# Shedding Light on How Intelligent Techniques can Support Technical Debt Management and Influence Software Quality Attributes

Danyllo Albuquerque
Federal University of Campina Grande (UFCG)
Campina Grande, Paraiba - Brazil
danyllo@copin.ufcg.edu.br

Ferdinandy Chagas
Federal Rural University of the Semi-Arid (UFERSA)
Pau dos Ferros, Rio Grande do Norte - Brazil
ferdinandy@ufersa.edu.br

Everton Guimaraes
The Pennsylvania State University
Malvern, Pennsylvania - USA
ezt157@psu.edu

Graziela Tonin
Federal University of Fronteira Sul (UFFS)
Chapecó, Santa Catarina - Brazil
graziela.tonin@uffs.edu.br

Mirko Perkusich
Federal University of Campina Grande
Campina Grande, Paraiba - Brazil
mirko@embedded.ufcg.edu.br

Hyggo Almeida and Angelo Perkusich
Federal University of Campina Grande
Campina Grande, Paraiba - Brazil
(hyggo,angelo)@embedded.ufcg.edu.br

## ABSTRACT

Technical Debt (TD) is a consequence of decision-making in the development process that can negatively impact Software Quality Attributes (SQA) in the long term. Technical Debt Management (TDM) is a complex task to minimize TD that relies on a decision process based on multiple and heterogeneous data that are not straightforward to synthesize. Recent studies show that Intelligent Techniques can be a promising opportunity to support TDM activities since they explore data for knowledge discovery, reasoning, learning, or supporting decision-making. Although these techniques can improve TDM activities, there is a need to identify and analyze solutions based on Intelligent Techniques to support TDM activities and their impact on SQA. For doing so, a Systematic Mapping Study was performed, covering publications between 2010 and 2020. From 2276 extracted studies, we selected 111 unique studies. We found a positive trend in applying Intelligent Techniques to support TDM activities being Machine Learning and Reasoning Under Uncertainty the most recurrent ones. Design and Code were the most frequently investigated TD types. TDM activities supported by intelligent techniques impact different characteristics of SQA, mainly Maintainability, Reliability, and Security. Although the research area is up-and-coming, it is still in its infancy, and this study provides a baseline for future research.

## CCS CONCEPTS

• **General and reference** → Empirical studies.

## KEYWORDS

Technical Debt, Intelligent Techniques, Systematic Mapping Study, Software Quality Attributes

## 1 INTRODUCTION

Technical debt (TD) is a metaphor reflecting technical compromises that can yield short-term benefits but may hurt the long-term health of a software system [5]. TD accumulation can negatively impact several types of Software Quality Attributes (SQA) [24], a critical success factor in software projects. Managing its impact includes identifying, monitoring, and measuring TD symptoms [7]. Even though Technical Debt Management (TDM) is a critical activity [5], many organizations do not have established TDM practices.

Given the diversity of everyday practices for software development, managing TD and SQA can be complex since it relies on a decision process based on multiple and heterogeneous data, which are hard to be gathered and synthesized [7][3].In this context, there is a promising opportunity to use Intelligent Techniques to support TDM activities and SQA management. These techniques explore data for knowledge discovery, reasoning, learning, planning, natural language processing, perception, or supporting decision-making [17]. We conjecture that these techniques can use the data produced in software development tasks to improve existing support for various TDM activities. Although intelligent techniques can be used for this purpose, there is a lack of knowledge of how researchers and practitioners can use them to improve TDM-associated activities and their impact on SQA.

This paper is an extension of a study originally presented at the International Conference on Technical Debt 2022 (TechDebt'22) [1]. However, the present study version contributes to the relationship between TDM and SQA. Cutting the results from the SQA perspective was the path to present an extra analysis outside the original scope. The remainder of this paper is structured as follows. Section 2 describes the main concepts required to understand this study, and existing related studies. Section 3 presents the methodology used to conduct the study. Section 4 discusses the results and findings. Section 5 presents the threats to validity whereas Section 6 presents the final remarks and future work.

## 2 BACKGROUND AND RELATED WORK

This section describes the main concepts to understand this study and reports some related work.

