Shedding Light on the Techniques for Building Bayesian Networks in Software Engineering

Thiago Rique
thiago.rique@ifpb.edu.br
Federal Institute of Paraíba
Princesa Isabel, PB, Brazil

Emanuel Dantas
emanuel.filho@ifpb.edu.br
Federal Institute of Paraíba
Campina Grande, PB, Brazil

Danyllo Albuquerque
danyllo.albuquerque@virtus.ufcg.edu.br
Intelligent Software Engineering
Group - VIRTUS/UFCG
Campina Grande, PB, Brazil

Mirko Perkusich
mirko@virtus.ufcg.edu.br
Intelligent Software Engineering
Group - VIRTUS/UFCG
Campina Grande, PB, Brazil

Kyller Gorgônio
kyller@virtus.ufcg.edu.br
Intelligent Software Engineering
Group - VIRTUS/UFCG
Campina Grande, PB, Brazil

Hyggo Almeida
hyggo@virtus.ufcg.edu.br
Intelligent Software Engineering
Group - VIRTUS/UFCG
Campina Grande, PB, Brazil

Angelo Perkusich
perkusic@virtus.ufcg.edu.br
Intelligent Software Engineering
Group - VIRTUS/UFCG
Campina Grande, PB, Brazil

ABSTRACT

Context Bayesian networks (BNs) have been used to tackle several software engineering (SE) problems, such as risk management and effort estimation. They enable reasoning under uncertainty and have the advantage of incorporating expert knowledge to build more accurate models when sufficient historical data are not available. Software practitioners often encounter a lack of substantial evidence concerning the usability, limitations, risks, and benefits of BNs, as is the case with many other topics in the SE literature. Therefore, there is a need to organize and systematize the existing knowledge in this area. **Objective** This paper aims to provide researchers and practitioners with an overview of the techniques for building BNs in SE. **Method** We conducted a tertiary study following the guidelines available in the SE literature. **Results** We examined six secondary studies. Our findings revealed that expert knowledge emerges as the predominant technique for structure learning and, in conjunction with learning from data using automated tools, is widely employed for parameter learning in BNs. **Conclusion** Despite the attention given to data-driven approaches in SE, it is worth acknowledging the significant value that expert knowledge continues to hold in constructing more accurate and robust models. This observation underscores potential opportunities for developing expert-driven solutions to enhance model building and foster the adoption of BNs in the software industry.

CCS CONCEPTS

• **Software and its engineering** → **Software creation and management**.

KEYWORDS

Bayesian networks, software engineering, tertiary study

1 INTRODUCTION

To improve software quality, increase productivity, and address software development issues, the software engineering (SE) community has proposed several paradigms. Due to the enormous amount of data generated by tools to support software development, one of these paradigms has gained attention: intelligent software engineering (ISE). In general, ISE concerns two different approaches [23]: (i) the application of artificial intelligence (AI) techniques to SE and (ii) the creation of SE solutions specifically designed for intelligent software.

Complementing the first approach (which is used in this study), Perkusich et al. [15] define intelligent technique in a broader perspective, referring to the exploration of data generated by SE tools for various purposes, such as knowledge discovery, reasoning, planning, and decision-making. Among such techniques, Bayesian networks (BNs) are widely acknowledged as practical tools that play a crucial role in supporting decision-making within the field of SE.

While machine learning and deep learning techniques have garnered considerable attention and achieved remarkable outcomes across diverse domains, their lack of interpretability presents challenges for practitioners. In contrast, BNs provide enhanced interpretability and comprehension of the knowledge embedded within their representations. The discernible distinctions and explicit causal relationships depicted by the underlying graph structure of BNs make it more accessible for practitioners to interpret and understand the decision-making process employed by these models [15].

Additionally, the characteristics of an organization and its software development culture influence the availability and use of data for decision-making. There are SE problems for which collecting data might be unfeasible, leading practitioners to consider other factors, such as their experience [17]. This points to another advantage of using BNs, which is the possibility of incorporating expert knowledge when sufficient historical data are not available. From a practical perspective, combining data-driven approaches with expert knowledge has been emphasized as a sensible approach for software development organizations to achieve meaningful decision-making improvements [17, 19].

