Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 5782587, 17 pages
https://doi.org/10.1155/2022/5782587

*Review Article*

# Evolution of Software Development Effort and Cost Estimation Techniques: Five Decades Study Using Automated Text Mining Approach

**Anil Jadhav,[1] Mandeep Kaur,[2,3] and Farzana Akter[4]**

[1]*Symbiosis Centre for Information Technology, Pune, Maharashtra, India*
[2]*Department. of Computer Science, Savitribai Phule Pune University, Pune, Maharashtra, India*
[3]*Permtech Research Solutions, Pune, Maharashtra, India*
[4]*Department of ICT, Bangabandhu Sheikh Mujibur Rahman Digital University, Kaliakair, Gazipur, Bangladesh*

Correspondence should be addressed to Farzana Akter; farzana@ict.bdu.ac.bd

Received 28 February 2022; Accepted 6 April 2022; Published 2 May 2022

Academic Editor: Amandeep Kaur

Copyright © 2022 Anil Jadhav et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Software development effort and cost estimation (SDECE) is one of the most important tasks in the field of software engineering. A large number of research papers have been published on this topic in the last five decades. Investigating research trends using a systematic literature review when such a large number of research papers are published is a very tedious and time-consuming task. Therefore, in this research paper, we propose a generic automated text mining framework to investigate research trends by analyzing the title, author's keywords, and abstract of the research papers. The proposed framework is used to investigate research trends by analyzing the title, keywords, and abstract of select 1015 research papers published on SDECE in the last five decades. We have identified the most popular SDECE techniques in each decade to understand how SDECE has evolved in the past five decades. It is found that artificial neural network, fuzzy logic, regression, analogy-based approach, and COCOMO methods are the most used techniques for SDECE followed by optimization, use case point, machine learning, and function point analysis. The NASA and ISBSG are the most used dataset for SDECE. The MMRE, MRE, and PRED are the most used accuracy measures for SDECE. Results of the proposed framework are validated by comparing it with the outcome of the previously published review work and we found that the results are consistent. We have also carried out a detailed bibliometric analysis and metareview of the review and survey papers published on SDECE. This research study is significant for the development of new models for cost and effort estimations.

## 1. Introduction

Software development effort and cost estimation (SDECE) is a process of estimating the effort and cost required for software development and is one of the most important activities of software engineering. There exist several research papers on this topic. Some papers talk about the software development effort estimation (SDEE) [1–9] and the others talk about the software development cost estimation (SDCE) [10–15]. It is very common that the terms "software effort estimation" and "software cost estimation" have been used interchangeably in the literature. However, software cost estimation is an outcome of software effort estimation [16]. The ability of sales consultants, presales consultants, project managers, delivery managers, and delivery heads to determine accurate costs depends on the amount of detailing and care that has been taken to estimate efforts. Estimating accurate effort and cost have an important role in the success of the software project. Over the past five decades, there has been a significant increase in the complexity of software projects. This has led to the design and implementation of numerous techniques for estimating the effort and cost of the software development and its consequent discussion in literature.

Initial papers published on this topic discussed techniques such as COCOMO, SEER, PRICE-S, Checkpoint,

SLIM, Delphi Technique, and COCOMO II [17]. There also exist techniques based on function point analysis (FPA) [18–21], use case point analysis (UCP) [22–24], and Work Breakdown Structure (WBS) [25]. Several research papers discussed analogy-based approach [26–32] and case-based reasoning (CBR) approach [33, 34]. The techniques such as regression [35–37], artificial neural network (ANN) [38–41], and Fuzzy logic [31, 42–45] have become more popular in the recent past. Machine learning (ML) based techniques have also been used very widely in the literature [3, 16, 46–49]. A large number of research papers also discussed the optimization techniques used for SDECE [50–54].

Thus, there exist several studies on SDEE, SDCE, systematic reviews on SDEE, and systematic reviews on SDCE. However, there is a lack of research that analyzes research trends and techniques that have evolved in the last five decades. There is also a need to do a systematic bibliometric analysis of articles published on SDECE in the last five decades. Analyzing such vast research papers published on this topic in the last five decades is a very tedious and time-consuming task. Considering this fact, in this research paper, we proposed a generic text-mining-based framework to analyze a vast range of articles and investigate research trends and techniques used for SDECE in the last five decades. The framework is based on natural language processing and it is an automated process. The advantage of using a text-mining approach is that it significantly reduces manual efforts required to investigate research trends and patterns from the corpus of the documents on specific topics like SDECE [55, 56]. This has motivated us to conduct this research study based on the text mining mechanism. The proposed framework is very generic and can be used in any other domain where a large number of research articles are published and need to be investigated in a manner that may be similar to this study. In this study, we analyzed 1015 research articles indexed in the Scopus database. The objectives, research questions, and contributions of this study are as follows.

### 1.1. Research Objectives

(1) To propose a generic automated text-mining framework to analyze a large number of research papers, for identifying changing research trends in technologies, methodologies, frameworks, tools, and techniques in an identified area or topic of any scientific or social science field. This paper was carried out to understand how SDECE techniques have evolved in the last five decades.

(2) To investigate frequently used techniques, accuracy measures, and datasets for SDECE using the proposed text mining framework

(3) To validate the proposed framework to ensure consistent outcomes

(4) To do a systematic bibliometric analysis of studies on SDECE

(5) To do a comprehensive metareview of the review and survey papers on SDECE

### 1.2. Research Questions (RQ).

We attempt to address following research questions in this study.

RQ1: What are the most frequently used SDECE techniques? What is research trend? How have SDECE techniques evolved in the last five decades?

RQ2: What are the most frequently used datasets in SDECE studies?

RQ3: What are the most frequently used accuracy measures in SDECE studies?

RQ4: What is the distribution of SDECE papers and their citations by document type?

RQ5: How many research papers are published on SDECE each year and in each decade since 1970?

RQ6: What is the distribution of citations of SDECE papers? This research question is further divided into sub-questions as follows:

RQ6.1: What is the distribution of journal and conference papers with zero citation and papers with one or more than one citation?

RQ6.2: What are highly cited papers?

RQ7: Who are the top authors in terms of the number of papers and number of citations?

### 1.3. Research Contributions

(1) We propose a generic text mining framework to investigate research trends by analysis title, keywords, and abstract of the research papers. This will help researchers from any domain to investigate research trends and patterns in an identified topic of the study.

(2) We have used a proposed framework to investigate the most frequently used techniques, datasets, and accuracy measures for SDECE in the last five decades. Decade-wise analysis is also done to understand the evolution of SDECE techniques in the last five decades.

(3) The study presents a comprehensive metareview of the review and survey papers on SDECE, which enables researchers to understand research trends and the contribution of researchers in this field.

(4) The study presents a comprehensive citation analysis of select 1015 Scopus indexed research papers published on SDECE during the period between the year 1974 and 2020.

The paper is organized as follows: in the second section, we present the research method. The third section presents a metareview of the review and survey papers. Results of the automated text mining framework and bibliometric analysis are presented in the fourth section. In the fifth section, we validate the results of the proposed text mining framework. The threats to the validity of the study are explained in the sixth section. Finally, we conclude paper in the seventh section.

## 2. Research Method

To achieve the stated objectives of this study we have analyzed select 1015 articles from the Scopus database. We had three options to select articles on SDECE from indexing databases including Scopus, Web of Science, and Google scholar. These are the three most popular and widely used online indexing databases by researchers. We decided to use the Scopus database because we could download all required data about research articles, such as the title of study, year of publication, the number of citations, source of the article (Journal, conference, etc.), author's keywords, abstract, document type, and authors information in CSV file format.

The search string used for finding the documents from Scopus database was decided by taking into account the objectives and research questions of the study. The search terms used were "software effort estimation" OR "software cost estimation". We used this search string because it was needed to limit the study to the research papers that discuss software effort estimation and software cost estimations. The search of documents was done on 23 May 2020. We exported the search results in CSV (excel) file format. In the title column of the extracted data, we found that some of the titles were not research papers but the titles belonged to conferences, symposiums, and workshops. So, we decided to remove those titles from our list. We removed the following type of titles from the search results: (i) 48 conference titles; (ii) 6 symposium titles; (iii) 2 annual conventions titles; (iv) 6 international work-shop titles; (v) 1 conference review title; and (vi) 1 conference note. We also found that in search results there were 26 non-English papers, so we removed those papers from the list. Thus, in total we removed 90 titles from the original search results and selected 1015 papers for purpose of this study. Later, by reading the title of the research papers, we checked whether all selected 1015 research papers are on SDECE and we found that all of them were relevant. Thus, other than the criteria that the paper should be on SDECE and written in the English language, we did not use any exclusion criteria.

The analysis to investigate research trends and techniques used for SDECE was done separately for title, keywords, and abstract of the research papers. We did the analysis separately because we wanted to check whether the results of the analysis based on the title, abstract, and keywords of the research papers are consistent or not. We used "wordcloud" and "tm" packages in "R" programming language for the text mining task. The bibliometric analysis of 1015 is done using the following information of the research papers: title; authors; year of publication; source title; cited by; affiliation; and document type. The detailed results of both bibliometric analysis and text mining are presented in Section 4.

## 3. Metareview of Review and Survey Papers on SDECE and Related Work

Several studies have been published on the topic SDECE in the last five decades. In this section, we present a detailed metareview of review and survey papers. Out of the selected 1015 articles, we found 39 review/survey papers, which include 13 journal articles, 25 conference papers, and one book chapter. Among the 39 review/survey papers, 9 papers that were published in year 2018 and 2019 did not receive any citation till May 2020. The remaining 30 papers received a total 1636 citations. The main findings of the 39 review/survey papers are presented in Table 1. For some studies, data such as duration of the study and number of papers reviewed were not available so we could not include those details in Table 1.