### 2.1 Technical Debt

**Definition**. The term was coined by Cunningham [8] when discussing with stakeholders the consequences of releasing poorly written code snippets to accelerate the development process. TD

*ISE '22, October 4, 2022, Proceeding-based Workshop, Brazil*
2022. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *ISE '22: 2nd Brazilian Workshop on Intelligent Software Engineering, October 4, 2022, Virtual, Brazil*, https://doi.org/10.5753/ise.2022.227051.

can have a positive or negative influence on the software quality attributes (SQA), and it may occur due to many factors, such as a lack of team members' knowledge of coding without following principles, patterns, and other best practices. Therefore, software organizations must perceive and manage TD in their projects [20].

**TD Types**. TD can be associated with different software artifacts such as requirements specification, source code, and test reports. For instance, Li *et al.* [16] investigated how TD is spread throughout different development phases and collected a large number of TD types and related instances.They classified TD into ten coarse-grained types: Requirement TD, Design TD, Code TD, Test TD, Defect TD, Architectural TD, Build TD, Infrastructure TD, Versioning TD, and Documentation TD [16][7]. The original study provides a detailed definition of each TD type.

**TDM Acttivities**. The TDM includes activities that prevent potential TD from being incurred [16][7]. TDM activities also comprise those that deal with the accumulated TD to make it visible and controllable. In addition, TDM activities help to keep a balance between the cost and value of the software project. The main TDM activities can be summarized as: TD Identification, TD Measurement, TD Prioritization, TD Prevention, TD Monitoring, TD Repayment, TD Representation/documentation, and TD Communication [16]. The original study provides a detailed definition of each TDM activity.

## 2.2 Intelligent Techniques

**Definition**. The term comprises a set of techniques to explore data for knowledge discovery, reasoning, learning, planning, natural language processing, perception, or supporting decision-making [17]. The guiding principle of Intelligent Techniques is to exploit this tolerance to achieve tractability, robustness, and low computation cost [15].

**Classification**. By using the previous study classification [17], these Techniques can be organized into 11 broad categories: (i) Machine Learning (e.g., regression analysis, K-means, and support vector machine), (ii) Reasoning Under Uncertainty (e.g., Bayesian network and fuzzy logic), (iii) Search and Optimization (e.g., Genetic algorithm and Linear programming), (iv) Natural Language Processing (e.g., Speech recognition and Text mining), (v) Mathematical Model (e.g., Conceptual model and Statistical model), (vi) Multiple Decision Criteria (e.g., Analytic Hierarchy Process - AHP), (vii) Rules (e.g., Decision Tree), (viii) Recommendation Systems (e.g., Recommender systems and Decision-makers), (ix) Multi-agent Systems (e.g., BDI systems and Cognitive Multi-Agent systems), (x) Semantic Network (e.g., Complex networks and graphs), and (xi) Cognitive Simulation (e.g., Data analytic and Comprehension model).

## 2.3 Software Quality Attributes

Software quality is the degree to which software possesses a desired combination of attributes (e.g., reliability, interoperability) [10]. There are relevant models for software quality, such as ISO/IEC 9126-1, the McCall model, and ISO/IEC 25010 [12]. The ISO/IEC 25010 includes all factors of other models, such as McCall and ISO/IEC 9126-1 [19]. Therefore, this model was used in this study. In the TD context, many works emphasize the negative impact on maintainability and reliability.

The ISO 25010 quality model contains eight quality attributes that represent dynamic and static properties of software. The characteristics are suitability, efficiency/performance, compatibility, usability, reliability, security, maintainability, and portability. The maintainability represents the availability of a product or system to be modified to improve it or adapt it to changes [12]. Many studies pointed to TD's impact on maintainability, mainly in commercial studies [13].

## 2.4 Our Contribution

TDM has been largely studied in the literature. For instance, Tom *et al.* [22] conducted a Systematic Mapping Study (SMS) to get a consolidated understanding of TD. Similarly, a systematic review was conducted in Ampatzoglou *et al.* [3], where the goal was to analyze research efforts on TD by considering the financial aspects. Then, Mendonça *et al.* [2] reported a systematic study investigating the TD types and TDM strategies.

Regarding Intelligent Techniques, Sorte *et al.* [21] and Feldt *et al.* [11] conducted studies to better understand the interaction between Software Engineering and Intelligent Techniques. Then, Perkusich *et al.* [17] analyzed the literature on applying Intelligent Techniques to Agile Software Development (ASD). More recently, Tsintzira et al. [23] conducted a literature review by analyzing the intersection of Machine Learning and TDM. Similarly, Azeem et al. [6] presented a literature review on Machine Learning Techniques for Code Smell Detection.

