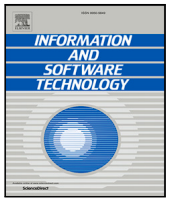
Information and Software Technology 177 (2025) 107586

Contents lists available at ScienceDirect

Information and Software Technology

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



Constructing the graphical structure of expert-based Bayesian networks in the context of software engineering: A systematic mapping study

Thiago Rique [a,b,*], Mirko Perkusich [b], Kyller Gorgônio [b], Hyggo Almeida [b], Angelo Perkusich [b]

[a] Federal Institute of Education, Science and Technology of Paraíba (IFPB), Princesa Isabel, PB, Brazil
[b] Intelligent Software Engineering Group (ISE/VIRTUS), Federal University of Campina Grande (UFCG), Campina Grande, PB, Brazil

ARTICLE INFO

Keywords:
Bayesian networks
Bayesian network structure
Expert knowledge
Software engineering
Systematic mapping

ABSTRACT

Context: In scenarios where data availability issues hinder the applications of statistical causal modeling in software engineering (SE), Bayesian networks (BNs) have been widely used due to their flexibility in incorporating expert knowledge. However, the general understanding of how the graphical structure, i.e., the directed acyclic graph (DAG), of these models is built from domain experts is still insufficient.
Objective: This study aims to characterize the SE landscape of constructing the graphical structure of BNs, including their potential for causal modeling.
Method: We conducted a systematic mapping study employing a hybrid search strategy that combines a database search with parallel backward and forward snowballing.
Results: Our mapping included a total of 106 studies. Different methods are commonly combined to construct expert-based BN structures. These methods span across data gathering & analysis (e.g., interviews, focus groups, literature research, grounded theory, and statistical analysis) and reasoning mechanisms (e.g., using idioms combined with the adoption of lifecycle models, risk-centric modeling, and other frameworks to guide BN construction). We found a lack of consensus regarding validation procedures, particularly critical when modeling cause–effect relationships from knowledge. Additionally, expert-based BNs are mainly applied at the tactical level to address problems related to software engineering management and software quality. Challenges in creating expert-based structures include validation procedures, experts' availability, expertise level, and structure complexity handling. Key recommendations involve empirical validation, participatory involvement, and balance between adaptation to organizational constraints and model construction requirements.
Conclusion: The construction of expert-based BN structures in SE varies in rigor, with some methods being systematic while others appear ad hoc. To enhance BN application, reducing expert knowledge subjectivity, enhancing methodological rigor, and clearly articulating the construction rationale is essential. Addressing these challenges is crucial for improving the reliability of causal inferences drawn from these models, ultimately leading to better-informed decisions in SE practices.

1. Introduction

Causal inference is a methodological approach used to identify and estimate cause–effect relationships between variables, moving beyond simple correlation to reveal the actual mechanisms by which changes in one variable lead to changes in another. Extensively used across various fields, including social sciences, epidemiology, economics, and machine learning [1], causal inference is grounded in Judea Pearl's seminal framework [2]. The gold standard for constructing causal models involves three major steps: modeling, identification, and estimation [3]. In the modeling step, causal assumptions are explicitly made, typically obtained through domain experts or extracted from data. The next steps refer to using statistical analysis (i.e., do-calculus [4]) to reduce the impact of confounding factors, remove spurious correlations, and estimate actual causal effects (i.e., statistical causal inference).

Software engineering (SE), characterized by complex decision-making and problem-solving, necessitates a robust approach to causal inference to enhance software quality and stakeholder satisfaction [5–7]. However, statistical causal inference encounters limitations at strategic and tactical levels, where high-quality, relevant data may be incomplete, scarce, or non-existent [8,9].

In scenarios with data constraints, Bayesian networks (BNs) offer a compelling alternative by enabling the integration of expert

* Corresponding author at: Federal Institute of Education, Science and Technology of Paraíba (IFPB), Princesa Isabel, PB, Brazil.
E-mail addresses: thiago.rique@ifpb.edu.br (T. Rique), mirko@virtus.ufcg.edu.br (M. Perkusich), kyller@virtus.ufcg.edu.br (K. Gorgônio), hyggo@virtus.ufcg.edu.br (H. Almeida), perkusic@virtus.ufcg.edu.br (A. Perkusich).

https://doi.org/10.1016/j.infsof.2024.107586
Received 30 November 2023; Received in revised form 9 September 2024; Accepted 18 September 2024
Available online 27 September 2024

knowledge for modeling causality. Although BNs are not inherently causal, they can represent causality effectively if the model accurately encapsulates all relevant variables and causal dependencies (i.e., the causal assumptions can be treated as causal effects). This capability is demonstrated in various domains where BNs have been successfully applied to model cause–effect relationships in the absence of comprehensive data (e.g., [10–16]). Furthermore, Lattimore and Rohde [17] illustrate that estimating causal effects can be achieved within the standard Bayesian paradigm by explicitly modeling assumptions about interventions, thereby aligning Bayesian methods with the do-calculus for causal estimation [17–22].

In the context of SE, effectively engineering expert knowledge to construct the BN graphical structure, i.e., the directed acyclic graph (DAG), is crucial [6,23]. The Expert-based Knowledge Engineering of Bayesian Networks (EKEBN) process provides a structured framework and has been utilized in various SE studies [24–28]. However, EKEBN often lacks detailed guidance on critical steps such as the elicitation of knowledge and the systematic analysis of information, which are essential for building robust BN structures. This oversight highlights a significant gap in the SE literature on BNs: while existing methodologies reference well-known knowledge acquisition methods like interviews and surveys, they fall short in detailing how these methods are adapted and validated within the SE context. This deficiency directly impacts the reliability and validity of the causal inferences these BNs are intended to support, underscoring the need for more comprehensive and transparent construction processes.

Based on this scenario, our study focuses on characterizing the SE landscape of constructing BN structures solely through expert knowledge engineering (i.e., excluding works where expert knowledge has been combined with observational data to learn the structure). We delve into how domain experts contribute to the development of these structures and seek to provide a clear picture of current practices, gaps, and opportunities in this area. By doing so, we aim to bridge the divide between theory and practice in applying BNs in SE, offering valuable insights for practitioners and researchers. Given this, we have defined the following research questions (RQs):

- RQ1: What methods have been employed to construct the structure of BNs using expert knowledge in SE?
- RQ2: How have researchers validated the effectiveness of the structure of BNs constructed using expert knowledge in SE?
- RQ3: At what level (strategic, tactical, or operational) are BNs constructed with expert knowledge being applied in SE?
- RQ4: In what areas of SE have BNs constructed with expert knowledge been most commonly applied?
- RQ5: What tools have been used to construct BN models based on expert knowledge in SE?
- RQ6: What challenges and limitations have been identified in constructing and applying BN structures using expert knowledge in SE?
- RQ7: What are the trends in the application of BNs constructed with expert knowledge in SE, and what future directions are suggested by the existing literature?

The contribution of our study is threefold. By categorizing the methods and practices used, we provide a structured framework for understanding how expert knowledge has been engineered into BN structures, thereby mapping the SE's current state of constructing expert-based BNs. We also highlight critical challenges and offer recommendations for expert knowledge use in building these models. This understanding is essential for our community and paves the way for new advancements in the field of BNs in SE. Furthermore, we analyze the level of rigor (or lack thereof) previous studies have employed in justifying their causal assumptions and discuss future directions for causal modeling with BNs in scenarios where data availability issues pose a challenge to traditional techniques.

Our results further support the findings of Tosun et al. [29], who noted a lack of systematic guidance for constructing BN models. While their work focused on software quality prediction, our study extends to the broader SE field, where we have identified significant gaps in model development by domain experts. Hence, establishing guidelines for developing expert-based BNs is still an open issue in SE.

The remainder of this paper is organized as follows: Section 2 outlines the background information and related work. Section 3 describes our systematic mapping protocol. Section 4 presents the results and analysis of our study. Section 5 discusses the implications of our findings for both research and practice. Section 6 addresses the potential threats to the validity of our study. Finally, Section 7 offers our concluding remarks.

2. Background and related work

This section outlines the preliminaries on BNs with a simple example, summarizes foundational concepts of causal modeling, provides an overview of BNs in the SE domain, and presents related work concerning other secondary studies on closely related topics.

2.1. Preliminaries on BNs

BNs are probabilistic graphical models that provide a structured framework for modeling problems that involve reasoning under uncertainty [30]. They are comprised of two essential components: a DAG that represents the qualitative structure of the network and conditional probability distributions associated with each node, representing the quantitative aspect.

Graphical structure: The graphical structure of a BN is a DAG where nodes represent random variables, and directed edges (arcs) between nodes represent probabilistic dependencies or causal relationships. This graph encodes the conditional independence assumptions in the domain [6].

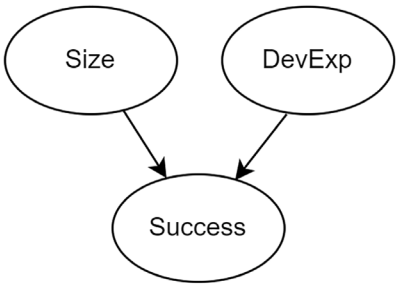
Conditional probability: BNs use conditional probability to model the probability distribution of each node in the network given the values of its parent nodes. This is expressed as $P(X|Parents(X))$, where X is a node, and $Parents(X)$ are its parent nodes. In the case of continuous nodes, this conditional probability can be defined as a distribution [31]. Also, discretization can be applied to convert continuous values into discrete intervals [6]. For discrete nodes, probability distributions are depicted through conditional probability tables (CPTs). These tables detail the probabilities associated with each discrete state a node can take on, considering all possible combinations of states for the parent nodes (if any). This approach enables the quantification of causal relationships and the uncertainty between nodes within the BN.

Example in the SE domain: Let us consider a BN applied to the SE problem of estimating the probability of a software project's success. Based on domain knowledge, this network can help us make predictions and perform various types of analyses. Next, we outline its components and functionalities (see Fig. 1):

- Node 1: Project Size (Size)
  - Represents the size of the software project (e.g., small, medium, large).
  - CPT encodes the probability distribution of project size.
- Node 2: Developer Experience (DevExp)
  - Represents the experience level of developers (e.g., junior, intermediate, senior).
  - CPT encodes the probability distribution of developer experience.
- Node 3: Project Success (Success)
  - Represents the binary outcome of the project (success or failure).

Size		
Small	Medium	Large
0.3	0.5	0.2

DevExp		
Junior	Intermediate	Senior
0.2	0.5	0.3



Success			
Size	DevExp	Success	Failure
Small	Junior	0.6	0.4
Small	Intermediate	0.75	0.25
Small	Senior	0.9	0.1
Medium	Junior	0.5	0.5
Medium	Intermediate	0.7	0.3
Medium	Senior	0.85	0.15
Large	Junior	0.3	0.7
Large	Intermediate	0.5	0.5
Large	Senior	0.75	0.25

Fig. 1. Simple BN example.

– CPT encodes the probability of project success given project size and developer experience.

The key part of the BN is the conditional probability distribution in Node 3 (Project Success). To specify this distribution, we would need expert knowledge or data to estimate the likelihood of project success based on project size and developer experience. Suppose our CPT for Node 3 (Project Success) is as defined in Fig. 1 (the values are fictitious; examples of more realistic scenarios can be found in [8,32,33]). Those conditional probabilities represent the likelihood of project success or failure given different combinations of project size and developer experience.

In this BN, we can perform different analyses, such as:

- Predictive analysis: Given the project size and developer experience, we can predict the probability of project success. For example, we can calculate $P(Success|Size = Small, DevExp = Senior)$ to estimate the likelihood of success (which is 0.9 in this case).
- What-if analysis: We can investigate the impact of changes on some nodes on the likelihood of project success. For instance, we can analyze how increasing the developer experience from junior to senior affects project success probabilities.
- Sensitivity analysis: By altering the probabilities in the CPTs, we can assess the sensitivity of project success to different factors, helping identify critical variables.

This BN provides a systematic way to model and reason about software project success, incorporating domain knowledge and probabilistic reasoning to inform decision-making and risk management in SE.

BNs can be constructed using expert knowledge, data, or a combination of both (hybrid models). Given the focus of this study, it is worth highlighting an existing approach for building expert-based models. The Expert-based Knowledge Engineering of Bayesian Networks (EKEBN) is a process that involves three main steps: structure building, uncertainty quantification, and model validation [34]. The first step iteratively constructs the BN structure through knowledge elicitation and representation meetings. In the second step, conditional probabilities are determined to quantify the relationships between factors in the BN. Finally, the third step validates the model resulting from the previous steps, determining if revisions are necessary. It should be noted, however, that while EKEBN outlines the steps involved, it does not provide detailed guidance on the methodology for analyzing data to convert expert knowledge into specific variables and relationships. This reinforces the need for additional guidelines or frameworks to assist in the practical application of expert knowledge in constructing BNs.

When building the DAG of large-scale BNs, the idioms described by Neil et al. [19] represent a key concept. An idiom is a "building block" or a pattern in the structure of a BN that encapsulates a specific type of reasoning or relationship among variables. Idioms are fundamental components that can be combined and reused to construct more complex and large-scale BNs efficiently. Neil et al. [19] identify five primary idioms in their work (see Fig. 2):

- Definitional/synthesis idiom: This idiom involves combining several variables into a single node to simplify the network and manage its complexity. It is often used when variables collectively define or influence another concept.
- Cause-consequence idiom: It models the relationship between causes and their direct effects, having observable consequences. This idiom is particularly useful in scenarios where a causal

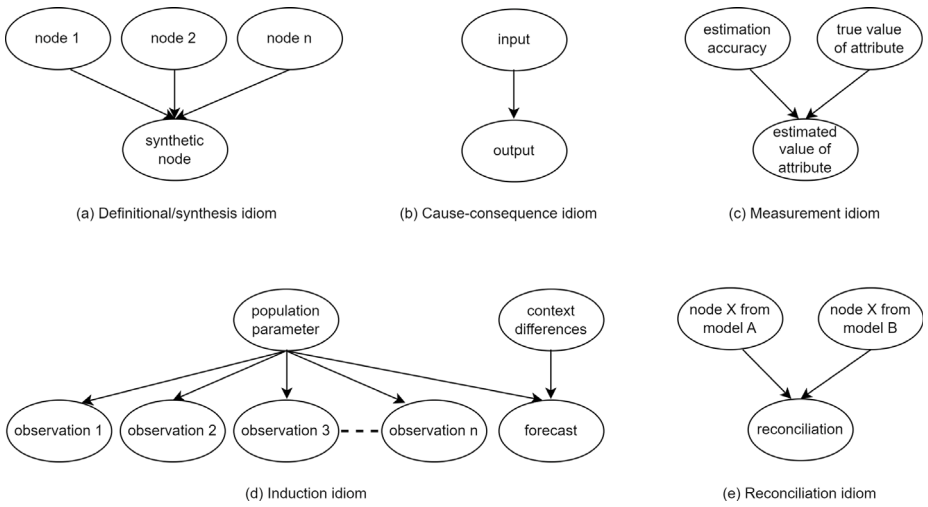


Fig. 2. Idioms identified by Neil et al. [19].

process needs to be represented, such as the impact of certain actions or conditions on an outcome.

