IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 20 August 2023, accepted 31 August 2023, date of publication 7 September 2023, date of current version 15 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3312716

TOPICAL REVIEW

Software Cost and Effort Estimation: Current Approaches and Future Trends

CHAUDHARY HAMZA RASHID [1], IMRAN SHAFI [2], JAMIL AHMAD [1], (Senior Member, IEEE), ERNESTO BAUTISTA THOMPSON [3,4,5], MANUEL MASIAS VERGARA [3,6,7], ISABEL DE LA TORRE DIEZ [8], AND IMRAN ASHRAF [9]

[1]Department of Computer Science, Abasyn University Islamabad Campus, Islamabad 44000, Pakistan
[2]College of Electrical and Mechanical Engineering, National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan
[3]Universidad Europea del Atlántico, 39011 Santander, Spain
[4]Universidad Internacional Iberoamericana, Campeche 24560, Mexico
[5]Universidade Internacional do Cuanza, Kuito, Bié, Angola
[6]Universidad Internacional Iberoamericana Arecibo, Sector Palaches 00613, Puerto Rico
[7]Universidad de La Romana, La Romana, Dominicana
[8]Department of Signal Theory, Communications and Telematics Engineering, University of Valladolid, 47011 Valladolid, Spain
[9]Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, South Korea

Corresponding authors: Imran Ashraf (ashrafimran@live.com) and Isabel de la Torre Diez (isator@tel.uva.es)

This work was supported by the European University of Atlantic.

ABSTRACT Software cost and effort estimation is one of the most significant tasks in the area of software engineering. Research conducted in this field has been evolving with new techniques that necessitate periodic comparative analyses. Software project success largely depends on accurate software cost estimation as it gives an idea of the challenges and risks involved in the development. The great diversity of ML and Non-ML techniques has generated a comparison and progressed into the integration of these techniques. Based on varying advantages it has become imperative to work out preferred estimation techniques to improve the project development process. This study aims to present a systematic literature review (SLR) to investigate the trends of the articles published in the recent one and a half decades and to propose a way forward. This systematic literature review has proposed a three-stage approach to plan (Tollgate approach), conduct (Likert type scale), and report the results from five renowned digital libraries. For the selected 52 articles, artificial neural network model (ANN) and constructive cost model (COCOMO) based approaches have been the favored techniques. The mean magnitude of relative error (MMRE) has been the preferred accuracy metric, software engineering, and project management are the most relevant fields, and the promise repository has been identified as the widely accessed database. This review is likely to be of value for the development, cost, and effort estimations.

INDEX TERMS Software cost estimation, systematic literature review, tollgate approach, Likert scale, quality assessment, software dependability, project planning.

I. INTRODUCTION

Software cost estimation is a difficult process that takes into account a variety of elements, including the scope of the project, the difficulty of the requirements, the team's experience, and the technologies being employed [1]. Although a lot of studies have been conducted in the area of software cost and effort estimation, still cost and effort overruns are a concern for software industries. Wrong estimations can lead to financial and time losses. There has been a lack of research that evaluates techniques that have evolved in the last decade and a half. The lack has generated a requirement to do a systematic literature review (SLR) of the published research on software cost and effort estimation during this period.

Accuracy is frequently a problem with evolving machine learning (ML) and Non-ML techniques, particularly when working on complex projects or projects with shifting requirements. For projects to be completed on schedule and

The associate editor coordinating the review of this manuscript and approving it for publication was Sangsoon Lim.

TABLE 1. Pros and cons of cost estimation techniques discussed in the literature.

Technique	Pros	Cons
COCOCMO	The drivers are very helpful to segregate and calculate the impact on different factors that affect the development.	Does not sufficiently involve the requirements' volatility and documentation.
FPA	Ability to assess the size of an application to evaluate its ability for meeting user requirements.	Generally, effort models are based on lines of code hence FPA needs to be converted.
Regression Analysis	Exceptionally good results for linearly separable data.	Prone to noise and overfitting.
Group Expert Judgment	Group Estimation gives better accuracy than individual estimates	Group estimates vary and depend highly on the expertise, bias, and experience of the group.
ANN	These are good to model with nonlinear data and a large number of inputs.	On smaller data set these models tend to overfit.
Blockchain	Increased transparency and speed.	High implementation costs.
Fuzzy Logic	Control is based on the actual output of the system, hence a flexible control loop.	Handling of imprecise data and inherent human interference.
Analogy Based Approaches	Estimation will be accurate if an analogous project is available. Useful when the data is small.	Difficult to find a near analogous project.
Ensemble	Empirically, a standard ensemble reduces the bias leading to better outcomes.	Expensive in terms of time and space with low interpretability and prediction.

within budget, accurate cost estimation has vital important and many organizations invest a lot in estimation to ensure timely development and customer satisfaction [2]. Considering this, it is important to identify accurate estimation techniques to address the cost and effort overrun issues.

This review aims to provide a comprehensive overview of the current state of research on software cost estimation and identify opportunities for future researchers to improve the accuracy and effectiveness of cost estimation methods. To compare the performance of different methods and evaluate their effectiveness, the SLR is conducted to identify the

1) Current trends of research in this field,
2) Accurate measures for cost and effort estimation,
3) Impact on industry and application fields,
4) Commonly accessed databases and,
5) Current gaps and future work.

Non-ML Methods evaluate a project's cost using a set of established formulas and parameters. ML methods, on the other hand, offer the potential to improve the accuracy of software cost estimation by learning from historical data and making predictions based on patterns [3]. These methods take into account a wide range of factors including project cost, project size, team experience, and technology stack, and can adapt to changing project requirements. The application of ML techniques in software cost estimation is a relatively new field, and there is room for more research to improve these methods and algorithms for different types of software projects [4]. In this study, the literature on software cost estimation using ML and Non-ML methods has been reviewed to discuss the challenges and limitations of these methods, such as the use of historical datasets and the difficulty of interpreting the results [4], [5].

In this systematic literature review, we review the existing literature on software cost estimation using ML and Non-ML methods including studies to compare the performance of different methods and evaluate their effectiveness in different contexts. We also discuss the challenges and limitations of these methods, such as the need for large amounts of historical data and the difficulty of interpreting the results [4], [5]. Ultimately, this review has aimed to provide a comprehensive overview of the current state of research on software cost estimation and identify opportunities for future researchers to improve the accuracy and effectiveness of cost estimation methods. We review articles published between 2009 and 2023 have been reviewed by prominent digital libraries [6], [7] including IEEE, Hindawi, Elsevier, ACM, and Web of Science. Inclusion and exclusion criteria based on period, language, source, impact, accessibility, and relevance to select relevant articles for the review have been considered.

A. TECHNOLOGICAL QUERIES

- RQ-1. What are the most common ML and Non-ML techniques for software cost estimation, and how do they compare to other cost estimation methods?
- RQ-2. What factors influence the accuracy of software cost estimation using ML methods, and how can these factors be optimized to improve cost estimation accuracy?
- RQ-3. Which organizations and industries have benefitted the most from the selected articles?
- RQ-4.What are the most commonly accessed repositories and datasets in these studies?

The following is the structure of the preceding paper: The existing literature on software cost estimation methods is discussed in Section II. The proposed methodology is discussed in Section III. Section IV presents the results. In the end, Section V concludes this study and provides future work.

II. LITERATURE REVIEW

Software effort estimation methods can generally be grouped into three broad categories [8]. The pros and cons of models from each of these categories have been described in Table 1.

A. EXPERT JUDGMENT-BASED APPROACHES

These methods rely on the skills and in-depth knowledge of seasoned personnel, like project managers or subject-matter specialists, to calculate the time needed for software development [9]. Delphi, analogous estimation, expert opinion poll, and parametric estimation are common expert judgment techniques. The responses are collected and summarized, without revealing the identities of the experts [10]. Analogous estimation uses historical data and leverages past project knowledge for new projects [11]. The expert opinion poll aims to arrive at a consensus by considering the knowledge of the experts [12].

B. ALGORITHMIC-BASED APPROACHES

These use mathematical models and algorithms to estimate software effort and often employ statistical techniques, machine learning algorithms, or parametric models to generate estimates based on the available data [13]. COCOMO, Function Point Analysis (FPA), and software estimation by analogy are examples of common algorithmic methodologies. COCOMO forecasts the effort, time, and expense needed for software development [14]. With increasing levels of accuracy and sophistication, COCOMO has developed into several variants, including Basic COCOMO, Intermediate COCOMO, and Detailed COCOMO [15]. A thorough summary of recent work on FPA is given in [16].

C. COMPUTATIONAL INTELLIGENCE

These methods have been used to increase the precision of cost and effort estimation [17]. The choice between these strategies depends on several variables, including the accessibility of data, the complexity of the project, and the expertise that is available inside the organization. Computational intelligence software cost estimation with Artificial Neural Networks (ANN) [18] leverages the power of machine learning to predict project effort and cost. They can handle non-linear relationships and adapt to different project contexts [19]. By analyzing historical data and identifying patterns, regression models can provide predictions and insights into the expected cost of software development projects [20].

Software cost and effort estimation has been studied by many researchers and numerous studies have been presented on this subject. A few of the articles have given an outline of the estimation approaches and not carried out an in-depth analysis [21], [22] while many have presented comprehensive research to give substantial outcomes and contributed to the body of knowledge [23], [24], [25]. Starting in the 1990s research in this area began with analogy-based approaches including case-based reasoning (CBR) [26]. Subsequently, considerable work has been done on versions of COCOMO by Boehm et al. [27]. FPA has also received attention during the same period [28]. In the recent past, ML-based methods have been used more extensively in the literature [29], [30], [31]. This comprises techniques such as regression [32], [33], [34], ANN [13], [35], [36], [37], Ensemble [31], [38], [39], and Blockchain [40]. The specific focus of research in all these methods has been on the optimization of errors [41], [42], [43], [44], [45].

Recently, blockchain has emerged as a new technology with its application in the field of software engineering [46], which includes its use in the improvement of software processes. Certain software development issues have also received researchers' attention like the improvement of software development processes by using Blockchain techniques [47]. This is an emerging field with research potential. Ahmed et al. [40] conducted an SLR using a Blockchain-Based Software Effort Estimation methodology on the expert opinions of 52 organizations and the results of the case study conclude that a lack of historical data, experts, and biases in a group prevents the organizations to perform estimation activities more effectively. It also suggests that the blockchain method estimation is more efficient as compared to traditional methods.

An SLR performed by Khan et al. [13] shows that ML method application in software effort estimation has increased since the 2000s and, the use of Non-ML methods has been less common for acquiring optimized results. While; two subsequent SLRs [48], [49] on the subject have analyzed the ML and Non-ML methods to compare the accuracy of both methods. Both these SLRs suggest that ML methods have better performance overall than Non-ML methods which are in lesser numbers. Ali and Gravino [24] in an SLR on ML methods concluded similar results that ML methods outperform Non-ML methods. While they have further segregated the techniques as ANN and SVM performed better than other ML approaches.

Jadhav et al. [23] indicated detailed research on 1015 articles in the past five decades and concluded that ANN, fuzzy logic, regression, analogy, and COCOMO are the most prevalent methods followed by use case point (UCP) and FPA. Their results have been confirmed by relating with published review work and found that the results were consistent.

Fernández-Diego et al. [25] updated the Usman et al. study [50] conducted on 73 articles from 2014 to. They highlighted that accuracy was a challenge in most of the articles. Although several articles showed satisfactory accuracy values, still some researchers continued to report inadequate results. While the cost and effort estimations are mainly considered in software domains, some applications in practical manufacturing and other fields are equally applicable and have proved beneficial. Huynh et al. [51] proposed a poly algorithm with fuzzy logic system, Grey-Taguchi method, and adaptive neuro-fuzzy inference system (ANFIS) for the estimation of parameters that affect the cost of design variables on the magnification ratio of compliant mechanisms of motion scope.

Similarly, research by The Ho et al. [52] proposes a methodology to detect "chatter" by using a multi-input convolutional neural network (CNN) via image and sound signals to classify data and to determine whether the mechanical machining is stable or vibrational. Research in corresponding fields is expanding as evident from diverse research including performance analysis of fuzzy c means clustering based ANFIS and Elman ANN in effort and cost estimation by Yang et al. [53], the establishment of a link between manufacturing and economic variables by cost estimation in

TABLE 2. Strengths and limitations of research articles discussed in the literature review.

