Elmidaoui S, Cheikhi L, Idri A *et al.* Machine learning techniques for software maintainability prediction: Accuracy analysis. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 35(5): 1147–1174 Sept. 2020. DOI 10.1007/s11390-020-9668-1

Machine Learning Techniques for Software Maintainability Prediction: Accuracy Analysis

Sara Elmidaoui[1], Laila Cheikhi[1,*], Ali Idri[1], and Alain Abran[2]

[1] *Software Project Management Team, École Nationale Supérieure d'Informatique et d'Analyse des Systémes, Madinate Al Irfane, Mohammed V University in Rabat, Agdal Rabat, BP 713, Morocco*

[2] *Department of Software Engineering & Information Technology, Ecole de Technologie Supérieure (ETS), Montréal H3C IK3, Canada*

E-mail: sarah.elmidaoui@gmail.com; {laila.cheikhi, ali.idri}@um5.ac.ma; alain.abran@etsmtl.ca

Received April 24, 2019; revised February 26, 2020.

Abstract Maintaining software once implemented on the end-user side is laborious and, over its lifetime, is most often considerably more expensive than the initial software development. The prediction of software maintainability has emerged as an important research topic to address industry expectations for reducing costs, in particular, maintenance costs. Researchers and practitioners have been working on proposing and identifying a variety of techniques ranging from statistical to machine learning (ML) for better prediction of software maintainability. This review has been carried out to analyze the empirical evidence on the accuracy of software product maintainability prediction (SPMP) using ML techniques. This paper analyzes and discusses the findings of 77 selected studies published from 2000 to 2018 according to the following criteria: maintainability prediction techniques, validation methods, accuracy criteria, overall accuracy of ML techniques, and the techniques offering the best performance. The review process followed the well-known systematic review process. The results show that ML techniques are frequently used in predicting maintainability. In particular, artificial neural network (ANN), support vector machine/regression (SVM/R), regression & decision trees (DT), and fuzzy & neuro fuzzy (FNF) techniques are more accurate in terms of PRED and MMRE. The *N*-fold and leave-one-out cross-validation methods, and the MMRE and PRED accuracy criteria are frequently used in empirical studies. In general, ML techniques outperformed non-machine learning techniques, e.g., regression analysis (RA) techniques, while FNF outperformed SVM/R, DT, and ANN in most experiments. However, while many techniques were reported superior, no specific one can be identified as the best.

Keywords accuracy criterion, accuracy value, machine learning technique, maintainability prediction

1 Introduction

Software maintenance activities begin, in general, after the first release is delivered to the end-user. The aim is to keep the software operational while it undergoes the changes that occur during its lifetime. Maintenance activities consume the major part of software lifecycle costs [1]. Research initiatives in software engineering have sought to reduce this cost by providing software that is easily modifiable. Maintainability, defined as the ease with which software can be modified[①], is one of the most important quality characteristics that software engineering should address[②]. Therefore, as stated by SWEBOK, specifying, reviewing, and controlling maintainability during software development is important in order to reduce this cost and improve maintainability [2].

However, maintainability is often overlooked during development, and mostly considered only at the later phase of the software lifecycle; this is not useful for controlling the quality of the software. Predicting software maintainability in an earlier phase, such

Survey

*Corresponding Author

[①] IEEE Std. 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology, 1990.

[②] ISO. Systems and Software engineering — Systems and Software Quality Requirements and Evaluation — System and Software Quality Models. ISO/IEC 25010, 2010.

as the design phase, has been the focus of attention for researchers and practitioners in an effort to improve software quality and reduce future maintenance costs. In 2009, of many software product maintainability prediction (SPMP) techniques proposed in the literature, none was proved to be superior in all cases[3–5]. Few studies were supported with accuracy criteria and the choice among SPMP techniques was not obvious. There is a general agreement on the need to design SPMP techniques that are more reliable and robust. This has motivated researchers to investigate new techniques for more accurate prediction of software maintainability.

A number of reviews have been conducted on this topic[3–13]. A secondary study was conducted in [14] and found that most of these reviews provide only a general overview or a roadmap of the studies without conducting an in-depth investigation of the aspects studied. Some of these reviews extracted the data for each primary study, such as [3, 4, 7], while others provided only a summary without a detailed analysis. Moreover, each review dealt with some (not all) but different aspects related to SPMP, such as measures, models/techniques, factors or attributes, maintainability definition, and accuracy criteria used. Based on the results of these reviews, maintainability prediction models/techniques, as well as maintainability key predictor measures/factors were the most studied aspects. Identified techniques fell into two categories: statistical learning and machine learning (ML). Statistical techniques, especially regression analysis[3–5], have been frequently used in empirical studies. The shift towards using ML techniques is gaining increasing attention as they offer algorithms that have the ability to enhance performance automatically through experience[15], and they facilitate the expression of the relationships between the maintainability of the software and its attributes when this relationship is not linear and does not seem to have any predetermined form[16]. In the last decade, ML techniques were used to test their ability to predict maintainability compared with statistical techniques. This motivated us to review ML techniques used for SPMP in order to understand how much has been achieved in this area and to identify related guidelines based on our findings.

Table 1 summarizes related work in terms of purpose, the type of reviews (systematic literature review (SLR), systematic mapping study (SMS), survey, review, roadmap, etc.), the period of collection of studies included in the reviews, the number of studies and the research questions (RQs) addressed. As can be seen in Table 1, all studies share the same topic, the maintainability of the software, but with different purposes. The period of collection varies among the studies as well as the number of primary studies. Only four studies have conducted a rigorous review process (three SLRs and one SMS) and addressed some research questions. The purpose of the two SLRs in [3, 4] is the same as in our work, but they differ in terms of the research questions addressed, to the exception of RQ2 that is in common with this study; however the response provided in [3, 4] is limited to a table that groups the data about techniques, accuracy measures and their corresponding values, cross-validation, and techniques reported to be superior, without providing any analysis. Moreover, these SLRs do not report the overall accuracy of ML techniques, the accuracy comparison of ML versus ML techniques, and the accuracy comparison of ML with statistical techniques.

To sum up, the key different objectives between prior reviews on maintainability prediction and our work are as follows.

1) We provide a classification of SPMP studies with respect to:

a) techniques (ML and statistical techniques) used, identifying the most commonly used ones, and providing trends in SPMP techniques,

b) frequently used cross validation methods, and

c) frequently used accuracy criteria.

2) Our work analyzes evidence regarding:

a) the overall prediction accuracy of SPMP ML techniques in terms of commonly-used accuracy criteria and their corresponding values, and

b) the prediction accuracy of SPMP ML techniques compared with that of statistical techniques and prediction accuracy comparison among different ML techniques to identify the set of techniques reported to be high-performance.

3) This study is the most comprehensive, up to date, covering 77 empirical studies published between 2000 and 2018.

To understand and facilitate the use of ML techniques in SPMP, this study analyzes and discusses the following research questions (RQs).

1) What are the most frequently-used SPMP techniques? (RQ1)

2) Which validation methods have been used for SPMP techniques? (RQ2)

3) What accuracy criteria have been used for SPMP techniques? (RQ3)

4) What is the overall accuracy of SPMP ML techniques? (RQ4)

Table 1. Summary of Related Work

Reference	Purpose	Type of Review	Period of Collection	Number of Studies	Research Questions (RQ) Addressed
[3]	To present a systematic review of software maintainability prediction and metrics	SLR	1985–2008	15	RQ1: evidence for maintainability techniques, RQ2: techniques, RQ2(a): accuracy measures, RQ2(b): numeric values, RQ2(c): cross validation, RQ2(d): technique reported superior, RQ3: factors and metrics, RQ3(a): stage, RQ3(b): type of predictors, and RQ(4): how maintainability is understood
[4]	To present a systematic review of maintainability prediction of relational database relational database driven applications	SLR	1985–2010	7	RQ1: evidence for maintainability techniques for RDBAs, RQ2: techniques for RDBAs, RQ2(a): accuracy measures and numeric values for RDBAs, RQ2(b): cross validation, RQ2(c): technique reported superior, RQ3: factors and metrics for RDBAs, RQ3(a): successful predictors, RQ3(b): stage, RQ3(c): type of predictors, and RQ4: how maintainability is understood in the context of RDBAs
[5]	To provide a survey of object-oriented software maintainability measurement in the past decade	Survey	2003–2012	36	No research questions provided
[7]	To provide a systematic review of coupling metrics for aspect-oriented programming in maintainability studies	SLR	Not provided	12	RQ1: external attributes used to indicate maintainability in aspect oriented (AO) programming, RQ2: coupling metrics for maintainability, RQ3: AO abstractions and mechanisms covered in the design of the used coupling metrics, and RQ4: do AO coupling metrics meet well-established theoretical validation criteria
[8]	To present a systematic mapping study on aspect-oriented software maintenance metrics	SMS	1992–2011	138	RQ1: metrics adopted to assess software maintainability on aspect-oriented (AO) programming, RQ2: metrics adopted to assess software maintainability on object-oriented (OO) programming, and RQ3: metrics that address OO maintainability and can be adapted to address AO maintainability
[9]	To present a survey of key factors affecting software maintainability	Survey	Not provided	Not provided	No research questions provided
[10]	To present a review of maintainability techniques for software development approaches	Survey	Not provided	Not provided	No research questions provided
[11]	To provide a roadmap of software system maintainability models	Roadmap	1970–2012	33	No research questions provided
[12]	To provide a review on appraisal techniques for web-based maintainability	Survey	Not provided	13	No research questions provided
[6]	To provide a review analysis of maintainability models for an object oriented system	Survey	1993–2011	23	No research questions provided
[13]	To present a review of maintainability quantification of object-oriented design: a revisit	Review	Not provided	Not provided	No research questions provided

5) Are there SPMP techniques reported to be superior in the literature? (RQ5)

The rest of the paper is structured as follows. Section 2 presents the methodology pursued throughout this study. Section 3 summarizes and discusses the findings of this study, including an overview of the selected studies and responses to RQ1–RQ5. Section 4 discusses threats to validity of the study. Section 5 summarizes the principal findings with implications for research and practice.

2 Methodology

This study reviews the use of ML techniques for predicting software product maintainability based on the review process suggested by Kitchenham and Charters[17] for conducting SLR. Fig.1 shows the re-

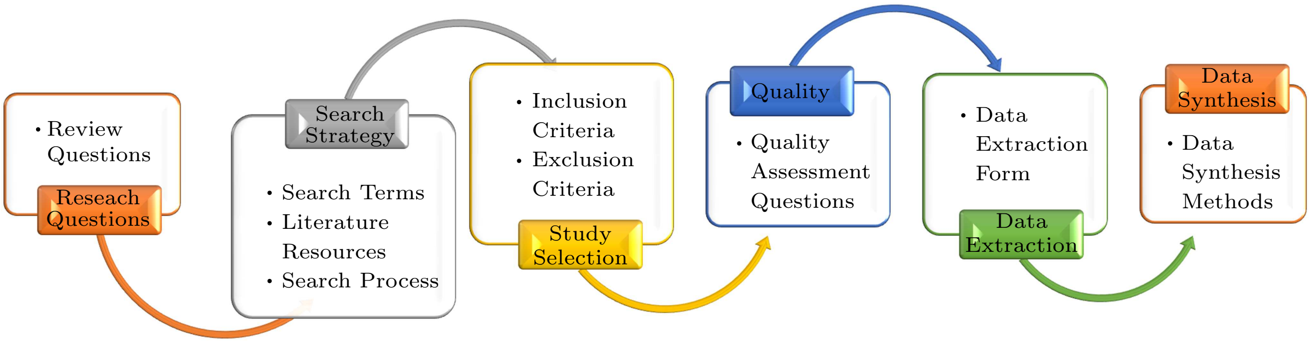


Fig.1. Systematic review process.

view process comprising the following six parts: 1) research questions, 2) search strategy, 3) study selection, 4) quality assessment, 5) data extraction, and 6) data synthesis. The detailed description of these steps will be explained in the following subsections.

2.1 Research Questions

To understand and facilitate the use of ML techniques in SPMP, this study analyzes and discusses five research questions (RQs) which are listed in Table 2, along with our main motivation for including them in the systematic review.

2.2 Search Strategy

The strategy used for searching primary studies to answer the research questions includes three steps. The first step is to define a search string. This search string was then applied to a set of selected electronic databases to extract all the relevant studies in the second step. The search process was divided into two stages to ensure that no relevant study had been left out in the third step. These three steps are described in detail below.

2.2.1 *Search Terms*

The search terms were derived using the following sub-steps[16]:

• identifying the main terms from the review questions listed above;

• identifying all alternative spellings and synonyms for main term;

• using the Boolean operator OR to join synonymous terms, in order to retrieve any record containing either (or all) of the terms;

• use the Boolean operator AND to concatenate the main terms, in order to retrieve any record containing all the terms.

The search string is formulated using the major terms and their corresponding alternative terms as follows.

• Set1 includes major term {maintainability}, and

Table 2. Research Questions and Motivations

ID	Research Question	Motivation
RQ1	What are the most frequently used SPMP techniques?	To identify: - most frequently used SPMP techniques (ML and statistical) - trends in SPMP techniques
RQ2	What are the most used validation methods for SPMP techniques?	To identify validation methods used (LOOCV, K-FCV, etc.)
RQ3	What are the most frequently used accuracy criteria for SPMP techniques?	To identify accuracy criteria used to evaluate the techniques (MMRE, MaxMRE, Pred, etc.)
RQ4	What is the overall accuracy of SPMP ML techniques?	To analyze: - accuracy context of SPMP ML techniques (accuracy values of Pred and MMRE criteria in general and for historical datasets) - accuracy of SPMP ML techniques - accuracy of SPMP ML techniques for UIMS and QUES historical datasets
RQ5	Are there SPMP techniques reported to be superior in the literature?	To provide: - overview of SPMP comparative studies (ML and statistical) - accuracy comparison among different ML techniques - accuracy comparison of ML with statistical techniques

the alternative terms {analyzability OR modifiability OR testability OR stability OR compliance}. These terms are included since they were used in previous SLRs[3, 4] and are considered as sub characteristics of maintainability in ISO 9126 Standards on software product quality.

• Set2 includes major term {empirical*} and alternative terms {evaluation* OR validation* OR experiment* OR control experiment OR case study OR survey}. We focus on empirical studies to discover how rigorously they performed and provided empirical validation of their findings, rather than concentrate on the technique used.

• Set3 includes major term {software product} and alternative terms {software OR application OR system OR software engineering) AND (predict* OR evaluat* OR assess* OR estimat* OR measur*}.

• Set4 includes {method* OR technique* OR model* OR tool* OR approach*}.

2.2.2 *Literature Resources*

To answer our research questions, nine electronic databases were chosen to perform an automated search using the search terms. The databases used were chosen because they were used by previous SLRs in software engineering[3, 4, 7, 16].

The following electronic databases were used for searching the primary studies: IEEE Xplore[3], Science Direct[4], Springer Link[5], Ebsco[6], ACM Digital Library[7], Google Scholar[8], Scopus[9], Jstore[10], DBLP[11].

To obtain up-to-date results, the searches were limited to primary studies published between 2000 and 2018. They were conducted separately in all databases based on title, abstract, and keywords except in Google scholar where the search was restricted to study titles.

2.3 Search Process

To identify the primary studies and to ensure the quality of the search, a two-stage search process was adopted.