- Measurement idiom: This idiom deals with the uncertainty in the measurement process. It is used when the goal is to represent the accuracy or reliability of measurements or observations. There are two alternatives to instantiate this idiom. The classic approach, described by Neil et al. [19], contains three nodes: estimation accuracy (A), the true value of the attribute (B), and the estimated value of the attribute (C), where A and B point to C. A simplified version is described by Fenton and Neil [35], where the node for estimation accuracy is not part of the model. Instead, the model may have multiple nodes for the estimated value of the attribute, called indicator nodes, which are pointed to by the node for the true value of the attribute. In practice, this version of the measurement idiom allows the knowledge engineer to identify multiple measures (or indicators) for a given attribute and, individually, model their construct validity (i.e., the extent to which the measure accurately assesses the attribute) by defining the conditional probability distributions.
- Induction idiom: This idiom is applied in situations where reasoning is based on patterns observed in a population to make inferences about individual instances. It is typically used for probabilistic reasoning based on historical or aggregated data.
- Reconciliation idiom: It is used for reconciling different models or sources of evidence about a single attribute. This idiom is essential when there are multiple approaches or perspectives that need to be integrated into a single coherent analysis.

These idioms, as identified by Neil et al., provide a structured approach to constructing BNs, allowing for the modeling of complex relationships and reasoning processes in a more manageable and modular manner.

2.2. Causal modeling and inference in SE

Causal modeling plays an essential role in understanding and predicting outcomes in SE, where complex relationships between factors such as software quality, development processes, and project outcomes need to be addressed. However, there is ongoing debate in the scientific community regarding the adequacy of traditional probability theory in representing causal problems. This has led to the development of advanced methods like Pearl's do-calculus, which addresses causal inference from observational data under specific assumptions related to causal structures and prior knowledge [4].

While Pearl's causal framework has been highly influential, particularly in fields with abundant data, its application in SE is still limited. This is primarily due to the fragmented and incomplete nature of data typically available in SE, which poses challenges for adopting data-heavy causal modeling techniques such as Structural Equation Modeling (SEM) [36] and Propensity Score Matching (PSM) [37]. These traditional methods require substantial amounts of high-quality data to establish reliable causal relationships, making them difficult to apply in SE scenarios where data is often scarce, particularly at the strategic and tactical levels [8,9].

Given these challenges, BNs offer a viable and flexible solution for causal modeling in data-scarce environments. A critical challenge is how to transform what Judea Pearl refers to as "causal assumptions" (i.e., hypotheses) during the modeling phase into robust cause–effect relationships. This transformation can be effectively managed by employing rigorous qualitative data collection and analysis methods [38,39]. This approach enhances the reliability of BNs by minimizing the impact of confounding factors and reducing the likelihood of spurious correlations. As a result, the directional arrows in BNs, which typically represent associations (i.e., correlations), can be interpreted with greater confidence as causal links when supported by well-substantiated expert knowledge [17–22].

The flexibility of BNs also allows for an iterative refinement process. Initial models based on expert knowledge can be further validated and improved as observational data is collected [16,40]. This process aligns with Pearl's framework, where causal assumptions can be statistically tested and refined over time, enhancing both the accuracy and applicability of the models. Hybrid methods that combine data-driven approaches with expert knowledge can further increase the robustness of BNs, making them indispensable for dynamic SE environments [41–43].

Moreover, BNs not only address data limitations but also provide a method that evolves with the availability of new data. This dual-phase approach — from expert-based construction to data-driven refinement — ensures that BNs remain relevant and adaptable, addressing the evolving requirements of SE processes. Thus, BNs offer a dynamic and practical framework for ongoing causal analysis, particularly suited for complex and evolving SE scenarios [44].

2.3. BNs in SE

SE is a complex and dynamic field that involves numerous uncertainties and dependencies among various factors, making it challenging to make informed decisions and predictions. As such, BNs find several applications in this domain, offering valuable insights into various SE topics. The most popular ones are software quality and software engineering management [6]. Concerning the former, BNs have been

applied to predict software faults, failures, and reliability. The latter includes subtopics like project planning and estimation. This encompasses resource allocation, cost, effort, schedule, and productivity prediction, as well as the performance of trade-off analyses using BNs [6].

Next, we present some software development aspects addressed by the application of BNs with examples from the literature:

- Risk assessment: By modeling the dependencies between project parameters and potential risks, these networks can provide a comprehensive understanding of project uncertainties. This helps project managers make more informed decisions about resource allocation, scheduling, and risk mitigation strategies. Examples: [45,46].
- Software quality assurance: BNs can be used to model dependencies between various software quality attributes, such as reliability, maintainability, and security. By analyzing these relationships, software engineers can identify potential areas for improvement and prioritize efforts to enhance overall software quality. Examples: [47,48].
- Fault detection and diagnosis: BNs can be employed to model the relationships between system events and potential faults. By monitoring system behaviors, software engineers can detect anomalies, diagnose faults, and expedite the troubleshooting process. Examples: [49,50].
- Requirements engineering: BNs find applications in requirements engineering by modeling dependencies between different requirements and their impact on the project's success. This aids in requirements prioritization and validation, ensuring that critical requirements are addressed early in the development process. Examples: [51,52].

BNs offer a promising avenue for improving the overall quality and success of software projects. As the SE field evolves, BNs will likely play an increasingly significant role in addressing complex challenges and uncertainties.

2.4. Related work

To the best of our knowledge, only one systematic mapping study (SMS) was performed focusing on BNs in SE from a broad perspective. However, we identified other secondary studies concerning the application of BNs to specific SE problems.

Radlinski [53] conducted a survey of 23 publications on BNs built for software development effort prediction. He briefly described and analyzed the models in terms of their purpose, type of structure, data/knowledge base for model construction, and performance measures for validation.

Misirli and Bener [54] presented an SMS on BNs for software quality prediction. Their mapping included 38 primary studies that they classified according to the techniques used for estimating the model parameters, the techniques used for building the BN structure, and the type of variables representing BN nodes.

Rodríguez et al. [55] conducted a review of BNs in software testing. They analyzed the selected literature (41 papers), grouping the different subareas and providing discussions regarding possible research paths.

Focusing on BNs in requirements engineering (RE), del Águila and del Sagrado [56] performed a literature review comprising 20 studies. They investigated the scope of the application of BNs to RE and the motivations behind their adoption, the undertaken steps to construct the qualitative and quantitative parts of BN models, and the advantages and disadvantages of using BNs in this specific SE topic.

Extending their prior mapping, Tosun et al. [29] performed a systematic literature review (SLR) on the applications of BNs to predict software quality. From a total of 10 primary studies, they investigated the use of BNs regarding dataset characteristics, techniques used for BN building (structure and parameter learning), use of tools, and validation techniques.

In their 20-year SMS, de Sousa et al. [7] reviewed the applications of BNs to software project management (SPM). They analyzed 109 publications investigating the motivations for using BNs, the problem domain and model scope that researchers have addressed, the stage of the software development life cycle in which BNs have been used, the venues of the publications, and the tools used for BN modeling.

Misirli and Bener [6] presented the only secondary study that covered the broad area of SE. The authors conducted a systematic mapping in which they analyzed 117 primary studies investigating the following facets of the application of BNs in SE: main challenges addressed, techniques for structure and parameter learning, and types of variables used as BN nodes.

It is noteworthy that the last of these seven studies is the one more closely related to ours both in terms of scope and research questions. It simultaneously covers the field of SE as a whole and investigates techniques used for BN structure construction. However, our study has a stricter scope, as we aim to investigate the construction of BN structures in SE using a specific technique, i.e., expert knowledge. Misirli and Bener [6] did not delve into this level of detail.

Other secondary studies have focused on the step of building the graphical structures of BNs (e.g., [57,58]), but they do not target any specific domain. Moreover, their focus is on algorithms for structure learning from data. Thus, apart from ours, we are unaware of any other secondary study addressing the construction of BN structures through expert knowledge. We hope our initiative serves as an inspiration to researchers in other domains in which expert knowledge plays a paramount role in BN structure construction, such as in SE.

Table 1 provides an overview of related work. We describe each study's scope and focus and indicate how the study maps to the seven RQs addressed by our systematic mapping. Different RQs are investigated in related studies, not or only partially related to our RQs (e.g., Radlinski [53] and Tosun et al. [29] investigate BN validation procedures in software development effort prediction and software quality prediction, respectively, whereas we are interested in SE, not focusing on a specific aspect; additionally, other secondary studies do not restrict their investigation to BNs constructed with expert knowledge as ours do). In this way, such an overview makes it more evident to what extent our study contributes to the body of knowledge on BNs in SE.

3. Research method

This section describes our research design. We performed an SMS following the guidelines by Peterson et al. [59]. Given that our research topic has been overlooked in the SE literature, this type of secondary study represents a suitable strategy since its primary purpose is to comprehensively understand a particular field, mapping and categorizing relevant studies based on predefined criteria. Mapping studies follow a principled and structured protocol similar to that of SLRs. However, the objectives and research questions of the former tend to be more general, and the criteria for their search strategies are less strict when compared to those of the latter [60].

3.1. Search strategy

As researchers need to perform secondary studies efficiently without compromising their quality, the SE community has invested significant effort in this matter and achieved promising results. Several researchers have provided guidelines and recommendations to support their peers in aggregating and synthesizing evidence through review studies (e.g., [59–61]). Recently, Mourão et al. [62] examined different search strategies to conduct SLRs in SE and found that using a hybrid strategy combining a database search on the Scopus digital library with parallel backward and forward snowballing (using Google Scholar) tends to strike an appropriate balance between the quality of results

Table 1
Overview of related work.

Reference	Scope	Study focus (as per RQs)	RQ1	RQ2	RQ3	RQ4	RQ5	RQ6	RQ7
Radlinski [53]	Software development effort prediction	Purpose, type of structure, data/knowledge base, and validation		✓					
Misirli and Bener [54]	Software quality prediction	Techniques for parameter learning, techniques for structure learning, and variable types							
Rodríguez et al. [55]	Software testing	Different aspects of software testing and quality that apply Bayesian concepts							
Del Águila and del Sagrado [56]	Requirements engineering	RE knowledge area, reasons for using BNs in RE, methods for adopting BNs in RE, advantages and disadvantages of using BNs in RE							
Tosun et al. [29]	Software quality prediction	Dataset characteristics, techniques for parameter learning, techniques for structure learning, BN tools, and validation techniques		✓			✓		
De Sousa et al. [7]	Software project management	Reasons for using BNs in SPM, project management domains, model scope, software life cycle phase, BN tools, where and when the study was published					✓		
Misirli and Bener [6]	Software engineering	SE challenges addressed, techniques for structure learning, techniques for parameter learning, and variable types				✓			
Our study	Software engineering	Methods for BN structure construction using expert knowledge, validation procedures, level of focus, SE problems addressed, BN tools, limitations and challenges of constructing expert-based BN structures, recommendations for expert knowledge use	✓	✓	✓	✓	✓	✓	✓

and the effort required for the review. In addition, the study of Wohlin et al. [63] strengthens the findings of Mourão et al. [62], indicating that a search on a representative digital library (e.g., Scopus) combined with snowballing for the identification of primary studies in systematic literature studies is a successful approach.

Based on these findings, we adopted such a hybrid strategy, searching for papers in the Scopus database, applying our exclusion criteria to filter the results, and performing parallel backward and forward snowballing on the remaining papers. As described in [62], in this parallel process, papers identified through backward snowballing were not subject to forward snowballing, and vice-versa. We took the study of Alonso et al. [64] as an inspiration as they adopted the same search strategy.

To search for primary studies, we adopted the search string originally used by Misirli and Bener [6] in their mapping study on BNs for evidence-based decision-making in SE. The decision to reuse this search string was based on the rationale that it had already been employed in a previous study in which its effectiveness had been empirically validated. This approach not only ensures the consistency of the search process but also builds upon a well-established foundation for identifying relevant literature in this domain. The search string is as follows:

("Bayesian network" OR "Bayesian net" OR "Bayes net" OR "Bayesian belief network") AND "software engineering"

Additionally, before performing the database search, a good practice is to check its validity against already known primary studies [61]. Given this, we tested Misirli and Bener's search string against ten known relevant papers, and it retrieved all of them. The list of papers used for this purpose is available in the review protocol included in our replication package.

3.2. Study selection

Next, we describe the study selection process (see Fig. 3). The search in Scopus returned a total of 446 document results. We then applied filters available in the search engine to exclude the documents belonging to subject areas not related to SE (e.g., medicine and environmental science), not written in English, and of a document type not considered in our review protocol. The advanced query in Scopus is provided in our supplementary material. After that filter application, a total of 287 documents remained, which were considered in the selection process.

The selection of papers was comprised of the following steps. First, our intention was to be less restrictive, reaching a broad spectrum of papers to serve as input for the next step. Thus, based on [6], we applied the following inclusion and exclusion criteria (IC/EC) in the first phase of the selection process:

- IC1: It addresses the use of BNs to tackle an SE problem.
- EC1: It does not address BNs in SE (e.g., papers employing Bayesian statistics without building a network, comparing BNs with other models, or using BNs applied to other domains).
- EC2: It is not a primary study.
- EC3: It is not written in English.
- EC4: It is not published in a peer-reviewed journal, conference, or workshop.

We applied the IC/EC at the title and abstract level. If it was not possible to make a decision about the inclusion of the paper, we applied the IC/EC at the content level. The 107 papers selected in the first phase formed the seed set that was subject to parallel backward and forward snowballing. In the first backward snowballing iteration, we analyzed 2,926 papers and included 71; in the second iteration, we analyzed 1,740 papers but included 10; in the third iteration, we analyzed 313 papers but included none. In the first forward snowballing iteration, we analyzed 4,387 papers and included 92; in the second iteration, we analyzed 1,219 papers but included 16; in the third iteration, we analyzed 231 papers but included 5; in the fourth iteration, we analyzed 55 papers but included none. The search and snowballing procedure together resulted in 301 papers (= 107 + 194). We then moved to the second phase of the selection process, in which we examined the 301 papers against the following IC/EC:

- IC1: It addresses the use of BNs whose structures were constructed based on expert knowledge to tackle an SE problem. Our definition of "expert knowledge" is based on the work of Misirli and Bener [6], which, in summary, includes manually extracting knowledge from experts (researchers, software practitioners, or both) or from the literature. Also, we considered expert judgment about the analysis of datasets, metrics, or software artifacts (e.g., performing statistical analysis of a dataset to identify relevant variables/relationships and using expert judgment to decide which ones will be included in the model or if it is necessary to include variables not present in the dataset). While data analysis

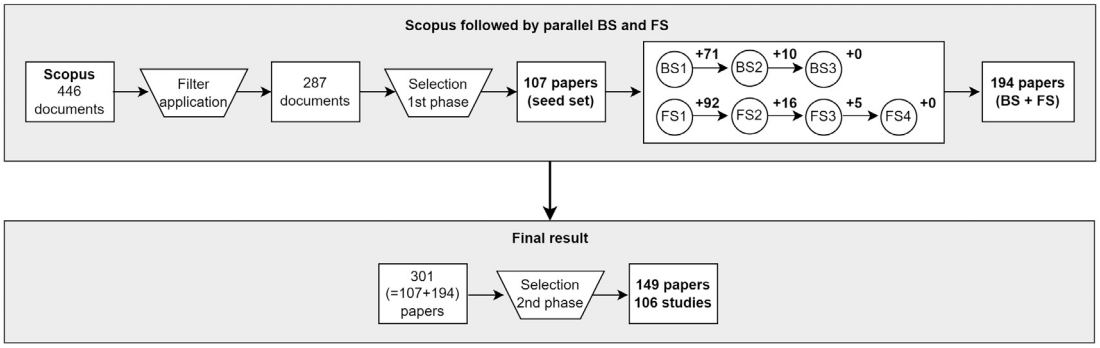


Fig. 3. Study selection process.

and statistical findings play a role, the final decision-making is driven by the expertise and judgment of the domain experts, which is a hallmark of an expert-based approach. We used the scheme proposed by Misirli and Bener [6] to classify the papers into the techniques employed for BN structure construction and kept the papers describing only the use of expert knowledge as we have just defined. Papers describing any of the other techniques, including their combinations with expert knowledge, were discarded.