Ref.	Strengths	Weaknesses
[11]	• A very well-referred article with more than 300 highly cited articles. • Each RQ has been justified with research gap and motivation which has supported the concept of research.	• Articles surveyed have been identified using expert opinion only, rather than applying string search or data mining. • Results have not been supported with figures which hamper the understandability of readers.
[23]	• The SLR is of 5 decades which gives a comprehensive bibliometric overview. • Strengths and weaknesses of different techniques have been given due importance in the article which is helpful for researchers.	• Only one database has been searched for articles that do not provide a sufficient variety of articles. • Some figures and tables are not arranged next to the related text hence causing confusion.
[24]	• The comparative analysis of each ML and Non-ML technique has been carried out in-depth to highlight the priorities of researchers. • Prediction accuracies of models and use in different timeframes have been indicated with recommendations for the research community.	• Bio-inspired feature selection algorithms have been included but have not been addressed sufficiently to add to the analysis. • Better figures could have been added to improve the understandability level of the SLR.
[27]	• Activities covered under various techniques have been defined elaborately in tabulated form. • It is a comprehensive guide for a general reader.	• The paper lacks a comparative analysis of the techniques surveyed. • Review strategy, paper selection criteria, and recommendations section are not available.
[50]	• Search strategy was applied to eight different libraries to shortlist articles which added to the diversity of the conclusion. • Specific description of keywords from primary studies along with references is a good addition that would help researchers to search the article easily.	• Cross-company data articles have not been investigated in the SLR which restricted the discussion in this domain. • None of the results has been supported with a graphical representation which reduces the understandability of a general reader.
[25]	• While updates of SLRs are very common in medicine, few have been found in software engineering and this study is a positive addition in this context. • Accuracy summary statistics of the SLR have been very elaborately defined which can assist future work on the subject.	• 22 cost factors mentioned in the SLR have not been explained and the selection criteria of the factors is not clear. • Paper employing the use of multiple domain datasets has not been included which restricts the wide applicability of results.
[40]	• Complete evaluation of current estimation techniques being followed at organizational and sub-organizational levels has been done. • Clear comparison of good and bad practices being followed for software estimation with a way forward to address the issue.	• Results were dependent on data provided by organizations that are prone to bias. • Study included expert opinions whose performance may vary depending on the level of experience.
[48]	• Methods understudy have been explicitly explained in a detailed manner. • The analysis table has been developed in a very comprehensible form.	• The number of studies included in the review is too small for accruing useful trends and outcomes. • Research methodology has not been clearly defined.
[49]	• The paper qualifies several ML models based on data mining techniques for a suitable choice. • Detailed results of each technique have been clearly defined and displayed in a comparative table for better comprehension.	• Literature review has not been included in the study. • Results are not well supported by illustrations and one of the figures is blurred.

mechanical production by H'mida et al. [54] and project cost estimation of 415 Chinese expressways using CNN algorithm by Xue et al. [55] are to name a few. The field is expanding and is likely to earn dividends in software, manufacturing, and construction domains. Table 2 gives a brief overview of the strengths and weaknesses of each of the articles included in this SLR.

III. METHODOLOGY

Following an authentic protocol is imperative in any SLR to minimize biases in the research. This SLR follows the review protocol suggested by Kitchenham et al. [56] as depicted in the flowchart, Figure 1. The review has been divided into three main stages including planning, execution, and analysis, as per the protocol.

A. SYSTEM OVERVIEW

As per Kitchenham's protocol, Research Questions (RQs) have been designed in the first stage i.e. the planning stage. To address the RQs, research methodology has been adopted in the second stage to identify related articles based on the RQs of the first stage. For the unbiased formulation of the review, search strings have been defined to search relevant articles. Relevant research articles were accessed from renowned literature sources. Inclusion and exclusion criteria were defined to decide the articles to be included or excluded. Then, the quality assessment matrix was defined to assess the quality of each article individually. Subsequently, in the third stage, relevant information was extracted from the selected articles based on RQs and kept in the data extraction table. Finally, the data was analyzed to accrue meaningful results.

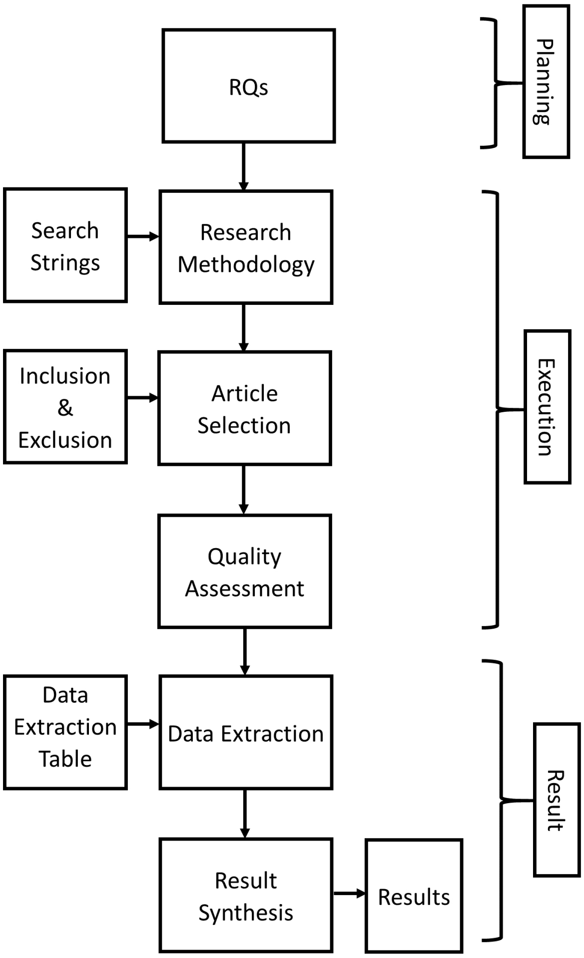

FIGURE 1. Flowchart of SLR protocol.

B. FRAMEWORK

his SLR follows a structured framework to ensure a comprehensive and rigorous analysis of articles sequentially incorporating formulation of the RQs, developing a search strategy, literature search, screening, selection, assessment, data extraction, synthesis, results reporting, discussions, and conclusion [57], [58]. These stages have been categorized into the following:

1) Research Questions
2) Research Methodology
3) Articles Selection
4) Quality Assessment
5) Data Extraction
6) Result and Discussions

The stages have been elaborated graphically in the flowchart as mentioned in the sub-section overview.

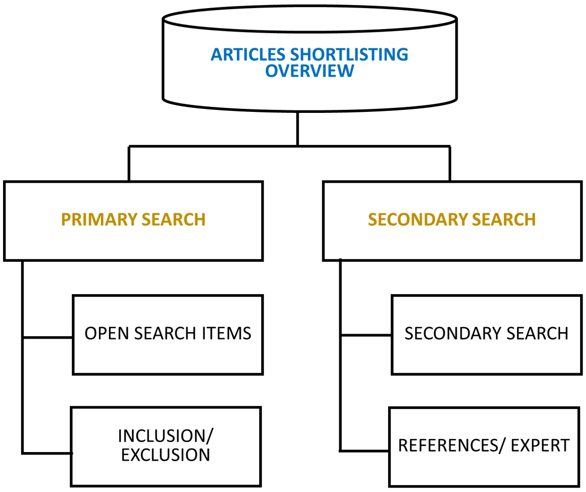

FIGURE 2. Overview of article selection.

C. DATABASES

The databases that are searched to obtain the references for the articles on software cost estimation are as follows

1) IEEE Xplore
2) Springer
3) ScienceDirect
4) ACM Digital Library
5) Hindawi

These databases have been selected based on their reputation for hosting high-quality scientific literature in the field of computer science, including software engineering and artificial intelligence [6], [7]. Furthermore, the inclusion and exclusion criteria have been applied to ensure that the articles selected for review meet specific quality and relevance standards. The databases have been accessed in two stages namely, primary and secondary as shown in Figure 2.

D. ARTICLE SEARCH

As done by alike systematic literature reviews, erudite search terms have been included by employing substitute terms and synonyms of related terms using the Boolean operator OR and joining the main terms via the Boolean operator AND [28]. The use of these Boolean strings has permitted to get almost all the studies obtainable in the databases and for any missing studies, selected studies can be used. We used query strings on the keywords related to the research topic and Boolean operators to search articles with the relevant title options. Below are the query strings used in each database

IEEE Xplore Digital Library

("All Metadata": Software effort estimation) OR ("All Metadata":Software cost estimation) OR ("All Metadata":Software project estimation) AND ("All Metadata":COCOMO) OR ("All Metadata Nonlogarithmic) AND ("All Metadata":MACHINE LEARNING) OR ("All Metadata":ANN) OR ("All Metadata":ARTIFICIAL NEURAL NETWORK) OR ("All Metadata":SURVEY VECTOR) OR ("All Metadata":DECISION TREE) OR ("All Metadata":FUZZY LOGIC)

Filters Applied: 2009 ' 2023

Springer

"software cost estimation OR software effort estimation OR software project estimation AND machine learning OR artificial intelligence OR neural network OR support vector regression OR fuzzy logic OR decision tree OR genetic algorithm OR swarm intelligence OR ensemble learning AND COCOMO OR SLIM OR Price-S OR Putnam Model AND empirical study OR systematic AND literature AND review OR survey AND data extraction OR data analysis"

Filter: within 2009 - 2023

Science Direct

"software effort estimation OR software project estimation AND machine learning OR neural network OR ensemble learning AND COCOMO OR Non algorithmic AND empirical study OR systematic literature"

Filters: "Computer Science AND Research Articles AND Open Access OR Open Archive AND 2009 till 2023"

ACM

"query": (software engineering OR software development) AND (cost estimation OR effort estimation OR size estimation OR software metrics) AND (neural network OR support vector regression OR fuzzy logic OR decision tree OR genetic algorithm)

"Filter": E-Publication Date: (12/01/2009 TO 05/31/2023)

Hindawi

(software engineering OR software development) AND (cost estimation OR effort estimation OR size estimation OR software metrics) AND (ANN OR support vector regression OR fuzzy logic OR decision tree OR genetic algorithm)

(Filters: Abstract only AND Research Articles AND between 2009 ' 2023)

These query strings include keywords related to software cost estimation, machine learning algorithms, software cost estimation frameworks, research methods, data extraction, and analysis. The studies we focused on post-2009 to date.

E. ARTICLE SHORTLISTING

The preliminary web search on the five databases brought out a cumulative 1,044,771 articles from various diverse fields. However, based on the article's titles' relevance and the initial identification brought the figure to 1295. An overview is shown in Figure 3. Here, the primary research phase starts with the tollgate approach [46]. The tollgate approach is a project management methodology that involves the use of predefined checkpoints or "tollgates" at various stages of a project. It provides a structured framework for monitoring and controlling projects, ensuring alignment with strategic objectives, managing risks, and enabling effective decision-making.

Further screening with the abstracts and duplication removal brought the total to 290 articles. Application of the inclusion and exclusion criteria in the next stage brought the figure to 47 articles. These steps have been mentioned as the primary search. Then with expert opinion and snowballing [59] the selected studies 10 more relevant articles were identified in the secondary search phase. Hence, based on primary and secondary searches, the total number of shortlisted articles for the SLR was brought to 57. Detail of the tollgate approach is tabulated in Table 3. Subsequently, five articles were removed from the quality assessment (QA) and the final number of studies was set to 52, which has been discussed in the quality assessment sub-section.

TABLE 3. Article shortlisting using "Tollgate approach."

Open search	Library	IEEE	Springer	Science direct	ACM	Hindawi	Total
	Preliminary Search	1670	1691	15,801	1,022,099	3450	1,044,711
Primary	Identification	210	315	223	373	174	1295
	Screening	53	85	47	67	38	290
	Inclusion /Exclusion	9	12	8	9	9	47
Secondary	Expert/Snowballing	10	15	12	11	9	57

F. INCLUSION CRITERIA

The inclusion criteria are based on various factors [60] for selecting the articles.

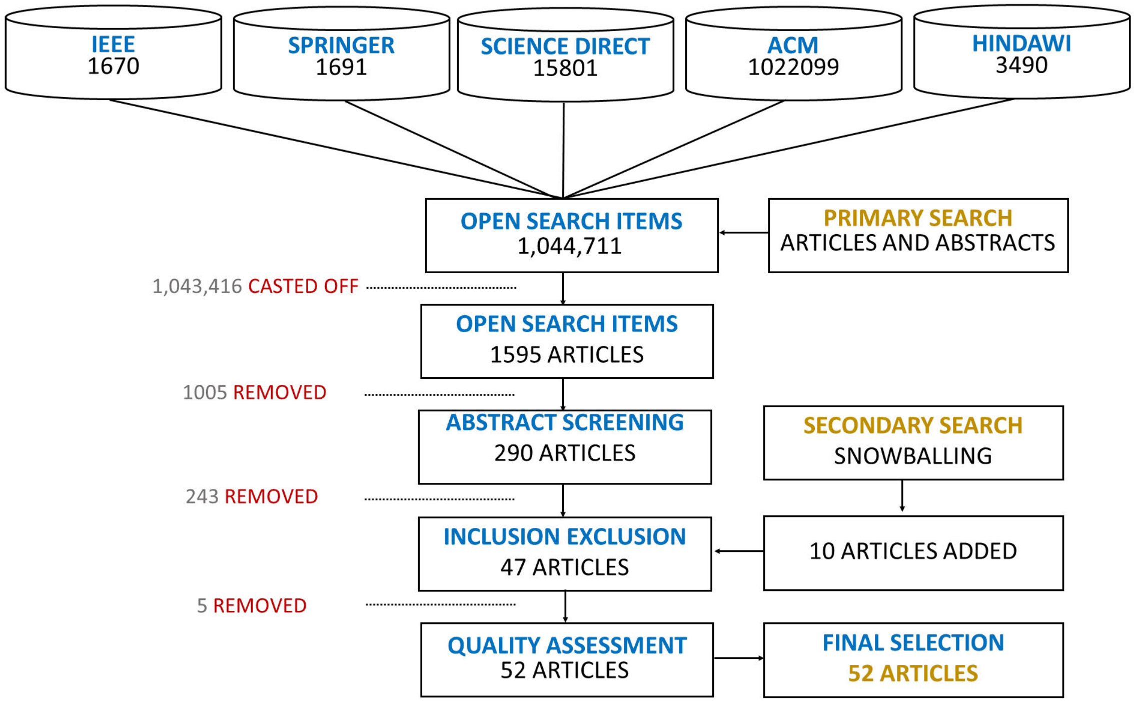

FIGURE 3. Tollgate approach.

1) **Period**: Articles published between the years 2009 and 2023 have been considered for this systematic literature review.
2) **Language**: Only articles published in the English language are included in the review.
3) **Source**: Articles from reputable digital libraries such as IEEE, Springer, Science Direct, ACM, and Hindawi are included in the review.
4) **Impact**: Only articles with a high impact factor, and relevance to the topic are included in the review.
5) **Accessibility**: Easily accessible articles available in full-text format have been included in the review.
6) **Relevance**: Articles that directly address the research question and meet the inclusion criteria are included in the review. Studies that are not related to the research question or those that do not meet the inclusion criteria have not been included in the review [50].