Some existing studies have used a text mining approach for identifying research trends in different areas. Garousi and Mantyala [55] used text mining to identify research themes, hot and cold topics in software engineering. The study conducted by Nie and Sun [85] used text mining to identify major academic branches and identify research trends in design research. Sehra et el. [56] conducted a study to identify research patterns and trends in software effort estimation using a text mining approach. The study was conducted by applying text mining on articles published during the period between 1996 and 2016. In all these studies, it is found that usage of the text mining is an adequate choice for better assessment when large number of articles needs to be assessed to understand the research trends, research themes, hot and cold topics in an identified research area. However, we found that there is a lack of research on (i) investigating research trend in SDECE in the last five decades; (ii) identifying the most popular SDECE techniques in each decade to understand how SDECE techniques have evolved in the last five decades; (iii) investigating research trends by analyzing title, keywords, and abstract of research papers separately to understand whether results are consistent or different; (iv) metareview of the review and survey papers published on SDECE in the last five decades; and (v) bibliometric analysis of the papers published in the last five decades. Therefore, in this study, we attempt to fill these gaps.

Based on a metareview of the review and survey papers, we have identified the most used (i) SDECE techniques, (ii) datasets, and (iii) accuracy measures for SDECE. The most used SDECE techniques are shown in Figure 1. The most used datasets and accuracy measures are given in Table 2.

The strengths and weaknesses of the commonly used SDECE techniques:

Based on the review of literature and our understanding about SDECE techniques, we present the strengths and weaknesses of the most used SDECE techniques.

(A) Linear regression: The strengths are as follows: (i) easy to understand, implement, interpret, and explain; (ii) can work well with small datasets; (iii) computationally not very expensive. The weaknesses are as follows: (i) assumes linear relationship and therefore not suitable when nonlinear relationship exists in the data; (ii) multicollinearity issue needs to be resolved, if it exists; (iii) sensitive to the outliers.

(B) Artificial neural network: The strengths are as follows: (i) it can learn complex relationship in the

Table 1: Findings of review and survey papers on SDECE.

| Research paper | Findings |
|---|---|
| | Journal papers |
| [17] | This study classifies cost estimation models into five different categories along with detailed explanation of each category. The techniques are classified as i) Model based approaches: SLIM, COCOMO, checkpoint, SEER; ii) expertise based models: Delphi, rule based; iii) learning based models: ANN, robust; iv) regression models: OLS and robust; and v) composite models: Bayesian and COCOMO II |
| [10] | The study reviewed 304 papers from 76 journals. Research papers published before April 2004 were included in the study by manual search. Focus of the review was to classify papers based on research topic, research approach, SDEE technique, and datasets used for the study. The study also listed important cost estimation journals, research topics, research approaches, estimation approaches, context of the study. |
| [16] | The study reviewed 84 articles during the period 1991 to 2010. Four different aspects of ML models were reviewed: ML technique; accuracy of estimation using ML technique; comparison of ML models; and estimation context. Finding of study are that accuracy of ML models is better than non-ML models/techniques. |
| [37] | Conducted systematic empirical analysis of 10 local and global models of SEE. Study found that the results obtained are different for local and global methods of SDEE because of different experiment design and datasets. |
| [41] | Reviewed 21 articles describing neural network based models for SEE. The study reports range of features used for SDEE using ANN. The important finding of the study are as follows: i) ANN gives better results compared to regression, classic COCOMO model, SLIM FPA; ii) most of the researcher used COCOMO dataset; iii) the most used accuracy measures are MMRE, MdMRE, MRE, pred, MMER; iv) the most used neural network is feed forward neural network; |
| [57] | The study reviewed 129 articles during the period 2000 to 2014 and discussed usefulness and limitations of the ISBSG dataset used for SEE. About 70% papers used ISBSG dataset for SDEE and 36% papers used ISBSG dataset to study its properties. 55% papers used ISBSG dataset and others used complementary datasets for SEE. The study also highlighted that the most common methods used for SDEE are regression, machine learning, and estimation by analogy. |
| [58] | Review period of this study was from 1991 to 2016. The study reported that because of changing nature of the software development and its complexity several estimation techniques are evolved. The study also reported that for improved results several data mining and machine learning techniques are used along with conventional methods of SEE. |
| [59] | The study reviewed 101 articles during the period 2006 to 2015. The study reviewed papers related to cost estimation using agile software development. The study reported most popular SDEE techniques, accuracy measures, and project success rate over the years. ANN and expert judgment are the most used techniques for agile SDECE. MRE, MMRE, MdMRE, and pred are most used accuracy measures. |
| [60] | Review period was from 2000 to 2017. The articles are reviewed with respect to type of soft computing or machine learning techniques used for SEE. The study reported that COCOMO, NASA, ISBSG, DEHANAI are the most used datasets and MMRE and PRED are most used evaluation metrics. It is also reported that ANN is most used estimation technique. |
| [61] | The study analyzed 20 papers on SDCE tools. The review concluded that most of the tools are based on COCOMO model. |
| [3] | The study reviewed models built using ML techniques for SEE. The study reviewed 75 papers during the period 1991 to 2017. The study found that i)ANN is widely used ML technique; ii) MMRE is widely used accuracy measure; iii)ANN and SVM outperformed the other techniques; iv) Regression is non-ML technique widely used for effort estimation. |
| [62] | The study reviewed 74 articles from the period 2000 to 2017. Eight types of techniques found to be used for SEE. The study found that i) most used datasets are ISBSG, COCOMO, NASA93, NASA, desharnias, albercht, sdr, China, kemerer, miyaki, maxwell, Finnish. ii) Most widely used methods are ANN, CBR, linear regression, fuzzy logic, GA, kNN, support vector regression, logistic regression, and decision tree. iii) Most used accuracy measures are MMRE, MdMRE, PRED. |
| [63] | Discussed issues of estimating cost of software projects. |
| | Conference papers |
| [64] | Paper reports survey results on SDEE technique used in JPL laboratory. It is found that i) most technical staff use informal analogy and high level partitioning of requirements, and ii) staff was better in estimating effort than size. |
| [5] | The findings are based on surveys on SDEE and the findings are as follows: i) 60–70% projects encounter effort or schedule overrun; ii)30–40% projects encounter cost overrun; iii) frequent method used for estimation is expert's judgment. |
| [65] | The study analyzed 112 projects from Chinese software industry. The survey investigated estimation methods, accuracy of method, and factors influencing adoption of certain method. The main findings are as follows: i) The large projects are prone to cost and schedule overrun, and ii) about 15% organizations used model based methods. |
| [66] | Paper provides compressive overview of analogy based SEE. Paper also discussed analogy based tool and systems, dataset quality and its relevance in predicting SEE. |
| [67] | This study reports the review of three parametric models used for SDEE namely: SLIM-putnam 1979, SEER-SEM 1989, SPR-knowledge plan 1999. |
| [68] | This study reports result of survey analysis on SDEE from industry perspective such as abilities of software organizations to apply SDEE technique and actually use techniques for effort estimation. The study also reports requirement of SDEE identified on the basis of survey and are compared with the requirements of existing methods. |

Table 1: Continued.

| Research paper | Findings |
|---|---|
| [69] | This study reports cost and schedule estimation approaches for component-based software development. Analysis of published work is done with respect to modeling techniques, data requirement, type of estimation, and lifecycle activities. |
| [70] | This survey reports results of reliability of expert's judgment for SDCE in a medium sized software company. The study also reported that cost estimation based on expert's judgment is unreliable. |
| [71] | The study reports overview and usefulness of ANN for SDEE and its accuracy. The study reviewed 19 articles from the period 2000 to 2014. |
| [72] | The focus of review was to determine whether use of feature weighting technique (FWT) in CBR improves SDEE prediction accuracy. The study concluded that use of FWT in CBR improves SDEE prediction accuracy. |
| [73] | The study reviewed articles pertaining to SDEE and concluded that every technique has its own advantages and disadvantages and there is no globally accepted single technique for SEE. The study reviewed 167 papers from the period 2000 to 2013. |
| [74] | The study reports statistics about usage of variables in ISBSG dataset for SEE. The study found that variables with missing values are less frequently used. |
| [75] | The study reviewed 16 articles only. Reviewed articles were classified using nine criteria for global software development (GSD). It is found that the dominant contribution of GSD research was the models and software development cost. |
| [76] | The study reviewed article on the tools and frameworks developed for SDEE using use case point model. |
| [59] | This study reviewed various soft computing techniques such as genetic algorithm, neural networks, fuzzy systems, particle swarm optimization used for SDEE in agile software development. The study found that soft computing techniques provide better accuracy estimation. |
| [77] | The study reviewed 15 articles. The findings of the study are as follows: i) company's use expert's judgment for SDEE; ii) need to improve algorithms and prediction techniques. |
| [78] | The study reports 8 common approaches used to find $k$ value for analogy based SDEE techniques. It is reported that due to varied performance of different approaches in finding $k$ values resulted in conflicting results. |
| [79] | The study conducted review of article to find which estimation method is best. It is found that use case point analysis approach is better that function point analysis and COCOMO model. |
| [80] | The study reviewed 10 articles. The survey analyzed contribution of papers in estimating effort with respect time, cost, and test. The study found that supervised learning algorithms are most popular for effort estimation. |
| [81] | The study reviewed 41 papers on ML based SDEE from the period between 2000 and 2017. The study discussed ML techniques, size metrics, benchmark datasets, and validation methods for SEE. It is found that i) most used techniques: Fuzzy logic, ANN, GA, anlogy based, SVR, bayesian network, regression tree, CBR; ii) dataset used: NASA, ISBSG, albrecth, COCOMO, desharnais, kemerer, kotengray, maxwell; iii) performance measures: MRE, MMRE, pred, MdMre, MMER, MSE, RMSE, standard deviation. |
| [82] | The review was conducted to understand importance of nonfunctional requirement in SDEE. The study identified nonfunctional requirements used in SDEE and how they are used. It is also found that use of nonfunctional requirements in SDEE brings down error by 30%. |
| [83] | The study reviews cost estimation techniques and presents strength and weakness of the techniques. |
| [4] | The study reviewed use case-based effort estimation methods and provides factors contributing to use case effort estimation. Provides inputs on criteria to evaluate accuracy and effectiveness of the models. |
| [3] | The study reviewed 30 articles on 'bio-inspired feature selection algorithms' during the period 2007 to 2018. It is fount that genetic algorithm (GA) and particle swarm optimization (PSO) are widely used bio-inspired algorithms. Results of GA and PSO are better than baseline estimation techniques. |
| [21] | The study discussed limitations and accuracy of the function point analysis method. |
| [84] | The study: i) Reviewed papers that describes models, processes, and practices and ii) proposed a general prediction process and framework for selecting predictive measures. |

data; (ii) feature engineering is not required to be done. The weaknesses are as follows: (i) large amount of data is required for training, therefore it is computationally expensive; (ii) it is difficult to understand the reasoning behind the results, so interpretation of the results is difficult; (iii) it may suffer from over-fitting problem; (iv) cannot deal with missing values; (v) categorical values need to be converted to the numeric type.