- EC1: It does not address the use of BNs whose structures were constructed using expert knowledge (e.g., papers (i) not providing sufficient information on which technique was used to construct the graph structure, (ii) describing other techniques such as structure learning from data, or (iii) focusing on the parameter learning step of BN construction).

We applied the IC/EC of the second phase at the content level since we could not identify the technique used for structure construction examining only the titles and abstracts of the papers. At the end of the selection process, a total of 149 papers were included in our final set. Given that multiple papers report different aspects of the same study, we decided to group them to avoid bias in the analysis of the results. Thus, our report is based on studies, not papers. To enable the grouping of papers into studies in a systematic and auditable manner, we define a "study" in this article as a research effort carried out by the same group of researchers with the aim of investigating the application of BNs to solve a specific SE problem. Furthermore, a study may have been published over the years in several papers that may present BNs constructed using the same approach, with refined or modified topologies, i.e., different versions of the model (or set of models as some studies may involve different projects and organizations), e.g., changing the configuration of variables/relationships or adding/removing variables. For instance, we grouped Freire et al. (P24) and Freire et al. (P25) into one study (S8). At the end of the grouping process, we had 106 unique studies. The complete list of selected papers is available in our supplementary material.

To mitigate bias in the study selection process, we first applied the iterative approach proposed by Pérez et al. [65]. Two researchers participated in this activity that aimed to examine the agreement level between them and avoid possible misunderstandings regarding the application of the IC/EC. Given that we divided our study selection into two phases with different sets of IC/EC, both researchers independently examined a set of 15 papers randomly selected from the pool of 287 papers obtained from the Scopus database (first phase) and another set of 15 papers from the pool of 301 papers achieved after the snowballing approach (second phase). During this process, they recorded their decisions on whether to include or exclude each paper. We then calculated the Cohen's Kappa coefficient (k) to measure the agreement level between them. If $k \leq 0.8$, the researchers involved in the process should discuss those papers in which they have disagreed and refine the IC/EC. A new set of 15 papers is selected and examined, and the process continues until $k > 0.8$, indicating that both researchers consistently applied the IC/EC. After achieving such a level of agreement, the remaining papers were examined solely by the first researcher. We executed this process for each phase and obtained $k = 1$ after performing one iteration. We believe this was due to the following reasons: in the first phase, we were intentionally inclusive concerning the IC/EC as we wanted to achieve a spectrum of papers as wide as possible; in the second phase, as we were specifically interested in papers describing expert-based BN structures, the classification scheme of Misirli and Bener [6] helped us apply a consistent rationale to exclude the papers not falling into our IC. Besides, both researchers participated in the design of the study protocol.

3.3. Data extraction

We extracted the following data from each paper:

- Source of publication (e.g., journal, conference, workshop)
- Authors
- Title
- Venue
- Year of publication
- Methods for BN structure construction (RQ1)
- Validation procedures (RQ2)
- The level of focus at which the BN was applied (RQ3)
- SE area (RQ4)
- BN tools (RQ5)
- Challenges and limitations (RQ6)
- Trends and future directions (RQ7)

One researcher (data extractor) extracted and recorded the data in a spreadsheet. Another researcher (data checker) was responsible for reviewing the extracted data, and in case of disagreement, both researchers had discussions until they reached a consensus.

3.4. Classification scheme

Different facets were derived based on our RQs. The facet regarding RQ1 is the method for expert-based structure construction. Examining only metadata such as titles, abstracts, and keywords proved inadequate for discerning the methods employed to construct the graphical structure of BNs. Therefore, we delved into the sections of the papers where the authors detailed their construction of expert-based BN structures and compiled a list of methods. Then, we inductively developed higher-level categories to encompass each method within this list. The forthcoming section provides a comprehensive account of our final classification with examples from the literature.

The facet regarding RQ2 is the validation procedure to evaluate the effectiveness of the BN structure constructed with expert knowledge. We carefully read the validation or model evaluation sections and

particularly searched for any performance measures and/or approaches used to assess the BNs.

For RQ3, the facet concerns the level of focus at which the BNs constructed with expert knowledge are applied in SE. We adopted the following classification [66,67]:

- *Strategic level:* At the strategic level, decisions are typically long-term and high-level, focusing on overall goals, directions, and policies. BNs can be used to support these decisions by modeling complex relationships and uncertainties. Example: A software company might use a BN constructed with expert knowledge to evaluate the potential success of entering a new market or developing a new product line. The network could incorporate factors like market trends, competition, technological innovation, and regulatory environment, allowing executives to make informed strategic decisions.
- *Tactical level:* Tactical decisions are medium-term and focus on how to implement the strategic goals. They often involve allocating resources, scheduling, and planning. Example: A project manager might use a BN to assess the risks associated with different development methodologies or tools for a specific project. By modeling the relationships between various factors such as team expertise, project complexity, budget constraints, and deadlines, the BN could help select the most appropriate development approach.
- *Operational level:* Operational decisions are short-term and focus on the day-to-day activities of the organization. They are often concerned with efficiency and effectiveness in executing tasks. Example: A software development team might use a BN to troubleshoot a recurring issue in a software system. By modeling the dependencies between different components and potential failure points, the BN could help identify the most likely root cause of the problem and guide the team in taking corrective action.

It is worth noting that the papers are not straightforward in stating the level of focus at which the BNs were applied. Thus, we relied on our inference capabilities as researchers to answer RQ3.

Reading the titles, abstracts, and keywords, we also classified the papers based on the knowledge areas defined in the Software Engineering Body of Knowledge (SWEBOK) [68]. Therefore, the facet for RQ4 is defined as the SE topic.

The facet related to RQ5 is the tooling support for building BN models constructed with expert knowledge. We read the sections in which the authors addressed BN construction and extracted a list of the tools used during this process.

The facet for RQ6 covers the challenges and limitations identified in building and applying expert-based BN structures in SE. We extracted a list based on the authors' descriptions by reading the sections in which they addressed the limitations of the proposed BNs.

Regarding RQ7, we read the sections in which the authors provided their final remarks and conclusions of the paper and extracted a list representing the facet related to future directions in the applications of BNs constructed with expert knowledge.

Based on these seven facets, we classified all primary studies and answered the RQs in the next section. One researcher made the classification for each of the seven aspects by reading the papers, and another researcher reviewed the classification. In case of doubtful assignments, both researchers read the paper and had discussions until they reached an agreement.

4. Results and analysis

This section presents the results and analysis of the systematic mapping. The list of selected studies is available in our supplementary material.[1] Our selection process returned 149 papers grouped into 106 unique studies published between 1996 and 2023.

[1] https://doi.org/10.5281/zenodo.13526475

4.1. RQ1: Expert-based methods for BN structure construction

The first RQ in this SMS investigates the state-of-the-art construction of expert-based BN structures in terms of methods/approaches employed in the SE domain. Next, we describe the different methods identified and extracted from the included studies. We inductively classified these methods into two dimensions: data gathering & analysis and reasoning mechanisms.

The **data gathering & analysis** dimension comprises methods for collecting and analyzing data related to the experience and viewpoints of domain experts and findings from the literature regarding SE aspects relevant to building the structure of a certain BN model. It also includes analyzing software data and artifacts. Table 2 lists the data gathering methods we identified. Next, we describe each of them:

Interviews: Interviews are a widely used method to elicit expert knowledge in SE. This method involves engaging with individuals with specialized knowledge and experience in various aspects of software development. After conducting interviews, the collected data are typically analyzed to extract meaningful insights and patterns. This analysis can help identify common themes, trends, and valuable nuggets of knowledge shared by the experts. In the specific context of this study, the data analysis aims to identify relevant variables and relationships that delineate the DAG of the BN modeling the SE problem at hand. However, we observed that the use of interviews among the selected studies tends to vary in terms of how knowledge is acquired from the data. Some studies adopted interviews in a sound, systematic approach, describing that the collected data were analyzed using well-established coding techniques available in the qualitative research literature (e.g., S3 and S5). Others only mentioned that the collected data were analyzed (e.g., S87) or stated that the knowledge engineers conducted interviews with domain experts to build the DAG of the BN model (e.g., S8, S24, S26, S69, and S81) without detailing the employed analysis procedures. As a method used to elicit expert knowledge that serves as a basis for model construction, it is important that the data resulting from the interview process are appropriately analyzed.

Focus group: Using focus groups to elicit expert knowledge involves gathering a group of subject matter experts to engage in a collaborative discussion about specific topics or issues related to software development. For example, in this study, we classified workshop and brainstorming sessions under the umbrella of focus groups, given their shared emphasis on group interactions and discussions. The studies primarily applied focus groups in two ways: to validate factors and to elicit factors. For instance, S2 utilized a focus group to validate relevant factors previously identified through a literature review. In another example, S3 demonstrated focus groups' versatility in both applications. Specifically, in P4, the authors used focus groups to complement the knowledge elicitation process, discussing and agreeing on a set of value factors initially identified through interviews. Conversely, in P7, the researchers chose to elicit value factors directly from focus groups, presenting this method as a viable alternative to conducting interviews for data collection.

Survey with practitioners: When it comes to eliciting expert knowledge to build BN structures, surveys serve as a valuable method for capturing the opinions and experiences of practitioners in a structured and quantifiable manner. The survey questions are carefully crafted to address specific aspects of interest, such as factors influencing decision-making or dependencies between different variables. For example, S62 adapted an existing model consisting of 24 common project risk factors used in a survey with practitioners who were asked to make adjustments by adding/removing factors to build a new model. S67 gathered data through a questionnaire to identify cause–effect relationships among relevant factors to construct their initial BN. In S104, the authors built a cross-company BN analyzing and coding textual survey data containing practitioners' responses regarding the main causes and effects of requirements engineering problems. As with interviews, reporting how survey data are analyzed varies from poor to

Table 2
Data gathering methods.

Method	Studies
Interview	[S1] [S3] [S5] [S8] [S11] [S24] [S26] [S33] [S67] [S69] [S74] [S81] [S86] [S87]
Focus group	[S2-S9] [S17] [S19] [S24] [S27] [S37] [S48] [S65] [S71] [S73-S75] [S77] [S81] [S99]
Survey with practitioners	[S62] [S67] [S68] [S72] [S77] [S86] [S89] [S101] [S103] [S104]
Literature research	[S2] [S4] [S6-S14] [S20] [S21] [S25] [S27] [S29] [S30] [S33] [S35-S38] [S41] [S42] [S44] [S50] [S52] [S54] [S57] [S58] [S64] [S67-S69] [S71-S73] [S75-S79] [S83] [S85-S87] [S89] [S91-S94] [S96] [S101] [S102] [S104] [S106]
Selection of variables from a dataset	[S4] [S6] [S38] [S40] [S42] [S97]
Extension of existing models	[S56] [S62]
Delphi method	[S7]
Case study	[S42]

comprehensive explanations in the studies. We address the role of data analysis methods later in this section.

Literature research: Another method commonly used to leverage expert knowledge for BN structure construction is literature research. Researchers and software practitioners rely on the evidence from the existing literature to reflect on variables and relationships to be incorporated into their models. The concept of literature research in our SMS encompasses different approaches. It can mean searching for insights or structured knowledge in the scientific literature (e.g., S25 and S69), guidelines (e.g., S7), reports (e.g., S101), and standards (e.g., S102), conducting a literature survey (e.g., S30), or performing a more rigorous approach that involves an SLR (e.g., S72). For example, S69 applied BNs to a risk management approach to estimate the necessary percentage of staff training and the associated training costs. The authors verified that staff may be required to undergo training due to risk factors identified based on research findings from other studies. The authors of S7 constructed BNs to assess Scrum-based software development methods and exploited the Scrum Guide for constructing their BN structures. To construct their BN for reliability assessment of safety-critical software of nuclear power plants, the authors of S102 reflected upon the IEEE Standard 1012. S72 conducted an SLR in which they identified more than 50 attributes of different categories influencing the estimation of user stories. Thus, we can see the differences in how this method is used and reported. Some studies limited themselves to indicating that the relevant factors in the model were identified through a literature review (e.g., S2) or from the existing literature (e.g., S4 and S8), giving a vague idea of the systematization and rigor of the search process.

Selection of variables from a dataset: The initial structure of a BN can be delineated based on the selection of variables from existing datasets relevant to the SE problem being tackled. Calling on their previous experience in constructing BN models, the authors of S4 used the set of variables from the Tukutuku dataset as a starting point to identify the factors considered by the domain expert when estimating effort in Web applications. Once the knowledge engineer sketched out and explained the variables, the domain expert suggested removing irrelevant variables and adding new ones. Another example is S38, in which the authors used the COCOMO II model variables and existing studies to propose a causal model for software effort prediction. Similarly, S40 constructed a BN for productivity estimation based on COCOMO 81 cost factors.

Extension of existing models: Experts can construct BN structures by extending existing models. This involves either adding new functionality or new aspects not present in the original model or adapting an existing model to serve as a framework for a new BN. For example, S29 developed BNs to model the knowledge and relationships among variables in software development projects, specifically for resource estimation. These networks enable the prediction of resources required for a desired quality level and, conversely, can also estimate the quality achievable with specified resources. While S29's BNs are valuable for resource estimation, their analysis is trial-based, involving various scenarios [69]. Consequently, S56 extended these BNs into decision networks, empowering project managers to make decisions regarding activities that could enhance process or software quality. Such decision networks aim to identify optimal solutions or the best decisions within specified cost criteria for a given project. This involved the incorporation of decision nodes and utility nodes into the subnets of the BNs in S29, facilitating the determination of the optimal decision or a set of software activities that maximizes utility. Another example is S62, which adapted a model containing common risk factors. The adapted model version was used as a basis for a survey with practitioners. The new BN was then created, taking into account the survey results.

Table 3
Data analysis methods.

Method	Studies
Grounded Theory	[S3] [S5] [S74] [S103] [S104]
Statistical analysis	[S30] [S34] [S42] [S54] [S68] [S77] [S91] [S106]
Software analysis	[S59]

Delphi method: Researchers can leverage the Delphi method to make the process of eliciting expert knowledge more systematic and less biased when compared to focus groups, for example. The authors of S7 decomposed the nodes of their BN into initial elements representing key software process factors which were presented to experts. Then, the authors used the Delphi method to refine the sets of elements through one-to-one discussions.

Case study: Case study research is an investigative approach that leverages quantitative methods, qualitative methods, or a blend of both, known as a mixed research approach. The goal is to gather information from various sources with the aim of examining a phenomenon within its natural settings [70]. To investigate the impact of structural quality on managerial maintenance indices, the authors of S42 conducted a case study involving 20 open-source software (OSS) projects. The aim of the study was to examine software metrics and explore their influence on maintenance production, duration, and productivity. They collected metrics data from a set of popular and active projects, building a dataset suitable for statistical analysis. One of the steps performed was selecting and fine-tuning the variables influencing the BN structure. Expert input was essential at this stage to determine the order of variables that shaped the causal relationships defined by the model.