• *Initial Search Process.* In the initial process, the candidate primary studies were identified by searching the nine electronic databases using the search string constructed by combining Set1 to Set4 using "AND". The retrieved studies were grouped together to form a set of candidate studies.

• *Second Search Process.* In this second process, the reference list of candidate studies that met the inclusion and exclusion criteria (defined in Subsection 2.3) was scanned based on the title to identify additional primary studies related to SPMP. This stage ensured that the search covered the maximum number of studies related to SPMP.

During the search process, we noticed that the search string is too long to be used in some electronic databases. To deal with this electronic databases limitation, we have tailored the search string depending on the electronic database used by splitting the whole search string, performing the search and then combining the results manually.

2.4 Study Selection

In this step, the studies that addressed the research questions were identified based on their titles, keywords, and abstracts first, and then the studies were read by two authors. The studies that met all inclusion criteria and none of the exclusion criteria were included. However, studies that met at least one of the exclusion criteria were excluded.

Inclusion criteria are:

• empirical studies addressing prediction or assessment of software product maintainability and/or its sub characteristics, and

• empirical studies using SPMP techniques.

Exclusion criteria are:

• studies that discuss the process of software maintenance,

• studies that concentrate on software maintainability generally and do not present a technique to predict the software maintainability,

• studies published before 2000,

[3] https://ieeexplore.ieee.org, Apr. 2020.

[4] https://www.sciencedirect.com, Apr. 2020.

[5] https://link.springer.com, Apr. 2020.

[6] https://www.ebsco.com/, Apr. 2020.

[7] https://dl.acm.org, Apr. 2020.

[8] https://scholar.google.ca/, Apr. 2020.

[9] https://www.scopus.com/, Apr. 2020.

[10] https://www.jstor.org, Apr. 2020.

[11] https://dblp.org/, Apr. 2020.

- short studies (2–3 pages),
- secondary studies, and
- studies by the same author: if results were the same in these studies, we used the most recent; otherwise we used all of these studies.

The decision to include or exclude a study was done by two researchers: if both evaluated a study as "Include" or "Exclude", the study was considered to be relevant or not, respectively. In all the other situations, the study was labeled as "Uncertain", which means that there was a disagreement among the researchers on its relevance. The results show a high level of agreement between the two researchers since only eight cases of disagreement were identified. To deal with disagreement cases, a discussion was held based on the full text until the two researchers reached an agreement. Of the eight "Uncertain" studies, six were retained and two were excluded. The application of the selection criteria to the candidate articles in the initial search stage resulted in 75 relevant studies. The scanning of the reference lists of these studies revealed seven more relevant studies.

2.5 Study Quality Assessment

In the previous step, the selection of studies was determined by their relevance to the research questions (RQs). Quality assessment (QA) criteria were then determined to assess the strength of individual studies, minimize bias, maximize internal and external validity, and guide the interpretation of findings [17]. The QA criteria were selected by considering suggestions given in [16–18], rephrased according to our SLR needs (which are presented in inclusion and exclusion criteria) and formed into a checklist (see Table 3).

The questions were ranked "Yes", "No", or "Partially", with associated scores of 1, 0, and 0.5, respectively. The maximum score for all questions is 8 and the minimum 0. Studies that scored greater than 50% of the maximum score were considered for the review as in [16] by Idri *et al.* and [3] by Riaz *et al.* The quality assessment was carried out by two researchers independently. All disagreements were discussed until a final consensus was reached. A set of 77 primary studies was selected and five with a quality score less than 50% (i.e., less than 4) were rejected. The summary of quality scores for the 77 selected primary studies is presented in Table 4 (see Table A1 in the Appendix for the detailed quality scores of each study).

Table 3. Quality Assessment Criteria Checklist

ID	Questions
QA1	Are the objectives of the study clearly described and appropriate?
QA2	Are the SPMP techniques well-presented and defined?
QA3	Is there more than one SPMP technique proposed and/or evaluated?
QA4	Are the accuracy criteria well-presented and discussed?
QA5	Are the validation methods well-presented?
QA6	Is the most accurate technique clearly stated?
QA7	Is there any comparative analysis conducted?
QA8	Are the findings of the study clearly stated and presented?

Table 4. Statistics of Quality Scores of Selected Studies

Quality Level	Number of Studies	Percent (%)
Very high ($6 \leqslant$ score $\leqslant 8$)	43	56
High ($5 \leqslant$ score < 6)	8	10
Medium ($4 \leqslant$ score < 5)	26	34
Total	**77**	**100**

2.6 Data Extraction and Data Synthesis

A data extraction form was established (see Fig.2) and filled with the information of each selected primary study for addressing the research questions. Two independent researchers performed the extraction by reading the full text. For any disagreement, a discussion was held until a consensus was reached between the authors. The extracted data was grouped into an excel file for data synthesis.

The purpose of data synthesis is to use visualization techniques (e.g., charts, frequency tables) to accumulate and combine facts from the selected primary studies in order to formulate an answer to a research question. Narrative summary including the collection of a number of studies that state similar and comparable viewpoints is used to report the principal findings of the study.

3 Results and Discussion

Fig.3 shows the number of studies obtained at each stage of the selection process. The search in the nine electronic databases resulted in 341 candidate studies. Our inclusion and exclusion criteria were applied and a set of 75 relevant studies were retained. The scanning of the reference lists of the relevant studies revealed seven additional studies. By performing quality assessment criteria to these 82 relevant studies, a set of 77

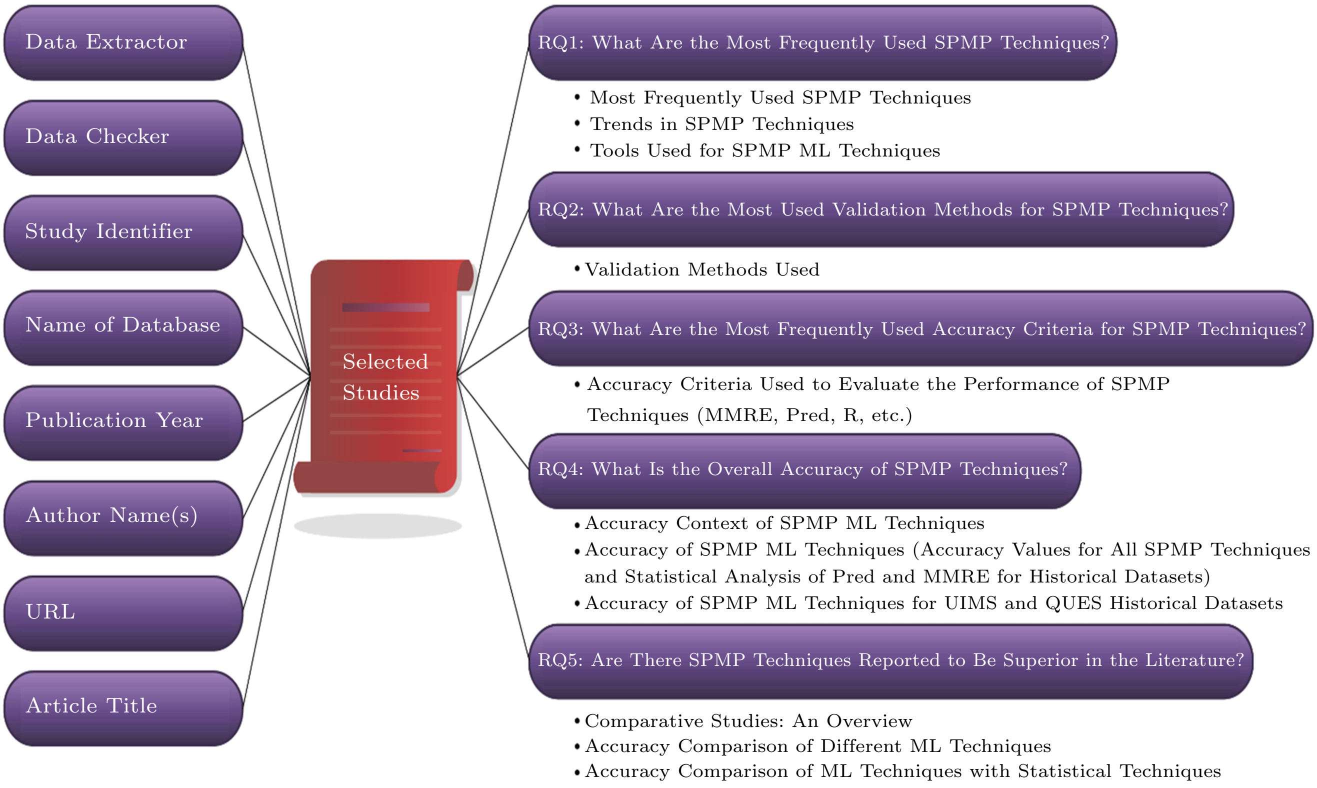

Data Extractor
Data Checker
Study Identifier
Name of Database
Publication Year
Author Name(s)
URL
Article Title
Selected Studies
RQ1: What Are the Most Frequently Used SPMP Techniques?
• Most Frequently Used SPMP Techniques
• Trends in SPMP Techniques
• Tools Used for SPMP ML Techniques
RQ2: What Are the Most Used Validation Methods for SPMP Techniques?
• Validation Methods Used
RQ3: What Are the Most Frequently Used Accuracy Criteria for SPMP Techniques?
• Accuracy Criteria Used to Evaluate the Performance of SPMP Techniques (MMRE, Pred, R, etc.)
RQ4: What Is the Overall Accuracy of SPMP Techniques?
• Accuracy Context of SPMP ML Techniques
• Accuracy of SPMP ML Techniques (Accuracy Values for All SPMP Techniques and Statistical Analysis of Pred and MMRE for Historical Datasets)
• Accuracy of SPMP ML Techniques for UIMS and QUES Historical Datasets
RQ5: Are There SPMP Techniques Reported to Be Superior in the Literature?
• Comparative Studies: An Overview
• Accuracy Comparison of Different ML Techniques
• Accuracy Comparison of ML Techniques with Statistical Techniques

Fig.2. Data extraction form.

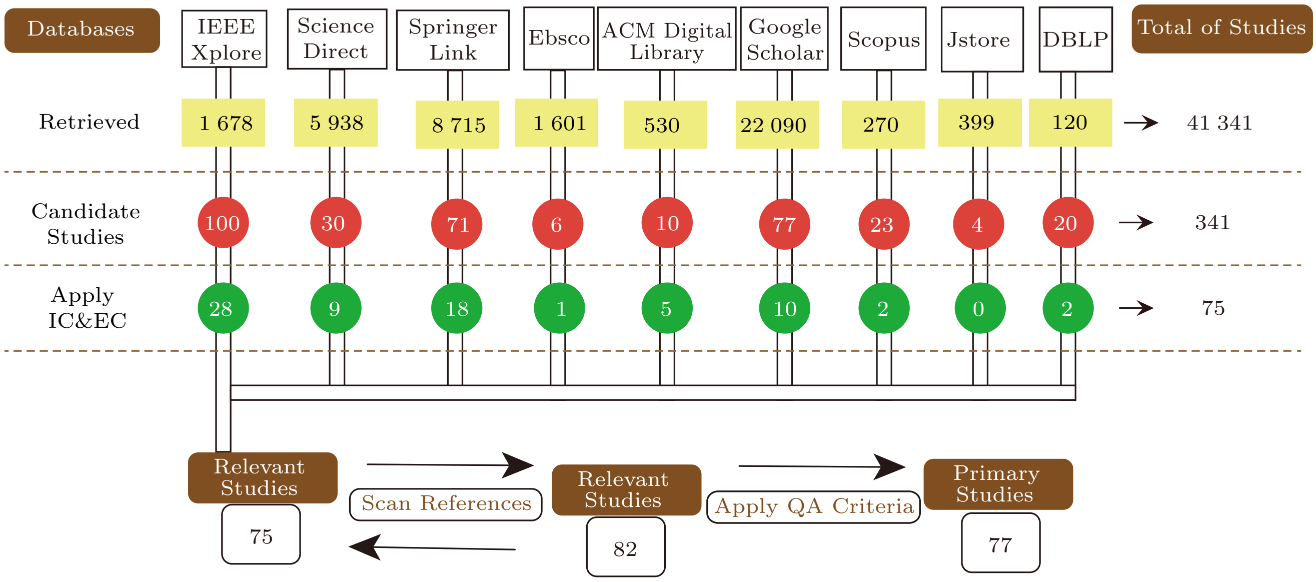


Fig.3. Study selection process.

primary studies, from journals and conferences, of acceptable quality were selected (see Fig.3). These are listed in chronological order in Table A1 in Appendix. This section presents and discusses the results of the review with regard to SPMP techniques (RQ1), validation methods (RQ2), accuracy criteria (RQ3), overall accuracy of SPMP ML techniques (RQ4) and the comparison of SPMP technique accuracy (RQ5). Table A2 in Appendix maps the acronyms of techniques to their extension terms, and Tables A3–A7 in Appendix summarize these points for each study.

3.1 Overview of Selected Studies

This subsection provides an overview of the 77 selected primary studies with regard to publication sources, empirical research approaches, and empirical context, including frequently-used datasets, frequently-used measures or factors (independent variables) and frequently-used measures of maintainability (dependent variable).

Recognized and stable publication sources were used by considering the computer science conference rankings (CORE)[12], and the Journal Citation Reports (JCR) lists. Shortlisted good journals include the Journal of Systems and Software, Software Quality Journal, and International Journal of Innovative Computing, Information, and Control. Examples of reputed international conferences in software maintainability include SIGSOFT Software Engineering Notes, International Software Metrics Symposium, and the International Conference on Software Maintenance. The authors believe that selected studies from such journals and conferences were most appropriate for this review due to their specialization and relevance to the subject matter.

Analysis of the content of the selected studies clearly shows that empirical research approaches to software product maintainability have evolved over the years. From 2000 to 2010, researchers focused mainly on proposing SPMP techniques, such as in [19–21]. From 2010 onwards, empirical studies, such as [22, 23], have evaluated and/or compared SPMP techniques. From 2012 to 2018 interest was mainly directed towards the evaluation of SPMP techniques based on previously completed software projects (called history-based evaluation), as in [24, 25]. Since many techniques were proposed and evaluated in that time frame, researchers and practitioners conducted comparative studies, such as [26–28], in order to identify the best techniques for predicting maintainability (Subsection 3.6).

To empirically evaluate SPMP techniques, many datasets were used, ranging from public datasets provided by researchers, freely available ones from open source systems/software, or private ones from large industrial projects. Many of these datasets were made available to the software engineering community for new research studies (i.e., referred to as historical datasets). In addition, some studies focused on the feature selection[29, 30] in order to select the relevant features and provide the best configuration (e.g., [31] and [32]), or on extraction selection (e.g., [27] and [33]).

Furthermore, it was also observed that Chidamber and Kemerer (48 studies), Li and Henry (32 studies), and source code size measures (15 studies) were frequently used factors or measures (as independent variables). Some traditional measures related to the procedural paradigm were also used, such as Halstead in [34,35], and McCabe's cyclomatic complexity in [19,36]. Some researches, such as [37, 38], considered documentation quality to predict maintainability. Recently, many studies, such as [39–41] have investigated the effect of refactoring (i.e., improving the quality of the code and design of the software while preserving its external behavior) on maintainability.