(C) Analogy-based approaches: the strengths are as follows: (i) Easy to understand the reasoning behind the outcome; (ii) can deal with outliers. The weaknesses are as follows: (i) computationally intensive; (ii) sensitive to the similarity function; (iii) categorical variables need to be converted to numeric type; (iv) cannot handle missing values; (v) difficult to get the solution if similar work has not been done in the past.

(D) Fuzzy logic: The strengths are as follows: (i) It is based on the theory of classes with soft boundaries so it can deal with uncertainty in the data caused by measurement error during data collection; (ii) it can also deal with uncertainty in the model; (iii) gives improved performance if combined with ML or non-ML models; (iv) it is similar to human reasoning process. Its only weakness is that it becomes computationally intensive when combined with ML or non-ML models.

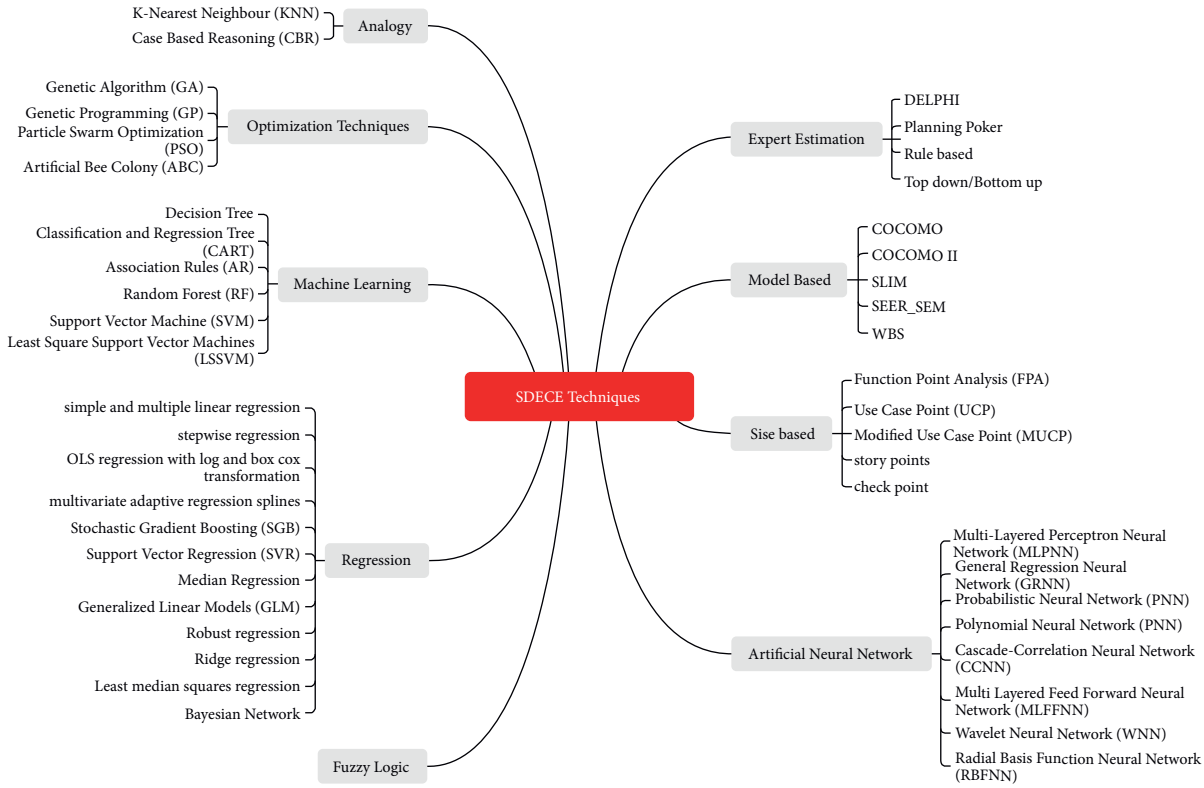



Figure 1: The SDECE techniques based on review.

Table 2: The datasets and accuracy measures.

| |
|---|
| The most commonly used accuracy measures: magnitude of relative error (MRE), mean magnitude relative error (MMRE), median magnitude relative error (MdMRE), percentage relative error deviation (PRED) |
| The most commonly used datasets: NASA, ISBSG, COCOMO, albrecht, desharnais, kemerer |

(E) ML (Machine learning) techniques: Tree-based model and SVM are the most used ML techniques for SDECE. Each ML technique has its own strengths and weaknesses. The decision tree (DT), CART, and RF techniques fall under tree-based ML models, and its strengths are as follows: (i) intuitive, so easy to understand and interpret the results of model; (ii) can handle both categorical and numeric data; (iii) suitable when nonlinear relation exists in the data; (iv) robust with the outliers or we can say it has the capability to deal with the outliers. Its weaknesses are as follows: (i) DT is prone to overfitting if the dataset is small; (ii) cannot deal with missing values; (iii) time complexity is high for a large dataset; (iv) a small change in data can have a large change in the model.

The strengths of SVM are as follows: (i) It is suitable for high dimensional data; (ii) learns nonlinear relationship in the data. The weaknesses are as follows: (i) memory intensive; (ii) may not scale well with large datasets.

(F) Optimization techniques: the strengths are as follows: (i) it is useful for feature weighting and feature selection; (ii) accuracy of the estimation improves if used in combination with ML or non-ML techniques. The weaknesses are as follows: (i) it is a nondeterministic approach, the so results may vary each time; (ii) computationally expensive.

(G) Model and size-based estimation: The strength is that it is very useful for project planning, control, and budgeting. Its weakness is that it is based on calibration of the past experience. Difficulty in estimation arises with unprecedented situation.

(H) Expertise-based estimation: Its strengths are that it is very useful when no quantifiable or empirical data is available. Its weakness is that it is purely based on knowledge and experience of the expert, so estimation is just opinion and it can be biased and may go wrong.

## 4. The Generic Automated Text-Mining Framework and Bibliometric Analysis

This section is divided into two parts: first part explains the generic automated text-mining framework to study the evolution of SDECE in the last five decades and the second part presents a bibliometric analysis of the selected 1015 research papers.

*4.1. The Generic Automated Text-Mining Framework for Identifying Research Trends and Patterns.* In this section, we present the generic automated text-mining framework and use it to investigate research trends and techniques used for SDECE by analyzing the title, abstract, and author's keywords of the selected 1015 research papers published in the last five decades. The framework is shown diagrammatically in Figure 2.

The text mining is applied to (i) title, (ii) abstract, and (iii) authors' keywords of research papers. We have used "tm," "RWeka," and "wordcloud" package in the "R" tool. The steps used for text mining are as follows:

Step 1: The title of research papers was first loaded in "R" from the CSV file downloaded from the Scopus database

Step 2: We then created a corpus of the documents, where each title is treated as a separate document

Step 3: The third step was text cleaning, and we performed following text cleaning tasks:

(i) Converting text to lower case

(ii) Removing punctuations, whitespace, numbers, and special characters from the text

(iii) Removing the stopwords. Stopwords are the words that occur very frequently in the document, such as "the," "this," "and", but do not help in extracting any meaningful insights from the text data

Step 4: The next step was to create tokens of the words and find their frequency using "NgramTokenizer" function in "Rweka" package in 'R'

Step 5: The last step was to create WordCloud using word-frequency table created in the third step

We repeated the above process for the abstracts and authors' keywords of the selected 1015 papers. We also performed decade-wise analysis using title, abstract, and keywords of the research papers published in each decade. We stored the results of Step 3 (word frequency table) in CSV file format so that we could cross-check WordCloud and word-frequency table.

We address the following research questions using the proposed framework:

RQ1: What are the most frequently used SDECE techniques? What is research trend and how SDECE techniques have evolved in the last five decades?

RQ2: What are the most frequently used datasets in SDECE studies?

RQ3: What are the most frequently used accuracy measures in SDECE studies?

Results of the text mining (i) using all 1015 papers and (ii) papers published in each decade are shown in Table 3. The first column in the table shows WordCloud using title, second column shows WordCloud using authors' keywords, and the third column shows WordCloud using the abstract of the research papers. The prominent words in each WordCloud are given just below the WordCloud for better understanding. These prominent words indicate the most commonly used/referred/discussed techniques for SDECE. The first row in Table 3 shows WorldCloud of the papers published between the year 1971 and May 2020. The other rows in Table 3 show WordCloud of research papers published in each decade.