Table 3 lists the data analysis methods that we describe next:

Grounded Theory: Grounded Theory (GT) is a qualitative research methodology that is often used for theory generation by systematically analyzing data without preconceived notions or predefined structures [71]. Researchers collect data through different methods, such as interviews, observations, and open-ended surveys. In the context of SE, we can find either full GT studies aimed at theory generation or the application of its principles to analyze qualitative data. The outcomes of both approaches have been used to build the DAG of BNs. For example, in S74, the authors built the core fragment of their BN model using the outcome of a full GT study in which they interviewed twenty-five professionals from ten organizations. The authors of S103 relied on survey data that were coded manually following GT principles. Thus, GT plays a role in constructing expert-based BN structures as it helps identify relevant variables represented by the key concepts and categories that

emerge from qualitative data. Relationships between these variables are explored and defined based on the grounded understanding of the SE phenomenon under study. Conversely, S5 collected interview data that were analyzed using principles from the GT method. The resulting codes represented factors to be used as part of the DAG construction step. The outcome of the analysis was used in a focus group in which the experts suggested removing irrelevant factors or adding new ones to the model.

Statistical analysis: Domain experts may reflect on the results of statistical analyses to define BN structures. Descriptive statistics, correlation analysis, regression analysis, and other statistical techniques are applied to the data. Experts evaluate the statistical results to validate or adjust their initial judgments. They may identify potential causal relationships or dependencies between variables. For example, S34 performed statistical analysis of the ISBSG dataset to identify variables influencing the proportions of different defect types. Also, experts analyzed the nature of these influences in terms of direction and strength. The analysis involved preparing a list of potential (continuous, nominal, and ordinal) predictors based on expert experience and data availability, besides examining categories of potential nominal predictors and setting new categories that group many similar low-frequency states for certain variables. Connections between predictors and dependent variables were identified using Spearman's rank correlation coefficient, Kruskal–Wallis analysis of variance, histograms, and box plots, and the validity of these connections was examined from a causal perspective. In addition to the previous measures, S34 used Phi, Cramer's V, and contingency coefficients to discern correlations and associations among predictors to incorporate only the independent ones into the model. Due to data issues, the study highlights an important point: instead of automatically generating the model from the ISBSG dataset, a more sensible strategy was to build the model incorporating expert knowledge and judgment about the statistical analysis results. Such an approach enabled the addition of predictors not present in the dataset or predictors that were present but not correlated/associated with dependent variables according to the analysis' results but influenced them based on the experts' beliefs.

Software analysis: Experts can reflect on software analysis results, such as static and/or dynamic analyses, to build BN models. For example, S60 employed static analysis to extract information about software elements' dependencies that served as a basis for the construction of the graph structure of the models.

The **reasoning mechanisms** dimension comprises methods that support domain experts in establishing the types of reasoning appropriate to the relationships they want to model in the DAG. In addition, it involves using techniques to either guide the experts' reasoning to specific aspects of DAG construction they want to focus on or assist in applying BN concepts. We classified the reasoning mechanisms using a two-step approach. First, we identified the idioms applied to construct the DAG of the BNs. Later, to have a better understanding of the conceptual model used as a reference while building the DAG of the BNs, we identified knowledge engineering and software engineering techniques that were employed by the knowledge engineers. Table 4 summarizes our findings and maps the identified idioms and techniques. For example, we identified that the measurement idiom was applied in combination with using a top-down approach to build the BN, Software Development Lifecycle (SDLC) frameworks, Goal-Question-Metric (GQM), abstraction sheets, and d-separation dependencies. Next, we summarize our results regarding the reasoning mechanisms identified.

Idioms: The use of idioms for constructing BN structures relies on the concept of generally applicable "building blocks" which function as solution patterns. Idioms describe distinct BN fragments that embody generic forms of uncertain reasoning employed by experts during BN construction and can be integrated to form more comprehensive BNs [19]. Only a few studies explicitly mentioned using idioms to build BNs, so we examined the BN structure and its description in the papers to identify them[2]. We identified the following types of idioms: definitional/synthesis idiom (e.g., S7 and S20), cause-consequence idiom (e.g., S20 and S21), measurement idiom (e.g., S27 and S48), induction idiom (e.g., S25), and reconciliation idiom (e.g., S43). It is worth mentioning that in terms of classifying the applied idioms, we strictly followed the idioms' structure presented in Neil et al. [19] and Fenton and Neil [35]. For example, S41 claimed that, for each factor, they defined an appropriate measurement scale, which could indicate that they applied the measurement idiom. However, when building the BN, instead of using the classic measurement idiom structure or indicator nodes, they simply decomposed the node for the factor into multiple nodes, and we interpreted it as if they applied the definitional/synthesis idiom instead. As a result, we found that the most used idioms in the SE domain are the cause-consequence (89.62%), definitional/synthesis (83.02%), and measurement idioms (23.58%). In our supplementary material, we detail how we identified each type of idiom for each study.

The prevalence of the cause-consequence idiom indicates that BNs are highly considered for modeling cause–effect relationships. However, few studies comprehensively explain how their causal assumptions are supported. For example, in S4 [24], the authors mention using the timeline of events to identify cause–effect relationships in the context of Web effort estimation. S74 [46] depicted the effects of technologies and risk mitigation strategies on risk factors. S104 [72] encourages the integration of cause–effect learning mechanisms into defect causal analysis meetings. An initial step is related to aggregating knowledge obtained from successive within-company causal analysis events to build a deeper understanding of defects' cause–effect relationships using BNs. The results of this approach culminated in the investigation of cross-company cause–effect relations for critical RE problems, involving the design of a globally distributed family of surveys to establish an empirical foundation on RE practices and issues. A rigorous qualitative analysis of the survey data helped increase the reliability of the causal relations embedded in their BN model. Another example is S29 [73]. Besides being based on the experts' understanding of cause and effect, many of the relationships between variables have been prompted by empirical results reported in various sources. Despite the importance of justifying the basis of their causal assumptions (as outlined by S4, S74, S104, and S29), this is not common practice in studies reporting BN applications in SE.

Parent divorcing: Parent divorcing is a technique employed in BN construction through which additional variables are added to the model to reduce the probability distributions and/or keep the BN structure simpler. For example, S4 highlights the use of divorcing as an optimization procedure performed by knowledge engineers when the BN structure is assumed to be close to final, but the number of probabilities that need to be elicited or learned for the network needs to be reduced. Another example is S85, which presents a BN-based approach to Web quality. The authors collected and refined quality characteristics, built the model structure based on them, and derived the model parameters. However, given the existence of a node with nine parents within one subnet of the model, it was necessary to restructure this subnet to avoid combinatory explosion during the elicitation of the node probability tables, which was done by creating synthetic nodes to simplify the BN fragment. Parent divorcing is a classic example of applying the definitional/synthesis idiom.

D-separation dependencies: Another approach employed during BN construction is examining the BN graph structure to verify that any identified d-separation dependencies align with the types of variables employed and the assumed causality. D-separation dependencies play a role in pinpointing variables influenced by evidence originating from other variables within the BN [74]. Examples of studies mentioning the use of this approach are S4, S12, and S94.

[2] We could not identify the idioms employed in S64 as it did not provide the BN structure.

Table 4
Reasoning mechanisms.

Idiom	Technique	Studies
Definitional/Synthesis	SDLC	[S27-S29] [S32] [S41] [S47] [S49] [S71] [S91] [S94] [S97] [S99] [S101] [S102] [S105] [S106]
	Risk-centric modeling	[S64] [S74] [S86] [S93] [S100] [S104]
	GQM	[S10] [S12] [S18] [S66] [S81] [S85]
	Quality model	[S53] [S75]
	Value model	[S3]
	AHP	[S79]
	Top-down approach	[S7] [S8]
	Parent divorcing	[S3] [S4] [S6] [S7] [S85] [S87]
	D-separation dependencies	[S4] [S12] [S94]
	None	[S2] [S5] [S9] [S13-S15] [S17] [S19] [S20] [S22] [S25] [S26] [S30] [S31] [S33] [S35-S40] [S42-S44] [S48] [S50-S52] [S54] [S56-S58] [S60] [S63] [S65] [S67] [S70] [S72] [S73] [S76-S78] [S80] [S84] [S88-S90] [S92] [S96] [S98] [S103]
Cause-consequence	SDLC	[S23] [S27-S29] [S32] [S41] [S47] [S49] [S71] [S91] [S94] [S97] [S99] [S101] [S102] [S105] [S106]
	Risk-centric modeling	[S64] [S86] [S93] [S100] [S104]
	GQM	[S10] [S12] [S18] [S66] [S81] [S85]
	Quality model	[S53]
	Value model	[S3]
	Top-down approach	[S7]
	Parent divorcing	[S3] [S4] [S6] [S7] [S85] [S87]
	D-separation dependencies	[S4] [S12] [S94]
	None	[S2] [S5] [S9] [S11] [S13-S15] [S17] [S19-S22] [S24-S26] [S30] [S31] [S33] [S35] [S37-S40] [S42-S46] [S48] [S50-S52] [S54-S61] [S63] [S65] [S67-S69] [S76-S78] [S80] [S82-S84] [S88-S90] [S92] [S95] [S96] [S98] [S103]
Measurement	SDLC	[S27] [S29] [S47] [S71] [S94] [S99] [S101] [S102] [S105]
	GQM	[S1] [S66] [S81]
	Top-down approach	[S7]
	Abstraction sheets	[S1]
	D-separation dependencies	[S94]
	None	[S16] [S17] [S20] [S22] [S25] [S31] [S33] [S34] [S48] [S56] [S70] [S98]
Induction	None	[S22] [S25]
Reconciliation	None	[S22] [S43]

SDLC: A popular reasoning framework for constructing BNs is using the structured phases of the SDLC as a reference, employed by 17 studies. It models the risks, tasks, and outcomes associated with each stage of software development, from requirements gathering and design to implementation and maintenance. BNs built within this framework not only reflect the sequential or iterative nature of software projects but also implicitly explain the causality assumptions between key software development activities, aiming to manage uncertainty and risk at each lifecycle phase. For example, S41 used the Rational Unified Process (RUP) as a reference for predicting project effort. They used the cause-consequence idiom to model the relationship between the main factors of the RUP lifecycle and the definitional/synthesis idiom to break down factors into measures. S27 modeled the waterfall lifecycle to predict software defects early in the development process and used the cause-consequence similarly to S41. However, they applied the definitional/synthesis idiom to organize the BN into subnets, and they had subnets for key processes of the SDLC such as "specification and documentation," "design and development," and "testing and rework." Further, they used indicator nodes to judge the qualities of different processes, e.g., for "testing process quality", the suggested indicators are "quality of documented test cases," "testing process well defined," and "testing staff experience" (see P48).

Risk-centric modeling: Six studies used a risk-centric modeling framework as the basis for building the DAG of the BN. This framework is centered around the direct relationships and interactions between risk elements such as threats, vulnerabilities, controls, and impacts. BNs constructed under this framework aim to capture and quantify the probabilistic dependencies between these risk factors, providing a focused analysis of how risks can affect a software system and the effectiveness of risk mitigation strategies. In this context, all the studies applied the definitional/synthesis and the cause-consequence idioms. For example, S74 built a BN for predicting risks in software projects, given the technologies used (i.e., risk drivers) and mitigation strategies employed. For this purpose, the definitional/synthesis idiom was applied to organize a hierarchical structure of risks (e.g., "integration risks" and "environment risks" as parent nodes of "infrastructure risks"). Further, the cause-consequence idiom was employed to model the cause–effect relationship between technologies (e.g., "JWT" and "Keystore") and mitigation strategies ("offline data persistence") and risk factors (e.g., "security risks").

GQM paradigm: The GQM paradigm was used by seven studies to construct BN models as it guides experts' reasoning when thinking about a BN structure in a goal-oriented approach. While applying the GQM, we identified that the definitional/synthesis, cause-consequence, and measurement idioms were employed. For example, S1 used the GQM paradigm with experts to select software metrics for their BN model, employing indicator nodes. Another example is S18. As the problem definition and key performance indicators (KPIs) are derived from the initial goal of the model, the authors applied the GQM paradigm to identify relevant KPIs (i.e., definitional/synthesis idiom) and their relations (i.e., cause-consequence idiom) to build the causal structure of the BN proposed in their study. S85 presented an approach for modeling Web quality using BNs. The authors gathered quality criteria specific to Web applications from existing work. They refined them by applying the GQM paradigm, which limited subjectivity and improved rigor through structured reasoning when confirming the selection of subcriteria and finding metrics to the model. Thus, the authors of S85 used the criteria collected and refined through the GQM process to build the model's graphical structure as they represented the nodes of the BN.

Top-down approach: A top-down approach was employed by S7 and S8. In both cases, they identified a top-level target node (i.e., for S7, Process quality, and for S8, Teamwork quality) and decomposed it into factors that could be observable by the models' users in a similar fashion as in GQM the Goals are decomposed into Questions and later into Metrics. Applying this approach has resulted in using the definitional/synthesis and cause-consequence idiom. For example, in S7, the definitional/synthesis was applied to break down a node

representing the personal characteristics of the Product Owner, a Scrum accountability, into several nodes (e.g., communicator and negotiator). Conversely, the cause-consequence idiom was employed when modeling cause–effect relationships with observable consequences, such as the Product Owner's impact on the Product Backlog's quality and the Product Backlog's quality in having effective Sprint Planning events.

Quality model: S53 and S75 used quality attributes such as reliability, usability, performance, and maintainability as the primary nodes in a BN. Both studies applied the definitional/synthesis and cause-consequence idioms. S53 built the DAG of the BN by sequencing the actions the software should perform, their causes, and test outcomes. S75 focused on estimating software strategic indicators (e.g., "product quality") by decomposing them into quality factors (e.g., "code quality", "testing status", and "software stability") and for each quality factor, applying the definitional/synthesis idiom decomposing them into measures.

Value model: S3 built BNs by modeling the various factors and their interdependencies that contribute to the overall value delivered by a software system or project. They used the cause-consequence idiom to model the effect of value factors on the delivered value. A common scenario was for a BN to include multiple value dimensions (i.e., nodes) such as "market competitiveness," "customer satisfaction," "business value," and "cost efficiency". As a result, such value dimensions were aggregated, using the definitional/synthesis idiom, into an "overall value" node.

Analytic Hierarchy Process (AHP): S79 built a BN by employing the AHP for selecting Commercial-Off-The-Shelf (COTS) software. For this purpose, they deconstructed complex decision-making into a multi-level hierarchical structure of objectives, evaluation criteria, and attributes. In particular, S79 built the BN with three levels. The upper level contains nodes that represent COTS candidates. The intermediate level contains nodes that represent evaluation criteria. Finally, the lowest level contains nodes that represent attributes associated with evaluation criteria. As a result, S79 demonstrated how AHP assists in constructing BNs by systematically identifying and organizing the factors influencing decision outcomes. By integrating AHP's structured approach to prioritization and the utility functions that capture the relative preferences or weights of criteria, the resulting BNs can effectively model the probabilistic relationships and trade-offs inherent in intricate decisions, thereby facilitating a comprehensive and quantitatively supported decision-making process.