In summary, many are studies interested in TD and related management activities. Similarly, we see the growing interest in studying the application of Intelligent Techniques in software engineering. However, to the best of our knowledge, there is no report in the literature associating Intelligent Techniques in the context of TDM and their relationship with SQA. In this sense, our study will begin the exploration of this gap and point out new perspectives in the field.

# 3 RESEARCH METHODOLOGY

To perform this research, we used a Systematic Mapping Study (SMS) by following the guidelines established in Petersen *et al.* [18]. The SMS approach is organized into three high-level steps: (i) Study Planning; (ii) Conducting the Review; and (iii) Mapping and Documenting Results.

## 3.1 Study Planning

**Research Questions**. The main objective of this study was to identify, classify, and analyze the state of the art in applying Intelligent Techniques to TDM activities and their relationship with SQA. To this end, we have formulated two Research Questions (RQ) to support our research investigation as follows.

**RQ1.** *What is the state of the art on the intersection of TDM and Intelligent Techniques?* This RQ focused on identifying what Intelligent Techniques, TD types, and TDM activities are in the spotlight. For this purpose, we derived two secondary RQs as follows: (i) *RQ1.1* What Intelligent Techniques have been employed to support TDM activities?; and (ii) *RQ1.2* What TD types have been supported by Intelligent Techniques-based solutions?

**RQ2.** *What SQA types the Intelligent Techniques provide support to?*. This question intends to create a catalog associating the use of Intelligent Techniques and what SQA can be impacted by using these techniques. For this purpose, we derived two secondary RQs as follows: (i) *RQ2.1* What is the relationship between SQA and Intelligent Techniques?; and (ii) *RQ2.2* What is the relationship between SQA and TD types?.

**Study protocol**. The study protocol was iteratively prepared, and researchers selected a sub-set of selected papers explored in the pilot study as input to conduct the "study protocol". Moreover, we adapted protocols of published secondary studies in the TD and Intelligent Techniques research areas to be more aligned with the literature [2, 7, 16, 17]. We refined the protocol items for each iteration based on the initial results. We observed the need for new keywords or terms to be included in the search string or even removal of some of them. Inclusion/exclusion criteria were applied during the screening and selection process. Later, in the next step, they were double-checked to make sure the results were consistent.

## 3.2 Conducting the Review

**Selecting Primary Studies**. Search strings were constructed following the population, intervention, comparison, and outcome (PICO) criteria [14]. The population and intervention search terms were joined using the "AND" Boolean operator to build the search string. We selected, evaluated, and extracted initial data from the selected studies with the support of the Start tool [9]. The supplementary material presents more details about the search string [4].

The SMS was limited to primary studies published between 2012 and 2020. Only scientific papers were considered for the screening process and primary studies selection. All duplicated studies were excluded. Next, we identified studies based on whether it contributes to the body of knowledge of Intelligent Techniques and TDM. Figure 1 shows the search process applied to four databases (i.e., ACM, IEEE Xplore, Scopus, and Engineering Village) to gather existing studies and provide support to answer the RQs. We executed tailored search queries in the four databases mentioned above based on the search string and keywords.

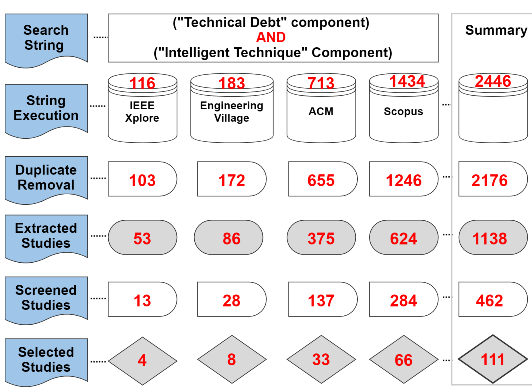


**Figure 1: Literature search process with search string.**

To ensure the inclusion of relevant studies, we defined a set of criteria within the scope of this study. The *inclusion criteria* can be summarized as (i) publications that contribute to the knowledge area of both Intelligent Techniques, TDM, and SQA; (ii) peer-reviewed publications; (iii) scientific papers (including experience reports); and (iv) publications that are written in English. Additionally, we defined the following *exclusion criteria*: (i) topics that are not related to Intelligent Techniques, TDM, or SQA; (ii) studies that merely mention these concepts without an investigation of the topic of Intelligent Techniques in the context of TDM and SQA; (iii) studies published in unrecognized (non-scientific) venues that are different from journal or conference publications; and (iv) duplicates of the same study (in this case, a more complete version of the study was selected).