Ways that maintainability (dependent variable) was expressed in the selected studies include: maintainability as a change measure (i.e., maintenance effort measured in terms of the number of lines changed per class during its maintenance history) was frequently used (25 studies), followed by expert opinion using an ordinal scale (10 studies), and then a maintainability index (three studies). In addition, some studies expressed maintainability in terms of the ISO 9126 sub characteristics, i.e., understandability and modifiability, such as [42].

To summarize, with respect to SPMP problems and corresponding datasets, we have noticed the followings.

- Most of the selected studies used historical datasets to predict maintainability in terms of change in lines of code (LOC) by counting the number of lines in the code that was changed per class. For instance, User Interface Management System (UIMS) and Quality Evaluation System (QUES) were the most frequently used historical datasets in predicting maintainability.

- Other studies used open source datasets, such as JEdit[13], JUnit[14], Log4j[15], and Ivy[16] and private data, such as File Letter Monitoring System (FLM), EASY

[12] http://www.core.edu.au/, Apr. 2020.

[13] http://www.jedit.org/, Apr. 2020.

[14] https://junit.org/, Apr. 2020.

[15] https://logging.apache.org/log4j/2.x/, Apr. 2020.

[16] http://ant.apache.org/ivy/index.html, Apr. 2020.

classes online services collection (Easy), Student Management System (SMS), Inventory Management System (IMS), and Angel Bill Printing (APB) to predict maintainability in terms of change in LOC also.

- Some studies predict maintainability of software using an ordinal scale based on expert opinion. Such studies generally used datasets from different software applications. The maintainability of the selected software applications was qualified as poor, average, very good or very high, high, medium, low, or excellent, average, bad, etc.
- Few studies predict maintainability in terms of understandability, modifiability and analyzability levels or time. These studies aim to build UML class diagram and/or sequence diagram maintainability prediction models based on the subject's rating. To this end, datasets used are UML class diagrams and/or sequence diagrams from different software applications.

3.2 SPMP Techniques (RQ1)

This subsection discusses the techniques used to predict software product maintainability in the 77 selected primary studies. The identified SPMP techniques are grouped into two main categories: ML and statistical techniques.

1) Statistical techniques were classified into regression analysis (RA), probability density function (PD), Gaussian mixture model (GMM), discriminant analysis (DA), weighted functions (WF) and stochastic model (SM).

2) Machine learning techniques, on the basis of [18, 43], were classified as artificial neural networks (ANN), case-based reasoning (CBR), regression and decision trees (DT), Bayesian networks (BN), evolutionary algorithm (EA), support vector machine and regression (SVM/R), fuzzy and neuro fuzzy (FNF), inductive rule based (IRB), ensemble methods (EM), and clustering methods (CM).

3.2.1 *Most Frequently Used SPMP Techniques*

Table 5 presents the selected studies related to each category. The most frequently used techniques were 57 studies (74%) in ML techniques compared with 37 studies (48%) in statistical techniques. It should be noted that if a study included both ML and statistical techniques it was counted in each category.

SPMP Statistical Techniques. Within the categories for statistical techniques (37 studies) regression analysis techniques were the most frequently used at 78% (29 studies) followed by the probability density function with 8% (three studies). Moreover, as shown in

Table 5. Distribution of Studies per SPMP Techniques

Category	Sub-Category	Reference	% (Number of Studies)
Statistical	Regression analysis (RA)	[19, 20, 24–26, 28, 30, 32, 34, 44–63]	78% (29)
	Probability density function (PD)	[40, 64, 65]	8% (3)
	Stochastic model (SM)	[41, 66]	5% (2)
	Discriminant analysis (DA)	[42]	3% (1)
	Gaussian mixture model (GMM)	[35]	3% (1)
	Weighted function (WF)	[67]	3% (1)
Machine learning	Artificial neural networks (ANN)	[23–27, 30–33, 44, 51, 54, 56, 57, 59, 61, 63, 68–81]	54% (31)
	Support vector machine/regression (SVM/R)	[24, 29, 30, 32, 35, 44, 56, 57, 59, 61, 62, 70, 74–76, 81–85]	35% (20)
	Regression and decision trees (DT)	[24, 25, 30, 32, 35, 44, 45, 51, 56, 57, 60, 61, 63, 70, 74, 75, 86]	30% (17)
	Fuzzy and neuro fuzzy (FNF)	[21, 23, 24, 27, 36–38, 76, 87–94]	28% (16)
	Ensemble methods (EM)	[24, 30, 32, 56, 59, 63, 74, 75]	14% (8)
	Bayesian networks (BN)	[25, 45, 59, 63, 70, 81]	11% (6)
	Case-based reasoning (CBR)	[24, 32, 63, 70, 95]	9% (5)
	Evolutionary algorithms (EA)	[39, 56, 57, 72, 81]	9% (5)
	Inductive rule based (IRB)	[24, 32, 81]	5% (3)
	Clustering method (CM)	[56, 96]	4% (2)

Fig.4(a), the multiple linear regression (MLR) technique in the RA category was the most frequently used at 30% (14 studies). Zhou and Leung[44] stated that MLR is “commonly used for modeling the relationship between two or more independent variables and a dependent variable by fitting a linear equation to observed data. The main advantages of this technique are its simplicity and its supportability by many popular statistical packages”.

Frequently used ML techniques (57 studies) were artificial neural networks (ANN) with 54% (31studies), followed by support vector machine and support vector regression (SVM/R) with 35% (20 studies), regression & decision trees (DT) with 30% (17 studies), and fuzzy & neuro fuzzy (FNF) with 28% (16 studies).

The distribution of the most frequently used ML

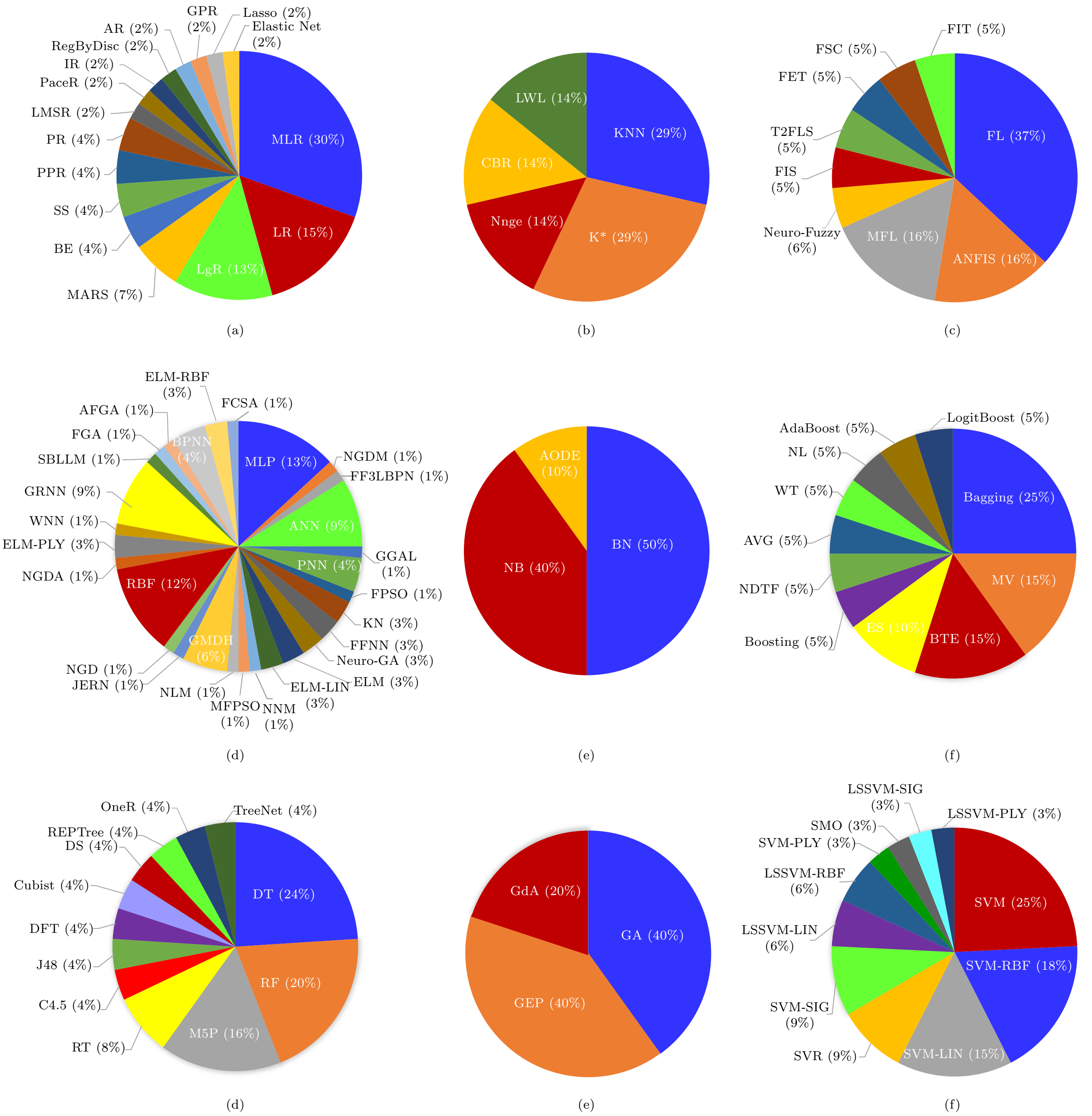


Fig.4. Distribution of the most-used sub categories of ML and statistical techniques in (a) RA, (b) CBR, (c) FNF, (d) ANN, (e) BN, (f) EM, (g) DT, (h) EA, and (i) SVM/R.

techniques for SPMP with respect to each identified category is presented in Figs.4(b)–4(i). As can be seen, the frequently used techniques were:

- MLP and RBF in ANN category (13% and 12% respectively),
- SVM in SVM/R category (25%),
- DT and RF in DT category (24% and 20% respectively),
- FL in FNF category (37%), bagging in EM category (25%),
- BN and NB in BN category (50% and 40% respectively),
- KNN and K* in CBR category (29% respectively),
- GA and GEP in EA category (40% respectively).

Therefore, a subset of nine top ML techniques were identified (i.e., used in more than three studies): MLP, RBF, SVM, DT, RF, FL, bagging, BN, and NB.

- MLP's main advantage is its adaptive nature, nonlinearity, parallel architecture, and fault tolerance[63].
- RBF has shown a "great promise in most sorts of problems because of its excellent learning capacity"[97].
- SVM aims to minimize empirical error and maximize geometric margin[74].
- DT can easily extract the "IF-THEN" rule[75].
- RF requires "very little preprocessing of data and no need to select a variable to start building a model. RF in itself selects the most useful variables"[63].
- FL offers significant advantages because of "its ability to naturally represent the human-provided qualitative linguistic knowledge with regard to the quality relationships and apply flexible inference rules"[76].
- Bagging technique "has produced good results whenever the learning algorithm is unstable"[56].
- NB is simple, easy to use and interpret; it is particularly appropriate when the dimensionality of the independent space is high[63] and "can often outperform other more sophisticated classification methods"[98].
- BN "can be a promising new technique for OO software maintainability prediction. This is due to its ability to explicitly represent uncertainty using probabilities, its ability to incorporate existing human expert knowledge into empirical data, and its ability to update the model when new information becomes available"[45].

3.2.2 Trends in SPMP Techniques

Investigation of the selected studies revealed that statistical techniques were used mostly between 2000 and 2007. However, these techniques work only when the relationship between the dependent and the independent variables is linear or has a predetermined form[16].

When ML techniques began to emerge, researchers studied both statistical and ML techniques in order to test their ability to predict maintainability. For instance,

- Kaur and Kaur[24] reported "classical parametric statistical data analysis methods may not be adequate. It is hypothesized that the use of ML algorithms or pattern recognition approaches that are essentially nonparametric may lead to better prediction accuracies."
- Van Koten and Gray[45] reported that ML techniques, e.g., BN achieved significantly a better prediction accuracy than the regression-based models.
- Kumar and Rath[31] studied a subset of ML techniques that were able to approximate the non-linear function with more precision.
- Malhotra and Chug[72] reported "the relationships between static software metrics and its maintainability are very complex and nonlinear, hence conventional statistical technique based models, which are purely based on quantity, would not help much to the problem. Instead, the use of ML algorithms to establish the relationship between metrics and maintainability would be a much better approach as these are based on quantity as well as quality."

From 2008 onwards the focus was mainly directed towards the use of ML techniques, especially comparative studies to accurately identify the best one for predicting maintainability. Such studies include [24–33, 35, 44, 45, 51, 54, 56, 57, 59–63, 68, 70–81, 84–86, 91, 93, 96]. This subset of comparative studies is investigated in more detail in RQ5.

In recent years, since every single technique has its advantages and drawbacks, the research trend has been to take advantage of the techniques used, mitigate their weaknesses and therefore obtain a more accurate technique.

New approaches based on using ensemble methods (i.e., combining different single base techniques based on some rules) were proposed in [24, 30, 32, 56, 59, 63, 74, 75]. For instance, Elish *et al.*[56] studied heterogeneous and homogeneous ensemble methods with different linear and non-linear combination rules for predicting software maintainability. Other studies proposed hybrid approaches integrating two or more techniques. For instance, ANN was combined with GA in [79, 80], and with FLANN, CSA, and PSO in [31]. FL was combined with ANN in S69[96].

To summarize, ANN, RA, SVM/R, DT, and FNF are the most frequently used techniques for predicting maintainability in 31, 29, 20, 17, and 16 selected studies respectively. A set of technique strengths (as per author opinion) is presented in Table 6. For instance:

• ANN has the ability to model complex non-linear relationships and is capable of approximating any measurable function.

• SVM/R has the ability to learn classification and regression tasks with high-dimensional data and is widely used in many domains.

• DT is simple, easy, and well comprehended.

• FNF can be used without any data or with little data.

• RA is easy and more dependable especially for cases involving more than two independent variables, which helps to build efficient maintainability techniques.

3.2.3 SPMP ML Tools

Seven tools SPMP ML were identified in the selected studies which are Matlab, Waikato Environment for Knowledge Analysis (Weka), Bayesian Discriminant Analysis (Bayda), Neuroshell 2, GMDH Shell, Decision Tree and Regression (DTReg), and CART 5.0. Matlab is the tool used most often (15 studies), followed by Weka (12 studies).

• Matlab[17] was developed by MathWorks. This tool is simple to use and provides many features such as matrix manipulation, algorithm implementation, and interface to programs written in other languages. From the selected studies, we found that this tool was used often to implement FL as well as SVM and ANN techniques.

• Weka[18] is a collection of ML algorithms for data mining tasks and provides tools for data preprocessing, classification, regression, clustering, association rules, and visualization.

The reminder tools are used only in one study.