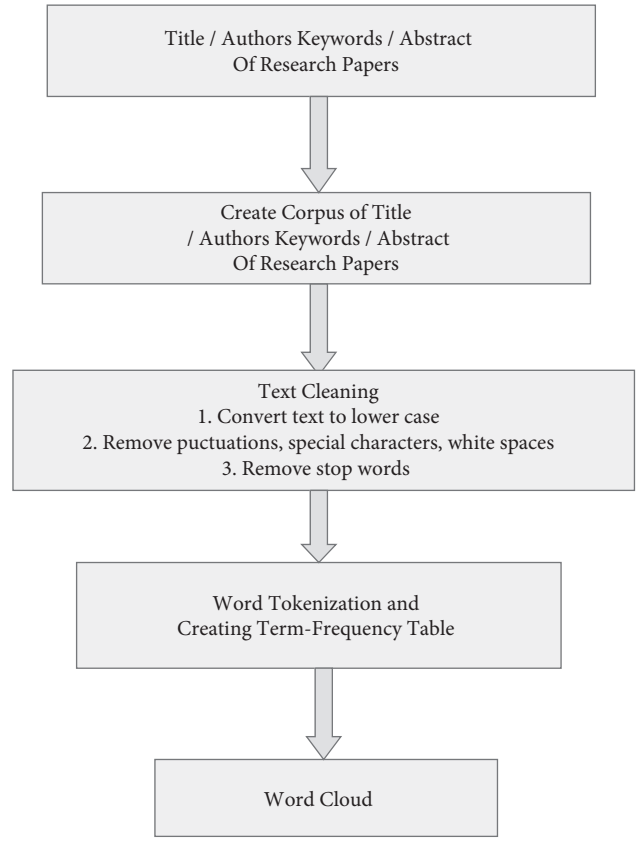



Figure 2: A Generic Text-Mining Framework to investigate changing research trends.

From Table 3, it is observed that the five most common techniques used for SDECE in the last five decades (1971 to May 2000) are fuzzy logic, artificial neural network, regression, analogy-based approach, and COCOMO model. The results also show that the other commonly used techniques are optimization, use case points, function point analysis, machine learning, COCOMO II, and CBR. There is a small variation in the most common techniques identified based on the analysis of the title, keywords, and abstract of the research papers. The SDECE technique mentioned in the title of the research paper generally indicates that the technique is proposed or used in the research paper for SDECE, whereas the techniques listed in the authors' keywords and abstract of the research paper may indicate that the technique is either proposed/used/referred/discussed in the paper or compared with other existing techniques. Therefore, we strongly believe that the title-based text mining approach gives us information about the technique proposed/used by the researcher for SDECE, whereas, the keywords- and abstract-based text mining results give us information about the most discussed/proposed/used/referred technique or it is compared with the other techniques.

Table 3: The text-mining results using the title, keywords, and abstracts of the research papers.

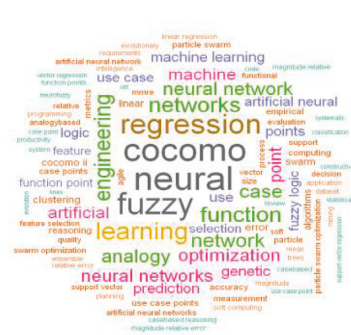
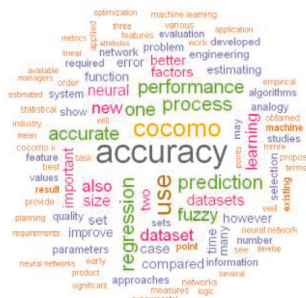
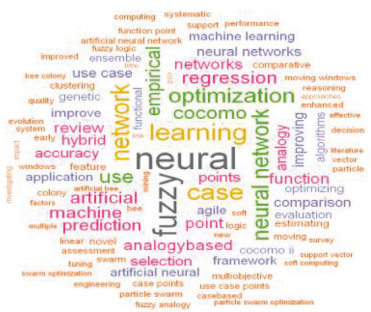
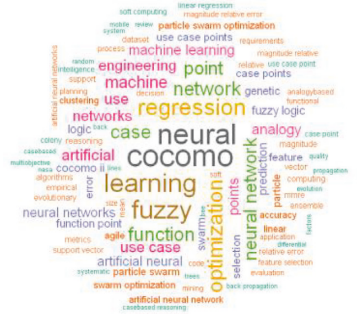
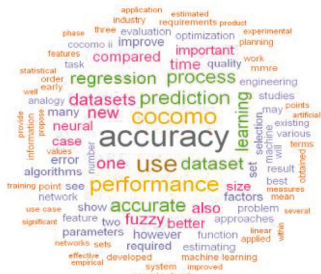
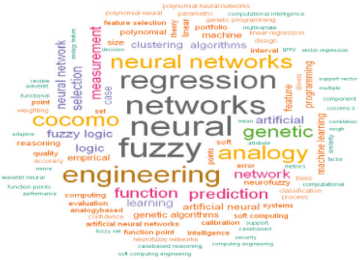
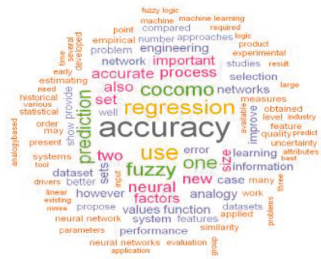
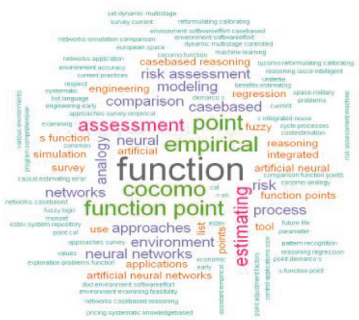
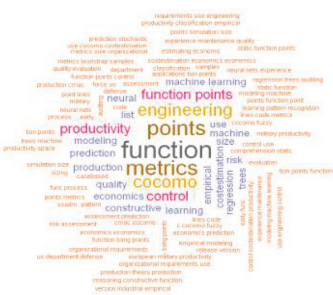
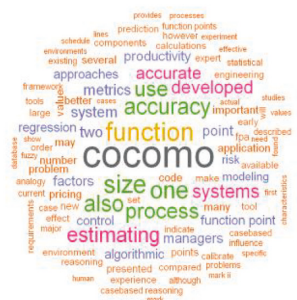
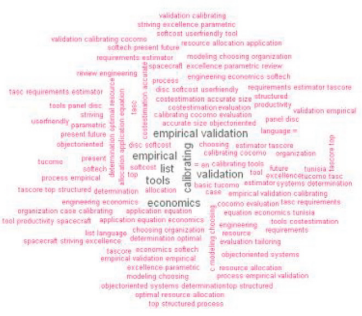
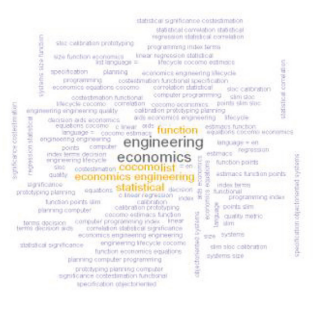
| Title | Keywords | Abstract |
|---|---|---|
| Papers between period 1971 and 2020 (number of papers: 1015) | | |
|  |  |  |
| Techniques: Fuzzy logic, ANN, regression, analogy based approach, cocomo, optimization, use case points, machine learning, function point analysis, cocomo ii, CBR, particle swarm optimization, feature selection, support vector. | Techniques: ANN, cocomo, fuzzy logic, regression, analogy based approach, optimization, ML, use case points, function point analysis, cocomo ii, clustering, particle swarm optimization, CBR, support vector, feature selection, soft computing, trees, ensemble, support vector, Dataset:NASA, metrics: MMRE, MRE | Techniques:cocomo, regression, fuzzy logic, ANN, analogy based apporach, ML, cocomo ii, optimization, use case points, clustering, feature selection, function point analysis, CBR, ensemble, particle swarm optmization, trees, support vector, dataset: ISBSG, NASA, Metrics:MMRE, pred, MRE |
| Papers between period 2011 and 2020 (number of papers: 629) | | |
|  |  |  |
| Techniques: ANN, fuzzy approach, optmization, cocomo, regression, analogy based, approach, use case points, ML, cocomo ii, GA, ensemble, function point, particle swarm optimization, CBR, artificial bee colony, dataset: ISBSG | Techniques: Cocomo, ANN, fuzzy approach, regression, optimization, use case points, analogy, machine learning, function point analysis, GA, cocmo ii, particle swarm optimization, clustering, support vector, metrics: MMRE, MRE, dataset: ISBSG, NASA | Techniques: Cocomo, regression, fuzzy approach, neural, optimization, cocomo ii, analogy based approach, machine learning, use case points, swarm, function point, Metrics: MMRE, pred, Dataset:ISBSG, NASA |

Table 3: Continued.

| Title | Keywords | Abstract |
|---|---|---|
| Papers between period 2001 and 2010 (number of articles: 306) | | |
| Title | Keywors | Abstract |
|  |  |  |
| Techniques: Fuzzy, ANN, regression, analogy based, cocomo, clustering, function point, soft computing, GP, GRA, machine learning, radial basis function, use case point, cocomo ii, CBR, metrics: MMRE | Techniques: ANN, fuzzy approach, regression, analogy, cocomo, GA, clustering, ML, function point, soft computing, genetic programming, linear regresion, polynomial NN, CBR, cocomo ii, SVR, metrics: MMRE, dataset: ISBSG | Techniques: Regression, fuzzy, cocomo, neural, analogy, ML, cocomo ii, function point, use case point, metrics: MMRE, dataset: ISBSG, NASA |
| Papers between period 1991 and 2000 (number of articles: 57) | | |
|  |  |  |
| Techniques: Cocomo, function point, analogy, case based, ANN, fuzzy, regression | Techniques: COCOMO, function point, ML, ANN, regression, trees | Techniques:Cocomo, function point, regression, analogy, case based, fuzzy logic, cocomo ii |
| Papers between period 1981 to 1990 (number of articles-21) | | |
|  |  |  |
| Techniques: Calibrating, economics, empirical | Techniques: Economics, engineering, cocomo | Techniques: Process, productivity, tool, cocomo, calibration |

However, the top five techniques based on the text analysis of the title, authors' keywords, and abstract of the research paper are the same.

*4.2. Evolution of SDECE Techniques.* The text-mining results based on the title of the research papers published during the period between 2011 and May 2020 show that ANN, fuzzy approach, optimization, COCOMO, and regression are the most used techniques. The text-mining results based on keywords and the abstract of the research papers show that COCOMO is the most discussed technique. The other widely used techniques are analogy-based approach, use case point, function point analysis, machine learning, COCOMO II, and GA.