G. EXCLUSION CRITERIA

The exclusion criteria used in this systematic literature review [61] are as follows.

1) Articles that are not related to software cost estimation using mentioned methods have been excluded from the review.
2) Articles that are published in languages other than English are excluded.
3) Duplicate articles are removed from the list.
4) Articles that are not peer-reviewed or are not published in reputable journals or conference proceedings are excluded.
5) Articles that are not accessible in full-text format are excluded.
6) Articles that do not meet the inclusion criteria are excluded.

Overall, the goal of the inclusion and exclusion criteria is to ensure that only relevant and high-quality articles are included [57] in the systematic literature review to address the research question effectively.

H. DATA SIFTING

On completion of the planning stage of the systematic literature review data sifting or sorting is done to stage 2, i.e. execution. In data sifting all the data is organized as per the libraries and it is ensured that all articles are complete. Before shifting to the quality assessment, the data is preprocessed through Zotero for further organizing as per requirements.

Zotero is a free and open-source reference management program that allows you to save and organize articles, annotate them, and generate bibliographies [58]. It is a popular tool among academics and scholars because it makes managing research materials simple and efficient. Users can import articles directly from databases, websites, and other sources, and Zotero automatically extracts important information such as author names, publication titles, and publication dates. Users can also manually enter information for items that cannot be imported automatically. It also has several features to aid in the organization and management of research materials, such as the ability to create collections and sub-collections to group relevant articles, add tags and notes for simple searching and reference, and produce bibliographies in a variety of formats.

I. QUALITY ASSESSMENT

The quality assessment (QA) of each article has been carried out to assess the reporting, rigor, credibility, and relevance [62]. The Likert-type scale has been employed for the assessment which is a type of survey response scale [63]. The scale typically consists of a series of statements or items that respondents are asked to rate on a scale. The Likert-type scale is widely used in scientific research and is particularly useful in studies where the researcher wants to quantify opinions about a particular topic. The scale is easy to administer and analyze, and it allows for a range of responses, making it more sensitive than a simple binary response scale [64].

The Likert-type scale was used to evaluate each study independently by two researchers as per the numerical assessment scale based on the Reporting, Rigor, Credibility, and Relevance [65] with each corresponding to a different set of questions that have been covered in the subsequent paragraphs. To mitigate the potential biases, the total scores of the two independent reviewers were averaged to grade each article. The procedure added to the robustness of comparative analysis as it offered a range of grades instead of a yes or no. A further explanation of the scale distribution has been described below:

1) REPORTING

The extent to which the article clearly and thoroughly reports the research methods and findings [66]. Articles scoring 40, 30, 20 and 10 have been considered as excellent, good, fair and poor respectively. Each of the articles have been graded as per criteria below. (maximum score: 40).

1) Clarity and completeness of the article's title and abstract (10 points)
2) Appropriateness and transparency of the methodology used (10 points)
3) Thoroughness and accuracy of data collection and analysis (10 points)
4) Adequacy and clarity of results and conclusions (10 points)

2) RIGOR

The extent to which the article has a strong and appropriate research design, methodology, and analysis [67]. Articles scoring 30, 22.5, 15, and 7.5 have been considered as excellent, good, fair, and poor respectively. Each of the articles has been graded as per the criteria below. (maximum score: 30).

1) Robustness and transparency of the statistical methods used (10 points)
2) Validity and reliability of the measures used (10 points)
3) Adequacy of sample size and representativeness (10 points)

3) CREDIBILITY

The extent to which the article is trustworthy and credible in its conclusions [68]. Articles scoring 20, 15, 10, and 5 have been considered as excellent, good, fair, and poor respectively. Each of the articles has been graded as per the criteria below. (maximum score: 20).

1) Relevance and credibility of the sources cited (10 points)
2) Appropriate and ethical handling of research data (10 points)

4) RELEVANCE

The extent to which the article is relevant and applicable to the research question and context [69]. Articles scoring 10, 7.5, 5, and 2.5 have been considered as excellent, good, fair, and poor respectively. Each of the articles has been graded as per the criteria below. (maximum score: 10).

1) Direct and significant relevance to the research question or topic (10 points)

These points have been assessed on a scale of four, which means that 25% points for each step i.e. poor, fair, good, and excellent. None of the assessments has been put equal to zero as articles selected on the inclusion and exclusion criteria were already well scrutinized. 52 articles (having more than 65% points-the cut-off score) qualified the assessment criteria out of a total of 57 and have been included for analysis. Each article has been assessed by two independent reviewers, and their scores have been averaged to obtain a final score for each category. The scores have been used to determine the overall quality of the article and its contribution to the systematic literature review.

J. QUALITY ASSESSMENT ANALYSIS

Yang et. al carried out an SLR on 241 studies between 2004 and 2018 to assess the QA criteria and suggested that the extent to which articles report findings, the rigor of methodology, credibility, and relevance of analysis define the quality assessment's reliability [70]. Similarly, these criteria have been followed in earlier studies [65], [66], [67], [68], [69]. By following these established protocols, biases can be reduced to produce credible and transparent results and internal and external validities can be improved [24].

Table 4 contains the quality assessment scores of selected articles as per mentioned protocol [101]. Studies have not been assigned binary scales of 0 and 1 scores, instead assessed on a sliding scale to give fair weightage to the research carried out by the researchers. The QA of 52 articles have been summarized in Table 4. The above calculation is based on a Likert-type scale with the definitions that an article would be called excellent if it fully meets or exceeds criteria (100%), it would be good if it meets criteria to a great extent (75%), it would be fair if it meets criteria to some extent, but with

TABLE 4. Quality assessment matrix of selected articles.

Reference	Clarity	Appropriateness	Thoroughness	Adequacy	**Reporting (40)**	Robustness	Validity	Sample size	**Rigor (30)**	Relevance	Ethical handling	**Credibility (20)**	Relevance	**Relevance (10)**	Total (100)
S-1 [29]	10	7.5	10	7.5	35	10	7.5	10	27.5	10	10	20	10	10	92.5
S-2 [71]	7.5	2.5	10	7.5	27.5	7.5	7.5	10	25	7.5	7.5	15	10	10	77.5
S-3 [25]	10	10	10	7.5	37.5	10	7.5	10	27.5	10	10	20	10	10	95
S-4 [30]	7.5	10	7.5	7.5	32.5	5	10	7.5	22.5	10	10	20	10	10	85
S-5 [40]	7.5	5	5	7.5	25	5	5	7.5	17.5	7.5	10	17.5	10	10	70
S-6 [72]	7.5	10	5	7.5	30	5	5	7.5	17.5	2.5	10	12.5	10	10	70
S-7 [73]	5	5	5	7.5	22.5	7.5	5	10	22.5	7.5	10	17.5	5	5	67.5
S-8 [31]	10	10	7.5	7.5	35	10	10	7.5	27.5	10	10	20	10	10	92.5
S-9 [13]	10	5	7.5	7.5	30	7.5	7.5	10	17.5	7.5	10	17.5	5	5	70
S-10 [74]	5	5	10	7.5	27.5	7.5	10	10	27.5	7.5	10	17.5	5	5	77.5
S-11 [35]	7.5	5	5	7.5	25	5	5	7.5	17.5	7.5	10	17.5	10	10	70
S-12 [38]	10	10	7.5	7.5	35	5	5	7.5	17.5	5	10	15	10	10	77.5
S-13 [75]	7.5	5	5	10	27.5	10	2.5	10	22.5	5	10	15	7.5	7.5	72.5
S-14 [76]	2.5	5	10	7.5	25	2.5	7.5	10	20	7.5	7.5	15	10	10	70
S-15 [77]	10	5	7.5	7.5	30	5	7.5	5	17.5	10	7.5	17.5	10	10	75
S-16 [36]	10	5	7.5	7.5	30	5	5	7.5	17.5	5	5	10	10	10	67.5
S-17 [78]	10	10	7.5	7.5	35	5	10	7.5	22.5	10	10	20	10	10	87.5
S-18 [32]	5	5	7.5	7.5	25	7.5	5	10	22.5	7.5	10	17.5	5	5	70
S-19 [72]	7.5	5	5	10	27.5	2.5	10	10	22.5	5	10	15	7.5	7.5	72.5
S-20 [79]	10	5	5	7.5	27.5	7.5	5	10	22.5	7.5	2.5	10	5	5	65
S-21 [39]	10	7.5	10	7.5	35	5	5	7.5	17.5	7.5	5	12.5	7.5	7.5	72.5
S-22 [80]	10	5	10	7.5	32.5	7.5	5	5	17.5	7.5	5	12.5	7.5	7.5	70
S-23 [81]	10	2.5	10	7.5	30	10	7.5	7.5	25	7.5	7.5	15	7.5	7.5	77.5
S-24 [82]	10	10	7.5	7.5	35	5	7.5	7.5	20	10	10	20	10	10	85
S-25 [37]	10	10	7.5	7.5	35	5	10	7.5	22.5	5	10	15	10	10	82.5
S-26 [83]	10	5	5	7.5	27.5	2.5	2.5	10	15	7.5	10	17.5	7.5	7.5	67.5
S-27 [84]	7.5	5	5	7.5	25	2.5	5	10	17.5	7.5	10	17.5	7.5	7.5	67.5
S-28 [48]	7.5	5	5	10	27.5	2.5	10	10	22.5	7.5	10	17.5	7.5	7.5	75
S-29 [85]	5	5	10	7.5	27.5	7.5	5	10	22.5	7.5	10	17.5	5	5	72.5
S-30 [86]	10	5	7.5	7.5	30	10	7.5	5	22.5	10	7.5	17.5	10	10	80
S-31 [49]	5	10	7.5	5	27.5	5	10	10	25	5	10	15	10	10	77.5
S-32 [87]	5	10	7.5	7.5	30	5	10	10	25	5	5	10	7.5	7.5	72.5
S-33 [88]	7.5	5	10	7.5	30	2.5	5	7.5	15	7.5	10	17.5	7.5	7.5	70
S-34 [89]	2.5	5	5	7.5	20	2.5	10	10	22.5	7.5	7.5	15	7.5	7.5	65
S-35 [90]	7.5	7.5	5	10	30	2.5	5	10	17.5	7.5	2.5	10	7.5	7.5	65
S-36 [91]	5	5	7.5	7.5	25	10	7.5	5	22.5	10	7.5	17.5	10	10	75
S-37 [92]	7.5	10	7.5	7.5	32.5	5	5	7.5	17.5	5	10	15	5	5	70
S-38 [33]	7.5	5	5	10	27.5	2.5	2.5	10	15	5	10	15	7.5	7.5	65
S-39 [93]	2.5	7.5	5	7.5	22.5	2.5	7.5	10	20	7.5	7.5	15	10	10	67.5
S-40 [34]	5	5	7.5	7.5	25	7.5	5	10	22.5	7.5	10	17.5	5	5	70
S-41 [94]	5	5	10	7.5	27.5	2.5	7.5	10	20	7.5	7.5	15	10	10	72.5
S-42 [95]	10	10	7.5	7.5	35	5	7.5	7.5	20	10	10	20	10	10	85
S-43 [96]	7.5	5	10	7.5	30	2.5	7.5	10	20	7.5	7.5	15	10	10	75
S-44 [97]	7.5	7.5	5	10	30	2.5	5	10	17.5	7.5	2.5	10	7.5	7.5	65
S-45 [98]	7.5	5	5	7.5	25	5	5	7.5	17.5	7.5	10	17.5	10	10	70
S-46 [99]	10	5	5	7.5	27.5	5	5	7.5	17.5	7.5	10	17.5	10	10	72.5
S-47 [100]	10	5	10	7.5	32.5	10	5	7.5	22.5	7.5	5	12.5	10	10	77.5
S-48 [41]	7.5	5	5	10	27.5	10	2.5	10	22.5	10	10	20	7.5	7.5	77.5
S-49 [42]	7.5	5	5	7.5	25	7.5	5	10	22.5	7.5	10	17.5	5	5	70
S-50 [43]	10	7.5	5	7.5	30	10	5	7.5	22.5	7.5	10	17.5	5	5	75
S-51 [44]	10	5	5	7.5	27.5	10	5	7.5	22.5	7.5	5	12.5	10	10	72.5
S-52 [45]	10	7.5	10	7.5	35	10	7.5	10	27.5	10	10	20	10	10	92.5
Averages	7.8	6.4	7.1	7.7	29.1	6.0	6.5	8.7	21.0	7.5	8.5	16.0	8.3	8.3	74.5
%ages	78	64	71	77	72.7	60	65	87	70	75	85	80	83	83	74.5

minor deficiencies (50%) and Poor if meets criteria to a small extent (25%).

It has been observed that the average QA score of all articles is 74.5% which is on the borderline of fair and good. A total of 22 articles (42%) scored more than 75%, hence they are good articles, while the remaining 32 articles (58%) are fair. It has been noted that none of the articles is excellent or poor but 4 articles scored more than 90% and hence are more prominent than the rest. The average scores of the reporting, rigor, credibility, and relevance assessment show that articles scored the highest average of 83% in 'Relevance to the RQs and context' while the lowest average of 72.7% was maintained in 'Reporting of methodology and findings'. Furthermore, 'Robustness and transparency of statistical methods' (a sub-criteria of rigor) scored the lowest average of 60% and, 'Adequacy of sample size and representation' (a sub-criteria of reporting) scored the highest average at 87%.

