32–43. doi:10.1016/j.infsof.2006.09.001
27. Fenton NE, Neil M (2014) Decision support software for probabilistic risk assessment using Bayesian networks. IEEE Softw 31(2):21–26. doi:10.1109/MS.2014.32
28. Galliers J, Sutcliffe A, Minocha S (1999) An impact analysis method for safety–critical user interface design. ACM Trans Comput Hum Interact 6(4):341–369. doi:10.1145/331490.331493
29. Ganesh J. Pai, J.B.D., Leteef K (2006) Analyses of milestone readiness levels in software IV&V. In: Proceedings of international symposium on software reliability engineering (ISSRE), workshop on software assessment
30. Glass RL (2002) Software engineering: facts and fallacies. Addison-Wesley Longman, Boston
31. Gregoriades A, Sutcliffe AG (2005) Scenario-based assessment of nonfunctional requirements. IEEE Trans Softw Eng 31(5):392–409. doi:10.1109/TSE.2005.59
32. Gregoriades A, Sutcliffe AG (2005) The system reliability analyser tool. Requir Eng 10(1):63–80. doi:10.1007/s00766-004-0200-x
33. Harrison R, da Cruz D, Henriques P, Pereira MJV, Liu SH, Menzies T, Mernik M, Rodriguez D (2012) Report from the first international workshop on realizing artificial intelligence synergies in software engineering (raise 2012). SIGSOFT Softw Eng Notes 37(5):34–35. doi:10.1145/2347696.2347697
34. Howison J, Conklin M, Crowston K (2006) Flossmole: a collaborative repository for floss research data and analyses. Int J Inf Technol Web Eng 1:17–26. doi:10.4018/jitwe.2006070102
35. Hui B, Boutilier C (2006) Who's asking for help? A Bayesian approach to intelligent assistance. In: Proceedings of the 11th international conference on intelligent user interfaces, IUI '06. ACM, New York, NY, USA, pp 186–193. doi:10.1145/1111449.1111491
36. IEEE I (1998) IEEE recommended practice for software requirements specifications. Tech. rep. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=720574
37. Jensen FV, Nielsen TD (2007) Bayesian networks and decision graphs, 2nd edn. Springer, New York
38. Katina PF, Keating CB, Jaradat RM (2014) System requirements engineering in complex situations. Requir Eng 19(1):45–62. doi:10.1007/s00766-012-0157-0
39. Kitchenham B, Brereton P (2013) A systematic review of systematic review process research in software engineering. Inf Softw Technol 55(12):2049–2075. doi:10.1016/j.infsof.2013.07.010
40. Kitchenham B, Brereton P, Budgen D, Turner M, Bailey J, Linkman SG (2009) Systematic literature reviews in software engineering—a systematic literature review. Inf Softw Technol 51(1):7–15. doi:10.1016/j.infsof.2008.09.009
41. Kitchenham B, Charters S (2007) Guidelines for performing systematic literature reviews in software engineering. Tech. Rep. EBSE 2007–001, Keele University and Durham University Joint Report
42. Kjrulff UB, Madsen AL (2008) Bayesian networks and influence diagrams. Springer, New York
43. Korb K, Nicholson A (2010) Bayesian artificial intelligence, 2nd edn. Chapman and Hall, London

44. Laskey KB, Barry P, Brouse P (2000) Breaking the knowledge bottleneck for Bayesian networks using language (UML) artifacts. In: 2nd international conference on software engineering and knowledge engineering, Chicago, IL
45. Laskey KB, Barry PS (2000) Development of Bayesian networks from unified modeling language artifacts. In: Proceedings of the twelfth software engineering/knowledge engineering 2000 conference
46. Lauría EJM, Duchessi PJ (2006) A Bayesian belief network for it implementation decision support. Decis Support Syst 42(3): 1573–1588. doi:10.1016/j.dss.2006.01.003
47. Lee S (2011) Probabilistic risk assessment for security requirements: a preliminary study. In: 2011 fifth international conference on secure software integration and reliability improvement (SSIRI), pp 11–20. doi:10.1109/SSIRI.2011.12
48. Lucas FJ, Molina F, Toval A (2009) A systematic review of UML model consistency management. Inf Softw Technol 51(12): 1631–1645. doi:10.1016/j.infsof.2009.04.009
49. Menzies T (2006) Improving IV&V techniques through the analysis of project anomalies: Bayes networks—preliminary report. Tech. rep., West Virginia University. http://menzies.us/pdf/06anomalies-bayes0.pdf
50. Nagy A, Njima M, Mkrtchyan L (2010) A Bayesian based method for agile software development release planning and project health monitoring. In: 2010 2nd international conference on intelligent networking and collaborative systems (INCOS), pp 192–199. doi:10.1109/INCOS.2010.99
51. Nalepa GJ, Nadas JC, Baumeister J (eds) Proceedings of 8th workshop on knowledge engineering and software engineering (KESE8) at the 20th Biennial European conference on artificial intelligence (ECAI 2012) Montpellier, France, August 28, 2011, CEUR workshop proceedings, vol 949. CEUR-WS.org (2011)
52. Neapolitan RE (2003) Learning Bayesian networks, illustrated, edition edn. Prentice Hall, Upper Saddle River