The text-mining results for the period between 2001 and 2010 show that fuzzy approach, ANN, regression, analogy-based, and COCOMO are the most used techniques. It is also observed that regression is the most discussed technique based on text analysis of the abstract. The title-based text mining results show that fuzzy-based approach is the most used technique, which is followed by ANN, regression, analogy, and COCOMO. The other widely used techniques during this period are clustering, function point analysis, soft computing, GP, machine learning, use case point, and COCOMO II.

The text-mining results for the period between 1991 and 2000 show that COCOMO and function point analysis are the most used techniques followed by analogy, CBR, ANN, regression, fuzzy logic, and ML techniques. As there exist very few select research papers (21) in our study for the period between 1981 and 1990, the WordCloud does not have a large set of words. However, the results show that COCOMO was the most popular method during this period. We did not apply text mining to the research papers published during the period between 1971 and 1980 because out of the total 1015 selected papers, we had only two papers for that period. The number of research papers published during this period may be large but we found only two papers in the set out of the selected 1015 papers.

Thus, the text mining results show that (i) for the initial period between 1981 and 1990 focus of the research was on calibration and productivity and COCOMO was the most used technique; (ii) during the period between 1991 and 2000 COCOMO became more popular and researchers also proposed functions point analysis, regression, analogy-based approach, CBR, and fuzzy-based techniques; (iii) fuzzy approach, ANN, and regression-based approaches became more popular during the period between 2001 and 2010 followed by COCOMO, analogy-based approach, clustering and ML techniques; (iv) during the period between 2011 and 2020, ANN-based approach was more popular followed by fuzzy logic, optimization, COCOMO, regression, and analogy based approach. Use case point, function point analysis, and machine learning were other popular techniques during that period; (v) for the last fifty years, i.e., for the period between 1971 and May 2020, the most popular technique based on analysis of the studied research papers are fuzzy logic, ANN, regression, analogy-based approach, COCOMO followed by optimization, use case point, function point, ML, COCOMO II, clustering, and CBR-based approaches.

We can map the evolution of SDECE techniques with the evolution of the programming languages. In the initial period (1970 to 1990) COCOMO was the most popular model because software systems were being developed using the assembly and procedure-oriented programming languages and COCOMO model is based on a number of lines of code written to develop the software system.

In the later stage (1991–2000), function point analysis, regression, analogy-based approaches became more popular because by that time a large number of software projects data was recorded and available for the estimation of the newly developed software systems. Regression is a statistical technique, which is used to estimate efforts and cost using historical software projects data, whereas in analogy-based approach, efforts are estimated by considering efforts required for similar systems/projects developed in the past. Function point analysis also became more popular because software systems were being developed using functional programming languages.

Later, during the period between 2001 and 2010, fuzzy logic and ANN techniques became more popular. Fuzzy logic was popular because it takes into account vagueness and imprecise information, and ANN was popular because some researchers were of the opinion that ANN gives more accurate estimation than the existing techniques. During the same period due to the emergence of machine learning techniques and the availability of the existing projects' data, people also started using different ML techniques for effort and cost estimation. Since researchers started using ML techniques, optimization also became more popular as it helped in selecting the most appropriate features. Also, as systems were being developed using object-oriented programming language, scholars started using use case point techniques for SDECE.

For the period between 2010 and 2020, researchers started using existing techniques in combination with the other existing techniques for better estimation. In the recent past, scholars have used deep learning techniques for prediction in other domains, but there is a lack of research in using deep learning techniques for SDECE. Therefore, we recommend that scholars should explore this option for SDECE.

We have also identified the most common datasets and accuracy measures used for SDECE. As researchers rarely use the dataset name and accuracy measures in the title of the research article, finding the most frequently used datasets and accuracy measures by applying text mining to the title of the research papers is difficult. However, researchers use the dataset name and accuracy measures in authors' keywords and abstract of the research papers. Therefore, we could find the most frequent datasets and accuracy measures by applying text mining on authors' keywords and the abstracts of the research papers. The most frequently used datasets and accuracy measures for each decade and for the period between 1974 to May 2020 are also given in Table 3. It is observed that (i) NASA and ISBSG are the most used datasets; and (ii) MMRE, MRE, and PRED are the most used accuracy measures for the period between

1971 and May 2020, and also for each decade starting from 1970s to 2020.

Using the text mining, we have also identified whether focus of the research was on the SDEE or SDCE. The results of the text mining for the same are presented in Table 4. The results show that (i) for the initial period from 1981 to 1990 focus of the research papers was on the cost estimation; (ii) for the period between 1991 to 2000 and 2001 to 2010 the focus was on cost estimation followed by effort estimation; (iii) for the period between 2011 and 2020 most studies discussed effort estimation than the cost estimation; (iv) for the past five decades, from 1974 to May 2020 most studies discussed effort estimation than the cost estimation. However, it is important to note that some researchers use these two terms interchangeably.

*4.3. Bibliometric Analysis.* In this section, we present the bibliometric analysis of select 1015 research papers published during the period between 1974 and May 2020 to address the research questions from RQ4 to RQ7.

RQ4: What is the distribution of SDECE papers and its citations by document type?

The distribution of the selected papers by document type is given in Table 5. The contribution of the journal and conference documents is 39.11% and 59.41%, respectively. However, in terms of the number of citations, journal papers received more citations (68.62%) as compared to the conference papers (31.04%). The contribution of the books and book chapters in terms of the number of papers as well as citations is very less.

RQ5: How many research papers are published on SDECE each year and each decade since 1970?

The distribution of papers published in the last five decades is given in Table 6. It is observed that about 92% of the papers were published in the last 2 decades. Figure 3 shows the graph of a number of papers published in each year since 1974. As we have included paper till May 2020, the number of papers published in the year 2020 is less as compared to the year 2019.

RQ6: What is the distribution of citations of SDECE papers?

This research question is further divided into five subquestions as follows.

RQ6.1: What is the distribution of journal and conference papers with zero citation and with one or more than one citation?

The number of citations of the research paper plays an important role in deciding the influence or impact of the research paper. Based on articles selected for this study, the count of citations for journal and conference articles is given in Table 7. It is observed that 22.36% of the papers have received zero citations. The proportion of the journal and conference articles having zero citations is 19.89% and 23.21%, respectively.

RQ6.2: What are highly cited papers?

We have identified highly cited papers using an average annual number of citations received by the paper per year since its publication. The top 5 articles based on the average annual number of citations are shown in Figure 4. The top five articles are published in the journal. It is also found that out of the top 10 articles, all articles were from the journal except one at the seventh position.

RQ7: Who are the top authors in terms of the number of papers and number of citations?

The contribution of authors is measured using two metrics: (i) number of articles published by the author and (ii) number of citations received by the author for all his articles selected in this study. The top ten authors based on these two metrics are given in Table 8. We have also created a WordCloud of authors using the authors column of the CSV file of select 1015 papers. The resulting WordCloud (Figure 5) of the author's contribution based on the number of papers matches with the manual calculations of the number of papers published by the top author (Refer Table 8).

The bibliometric analysis of select 1015 papers shows that (i) impact of journal papers in terms of the number of citations is more than the conference papers, though the number of conference papers is more than the journal papers; (ii) IEEE transaction on software engineering, Information and Software technology, Journal of Systems and Software are the top journal sources for the research on software development effort and cost estimation; (iii) Jorgensen, Boehm, and Shepperd are the most cited researchers whereas Idri, Angeles, and Keung have published the maximum number of research papers on SDECE.

## 5. Validation of the Framework

In this section, we validate the results of the proposed automated text mining framework by comparing it with the (i) results/outcome of the comprehensive systematic literature reviews (SLRs) done in the past; and (ii) results obtained manually by reading the title of all selected 1015 research papers.

*5.1. Validation Using past SLRs.* A summary of the results based on five selected comprehensive systematic literature reviews conducted in the past is shown diagrammatically in Figure 6.

The five SLRs conducted in the past show that (i) regression, ANN, fuzzy logic, analogy-based approach, CBR, DT, SVR, GA, and GP are the most used techniques; (ii) MRE, MMRE, Pred, and MdMRE are the most used accuracy measures; and (iii) NASA, ISBSG, and COCOMO are the most used datasets for SDECE.

Out of five comprehensive SLRs, three studies show that regression is the most used non-ML technique for SDECE. Two studies that reviewed research papers published between the year 2000 and 2017 show that (i) the most used SDECE techniques are regression, ANN, DT, fuzzy logic, analogy, and CBR-based approaches; (ii) the most used datasets are NASA, ISBSG, and COCOMO; and (iii) the most used accuracy measures are MRE, MMRE, and Pred.

*5.2. Validation by Reading Title of the Research Papers.* The results obtained manually by reading the title of the selected 1015 research papers are shown in Figure 7. The

Table 4: What is research focus: effort or cost estimation?