**Data Extraction/Synthesis**. To provide adequate support to answer our RQs, we first provided an overview of the extracted data, as illustrated in Table 1), row ID's *Pr1-Pr7*. Furthermore, we used other classifications that have been successfully used in other studies [2, 7, 16]. *For Pr8 and Pr9*, we used the classification presented by Perkusich *et al.* [17]. *For Pr10, Pr11, and Pr12*, we adopted a classification widely utilized by other secondary studies in the TD research area such as [2, 7, 16]. Finally, *For PR13* we used the framework categorization described in ISO/IEC 25010.

## 3.3 Data Analysis

**Analysing Primary Studies**. We performed the data extraction to collect detailed information from primary studies. The first author performed the data extraction whereas the second author checked the extracted data. After our initial aggregation, we reviewed the recommendations found in the individual studies. We also looked for general trends that had not been previously discussed but indicated an issue that needed to be addressed in the study protocol. We integrated the results from our synthesis with the recommendations we found in the individual studies. These recommendations were used to specify changes required to the current study protocol.

**Reporting Study Results**. The first step consisted in planning the report and included specifying the audience and determining what sort of document would best suit their needs. The second step was writing the actual report. The ways of presenting the data were discussed among the authors and their support for responses to the various RQs defined for the study. Finally, the last step was validating the reports through internal and external reviews to assess their quality.

# 4 RESULTS AND DISCUSSION

This section presents the main results used to support the Research Questions (RQ1-RQ2) based on the data collected from the primary studies. Before exploring the RQs, we will explore demographic data associated with the 111 primary studies included in the SMS. Figure (2) shows the distribution of selected studies over the period from 2012 to 2020. This figure provides clear information on the trend of the number of published studies on Intelligent Techniques applied in the TD context. The number of published studies on TDM has increased since 2012, representing an increasing interest and relevance of the research topic. Further, 25% of the primary studies from conferences were published on MTD, TechDebt, and ICSE. Moreover, 30% of the primary studies from journals were published on TSE and JSS. Due to space constraints, we provide more result details in the supplementary material [4].

## 4.1 TD and Intelligent Techniques (RQ1)

**Intelligent techniques to support TDM activities (RQ1.1)**. The most popular Intelligent Techniques are *Machine Learning*, *Reasoning Under Uncertainty*, *Natural Language Processing* and *Mathematical Model*, which were applied in 52%, 15%, 13% and 12% of the studies, respectively.

We noticed that *Machine Learning* was employed in most studies. Based on this sample, approximately 55% of the solutions using Machine Learning were classified as *Supervised Learning Techniques*, such as Regression Analysis (35%), Neural Networks (19%), and Support Vector Machine (6%). In turn, around 3% of the solutions using

**Table 1: Overview of extracted data.**

| ID | Property | Format/value | RQ |
|---|---|---|---|
| Pr1 - Pr5 | (i) Publication ID, (ii) Title, (iii) Abstract, (iv) Keywords and (v) Metadata | (i) It is a number, (ii) It is a phrase, (iii) It is a text, (iv) A set of words, and (v) URL, Volume, Pages, DOI, ISSN | - |
| Pr6 - Pr7 | (i) Publication Year and (ii) Source | (i) Number and (ii) Conference/Journal | - |
| Pr8 | Intelligent Techniques category employed to support TDM activities | For instance, Search and optimization, Mathematical model, Multi-agent system, Rules, Recommendation system, Natural language processing, Semantic network, and Cognitive simulation | RQ1 |
| Pr9 | Intelligent Techniques subcategory employed to support TDM activities | For instance, Machine Learning (Regression Analysis, SVM, KNN, Decision Tree) and Reasoning under uncertainty (Statistical model, Bayesian network, weighted dependency graph). | RQ1 |
| Pr10 | TDM Activity supported by Intelligent Techniques | TD repayment,TD identification,TD measurement, TD monitoring, TD prioritization,TD communication, TD prevention,TD representation/documentation, | RQ1 |
| Pr11 | TD types supported by Intelligent Techniques | Requirement TD, Architectural TD, Design TD, Code TD, Test TD, Build TD, Documentation TD, Infrastructure TD, Defect TD | RQ1 |
| Pr12 | TD sub-types supported by Intelligent Techniques | For instance, Requirement TD (over-engineering, Not specified), and Design TD (Code smells, Complex classes or methods, Not specified). | RQ1 |
| Pr13 | SQA | SQA classification is based on ISO/IEC 25010. | RQ2 |