53. Nuseibeh B, Easterbrook S (2000) Requirements engineering: a roadmap. In: Proceedings of the conference on the future of software engineering, ICSE '00. ACM, New York, NY, USA, pp 35–46. doi:10.1145/336512.336523
54. Olson J, Rozenblit J (1998) Framework for hardware/software partitioning utilizing bayesian belief networks. In: 1998 IEEE international conference on systems, man, and cybernetics, 1998, vol 4, pp 3983–3988. doi:10.1109/ICSMC.1998.726711
55. Omoronyia I, Sindre G, Stålhane T (2011) Exploring a Bayesian and linear approach to requirements traceability. Inf Softw Technol 53(8):851–871. doi:10.1016/j.infsof.2011.03.001
56. Pai G, Bechta-Dugan J, Lateef K (2005) Bayesian networks applied to software IV & V. In: Proceedings of the 29th annual NASA/IEEE software engineering workshop, pp 293–304. doi:10.1109/SEW.2005.20
57. Pai M, McCulloch M, Enanoria W, Colford J (2004) Systematic reviews of diagnostic test evaluations: what's behind the scenes? Evid Based Med 9(4):101–103. doi:10.1136/ebm.9.4.101
58. Pearl J (1988) Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann, San Francisco
59. Pendharkar P, Subramanian G, Rodger J (2005) A probabilistic model for predicting software development effort. IEEE Trans Softw Eng 31(7):615–624. doi:10.1109/TSE.2005.75
60. Punter T, Trendowicz A, Kaiser P (2002) Evaluating evolutionary software systems. In: 4th international conference on product focused software process improvement PROFES 2002. doi:10.1007/3-540-36209-6_23
61. Radlinski L (2011) A framework for integrated software quality prediction using Bayesian nets. In: Murgante B, Gervasi O, Iglesias A, Taniar D, Apduhan B (eds) Computational science and its applications—ICCSA 2011, lecture notes in computer science, vol 6786. Springer, Berlin, pp 310–325. doi:10.1007/978-3-642-21934-4_26
62. Ruhe G, Saliu MO (2005) The art and science of software release planning. IEEE Softw 22(6):47–53. doi:10.1109/MS.2005.164
63. Si G, Xu J, Yang J, Wen S (2014) An evaluation model for dependability of internet-scale software on basis of Bayesian networks and trustworthiness. J Syst Softw 89:63–75. doi:10.1016/j.jss.2013.08.035
64. Sutcliffe A, Galliers J, Minocha S (1999) Human errors and system requirements. In: Proceedings of the IEEE international symposium on requirements engineering, 1999, pp 23–30. doi:10.1109/ISRE.1999.777982
65. Sutcliffe A, Gregoriades A (2002) Validating functional system requirements with scenarios. In: Proceedings of the IEEE joint international conference on requirements engineering, 2002, pp 181–188. doi:10.1109/ICRE.2002.1048521
66. Tang Y, Feng K (2009) An expert system based approach to modeling and selecting requirement engineering techniques. In: Liu W, Luo X, Wang F, Lei J (eds) Web information systems and mining, lecture notes in computer science, vol 5854. Springer, Berlin, pp 19–30. doi:10.1007/978-3-642-05250-7_3
67. Tang Y, Feng K, Cooper K, Cangussu J (2009) Requirement engineering techniques selection and modeling an expert system based approach. In: International conference on machine learning and applications, 2009, ICMLA '09, pp 705–709. doi:10.1109/ICMLA.2009.102
68. Wang J, Wang Q (2014) Analyzing and predicting software integration bugs using network analysis on requirements dependency network. Requir Eng 1–24. doi:10.1007/s00766-014-0215-x
69. Wen J, Li S, Lin Z, Hu Y, Huang C (2012) Systematic literature review of machine learning based software development effort estimation models. Inf Softw Technol 54(1):41–59. doi:10.1016/j.infsof.2011.09.002
70. Wiegers K (2003) Software requirements: practical techniques for gathering and managing requirements throughout the product development cycle. Pro-best practices. Microsoft Press, New York
71. Wohlin C, Runeson P, Host M, Ohlsson M, Regnell B, Wesslen A (2000) Experimentation in software engineering: an introduction. Kluwer Academic, London
72. Wooff D, Goldstein M, Coolen F (2002) Bayesian graphical models for software testing. IEEE Trans Softw Eng 28(5): 510–525. doi:10.1109/TSE.2002.1000453
73. Wu H, Liu L, Ma W (2010) Optimizing requirements elicitation with an i* and Bayesian network integrated modelling approach. In: 2010 IEEE 34th annual computer software and applications conference workshops (COMPSACW), pp 182–188. doi:10.1109/COMPSACW.2010.40
74. Zhang J, Zhang X, Lin KJ (2011) An efficient Bayesian diagnosis for QOS management in service-oriented architecture. In: 2011 IEEE international conference on service-oriented computing and applications (SOCA), pp 1–8. doi:10.1109/SOCA.2011.6166214
75. Zhang Z, Liu Y (2012) Application of active learning strategy and formalization method in requirement analysis. In: 2012 IEEE symposium on robotics and applications (ISRA), pp 958–960. doi:10.1109/ISRA.2012.6219353