Practitioners in the field of SE often encounter a lack of substantial evidence concerning the usability, limitations, risks, and benefits of BNs, as is the case with many other topics in the SE literature [11]. Consequently, a critical demand of the SE community is to support and improve the decision-making process of software practitioners when considering the implementation and adoption of specific technologies such as BNs [9]. Therefore, there is a need to organize and systematize the existing knowledge in the area of BNs for SE.

To effectively organize and consolidate the knowledge in this field, conducting a tertiary study proves to be a valuable approach. A tertiary study involves a meticulous investigation that scrutinizes existing secondary studies to address broader research questions [8]. As far as we know, no tertiary study has been conducted to identify and catalog individual secondary studies in this significant research domain.

This tertiary study aims *to provide researchers and practitioners with insights into how BNs have been used in SE.* The contribution of this study is an updated overview of the main techniques used in the construction of BNs, allowing new efforts to be made by the SE community to address the limitations and discover new opportunities regarding the use of these techniques to create more effective models.

The subsequent sections of this paper are structured as follows: Section 2 furnishes a comprehensive account of the employed review method. Section 3 presents the outcomes derived from the data extraction process and engages in a discussion concerning the research questions. Section 4 delineates the limitations observed within this study. Lastly, Section 5 concludes the paper by summarizing the findings and their implications.

2 METHOD

In accordance with the guidelines presented by Kitchenham and Charters [7], we conducted a tertiary study. In addition, we took inspiration from exemplary tertiary studies in SE (e.g., [3, 5]).

2.1 The Research Questions

We aimed to answer the following question: *"How have BNs been used in SE?"* To do so, we depicted it into the following research questions (RQs):

- RQ1: What are the techniques used for structure learning?
- RQ2: What are the techniques used for parameter learning?

2.2 The Search Process

To conduct the search process, we adopted a hybrid strategy that combined a database search using the Scopus digital library and a snowballing approach. Scopus is renowned for its extensive coverage of scientific and technical literature, encompassing a wide array of journals, conferences, and research sources in the field of SE. We chose the hybrid search strategy based on Wohlin et al. [22]. Our initial search focused on the Scopus database, where we sought all pertinent secondary studies related to BNs in SE without restricting the search to a predefined period of time. To devise effective search terms, we considered three key aspects: Bayesian networks, secondary studies, and software engineering. However, we encountered a challenge wherein some papers might not explicitly mention "software engineering" or "software development". Instead, they may use terms related to specific SE problem domains to which BNs are applied (e.g., effort estimation). To mitigate the risk of missing relevant studies, we sought assistance from ChatGPT [1] to identify the most commonly addressed SE problems through the application of BNs. The results obtained from ChatGPT were then used to construct our search string's software engineering subset. In order to validate ChatGPT's output, we initially leveraged our expertise as SE researchers to identify any potential inconsistencies. Subsequently, we conducted an informal search and found no evidence of incoherence, as studies for the suggested problems were retrieved, corroborating the output provided by ChatGPT. Thus, the search string was defined as follows:

(("Bayesian network" OR "Bayesian net" OR "Bayes net" OR "Bayesian belief network") ***AND*** *("systematic review" OR "systematic literature review" OR "systematic map" OR "systematic mapping" OR "mapping study" OR "literature survey")* ***AND*** *("software" OR "reliability" OR "testing" OR "fault diagnosis" OR "requirements engineering" OR "project management" OR "quality assessment" OR "cost estimation"))*

2.3 Study Selection

The inclusion and exclusion criteria were the following:

- Inclusion criteria
  - The study must be directly related to the application of BNs in SE; and
  - The study must be a secondary study.
- Exclusion criteria
  - Published in a non-peer-reviewed channel; or
  - Reviews that appeared as abstracts, work-in-progress papers, posters, and papers not written in English.

Figure 1 presents an overview of the paper selection process. Initially, our search on the Scopus database generated a total of 248 document results. To refine the search, we utilized a filter provided by the search engine, excluding documents from subject areas unrelated to SE (e.g., medicine). The search string used for filtering is presented in our supplementary material. Consequently, we narrowed down the results to 29 documents, encompassing the subject areas of "computer science", "engineering", and "multidisciplinary".