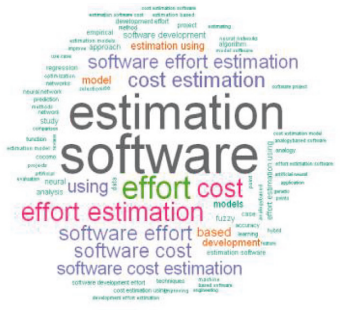
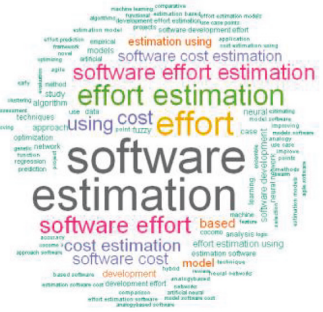
| Period: 1974–2020 | Period: 2011–2020 | Period: 2001–2010 |
|---|---|---|
|  |  |  |
| Effort estimation, cost estimation | Effort estimation, cost estimation | Cost estimation, effort estimation |
| Period: 1991–2000 | Period: 1981–1990 | |
|  |  | |
| Cost estimation, effort estimation | Cost estimation | |

Table 5: Distribution of papers and citations by document type.

| Document type | Articles | Citations |
|---|---|---|
| Journal articles | 397 (39.11%) | 11646 (68.62%) |
| Conference papers | 603 (59.41%) | 5267 (31.04%) |
| Book chapters | 14 (1.38%) | 48 (0.28%) |
| Books | 1 (0.10%) | 10 (0.06%) |
| Total | 1015 | 16971 |

Table 6: Distribution of the papers published in each decade.

| Decade | Conference articles | Journal articles | Book and book chapters | Total |
|---|---|---|---|---|
| 2011–2020 | 383 | 236 | 10 | 629 (61.97%) |
| 2001–2010 | 196 | 106 | 4 | 306 (30.14%) |
| 1991–2000 | 17 | 39 | 1 | 57 (05.61%) |
| 1981–1990 | 7 | 14 | 0 | 21 (02.06%) |
| 1971–1980 | 2 | 0 | 0 | 2 (0.19%) |

results show that fuzzy logic, analogy-based approach, ANN, regression, and optimization techniques are the most used techniques for SDECE. We did not validate the most-used datasets and accuracy measures by reading the title of the research papers because usually it is not mentioned in the title of the research paper.

Thus, the careful examination of the results obtained manually by reading the title of research papers and the outcome of the five selected SLRs shows that these results are almost similar with the results obtained using the proposed text-mining approach. Therefore, we strongly believe that the proposed automated text-mining framework is very useful to investigate research trends in an identified research area and makes our job easy compared to amount of time and efforts required to do so by systematic literature review method.

## 6. Discussions

We have used the following search strings to search literature from the Scopus database: "software effort estimation" OR "software cost estimation". The search string was designed by considering the objectives and research questions of the study. As papers are selected only from the

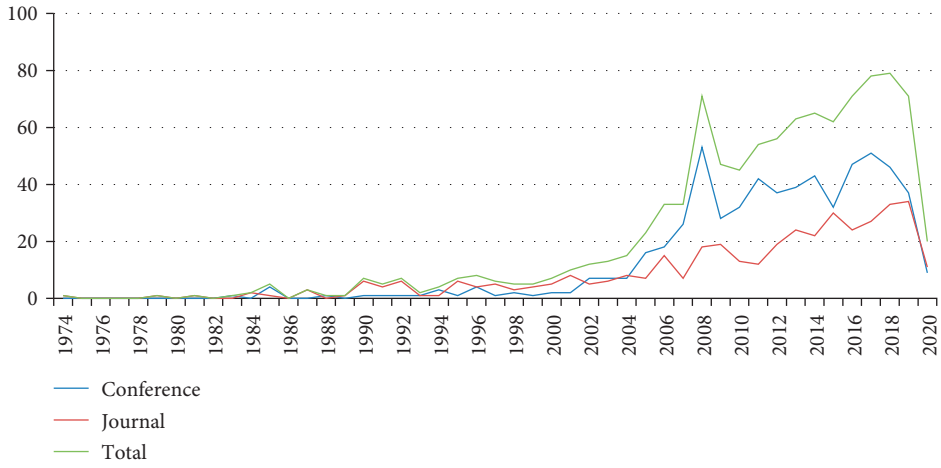



Figure 3: No. of papers per year.

Table 7: Distribution of papers with zero and with one or more than one citation.

| | Journal articles | Conference articles | Book chapters | Books | Total |
|---|---|---|---|---|---|
| Papers with one or more than one citations | 318 | 463 | 6 | 1 | 788 |
| Papers with zero citations | 79 | 140 | 8 | 0 | 227 |
| Total | 397 | 603 | 14 | 1 | 1015 |

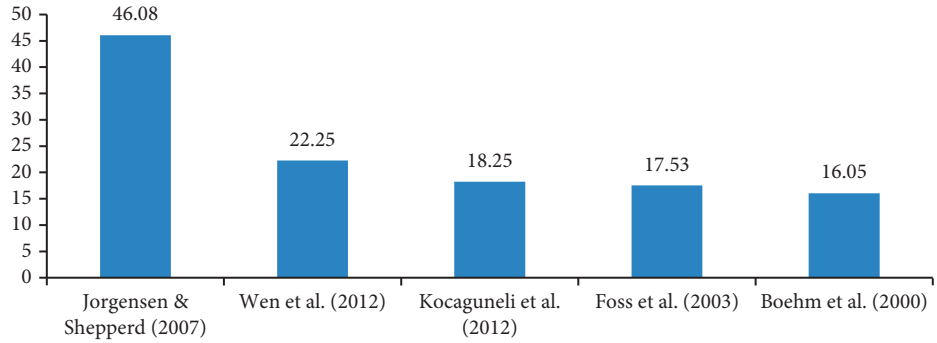



Figure 4: Highly cited papers.

Table 8: Contribution of the authors by the number of papers and citations.

| Authors by number of papers | | Authors by no. of citations | |
|---|---|---|---|
| Author name | No. of papers | Author name | No. of citations |
| Idri | 35 | Jorgensen | 1378 |
| Angelis | 32 | Boehm | 1227 |
| Keung | 30 | Shepperd | 875 |
| Mitts | 23 | Kitchenham | 831 |
| Boehm | 20 | Angelis | 808 |
| Azzeh | 20 | Menzies | 671 |
| Abran | 20 | Keung | 658 |
| Lokan | 18 | Kocaguneli | 553 |
| Jorgensen | 17 | Kemerer | 523 |
| Nassif | 17 | Stamelos | 472 |


Figure 5: Contribution of authors by the number of papers.

Scopus database, there is a possibility that we may have missed some relevant papers which could be a potential threat to this study. However, as Scopus is the largest abstract and citation database for peer-reviewed literature, it can offer the widest coverage of literature that one can achieve using a single search engine and it also mitigates the exclusion of relevant important papers [86, 87]. Further, we did not apply any exclusion criteria on the searched results except that the paper should be written in English language, and the focus of the study should be on SDECE. Therefore, we believe that there was no bias in the paper selection process. When we checked the final list of selected papers

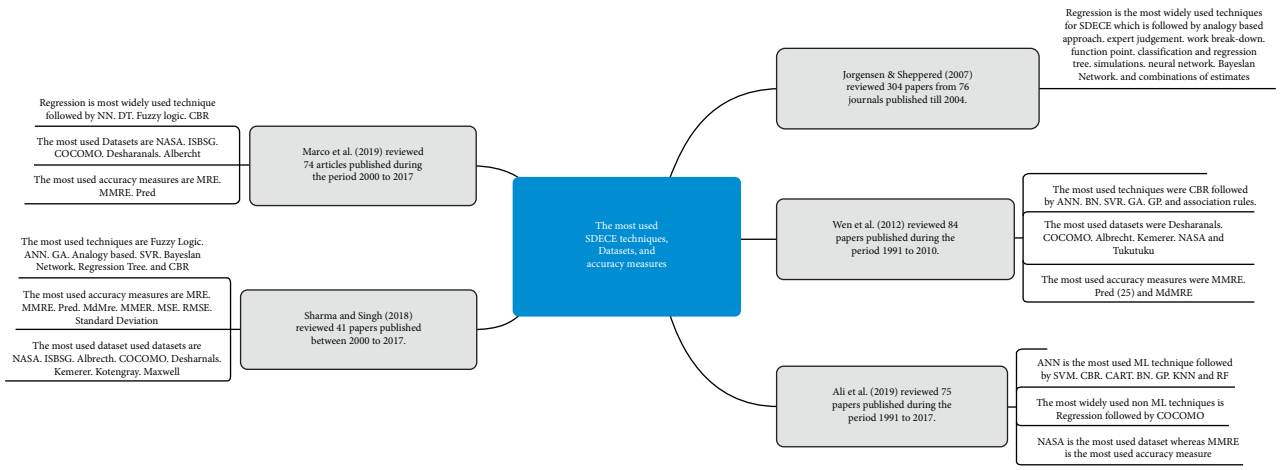



Figure 6: Outcomes of the past five SLRs.

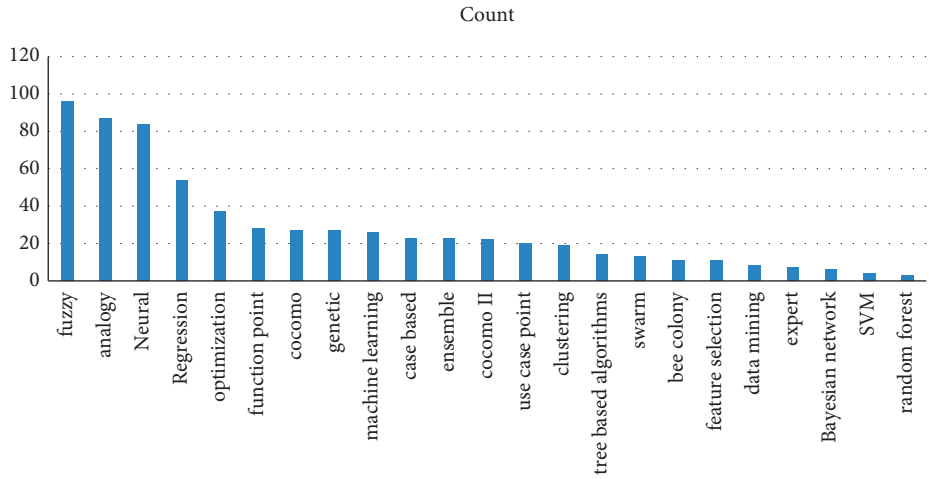



Figure 7: Most used SDECE techniques identified by reading the title of the papers.

and found that all the selected papers were relevant to meet the objectives of the study. We strongly believe that our analysis results would not be much different had we included more papers from the Scopus or other indexing databases because the number of papers selected in the study is 1015, which is a very good number to achieve research objectives. Another threat to the study with respect to using text mining for investigating the most popular technique in each decade is that if scholars use the name of the technique in a slightly different way than the usual one, then that technique would be treated as a different one. However, in most manuscripts, the techniques are named/referred in the same manner barring a few cases. Therefore, that will not affect much in capturing the overall research trend.