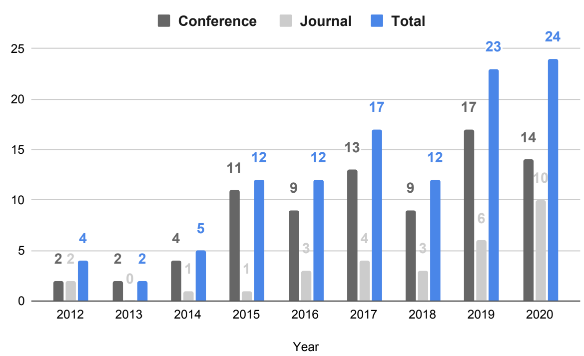


**Figure 2: Distribution of selected studies over time period.**

Machine Learning were classified as *Clustering* techniques, such as K-means algorithms. Finally, about 23% of the studies used *Tree* techniques such as Decision Tree, Random Tree, and Random Forest.

Similarly, the most popular *Reasoning Under Uncertainty* technique was the *Statistical Model*. The corresponding number of studies and the sub-technique employed can be summarized as follows: *Statistical Model* (67%), *Bayesian Network* (27%), and *Weighted Dependency Graph*(13%). We also observed that *Natural Language Processing* was employed to support TDM activities in 12% of the primary studies. Finally, other techniques such as *Semantic Network* had three related studies, *Recommendation system* and *Multi-Agent System* had only one related study, while *Cognitive Simulation* had no related study.

About 97 of the primary studies (87%) reported solutions using *only one* type of Intelligent Technique to support TDM activities. This finding may indicate that applying Intelligent Techniques to support TDM activities can be further explored as a promising research topic. Similarly, we did not find in the literature a significant number of solutions (only 12 primary studies representing about 10% of the total) that are being developed to take advantage of the best features of *two* types of Intelligent Techniques. Interestingly, we have not found solutions that offer a combination of three or more Intelligent Techniques to support TDM.

**TD Types Supported by Intelligent Techniques (RQ1.2).** Based on the selected studies, we found many TD types and instances at different levels of abstraction. We notice that *Design TD* was mentioned in approximately about 57% of the studies. Similarly, *Code TD*, *Architectural TD*, and *Defect TD* are the subsequent ones, occurring, respectively, in 31%, 21%, and 14% of the primary studies. In turn, *Documentation TD*, *Build TD*, and *Infrastructure TD* were the less mentioned TD types. The number of studies associated with these types is less than 10%. Note that the total can exceed 100% since the same study can provide support to more than one TD type.

Comparing these results against other secondary studies, we observed a relevant comparative reference for our research findings. First, Alves *et al.* [2] pointed out Design TD, Architectural TD, and Documentation TD were investigated, respectively, in 20%, 24%, and 13% of the 111 selected studies. Li *et al.* [16] revealed that Code TD, Architectural TD, and Design TD were reported, respectively, in 80%, 65%, and 60% of the selected studies. Their results are aligned with what we observed in his SMS. Regarding the most studied TD types (top-3), these results are also similar to previous studies reported in the literature. However, when analyzing the top-5 most studied TD types, we noticed differences compared to these aforementioned secondary studies. For instance, Test TD and Defect TD received great attention, mainly in the last four years, representing about 14% and 11% of the selected studies considered in this SMS.

An important finding is that 63% of the studies support *only one* TD type, 26% of the studies provide support for the *two* TD types, while the remaining 11% of the studies provide support for *three or more* TD types simultaneously. Therefore, we noticed that the academic community lacks systematic awareness to deal with TD through well-defined and interconnected activities. As previously mentioned, most research was limited to applying Intelligent Techniques to address only one activity in the TDM process. One possible reason may be the complexity of collecting information on certain types of TD and automating decision-making considering this information. The most supported TD types are code TD and Design TD. These TD types are already widely covered and mapped by metrics that can support decision-making more assertively.

## 4.2 Intelligent Techniques and SQA (RQ2)

**Relationship between Intelligent Techniques and SQA (RQ2.1).** This section describes the results associated with Software Quality Attributes (SQA) which can be impacted by the TDM activities supported by Intelligent Techniques described in the selected studies. The SQA classification followed the standard ISO/IEC 25010 which comprises eight main quality characteristics. The frequency in which these attributes are described in the primary studies are: Maintainability (85%), Reliability (44%), Suitability (42%), Security (32%), Performance (20%), Compatibility (9%), Usability (9%), and Portability (9%). According to described in Figure 3, we can notice that Machine Learning, Recommender systems, and Rules were the most used intelligent techniques that impact all the SQA considered in this study. It is essential to mention that these three techniques represent about 45% of the studies related to maintainability and about 22% of the studies related to reliability.