We applied the inclusion and exclusion criteria by screening the papers' titles and abstracts, achieving a total of 5 studies that met the requirements. Further, we employed snowballing using the 5 papers as the seed set. Both backward and forward snowballing techniques were used, with Google Scholar [2] serving as the data source. Two studies from the first iteration were deemed suitable and incorporated as the seed set for the subsequent iteration. However, this second iteration did not yield any additional included papers. At the end of the snowballing process, a total of 7 studies were ultimately included. The final step involved conducting

[1] https://chat.openai.com/
[2] https://scholar.google.com/

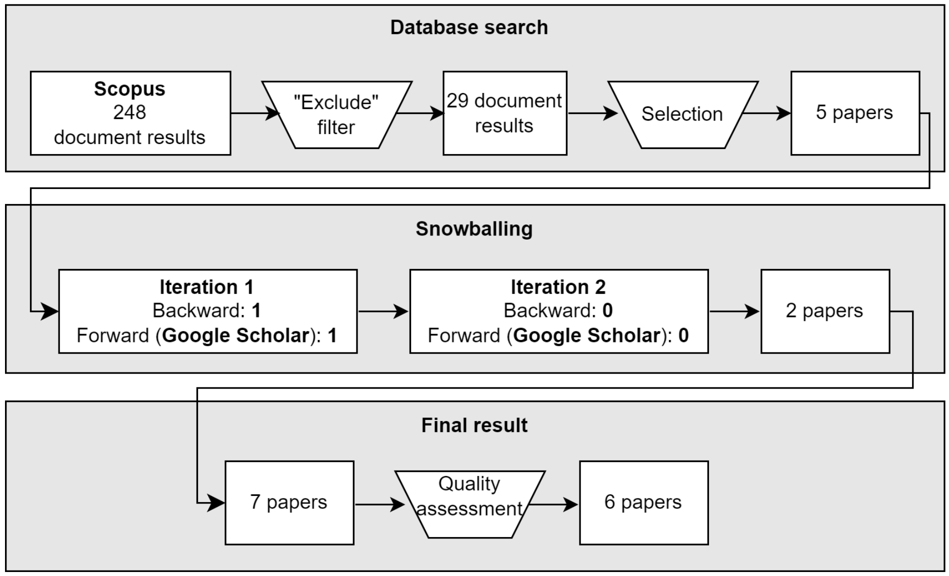

Figure 1: Number of papers in study selection.

a quality assessment, as outlined in Section 2.4. As a result of this evaluation, one (1) paper was deemed unsuitable and subsequently excluded because it did not meet the minimum score based on [5]. This process left us with a final set of 6 secondary studies that were considered for further analysis.

2.4 Quality Assessment

We used the quality criteria established in the tertiary study by Kitchenham et al. [8]. Four specific criteria (in the form of questions (Q#)) were employed to assess the quality of each secondary study:

- Q1. Are the review's inclusion and exclusion criteria described and appropriate?
- Q2. Is the literature search likely to have covered all relevant studies?
- Q3. Did the reviewers assess the quality/validity of the included studies?
- Q4. Were the basic data/studies adequately described?

The evaluation process was applied to the secondary studies in order to assess their adherence to the predefined quality criteria. Each criterion was assigned a numerical score using a point system: Yes (Y) = 1 point, Partial (P) = 0.5 point, No (N) = 0 points. The overall quality score was calculated by summing up the scores for each individual criterion. Consequently, the total quality score for each study ranged from 0 (very poor quality) to 4 (very good quality).

2.5 Data Extraction and Analysis Process

All necessary information for further synthesis was recorded using a Microsoft Excel spreadsheet [3]. We extracted the following data (when provided) from all included secondary studies:

- Bibliographic information (title, abstract, publication year, publication type: conference/journal/workshop)
- Authors and affiliations (organization, country)
- Type of study (systematic literature review, systematic mapping study, literature survey)
- Number of primary studies
- Years covered
- How BNs were applied
  - techniques for structure learning
  - techniques for parameter learning
- Quality score.

The data extraction process was primarily carried out by the first author (data extractor). Following this, the second author (data checker) carefully reviewed the extracted data. To address potential biases in the data, any doubts or uncertainties that emerged were thoroughly discussed among the researchers until an agreement was reached, ensuring a comprehensive examination and resolution of any concerns.

We employed descriptive analysis for the quantitative data (e.g., the number of primary studies in each secondary study) and a thematic analysis approach [1] to analyze the qualitative data. For example, we conducted a categorization process in which the techniques for structure learning were grouped into similar themes.