Table 6. Strengths of Frequently Used Techniques for SPMP

SPMP Technique		Strength
Artificial neural network (ANN)	–	ANN is used for applications where formal analysis would be difficult or impossible, such as pattern recognition, nonlinear system identification, and control. The learning function can be applied to individual weights and biases within a network [22].
	–	ANN acts as an efficient predictor of dependent and independent variables due to its modeling characteristics, i.e., the ability to model complex functions [80].
	–	The neural network has self-learning and self-adapting ability, and ANN is robust and can suppress noise more effectively [75].
Support vector machine & support vector regression (SVM/R)	–	SVR analyzes data and recognizes patterns, which are used for classification and regression analysis [29, 59].
	–	SVM is well founded theoretically because it is based on well-developed statistical learning theory [35, 70].
	–	SVMs aim to minimize the empirical error and maximize the geometric margin [56, 74].
	–	SVM has the ability of learning classification and regression tasks with high-dimensional data and it is widely used in many domains [75].
	–	SVM offers more powerful regression capabilities [76].
	–	SVM is based on a robust estimator [24].
Decision tree (DT)	-	DT is simple and well understood [75].
	–	DT can easily extract the "IF-THEN" rule [75].
Fuzzy and neuro fuzzy	–	FNF can be constructed without data or with little data [21, 89].
(FNF)	–	FNF can adapt to new environments when data become available [21].
Regression analysis (RA)	–	Computations using regression techniques prove to be easier and more dependable, especially for cases involving more than two independent variables [55].
	–	"One of the advantages of regression is that we use continuous scale, so we expect more precise results" [99].

[17] https://www.mathworks.com/products/matlab.html, Apr. 2020.

[18] https://www.cs.waikato.ac.nz/ml/weka/, Apr. 2020.

- Neuroshell 2[19] is a legacy neural network product targeted towards computer science instructors and students.
- Bayda[20] is a software package developed in the University of Helsinki and used to construct a special type of BN called NB.
- DTReg[21] was developed by Devdigital in 2014 which handles both classification and regression problems.
- GMDH Shell[22] is a professional and easy-to-use open source tool for data mining.
- CART 5.0[23] is a decision tree tool used to generate predictive models.

There seem to be few tools for SPMP ML techniques, based on the results we obtained. Only two tools can be used for different ML techniques, while other tools implement specific ones. In addition, these two tools may not give all possible configurations of ML techniques and cannot deal with the increasing number of techniques. This lack of configurability and scalability may limit the use of SPMP ML techniques by practitioners. Thus, using programming languages such as R and Python is required because of their simplicity, consistency, and accessibility to libraries and frameworks for ML.

3.3 Validation Methods Used for SPMP Techniques (RQ2)

To validate the accuracy of SPMP techniques, the selected studies used a number of cross validation methods. Once the predictive model is built using the training set, it needs to be validated to test its effectiveness to predict new cases. Fig.5 shows the distribution of 32 out of the 77 selected studies that provided the information about the cross validation method used.

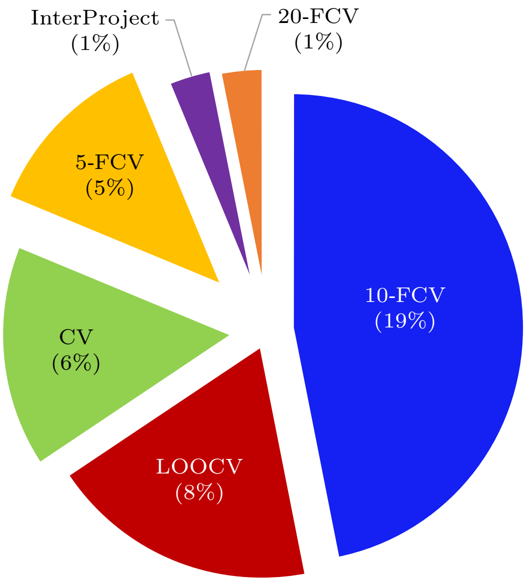


Fig.5. Cross validation methods per selected studies.

- N-fold cross validation (i.e., the dataset is divided into N equal parts where the training dataset contains $(N-1)$ parts with the remaining one used for testing) was the most used method, including:
  - ○ 10-fold cross validation (10-FCV): 15 studies [31, 32, 35, 45, 51, 52, 56, 57, 59, 61, 63, 69, 70, 74, 80];
  - ○ 5-fold cross validation (5-FCV): four studies ([30, 31, 80, 93]);
  - ○ 20-fold cross validation: one study of [85].
- The Leave-One-Out Cross Validation (LOOCV) (i.e., the training data is the whole dataset minus one single observation used as the test data): six studies ([24, 26, 44, 48, 56, 86]).
- One study [63] used inter-project validation (Inter-Project) (e.g., one dataset used for the model training and another dataset used for model validation). This kind of validation methods involve using datasets of a specific type of software for training and another dataset of a different type of software for validation. The purpose is to save resources by using inter project validation to assess the possibility of using developed models on different software datasets [63].
- The type of cross validation (CV) method of five studies ([33, 68, 71, 75, 77]) cannot be identified clearly.

3.4 Accuracy Criteria Used for SPMP Techniques (RQ3)

For any SPMP technique proposed, maintainability values predicted using datasets differ from the actual values in most cases. These two values (predicted and actual) if not equal may be close enough. To determine how accurate the proposed SPMP techniques are, a set of 53 accuracy criteria were used in the 77 selected studies (see details in Table A3 in Appendix).

[19] http://www.wardsystems.com/neuroshell2.asp, Apr. 2020.

[20] https://www.kdnuggets.com/software/bayesian.html, Apr. 2020.

[21] https://www.dtreg.com/, Apr. 2020.

[22] http://www.gmdhshell.com/, Apr. 2020.

[23] https://www.g6g-softwaredirectory.com/ai/data mining/20053A1SalfordSystsCart5NewEnh.php, Apr. 2020.

Since there are several ML techniques, different accuracy criteria have been used in empirical validations. The selection of these criteria may depend on several factors:

- objectives of ML techniques (regression, classification, clustering, or association);
- characteristics of datasets (imbalanced or balanced datasets);
- efficiency of the accuracy criteria used as reported in the literature.

Fig.6 presents the most frequently used accuracy criteria with their corresponding frequencies in the selected studies. The commonly used accuracy criteria are:

- 27% mean magnitude of relative error (MMRE),
- 21% percentage relative error deviation (Pred(p)) criteria (i.e., including $Pred(25/30)$,
- 16% mean absolute error (MAE),
- 14% coefficient of correlation (r, R, R-value),
- 12% root mean square error (RMSE),
- 10% mean absolute relative error (MARE),
- 10% maximum value of magnitude of relative error (MaxMRE),
- 10% R-square and accuracy,
- 9% recall,
- 6% F-measures, precision and AUC respectively.

In general, studies with statistical techniques commonly use accuracy criteria such as R and R-square, while those with ML techniques commonly use new criteria such as MMRE and Pred. Consequently, we selected these criteria to evaluate the prediction accuracy of ML techniques and answer RQ4 (see Subsection 3.5 for more details).

MMRE and $Pred(m)$ are based on the MRE criteria.

- MRE is a normalized measure of the discrepancy between actual values and predicted values, and MMRE is the mean magnitude of relative error, defined as:

$$MRE_i = \frac{|y_i - \hat{y}_i|}{y_i}, \quad (1)$$

$$MMRE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i}, \quad (2)$$

where y_i is the actual value and $\hat{y}_i$ is the predicted value.

$Pred(25/30)$ is the percentage of the predicted value with MRE less than or equal to 25%/30%.

$$Pred(m) = \frac{k}{n}, \quad (3)$$

where n is the total number of instances in the dataset, m is the specified value (25% or 30%), and k is the number of instances whose MRE is less than or equal to m.

Statistical Tests. As stated previously, the accuracy of SPMP techniques is measured in terms of difference between actual and predicted values. When different techniques are used to build prediction models, statistical tests are used to verify if differences among the results of the used techniques are statistically significant. Fig.7 provides the list of statistical tests identified from the selected studies as well as their frequency. As can be seen, Wilcoxon matched-pair signed-rank test, Friedman test, statistical hypothesis test, and t-test

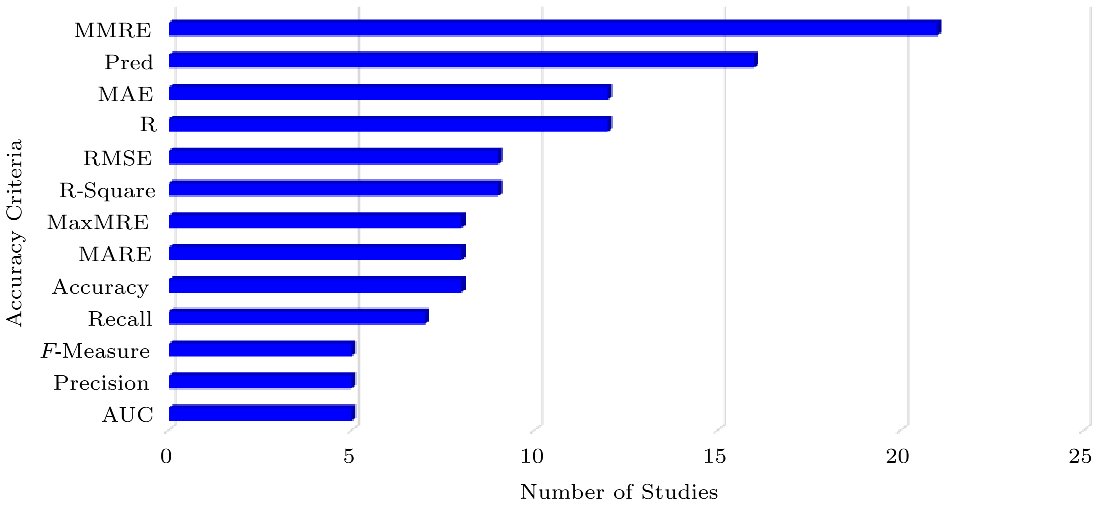


Fig.6. Most frequently used accuracy criteria per selected study.

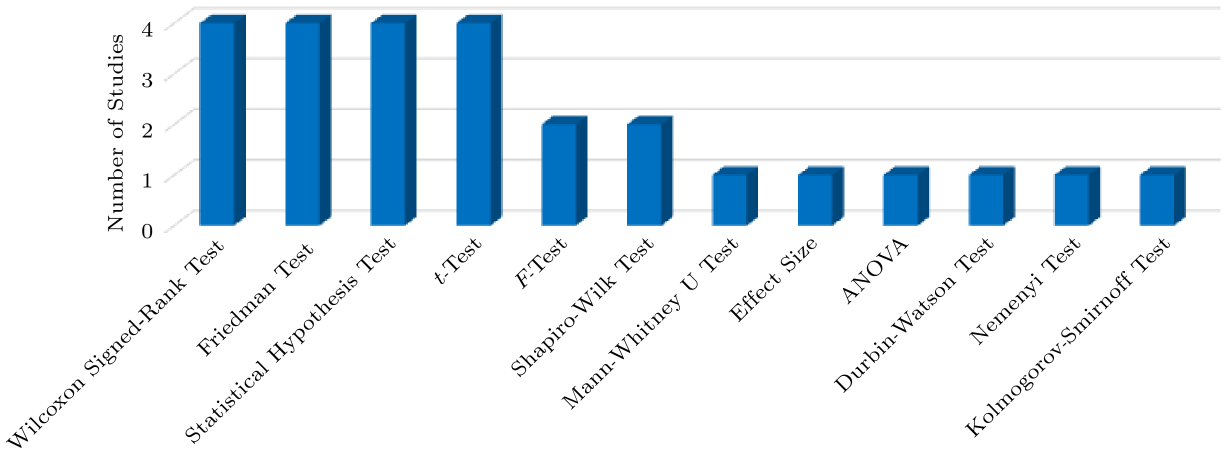


Fig.7. Statistical tests per selected study.

were used in four studies (5%) respectively, Shapiro-Wilk test and f-test were used in two studies (3%) respectively, and the remainder statistical test, such as Mann-Whitney U test, effect size, ANOVA, Durbin-Watson test, Nemenyi test, and Kolmogorov-Smirnoff were used in one study (1%) respectively.

3.5 Overall Accuracy of SPMP ML Techniques (RQ4)

The selection of criteria for defining an accuracy evaluation method for SPMP techniques is very challenging[16]. The performance of ML techniques in predicting SPMP can be affected by several factors related to prediction context[16, 18] such as the size of dataset, missing value, outlier, evaluation method used, accuracy criterion, and categorical feature, technique. Many selected SPMP studies used MMRE and Pred to determine the accuracy of their prediction techniques. MMRE was used in 20 of the studies (26%), $Pred(25)$ was used in 14 of the studies (18%), and $Pred(30)$ was used in 12 of the studies (16%). Due to the dominance of these accuracy criteria, we selected them to provide answers to RQ4.

According to Conte *et al.*, "MMRE has been regarded as a versatile assessment criterion and has a number of advantages, for example it can be used to make comparisons across datasets and all kinds of prediction model types and is independent of measuring unit and scale independent"[100]. Port and Korte reported that "Pred is immune to large variances from outliers in the data (i.e., it is more robust)"[101]. Moreover, for a prediction model to be considered accurate, either MMRE $\leqslant 0.25$ and/or either $Pred(25) \geqslant 75\%$ or $Pred(30) \geqslant 70\%$ are/is needed to be achieved[100, 102]. In general, a high value of Pred or a low value of MMRE indicates better prediction accuracy.

To provide answers to RQ4, the analysis of a subset of 20 (out of 77) selected studies on ML techniques was conducted, which provided the corresponding numerical values of MMRE and/or Pred criterion for each evaluation, as detailed in the following subsections (see Table A4 and Table A5 in Appendix for more details). It should be noted that one study may involve many evaluations (referred to as experiments henceforth).

3.5.1 *Accuracy Context of SPMP ML Techniques*

The accuracy of a prediction technique depends on several parameters such as 1) the technique, 2) the dataset, and 3) the accuracy criterion[16, 18]. These parameters are discussed in this subsection. To analyze the distribution of MMRE, $Pred(25)$, and $Pred(30)$ of SPMP ML techniques, we drew box plots corresponding to each of these accuracy criteria using the prediction accuracy values of the 20 selected primary studies.