Combining different methods for building BN structures is a common practice. For example, researchers have conducted focus groups with experts. In these focus groups, experts assess variables from existing datasets to select the relevant ones or validate factors elicited through interview data analyzed with GT principles. So, the methods/approaches are combined in a way that they coexist to enhance knowledge elicitation. Figs. 4 and 5 show an overview of the identified methods and how they are combined to elicit and organize the expert knowledge that serves as a basis for structure construction. The numbers in the nodes represent the number of studies using the method, whereas the numbers in the arcs represent the number of studies in which the two methods were combined. Further, the node size is directly proportional to the number of studies that applied the technique it represents. Such a representation helps us emphasize an important point we introduced earlier in this section: the lack of consistency regarding the rigor in applying such methods. For example, many studies reported conducting interviews with experts, but very few provided an explanation of how the collected data were analyzed. This can be seen in Fig. 4, as the number of studies conducting interviews exceeds the number of studies analyzing the collected data. Similarly, the number of studies conducting surveys with practitioners differs from the number of studies analyzing their responses (at least the ones properly explaining that they did and how). The searches in the existing literature also vary considerably, sometimes suggesting a simple consultation was performed and sometimes making it clear that a collaborative effort was invested in an SLR.

Such a variation in structure construction and how studies report it poses a challenge to mapping and classifying the employed methods. It is also worth highlighting that some studies do not mention how expert knowledge was exploited, already providing the BN structure with the final variables and relationships without detailed explanations. In these cases, we classified the studies into the category labeled "Not specified". Another identified situation is when the study states that the model variables were identified from the literature and that the model structure reflects expert knowledge (no explanation of "how it was done" is provided). Thus, we could not assign any method to how domain experts contributed to building the DAG. We could only attest that the authors analyzed the existing literature and gathered the model variables from it.

From a high-level perspective, studies generally divide the construction of the expert-based structure of BNs into two steps: identifying the variables and defining the relationships among them. However, from a lower-level perspective, the approach used to execute these steps varies throughout the studies. Some structures are built by consistently applying different methods to achieve a DAG that reflects the experts' mental models as genuinely as possible. Others rely less on these methods or appear to be constructed ad hoc. Another important step of the DAG construction is defining the states each variable will take on. How these states are defined is also neglected. This may indicate that researchers often make arbitrary decisions to build BN structures according to their understanding of the problem, constraints, and capabilities, to the detriment of more solid and rigorous approaches that would lead to structures that are more faithful to the modeled problem domain.

Such a lack of systematic guidance in the context of SE contributes to the described scenario of constructing expert-based BN structures. Not taking the required care in limiting subjectivity and improving process rigor can result both in structures that are too complex, hindering reasoning under uncertainty through the model and impacting the construction of probability tables, and in structures that are too simple, which can compromise the model's ability to support practitioners' decision-making, hindering the model's long-term use in the organizations. Hence, the variations reported in the studies regarding the expert-based structure construction of BNs in SE point out the investigation of new solutions to this problem as a potential area for further research.

In general, we observed that the insufficient information about BNs and the rationale behind the decisions to construct them impedes their expansion in SE. As proposed by Tosun et al. [29], a comprehensive framework outlining key elements of BN construction would benefit our community. This framework could serve as a valuable resource, aiding researchers and practitioners in leveraging BNs to tackle various challenges within the SE domain. Table 5 summarizes the main findings associated with RQ1.

4.2. RQ2: Validation techniques

Our results show that researchers and practitioners building BNs use a broad spectrum of validation procedures. So, our aim here is not to present them quantitatively by providing an exhaustive list of all types of validation we identified. Instead, by identifying them, we primarily focus on understanding if the validation process is well-established within the SE community, e.g., presenting the most used procedures and observing if there is an agreement regarding the more appropriate techniques to validate the BN models. The ones interested in an exact quantitative picture of the validation procedures can consult our supplementary material.

We observed that the studies report a set of performance assessment measures, statistical tests, and other model validation measures, which are calculated based on a comparison between actual data and model prediction. Next, we summarize them.

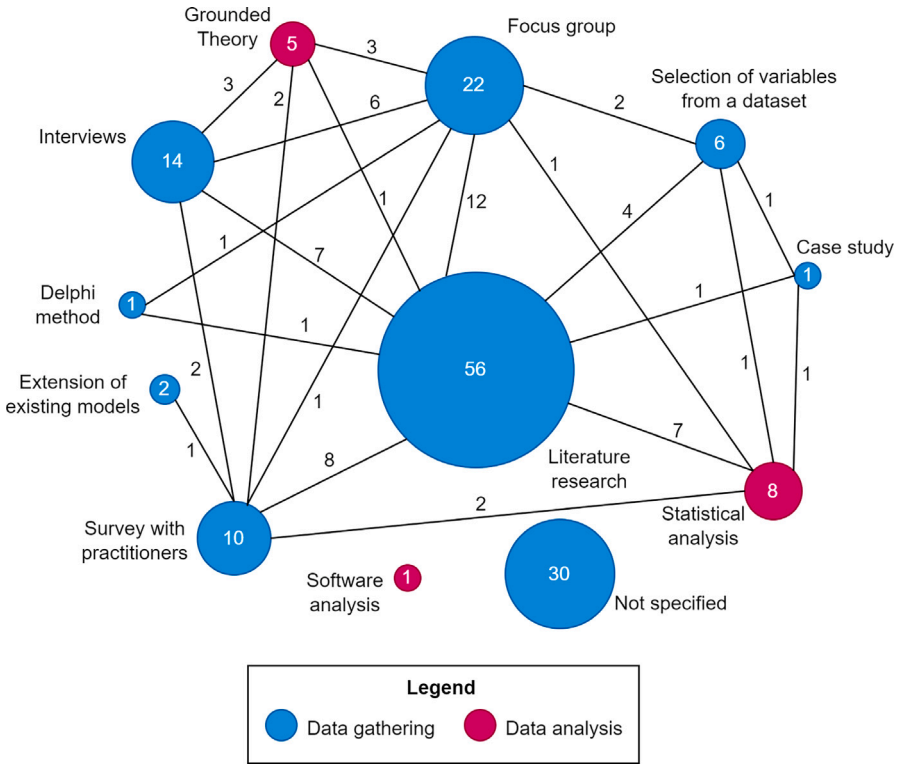


Fig. 4. Data gathering & analysis.

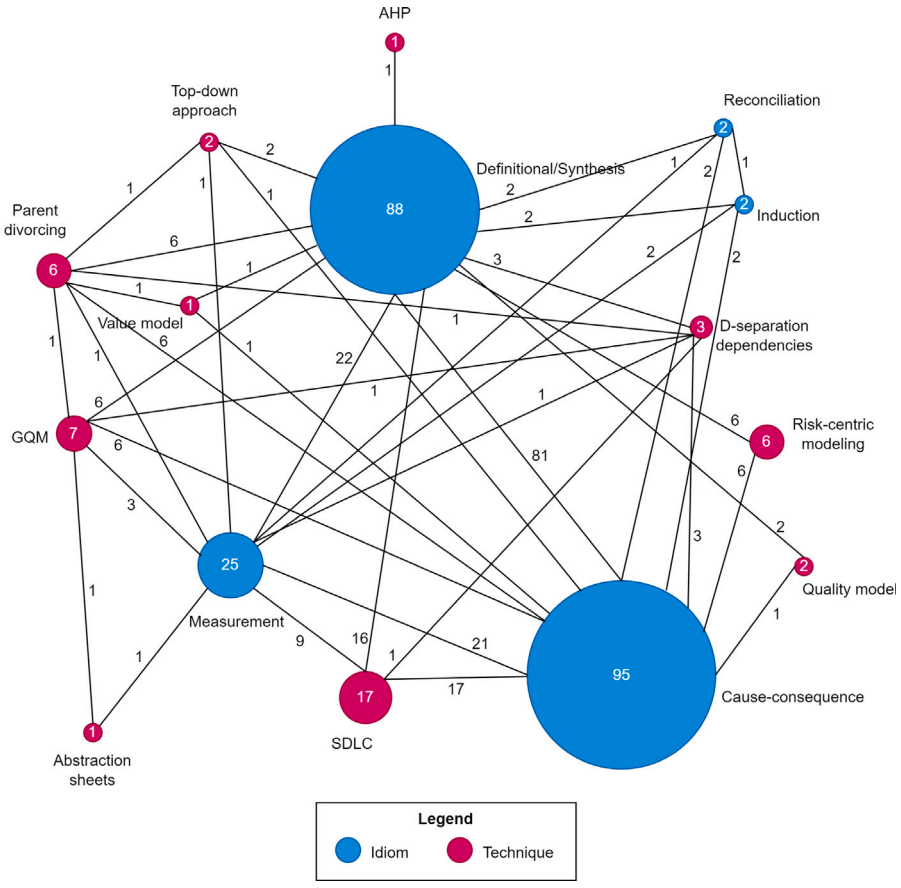


Fig. 5. Reasoning mechanisms.

Table 5
Takeaway points - Expert-based methods for BN structure construction (RQ1).

No.	Finding description
5.1	Utilization of interviews, focus groups, surveys, literature research, and other methods for expert knowledge elicitation.
5.2	Only 14 (out of 106) described the methods employed to analyze the gathered data (i.e., Grounded Theory, statistical, and software analysis), with varying levels of detail in reporting.
5.3	Dominance of cause-consequence and definitional/synthesis idioms, though often not explicitly mentioned in studies.
5.4	Techniques like parent divorcing and examining d-separation dependencies are used to optimize BN structure and align it with the types of variables and assumed causality.
5.5	Adoption of SDLC, risk-centric modeling, and other frameworks to guide BN construction.
5.6	Studies often combine different methods, such as focus groups and interviews, to enhance the knowledge elicitation process for BN structure construction.
5.7	Absence of consistent guidelines in constructing expert-based BN structures, leading to varied approaches.
5.8	General neglect in studies to discuss how the states of nodes are defined, crucial for BN accuracy.
5.9	The findings point to a need for a comprehensive framework outlining key elements of BN construction in SE to aid researchers and practitioners.

Table 6
Takeaway points - Validation techniques (RQ2).

No.	Finding description
6.1	A prevalent method for validating BNs often involves the creation of hypothetical scenarios to test the model's predictions and evaluate its behavior under various conditions.
6.2	Validation commonly focuses on predictive performance, using statistical measures like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and others to assess the accuracy of the BN's predictions.
6.3	There is a notable absence of universally accepted measures for evaluating BN models in the SE community, suggesting the need for standardization.
6.4	Validation typically includes an iterative cycle, where experts provide feedback to refine the model based on test results, leading to adjustments in parameters or the model's structure.

Table 7
Level of focus at which expert-based BNs are applied in SE.

Level of focus	#Studies
Tactical	94 (88.68%)
Strategic	12 (11.32%)
Operational	0

A significant number of studies validated their BNs through model walkthrough and outcome adequacy (also referred to as simulated scenarios and predictive accuracy, respectively, in some studies). Validation procedures also included case studies and comparisons with other methods. Many studies used toy examples, describing model usage without validating it against actual data. Other procedures and measures include Mean Absolute Error (MAE), Relative Absolute Error (RAE), Mean Magnitude of Relative Error (MMRE), Root Mean Squared Error (RMSE), Root Relative Squared Error (RRSE), Magnitude of Relative Error (MRE), Balanced Mean Magnitude of Relative Error (BMMRE), Median Magnitude of Relative Error (MdMRE), Mean Magnitude of Relative Error relative to the Estimate (MEMRE), Median Magnitude of Relative Error relative to the estimate (MdEMRE), Prediction at level k (Pred(k)), Sensitivity, Precision, Recall, k-fold cross validation, Error rate, Accuracy rate, Fisher's exact test, Coefficient of determination (R^2), and many others.

Given the wide range of validation procedures and measures, we do not provide an explanation for each of them (the reader can consult the list of studies for details regarding BN validation). This variation indicates a lack of consensus in the SE community on the most appropriate techniques for evaluating BN models. This is in line with the findings of Tosun et al. [29], even though the authors focused on BNs for software quality prediction, and we considered the whole domain of SE. As our focus is on BNs with structures constructed using expert knowledge, it is worth mentioning that, in general, the validation procedures do not include a specific evaluation of the DAG. We are aware that building BNs is an iterative process, and validating the DAG is commonly based on the experts' opinions until a final version is achieved. However, this validation has the same shortcomings we identified in the applications of the structure construction methods, i.e., lack of necessary rigor. There is no well-established procedure specific to validate the DAG. We commonly observed that some studies reported the need for DAG adjustment after the complete BN was built and the inferences were tested. When unsatisfactory outcomes are achieved, researchers calibrate parameters or change the DAG until good results are obtained. Thus, we believe that investigating sound and more rigorous DAG validation is an area worth putting effort into. Table 6 summarizes the main findings associated with RQ2.

4.3. RQ3: Level of focus

We found that expert-based BNs in SE are predominantly applied at the tactical level (88.68%) and, to a lesser extent, at the strategic level (11.32%), with no applications at the operational level (see Table 7).

The high number of studies applying BNs at the tactical level indicates that researchers and practitioners often rely on the experience and expertise of domain experts to address managerial aspects of software development. This may be due to the tactical level being more directly applicable to solving specific problems or making medium-term decisions within software projects. The relatively lower number

Table 8
Takeaway points - Level of focus (RQ3).

No.	Finding description
8.1	The majority of studies (88.68%) apply expert-based BNs at the tactical level, underscoring their use in addressing managerial aspects of software development. This trend highlights the reliance on expert knowledge for solving specific problems or making mid-term decisions within software projects.
8.2	Only 11.32% of studies explore BNs at the strategic level, indicating a potential area for further research and application.
8.3	No applications of expert-based BNs at the operational level, likely due to the preference for data-driven approaches like machine learning in data-rich environments.

Table 9
SE topics covered in the studies.

Knowledge area	#Studies
Software engineering management	41 (38.68%)
Software quality	36 (33.96%)
Software maintenance	8 (7.55%)
Software requirements	7 (6.6%)
Software testing	6 (5.66%)
Software design	5 (4.72%)
Software engineering process	2 (1.89%)
Engineering foundations	1 (0.94%)

of studies applying BNs at the strategic level suggests a gap or limiters in practical applications concerning using expert-based BNs for making long-term decisions.

The absence of studies applying expert-based BNs at the operational level in SE might reflect the prevalent perception that this level, which concerns daily activities and processes, is not the primary focus for BN applications. This trend could be due to the operational domain's typical abundance of data, making data-driven approaches like machine learning more suitable and effective than expert-based systems. Consequently, the unique advantages of expert-based BNs, such as capturing nuanced expert knowledge, might be overshadowed by the efficacy and applicability of machine learning in data-rich operational environments. It is worth mentioning that our results do not mean that BNs have not been applied at the operational level. We did not find studies focusing on this level because we focused on expert-based BNs. Table 8 summarizes the main findings associated with RQ3.

4.4. RQ4: SE areas of expert-based BN applications

Following the SWEBOK classification, Table 9 outlines SE topics where BNs with expert-based structures have been applied, emphasizing their popularity based on study counts. The most frequently addressed topic is software engineering management, covering different aspects such as effort, schedule, and cost estimation, resource allocation, risk management, and trade-off analyses. Following closely is the utilization of BNs in predicting faults, failures, and software reliability within the context of software quality.