3 RESULTS AND DISCUSSION

Table 1 presents a summary of the secondary studies examined in our review. It includes information such as the total quality score, year of publication, type of publication venue, number of primary studies, and years covered for each study. Out of the six secondary studies analyzed, four were published in journals, while the remaining two appeared in workshop proceedings. In terms of study types, we identified three systematic mapping studies

[3]https://doi.org/10.5281/zenodo.8174322

Table 1: Secondary studies on BN4SE

Id	Quality	Year	Venue[1]	Type[2]	#Studies	#Years
S1 [16]	2.5	2010	J	LS	23	2002-2010
S2 [12]	2.5	2014	W	SMS	38	1999-2012
S3 [11]	2.5	2014	J	SMS	117	1999-2012
S4 [18]	2.5	2015	W	SLR	41	2010-2015
S5 [21]	4.0	2017	J	SLR	10	1999-2014
S6 [2]	4.0	2022	J	SMS	109	1999-2018

[1] J = Journal; W = Workshop
[2] SLR = Systematic Literature Review; SMS = Systematic Mapping Study; LS = Literature Survey

(SMSs) [2, 11, 12], two systematic literature reviews (SLRs) [18, 21], and one literature survey (LS) [16].

One question guided our investigation: *"How have BNs been used in SE?"* To address the question at hand, the following subsections explore two aspects related to the construction of BNs, each one focusing on the techniques for a specific aspect. Such techniques are illustrated in Figure 2. In line with the terminology employed by Misirli and Bener [11, 12], as well as Tosun et al. [21], we maintained the distinction between structure learning and parameter learning. Structure learning pertains to the construction of the BN's structure, including the identification of nodes and their relationships. On the other hand, parameter learning involves (i) defining prior and conditional distributions for nodes within the network and (ii) performing inference, which entails estimating the parameters of the distributions for continuous variables or conditional probability tables for categorical variables.

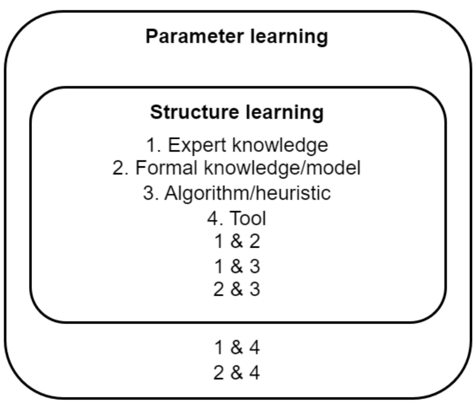

Figure 2: Techniques for structure and parameter learning.

3.1 RQ1: Techniques for structure learning

As explained next, common techniques for constructing the BN structure involve using expert knowledge, data (through an algorithm or a tool), formal knowledge, and combinations of these techniques.

In his survey, Radlinski (S1, [16]) categorized the primary studies based on two factors: expert knowledge and data. The majority of BN models were constructed primarily using expert knowledge, while some models combined expert knowledge and data. BN models were rarely built solely based on data. Misirli and Bener (S2 [12] and S3 [11]), as well as Tosun et al. (S5 [21]), classified the approaches for BN construction in the primary studies according to the following types, ranked from most to least commonly used: expert knowledge, usage of formal knowledge/model, learning from data with an algorithm/heuristic, and learning from data via an automated tool. Various combinations of these techniques were also employed, such as utilizing expert knowledge and learning from data with an algorithm/heuristic, expert knowledge and formal knowledge/model, or formal knowledge/model and learning from data with an algorithm/heuristic. Rodriguez et al. (S4 [18]) did not provide details on how the BN structures were constructed, while Sousa et al. (S6 [2]) only mentioned the use of data and/or expert knowledge without explicitly referring to the construction of BN structures.

The results suggest that expert knowledge emerges as the prevailing technique for building the structure of BNs (e.g., [4]). Given the nature of SE, which demands substantial expertise, domain experts with experience in similar projects and a deep understanding of variable relationships are invaluable in this regard.

The integration of expert knowledge and data represents another frequently employed strategy for constructing BN models. This approach enables experts to validate their assumptions while incorporating valuable information from the available data into the model. By combining expert knowledge and data, not only can the accuracy of the model be enhanced, but also potential biases inherent in the expert knowledge can be recognized and rectified, leading to more robust and reliable results.

It is worth mentioning that constructing BN models exclusively based on data is a rarity. This observation can be attributed to several factors. Firstly, SE data tends to be incomplete, noisy, or inaccurate, posing challenges to relying solely on data-driven approaches [17]. Additionally, SE involves intricate interactions among variables that extend beyond the capabilities of statistical models alone. Hence, incorporating expert knowledge becomes crucial in capturing and understanding the complexities inherent in SE data, complementing the limitations of purely data-driven models.