## 7. Conclusion

In this research article, a generic automated text mining framework is proposed to investigate the research trends by analyzing the title, keywords, and abstract of research papers published in an identified area. The proposed framework is used to investigate research trends by analyzing select 1015 research papers published on SDECE in the last five decades. It is found that fuzzy logic, artificial neural networks (ANN), regression, analogy, and COCOMO are the most popular techniques followed by use case point, function point analysis, and machine learning-techniques. The NASA and ISBSG are the most used datasets while MMRE, MRE, and PRED are the most used accuracy measures. It is observed that there is a lack of research on using deep learning techniques for software effort and cost estimation. Therefore, we recommend research scholars to explore deep learning techniques for software development effort and cost estimation. The analysis is also carried out to investigate the most used techniques, datasets, and accuracy measures in each decade to understand how SDECE techniques have evolved in the last five decades.

The results of the proposed framework are validated by comparing it with the outcome of previously published review work, and we have found that the results are consistent. Therefore, the proposed text mining framework is beneficial for futuristic study and can reduce the efforts required to investigate research trends on the topic of an identified research area. To uncover research trends, we have

analyzed the titles, keywords, and abstracts of the research papers separately and found that there is no significant difference in the outcome except slight change in the rank of the most popular SDECE techniques. The detailed bibliometric analysis is also performed along with the metareview of the survey papers, which aids to determine the most relevant papers, venues, authors, and contributions of researchers in the field of the proposed research. A study is recommended to uncover the research patterns and trends by analyzing numerous research papers collected from different electronic databases as this study is limited to research papers collected only from the Scopus database.

## Data Availability

The data are available for the experimental study.

## Conflicts of Interest

The authors have nothing to declare as conflicts of interest with respect to this manuscript.

## References


[1] S. Di Martino, F. Ferrucci, C. Gravino, and F. Sarro, "Assessing the effectiveness of approximate functional sizing approaches for effort estimation," *Information and Software Technology*, vol. 123, Article ID 106308, 2020.

[2] A. B. Nassif, M. Azzeh, A. Idri, and A. Abran, "Software development effort estimation using regression fuzzy models," *Computational Intelligence and Neuroscience*, vol. 2019, Article ID 8367214, 17 pages, 2019.

[3] A. Ali and C. Gravino, "A systematic literature review of software effort prediction using machine learning methods," *Journal of Software: Evolution and Process*, vol. 31, no. 10, 2019.

[4] H. L. T. K. Nhung, H. T. Hoc, and V. V. Hai, "A review of use case-based development effort estimation methods in the system development context," in *Intelligent Systems Applications in Software Engineering*, pp. 484–499, Springer, Scam, Switcherland, 2019.

[5] K. Molokken and M. Jorgensen, "A review of software surveys on software effort estimation," in *Proceedings of the 2003 International Symposium on Empirical Software Engineering, ISESE*, Rome, Italy, September, 2003.

[6] E. Kocaguneli, T. Menzies, and J. W. Keung, "On the value of ensemble effort estimation," *IEEE Transactions on Software Engineering*, vol. 38, no. 6, pp. 1403–1416, 2011.

[7] K. Dejaeger, W. Verbeke, D. Martens, and B. Baesens, "Data mining techniques for software effort estimation: a comparative study," *IEEE Transactions on Software Engineering*, vol. 38, no. 2, pp. 375–397, 2011.

[8] Y. Shan, R. I. McKay, C. J. Lokan, and D. L. Essam, "Software project effort estimation using genetic programming," *IEEE 2002 International Conference on Communications, Circuits and Systems and West Sino Expositions*, vol. 2, pp. 1108–1112, 2002.

[9] B. Alsaadi and K. Saeedi, "Data-driven effort estimation techniques of agile user stories: a systematic literature review" Artificial Intelligence Review," vol. 1-32, 2022.

[10] M. Jorgensen and M. Shepperd, "A systematic review of software development cost estimation studies," *IEEE Transactions on Software Engineering*, vol. 33, no. 1, pp. 33–53, 2006.

[11] C. F. Kemerer, "An empirical validation of software cost estimation models," *Communications of the ACM*, vol. 30, no. 5, pp. 416–429, 1987.

[12] B. Boehm, B. Clark, E. Horowitz, C. Westland, R. Madachy, and R. Selby, "Cost models for future software life cycle processes: cocomo 2.0," *Annals of Software Engineering*, vol. 1, no. 1, pp. 57–94, 1995.

[13] N. Mittas and L. Angelis, "Ranking and clustering software cost estimation models through a multiple comparisons algorithm," *IEEE Transactions on Software Engineering*, vol. 39, no. 4, pp. 537–551, 2012.

[14] C. Pareta, N. S. Yaadav, A. Kumar, and A. K. Sharma, "Predicting the accuracy of machine learning algorithms for software cost estimation," in *Advances in Intelligent Systems and Computing*, pp. 605–615, Springer, Singapore, 2019.

[15] A. A. Fadhil, R. G. H. Alsarraj, and A. M. Altaie, "Software cost estimation based on dolphin algorithm," *IEEE Access*, vol. 8, Article ID 75279, 2020.

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

[17] B. Boehm, C. Abts, and S. Chulani, "Software development cost estimation approaches—a survey," *Annals of Software Engineering*, vol. 10, no. 1-4, pp. 177–205, 2000.

[18] G. R. Finnie, G. E. Wittig, and J.-M. Desharnais, "A comparison of software effort estimation techniques: using function points with neural networks, case-based reasoning and regression models," *Journal of Systems and Software*, vol. 39, no. 3, pp. 281–289, 1997.

[19] R. Betteridge, "Successful experience of using function points to estimate project costs early in the life-cycle," *Information and Software Technology*, vol. 34, no. 10, pp. 655–658, 1992.

[20] A. J. Albrecht and J. E. Gaffney, "Software function, source lines of code, and development effort prediction: a software science validation," *IEEE Transactions on Software Engineering*, vol. SE-9, no. 6, pp. 639–648, 1983.

[21] V. Van Hai, H. Le Thi Kim Nhung, and H. T. Hoc, "A review of software effort estimation by using functional points analysis," in *Computational Statistics and Mathematical Modeling Methods in Intelligent Systems*, pp. 408–422, Springer, Cham, Switcherland, 2019.

[22] M. Ochodek, J. Nawrocki, and K. Kwarciak, "Simplifying effort estimation based on use case points," *Information and Software Technology*, vol. 53, no. 3, pp. 200–213, 2011.

[23] A. B. Nassif, L. F. Capretz, and D. Ho, "Estimating software effort based on use case point model using sugeno fuzzy inference system," in *Proceedings of the 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, pp. 393–398, Boca Raton, FL, USA, November, 2011.

[24] M. Azzeh and A. B. Nassif, "A hybrid model for estimating software project effort from Use Case Points," *Applied Soft Computing*, vol. 49, pp. 981–989, 2016.

[25] W. T. Lee, K. H. Hsu, J. Lee, and J. Y. Kuo, "Applying software effort estimation model based on work breakdown structure," in *Proceedings of the 2012 Sixth International Conference on Genetic and Evolutionary Computing*, pp. 192–195, IEEE, Kitakyushu, Japan, August, 2012.

[26] F. Walkerden and R. Jeffery, "An empirical study of analogy-based software effort estimation," *Empirical Software Engineering*, vol. 4, no. 2, pp. 135–158, 1999.

[27] M. Shepperd, C. Schofield, and B. Kitchenham, "Effort estimation using analogy," in *Proceedings of IEEE 18th International Conference on Software Engineering*, pp. 170–178, IEEE, Berlin, Germany, March, 1996.

[28] L. Angelis and I. Stamelos, *Empirical Software Engineering*, vol. 5, no. 1, pp. 35–68, 2000.

[29] E. Kocaguneli, T. Menzies, A. Bener, and J. W. Keung, "Exploiting the essential assumptions of analogy-based effort estimation," *IEEE Transactions on Software Engineering*, vol. 38, no. 2, pp. 425–438, 2011.

[30] J. Li, G. Ruhe, A. Al-Emran, and M. M. Richter, "A flexible method for software effort estimation by analogy," *Empirical Software Engineering*, vol. 12, no. 1, pp. 65–106, 2007.

[31] M. Azzeh, D. Neagu, and P. I. Cowling, "Analogy-based software effort estimation using Fuzzy numbers," *Journal of Systems and Software*, vol. 84, no. 2, pp. 270–284, 2011.

[32] I. Myrtveit and E. Stensrud, "A controlled experiment to assess the benefits of estimating with analogy and regression models," *IEEE Transactions on Software Engineering*, vol. 25, no. 4, pp. 510–525, 1999.

[33] Y. F. Li, M. Xie, and T. N. Goh, "A study of mutual information based feature selection for case based reasoning in software cost estimation," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5921–5931, 2009.

[34] D. Wu, J. Li, and C. Bao, "Case-based reasoning with optimized weight derived by particle swarm optimization for software effort estimation," *Soft Computing*, vol. 22, no. 16, pp. 5299–5310, 2018.

[35] P. Sentas, L. Angelis, I. Stamelos, and G. Bleris, "Software productivity and effort prediction with ordinal regression," *Information and Software Technology*, vol. 47, no. 1, pp. 17–29, 2005.

[36] A. B. Nassif, D. Ho, and L. F. Capretz, "Towards an early software estimation using log-linear regression and a multilayer perceptron model," *Journal of Systems and Software*, vol. 86, no. 1, pp. 144–160, 2013.

[37] O. Fedotova, L. Teixeira, and H. Alvelos, "Software effort estimation with multiple linear regression: review and practical application," *Journal of Information Science and Engineering*, vol. 29, no. 5, pp. 925–945, 2013.