Comparing our findings with Misirli and Bener's [6], we noted that after ten years, software engineering management and software quality continue to be areas of great interest for the application of BNs in SE, with the preponderance of the former when we focus on expert knowledge as the technique used for structure construction. The fact that software engineering management is the most popular topic when considering the use of expert knowledge to build BN solutions supports certain claims regarding the needs of managers in software development. For example, Rique et al. [75] found that, due to several challenges regarding data availability, managers rely on a set of factors to make decisions, including their personal experience and domain experts' opinions. Based on their results, they claim that data are relevant but not sufficient for making managerial decisions, pointing to the need for hybrid solutions such as BNs. Our findings strengthen the ones of Rique et al. as they unveil the representative role of expert knowledge in applying BNs to address the managerial aspects of SE. Table 10 summarizes the main findings associated with RQ4.

4.5. RQ5: BN tools

Table 11 lists the tools used to build the BN models and the number of studies in which they were employed. While the table highlights the variety of tools used, it is imperative to delve deeper into the implications of these choices for the research community. The most popular tools are AgenaRisk, Netica, and Hugin, which were also reported in the works of Tosun et al. [29] and de Sousa et al. [7]. Some studies have also implemented their own tools to manipulate the BN models.

When defining this RQ, we aimed to identify the most used software tools to build BNs whose structures were constructed using expert knowledge. Although the studies have commonly focused on the probability tables and inference aspects of BNs when mentioning these tools in the papers, they are expected to provide their users with graphical interface capabilities to support the visualization of the graph structure of the models.

The choice of BN tools plays a pivotal role in determining the quality and outcomes of research. Tools must align not only with the technical requirements of the research but also with the expertise of the users. Selection should be guided by a tool's capacity for expert knowledge elicitation, data integration, and effective representation of BN models. The right tool can significantly enhance the depth and breadth of research findings.

An intriguing aspect of our findings is the prevalent use of paid tools like AgenaRisk for building BNs. AgenaRisk has an easy-to-use GUI, which allows users to easily manipulate the DAG and also contains powerful features for defining the conditional probabilities and performing simulations. However, this preference raises questions about the accessibility of these tools, especially for researchers or practitioners operating under constrained budgets. The financial implications of tool selection could significantly influence the scope and scale of BN research, potentially limiting the diversity of participants and perspectives in this field. In some cases, financial constraints might lead researchers to use tools that lack robust native GUI features, like Weka and JavaBayes. It is worth mentioning that such GUI features are crucial for effective prognosis and diagnosis in BNs. The absence of such features can limit the practicality and applicability of these tools in complex BN tasks, potentially restricting their use to more theoretical or simplified applications. Future tool development should prioritize comprehensive GUI capabilities to enhance usability and analytical power.

As researchers navigate these GUI limitations, they must also contend with the ever-changing landscape of available BN tools, which is as evidenced by the discontinuation of tools like PowerSoft and SERENE, and the merger of GeNie and SMILE into BayesFusion. These changes not only affect the availability of these tools but also pose challenges in maintaining consistency across longitudinal studies and replicating previous research. This dynamic nature underscores the importance of choosing tools with stable support and development trajectories for long-term research projects.

Table 10
Takeaway points - SE areas of expert-based BN applications (RQ4).

No.	Finding description
10.1	38.68% of studies apply expert-based BNs in software engineering management, addressing effort, schedule, cost estimation, resource allocation, risk management, and trade-off analyses.
10.2	Software quality, including fault, failure, and reliability prediction, is a key focus area, with 33.96% of studies emphasizing BN applications.
10.3	Less explored problems include ones related to software maintenance, design, requirements, and testing.
10.4	The continuing emphasis on software engineering management and quality over a decade reflects the persistent need for expert-based decision support in these areas.

Table 11
BN tools.

Tool	#Studies
AgenaRisk	19
Netica	16
Hugin	12
Own tool	10
SERENE	4
GeNie	3
Elvira	2
Smile	2
Weka	2
PowerSoft	2
Bayes Server	1
B-course	1
BNJ	1
JavaBayes	1
Julia	1
UnBBayes	1
WinBugs	1
Not specified	44

Finally, a concerning observation is the non-disclosure of BN tools in 44 studies. This lack of reporting undermines the principles of transparency and replicability in scientific research. It hampers the ability of other researchers to validate, replicate, or build upon previous work. We advocate for a more stringent approach to documenting tool usage in BN research, emphasizing the need for complete and transparent reporting. Table 12 summarizes the main findings associated with RQ5.

4.6. RQ6: Challenging aspects of applying expert-based BNs

We identified some challenges researchers and practitioners face when building BN structures using expert knowledge. Table 13 lists them in terms of study counts. Next, we provide a brief description of these challenges.

Validation procedures: We captured several aspects regarding validation procedures described in the studies. For example, in S1, validation was only conceptual. S7 highlighted that the number of simulated scenarios and case studies might have been insufficient to make conclusions about the BN effectiveness for any Scrum-based software project. The authors of S8 mentioned the need to validate their BN in real projects. S11 emphasized the difficulty in obtaining reliable data to validate the proposed BN model. We observed that the validation aspects refer to the complete BN, after the probability distributions are elicited and the models are ready to perform inferences. This way, the need to calibrate the parameters or adjust their graph structures is based on the outcome of the models. Thus, specific validation of the structure's effectiveness is not performed. If the models' outcome is satisfactory, the BN structures are assumed to be appropriate for the job they were designed for.

Experts' availability: It is not always possible to have domain experts available to participate in BN construction. For example, S3 stated that the main challenge was the availability of key stakeholders to attend the knowledge elicitation meetings as they are in charge of strategic decision-making in the organization and have limited time. Besides busy schedules, other aspects that can influence experts' availability are business and marketing constraints. S4 advocates that in the construction of hybrid models (in that case, expert-based structure and data-driven probabilities), more than one expert, preferably from several companies, should participate in the graph structure construction. However, such situations are sometimes unfeasible, e.g., when experts are from consulting companies sharing the same market.

Expertise level: To obtain an effective effort estimation model, S4 highlighted that the most important aspect relates to the experts' knowledge of such a domain. According to the authors, BNs built by experienced experts will likely demand minimal validation. Regarding this challenge, the authors of S80 mentioned that the dependency on human capabilities may have threatened the validity of their results due to a lack of information or expertise, recognizing that other approaches could be used, e.g., formal methods of causality discovery and structure learning from data (even though the difficulty in obtaining data of required volume and quality remains as we discussed previously in this paper). So, having experts available to participate in the knowledge elicitation process is insufficient. It is crucial that the ones involved in the model construction have a deep and comprehensive understanding of the problem domain being addressed. Although many years of experience can help in facing this challenge, the research community can investigate strategies to support experts during the activity of defining variables and relationships for BN structures. One alternative that has been proven promising in addressing several aspects of SE is the use of large language models (LLMs).

Complexity handling: As the number of variables and relationships increases in the expert-based structure, so do the probabilities to be elicited. Although parameter learning can be performed automatically in the presence of historical data, handling the structure complexity is critical to ensure the feasibility of building the model. For example, S4 highlighted that this is an issue when building purely expert-based models, as the entire construction and validation of a BN can be effort- and time-consuming and last several months. The largest amount of time involves eliciting the probabilities, which are impacted by the BN structure. The authors addressed this challenge by creating new nodes suggested by the knowledge engineer and validated by the domain expert, who also proposed additional changes to simplify the BN structure. S7 also recognized this challenge as the authors pointed out that not considering the unnecessary complexity of the BN structure was a limitation of an initial version of their model. Complexity handling is not only necessary to reduce the effort to elicit the probabilities. BNs can assist in modeling SE problems of inherently high-level complexity, which influences the structure construction of the models. This is highlighted by S81 when the authors mentioned that the expert might have fatigued during the DAG construction due to its duration. If proper care is not taken, this can lead to low-level models.

Structure representativeness: Another relevant point is how representative of the SE phenomenon the BN structure is. Although handling complexity by limiting the number of variables and relationships is important, we should take care when deciding which factors are fundamental to the model. That was one of the challenges faced by S7 during

Table 12
Takeaway points - BN tools (RQ5).

No.	Finding description
12.1	AgenaRisk, Netica, and Hugin are the most commonly used tools for building BNs, highlighting their prominence in the field.
12.2	The preference for paid tools like AgenaRisk raises concerns about accessibility for researchers with limited budgets. This choice implies that financial considerations play a significant role in tool selection.
12.3	Some tools, such as Weka and JavaBayes, lack robust native GUI features, crucial for effective prognosis and diagnosis in BNs. This limitation suggests that these tools might be more suitable for theoretical or simplified applications rather than complex BN tasks.
12.4	The discontinuation of certain tools (e.g., PowerSoft, SERENE) and the merger of others (e.g., GeNie and SMILE into BayesFusion) indicates a dynamic and evolving landscape of BN tools. This evolution can impact the consistency and replication of longitudinal studies.
12.5	The non-disclosure of BN tools in a significant number of studies (44) hinders transparency and replicability in scientific research. This trend underscores the need for clearer documentation and reporting of tool usage in BN research.

Table 13
Challenges of constructing the structure of expert-based BNs.

Challenge	#Studies
Validation procedures	14
Experts' availability	4
Expertise level	3
Complexity handling	3
Structure representativeness	1
Bias mitigation	1

Table 14
Recommendations and future directions for expert knowledge use.

Recommendation	#Studies
Validation through empirical studies	6
Participation of BN experts	2
Attempt for broader generalizability of models	2
Elicitation process beneficial to experts and companies	2
Model customization for specific contexts	1
Validation datasets built through questionnaires	1
Participatory involvement	1
Trade-off between adaptation to organizational constraints and model construction requirements	1

the construction of their model, as some key agile practices and metrics were not considered at first to be included in the DAG.

Bias mitigation: This challenge relates to the attention and care that should be dedicated to ensuring that the instruments used to gather information for model construction appropriately represent the organizational context in which the BN models will be used. For example, S20 suggested taking additional care during the design of the data collection process to acquire expert knowledge. The goal was to avoid bias during data gathering through questionnaires, as the one previously used by the authors possibly was not representative enough of their industrial cases. Although this situation occurred in the parameter learning step, bias mitigation strategies such as the one described in S20 apply during structure construction.

4.7. RQ7: Recommendations and future directions in using expert knowledge

Next, we briefly describe future steps or recommendations for expert knowledge use considered in the studies concerning the development and applications of BNs in SE. Table 14 lists them together with the number of studies in which they were mentioned.

Validation through empirical studies: As observed in the previous section, validation procedures constitute a challenge in constructing expert-based BNs. Obtaining reliable data to validate the models is difficult, leading to merely conceptual validations or requiring more empirical ones to draw consistent conclusions about their effectiveness from a broader perspective. Some studies have considered leveraging expert knowledge by using more simulated scenarios and performing additional case studies to evaluate the proposed BNs. As a common practice in SE research for validating new technologies in industrial settings, researchers can design empirical studies involving the participation of domain experts to collect evidence regarding the performance of BN models. For example, the authors of S1 pointed out as future steps the execution of empirical validation by collecting data from practitioners and tools to assess how their solution enhanced decision-making. The authors of S7 and S87 mentioned that evaluating their BN models through industrial case studies is part of their plans for future work.

Participation of BN experts: A recommendation we can provide (that we did not find in most of the studies, at least not explicitly stated) is the participation of people with specialized knowledge in building BNs. S3 is an example study addressing this issue. In their specific context, the authors highlighted the need for a company to have the support of BN experts in the structure construction, usage of BN tools, and model validation before integrating the BN into their Value tool and validating it with stakeholders.

Attempt for broader generalizability of models: S4 is a study that has made efforts in this direction. The authors have completed the construction of different expert-based BNs for Web effort estimation, merging their structures to identify common predictors and causal relations. The idea is to investigate the feasibility of constructing a large and unified BN model. S4 also highlights that building more general models also involves the participation of experts from more than one company to enable diverse viewpoints. Otherwise, the BN may not be general enough to be widely employed in different companies.

Elicitation process beneficial to experts and companies: Exploiting expert knowledge for BN structure construction has been seen as beneficial to experts and consequently to their companies, as it enables them to reason under uncertainty in more systematic ways, thinking about the problem domain being modeled from a perspective that may not have been considered before. Thus, this process can change how experts look at certain aspects of software development, leading to a better understanding of them and, in turn, a different treatment by software companies. For example, S4 reported that the domain expert involved in the elicitation process was not completely aware of the factors subjectively considered when estimating effort before he participated in the model construction, often leading him to provide unrealistic estimates to clients. So, the expert knowledge elicitation itself was deemed beneficial. S53 found this process valuable as it provides a logical basis for test construction, advocating that structuring the testing problem as such is essential regardless of the intention to use BNs or not.

Model customization for specific contexts: If, on the one hand, it is important to be capable of building models with broader applicability

by gathering viewpoints and expertise of practitioners from different companies, on the other hand, model customization to fit specific contexts has its value and is in line with the paradigm of context-driven research [76]. For example, S7 included the industry's agile best practices in generic models. Still, it recommended agile process tailoring to adapt to a specific context, as the authors did by building a new version of a BN based on a previous one, which led to improved accuracy of the model's inferences. They concluded that their model is flexible and complies with process tailoring, a common practice in the software industry.

Validation datasets built through questionnaires: In many cases, obtaining sufficient data as well as similar predictive models against which to evaluate the predictive accuracy of BNs is a big challenge. That was the explanation given by S33 for not performing this type of validation. The authors called attention to the high volume of empirical data regarding several details of software projects and their development required for assessing software quality prediction. Due to this reason, the authors planned to leverage expert knowledge through a questionnaire and use the results as a validation dataset.

Participatory involvement: When using expert knowledge to build BN models, it is important to guarantee the direct involvement of stakeholders without disrupting their regular activities. As shared by S75, this is instrumental in stimulating the company's willingness to engage in research initiatives and gaining industrial participation in constructing and validating such models.

Trade-off between adaptation to organizational constraints and model construction requirements: Another lesson shared by S75 is to demonstrate flexibility and creativity in navigating organizational constraints and adhering to both model construction requirements and organizational needs. For example, recognizing that the quality of their estimation model relies on domain experts' knowledge, the authors emphasized this aspect to the team leader. This led to careful personnel assignment, with adaptations in guidelines and instruments to accommodate the assigned personnel's specific daily schedules and constraints. Furthermore, considering the typical restrictions on access to project data and the prevalence of non-disclosure agreements in most organizations, they implemented secure protocols to anonymize data while ensuring their integrity was maximized.

5. Implications for research and practice

This section discusses this study's implications for the SE research community (Section 5.1) and industrial practitioners (Section 5.2).