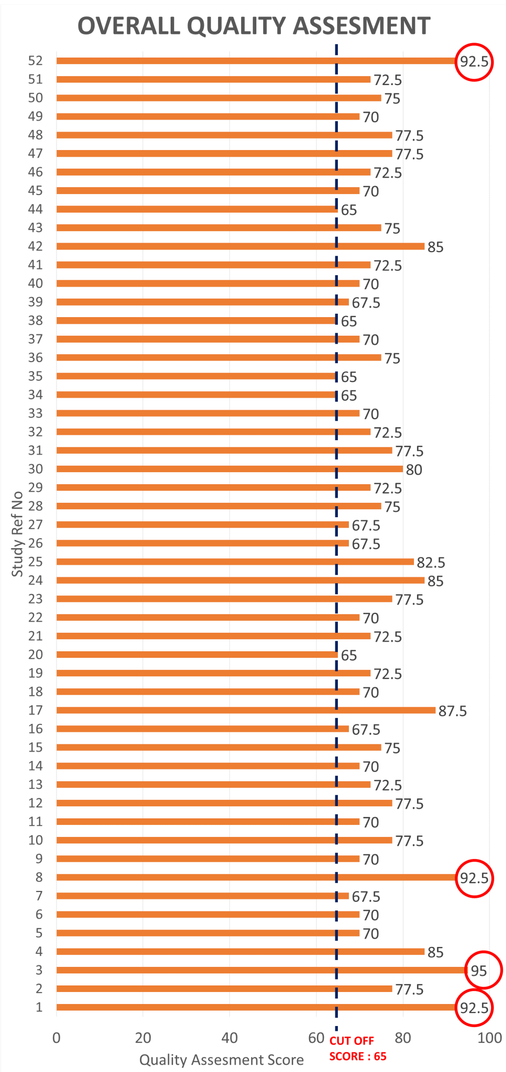

FIGURE 4. Cumulative quality assessment grading (Total 100 points).

Figure 4 shows an overview of the QA of the 52 articles with study reference numbers on the Y axis and scores on the X axis. The black dotted line shows the cut-off score i.e. 65. A cumulative QA grading has been depicted in this histogram for visual comprehension as the top four 'good' articles have been highlighted in red circles. Furthermore, in subsequent figures 5, 6, 7, and 8, the top score line has been indicated by golden, the average by red, and the lowest by a green dotted line. The highest and lowest scores have been marked with red circles.

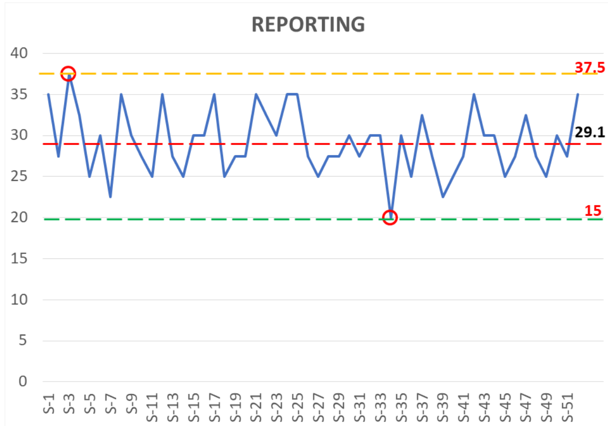

FIGURE 5. Reporting out of total 40 points.

The average score out of a total of 40 points (pt) in Reporting remained at 29.1 pt (72.7%), while one article scored the highest at 37.5 pt and one scored as low as 20 pt as depicted by the line graph in Figure 5.

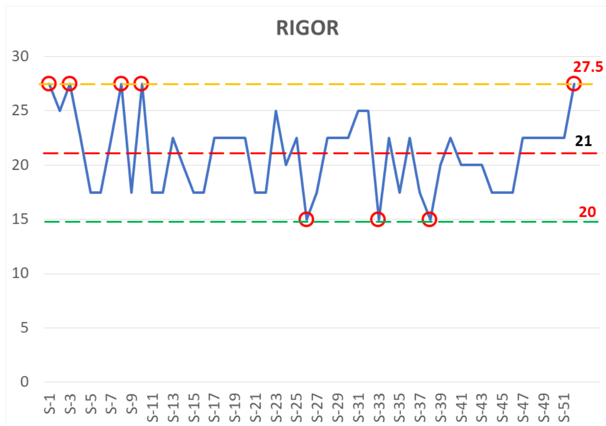

FIGURE 6. Reporting out of total 40 points.

Figure 6 shows the trend of Rigor, in which 5 articles scored 27.5 pt while 3 articles remained at 15 pt, with a cumulative average of 21.0 (70%) out of a total of 30 pt.

In Credibility, the average remained at 16.0 pt (80%), with 9 articles scoring 20 pt and 5 scoring low at 10 pt as shown in Figure 7.

Relevance attained the best average of 8.3 pt (83%) out of a total of 10 pt, the reason being that all the articles selected for the SLR were kept based on inclusion criteria which ensured relevance. Relevance scores have been depicted in Figure 8. Red circles have not been inserted as too many circles would reduce the clarity of the graph.

K. SHORTLISTED ARTICLES

Shortlisting based on quality assessment has been carried out to identify the most suitable articles for this review. Following the assessment, 52 articles have been shortlisted, based on

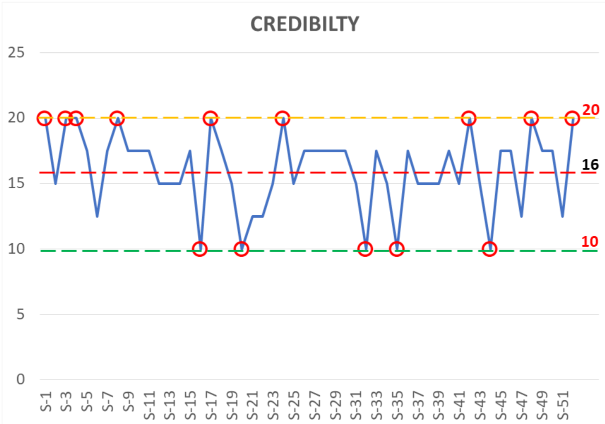

FIGURE 7. Reporting out of total 40 points.

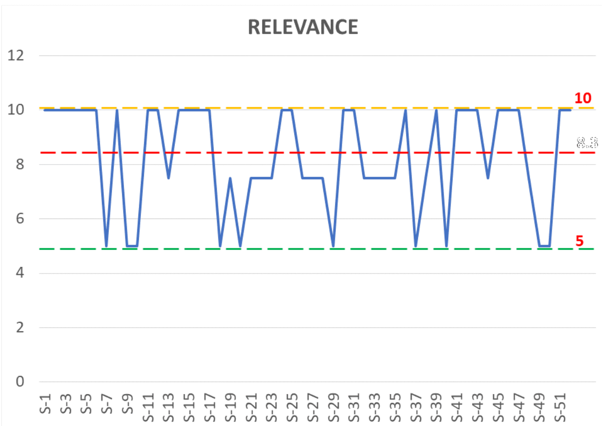

FIGURE 8. Reporting out of total 40 points.

their quality to meet the established criteria, i.e. the cut-off points of 65 out of a total of 100, as depicted in Figure 5. The shortlisting process was crucial in ensuring that the study was based on high-quality research and that the findings were credible and relevant. The shortlisted articles were then subjected to a more detailed analysis, which involved extracting data that were relevant to the research topic.

The analysis of the shortlisted articles has been carried out systematically, to ensure that the articles get reviewed rigorously and meet the highest standards of quality [68]. The findings from the selected articles were used to develop a comprehensive understanding of the research topic. A comparison of four good articles which scored more than 90% has been presented in Table 5.

IV. RESULTS AND DISCUSSIONS

The research format is empirical. Empirical research uses data collection and analysis to address research topics, which are consistent with the adopted methodology. Using a continuum between quantitative and qualitative approaches is a popular strategy to categorize research when using a yardstick to describe the study type [102]. While qualitative research

TABLE 5. Overview of prominent 'good ' articles.

Ref.	Objectives	Models	Dataset	Evaluation Metrics
[29]	To improve MMRE in effort estimation, as well as to find the simplest possible architecture for optimized learning.	Taguchi's Orthogonal Arrays and ANN	COCOMO, NASA and Kermer.	MMRE, MRE, and PRED
[25]	To analyze methods used to estimate size or effort in Agile Software Development methods.	Scrum, XP, TDD, Agile Unified Process, Kanban and Distributed ASD	COCOMO and NASA	MRE and PRED
[31]	To employ ensemble models that would yield better results than standalone models.	Use Case Point, Expert Judgment, and ANN	ISBSG	MMRE and PRED
[38]	To validate an automated genetic framework, and then conduct a sensitivity analysis across different genetic configurations.	SMO and M5P	ISBSG	MMRE, MdMRE, MMAR and Pred

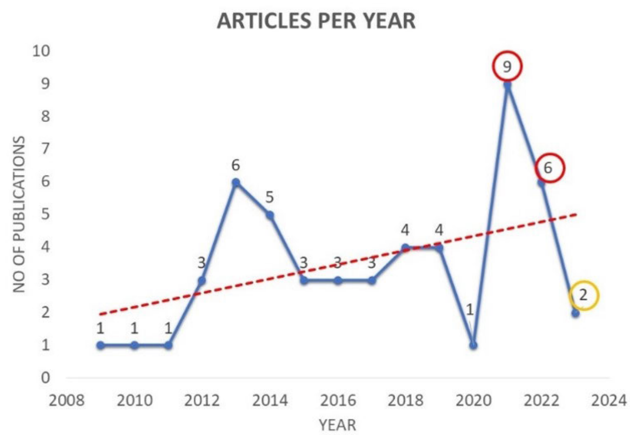

FIGURE 9. Trend in the selected articles over the years.

focuses on gathering non-numerical data that can be analyzed for themes and patterns, quantitative research often entails gathering numerical data that can be statistically analyzed.

In the case of this research, it is primarily quantitative in nature, as the goal is to analyze and compare the effectiveness [103] of different cost estimation techniques to proffer a way forward for future research work. However, there are some qualitative aspects to the research, such as analyzing the reasoning and assumptions underlying different cost estimation techniques. Below is a breakdown of the articles in the light of RQs.

To assist in understanding the evolution of research in this domain in general and the RQs in particular a graphical representation has been presented in Figure 9 which highlights trends in the selected articles over the years. The graph shows

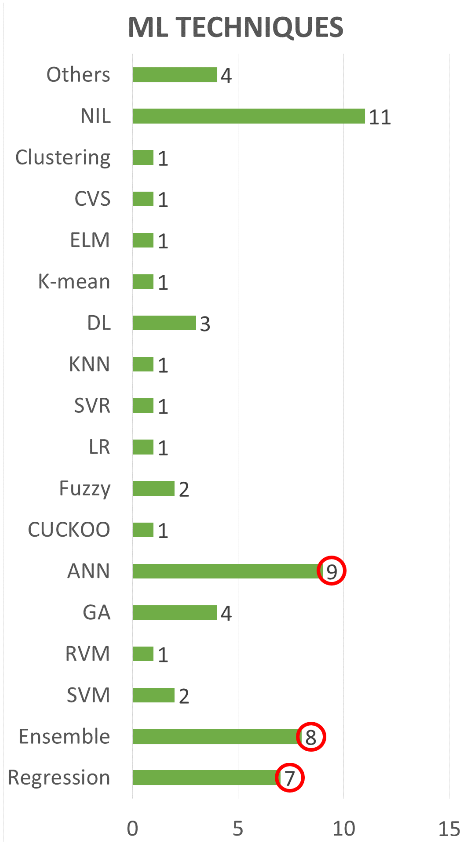

FIGURE 10. Summary of machine learning approaches used for software cost estimation.

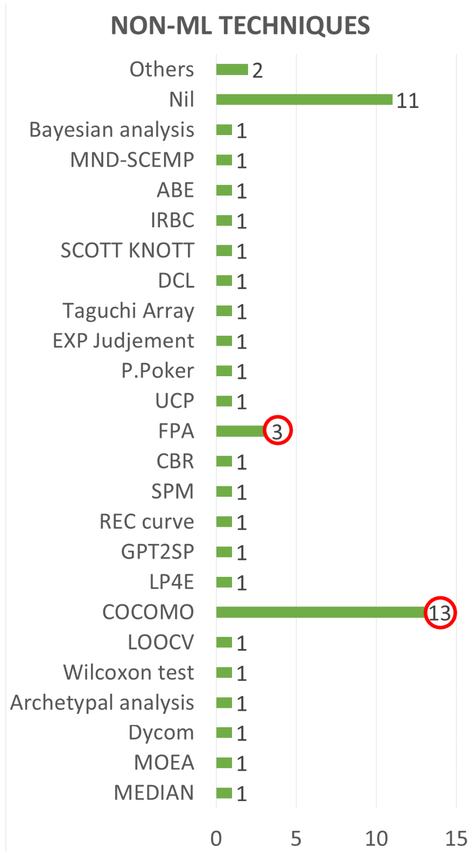

FIGURE 11. Summary of non-machine learning techniques.

TABLE 6. Top ML techniques.

No.	Technique	No. of articles
1.	ANN	9 articles
2.	Ensemble	8 articles
3.	Regression	7 articles

TABLE 7. Top non-ML techniques.

No.	Technique	No. of articles
1.	COCOMO	13 articles
2.	Function Point Analysis	3 articles
3.	Others	2 articles

that the research in the field has increased in the past three years which is a positive trend as depicted by the red dotted trend line. The highest number of publications has been in the year 2021 with nine publications, followed by six in 2022 and 2013. The circle on two articles of 2023 has been marked in yellow color to depict that the year is still not complete as the query string search was based on the results of the first five months of the year 2023.