As shown in Fig.8, the medians of the accuracy values of SPMP ML techniques are around 43% for MMRE, 58% for $Pred(25)$, and 57% for $Pred(30)$. We recall that, unlike MMRE, a higher value of Pred indicates a better prediction accuracy. It can also be seen that the distribution of SPMP techniques indicates positive skewness for MMRE since the median is closer to the lower quartile, and negative skewness for $Pred(25)$ since its median is closer to the higher quartile. For $Pred(30)$, the SPMP ML techniques are nearly in the center of the boxes, which indicates that they are symmetrically distributed around the median. Moreover, the lower and the upper quartiles are far from one

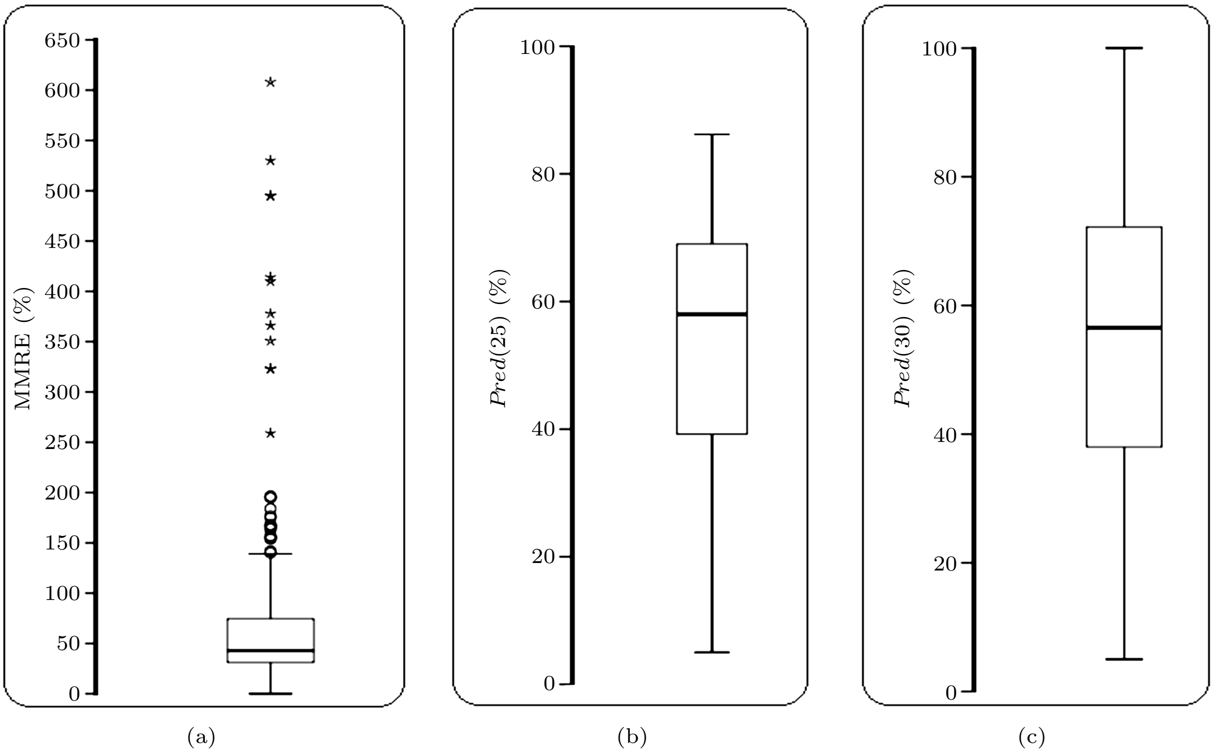


Fig.8. Box plots of (a) MMRE, (b) *Pred*(25) and (c) *Pred*(30).

another for MMRE, which suggest that there is a large variation of its values (196 experiments) compared with those for *Pred*(25) and *Pred*(30) (159 and 155 experiments respectively). This can be explained by different SPMP evaluation contexts, i.e., different SPMP techniques applied on a variety of datasets.

To further analyze the accuracy of SPMP techniques, Table 7 provides the detailed statistics of MMRE, *Pred*(25) and *Pred*(30) for the historical datasets in the selected primary studies. We recall that, unlike MMRE, a higher value of Pred indicates a better prediction accuracy. As can be seen, the mean of the prediction accuracy values varies from 27% to 47% for MMRE, from 41% to 74% for *Pred*(25) and from 47% to 65% for *Pred*(30) for all historical datasets except UIMS.

3.5.2 Accuracy of SPMP ML Techniques

To gain insight into the SPMP accuracy, box plots are used to illustrate the distribution of MMRE, *Pred*(25) and *Pred*(30) values per SPMP ML technique category (Table A5 in Appendix provides the values extracted from the 20 selected primary studies). The box plots include only ML techniques having more than three values of MMRE, and/or *Pred*(25), and/or *Pred*(30). We recall that, unlike MMRE, a higher value of Pred indicates a better prediction accuracy.

Fig.9(a) presents the box plot of the accuracy of SPMP ML techniques measured in MMRE. It can be seen that FNF is more accurate (median MMRE = 37%), followed by ANN (median MMRE = 40%), SVM/R (median MMRE = 42%), DT (median MMRE = 54%). CBR (median MMRE = 63%), and IRB (median MMRE = 92%). Note that the MMRE outliers for ANN came from seven studies ([24,44,56,61,71,74,77]). For SVM/R, the MMRE outliers came from five studies ([24, 44, 56, 61, 74]) and for DT techniques from three studies ([24, 44, 61]). For CBR and IRB, the MMRE outliers came from one study ([24]).

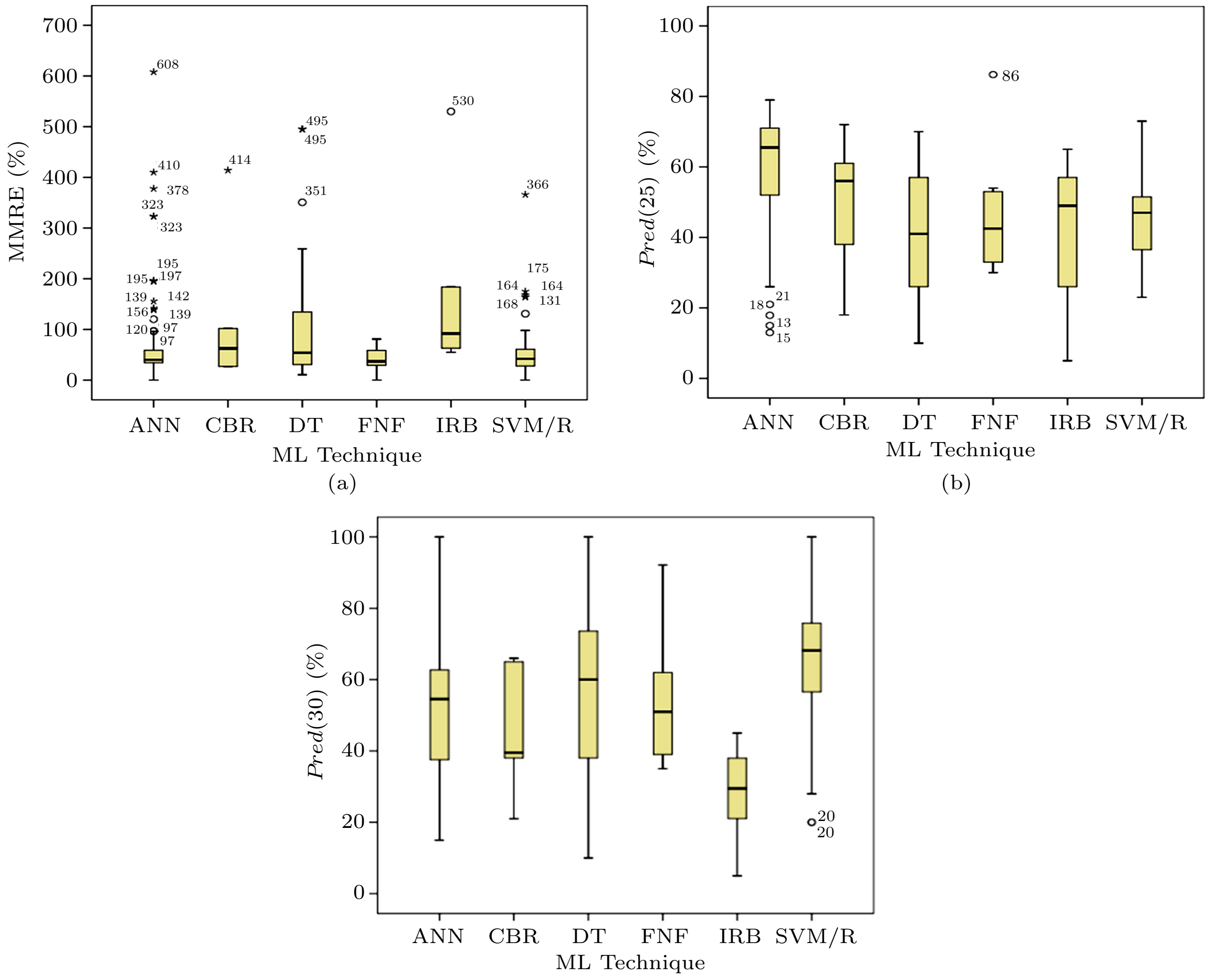
Fig.9(b) presents the box plot of the accuracy of SPMP ML techniques measured in *Pred*(25). It can be seen that ANN is more accurate (median *Pred*(25) = 66%), followed by CBR (median *Pred*(25) = 56%), IRB (median *Pred*(25) = 49%), SVM/R (median *Pred*(25) = 47%), FNF (median *Pred*(25) = 43%) and DT (median *Pred*(25) = 41%). Note that the *Pred*(25) outliers came from three studies [24, 44, 77] for ANN techniques and from one study [91] for FNF.

Based on the MMRE criteria (Fig.9(a)), it can be seen that SPMP techniques are symmetrically dis-

Table 7. Statistics for MMRE, *Pred*(25) and *Pred*(30) for Historical Datasets

Historical Dataset	MMRE					*Pred*(25)					*Pred*(30)				
	Number of Values	Min (%)	Max (%)	Mean (%)	Median (%)	Number of Values	Min (%)	Max (%)	Mean (%)	Median (%)	Number of Values	Min (%)	Max (%)	Mean (%)	Median (%)
UIMS	**40**	**0**	**323**	**119**	**116**	**27**	**5**	**45**	**25**	**26**	**35**	**5**	**47**	**27**	**23**
QUES	42	22	89	47	43	31	24	62	41	39	40	27	66	47	45
JEdit	-	-	-	-	-	11	42	75	60	59	-	-	-	-	-
JUnit	-	-	-	-	-	11	49	74	60	61	-	-	-	-	-
Log4j	-	-	-	-	-	10	65	77	72	73	-	-	-	-	-
Ivy	-	-	-	-	-	11	48	78	60	62	-	-	-	-	-
ABP	4	29	40	35	35	4	66	74	70	70	-	-	-	-	-
IMS	4	29	35	32	31	4	63	71	68	70	-	-	-	-	-
SMS	4	38	41	40	40	4	69	77	72	70	-	-	-	-	-
EASY	4	36	46	40	40	4	63	77	71	72	-	-	-	-	-
FLM	4	39	47	42	42	4	69	78	74	74	-	-	-	-	-
UIMS-QUES	4	21	41	27	23	4	34	69	59	67	4	40	75	65	72

Note: Since the *Pred*(30) values for ABP, IMS, SMS, EASY, JEdit, JUnit, Log4j, and FLM datasets were not provided in their corresponding studies, they are not presented in Table 7. For the same reason, the statistics for MMRE of JEdit, JUnit, and Log4j are not described in Table 7. Due to the small number of MMRE and *Pred*(30) values for Ivy, they are not presented in Table 7. Bold values indicate that this dataset achieves the lowest results.



(a)

(b)

(c)

Fig.9. Box plots of (a) MMRE, (b) *Pred*(25), and (c) *Pred*(30).

tributed around the median for CBR and SVM/R techniques, while the distribution of MMRE for ANN, DT, FNF, and IRB exhibits positive skewness, since their medians are closer to the lower quartile. Furthermore, the MMREs of ANN, SVM/R, and FNF have fewer variations since they have relatively narrower ranges of values and shorter boxes, compared with CBR, DT, and IRB techniques.

Fig.9(b) shows that *Pred*(25) is nearly in the center of the boxes for the FNF and DT techniques. In other words, the *Pred*(25) values are symmetrically distributed around the medians. The ANN, FNF, and SVM/R boxes are shorter and have narrower ranges of values, which indicates that their *Pred*(25) values have less variation than those of CBR, DT, and IRB. Also, the distribution of *Pred*(25) values indicates the negative skewness for ANN, CBR, IRB, and SVM/R techniques since the medians are closer to the higher quartile.

Fig.9(c) shows the box plot of *Pred*(30) where the median is 68% for SVM/R, 60% for DT, around 55% for ANN, 51% for FNF, 40% for CBR and 30% for IRB. Furthermore, FNF and IRB are systematically distributed about the median, since the observations split evenly at the median. Most of the observations are concentrated on the low end of the scale for CBR, which indicates positive skewness. Also, the distribution of *Pred*(30) values indicates negative skewness for DT, ANN, and SVM/R since their corresponding medians are closer to the higher quartile. Note that the *Pred*(30) outliers for SVM/R came from two studies, [74] and [56].

To sum up, according to the box plots in Figs.9(a)–9(c), more accurate ML techniques for SPMP are FNF and ANN in terms of MMRE, ANN, and CBR in terms of *Pred*(25), and SVM/R and DT in terms of *Pred*(30).

Besides, as a complement to the box plots used to understand the estimation accuracy of ML techniques, statistical analysis was used. Outliers were removed before calculating the statistics. Table 8 presents the minimum, maximum, mean, and median of MMRE, *Pred*(25), and *Pred*(30) for the SPMP ML techniques. To facilitate the analysis of the accuracy criteria, only techniques with more than three values were included in the analysis. We recall that, unlike MMRE, a higher value of Pred indicates a better prediction accuracy. As can be seen in Table 8, ANN, FNF, and SVM/R techniques have MMRE arithmetic means ranging from 36% to 41%. These techniques are more accurate compared with DT, CBR, and IBR, which have MMRE arithmetic means ranging from 56% to 79%. For *Pred*(25), all techniques had their arithmetic means ranging from 40% to 63%, while for *Pred*(30), all ML SPMP techniques had arithmetic means ranging from 45% to 68% except for IRB with 28%.

ANN was compared with DT and SVM/R since they were discussed in a large number of experiments (86, 43, and 32 respectively). Note that:

- ANN outperformed DT and SVM/R in terms of median *Pred*(25).
- SVM/R outperformed ANN and DT in terms of median MMRE and median *Pred*(30).

FNF was compared with CBR and IRB since they have approximately the same number of values (13 for CBR and IRB respectively and 12 for FNF). Note that:

- FNF outperformed CBR and IRB in terms of median MMRE and median *Pred*(30).
- CBR outperformed FNF and IRB in terms of median *Pred*(25).

Table 9 provides the number of studies that evaluated each SPMP ML technique and the number of evaluations (also referred to as "experiments" within some studies) performed within these studies. It shows that the number of evaluations was higher than the number of studies conducted. In addition, a single study ([24]) performed six evaluations for CBR and IRB respectively. The use of various evaluations from the same study may give biased results in terms of accuracy.

Table 8. Statistics for MMRE, *Pred*(25), and *Pred*(30)

ML Technique	MMRE					*Pred*(25)					*Pred*(30)				
	Number of Values	Min (%)	Max (%)	Mean (%)	Median (%)	Number of Values	Min (%)	Max (%)	Mean (%)	Median (%)	Number of Values	Min (%)	Max (%)	Mean (%)	Median (%)
ANN	76	0	96	41	38	86	26	79	63	67	55	15	100	51	55
DT	41	10	259	70	53	21	10	70	40	41	43	10	100	56	60
SVM/R	28	0	98	36	33	11	23	73	46	47	32	28	100	68	70
FNF	12	0	81	41	37	7	30	54	41	34	8	35	92	54	51
IRB	5	55	184	79	89	13	5	65	42	49	6	5	45	28	30
CBR	5	27	102	56	56	13	18	72	51	56	6	21	66	45	40

Table 9. Number of Studies with Number of Evaluations

	ANN	DT	SVM/R	FNF	CBR	IRB
Number of studies	14	8	6	5	2	2
Number of evaluations	86	43	32	12	13	13

In order to avoid this bias, the analysis of SPMP ML technique accuracy was conducted based on the number of studies instead of the number of evaluations. As can be seen from Table 9, ANN, DT, SVM/R, and FNF were the techniques most often used in predicting software product maintainability (more than two studies). Table 8 reveals that:

- for *Pred*(25): ANN outperformed SVM/R, FNF, and DT, while
- for MMRE: SVM/R outperformed FNF, ANN, and DT,
- for *Pred*(30): SVM/R outperformed DT, ANN, and FNF.

In summary, our analysis suggests that in general, ANN, SVM/R, and FNF are more accurate ML techniques for SPMP.