**Relationship between TD types and SQA (RQ2.2)**. We identified that the main TD types considered in the studies were Design TD (56%) and Code TD (34%), both related to source-code and affect software maintainability, which directly or indirectly affects other SQAs (e.g. reliability and security). We observed that TD items affect different characteristics of software quality due to its impact in many software artifacts. As reported by Alves *et al.*[2], TD items impact from requirements and architectural documents to source-code, tests, and code versioning systems. Hence, to avoid the negative impact, we need various strategies to find several TD types and support the whole TDM process.

**Joint Analysis**. Figure 3 provides a joint analysis about TD types, SQA, and Intelligent Techniques. We notice that Maintainability, Security, Reliability, and Suitability are impacted by almost all TD types, but being the most recurrent Design TD (52%), Code TD (about 30%), and Architectural TD (about 20%). It is important to mention that Code and Design TD have a significant impact on Maintainability. Since these TD types directly impact code artifacts, this SQA is more impacted as well. The relationship between TD types and the impact on SQA may be important for industry practitioners. For example, if a specific type of SQA is required in a given software project, the TD types that impact that specific type of SQA must be appropriately prioritized and managed.

Similarly, we can notice Machine Learning, Rules, Recommender Systems were used to assist all SQA types. For instance, Machine Learning was used to support TDM activities, impacting directly Maintainability (about 30%) and Reliability (21%), whereas Rules and Recommendation Systems were related to Reliability (15%) and Security (about 12%). Analyzing the data from another aspect, we can see that Maintainability, Reliability, and Security received support by almost types of Intelligent Techniques. In other words, Maintainability received support by 12 different Intelligent Techniques. Reliability and Security received support by 11 and 10 different Intelligent Techniques. This information might be useful in the sense of choosing proper Intelligent Techniques to raise a certain SQA.

A more in-depth analysis of the effectiveness of these intelligent techniques in the TDM may be interesting to provide notes on which techniques fit with certain TD types and, consequently, contribute to certain SQA. We did not intend to carry out this analysis in the present study. For this reason, although some intelligent techniques are less used for certain types of TD, no conclusion can be made about the effectiveness (or not) of using this technique. For research implications, we believe that the practical demonstration of the relationship between intelligent techniques, TD, and SQA can draw the attention of the scientific community to provide new research related to the junction of these three topics.

## 5 THREATS TO VALIDITY

This section discusses validity following the classification schema proposed by Wohlin *et al.* [26].

*Construct Validity.* We aimed to retrieve as many relevant studies as possible to avoid any possible literature selection bias. We faced a challenge in determining the scope of our study as the notion of "TD" and "Intelligent Techniques" means different things to different research communities. Therefore, to cover them all and avoid bias, we searched the literature based on relevant terms and combined them in our search string. While this strategy significantly increased the search effort [25], it enabled us to find a comprehensive set of relevant studies.

*Internal Validity*: To mitigate the bias in study selection, a pilot selection was performed to ensure that the researchers reached a consensus on understanding the selection criteria. Also, the study protocol was discussed among the researchers to ensure a common understanding of study selection. Moreover, two researchers conducted the process independently in the second and final rounds of study selection. Afterward, they compared and discussed their selection results to mitigate potential personal bias. It is important to mention that whenever there was a disagreement between two reviewers regarding the criteria for selecting/extracting the studies, a third reviewer was triggered to resolve such disagreement.

*External Validity.* The results can not be generalized. Although the diversity of settings, Intelligent Techniques have not been widely applied to support TDM activities. *Reliability*: The threat to the data synthesis and reporting reliability has been mitigated based on discussion and peer review of the data extracted by the researchers, a structured template for data synthesis, and several steps in the scheme and process that were refined and evaluated.

## 6 FINAL REMARKS

This paper presented the results of a Systematic Mapping Study aiming to identify and analyze the studies that report Intelligent Techniques applied for supporting TDM and their relationship with SQA. Our study yielded a set of 111 primary studies considering the time-frame between 2010 and 2020. From their analysis, we shed light on an initial discussion regarding the intersection between these research areas.