A. RQ-1: WHAT ARE THE MOST COMMON ML AND NON-ML TECHNIQUES FOR SOFTWARE COST ESTIMATION, AND HOW DO THEY COMPARE TO OTHER COST ESTIMATION METHODS?

Many ML techniques including ANN, Ensemble, Support Vector Machines (SVM), Fuzzy Logic, Genetic Algorithms (GA), etc. have been used in the reviewed articles.

As depicted in Figure 10 and Table 6, the most common ML technique is ANN with 18% usage (9 articles), followed by Ensemble at 16% (8 articles), Regression at 14% (7 articles), and others at 8%.

As depicted in Figure 11 and Table 7, the most common ML technique has been ANN with 18% usage (9 articles), followed by Ensemble at 16% (8 articles), Regression at 14% (7 articles), and others at 8%.

In Non-ML techniques, 33% (13 articles) of the studies used COCOMO followed by FPA at 8% (3 articles). Remaining techniques including CBR (2 articles), Archetypal Analysis, Wilcoxon Test, Use Case Points (UCP), Expert Judgement, etc. remained at a constant 3% usage. These results have been depicted in Figure 11 and leading techniques have been listed in Table 8.

As some studies have used ML, some have used Non-ML techniques while others have used a combination of both. So, it is important to note that the use of techniques can be categorized into three groups:

1) 18% (9) of studies used only Non-ML methods.
2) 28% (15) of studies used only ML techniques.

TABLE 8. A comparison of best techniques.

Best technique	Outpefomred
Regression	CVS [72], ANN [77], [81], SVR [49], Ensemble [81], FPA [72], CBR [34], COCOMO [49]
ANN	Regression [31], [34], [49], Archetypal Analysis [84], COCOMO [29], SVR [31], SVM [93], CBR [91]
Fuzzy Logic	ANN [32], [77], COCOMO [90], Regression [80], Voting Model [77]
GA	FPA [82], COCOMO [13], [75], [78], CBR [41]
Ensemble	ANN [38], [79], Regression [39], [79]
CUCKOO	COCOMO [87], [97]
PSO	COCOMO [13]
DL	GP2SP [37]
Dycom	Regression [37]
EAM	COCOMO [88]
GP2SP	DL [37]
PSO	GA [41]
CBR	ANN [91]
COCOMO II	COCOMO [71]

3) 54% (28) of studies applied a combination of both ML and Non-ML techniques.

The majority of the studies (54%) have used a combination of techniques, which were compared with each other for accuracy. The techniques mentioned in the leftmost column outperformed the techniques in the remaining columns (with lower MMRE values). For example, regression in the first column of the first row produced better results than the other techniques (ANN, CVS, Ensemble, etc.) in eight different experiments. Similarly, ANN performed better than six models in eight different experiments. Fuzzy and GA were the next better approaches, which had better performance in five experiments, each. It has been observed that ML techniques have mostly outperformed Non-ML techniques.

Practical implications of the preferences of ANN and COCOMO have been observed in various software and non-software engineering fields. These techniques not only address the challenges and risks associated with software cost estimation but their application impact project success in real-world scenarios, e.g. global software development [71], construction projects [55], defense industries [74], cross-company database management [99], etc.

B. RQ-2: WHAT FACTORS INFLUENCE THE ACCURACY OF SOFTWARE COST ESTIMATION USING MACHINE LEARNING METHODS, AND HOW CAN THESE FACTORS BE OPTIMIZED TO IMPROVE COST ESTIMATION ACCURACY?

Idri et al. [104] indicated several factors which can influence the accuracy of software cost estimation when using machine learning methods. Optimizing these factors can help improve the accuracy of cost estimation. Here are some key factors to consider:

1) **Data quality and quantity**: Sufficient and representative data is necessary to capture the various factors that contribute to software cost [105], [106]. The outcome of ML techniques specifically, of ANN is largely dependent on this factor [9], [22].
2) Feature selection: Identifying the relevant features that have a significant impact on software costs, such as project size, complexity, team experience, and development methodology is important [107]. These features are more pronounced in the algorithmic methods e.g. cost drivers of COCOMO [2], but their efficacy is not less in ML techniques [9].
3) **Model selection and tuning**: Tuning of factors is of importance in both ML and Non-ML techniques e.g. cost driver tuning in the COCOMO model and bias in ANN [108]. Furthermore, the selection of the right model is the most crucial step to achieving accurate estimates. Experimenting with different algorithms, and fine-tuning their parameters to find the best configuration is necessary to achieve the desired accuracy.
4) **Evaluation metrics**: These are an important component of an SLR and are used to highlight how reliable a particular effort estimation model is [24]. Considering the specific requirements of the cost estimation problem and selecting the metrics that align with the project's goals and constraints is a crucial factor to improve accuracy.

Several evaluation metrics have been used in the articles in our SLR. Prominently, MMRE and MAE have been widely used, which are based on relative and absolute errors. The lower the MMRE and MAE, the better an estimation model would be. On the contrary, in Pred while measuring the percentage of the predicted value, higher values depict better accuracy. Other accuracy measures employed in the selected studies include the magnitude of relative error (MRE), root mean squared error (RMSE), logarithmic standard deviation (LSD), mean square error (MSE), and mean of balanced relative error (MBRE).

A summary of the evaluation metrics used in the reviewed articles has been shown in Figure 12. It can be seen that MMRE has been the most widely used i.e. 35% (18) articles followed by MAE 15% (8) studies. Definitions and equations of these three prominent metrics have been mentioned below:

The magnitude of Relative Error (MRE), as represented in equation 2 is the ratio of the absolute error to the actual measurement.

$$MRE = \frac{ActualEffort - EstimatedActualEffort}{ActualEffort} \qquad (1)$$

Once, the MRE has been calculated for selected N projects (equation 1), then the Mean Magnitude of Relative Error (MMRE) is the average of N projects, as defined in equation 2 [104].

$$MMRE = \frac{1}{N}\sum_{i=1}^{N} MREi \qquad (2)$$

The Mean Absolute Error (MAE) describes the difference between the actual and predicted values and then finds the average of it as depicted in equation 3, where yi is the prediction, xi is the actual value and n is the total number

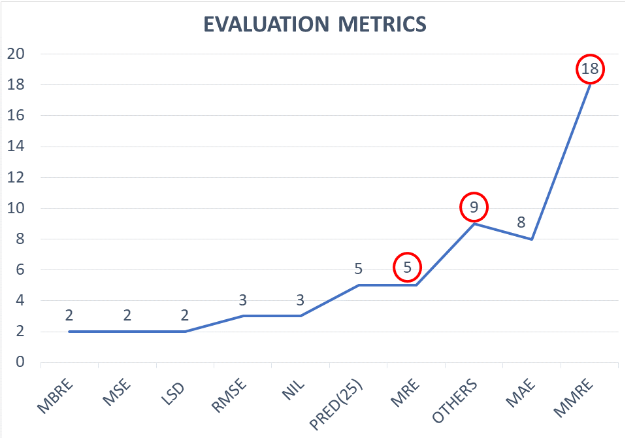

FIGURE 12. Count of error estimation techniques.

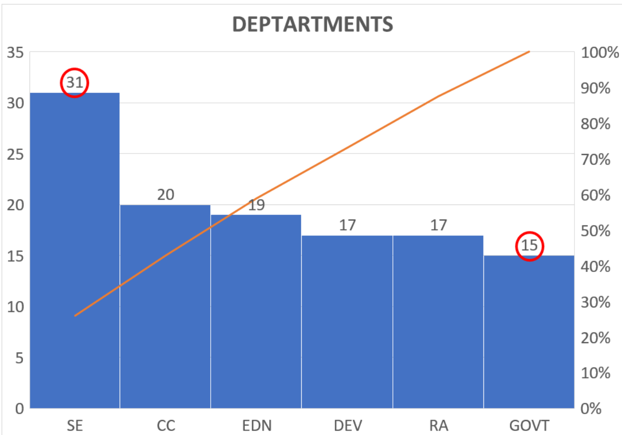

FIGURE 13. Distribution of related departments.

TABLE 9. Summary of departments.

Department	No. of articles
Software Engineering	31 articles
Cross Company Management	20 articles
Risk Analysis	17 articles

of points.

$$MAE = \frac{sum_{i=1}^{n} |yi - xi|}{n} \tag{3}$$

C. RQ-3: WHICH ORGANIZATIONS AND INDUSTRIES HAVE BENEFITTED THE MOST FROM THE SELECTED ARTICLES?

Software effort and cost estimation is a versatile domain that directly or indirectly relates to many organizations and industries [72]. Software engineering and project management are the most frequently related fields to software cost estimation, as these involve the prediction of effort, resources, and time for development and management.

The articles included in this review have a similar relation to diverse fields including software engineering (60%), cross-company data management (38%), Education (36%) remained prominent while governmental departments were the least related at 28%. These figures have been depicted graphically in Figure 13 and tabulated in Table 9. The other less prominent departmental fields include development and risk analysis at 32% each.

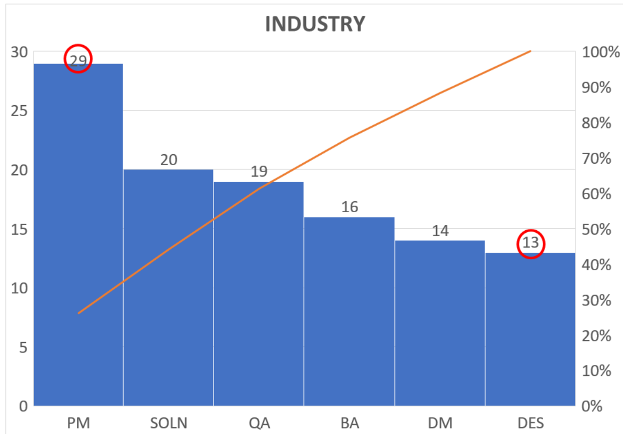

FIGURE 14. Distribution of related industries.

TABLE 10. Summary of industries.

Industry	No. of articles
Project Management	29 articles
Solutions	20 Articles
Quality Assurance	16 articles

In the industrial fields, project management (56%), solutions (38%) and, quality assurance (31%) were the frequently used fields while designing remained the least related at 25%. The breakdown of the articles by the industrial sector is represented in Figure 14 and a summary of prominent industries is given in Table 10. The other fields include business administration and database management at 30% and 27% respectively.

D. RQ-4: WHAT ARE THE MOST COMMONLY ACCESSED REPOSITORIES AND DATASETS IN THESE STUDIES?

More than 10 data repositories which include approximately 20 datasets have been accessed in the under-review articles. Each repository is having a diverse number of datasets and variables. Researchers use different datasets as per the requirements of the experiments.

PROMISE and ISBSG remained the most accessed repositories with 53% (24) and 27% (12) use respectively. These two most accessed repositories have been indicated by red circles in Figure 15. Other repositories include Tuktuku, Cran-R, Jira, Codeproject, CMMI, etc. While 13% of researchers did not access repositories but rather carried out reviews on previously carried out experiments.

Furthermore, 33% of researchers accessed NASA dataset for experimentation, which remained the most widely used dataset. The other prominent datasets include COCOMO,

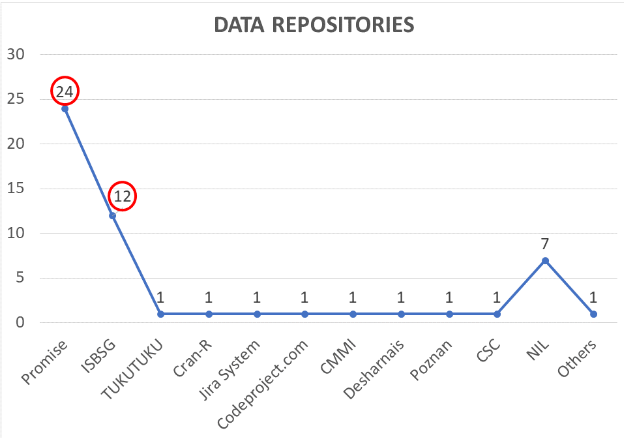

FIGURE 15. Most accessed repositories.

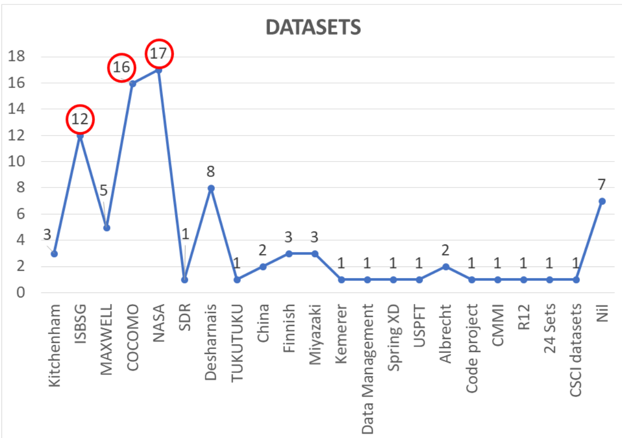

FIGURE 16. Most used datasets.

Maxwell, Desharnais, and Kitchenham. The top three most accessed datasets have been marked in red circles in Figure 16. A brief description of top repositories and datasets is mentioned below.

1) PROMISE Repository: The PROMISE (Project Repository for Software Engineering) repository is a well-known and widely used repository that provides a collection of datasets related to software engineering research [109]. It includes datasets on software cost estimation, defect prediction, effort estimation, and other software engineering topics.
2) ISBSG Repository: The International Software Benchmarking Standards Group (ISBSG) repository is a valuable resource that provides access to a wide range of historical software project data [110]. It includes data on project size, effort, duration, and other relevant metrics, allowing researchers to analyze and model software cost estimation.
3) NASA Dataset: NASA contains datasets collected from various NASA software projects [111]. It includes data on project characteristics, development effort, defects, and other relevant metrics, providing valuable insights for software cost estimation research.
4) Other prominent repositories include Cran-R [112], Jira Systems [113], Codeproject [114], Desharnais [115], and Mendeley [116]. Each of these has data access to datasets with diverse variables consisting of historical data.