Another frequently employed technique for constructing the structure of BNs involves the utilization of formal knowledge/models (e.g., [14]). This approach entails leveraging formal designs, such as diagrams or conceptual models used in the software development life cycle, to define the relationships between variables [11]. By incorporating formal models, a more precise and structured representation of the variable relationships can be achieved, leading to enhanced accuracy and reliability in the BN model.

The technique of learning from data with an algorithm/heuristic (e.g., [20]) entails employing algorithms or heuristics to discern the relationships between variables from the available data. This approach proves highly valuable, especially in scenarios where substantial data are accessible, and the relationships between variables are intricate and challenging to capture through expert knowledge or formal models.

Lastly, the results indicate that combinations of these techniques (e.g., [10, 13]) are also prevalent. For instance, the combination of expert knowledge and learning from data with an algorithm/heuristic is commonly used to construct more accurate BN models. Similarly, by combining formal knowledge/models with learning from data using an algorithm/heuristic, a more structured and precise representation of the relationships between variables can be achieved.

In summary, the selection of techniques for constructing the structure of BNs in SE depends on factors such as data availability, the complexity of variable relationships, and the expertise of domain experts. By comprehending the advantages and limitations of each technique, software engineers can make informed decisions and opt for the most suitable approach for building BN models.

3.2 RQ2: Techniques for parameter learning

The techniques used for parameter learning are the same as for structure learning, including additional combinations.

Regarding the definition of model parameters (i.e., probability distributions), various techniques have been explored. According to Radlinski (S1 [16]), the most prevalent approach involves incorporating expert knowledge alongside empirical data. Following this, there are methods that solely rely on empirical data and employ parameter learning algorithms. Additionally, some approaches combine empirical data with expert adjustments, while the utilization of expert knowledge alone is relatively uncommon. Misirli and Bener (S2 [12] and S3 [11]) and Tosun et al. (S5 [21]) identified techniques similar to those found in the structure learning step. These techniques include the use of expert knowledge, formal knowledge, learning from data with algorithms or heuristics, and learning from data through automated tools. Often, a combination of them is employed. For a comprehensive view of the statistics pertaining to each of these techniques, we summarize the findings from the broader study conducted by Misirli and Bener [11]: expert knowledge and learning from data using an automated tool (27.35%), learning from data using an automated tool (19.66%), learning from data with an algorithm/heuristic (17.95%), expert knowledge (15.38%), expert knowledge and learning from data with an algorithm/heuristic (4.27%), expert knowledge and usage of some other type of formal knowledge (2.56%), usage of some other type of formal knowledge and learning from data using an automated tool (1.71%), usage of some other type of formal knowledge and learning from data with an algorithm/heuristic (0.85%), usage of some other type of formal knowledge (0.85%), not specified (9.40%). Sousa et al. (S6 [2]) only mentioned using data and/or expert knowledge without explicitly referring to the parameter learning step.

Misirli and Bener [11] highlight the difficulty in precisely differentiating between parameter learning approaches found in the literature. For example, studies can report using a certain technique for setting the distribution of parent nodes while using a different technique for the rest of the nodes (e.g., [13]). Similarly, some authors can claim they used historical data only to compute the probability tables but mention the usage of a tool that uses an embedded inference algorithm. As a result, a significant number of studies resort to a multi-approach strategy for parameter learning [11].

Similar to structure learning, expert knowledge is a prevalent source frequently utilized for parameter learning, often in combination with other techniques. The findings reveal that experts commonly set the distributions, whereas the inference for the final (output) node is accomplished through learning from data using exact and approximate algorithms or via an automated tool (e.g., [4]).

4 LIMITATIONS

Given that this is a tertiary study, we specifically analyzed secondary studies on the application of BNs in SE. As a result, if a widely discussed application of BNs had many primary studies dedicated to it but lacked a published secondary study, we did not consider it in our analysis. Additionally, due to the scarcity of published secondary studies on the topic, we expanded our scope beyond SLRs and SMSs to also include LSs. The inclusion of these sources was subject to rigorous quality assessment criteria to ensure the reliability and soundness of the results.