3.5.3 Accuracy of SPMP ML Techniques on UIMS and QUES Historical Datasets

Two historical datasets, UIMS and QUES were frequently used in empirical studies, and a set of 16 studies provided the accuracy in terms of MMRE, *Pred*(25), and *Pred*(30) values. The statistical analysis presented in Table 10 includes SPMP ML techniques having more than two values. Table 10 shows the followings.

- FNF outperformed ANN and CBR in terms of median MMRE.
- ANN and CBR outperformed FNF in terms of median *Pred*(25).
- FNF outperformed SVM/R and CBR in terms of median *Pred*(30).

This finding confirms that ANN and FNF are more accurate. Most of the studies used the Li and Henry datasets[103]. Therefore, the analysis of results is approximately the same as in Subsection 3.5.2. In fact, these two datasets are about one project, and about software systems developed using the ADA programming language. As stated by Kaur and Kaur[24], this can be problematic in terms of generalizing the review results to software systems developed using other object-oriented programming languages.

3.5.4 Synthesis

A lower MMRE or a higher Pred indicates a more accurate prediction. On the performance of SPMP ML techniques measured in MMRE and *Pred*(25) or *Pred*(30) (Subsection 3.5.2):

- ANN and SVM/R are more accurate (around 40% of median MMRE and 66% of median *Pred*(25) for ANN, and around 42% of median MMRE and 68% of median *Pred*(30) for SVM/R).
- FNF and DT (decision tree) (around 37% of median MMRE and 51% of median *Pred*(30) for FNF, and around 54% of median MMRE and 60% of median *Pred*(30) for DT),
- CBR and IRB (around 63% of median MMRE and 56% of median *Pred*(25) for CBR, and around 92% of median MMRE and 49% of median *Pred*(25) for IRB).

These findings about ANN and SVM/R are highly consistent with the results reported in [18], but in the context of software development effort estimation. Wen *et al.*[18] found in their SLR that ANN and SVR are more accurate ML techniques (with median MMRE around 35% and median *Pred*(25) around 70%).

However, we should avoid drawing such potentially misleading conclusions. Different studies may have evaluated the same ML techniques, but in different ways and from different aspects. We have reported in this paragraph some of the strengths and weaknesses identified in the selected primary studies considering that they represent their authors' opinions whereas

Table 10. Descriptive Statistics of Accuracy of ML Techniques on UIMS and QUES Datasets

ML Technique	MMRE					*Pred*(25)					*Pred*(30)				
	Number of Values	Min (%)	Max (%)	Mean (%)	Median (%)	Number of Values	Min (%)	Max (%)	Mean (%)	Median (%)	Number of Values	Min (%)	Max (%)	Mean (%)	Median (%)
ANN	35	21	142	58	42	18	13	69	37	37	26	15	75	38	38
DT	16	38	259	104	70	14	10	58	30	30	18	10	65	34	36
FNF	12	0	81	41	37	7	30	54	41	34	8	35	92	54	51
SVM/R	8	35	168	99	88	4	23	39	32	33	8	20	57	39	41
CBR	5	27	102	56	56	6	18	62	41	37	6	21	66	45	40
IRB	5	55	184	97	89	6	5	41	24	25	6	5	45	28	30

some authors may have a favorable bias towards some techniques. For instance:

- ANN has the ability to model complex non-linear relationships and is capable of approximating any measurable function[80]. But, it is highly dependent on the chosen network architecture[26], and on other issues such as local minimum, improper learning rate and over-fitting[77].
- SVM/R has the ability to learn classification and regression tasks with high-dimensional data[75], and minimizes empirical error and maximizes the geometric margin[56, 74].
- FNF can be used without any data or with little data[21, 89] and it is well founded theoretically, as it is based on well-developed statistical learning theory[35, 70]. However, the loss of interpretability in fuzzy modeling is mainly in the process simulation and control domain[76].

Moreover, most of the studies used the UIMS and the QUES datasets provided by Li and Henry[103] (Subsection 3.5.3) each of which is a small and about one project. Therefore, the conclusion drawn about ANN performance should be considered carefully and cannot be generalized since the favorable context of ANN is with large datasets[16, 18]. Furthermore, the obtained results are mainly based on historical datasets of software projects, and most of these datasets used are too obsolete to be representative of recent trends in software maintainability. Therefore, further studies are encouraged to use case studies, experiments and real-life evaluations of ML techniques in industry and to take into account both the availability of the datasets and how representative they are.

3.6 SPMP Techniques Reported to Be Superior (RQ5)

As stated before, the performance of ML techniques in predicting SPMP is empirically context-dependent. Thus, several factors can affect the prediction results such as the size of datasets, missing values, outliers, evaluation methods used, accuracy criteria, categorical features and the technique itself. In previous published work, we have conducted the accuracy comparison of SPMP techniques taking into account some of these factors[104], i.e., the same datasets, measures, accuracy criteria, and the same software development paradigm regardless of the techniques. In this study, the purpose of the accuracy comparison of ML SPMP techniques is to deal with only the techniques regardless of other factors.

3.6.1 Comparative Studies: An Overview

This subsection provides an overview of all identified comparatives studies (41 out of 77 selected studies) regarding the number of techniques used within each study, the most compared techniques and those reported superior.

A wide variation was noted with regard to the number of techniques used within each study. For instance, a single study ([24]) evaluated 25 techniques, two studies ([32] and [30]) evaluated 12 and 18 studies, respectively, 13 studies ([25, 27, 31, 44, 54, 56, 57, 59, 60, 63, 70, 76, 81]) evaluated 10 to five techniques, 18 studies ([26, 28, 29, 35, 45, 51, 61, 62, 68, 72, 74, 75, 77–79, 84, 85, 96]) evaluated four to two techniques, and seven studies ([33, 71, 73, 80, 86, 91, 93]) evaluated a single technique.

Regarding the most compared techniques, ANN was the most commonly used in comparative studies (29 studies), followed by RA (26 studies), DT (24 studies), SVM/R (22 studies), BN (15 studies), FNF (five studies), CBR and EA (three studies respectively), IRB and CM (two studies respectively).

- For ML techniques, BN in the BN category (14 studies), RT in the DT category (11 studies), RBF, ANN, and MLP were frequently used for the comparison in the ANN category (nine studies for RBF and ANN, and eight for MLP), SVM and SVR in SVM/R category (eight studies respectively), ANFIS in the FNF category (three studies), GEP in the EA category (two studies) and KMC in the CM category (two studies).
- For statistical techniques, MARS, BE, SS, and MLR were the most frequently used techniques for the comparison in the RA category (11 studies for MARS and 10 for BE, SS, and MLR respectively).

With respect to the most accurate techniques, a subset of 46 techniques were reported to be superior in 41 primary studies, in a total of 136 experiments. Note that one study may involve more than one experiment. ANN was reported accurately in 38 experiments, SVM/R and DT in 29 experiments respectively, RA in 14 experiments, FNF and EA in nine experiments respectively, BN in five experiments, CBR in two experiments, and CM in one experiment.

- The more accurate ML techniques for SPMP were M5P in the DT category (16 experiments), SVR in the SVM/R category (nine experiments), MLP and GMDH in the ANN category (nine and eight experiments respectively), GEP in the EA category (five experiments), BN in the BN category (four experiments), MFL in the

FNF category (three experiments), K* in the CBR category (two experiments), and XMC in the CM category (one experiment).

• For statistical techniques, MARS is more accurate in the RA category (seven experiments).

The following subsections provide accuracy comparison between ML techniques and non-ML techniques (statistical), and comparison among different ML techniques.

3.6.2 *Accuracy Comparison Between ML and Statistical Techniques*

Frequently-used ML techniques such as ANN, DT, SVR/M, FNF, IRB, and CBR (in more than two experiments) were compared with RA, the most frequently used statistical technique. The details of the comparison are provided in Table A6 in Appendix, and the overall results, obtained by counting the number of experiments in which the ML technique outperforms (shown by bars above the zero-line) or underperforms (shown by bars below the zero-line) the RA technique, are summarized in Fig.10. We adopted the same analysis as that used by Wen *et al.* "One model is said to outperform another in an experiment if the MMRE value of the first model achieves at least 5% improvement over that of the second model"[18].

Fig.10 shows that, based on the MMRE accuracy criterion, the majority of the experiments reveal that ML techniques outperform statistical ones in 58% of experiments (249 out of 426) while they underperform them in 42% (177 out of 426).

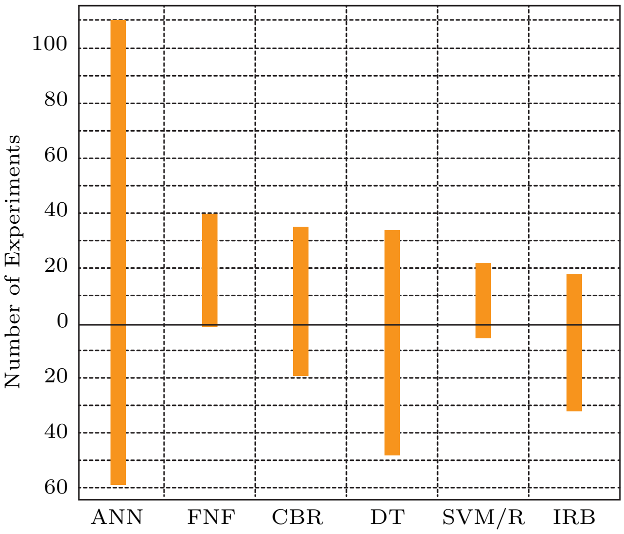


Fig.10. Comparison of MMRE between ML techniques and RA (the bars above the zero line indicate that ML techniques are more accurate, whereas the bars below the zero line indicate that RA is more accurate).

The results show that ANN outperforms RA in 65% (110 out of 169 experiments), FNF in 98% (40 out of 41) of experiments, and SVM/R in 81% (22 out of 27 experiments). In addition, RA outperforms DT in 59% of experiments (48 out of 81) while it underperforms them in 41% (33 out of 81). It is difficult to determine whether CBR and IRB are more accurate than RA because even if the number of experiments reporting CBR outperformed RA is higher and the number of experiments reporting RA outperformed IRB is also higher, the results come from a single study ([24]). Hence, this finding limits the possibility of generalizing the comparison results.

According to the above results, it can be concluded that in general ML techniques outperformed statistical ones. However, the number of studies included in the comparison was only 27 out of 77, a potential threat to the reliability of these findings.

3.6.3 *Accuracy Comparison Among ML Techniques*

We adopted the same analysis as used in the comparisons between ML and non-ML techniques. Fig.11 presents the results of comparison based on MMRE values, together with the corresponding number of supporting experiments. Included techniques are those used in more than two experiments. Details are presented in Table A7 in Appendix.

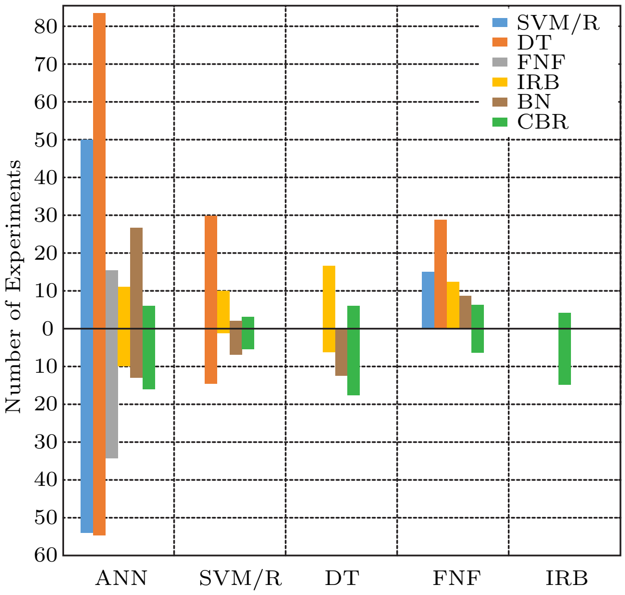


Fig.11. Comparisons of MMRE among ML techniques (the bars above the zero line indicate that techniques in horizontal axis are more accurate, whereas the bars below the zero line indicate that techniques in horizontal axis are less accurate).

The main results of this comparison (see Fig.11) are summarized as follows. FNF outperformed ANN in most experiments. FNF outperformed SVM/R, DT, IRB, and BN in all experiments. However, the finding for FNF vs SVM/R is inconsistent with that found in Subsection 3.5.2 where SVM/R performed better than FNF (see Table 8 in Subsection 3.5.2). This inconsistency may be explained by a lack of studies comparing FNF with SVM/R techniques, and all experiments demonstrating the superiority of FNF over SVM/R came from the FNF studies, thereby it is possible that some have a bias towards FNF. The results also show that the IRB technique did not outperform any of the ML techniques. Furthermore, comparisons among different ML techniques are rare; hence it is difficult to determine which may be more accurate.

The results of the comparison show that no technique was definitively better than another. This is related to the experimental context of the studies. In this context, a study was conducted in order to compare some techniques reported to be superior and which have the same experimental context, i.e., the same datasets, measures, accuracy criteria, and the same software development paradigm as in [104], i.e., the same independent variables (Chidamber and Kemerer, and Li and Henry metrics), the same dependent variable (change), the same accuracy criteria (MMRE, $Pred(25)$, $Pred(30)$), and the same software development paradigm (object-oriented). From this comparison, we found that MFL, K*, and KNN techniques achieved better accuracy prediction compared with the other techniques for the QUES dataset, while BN and ELM techniques were more accurate compared with other techniques on the UIMS dataset. The results show that technique performance is effectively context-dependent, which means the best technique should be chosen for that particular context[105]. Moreover, Briand *et al.*[106] stated "software engineering datasets suffer from problems of heteroscedasticity, missing values, lack of precision of data collection process, different scales of predictor variables and unknown probability distributions of predictor variables". Therefore, further studies are needed to find the most accurate technique for predicting maintainability of a software product. Moreover, when comparing different techniques, we may need to consider the real problem that the technique applies to and also the data that the technique can leverage.

4 Threats to Validity

There are several threats to the validity of our study for software maintainability prediction techniques, based on [16, 107]. One potential threat is internal. The selection was carried out up to 2018 using different databases to identify the relevant empirical studies on SPMP techniques. However, the list of studies selected may not be complete and a suitable study may have been left out, especially for the 2018 publications which were not available online at the time of our research study.

External validity is related to the completeness of the selected data. This bias was avoided by reporting the data that were clearly stated by the primary studies and used without making any assumptions. Another point concerns the generalization of the results. For example for techniques reported as superior, each study used a different experimental setting: the dataset (dependent and independent variables), the technique or the algorithm used to build the model and the validation procedure. The accuracy values were extracted from selected studies that validated the empirically proposed techniques for software maintainability prediction. Since we refrained from deriving or adjusting any data or values, the comparison between SPMP techniques was considered impartial.

Construct validity is related to the accuracy criteria used for the analysis where MMRE and Pred are based on the magnitude relative error (MRE). The literature reported some weaknesses of these criteria such as ignoring dataset quality[16] and assuming that the prediction model is able to predict the accuracy up to 100%[108]. MMRE has been criticized to be sensitive to outliers[101]. However, they are commonly used in the selected studies, which has allowed us to conduct our analysis and comparison.