5.1. Implications for researchers

Our SMS has identified several key areas where current research on BNs in SE can be improved. Here are specific actions researchers can take to address these gaps:

- **Transparency and reproducibility in BN research**: Current literature often lacks sufficient transparency in methodology and results, hindering the proper assessment of the studies' quality and their reproducibility, as previously observed by Tosun et al. [29]. In particular, when employing expert-based BNs for causal modeling, the absence of transparency regarding data gathering and analysis is critical since the rigor employed in these steps is crucial for reliable cause–effects, free of confounding factors and spurious correlations. This finding indicates the need for researchers to develop guidelines for constructing and reporting BNs in SE. It is worth mentioning that Tosun et al. [29] suggested a list of decision points that should be satisfied when reporting BNs, but it targets data-driven BNs and is not empirically validated. We recommend the development of guidelines similar to what we have for traditional empirical software engineering research (e.g., [59,61,77,78]) but for BNs (i.e., expert-based, data-driven, or hybrid), detailing the steps for eliciting expert knowledge, detailing the rationale behind causal assumptions, integrating data, and validating the model.
- **Enhancing objectivity in cause-consequence idiom applications**: The application of the cause-consequence idiom in BNs, when based solely on expert knowledge, poses a significant risk of introducing subjective biases that can skew the model's causality interpretations. This risk is heightened as these causal interpretations are not typically verified by robust statistical techniques such as do-calculus, Difference-in-Differences (DiD), or Granger Causality Tests. To mitigate these risks, researchers should integrate established qualitative research methodologies, such as thematic analysis [79] and GT [71], more deeply into the process of constructing these BNs. This approach helps systematically organize and validate the causal links suggested by experts, ensuring that these links are grounded in a comprehensive analysis of qualitative data rather than subjective interpretations alone. Additionally, incorporating best practices from qualitative research will enhance the transparency and replicability of the causal assumptions, making the BNs more robust and defendable in scientific and practical applications.
- **Generalizability vs. customization of models**: Developing BN models in SE requires a delicate balance between creating generalizable frameworks and tailoring them to specific industry contexts. Researchers should aim to develop adaptable BN frameworks with modular components that allow for easy adjustments to diverse organizational needs while maintaining a core structure that is broadly applicable, for instance, as done by Fenton et al. [73] and Perkusich et al. [80]. An alternative is to focus on context-specific BNs but enhance their transferability (i.e., the extent to which they can be transferred to other settings) by being transparent regarding their scope and construction procedures. This approach makes the models specific enough to provide meaningful, context-specific insights and ensures they remain general enough for broad applicability.
- **Empirical validation and iterative development of BN models**: There is a notable deficiency in the empirical validation of BN models, which questions the reliability of their predictions. Future research should focus on establishing rigorous empirical validation techniques for BNs, including developing guidelines for conducting and reporting on simulation studies and real-world case studies to test the effectiveness of BN models in practical SE settings. Moreover, aligning with the agile mindset prevalent in software development, the construction of BNs can be approached as an iterative process. Initially, BNs may be constructed based solely on expert knowledge when data is scarce. As the system is used and more observational data is collected, these initial models can be refined—both in terms of structure and parameters—using a combination of expert knowledge and data. This refinement process can further benefit from integrating statistical causal inference techniques. Therefore, there is a critical need for guidelines and tools that support this iterative development process of BNs. Such guidelines would detail steps for integrating data-driven and knowledge-driven approaches, thereby enabling a continuous improvement cycle that enhances the BNs' accuracy and relevance over time.
- **Exploring underutilized domains and applications**: BNs are predominantly applied in specific areas like software quality prediction, leaving other potential applications underexplored. Researchers should investigate the application of BNs in underrepresented domains within SE, such as requirements engineering, software maintenance, or security risk analysis. These initiatives should involve not only applying BNs but also adapting and testing their effectiveness in these new contexts.

- **Making model artifacts available**: A critical shortcoming we observed is that making model artifacts available is not common, compromising the practical use of these models by practitioners. So, we recommend researchers make the artifacts developed during model construction available to increase the applicability and usefulness of BNs to address SE tasks. An example of such good practice is the work of Manzano et al. [81], which provides software artifacts and tools implemented to support the development of BNs for estimating software strategic indicators. Additionally, an interesting challenge arises with the coupling of models to specific BN tools. This coupling creates barriers for practitioners who lack access to the specific tool used to produce the model, thereby limiting accessibility and broader applicability. Models created in a universal format could be used across various BN tools, enhancing interoperability, similar to how CSV or JSON enable seamless data exchange across different software applications. This flexibility would allow practitioners to choose the best tools for their specific needs without worrying about compatibility issues.
- **Integrating Generative AI for enhancing BNs**: With the recent advent of Generative Artificial Intelligence (GAI) technologies, it is possible to explore their integration to tackle complexities and biases inherent in BNs. For instance, researchers can explore how to incorporate Large Language Model (LLM) solutions to identify new variables, relationships, or parameter adjustments, which can be evaluated by domain experts for inclusion in the models. This integration can help diversify the perspectives included in the BN and enhance its overall robustness.

These implications highlight the urgent need for a paradigm shift in the research and application of BNs within SE, particularly in the case of modeling causality relying only on domain experts. By addressing these critical gaps, researchers can significantly enhance the methodological rigor, transparency, and applicability of BNs, paving the way for more reliable and actionable insights in SE practices. This not only fosters a systematic and collaborative research environment but also bridges the gap between theoretical research and practical implementation. Consequently, it positions BNs as a pivotal tool in tackling complex challenges in SE, particularly at the strategic and tactical levels, where data is scarce or non-existent.

5.2. Implications for practitioners

This study's focus on the construction and application of expert-based BNs in SE yields several practical implications, especially for practitioners engaged in causal analysis in data-scarce environments:

- **Strategic and tactical decision-making**: In strategic and tactical decision-making, where data is typically limited or non-existent, BN is a viable alternative for performing causal analysis. Practitioners should leverage expert knowledge to construct initial causal models, using BNs to simulate different scenarios and predict outcomes, thus aiding high-stakes decision-making.
- **Participatory model construction**: Practitioners should get involved directly in the BN construction process. This inclusion ensures that the models are not only technically accurate but also culturally and contextually relevant to the organization. Workshops and training sessions can be organized to educate team members on the principles of BNs and gather their input in model formulation.
- **Knowledge sharing and collaborative decision-making**: Organizations can visualize BNs as a means for structured knowledge persistence and sharing. By eliciting knowledge from software engineers and modeling it using BNs, organizations can benefit from transforming the knowledge once implicit in the engineers' brains into assets. Further, the BNs can support software engineers in making complex decisions. Finally, as decisions are made and feedback is incorporated into the BNs, they continuously become more effective. Organizations can kick off knowledge management initiatives centered on developing and maintaining BNs. Practitioners can use internal seminars and regular meetings to review and update BN models as new knowledge is acquired, similar to what is done in business intelligence efforts.
- **Tool availability and accessibility**: The diverse range of both paid and free tools highlights the need for accessible and versatile BN tools that balance functionality with budget constraints. We recommend practitioners choose tools that offer robust GUI features to enhance usability for effective prognosis and diagnosis. A popular free alternative is Netica, while AgenaRisk and Hugin are widely used paid options that typically include more advanced features.
- **Leveraging catalog of BN examples**: BNs have aided practitioners in performing several tasks. These models can assist software teams' stakeholder communication, reduce software estimation workload [82], support defect causal analysis [72], and predict the stability of software requirements specifications [51]. The list of applications is diverse. Our study offers a catalog of expert-based BN examples that SE practitioners can directly adopt or tailor to suit their organizational needs. For example, a software development team aiming to enhance their resource decisions may adopt the model proposed by Fenton et al. [73]. However, it is crucial to check whether the BN's executable files are available and compatible with the tools they use; for instance, Fenton et al. [73] employed AgenaRisk. Suppose the executable files are not accessible or the associated tool does not meet the practitioners' requirements. In that case, the practitioner can use the underlying model as a blueprint to develop their own tailored BN using a more suitable tool. This approach ensures that BNs are not only theoretically sound but are also practically applicable and customized to meet specific operational needs. Also, it can be applied to several SE tasks, offering new avenues to optimize processes and better understand the dynamics of the underlying practices.

In summary, the implications outlined underscore the adaptability and strategic importance of BN in addressing the challenges of causal analysis in environments with sparse data in the context of SE. By incorporating these practices, practitioners can enhance decision-making processes, improve knowledge management, and leverage cutting-edge technologies to refine and adapt their causal models. This study not only highlights the theoretical foundations of expert-based BNs but also demonstrates their practical utility in real-world settings, encouraging a proactive and informed approach to their application in SE. Practitioners are urged to engage with these strategies actively, ensuring that the benefits of BNs are fully realized in enhancing the robustness and efficacy of their projects.

6. Threats to validity

In this section, we explore potential threats to the validity of our SMS and the steps taken to mitigate them. We used the classification schema proposed by Ampatzoglou et al. [83] that distinguishes the following validity aspects in a secondary study: study selection validity, data validity, and research validity.

Study selection validity: As in any secondary study, there is a potential risk of having missed papers or included irrelevant ones, which could compromise the completeness and correctness of our results. The use of ineffective search strings and the selection of inappropriate digital libraries are concerns that should be addressed. In this study, we piloted and used a search string that proved effective, as it was previously adopted in Misirli and Bener's mapping study [6]. Yet there is the possibility of having missed relevant studies because papers may use

terms related to the specific SE problem tackled by the application of BNs (e.g., software maintenance, effort estimation, and software quality prediction) instead of using the broad term "software engineering" in their titles, abstracts, and keywords. Also, using only the Scopus database poses a threat to our study's validity. To address these issues, we complemented our search with parallel backward and forward snowballing [62–64]. Despite our efforts, possible Scopus indexing issues and limitations of parallel snowballing remain potential threats to our search process. To mitigate bias in the inclusion of papers, we followed the methodology proposed by Pérez et al. [65] to achieve an acceptable level of agreement between researchers concerning the understanding and application of our IC/EC. In addition, during the first phase of our selection process, we employed broad IC/EC to be as inclusive as possible regarding the set of papers serving as input to the snowballing approach to achieve a more representative final set of primary studies.

Data validity: Unverified data extraction and inadequate classification schemas are among the most critical threats to the validity of the data, as they could produce unreliable results and conclusions. We operationalized the mapping protocol using spreadsheets to record the data. One researcher conducted data extraction collaboratively with another researcher (one validating the work of the other). Moreover, the selected studies and the extracted data are easily accessible. To improve reliability, we provided excerpts from the papers on which we based to answer our RQs and make conclusions regarding the study's results. For our categorization, we followed an inductive approach (except for RQ3 and RQ4 since their classification was predefined). One researcher classified the collected data into themes which were revised by another researcher and refined collaboratively until a consensus was reached. Despite this, there remains a risk that our classification may not fully represent the entire domain due to subjective bias.

Research validity: Replicability is a major concern in our study due to the need for both qualitative and quantitative data analysis and synthesis to answer the proposed research questions. To address this issue, we developed a detailed protocol based on well-established guidelines [59,61] for conducting mapping studies. Additionally, our review protocol and the intermediate spreadsheets used to control the study selection process and manage the extracted data are available and auditable in our replication package.

7. Conclusion

This paper presented the results of an SMS on constructing the structure of expert-based BNs in the context of SE. We selected and analyzed 149 papers grouped into 106 unique studies published between 1996 and 2023. We extracted information related to methods employed to engineering expert knowledge for constructing the BN structures, validation techniques, level of focus at which expert-based BNs have been applied in SE, the SE topics covered by the application of BNs, challenging aspects of and recommendations for using expert knowledge to create BN solutions.

Our results show that a combination of different methods is commonly used to construct expert-based BN structures. We classified such methods into two dimensions: data gathering & analysis and reasoning mechanisms. The former includes methods such as focus groups, interviews, surveys with practitioners, GT, and statistical analysis. The latter includes the use of idioms and techniques such as GQM and risk-centric modeling used to guide the experts' reasoning when defining variables and relationships to form the DAG of BNs. Also, the prevalence of the cause-consequence idiom reveals the potential of BNs as a prominent tool for causal modeling in SE. The rigor applied to the construction methods varies significantly in the studies, from thorough and robust approaches to BN structures that appear to be constructed in an ad hoc manner.

Other relevant findings are as follows: (i) there is a lack of consensus regarding BN validation as researchers use a wide spectrum of validation procedures and measures, and there is no specific evaluation focusing on the structure effectiveness; (ii) expert-based BNs are mainly applied at the tactical level to tackle SE problems in the context of software engineering management and software quality; (iii) validation procedures, experts' availability, expertise level, complexity handling, structure representativeness, and bias mitigation represent challenging aspects of expert-based BNs; and (iv) recommendations for using expert knowledge include validation through empirical studies, participation of BN experts, attempt for broader generalizability of models, elicitation process beneficial to the companies, model customization, validation datasets built through questionnaires, participatory involvement, and a trade-off between adaptation to organizational constraints and model construction requirements.

Overall, there is substantial room for improvement in constructing expert-based BN structures. To foster a broader application of BNs in SE, it is essential to minimize the inherent bias in expert knowledge, enhance the rigor applied to the methods, and provide a more detailed rationale for model construction. A paradigm shift is necessary to address these challenges and increase the reliability of causal inferences derived from these models, ultimately resulting in better-informed decisions in SE practices.

CRediT authorship contribution statement

Thiago Rique: Writing – original draft, Visualization, Methodology, Formal analysis, Conceptualization, Data curation, Project administration. **Mirko Perkusich:** Writing – review & editing, Validation, Supervision, Methodology, Conceptualization, Project administration. **Kyller Gorgônio:** Writing – review & editing, Supervision. **Hyggo Almeida:** Writing – review & editing. **Angelo Perkusich:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

We have shared the link to our data in the manuscript.

Acknowledgments

Rique is supported by the IFPB qualification incentive program (PIQIFPB) - public notice 109/2021.

References

[1] L. Yao, Z. Chu, S. Li, Y. Li, J. Gao, A. Zhang, A survey on causal inference, ACM Trans. Knowl. Discov. Data (TKDD) 15 (5) (2021) 1–46.
[2] J. Pearl, Causality, Cambridge University Press, 2009.
[3] J. Siebert, Applications of statistical causal inference in software engineering, Inf. Softw. Technol. (2023) 107198.
[4] J. Pearl, The do-calculus revisited, 2012, arXiv preprint arXiv:1210.4852.
[5] N.E. Fenton, M. Neil, Software metrics: roadmap, in: Proceedings of the Conference on the Future of Software Engineering, 2000, pp. 357–370.
[6] A.T. Misirli, A.B. Bener, Bayesian networks for evidence-based decision-making in software engineering, IEEE Trans. Softw. Eng. 40 (6) (2014) 533–554.
[7] A.L. de Sousa, C.R. de Souza, R.Q. Reis, A 20-year mapping of Bayesian belief networks in software project management, IET Softw. 16 (1) (2022) 14–28.
[8] T. Schulz, Ł. Radliński, T. Gorges, W. Rosenstiel, Predicting the flow of defect correction effort using a Bayesian network model, Empir. Softw. Eng. 18 (2013) 435–477.
[9] I. Figalist, C. Elsner, J. Bosch, H.H. Olsson, Breaking the vicious circle: A case study on why AI for software analytics and business intelligence does not take off in practice, J. Syst. Softw. 184 (2022) 111135.
[10] C.W. Mehling, S. Pieper, S. Ihlenfeldt, Concept of a causality-driven fault diagnosis system for cyber-physical production systems, in: 2023 IEEE 21st International Conference on Industrial Informatics, INDIN, IEEE, 2023, pp. 1–8.

[11] H. Jouni, L. Jouffe, E. Tancrede-Bohin, P. André, S. Benamor, P.-P. Cabotin, J. Chen, Z. Chen, K. Conceiçao, N. Dlova, et al., Predicting the evolution of clinical skin aging in a multi-ethnic population: Developing causal Bayesian networks using dermatological expertise, Skin Res. Technol. 30 (2) (2024) e13602.