TABLE 11. Comparison of systematic literature reviews.

Reference	Duration	No. of articles	Preferred techniques	Accuracy metric	Preferred dataset
Current SLR	2009-2023	52	COCOMO (33%), ANN (18%), Ensemble (16%), Regression (14%)	MMRE (35%), MAE (15%), MAE (10%), PRED (10%)	NASA (33%), COCOMO (29%)
Jadav et al. [23]	1974-2020	1015	Fuzzy logic (10%), ANN (8%), Regression (5%), Analogy (9%), and COCOMO (3%)	MMRE (45%), MRE (20%), and PRED (30%)	NASA (30%), ISBSG (40%)
Ali et al. [24]	1991-2017	75	ANN (60%), SVM (25%), CBR (17%), Regression models (26%)	MMRE (69%), Pred(25) (61%), MdMRE (31%)	NASA (22%), COCOMO (21%), ISBSG (21%)
Wen et al. [70]	1991-2010	84	CBR (37%), ANN (26%), DT (17%), Regression models (36%)	MMRE (89%), Pred(25) 65%, MdMRE (31%)	Desharnais (28%), COCOMO (22%), ISBSG (20)
Marco et al. [117]	2000-2017	74	Regression (25%), ANN (21%), DT (13%), Fuzzy logic (10%), CBR(7%)	MRE (40%), MMRE (80%), Pred (70%)	NASA (17.5%), ISBSG (15%), COCOMO (17.5%)

E. COMPARISON

To highlight the contributions of this study, a summary of the outcomes of this study and four comprehensive SLRs has been presented in Table 11.

The summary draws a comparison between the outcomes and validates the results. The results have been compared in the context of the RQs, research duration, and no of articles. The table compared the following outcomes:

1) Duration of research;
2) Number of articles included in SLR;
3) Most preferred techniques;
4) Most preferred accuracy metrics;
5) Most widely used datasets.

The comparison shows that

1) Out of five SLRs three have been conducted for a duration of 15-20 years. While two SLRs have been conducted for 26 and 46 years.
2) Four SLRs reviewed less than 100 articles, while one reviewed more than 1000 articles.
3) ANN has been the most widely used ML technique in three SLRs followed by Fuzzy Logic while CBR and COCOMO were the most widely used Non-ML techniques.
4) MMRE has been the most preferred accuracy metric in four SLRs while MRE is in one, followed by Pred.
5) NASA has been the most accessed dataset in four out of five SLRs, followed by COCOMO and ISBSG.

The comparison shows that the results obtained in this SLR are almost similar to the results of other SLRs.

F. STUDY LIMITATIONS

All empirical studies are subject to certain limitations due to potential biases, constraints in data collection, or limitations of the chosen techniques. Prominent limitations have been mentioned here for a better perspective.

As the selected studies are software cost and effort prediction-based hence, the process maturity of the level of the under-study industries and their organizational biases are the major limitations of this SLR. Generally, these factors are beyond the control of researchers but they affect the outcomes. Therefore, possible efforts must be made to mitigate these biases.

It has been observed that most organizations do not openly provide historical data and hence the availability of data has restricted the maturity of results and impacted the reliability of the current research. Also, if a researcher has used a slightly different term in the title, then the difference in terminology affected the identification of a specific technique.

Another limitation has been the use of limited query strings, which could lead to missing out on some relevant articles. This can be termed as a bias in the selection procedure. Although the proper definition of query strings and search strategy as per Kitchenham's [56] guidelines have been followed but it is not possible to ensure the inclusion of all relevant articles in all the databases. The selection and a thorough quality assessment were carried out in pairs to minimize potential biases, but some partiality may still exist.

G. SUMMARY OF FINDINGS

1) INFERENCES

Based on the QA, analysis, and comparison of 52 selected articles, the following findings have been inferred:

1) **Techniques**: 33% of articles utilized the COCOMO for software cost estimation, followed by ANN (18%). Hence, it has been inferred that although Non-ML models have been outperformed by ML models, still these are being used in the research community.
2) **Evaluation Metrics**: MMRE was the most widely used evaluation metric employed to ascertain the validity and accuracy of the proposed model. Hence, it has been concluded that all the articles used quantitative analysis for software cost estimation, while a few articles employed qualitative and quantitative both.
3) **Applicability**: Software engineering organizations and project managers in industries are the most related subjects of cost estimation research, which implies that software cost and effort estimation methods are accurately addressing the core issue of predicting development cycles.
4) **Databases**: The majority of the articles (90%) used historical data (repositories), while a few articles (10%) used simulated or current data, which had restricted access. Hence, it has been inferred that efforts have not been made to access better and more comprehensive datasets by the researchers.
5) **Low score in QA:** Low average of 72.7% was observed in 'Reporting of methodology and findings'. Furthermore, within the assessment criteria of rigor the sub-criteria 'Robustness and transparency of statistical methods' scored the lowest average of 60%, which shows that researchers do not report methods and outcomes explicitly hence transparency is compromised.

2) GAP ANALYSIS

Based on the analysis of the articles and inferences, potential gaps in the current research on software cost estimation have been identified. Key gaps are described below:

1) Limited use of evolving techniques: As indicated in RQ-1 that most ML techniques have outperformed the non-ML techniques, but still a greater number of studies have used non-ML techniques. Keeping in view the better performance the emerging fields of ML, including GA and Blockchain techniques need more exploration because the accuracy of the software estimation improves when relevant features of the datasets are selected [24]. Studies in the fields of GA and Blockchain as compiled by Ahmed et al. [40] have gained attention in various software engineering fields but are still used very less often in software effort and cost estimation.
2) Dearth of comprehensive evaluation: The evaluation of many studies was restricted to a limited range of projects and methods. Studies that encompass a more comprehensive comparison on the lines of Elish et al. [38] are recommended. Although a variety of ML and Non-ML techniques have been used by researchers, experiments have not been conducted on which technique performs better, concerning different

types and sizes of datasets. It needs to be analyzed which techniques outperform others on small datasets and vice versa.
3) Applicability to multiple fields: As indicated in RQ-3, software engineering, and project management are very important domains where cost estimation has been widely used, but emerging domains like database management and designing (28% each) may be given more attention by researchers. Similarly, governmental projects (29%) have been the least researched with cost estimation. As governmental schemes impact the largest part of society members these require a more focused approach by researchers as in the case of Cibir and Ayyildiz [74].
4) Absence of latest and comprehensive datasets: Detailed and high-quality databases are crucial in effort estimation. However, it was observed that 90% (47 articles) of the studies have relied on old datasets (Albrecht, COCOMO, Finnish, Kemerer, Maxwell, NASA), which are not current and detailed descriptions of project features is not available in those datasets. Furthermore, they do not represent cross-company data sharing. It is therefore imperative for the researchers to work on current datasets which are likely to introduce new factors in the research as proposed by Minku and Yao [80]. It is further recommended that researchers having access to comprehensive datasets share the proprietary datasets on accessible forums, e.g. PROMISE repository after the removal of the confidential features.
5) Clarity in reporting of methodology and findings: As mentioned in the QA section that 72.7% of articles report methodology clearly, which means that approximately 27.8% of articles lack clear reporting. Similarly, 40% of articles lack robustness and transparency of the statistical methods. Therefore, it is recommended that researchers focus more on these two aspects to further enhance the research methodology and add to the body of knowledge.

Overall, there is still significant room for improvement in the area of software cost estimation using different types of techniques and addressing these gaps could lead to more accurate and reliable cost estimation models.

3) CHALLENGES

Some challenges in the light of the research gap that needs to be addressed in future research include:

1) **Drawing accurate results**: One of the challenges in the implementation of evolving ML techniques is to achieve accurate and efficient results that can handle the complexity of projects. Although, it has been observed that ML models are autonomous and outperform the Non-ML models; but, are prone to estimation errors. Effort series forecasting in software estimation is a challenge for ML models [118]. Furthermore, determining the right model for different types of datasets also remains a challenge [119], [120].
2) **Adopting evaluation benchmarks**: Validating estimation models on benchmarks for model performance and scalability evaluation is essential to ensure accuracy. To induce comprehensive evaluation as indicated in the gap analysis, researchers need to adopt established benchmark checklists. Nonadopting standard is a challenge that needs to be mitigated to set a course of action for future research work, as proposed by Minku and Yao [80] and Hasselbring [121].
3) **Handling uncertainty**: Software development is inherently uncertain, which needs to be accounted for in cost estimation models. As it has been suggested that researchers need to expand the research to emerging techniques, it is pertinent to highlight that uncertainty will be an inherent challenge in that [119].
4) **Data availability**: The availability of multi-featured comprehensive datasets is one of the major challenges which is being faced by the research community. Obtaining such data is a difficult task as many organizations keep data undisclosed based on privacy concerns. Therefore, researchers with access to private data sets are encouraged to share the data sets after the replacement of confidential features with false values.
5) **Quality of research methodology**: As indicated in the research gap, researchers need to adopt robust and standard protocols as indicated by Kitchenham [101]. Here, researchers need to draw a balance between the importance of clearly reporting methodology along with giving ample importance to the analysis of research.

Addressing these challenges will help in developing more accurate and effective software cost estimation models that can be used to improve project planning and management and add to the body of knowledge.

V. CONCLUSION AND FUTURE WORK

We have performed an SLR on software cost and effort estimation using ML and Non-ML techniques from the past decade and a half. The review identified and analyzed a total of 52 studies from five renowned digital libraries. Although many SLRs [21], [22], [23], [24] have been published but a substantial amount of work has been done in the last 15 years in the field of software effort estimation which needed to be reviewed. Moreover, there are differences between the approaches and conduct of different SLRs. Hence this SLR has been conducted to investigate the current research trends in software cost and effort estimation to indicate the most widely used estimating approaches, datasets, and application areas to suggest a future way fwd. Project planning and decision-making can both benefit from the findings of this study.

It has been found that ANN; and COCOMO are the most popular techniques followed by Ensemble and FPA. Also, ANN has outperformed several ML and Non-ML techniques. The MMRE is the most commonly used accuracy metric. Software engineering; and project management are the most relevant fields, and the PROMISE repository has been identified as the preferred database. The most widely used dataset indicated in this SLR is NASA.

Research communities appreciate the use of advanced technologies to improve the estimation results. While widely used approaches have been examined, there is still a limitation to this SLR that emerging technologies like blockchain and bio-inspired feature selection algorithms have not been examined in depth. In future work, we intend to perform an SLR on blockchain-based software effort estimation methods and bio-inspired feature selection algorithms that have been used recently in a few studies [29], [40] to address the software cost and effort estimation problems.

REFERENCES

[1] A. K. Bardsiri and S. M. Hashemi, "Software effort estimation: A survey of well-known approaches," *Int. J. Comput. Sci. Eng.*, vol. 3, no. 1, pp. 46–50, 2014.

[2] P. Singal, A. C. Kumari, and P. Sharma, "Estimation of software development effort: A differential evolution approach," *Proc. Comput. Sci.*, vol. 167, pp. 2643–2652, 2020.

[3] L. Cao, "Estimating efforts for various activities in agile software development: An empirical study," *IEEE Access*, vol. 10, pp. 83311–83321, 2022.

[4] S. S. Gautam and V. Singh, "Adaptive discretization using golden section to aid outlier detection for software development effort estimation," *IEEE Access*, vol. 10, pp. 90369–90387, 2022.

[5] J. Shah and N. Kama, "Extending function point analysis effort estimation method for software development phase," in *Proc. 7th Int. Conf. Softw. Comput. Appl.*, Feb. 2018, pp. 77–81.

[6] M. Kyrillidou and S. Giersch, "Developing the DigiQUAL protocol for digital library evaluation," in *Proc. 5th ACM/IEEE-CS Joint Conf. Digit. Libraries*, Jun. 2005, pp. 172–173.

[7] S. Sinha, B. Lacquet, and B. T. J. Maharaj, "IEEE xplore digital library indexes the transactions of the SAIEE, 1909 to date," *SAIEE Afr. Res. J.*, vol. 112, no. 4, pp. 158–159, Dec. 2021.

[8] I. F. de Barcelos Tronto, J. D. S. da Silva, and N. Sant'Anna, "An investigation of artificial neural networks based prediction systems in software project management," *J. Syst. Softw.*, vol. 81, no. 3, pp. 356–367, Mar. 2008.

[9] M. Tsunoda, A. Monden, J. Keung, and K. Matsumoto, "Incorporating expert judgment into regression models of software effort estimation," in *Proc. 19th Asia–Pacific Softw. Eng. Conf.*, vol. 1, Dec. 2012, pp. 374–379.

[10] P. G. Moore, "The Delphi method: Techniques and applications," *J. Roy. Stat. Soc. A, Gen.*, vol. 140, no. 2, pp. 257–258, 1977.

[11] M. Jorgensen and M. Shepperd, "A systematic review of software development cost estimation studies," *IEEE Trans. Softw. Eng.*, vol. 33, no. 1, pp. 33–53, Jan. 2007.

[12] M. G. Stochel, "Reliability and accuracy of the estimation process–wideband delphi vs. wisdom of crowds," in *Proc. IEEE 35th Annu. Comput. Softw. Appl. Conf.*, Jul. 2011, pp. 350–359.

[13] M. S. Khan, F. Jabeen, S. Ghouzali, Z. Rehman, S. Naz, and W. Abdul, "Metaheuristic algorithms in optimizing deep neural network model for software effort estimation," *IEEE Access*, vol. 9, pp. 60309–60327, 2021.