5 Conclusions

A systematic review of SPMP ML techniques was conducted based on a set of 77 selected studies collected from 2000 to 2018. Results of the review were discussed with regard to SPMP techniques (RQ1), validation methods (RQ2), accuracy criteria (RQ3), the overall accuracy of SPMP ML techniques (RQ4), and the comparison of SPMP technique accuracy (RQ5). The main findings are summarized as follows.

What Are the Most Frequently Used SPMP Techniques? ML techniques are the most frequently used techniques followed by statistical methods. Regression analysis (RA), especially multiple linear regression (MLR) was the most frequently used statistical technique. The most commonly used ML techniques in SPMP were ANN, SVM/R, DT, and FNF. Also, few hybrid techniques were proposed in the selected studies, while ensemble techniques were even less used. For the tools support for SPMP techniques, we found that Matlab and Weka were the most frequently used tools for ML techniques.

What Are the Validation Methods Most Used for SPMP Techniques? The N-fold and leave-one-out cross-validation methods were used to validate SPMP techniques. Inter-project validation was recently used as a validation method to assess the possibility of using developed models on different software datasets.

What Are the Most Frequently Used Accuracy Criteria for SPMP Techniques? Many criteria were applied to assess the accuracy of maintainability prediction. The results indicated that MMRE, $Pred$ (including $Pred(25)$, $Pred(30)$, and $Pred(75)$), MAE, R, RMSE, MARE, MaxMRE, R-square, Recall, F-measures, Precision, and AUC were the most commonly used. In addition, MMRE, $Pred(25)$, and $Pred(30)$ were the top 3 used in SPMP empirical studies. Besides, many statistical tests were used in order to verify the differences in accuracies of SPMP techniques, and the most used ones are Wilcoxon matched-pair signed-rank test, Friedman test, statistical hypothesis test, and t-test.

What Is the Overall Accuracy of SPMP ML Techniques? The overall prediction accuracy was analyzed based on the prediction accuracy values of MMRE, $Pred(25)$ and $Pred(30)$.

- Based on historical datasets, ML techniques achieve acceptable results, with the mean MMRE ranging from 27% to 47%, the mean $Pred(25)$ ranging from 41% to 74%, and the mean $Pred(30)$ ranging from 47% to 65% (except UIMS).

- The statistical analysis of ML techniques for all datasets suggests that ANN, SVM/R, and FNF are more accurate.

- The statistical analysis of ML techniques, using the most frequently used datasets (UIMS and QUES), suggests that FNF and ANN are more accurate.

Are There SPMP Techniques Reported to Be Superior in the Literature? Up to 53% of the selected empirical studies were comparatives studies, with the most compared techniques being ANN, RA, DT, and SVM/R.

- RA techniques (such as MARS, BE, and SS), ANN techniques (such as MLR, RBF, ANN, and MLP), DT (such as RT), and SVM/R (such as SVM and SVR) were the most often used in comparative studies.

- Accurate SPMP techniques were ANN, SVM/R, DT, and RA (in particular, MLP and GMDH were reported to be superior in the ANN category, SVR in the SVM/R category, M5P in the DT category, and MARS in the RA category).

- Accuracy comparison of ML with statistical techniques showed that in general the former outperformed the latter in terms of the RA technique.

- Accuracy comparison among different ML techniques showed that FNF outperformed SVM/R, DT, IRB, and BN in all experiments while outperforming ANN in most experiments.

In addition to these above findings, the following research gaps were identified.

- Despite the large number of SPMP studies, we were unable to identify the best techniques or guidelines to build the most accurate. Indeed, all SPMP techniques are prone to errors as they depend to some degree on the experimental context. No single technique can give a “right” result in all circumstances. Therefore, generalized results regarding the prediction of software product maintainability are still not available to meet software industry needs and the expectations for software maintainability prediction. In addition, few studies have investigated the effectiveness of hybrid and ensemble techniques. Ensemble techniques use many single base techniques, taking the advantages of each and mitigating their weaknesses in order to obtain a more accurate technique. Thus, proposing new approaches based on ensembles techniques may be fruitful.

- The identification of the best SPMP techniques relies mainly on the dataset being investigated since a technique may behave differently from one dataset to another, which makes them unstable. Frequently-used datasets are UIMS and QUES, based on projects developed in ADA. Datasets using new technologies such as Java represent a serious challenge. Moreover, none of the studies considered the effect of dataset properties, such as the size, missing data, and outliers, on the accuracy of SPMP techniques. One of the important points with datasets is data preprocessing which allows handling missing data. Many ways are available as seen in the literature such as deletion and imputation. Also, a dataset includes many attributes, some of which may be irrelevant to the problem being studied, while others may be redundant. By using feature selection, only the most meaningful attributes are selected, which gives better accuracy of the technique or even increases it. Selection can be carried out in many ways including filter methods, wrapper methods, and embedded methods. Moreover, feature extraction using principal component analysis has also been used in some studies. This method extracts new features based on the existing data. Feature selection and extraction have been used recently, but only in a few selected studies.

- Most comparative studies conducted on SPMP techniques reported only the most accurate by analyzing the values of the accuracy criteria used. Few studies used statistical significance tests to validate the findings. For instance, the Friedman test was used to statistically investigate the difference in the accuracy between techniques, while the pair-wise Wilcoxon signed rank test was used to identify how these techniques differ.

- Many measures and indicators were used in the empirical studies. But, unfortunately, some authors used different abbreviations for the same method. For example, a number of methods may be named NOM, NM or NMETH, where all these abbreviations mean the same thing; that is to measure the number of methods (NOM, NM or NMETH) in a system. These inconsistencies may cause ambiguities for the interpretation of the results and for comparative studies. Hence, it is imperative to unify the terminology and classify terms into a measurement framework that will assist researchers to build predictive models more easily.

- Maintainability as a quality characteristic of a software product was expressed differently in various selected studies. Several researchers have proposed definitions either by using or adapting ISO standards, such as ISO 9126 or ISO 25010 maintainability characteristics or sub-characteristics, or proposing their own definition based on their experience in the domain. Measures for quality characteristics and sub-characteristics are in general proposed by authors to determine the maintainability of software. However, no detailed structural model that unifies all the maintainability characteristics exists in the literature or industry in general. Thus, there is the need to improve the maintainability quality model based on ISO 25 010 and the definitions of maintainability and its sub characteristics provided in the literature.

References

[1] Abran A, Nguyenkim H. Measurement of the maintenance process from a demand-based perspective. *J. Softw. Maint. Res. Pract.*, 1993, 5(2): 63-90.

[2] Abran A, Bourque P, Dupuis R, Moore J W. Guide to the Software Engineering Body of Knowledge — SWEBOK. IEEE Press, 2001.

[3] Riaz M, Mendes E, Tempero E. A systematic review of software maintainability prediction and metrics. In *Proc. the 3rd International Symposium on Empirical Software Engineering and Measurement*, October 2009, pp.367-377.

[4] Riaz M. Maintainability prediction of relational database-driven applications: A systematic review. In *Proc. the 16th International Conference on Evaluation & Assessment in Software Engineering*, May 2012, pp.263-272.

[5] Orenyi B A. Basri S, Jung L T. Object-oriented software maintainability measurement in the past decade. In *Proc. the 2012 International Conference on Advanced Computer Science Applications and Technologies*, Nov. 2012, pp.257-262.

[6] Dubey S K. Sharma A, Rana A. Analysis of maintainability models for object oriented system. *Int. J. Comput. Sci. Eng.*, 2011, 3(12): 3837-3844.

[7] Burrows R, Garcia A, Taïani F. Coupling metrics for aspect-oriented programming: A systematic review of maintainability studies. In *Proc. the 4th International Conference on Evaluation of Novel Approaches to Software Engineering*, May 2009, pp.277-290.

[8] Saraiva J, Barreiros E, Almeida A *et al.* Aspect-oriented software maintenance metrics: A systematic mapping study. In *Proc. the 16th International Conference on Evaluation & Assessment in Software Engineering*, May 2012, pp.253-262.

[9] Kumar B. A survey of key factors affecting software maintainability. In *Proc. the 2012 International Conference on Computing Sciences*, Sept. 2012, pp.261-266.

[10] Tiwari G, Sharma A. Maintainability techniques for software development approaches — A systematic survey. *Int. J. Comput. Appl. Special Issue on Issues and Challenges in Networking, Intelligence and Computing Technologies*, 2012, ICNICT(4): 28-31.

[11] Saini M, Chauhan M. A roadmap of software system maintainability models. *Int. J. Softw. Web Sci.*, 2013, 2(3): 69-73.

[12] Vern R, Dubey S K. A review on appraisal techniques for web based maintainability. In *Proc. the 5th Int. Conf. Conflu. Next Gener. Inf. Technol. Summit*, Sept. 2014, pp.795-799.

[13] Rajendra K, Namrata D. Maintainability quantification of object oriented design: A revisit. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, 2014, 4(12): 461-466.

[14] Elmidaoui S, Cheikhi L, Idri A. Software product maintainability prediction: A survey of secondary studies. In *Proc. the 4th International Conference on Control, Decision and Information Technologies*, April 2017, pp.702-707.

[15] Zhang D, Tsai J J P. Machine Learning Applications in Software Engineering. World Scientific Publishing Co., 2005.

[16] Idri A, Amazal F A, Abran A. Analogy-based software development effort estimation: A systematic mapping and review. *Inf. Softw. Technol.*, 2015, 58: 206-230.

[17] Kitchenham B, Charters S. Guidelines for performing systematic literature reviews in software engineering. Technical Report, Keele University and University of Durham, 2007. http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=AE5CA418E8C8629591E397A8ECF1450E?doi=10.1.1.117.471&rep=rep1&type=pdf, Dec. 2019.

[18] Wen J, Li S, Lin Z, Hu Y, Huang C. Systematic literature review of machine learning based software development effort estimation models. *Inf. Softw. Technol.*, 2012, 54(1): 41-59.

[19] Muthanna S, Kontogiannis K, Ponnambalam K, Stacey B. A maintainability model for industrial software systems using design level metrics. In *Proc. the 7th Working Conference on Reverse Engineering*, November 2000, pp.248-256.

[20] Genero M, Piattini M, Manso E, Cantone G. Building UML class diagram maintainability prediction models based on early metrics. In *Proc. the 9th International Symposium on Software Metrics*, Sept. 2003, pp.263-275.

[21] Sharma A, Grover P S, Kumar R. Predicting maintainability of component-based systems by using fuzzy logic. In *Proc. the 2nd International Conference on Contemporary Computing*, August 2009, pp.581-591.

[22] Sharawat M S. Software maintainability prediction using neural networks. *International Journal of Engineering Research and Applications*, 2012, 2(2): 750-755.

[23] Al-Jamimi H A, Ahmed M. Prediction of software maintainability using fuzzy logic. In *Proc. the 3rd IEEE International Conference on Computer Science and Automation Engineering*, June 2012, pp. 702-705.

[24] Kaur A, Kaur K. Statistical comparison of modelling methods for software maintainability prediction. *Int. J. Softw. Eng. Knowl. Eng.*, 2013, 23(6): 743-774.

[25] Kaur A, Kaur K, Pathak K. Software maintainability prediction by data mining of software code metrics. In *Proc. the 2014 International Conference on Data Mining and Intelligent Computing*, Sept. 2014.

[26] Wang L, Hu X, Ning Z, Ke W. Predicting object-oriented software maintainability using projection pursuit regression. In *Proc. the 1st International Conference on Information Science and Engineering*, Dec. 2009, pp.3827-3830.

[27] Kaur A, Kaur K, Malhotra R. Soft computing approaches for prediction of software maintenance effort. *Int. J. Comput. Appl.*, 2010, 1(16): 69-75.

[28] Kaur A, Kaur K, Pathak K. A proposed new model for maintainability index of open source software. In *Proc. the 3rd International Conference on Reliability, Infocom Technologies and Optimization*, Oct. 2014.

[29] Kumar L, Krishna A, Rath S K. The impact of feature selection on maintainability prediction of service-oriented applications. *Serv. Oriented Comput. Appl.*, 2017, 11(2): 137-161.

[30] Kumar L, Ashish S. A comparative study of different source code metrics and machine learning algorithms for predicting change proneness of object oriented systems. arXiv:1712.07944, 2017. https://arxiv.org/abs/1712.07944, Dec. 2019.

[31] Kumar L, Rath S K. Hybrid functional link artificial neural network approach for predicting maintainability of object-oriented software. *J. Syst. Softw.*, 2016, 121: 170-190.

[32] Chug A, Malhotra R. Benchmarking framework for maintainability prediction of open source software using object oriented metrics. *Int. J. Innov. Comput. Inf. Control*, 2016, 12(2): 615-634.

[33] Dubey S K, Rana A, Dash Y. Maintainability prediction of object-oriented software system by multilayer perceptron model. *ACM SIGSOFT Softw. Eng. Notes*, 2012, 37(5): 1-4.

[34] Misra S C. Modeling design/coding factors that drive maintainability of software systems. *Softw. Qual. J.*, 2005, 13(3): 297-320.

[35] Jin X, Liu Y, Ren J, Xu A, Bie R. Locality preserving projection on source code metrics for improved software maintainability. In *Proc. the 19th Australasian Joint Conference on Artificial Intelligence*, Dec. 2006, pp.877-886.

[36] Mittal H, Bhatia P. Software maintainability assessment based on fuzzy logic technique. *ACM SIGSOFT Softw. Eng. Notes*, 2009, 34(3): 1-5.

[37] Dahiya S S, Chhabra J K, Kumar S. Use of genetic algorithm for software maintainability metrics' conditioning. In *Proc. the 15th International Conference on Advanced Computing and Communications*, Dec. 2007, pp.87-92.

[38] Pratap A, Chaudhary R, Yadav K. Estimation of software maintainability using fuzzy logic technique. In *Proc. the 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques*, February 2014, pp.486-492.

[39] Sandhya T, Anuradha C. Sequencing of refactoring techniques by Greedy algorithm for maximizing maintainability. In *Proc. the 2016 International Conference on Advances in Computing, Communications and Informatics*, September 2016, pp.1397-1403.

[40] Szöke G, Antal G, Nagy C, Ferenc R, Gyimóthy T. Empirical study on refactoring large-scale industrial systems and its effects on maintainability. *J. Syst. Softw.*, 2017, 129: 107-126.

[41] Hegedüs P, Kádár I, Ferenc R, Gyimóthy T. Empirical evaluation of software maintainability based on a manually validated refactoring dataset. *Inf. Softw. Technol.*, 2018, 95: 313-327.

[42] Kiewkanya M, Jindasawat N, Muenchaisri P. A methodology for constructing maintainability model of object-oriented design. In *Proc. the 4th International Conference on Quality Software,* September 2004, pp.206-213.

[43] Malhotra R. A systematic review of machine learning techniques for software fault prediction. *Appl. Soft Comput.*, 2015, 27: 504-518.

[44] Zhou Y, Leung H. Predicting object-oriented software maintainability using multivariate adaptive regression splines. *J. Syst. Softw.*, 2007, 80(8): 1349-1361.

[45] van Koten C, Gray A R. An application of Bayesian network for predicting object-oriented software maintainability. *Inf. Softw. Technol.*, 2006, 48(1): 59-67.

[46] Hayes J H, Zhao L. Maintainability prediction: A regression analysis of measures of evolving systems. In *Proc. the 21st IEEE International Conference on Software Maintenance*, Sept. 2005, pp.601-604.