[12] F.U. Küçüker, B. Yet, Reliability prediction for aircraft fleet operators: A Bayesian network model that combines supplier estimates, maintenance data and expert judgement, J. Oper. Res. Soc. 74 (10) (2023) 2187–2198.

[13] S. Nepal, L.Y. Pomara, N.P. Gould, D.C. Lee, Wildfire risk assessment for strategic forest management in the southern United States: A Bayesian network modeling approach, Land 12 (12) (2023) 2172.

[14] Y. Wu, S. Mascaro, M. Bhuiyan, P. Fathima, A.O. Mace, M.P. Nicol, P.C. Richmond, L.-A. Kirkham, M. Dymock, D.A. Foley, et al., Predicting the causative pathogen among children with pneumonia using a causal Bayesian network, PLoS Comput. Biol. 19 (3) (2023) e1010967.

[15] L.D. Burgoon, M. Angrish, N. Garcia-Reyero, N. Pollesch, A. Zupanic, E. Perkins, Predicting the probability that a chemical causes steatosis using adverse outcome pathway Bayesian networks (AOPBNs), Risk Anal. 40 (3) (2020) 512–523.

[16] A. de Waal, H. Koen, P. de Villiers, H. Roodt, N. Moorosi, G. Pavlin, Construction and evaluation of Bayesian networks with expert-defined latent variables, in: 2016 19th International Conference on Information Fusion (Fusion), IEEE, 2016, pp. 774–781.

[17] F. Lattimore, D. Rohde, Replacing the do-calculus with Bayes rule, 2019, arXiv preprint arXiv:1906.07125.

[18] K.B. Laskey, S.M. Mahoney, Network fragments for knowledge-based construction of belief networks, in: Proceedings of the AAAI Symposium on Mixed-Initiative Reasoning, 1998.

[19] M. Neil, N. Fenton, L. Nielson, Building large-scale Bayesian networks, Knowl. Eng. Rev. 15 (3) (2000) 257–284.

[20] S. Nadkarni, P.P. Shenoy, A causal mapping approach to constructing Bayesian networks, Decis. Support Syst. 38 (2) (2004) 259–281.

[21] M.J. Flores, A.E. Nicholson, A. Brunskill, K.B. Korb, S. Mascaro, Incorporating expert knowledge when learning Bayesian network structure: a medical case study, Artif. Intell. Med. 53 (3) (2011) 181–204.

[22] H. Hu, L. Kerschberg, Improving causal Bayesian networks using expertise in authoritative medical ontologies, ACM Trans. Comput. Healthcare 4 (4) (2023) 1–32.

[23] T. Rique, E. Dantas, D. Albuquerque, M. Perkusich, K. Gorgônio, H. Almeida, A. Perkusich, Shedding light on the techniques for building Bayesian networks in software engineering, in: Anais do III Workshop Brasileiro de Engenharia de Software Inteligente, SBC, 2023, pp. 1–6.

[24] E. Mendes, Using knowledge elicitation to improve web effort estimation: Lessons from six industrial case studies, in: 2012 34th International Conference on Software Engineering, ICSE, IEEE, 2012, pp. 1112–1121.

[25] E. Mendes, Building a web effort estimation model through knowledge elicitation, in: International Conference on Enterprise Information Systems, Vol. 2, SCITEPRESS, 2011, pp. 128–135.

[26] E. Mendes, C. Pollino, N. Mosley, Building an expert-based web effort estimation model using Bayesian networks, in: 13th International Conference on Evaluation and Assessment in Software Engineering (EASE) 13, 2009, pp. 1–10.

[27] E. Mendes, M. Perkusich, V. Freitas, J. Nunes, Using Bayesian network to estimate the value of decisions within the context of value-based software engineering, in: Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018, 2018, pp. 90–100.

[28] R. Saraiva, A. Medeiros, M. Perkusich, D. Valadares, K.C. Gorgônio, A. Perkusich, H. Almeida, A Bayesian networks-based method to analyze the validity of the data of software measurement programs, IEEE Access 8 (2020) 198801–198821.

[29] A. Tosun, A.B. Bener, S. Akbarinasaji, A systematic literature review on the applications of Bayesian networks to predict software quality, Softw. Qual. J. 25 (2017) 273–305.

[30] M. Perkusich, L.C. e Silva, A. Costa, F. Ramos, R. Saraiva, A. Freire, E. Dilorenzo, E. Dantas, D. Santos, K. Gorgônio, et al., Intelligent software engineering in the context of agile software development: A systematic literature review, Inf. Softw. Technol. 119 (2020) 106241.

[31] B.R. Cobb, R. Rumí, A. Salmerón, Bayesian network models with discrete and continuous variables, Adv. Probabilistic Graph. Models (2007) 81–102.

[32] E. Mendes, E. Mendes, Introduction to Bayesian networks, in: Practitioner's Knowledge Representation: A Pathway to Improve Software Effort Estimation, Springer, 2014, pp. 61–71.

[33] N. Fenton, M. Neil, W. Marsh, P. Hearty, Ł. Radliński, P. Krause, On the effectiveness of early life cycle defect prediction with Bayesian nets, Empir. Softw. Eng. 13 (2008) 499–537.

[34] E. Mendes, E. Mendes, Expert-based knowledge engineering of bayesian networks, in: Practitioner's Knowledge Representation: A Pathway to Improve Software Effort Estimation, Springer, 2014, pp. 73–105.

[35] N. Fenton, M. Neil, Risk Assessment and Decision Analysis with Bayesian Networks, Crc Press, 2018.

[36] K.A. Bollen, Structural Equations with Latent Variables, vol. 210, John Wiley & Sons, 1989.

[37] P.R. Rosenbaum, D.B. Rubin, The central role of the propensity score in observational studies for causal effects, Biometrika 70 (1) (1983) 41–55.

[38] J. Grundspenkis, Causal domain model driven knowledge acquisition for expert diagnosis system development, J. Intell. Manuf. 9 (1998) 547–558.

[39] J.A. Maxwell, Using qualitative methods for causal explanation, Field Methods 16 (3) (2004) 243–264.

[40] E. Almeida, P. Ferreira, T.T. Vinhoza, I. Dutra, P. Borges, Y. Wu, E. Burnside, Expert bayes: Automatically refining manually built Bayesian networks, in: 2014 13th International Conference on Machine Learning and Applications, IEEE, 2014, pp. 362–366.

[41] D. Heckerman, A tutorial on learning with Bayesian networks, in: Innovations in Bayesian networks: Theory and applications, Springer, 2008, pp. 33–82.

[42] B. O'Gorman, R. Babbush, A. Perdomo-Ortiz, A. Aspuru-Guzik, V. Smelyanskiy, Bayesian network structure learning using quantum annealing, Eur. Phys. J. Spec. Top. 224 (2015) 163–188.

[43] C. Xiao, Y. Jin, J. Liu, B. Zeng, S. Huang, Optimal expert knowledge elicitation for Bayesian network structure identification, IEEE Trans. Autom. Sci. Eng. 15 (3) (2018) 1163–1177.

[44] C.F. Zheng, L. Jiang, L.Q. Jiang, Z.J. Wu, Application and research of Bayesian network in data mining, Adv. Mater. Res. 532 (2012) 738–742.

[45] C. Kumar, D.K. Yadav, A probabilistic software risk assessment and estimation model for software projects, Procedia Comput. Sci. 54 (2015) 353–361.

[46] E. Dantas, A.S. Neto, M. Perkusich, H. Almeida, A. Perkusich, Using Bayesian networks to support managing technological risk on software projects, in: Anais do I Workshop Brasileiro de Engenharia de Software Inteligente, SBC, 2021, pp. 1–6.

[47] H.G. Kang, S.H. Lee, S.J. Lee, T.-L. Chu, A. Varuttamaseni, M. Yue, S. Yang, H.S. Eom, J. Cho, M. Li, Development of a Bayesian belief network model for software reliability quantification of digital protection systems in nuclear power plants, Ann. Nucl. Energy 120 (2018) 62–73.

[48] A. Senathi, G. Vinod, D. Jadhav, Software reliability based on software measures applying Bayesian technique, in: Proceedings of the Second International Conference on Computer and Communication Technologies: IC3T 2015, Volume 3, Springer, 2016, pp. 167–173.

[49] C. Kumar, D.K. Yadav, Software defects estimation using metrics of early phases of software development life cycle, Int. J. Syst. Assur. Eng. Manag. 8 (2017) 2109–2117.

[50] S. Chatterjee, B. Maji, A bayesian belief network based model for predicting software faults in early phase of software development process, Appl. Intell. 48 (8) (2018) 2214–2228.

[51] J. del Sagrado, I.M. del Aguila, Stability prediction of the software requirements specification, Softw. Qual. J. 26 (2018) 585–605.

[52] R. Fatima, F. Zeshan, A. Ahmad, M. Hamid, I. Filali, A.A. Alhussan, H.A. Abdallah, Requirement change prediction model for small software systems, Computers 12 (8) (2023) 164.

[53] L. Radlinski, A survey of bayesian net models for software development effort prediction, Int. J. Softw. Eng. Comput. 2 (2) (2010) 95–109.

[54] A.T. Misirli, A.B. Bener, A mapping study on Bayesian networks for software quality prediction, in: Proceedings of the 3rd International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering, 2014, pp. 7–11.

[55] D. Rodriguez, J. Dolado, J. Tuya, Bayesian concepts in software testing: An initial review, in: Proceedings of the 6th International Workshop on Automating Test Case Design, Selection and Evaluation, 2015, pp. 41–46.

[56] I.M. Del Águila, J. Del Sagrado, Bayesian networks for enhancement of requirements engineering: a literature review, Requir. Eng. 21 (2016) 461–480.

[57] M. Scanagatta, A. Salmerón, F. Stella, A survey on Bayesian network structure learning from data, Progr. Artif. Intell. 8 (2019) 425–439.

[58] N.K. Kitson, A.C. Constantinou, Z. Guo, Y. Liu, K. Chobtham, A survey of Bayesian network structure learning, Artif. Intell. Rev. (2023) 1–94.

[59] K. Petersen, S. Vakkalanka, L. Kuzniarz, Guidelines for conducting systematic mapping studies in software engineering: An update, Inf. Softw. Technol. 64 (2015) 1–18.

[60] B.A. Kitchenham, D. Budgen, O.P. Brereton, Using mapping studies as the basis for further research–a participant-observer case study, Inf. Softw. Technol. 53 (6) (2011) 638–651.

[61] B. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering, Tech. Rep. EBSE-2007-01, Schoolf of Computer Science and Mathematics, Keele University, 2007.

[62] E. Mourão, J.F. Pimentel, L. Murta, M. Kalinowski, E. Mendes, C. Wohlin, On the performance of hybrid search strategies for systematic literature reviews in software engineering, Inf. Softw. Technol. 123 (2020) 106294.

[63] C. Wohlin, M. Kalinowski, K.R. Felizardo, E. Mendes, Successful combination of database search and snowballing for identification of primary studies in systematic literature studies, Inf. Softw. Technol. 147 (2022) 106908.

[64] S. Alonso, M. Kalinowski, B. Ferreira, S.D. Barbosa, H. Lopes, A systematic mapping study and practitioner insights on the use of software engineering practices to develop MVPs, Inf. Softw. Technol. (2022) 107144.

[65] J. Pérez, J. Díaz, J. Garcia-Martin, B. Tabuenca, Systematic literature reviews in software engineering—Enhancement of the study selection process using Cohen's kappa statistic, J. Syst. Softw. 168 (2020) 110657.

[66] I. Figalist, C. Elsner, J. Bosch, H.H. Olsson, Business as unusual: a model for continuous real-time business insights based on low level metrics, in: 2019 45th Euromicro Conference on Software Engineering and Advanced Applications, SEAA, IEEE, 2019, pp. 66–73.
[67] I. Figalist, C. Elsner, J. Bosch, H.H. Olsson, Fast and curious: A model for building efficient monitoring-and decision-making frameworks based on quantitative data, Inf. Softw. Technol. 132 (2021) 106458.
[68] P. Bourque, R.E. Fairley (Eds.), SWEBOK: Guide to the Software Engineering Body of Knowledge, IEEE Computer Society, Los Alamitos, CA, 2014, Version 3.0, URL http://www.swebok.org/.
[69] T. Noothong, D. Sutivong, Software project management using decision networks, in: Sixth International Conference on Intelligent Systems Design and Applications, Vol. 2, IEEE, 2006, pp. 1124–1129.
[70] P. Runeson, M. Host, A. Rainer, B. Regnell, Case Study Research in Software Engineering: Guidelines and Examples, John Wiley & Sons, 2012.
[71] C. Urquhart, Grounded theory for qualitative research: A practical guide, in: Grounded Theory for Qualitative Research, Sage publications, 2022, pp. 1–100.
[72] M. Kalinowski, P. Curty, A. Paes, A. Ferreira, R. Spínola, D.M. Fernández, et al., Supporting defect causal analysis in practice with cross-company data on causes of requirements engineering problems, in: 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP), IEEE, 2017, pp. 223–232.
[73] N. Fenton, W. Marsh, M. Neil, P. Cates, S. Forey, M. Tailor, Making resource decisions for software projects, in: Proceedings. 26th International Conference on Software Engineering, IEEE, 2004, pp. 397–406.
[74] E. Mendes, M. AbuTalib, S. Counsell, Applying knowledge elicitation to improve web effort estimation: A case study, in: 2012 IEEE 36th Annual Computer Software and Applications Conference, IEEE, 2012, pp. 461–469.
[75] T. Rique, M. Perkusich, E. Dantas, D. Albuquerque, K. Gorgônio, H. Almeida, et al., On adopting software analytics for managerial decision-making: A practitioner's perspective, IEEE Access 11 (2023) 73145–73163, http://dx.doi.org/10.1109/ACCESS.2023.3294823.
[76] L. Briand, D. Bianculli, S. Nejati, F. Pastore, M. Sabetzadeh, The case for context-driven software engineering research: generalizability is overrated, IEEE Softw. 34 (5) (2017) 72–75.
[77] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, Empir. Softw. Eng. 14 (2009) 131–164.
[78] J.S. Molléri, K. Petersen, E. Mendes, An empirically evaluated checklist for surveys in software engineering, Inf. Softw. Technol. 119 (2020) 106240.
[79] D.S. Cruzes, T. Dyba, Recommended steps for thematic synthesis in software engineering, in: 2011 International Symposium on Empirical Software Engineering and Measurement, IEEE, 2011, pp. 275–284.
[80] M. Perkusich, K. Gorgonio, H. Almeida, A. Perkusich, A framework to build Bayesian networks to assess scrum-based software development methods, in: Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE, 2017, pp. 67–73.
[81] M. Manzano, C. Ayala, C. Gómez, A. Abherve, X. Franch, E. Mendes, A method to estimate software strategic indicators in software development: An industrial application, Inf. Softw. Technol. 129 (2021) 106433.
[82] E. Weflen, C.A. MacKenzie, I.V. Rivero, An influence diagram approach to automating lead time estimation in Agile Kanban project management, Expert Syst. Appl. 187 (2022) 115866.
[83] A. Ampatzoglou, S. Bibi, P. Avgeriou, M. Verbeek, A. Chatzigeorgiou, Identifying, categorizing and mitigating threats to validity in software engineering secondary studies, Inf. Softw. Technol. 106 (2019) 201–230.