**(RQ1)**. The most frequently applied Intelligent Techniques were Machine Learning, Reason Under Uncertainty, and Search and Optimization, corresponding to 60% of the primary studies. Similarly, we observed the main TD types studied were Design TD (57%), Code TD (31%), and Architectural TD (20%). Most studies provided support for only one TD type (around 70%). **(RQ2)**. We observed that TD affects different characteristics of SQA, mainly Maintainability, Reliability, and Security. Machine Learning, Rules, and Recommendation systems were the most prominent Intelligent Techniques to monitor or improve the SQA. The presence of Design TD, Code TD, and Architectural TD impact all SQA proposed by ISO/IEC 25010.

In future studies, we intend to assess more in-depth related aspects of Intelligent Techniques in the context of TDM and their relationship with SQA. For example, we can evaluate the maturity of the research based on some characteristics such as empirical research types, types of research contributions, and research validation. In addition, we plan to analyze some aspects (e.g., level of automation, explanation level, and point of application) that aim to assess the risks involved in applying Intelligent Techniques to support TDM activities.

## REFERENCES

[1] D. Albuquerque, E. Guimaraes, G. Tonin, M. Perkusich, H. Almeida, and A. Perkusich. Comprehending the use of intelligent techniques to support technical debt

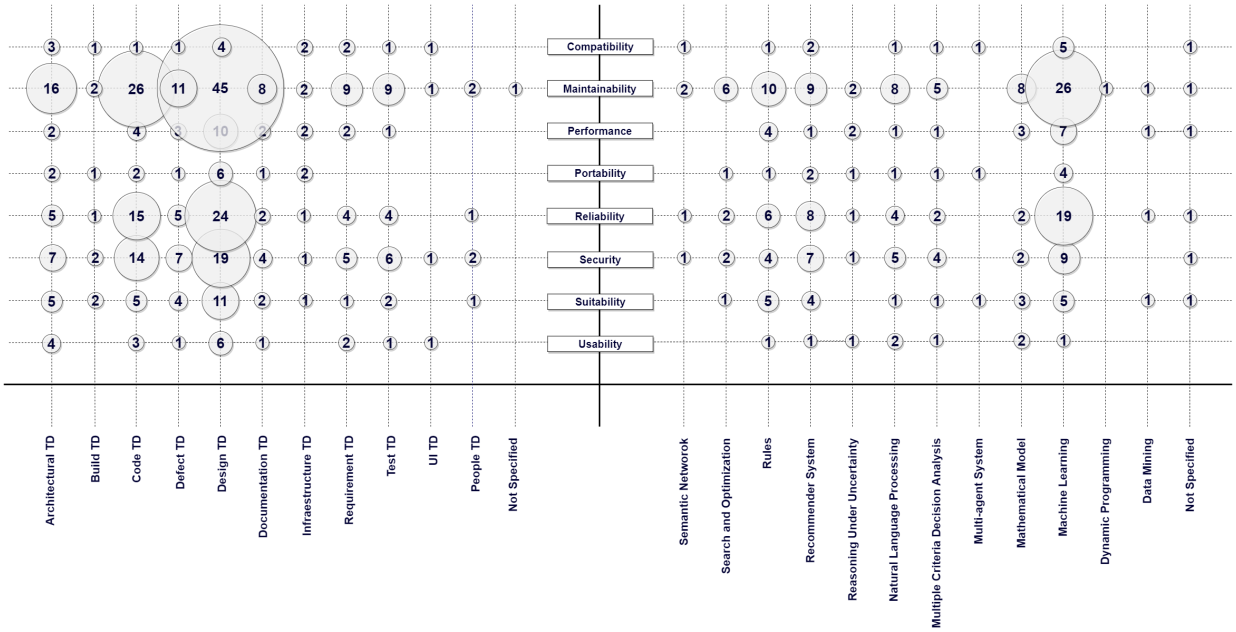


**Figure 3: Relationship between Software Quality Attributes, Intelligent Techniques and TD types.**


management. In *International Conference on Technical Debt (TechDebt'22)*, 2022.

[2] N. S. Alves, T. S. Mendes, M. G. de Mendonça, R. O. Spínola, F. Shull, and C. Seaman. Identification and management of technical debt: A systematic mapping study. *Information and Software Technology*, 70:100 – 121, 2016.

[3] A. Ampatzoglou, A. Ampatzoglou, A. Chatzigeorgiou, and P. Avgeriou. The financial aspect of managing technical debt: A systematic literature review. *Information and Software Technology*, 64:52–73, 2015.