[14] M. Ullah, R. Ali, Abdullah, M. Ahmad, T. Khan, and F. U. Mulk, "Software cost estimation—A comparative study of COCOMO-II and Bailey-Basili models," in *Proc. Int. Conf. Adv. Emerg. Comput. Technol. (AECT)*, Feb. 2020, pp. 1–5.

[15] R. K. Sachan, A. Nigam, A. Singh, S. Singh, M. Choudhary, A. Tiwari, and D. S. Kushwaha, "Optimizing basic COCOMO model using simplified genetic algorithm," *Proc. Comput. Sci.*, vol. 89, pp. 492–498, 2016.

[16] A. P. Subriadi and A. Y. P. Putri, "The need to critical review of function point analysis," in *Proc. Int. Seminar Res. Inf. Technol. Intell. Syst. (ISRITI)*, Nov. 2018, pp. 67–71.

[17] V. Van Hai, H. L. T. K. Nhung, Z. Prokopova, R. Silhavy, and P. Silhavy, "Toward improving the efficiency of software development effort estimation via clustering analysis," *IEEE Access*, vol. 10, pp. 83249–83264, 2022.

[18] M. Rahman, P. P. Roy, M. Ali, and H. Sarwar, "Software effort estimation using machine learning technique," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 4, pp. 1–6, 2023.

[19] H. L. T. K. Nhung, V. Van Hai, R. Silhavy, Z. Prokopova, and P. Silhavy, "Parametric software effort estimation based on optimizing correction factors and multiple linear regression," *IEEE Access*, vol. 10, pp. 2963–2986, 2022.

[20] C. Yu and W. Yao, "Robust linear regression: A review and comparison," *Commun. Statist. Simul. Comput.*, vol. 46, no. 8, pp. 6261–6282, Sep. 2017.

[21] N. Halimawan, F. W. P. Dharma, Wincent, and N. Surantha, "Model and dataset trend in software project effort estimation—A systematic literature review," *ICIC Int.*, vol. 15, no. 10, pp. 1109–1117, 2021.

[22] M. Fransoy and F. Pinciroli, "Sistematic mapping protocol–estimation accuracy on software development using agile technologies," 2022, *arXiv:2208.12164*.

[23] A. Jadhav, M. Kaur, and F. Akter, "Evolution of software development effort and cost estimation techniques: Five decades study using automated text mining approach," *Math. Problems Eng.*, vol. 2022, pp. 1–17, May 2022.

[24] A. Ali and C. Gravino, "A systematic literature review of software effort prediction using machine learning methods," *J. Software: Evol. Process*, vol. 31, no. 10, p. e2211, Oct. 2019.

[25] M. Fernández-Diego, E. R. Méndez, F. González-Ladrón-De-Guevara, S. Abrahão, and E. Insfran, "An update on effort estimation in agile software development: A systematic literature review," *IEEE Access*, vol. 8, pp. 166768–166800, 2020.

[26] J. Kolodner, "An introduction to case-based reasoning," *Artif. Intell. Rev.*, vol. 6, pp. 3–34, Mar. 1992, doi: 10.1007/BF00155578.

[27] B. Boehm, C. Abts, and S. Chulani, "Software development cost estimation approaches—A survey," *Ann. Softw. Eng.*, vol. 10, nos. 1–4, pp. 177–205, 2000.

[28] F. J. Heemstra and R. J. Kusters, "Function point analysis: Evaluation of a software cost estimation model," *Eur. J. Inf. Syst.*, vol. 1, no. 4, pp. 229–237, Dec. 1991.

[29] N. Rankovic, D. Rankovic, M. Ivanovic, and L. Lazic, "A new approach to software effort estimation using different artificial neural network architectures and Taguchi orthogonal arrays," *IEEE Access*, vol. 9, pp. 26926–26936, 2021.

[30] H. D. P. De Carvalho, R. Fagundes, and W. Santos, "Extreme learning machine applied to software development effort estimation," *IEEE Access*, vol. 9, pp. 92676–92687, 2021.

[31] S. S. Ali, J. Ren, K. Zhang, J. Wu, and C. Liu, "Heterogeneous ensemble model to optimize software effort estimation accuracy," *IEEE Access*, vol. 11, pp. 27759–27792, 2023.

[32] A. B. Nassif, M. Azzeh, A. Idri, and A. Abran, "Software development effort estimation using regression fuzzy models," *Comput. Intell. Neurosci.*, vol. 2019, pp. 1–17, Feb. 2019.

[33] N. Mittas and L. Angelis, "Visual comparison of software cost estimation models by regression error characteristic analysis," *J. Syst. Softw.*, vol. 83, no. 4, pp. 621–637, Apr. 2010.

[34] A. Sharma and N. Chaudhary, "Prediction of software effort by using non-linear power regression for heterogeneous projects based on use case points and lines of code," *Proc. Comput. Sci.*, vol. 218, pp. 1601–1611, 2023.

[35] C. S. Yadav and R. Singh, "Prediction model for object oriented software development effort estimation using one hidden layer feed forward neural network with genetic algorithm," *Adv. Softw. Eng.*, vol. 2014, pp. 1–6, Jun. 2014.

[36] A. Iftikhar, M. Alam, R. Ahmed, S. Musa, and M. M. Su'ud, "Risk prediction by using artificial neural network in global software development," *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–25, Dec. 2021.

[37] H. Phan and A. Jannesari, "Heterogeneous graph neural networks for software effort estimation," 2022, *arXiv:2206.11023*.

[38] M. O. Elish, T. Helmy, and M. I. Hussain, "Empirical study of homogeneous and heterogeneous ensemble models for software development effort estimation," *Math. Problems Eng.*, vol. 2013, pp. 1–21, Jan. 2013.

[39] L. L. Minku and X. Yao, "An analysis of multi-objective evolutionary algorithms for training ensemble models based on different performance measures in software effort estimation," in *Proc. 9th Int. Conf. Predictive Models Softw. Eng.*, Oct. 2013, pp. 1–10.

[40] M. Ahmed, N. Iqbal, F. Hussain, M.-A. Khan, M. Helfert, Imran, and J. Kim, "Blockchain-based software effort estimation: An empirical study," *IEEE Access*, vol. 10, pp. 120412–120425, 2022.

[41] S. K. Palaniswamy and R. Venkatesan, "Hyperparameters tuning of ensemble model for software effort estimation," *J. Ambient Intell. Humanized Comput.*, vol. 12, pp. 6579–6589, Jun. 2021.

[42] A. Sharma and D. S. Kushwaha, "An empirical approach for early estimation of software testing effort using SRS document," *CSI Trans. ICT*, vol. 1, no. 1, pp. 51–66, Mar. 2013.

[43] M. Sadiq and M. Shahid, "A systematic approach for the estimation of software risk and cost using esrcTool," *CSI Trans. ICT*, vol. 1, no. 3, pp. 243–252, Sep. 2013.

[44] T. Urbanek, Z. Prokopova, R. Silhavy, and V. Vesela, "Prediction accuracy measurements as a fitness function for software effort estimation," *SpringerPlus*, vol. 4, no. 1, pp. 1–17, Dec. 2015.

[45] J. Murillo-Morera, C. Quesada-López, C. Castro-Herrera, and M. Jenkins, "A genetic algorithm based framework for software effort prediction," *J. Softw. Eng. Res. Develop.*, vol. 5, no. 1, pp. 1–33, Dec. 2017.

[46] F. Liu, S. He, Z. Li, and Z. Li, "An overview of blockchain efficient interaction technologies," *Frontiers Blockchain*, vol. 6, Feb. 2023, Art. no. 996070.

[47] M. J. H. Faruk, S. Subramanian, H. Shahriar, M. Valero, X. Li, and M. Tasnim, "Software engineering process and methodology in blockchain-oriented software development: A systematic study," in *Proc. IEEE/ACIS 20th Int. Conf. Softw. Eng. Res., Manage. Appl. (SERA)*, May 2022, pp. 120–127.

[48] A. Saeed, W. H. Butt, F. Kazmi, and M. Arif, "Survey of software development effort estimation techniques," in *Proc. 7th Int. Conf. Softw. Comput. Appl.*, Feb. 2018, pp. 82–86.

[49] Z. A. Khalifelu and F. S. Gharehchopogh, "Comparison and evaluation of data mining techniques with algorithmic models in software cost estimation," *Proc. Technol.*, vol. 1, pp. 65–71, 2012.

[50] M. Usman, E. Mendes, F. Weidt, and R. Britto, "Effort estimation in agile software development: A systematic literature review," in *Proc. 10th Int. Conf. Predictive Models Softw. Eng.*, 2014, pp. 82–91.

[51] N. Thai Huynh, T. V. T. Nguyen, and Q. Manh Nguyen, "Optimum design for the magnification mechanisms employing fuzzy logic–ANFIS," *Comput., Mater. Continua*, vol. 73, no. 3, pp. 5961–5983, 2022.

[52] Q. N. The Ho, T. T. Do, P. S. Minh, V.-T. Nguyen, and V. T. T. Nguyen, "Turning chatter detection using a multi-input convolutional neural network via image and sound signal," *Machines*, vol. 11, no. 6, p. 644, Jun. 2023.

[53] Y. Yang, M. He, M. Li, Q. Wang, and B. Boehm, "Phase distribution of software development effort," in *Proc. 2nd ACM-IEEE Int. Symp. Empirical Softw. Eng. Meas.*, Oct. 2008, pp. 61–69.

[54] F. H'mida, P. Martin, and F. Vernadat, "Cost estimation in mechanical production: The cost entity approach applied to integrated product engineering," *Int. J. Prod. Econ.*, vol. 103, no. 1, pp. 17–35, Sep. 2006.

[55] X. Xue, Y. Jia, and Y. Tang, "Expressway project cost estimation with a convolutional neural network model," *IEEE Access*, vol. 8, pp. 217848–217866, 2020.

[56] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering—A systematic literature review," *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 7–15, 2009.

[57] L. Yang, H. Zhang, H. Shen, X. Huang, X. Zhou, G. Rong, and D. Shao, "Quality assessment in systematic literature reviews: A software engineering perspective," *Inf. Softw. Technol.*, vol. 130, Feb. 2021, Art. no. 106397.

[58] J. K. Owens, "Systematic reviews: Brief overview of methods, limitations, and resources," *Nurse Author Editor*, vol. 31, nos. 3–4, pp. 69–72, Dec. 2021.

[59] C. Wohlin, M. Kalinowski, K. Romero Felizardo, and E. Mendes, "Successful combination of database search and snowballing for identification of primary studies in systematic literature studies," *Inf. Softw. Technol.*, vol. 147, Jul. 2022, Art. no. 106908.

[60] P. V. Torres-Carrión, C. S. González-González, S. Aciar, and G. Rodríguez-Morales, "Methodology for systematic literature review applied to engineering and education," in *Proc. IEEE Global Eng. Educ. Conf. (EDUCON)*, Apr. 2018, pp. 1364–1373.

[61] C.-Y. Chang, C.-L. Lai, and G.-J. Hwang, "Trends and research issues of mobile learning studies in nursing education: A review of academic publications from 1971 to 2016," *Comput. Educ.*, vol. 116, pp. 28–48, Jan. 2018.

[62] J. Higgins. (Mar. 2011). *Cochrane Handbook for Systematic Reviews of Interventions. Version 5.1. 0*. The Cochrane Collaboration. [Online]. Available: https://www.cochrane-handbook.org

[63] G. M. Sullivan and A. R. Artino, "Analyzing and interpreting data from likert-type scales," *J. Graduate Med. Educ.*, vol. 5, no. 4, pp. 541–542, Dec. 2013.

[64] J. Carifio and R. Perla, "Resolving the 50-year debate around using and misusing Likert scales," *Med. Educ.*, vol. 42, no. 12, pp. 1150–1152, Dec. 2008.

[65] C. Batini, C. Cappiello, C. Francalanci, and A. Maurino, "Methodologies for data quality assessment and improvement," *ACM Comput. Surveys*, vol. 41, no. 3, pp. 1–52, Jul. 2009.

[66] E. von Elm, D. G. Altman, M. Egger, S. J. Pocock, P. C. Gøtzsche, and J. P. Vandenbroucke, "The strengthening the reporting of observational studies in epidemiology (STROBE) statement: Guidelines for reporting observational studies," *Lancet*, vol. 370, no. 9596, pp. 1453–1457, Oct. 2007.

[67] M. Egger, M. Schneider, and G. D. Smith, "Meta-analysis spurious precision? Meta-analysis of observational studies," *BMJ*, vol. 316, no. 7125, pp. 140–144, Jan. 1998.

[68] S. Cuschieri, "WASP (write a scientific paper): Understanding research metrics," *Early Human Develop.*, vol. 118, pp. 67–71, Mar. 2018.

[69] C. Kelly and M. Jennions, "The H index and career assessment by numbers," *Trends Ecology Evol.*, vol. 21, no. 4, pp. 167–170, Apr. 2006.

[70] J. Wen, S. Li, Z. Lin, Y. Hu, and C. Huang, "Systematic literature review of machine learning based software development effort estimation models," *Inf. Softw. Technol.*, vol. 54, no. 1, pp. 41–59, Jan. 2012.

[71] J. A. Khan, S. U. R. Khan, T. A. Khan, and I. U. R. Khan, "An amplified COCOMO-II based cost estimation model in global software development context," *IEEE Access*, vol. 9, pp. 88602–88620, 2021.

[72] P. Silhavy, R. Silhavy, and Z. Prokopova, "Categorical variable segmentation model for software development effort estimation," *IEEE Access*, vol. 7, pp. 9618–9626, 2019.