To ensure the validity of our conclusions, we undertook a thorough search process in a highly regarded digital database without imposing any restrictions on the publication year. Our intention was to encompass a wide spectrum of relevant papers. Additionally, we adopted a snowballing approach during our search to minimize the possibility of overlooking any pertinent evidence. This strategy effectively strikes a balance between the quality of results and the effort invested in the review process, being a suitable alternative when searching for evidence in this type of study [22].

In our search, we initially considered using the terms "software engineering" and "software development" as the SE-related subset of our search string, following the practices of other SE tertiary studies (e.g., [5, 6, 24]). However, recognizing that these specific terms might not always be present in the papers, we used ChatGPT to identify the most common SE problems addressed using BNs and integrated its insights to broaden our search scope. Nonetheless, there was a potential risk of missing studies applying BNs to specific problems not covered by ChatGPT. To address this concern, we also included the broad term "software" in our search string, thereby reducing the likelihood of overlooking relevant studies for our review.

Given that the first author was responsible for data extraction, a valid concern arises about the potential influence of personal or author bias on the accuracy of the results. To mitigate this possible issue, the second author conducted a thorough review of the extracted data. In cases where conflicts or disagreements arose during the review process, we proactively resolved them through collaborative discussions in a joint meeting. This approach ensured the objectivity of the data extraction process and minimized the impact of individual biases.

The influence of researcher bias on data interpretation is also a potential threat. We extracted themes from studies that presented them in categorized lists or tables. This method facilitated the thematic analysis by streamlining the identification and grouping of related themes. In cases where a theme could not be associated with an existing category, we agreed that a new category was created. While our methodical approach aimed to minimize the impact of researcher bias on data interpretation, it is worth noting that half of the included secondary studies were conducted by the same group of researchers. The same categories for structure learning and parameter learning appear in all three studies and, as a consequence, were the dominant categories in our analysis.

5 CONCLUSIONS

We analyzed and summarized information from six secondary studies on BNs for SE. Our results indicated that expert knowledge is the most used technique for structure learning and the most used for parameter learning in combination with learning from data via an automated tool, indicating that studies applying BNs to SE tend to use a multi-approach fashion to build more accurate models.

By utilizing our results, researchers and practitioners can develop effective research strategies to advance the current state-of-the-art and enhance the overall impact of BNs on the SE practice. Our future steps include investigating the adoption of BNs in the software industry, besides identifying gaps and opportunities in this significant research field.

Despite the attention that has been given to data-driven approaches in SE, expert knowledge has proved invaluable for building more accurate and robust models. For example, in software defect prediction, imagine a software development project with a limited dataset of historical defects. An expert software engineer, familiar with the codebase and development practices, can provide valuable insights into the likely sources of defects. This expertise can guide the selection of relevant variables in the BN, such as code complexity metrics, developer experience, and specific code modules prone to issues. Without expert input, these critical factors may be overlooked, leading to less accurate predictions. This conclusion sheds light on potential opportunities for developing expert-driven solutions to build enhanced models and leverage the adoption of BNs in the software industry.

ACKNOWLEDGMENTS

Thiago Rique is supported by the qualification incentive program of the Federal Institute of Paraíba (public notice 109/2021).

REFERENCES

[1] Daniela S Cruzes and Tore Dyba. 2011. Recommended steps for thematic synthesis in software engineering. In *2011 international symposium on empirical software engineering and measurement*. IEEE, 275–284.

[2] Abraham LR de Sousa, Cleidson RB de Souza, and Rodrigo Q Reis. 2022. A 20-year mapping of Bayesian belief networks in software project management. *IET Software* 16, 1 (2022), 14–28.

[3] Eliezer Dutra, Bruna Diirr, and Gleison Santos. 2021. Human factors and their influence on software development teams-a tertiary study. In *Proceedings of the XXXV Brazilian Symposium on Software Engineering*. 442–451.

[4] Norman Fenton, Martin Neil, William Marsh, Peter Hearty, Łukasz Radliński, and Paul Krause. 2008. On the effectiveness of early life cycle defect prediction with Bayesian Nets. *Empirical Software Engineering* 13 (2008), 499–537.

[5] Rashina Hoda, Norsaremah Salleh, John Grundy, and Hui Mien Tee. 2017. Systematic literature reviews in agile software development: A tertiary study. *Information and software technology* 85 (2017), 60–70.

[6] Helvio Jeronimo Junior and Guilherme Horta Travassos. 2022. Consolidating a common perspective on Technical Debt and its Management through a Tertiary Study. *Information and Software Technology* 149 (2022), 106964.