[47] Genero M, Manso E, Visaggio A, Canfora G, Piattini M. Building measure-based prediction models for UML class diagram maintainability. *Empir. Softw. Eng.*, 2007, 12(5): 517-549.

[48] Zhou Y, Xu B. Predicting the maintainability of open source software using design metrics. *Wuhan Univ. J. Nat. Sci.*, 2008, 13(1): 14-20.

[49] Rizvi S W A, Khan R A. Maintainability estimation model for object-oriented software in design phase (memood). arXiv:1004.4447, 2010. https://arxiv.org/pdf/1004.4447, Dec. 2019.

[50] Tagoug N. Maintainability assessment in object-oriented system design. In *Proc. the International Conference on Information Technology and e-Services*, March 2012, pp.1-5.

[51] Bakota T, Hegedüs P, Kortvelyesi P, Ferenc R, Gyimóthy T. A probabilistic software quality model. In *Proc. the 27th IEEE International Conference on Software Maintenance*, Sept. 2011, pp.243-252.

[52] Al-Dallal J. Object-oriented class maintainability prediction using internal quality attributes. *Inf. Softw. Technol.*, 2013, 55(11): 2028-2048.

[53] Kumar R, Dhanda N. Maintainability measurement model for object-oriented design. *International Journal of Advanced Research in Computer and Communication Engineering*, 2015, 4(5): 68-71.

[54] Malhotra R, Chug A. A metric suite for predicting software maintainability in data intensive applications. In *Transactions on Engineering Technologies*, Kim H K, Ao S L, Amouzegar M A (eds.), Springer, 2014, pp.161-175.

[55] Misra S, Egoeze F. Framework for maintainability measurement of web application for efficient knowledge-sharing on Campus Intranet. In *Proc. the 14th International Conference on Computational Science and Its Applications*, June 2014, pp.649-662.

[56] Elish M O, Aljamaan H, Ahmad I. Three empirical studies on predicting software maintainability using ensemble methods. *Soft Comput.*, 2015, 19(9): 2511-2524.

[57] Sandhya T, Anuradha C. Predicting maintainability of open source software using Gene Expression Programming and bad smells. In *Proc. the 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)*, Sept. 2016, pp.452-459.

[58] Almugrin S, Albattah W, Melton A. Using indirect coupling metrics to predict package maintainability and testability. *J. Syst. Softw.*, 2016, 121: 298-310.

[59] Kumar L, Rath S K, Sureka A. Empirical analysis on effectiveness of source code metrics for predicting change-proneness. In *Proc. the 10th Innovations in Software Engineering Conference*, February 2017, pp.4-14.

[60] Kanika G, Anuradha C. Evaluation of instance-based feature subset selection algorithm for maintainability prediction. In *Proc. the International Conference on Advances in Computing, Communications and Informatics*, Sept. 2017, pp.1482-1487.

[61] Reddy B R, Ojha A. Performance of maintainability index prediction models: A feature selection based study. *Evol. Syst.*, 2017, 10(2): 179-204.

[62] Kumar L, Santanu K R, Sureka A. Using source code metrics and multivariate adaptive regression splines to predict maintainability of service oriented software. In *Proc. the 18th IEEE International Symposium on High Assurance Systems Engineering*, January 2017, pp.88-95.

[63] Malhotra R, Jangra R. Prediction and assessment of change prone classes using statistical and machine learning techniques. *J. Inf. Process. Syst.*, 2017, 13(4): 778-804.

[64] Bakota T, Hegedüs P, Ladányi G, Körtvélyesi P, Ferenc R, Gyimóthy T. A cost model based on software maintainability. In *Proc. the 28th IEEE International Conference on Software Maintenance*, Sept. 2012, pp.316-325.

[65] Hegedüs P, Bakota T, Ladányi G, FaragóC, Ferenc R. A drill-down approach for measuring maintainability at source code element level. *Electronic Communication of the European Association of Software Science and Technology*, 2013, 60: Article No. 2.

[66] Shibata K, Rinsaka K, Dohi T, Okamura H. Quantifying software maintainability based on a fault-detection/correction model. In *Proc. the 13th Pacific Rim International Symposium on Dependable Computing*, Dec. 2007, pp.35-42.

[67] di Lucca G A, Fasolino A R, Tramontana P, Visaggio C A. Towards the definition of a maintainability model for web applications. In *Proc. the 8th European Conference on Software Maintenance and Reengineering*, March 2004, pp.279-287.

[68] Thwin M M T, Quah T S. Application of neural networks for estimating software maintainability using object-oriented metrics. In *Proc. the 5th International Conference on Software Engineering and Knowledge Engineering*, July 2003, pp.69-73.

[69] Aggarwal K K, Singh Y, Kaur A, Malhotra R. Application of artificial neural network for predicting maintainability using object-oriented metrics. *Int. J. Comput. Electr. Autom. Control Inf. Eng.*, 2008, 2(10): 3552-3556.

[70] Tian Y, Chen C, Zhang C. AODE for source code metrics for improved software maintainability. In *Proc. the 4th International Conference on Semantics, Knowledge and Grid*, Dec. 2008, pp.330-335.

[71] Olatunji S O, Rasheed Z, Sattar K A, Al-Mana A M, Alshayeb M, El-Sebakhy E A. Extreme learning machine as maintainability prediction model for object-oriented software systems. *J. Comput.*, 2010, 2(8): 49-56.

[72] Malhotra R, Chug A. Software maintainability prediction using machine learning algorithms. *Softw. Eng. An Int. J.*, 2012, 2(2): 19-36.

[73] Dash Y, Dubey S K, Rana A. Maintainability prediction of object oriented software system by using artificial neural network approach. *Int. J. Soft Comput. Eng.*, 2012, 2(2): 420-423.

[74] Aljamaan H, Elish M O, Ahmad I. An ensemble of computational intelligence models for software maintenance effort prediction. In *Proc. the 12th International Work-Conference on Artificial Neural Networks*, June 2013, pp.592-603.

[75] Ye F, Zhu X, Wang Y. A new software maintainability evaluation model based on multiple classifiers combination. In *Proc. the 2013 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering*, July 2013, pp.1588-1591.

[76] Ahmed M A, Al-Jamimi H A. Machine learning approaches for predicting software maintainability: A fuzzy-based transparent model. *IET Softw.*, 2013, 7(6): 317-326.

[77] Olatunji S O. Sensitivity-based linear learning method and extreme learning machines compared for software maintainability prediction of object-oriented software systems. *ICTACT J. Soft Comput.*, 2013, 3(3): 514-523.

[78] Malhotra R, Chug A. Application of group method of data handling model for software maintainability prediction using object oriented systems. *Int. J. Syst. Assur. Eng. Manag.*, 2014, 5(2): 165-173.

[79] Kumar L, Rath S K. Neuro — Genetic approach for predicting maintainability using Chidamber and Kemerer software metrics suite. In *Proc. the 11th International Conference on Computing and Information Technology*, July 2015, pp.31-40.

[80] Kumar L, Naik D K, Rath S K. Validating the effectiveness of object-oriented metrics for predicting maintainability. *Procedia Comput. Sci.*, 2015, 57: 798-806.

[81] Jain A, Tarwani S, Chug A. An empirical investigation of evolutionary algorithm for software maintainability prediction. In *Proc. the 2016 IEEE Students' Conference on Electrical, Electronics and Computer Science*, March 2016, pp.1-6.

[82] Jin C, Liu J A. Applications of support vector mathine and unsupervised learning for predicting maintainability using object-oriented metrics. In *Proc. the 2nd International Conference on Multimedia and Information Technology*, April 2010, pp.24-27.

[83] Chandra D. Support vector approach by using radial kernel function for prediction of software maintenance effort on the basis of multivariate approach. *Int. J. Comput. Appl.*, 2012, 51(4): 21-25.

[84] Kumar L, Kumar M, Rath S K. Maintainability prediction of web service using support vector machine with various kernel methods. *Int. J. Syst. Assur. Eng. Manag.*, 2017, 8(2): 205-222.

[85] Kumar L, Kumar S R, Sureka A. Using source code metrics to predict change-prone web services: A case-study on eBay services. In *Proc. the 2017 IEEE Workshop on Machine Learning Techniques for Software Quality Evaluation*, February 2017, pp.1-7.

[86] Elish M O, Elish K O. Application of TreeNet in predicting object-oriented software maintainability: A comparative study. In *Proc. the 13th European Conference on Software Maintenance and Reengineering*, March 2009, pp.69-78.

[87] Cai L, Liu Z, Zhang J, Tong W, Yang G. Evaluating software maintainability using fuzzy entropy theory. In *Proc. the 9th IEEE/ACIS International Conference on Computer and Information Science*, Aug. 2010, pp.737-742.

[88] Dhankhar P, Mittal H, Mittal A. Maintainability prediction for object oriented software. *Int. J. Adv. Eng. Sci.*, 2011, 1(1): 8-11.

[89] Dubey S K, Rana A. A fuzzy approach for evaluation of maintainability of object oriented software system. *Int. J. Comput. Appl.*, 2012, 49(21): 1-6.

[90] Hao X L, Zhu X D, Liu L. Research on software maintainability evaluation based on fuzzy integral. In *Proc. the 2013 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering*, July 2013, pp.1279-1282.

[91] Olatunji S O, Selamat A. Type-2 fuzzy logic based prediction model of object oriented software maintainability. In *Proc. the 13th International Conference on Intelligent Software Methodologies, Tools and Techniques*, Sept. 2015, pp.329-342.

[92] Kundu S, Tyagi K. Maintainability assessment for software by using a hybrid fuzzy multi-criteria analysis approach. *Manag. Sci. Lett.*, 2017, 7(6): 255-274.

[93] Kumar L, Rath S K. Software maintainability prediction using hybrid neural network and fuzzy logic approach with parallel computing concept. *Int. J. Syst. Assur. Eng. Manag.*, 2017, 8(S2): 1487-1502.

[94] Yenduri G, Madhwaraj G. An authoritative method using fuzzy logic to evaluate maintainability index and utilizability of software. *Adv. Model. Anal. B*, 2017, 60(3): 566-580.

[95] Yu H, Peng G, Liu W. An application of case based reasoning to predict structure maintainability. In *Proc. the 2009 International Conference on Computational Intelligence and Software Engineering*, Dec. 2009.

[96] Mehra A, Dubey S K. Maintainability evaluation of object-oriented software system using clustering techniques. *Int. J. Comput. Technol.*, 2013, 5(2): 136-143.

[97] Lee C C, Chung P C, Tsai J R, Chang C I. Robust radial basis function neural networks. *IEEE Trans. Syst. Man, Cybern. Part B*, 1999, 29(6): 674-685.

[98] Murphy K P. Naive Bayes classifiers. https://www.cs.ubc.ca/∼murphyk/Teaching/CS340-Fall06/reading/NB.pdf, Dec. 2019.

[99] Hegedüs P, Ladányi G, Siket I, Ferenc R. Towards building method level maintainability models based on expert evaluations. In *Proc. the International Conferences on Computer Applications for Software Engineering, Disaster Recovery, and Business Continuity*, Nov. 2012, pp.146-154.

[100] Conte S D, Dunsmore H E, Shen Y E. Software Engineering Metrics and Models. Benjamin-Cummings Publishing Co., 1986.

[101] Port D, Korte M. Comparative studies of the model evaluation criterions MMRE and PRED in software cost estimation research. In *Proc. the 2nd ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, October 2008, pp.51-60.

[102] MacDonell S G. Establishing relationships between specification size and software process effort in CASE environments. *Inf. Softw. Technol.*, 1997, 39(1): 35-45.

[103] Li W, Henry S. Object-oriented metrics that predict maintainability. *J. Syst. Softw.*, 1993, 23(2): 111-122.

[104] Elmidaoui S, Cheikhi L, Idri A. Accuracy comparison of empirical studies on software product maintainability prediction. In *Proc. the World Conference on Information Systems and Technologies*, March 2018, pp.26-35.

[105] Shepperd M, Kadoda G. Comparing software prediction techniques using simulation. *IEEE Trans. Softw. Eng.*, 2001, 27(11): 1014-1022.

[106] Briand L C, Brasili V R, Hetmanski C J. Developing interpretable models with optimized set reduction for identifying high-risk software components. *IEEE Trans. Softw. Eng.*, 1993, 19(11): 1028-1044.

[107] Elmidaoui S, Cheikhi L, Idri A, Abran A. Empirical studies on software product maintainability prediction: A systematic mapping and review. *e-Informatica Softw. Eng. J.*, 2019, 13(1): 141-202.

[108] Keung J W. Theoretical maximum prediction accuracy for analogy-based software cost estimation. In *Proc. the 15th Asia-Pacific Software Engineering Conference*, Dec. 2008, pp.495-502.

Sara Elmidaoui got her Bachelor's degree in mathematics and computer science in 2012 at Faculty of Sciences, University Ibn Zoher in Agadir, Morocco, and her Master's degree in engineering design and application development from Faculty of Sciences and Technologies, University Hassan 1, Settat, Morocco, in 2014. Sara is currently a Ph.D. student at École Nationale Supérieure d'Informatique et d'Analyse des Systeèmes (ENSIAS), Mohammed V University in Rabat, Morocco.

Laila Cheikhi is a professor at École Nationale Supérieure d'Informatique et d'Analyse des Systémes (ENSIAS), Mohammed V University in Rabat, Morocco. She received her M.Sc. degree in 2004 and Ph.D. degree in 2008 both in software engineering from École de Technologie Supérieure (ETS) Montréal. She has over eight years of experience in computer engineering at the Ministry of Finance of Morocco. Her research interests include software quality models, software metrics, software engineering ISO standards, software product and process quality, software engineering principles and data analysis.

Ali Idri is a professor at Computer Science and Systems Analysis School (École Nationale Supérieure d'Informatique et d'Analyse des Systèmes (ENSIAS), Mohammed V University in Rabat, Morocco). He received DEA (Master) (1994) and Doctorate of 3rd Cycle (1997) degrees in computer science, both from Mohammed V University in Rabat. He has received his Ph.D. degree (2003) in cognitive computer sciences from University of Quebec at Montreal. He is the head of the Software Project Management research team since 2010. He is the chairman of the 10th International Conference in Intelligent Systems: Theories and Application (SITA 2015) and he serves as a member of program committee of major international journals and conferences. His research interests include software effort/cost estimation, software metrics, software quality, computational intelligence in software engineering, data mining, e-health. He has published more than 90 papers in several international journals and conferences.

Alain Abran is a professor and the director of the Software Engineering Research Laboratory at the École de Technologie Supérieure (ETS). He is currently co-executive editor of the Guide to the Software Engineering Body of Knowledge project. He is also actively involved in software engineering standards as the international secretary for ISO/IEC JTC1 SC7—Software and System Engineering; he is also co-chair of the Common Software Measurement International Consortium (COSMIC). Dr. Abran has more than 20 years of industry experience in information systems development and software engineering. Dr. Abran holds his Ph.D. degree in electrical and computer engineering (1994) from École Polytechnique de Montréal (Canada) and Master's degrees in management sciences (1974) and electrical engineering (1975) from University of Ottawa. His research interests include software productivity and estimation models, software engineering foundations, software quality, software functional size measurement, software risk management and software maintenance management.

Appendix

The appendix is available at: https://github.com/-SarahElmidaoui/Machine-Learning-Techniques-for-Software-Maintainability-Prediction-Accuracy-Analysis-Appendix/blob/master/9668%20ID-Rx-source-Appendix.pdf.