[73] J. A. Khan, S. U. R. Khan, J. Iqbal, and I. U. Rehman, "Empirical investigation about the factors affecting the cost estimation in global software development context," *IEEE Access*, vol. 9, pp. 22274–22294, 2021.

[74] E. Cibir and T. E. Ayyildiz, "An empirical study on software test effort estimation for defense projects," *IEEE Access*, vol. 10, pp. 48082–48087, 2022.

[75] B. K. Singh, S. Tiwari, K. K. Mishra, and A. K. Misra, "Tuning of cost drivers by significance occurrences and their calibration with novel software effort estimation method," *Adv. Softw. Eng.*, vol. 2013, pp. 1–10, Dec. 2013.

[76] H. Najadat, I. Alsmadi, and Y. Shboul, "Predicting software projects cost estimation based on mining historical data," *Int. Scholarly Res. Notices*, vol. 2012, Apr. 2012, Art. no. 823437.

[77] L. F. Capretz and V. Marza, "Improving effort estimation by voting software estimation models," *Adv. Softw. Eng.*, vol. 2009, pp. 1–8, Sep. 2009.

[78] C. A. U. Hassan, M. S. Khan, R. Irfan, J. Iqbal, S. Hussain, S. S. Ullah, R. Alroobaea, and F. Umar, "Optimizing deep learning model for software cost estimation using hybrid meta-heuristic algorithmic approach," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–20, Oct. 2022.

[79] L. Song, L. L. Minku, and X. Yao, "The impact of parameter tuning on software effort estimation using learning machines," in *Proc. 9th Int. Conf. Predictive Models Softw. Eng.*, Oct. 2013, pp. 1–10.

[80] L. L. Minku and X. Yao, "How to make best use of cross-company data in software effort estimation?" in *Proc. 36th Int. Conf. Softw. Eng.*, May 2014, pp. 446–456.

[81] L. Song, L. L. Minku, and X. Yao, "The potential benefit of relevance vector machine to software effort estimation," in *Proc. 10th Int. Conf. Predictive Models Softw. Eng.*, Sep. 2014, pp. 52–61.

[82] F. Sarro, A. Petrozziello, and M. Harman, "Multi-objective software effort estimation," in *Proc. IEEE/ACM 38th Int. Conf. Softw. Eng. (ICSE)*, May 2016, pp. 619–630.

[83] A. Agrawal, V. Jain, and M. Sheikh, "Quantitative estimation of cost drivers for intermediate COCOMO towards traditional and cloud based software development," in *Proc. 9th Annu. ACM India Conf.*, Oct. 2016, pp. 85–95.

[84] N. Mittas, V. Karpenisi, and L. Angelis, "Benchmarking effort estimation models using archetypal analysis," in *Proc. 10th Int. Conf. Predictive Models Softw. Eng.*, Sep. 2014, pp. 62–71.

[85] F. Sarro and A. Petrozziello, "Linear programming as a baseline for software effort estimation," *ACM Trans. Softw. Eng. Methodology*, vol. 27, no. 3, pp. 1–28, Jul. 2018.

[86] B. Sigweni, M. Shepperd, and T. Turchi, "Realistic assessment of software effort estimation models," in *Proc. 20th Int. Conf. Eval. Assessment Softw. Eng.*, Jun. 2016, pp. 1–6.

[87] A. Puspaningrum and R. Sarno, "A hybrid cuckoo optimization and harmony search algorithm for software cost estimation," *Proc. Comput. Sci.*, vol. 124, pp. 461–469, 2017.

[88] T. Singh, R. Singh, and K. K. Mishra, "Software cost estimation using environmental adaptation method," *Proc. Comput. Sci.*, vol. 143, pp. 325–332, 2018.

[89] X. Qin and M. Fang, "Summarization of software cost estimation," *Proc. Eng.*, vol. 15, pp. 3027–3031, 2011.

[90] S. Malathi and S. Sridhar, "Optimization of fuzzy analogy in software cost estimation using linguistic variables," *Proc. Eng.*, vol. 38, pp. 177–190, 2012.

[91] J. Huang, Y.-F. Li, and M. Xie, "An empirical analysis of data preprocessing for machine learning-based software cost estimation," *Inf. Softw. Technol.*, vol. 67, pp. 108–127, Nov. 2015.

[92] M. Jørgensen, "Measurement of software development effort estimation bias: Avoiding biased measures of estimation bias," 2022, *arXiv:2203.15864*.

[93] N. Mittas, E. Papatheocharous, L. Angelis, and A. S. Andreou, "Integrating non-parametric models with linear components for producing software cost estimations," *J. Syst. Softw.*, vol. 99, pp. 120–134, Jan. 2015.

[94] A. Kaushik, P. Kaur, N. Choudhary, and Priyanka, "Stacking regularization in analogy-based software effort estimation," *Soft Comput.*, vol. 26, pp. 1197–1216, Jan. 2022.

[95] T. Lee, T. Gu, and J. Baik, "MND-SCEMP: An empirical study of a software cost estimation modeling process in the defense domain," *Empirical Softw. Eng.*, vol. 19, no. 1, pp. 213–240, Feb. 2014.

[96] L. Gren and R. B. Svensson, "Is it possible to disregard obsolete requirements? A family of experiments in software effort estimation," *Requirements Eng.*, vol. 26, pp. 459–480, Apr. 2021.

[97] S. Kumari and S. Pushkar, "Cuckoo search based hybrid models for improving the accuracy of software effort estimation," *Microsyst. Technol.*, vol. 24, no. 12, pp. 4767–4774, Dec. 2018.

[98] L. L. Minku and X. Yao, "Which models of the past are relevant to the present? A software effort estimation approach to exploiting useful past models," *Automated Softw. Eng.*, vol. 24, pp. 499–542, Dec. 2017.

[99] L. L. Minku, "A novel online supervised hyperparameter tuning procedure applied to cross-company software effort estimation," *Empirical Softw. Eng.*, vol. 24, no. 5, pp. 3153–3204, Oct. 2019.

[100] K. E. Rao and G. A. Rao, "RETRACTED ARTICLE: Ensemble learning with recursive feature elimination integrated software effort estimation: A novel approach," *Evol. Intell.*, vol. 14, no. 1, pp. 151–162, Mar. 2021.

[101] S. Keele, "Guidelines for performing systematic literature reviews in software engineering," School Comput. Sci. Math., Keele Univ., London, U.K., Tech. Rep. EBSE-2007-01, 2007.

[102] L. McLeod, S. G. MacDonell, and B. Doolin, "Qualitative research on software development: A longitudinal case study methodology," *Empirical Softw. Eng.*, vol. 16, no. 4, pp. 430–459, Aug. 2011.

[103] M. Di Penta and D. A. Tamburri, "Combining quantitative and qualitative studies in empirical software engineering research," in *Proc. IEEE/ACM 39th Int. Conf. Softw. Eng. Companion (ICSE-C)*, May 2017, pp. 499–500.

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

[105] K. E. Maani, M. S. Putterill, and D. G. Sluti, "Empirical analysis of quality improvement in manufacturing," *Int. J. Quality Rel. Manage.*, vol. 11, no. 7, pp. 19–37, Oct. 1994.

[106] J. H. F. Meyer, M. P. Shanahan, and R. C. Laugksch, "Students' conceptions of research. I: A qualitative and quantitative analysis," *Scandin. J. Educ. Res.*, vol. 49, no. 3, pp. 225–244, Jul. 2005.

[107] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Comput. Surv.*, vol. 50, no. 6, pp. 1–45, Jan. 2018.

[108] J. L. Garriga and M. Soroush, "Model predictive control tuning methods: A review," *Ind. Eng. Chem. Res.*, vol. 49, no. 8, pp. 3505–3515, Apr. 2010.

[109] J. S. Shirabad and T. Menzies, "The PROMISE repository of software engineering databases," School Inf. Technol. Eng., Univ. Ottawa, Ottawa, ON, Canada, Tech. Rep. 5, 2005. [Online]. Available: http://promise.site.uottawa.ca/SERepository

[110] ISBSG. (2022). *ISBSG Development & Enhancement Repository Release 13*. [Online]. Available: http://isbsg.org

[111] NASA. (2022). *NASA Data Repository*. [Online]. Available: https://data.nasa.gov/

[112] Cran-R. (2022). *Cran-R Project*. [Online]. Available: https://cran.rproject.org/web/packages/tree/tree.pdf

[113] Jira. (2022). *Jira System*. [Online]. Available: https://jira.lsstcorp.org/browse/DM-2157

[114] (2022). *Code Project*. [Online]. Available: http://www.codeproject.com/

[115] (2022). *Source Forge*. [Online]. Available: http://www.sourceforge.com/

[116] (2022). *Mendeley*. [Online]. Available: https://data.mendeley.com/datasets/m2zr2ns5k7/1

[117] R. Marco, N. Suryana, and S. S. S. Ahmad, "A systematic literature review on methods for software effort estimation," *J. Theor. Appl. Inf. Technol.*, vol. 97, no. 2, pp. 434–464, 2019.

[118] V. Cerqueira, N. Moniz, and C. Soares, "VEST: Automatic feature engineering for forecasting," *Mach. Learn.*, vol. 110, pp. 1–23, Apr. 2021, doi: 10.1007/s10994-021-05959-y.

[119] M. Humayun and C. Gang, "Estimating effort in global software development projects using machine learning techniques," *Int. J. Inf. Educ. Technol.*, vol. 2, no. 3, pp. 208–211, 2012.

[120] A. Kariluoto, J. Kultanen, J. Soininen, A. Pärnänen, and P. Abrahamsson, "Quality of data in machine learning," in *Proc. IEEE 21st Int. Conf. Softw. Qual., Rel. Secur. Companion (QRS-C)*, Dec. 2021, pp. 216–221.

[121] W. Hasselbring, "Benchmarking as empirical standard in software engineering research," in *Proc. Eval. Assessment Softw. Eng.*, 2021, pp. 365–372.

CHAUDHARY HAMZA RASHID received the B.E. degree in mechatronics engineering from the College of Electrical and Mechanical Engineering, NUST, Islamabad, Pakistan, in 2013, and the M.Sc. degree in management sciences from the University of Sargodha, in 2019. He is currently pursuing the M.S. degree in computer science with Abasyn University Islamabad Campus. His research interests include software project management, software cost estimation, and their application in software development.

IMRAN SHAFI received the bachelor's degree in aeronautical engineering from the College of Aeronautical Engineering, National University of Sciences and Technology (NUST), Pakistan, and the M.S. and Ph.D. degrees in computer engineering from the University of Engineering and Technology (UET) Taxila, Pakistan, in 2003 and 2009, respectively. He earned advanced certificates in disciplines of computer networks and databases from Stanford University, CA, USA, in 2011 and 2012, respectively. He is currently a Professor and the Chief Executive of the Vehicle Design and Manufacturing Laboratory at the College of Electrical and Mechanical Engineering, NUST. He works on high-resolution time–frequency signal processing and supervised neural networks. He has authored various book chapters and numerous articles in international peer-reviewed scientific journals and conferences of impact factor. His interests include artificial intelligence, vehicular designs, cognitive-communication, and digital signal processing. He is serving as a reviewer for prestigious IEEE, IET, Elsevier, and Springer journals and conferences.

JAMIL AHMAD (Senior Member, IEEE) received the M.Sc. degree in information technology from the University of Warwick, U.K., the M.Sc. degree in computer science from the University of Peshawar, and the Ph.D. degree in artificial neural networks from the Department of Electrical and Electronics Engineering, King's College London, U.K. He is currently the Head of Abasyn University Islamabad Campus, Pakistan. He received grants for his research and academic projects from various organizations, including the National Information and Communication Technology (ICT) Research and Development Fund, Pakistan, the HEC-British Council Linkage Program, and the U.K., Government under PMI 2 Program. He is a fellow of the British Computer Society, a Chartered Engineer (C.Eng.), and a member of IET and ACM. He was a recipient of the Charles Wallace Pakistan Trust Visiting Fellowship, U.K., from 2012 to 2013, and the International Visiting Leadership Program (IVLP) Fellowship, USA, from August 2007 to September 2007.

ERNESTO BAUTISTA THOMPSON is with Universidad Europea del Atlántico, Santander, Spain. He is also affiliated with Universidad Internacional Iberoamericana, Campeche, Mexico, and Universidade Internacional do Cuanza, Kuito, Bié, Angola.

MANUEL MASIAS VERGARA is with Universidad Europea del Atlántico, Santander, Spain. He is also affiliated with Universidad Internacional Iberoamericana Arecibo, Puerto Rico, and Universidad de La Romana, La Romana, Dominicana.

ISABEL DE LA TORRE DIEZ is currently a Professor with the Department of Signal Theory and Communications and Telematic Engineering, University of Valladolid, Spain, where she is also the Leader of the GTe Research Group (http://sigte.tel.uva.es). Her research interests include the design, development, and evaluation of telemedicine applications, services and systems, e-health, m-health, electronic health records (EHRs), EHRs standards, biosensors, cloud and fog computing, data mining, quality of service (QoS), and quality of experience (QoE) applied to the health field.

IMRAN ASHRAF received the M.S. degree (Hons.) in computer science from the Blekinge Institute of Technology, Karlskrona, Sweden, in 2010, and the Ph.D. degree in information and communication engineering from Yeungnam University, Gyeongsan, South Korea, in 2018. He was a Postdoctoral Fellow with Yeungnam University. He is currently an Assistant Professor with the Information and Communication Engineering Department, Yeungnam University. His research areas include positioning using next-generation networks, communication in 5G and beyond, location-based services in wireless communication, smart sensors (LIDAR) for smart cars, and data analytics.