[7] Barbara Kitchenham, Stuart Charters, et al. 2007. Guidelines for performing systematic literature reviews in software engineering.

[8] Barbara Kitchenham, Rialette Pretorius, David Budgen, O Pearl Brereton, Mark Turner, Mahmood Niazi, and Stephen Linkman. 2010. Systematic literature reviews in software engineering–a tertiary study. *Information and software technology* 52, 8 (2010), 792–805.

[9] Barbara A Kitchenham, Tore Dyba, and Magne Jorgensen. 2004. Evidence-based software engineering. In *Proceedings. 26th International Conference on Software Engineering*. IEEE, 273–281.

[10] Ghazwa Malak, Houari Sahraoui, Linda Badri, and Mourad Badri. 2010. Modeling web quality using a probabilistic approach: An empirical validation. *ACM Transactions on the Web (TWEB)* 4, 3 (2010), 1–31.

[11] Ayse Tosun Misirli and Ayse Basar Bener. 2014. Bayesian networks for evidence-based decision-making in software engineering. *IEEE Transactions on Software Engineering* 40, 6 (2014), 533–554.

[12] Ayse Tosun Misirli and Ayşe Başar Bener. 2014. 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*. 7–11.

[13] Sirsendu Mohanta, Gopika Vinod, AK Ghosh, and Rajib Mall. 2010. An approach for early prediction of software reliability. *ACM SIGSOFT Software Engineering Notes* 35, 6 (2010), 1–9.

[14] Ganesh J Pai and Joanne Bechta Dugan. 2007. Empirical analysis of software fault content and fault proneness using Bayesian methods. *IEEE Transactions on software Engineering* 33, 10 (2007), 675–686.

[15] Mirko Perkusich, Lenardo Chaves e Silva, Alexandre Costa, Felipe Ramos, Renata Saraiva, Arthur Freire, Ednaldo Dilorenzo, Emanuel Dantas, Danilo Santos, Kyller Gorgônio, et al. 2020. Intelligent software engineering in the context of agile software development: A systematic literature review. *Information and Software Technology* 119 (2020), 106241.

[16] Lukasz Radlinski. 2010. A survey of bayesian net models for software development effort prediction. *International Journal of Software Engineering and Computing* 2, 2 (2010), 95–109.

[17] Thiago Rique, Mirko Perkusich, Emanuel Dantas, Danyllo Albuquerque, Kyller Gorgônio, Hyggo Almeida, and Angelo Perkusich. 2023. On Adopting Software Analytics for Managerial Decision-Making: A Practitioner's Perspective. *IEEE Access* (2023).

[18] Daniel Rodriguez, Javier Dolado, and Javier Tuya. 2015. Bayesian concepts in software testing: An initial review. In *Proceedings of the 6th International Workshop on Automating Test Case Design, Selection and Evaluation*. 41–46.

[19] Richard Berntsson Svensson, Robert Feldt, and Richard Torkar. 2019. The unfulfilled potential of data-driven decision making in agile software development. In *Agile Processes in Software Engineering and Extreme Programming: 20th International Conference, XP 2019, Montréal, QC, Canada, May 21–25, 2019, Proceedings 20*. Springer, 69–85.

[20] Nuria Torrado, Michael P Wiper, and Rosa E Lillo. 2012. Software reliability modeling with software metrics data via Gaussian processes. *IEEE Transactions on Software Engineering* 39, 8 (2012), 1179–1186.

[21] Ayse Tosun, Ayse Basar Bener, and Shirin Akbarinasaji. 2017. A systematic literature review on the applications of Bayesian networks to predict software quality. *Software Quality Journal* 25 (2017), 273–305.

[22] Claes Wohlin, Marcos Kalinowski, Katia Romero Felizardo, and Emilia Mendes. 2022. Successful combination of database search and snowballing for identification of primary studies in systematic literature studies. *Information and Software Technology* 147 (2022), 106908.

[23] Tao Xie. 2018. Intelligent software engineering: Synergy between AI and software engineering. In *Proceedings of the 11th Innovations in Software Engineering Conference*. 1–1.

[24] Elijah Zolduoarrati, Sherlock A Licorish, and Nigel Stanger. 2023. Secondary studies on human aspects in software engineering: A tertiary study. *Journal of Systems and Software* 200 (2023), 111654.