[4] Anonymous. Supplementary Material - SMS on IF for TDM. Dataset. Online available. https://doi.org/10.6084/m9.figshare.12367277.v3. 5 2020.

[5] P. Avgeriou, P. Kruchten, I. Ozkaya, and C. Seaman. Managing technical debt in software engineering (dagstuhl seminar 16162). In *Dagstuhl Reports*, volume 6. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[6] M. I. Azeem, F. Palomba, L. Shi, and Q. Wang. Machine learning techniques for code smell detection: A systematic literature review and meta-analysis. *Information and Software Technology*, 108:115–138, 2019.

[7] W. N. Behutiye, P. Rodríguez, M. Oivo, and A. Tosun. Analyzing the concept of technical debt in the context of agile software development: A systematic literature review. *Information and Software Technology*, 82:139–158, 2017.

[8] W. Cunningham. The wycash portfolio management system. *ACM SIGPLAN OOPS Messenger*, 4(2):29–30, 1992.

[9] S. Fabbri, C. Silva, E. Hernandes, F. Octaviano, A. Di Thommazo, and A. Belgamo. Improvements in the start tool to better support the systematic review process. In *In Proc. of the 20th Int'l Conf. on Evaluation and Assessment in Soft. Eng.*, pages 1–5, 2016.

[10] M. Falco and G. Robiolo. Building a catalogue of iso/iec 25010 quality measures applied in an industrial context. In *Journal of Physics: Conference Series*, volume 1828, page 012077. IOP Publishing, 2021.

[11] R. Feldt, F. G. de Oliveira Neto, and R. Torkar. Ways of applying artificial intelligence in software engineering. In *2018 IEEE/ACM 6th Int'l Workshop on Realizing Artificial Intelligence Synergies in Soft. Eng (RAISE)*, pages 35–41. IEEE, 2018.

[12] I. O. for Standardization, S. Technical Committee ISO/IEC JTC 1, Information technology. Subcommittee SC 7, and systems engineering. *Systems and Software Engineering: Systems and Software Quality Requirements and Evaluation (SQuaRE): System and Software Quality Models*. ISO, 2011.

[13] A. Kaur. A systematic literature review on empirical analysis of the relationship between code smells and software quality attributes. *Archives of Computational Methods in Engineering*, 27(4):1267–1296, 2020.

[14] B. Kitchenham and P. Brereton. A systematic review of systematic review process research in software engineering. *Information and Software Technology*, 55(12):2049–2075, 2013.

[15] P. Kumar and S. Singh. An emerging approach to intelligent techniques—soft computing and its application. In *Internet of Things and Big Data Applications*, pages 171–182. Springer, 2020.

[16] Z. Li, P. Avgeriou, and P. Liang. A systematic mapping study on technical debt and its management. *Journal of Systems and Software*, 101:193–220, 2015.

[17] 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. *Information and Software Technology*, 119:106241, 2020.

[18] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson. Systematic mapping studies in software engineering. In *12th Int'l Conf. on Evaluation and Assessment in Soft. Eng. (EASE) 12*, pages 1–10, 2008.

[19] A. A. Pratama and A. B. Mutiara. Software quality analysis for halodoc application using iso 25010: 2011. *International Journal of Advanced Computer Science and Applications*, 12(8), 2021.

[20] V. M. Silva, H. J. Junior, and G. H. Travassos. A taste of the software industry perception of technical debt and its management in brazil. *Journal of Software Engineering Research and Development*, 7:1–1, 2019.

[21] B. W. Sorte, P. P. Joshi, and V. Jagtap. Use of artificial intelligence in software development life cycle: a state of the art review. *Int'l Journal of Advanced Eng. and Global Technology*, pages 398–403, 2015.

[22] E. Tom, A. Aurum, and R. T. Vidgen. A consolidated understanding of technical debt. In *20th European Conference on Information Systems, ECIS 2012, Barcelona, Spain, June 10-13, 2012*, page 16, 2012.

[23] A.-A. Tsintzira, E.-M. Arvanitou, A. Ampatzoglou, and A. Chatzigeorgiou. Applying machine learning in technical debt management: Future opportunities and challenges. In *International Conference on the Quality of Information and Communications Technology*, pages 53–67. Springer, 2020.

[24] K. Wiegers and J. Beatty. *Software requirements*. Pearson Education, 2013.

[25] C. Wohlin. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *In Proc of the 18th Int'l Conf. on evaluation and assessment in Soft. Eng*, pages 1–10, 2014.

[26] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012.