[38] H. Park and S. Baek, "An empirical validation of a neural network model for software effort estimation," *Expert Systems with Applications*, vol. 35, no. 3, pp. 929–937, 2008.

[39] 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," *Journal of Systems and Software*, vol. 81, no. 3, pp. 356–367, 2008.

[40] N. Tadayon, "Neural network approach for software cost estimation," *IEEE International Conference on Information Technology: Coding and Computing (ITCC'05)-Volume II*, vol. 2, pp. 815–818, 2005.

[41] V. S. Dave and K. Dutta, "Neural network based models for software effort estimation: a review," *Artificial Intelligence Review*, vol. 42, no. 2, pp. 295–307, 2014.

[42] M. A. Ahmed, M. Omolade Saliu, and J. AlGhamdi, "Adaptive fuzzy logic-based framework for software development effort prediction," *Information and Software Technology*, vol. 47, no. 1, pp. 31–48, 2005.

[43] C. S. Reddy and K. V. S. N. Raju, "An improved fuzzy approach for COCOMO's effort estimation using Gaussian membership function," *Journal of Software*, vol. 4, no. 5, pp. 452–459, 2009.

[44] C. López-Martín, C. Yáñez-Márquez, and A. Gutiérrez-Tornés, "Predictive accuracy comparison of fuzzy models for software development effort of small programs," *Journal of Systems and Software*, vol. 81, no. 6, pp. 949–960, 2008.

[45] X. Huang, D. Ho, J. Ren, and L. F. Capretz, "Improving the COCOMO model using a neuro-fuzzy approach," *Applied Soft Computing*, vol. 7, no. 1, pp. 29–40, 2007.

[46] K. Srinivasan and D. Fisher, "Machine learning approaches to estimating software development effort," *IEEE Transactions on Software Engineering*, vol. 21, no. 2, pp. 126–137, 1995.

[47] B. Baskeles, B. Turhan, and A. Bener, "Software Effort Estimation Using Machine Learning Methods," in *Proceedings of the 22nd International Symposium on Computer and Information Sciences*, pp. 1–6, IEEE, Ankara, Turkey, November, 2007.

[48] F. A. Amazal and A. Idri, "Estimating software development effort using fuzzy clustering-based analogy," *Journal of Software: Evolution and Process*, vol. 33, no. 4, 2021.

[49] R. R. Sinha and R. K. Gora, "Software effort estimation using machine learning techniques," in *Proceedings of the Advances in Information Communication Technology and Computing*, pp. 65–79, Noida, India, January, 2021.

[50] A. L. I. Oliveira, P. L. Braga, R. M. F. Lima, and M. L. Cornélio, "GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation," *Information and Software Technology*, vol. 52, no. 11, pp. 1155–1166, 2010.

[51] S.-J. Huang and N.-H. Chiu, "Optimization of analogy weights by genetic algorithm for software effort estimation," *Information and Software Technology*, vol. 48, no. 11, pp. 1034–1045, 2006.

[52] C. V. M. K. Hari and P. V. G. D. Reddy, "A fine parameter tuning for COCOMO 81 software effort estimation using particle swarm optimization," *Journal of Software Engineering*, vol. 5, no. 1, pp. 38–48, 2011.

[53] S. Chalotra, S. K. Sehra, Y. S. Brar, and N. Kaur, "Tuning of cocomo model parameters by using bee colony optimization," *Indian Journal of Science and Technology*, vol. 8, no. 14, 2015.

[54] D. K. K. Reddy and H. S Behera, "Software effort estimation using particle swarm optimization: advances and challenges," in *Computational Intelligence in Pattern Recognition"*, pp. 243–258, Springer, Singapore, 2020.

[55] V. Garousi and M. V. Mäntylä, "Citations, research topics and active countries in software engineering: a bibliometrics study," *Computer Science Review*, vol. 19, pp. 56–77, 2016.

[56] S. K. Sehra, Y. S. Brar, N. Kaur, and S. S. Sehra, "Research patterns and trends in software effort estimation," *Information and Software Technology*, vol. 91, pp. 1–21, 2017.

[57] M. Fernández-Diego and F. González-Ladrón-De-Guevara, "Potential and limitations of the ISBSG dataset in enhancing software engineering research: a mapping review," *Information and Software Technology*, vol. 56, no. 6, pp. 527–544, 2014.

[58] S. Rajper and Z. A. Shaikh, "Software development cost estimation: a survey," *Indian Journal of Science and Technology*, vol. 9, no. 31, 2016.

[59] S. Bilgaiyan, S. Sagnika, S. Mishra, and M. Das, "A systematic review on software cost estimation in agile software," *Development" Journal of Engineering Science & Technology Review*, vol. 10, no. 4, 2017.

[60] V. Venkataiah, M. Ramakanta, and M. Nagaratna, "Review on intelligent and soft computing techniques to predict software cost estimation," *IJAER*, vol. 12, no. 22, Article ID 12665, 2017.

[61] H. Hakimi, M. Kamalrudin, A. Pradana, and S. Sidek, "A review of using software cost estimation tools in software development process," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 119, 2018.

[62] R. Marco, N. Suryana, and S. S. S. Ahmad, "A systematic literature review on methods for software effort estimation," *Journal of Theoretical and Applied Information Technology*, vol. 97, no. 2, 2019.

[63] M. A. Saleem, T. Alyas, R. Asfandayar Ahmad et al., "Systematic literature review of identifying issues in software cost estimation techniques," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 8, 2019.

[64] J. Hihn and H. Houabib-agahi, "Cost estimation of software intensive projects: a survey of current practices," in *Proceedings of the 13th International Conference on Software Engineering*, pp. 276–287, Austin, TX, USA, May, 1991.

[65] D. Yang, Q. Wang, M. Li, Y. Yang, K. Ye, and J. Du, "A survey on software cost estimation in the Chinese software industry," in *Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, pp. 253–262, Kaiserslautern, Germany, January, 2008.

[66] J. Keung, "Software Development Cost Estimation Using Analogy: A Review," in *Proceedings of the Australian Software Engineering Conference*, pp. 327–336, Gold Coast, QLD, Australia, April, 2009.

[67] P. Rodríguez-Soria, J. J. Cuadrado-Gallego, J. A. G. de Mesa, and B. Martín-Herrera, "A Review of Parametric Effort Estimation Models for the Software Project Planning Process," in *Proceedings of the 22nd International Conference on Software Engineering & Knowledge Engineering SEKE*, pp. 135–140, San Francisco, CA, USA, January, 2010.

[68] A. Trendowicz, J. Münch, and R. Jeffery, "State of the practice in software effort estimation: a survey and literature review," in *Proceedings of the IFIP Central and East European Conference on Software Engineering Techniques*, pp. 232–245, Springer, Berlin, Heidelberg, Germany, September, 2008.

[69] T. Wijayasiriwardhane, R. Lai, and K. C. Kang, "Effort estimation of component-based software development - a survey," *IET Software*, vol. 5, no. 2, pp. 216–228, 2011.

[70] P. Faria and E. Miranda, "Expert Judgment in Software Estimation during the Bid Phase of a Project--An Exploratory Survey," in *Proceedings of the 2012 Joint Conference of the 22nd International Workshop on Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement*, pp. 126–131, IEEE, Nagoya, Japan, Octobar, 2012.

[71] H. Hamza, A. Kamel, and K. Shams, "Software effort estimation using artificial neural networks: a survey of the current practices," in *Proceedings of the 2013 10th International Conference on Information Technology: New Generations*, pp. 731–733, IEEE, Las Vegas, NV, USA, April, 2013.

[72] B. Sigweni and M. Shepperd, "Feature weighting techniques for CBR in software effort estimation studies: a review and empirical evaluation," in *Proceedings of the 10th International Conference on Predictive Models in Software Engineering*, pp. 32–41, Torino, Italy, 2014, September.

[73] H. Rastogi, S. Dhankhar, and M. Kakkar, "A survey on software effort estimation techniques," in *Proceedings of the 2014 5th International Conference-Confluence the Next Generation Information Technology Summit*, pp. 826–830, IEEE, Noida, India, September, 2014.

[74] F. González-Ladrón-de-Guevara and M. Fernández-Diego, "ISBSG variables most frequently used for software effort estimation: a mapping review," in *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pp. 1–4, New York, USA, September, 2014.

[75] M. El Bajta, A. Idri, J. L. Fernández-Alemán, J. N. Ros, and A. Toval, "Software cost estimation for global software development a systematic map and review study," in *Proceedings of the 2015 International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE)*, pp. 197–206, Barcelona, Spain, April, 2015.

[76] M. Saroha and S. Sahu, "Tools & methods for software effort estimation using use case points model—a review," in *Proceedings of the International Conference on Computing, Communication & Automation*, pp. 874–879, Barcelona, Spain, April, 2015.

[77] K. Usharani, V. V. Ananth, and D. Velmurugan, "A survey on software effort estimation," in *Proceedings of the 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pp. 505–509, Chennai, India, March, 2016.

[78] B. Chinthanet, P. Phannachitta, Y. Kamei et al., "A review and comparison of methods for determining the best analogies in analogy-based software effort estimation," in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pp. 1554–1557, Pisa, Italy, April, 2016.

[79] T. Arnuphaptrairong, "A literature survey on the accuracy of software effort estimation models," *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 2, pp. 16–18, 2016.

[80] A. Saeed, W. H. Butt, F. Kazmi, and M. Arif, "Survey of software development effort estimation techniques," in *Proceedings of the 2018 7th International Conference on Software and Computer Applications*, pp. 82–86, Kuantan, Malaysia, Febrauary, 2018.

[81] P. Sharma and J. Singh, "Systematic Literature Review on Software Effort Estimation Using Machine Learning Approaches," in *Proceedings of the 2017 International Conference on Next Generation Computing and Information Systems (ICNGCIS)*, pp. 43–47, IEEE, Jammu, India, December, 2017.

[82] S. Silva and C. Mario, "Use of non-functional requirements in software effort estimation: systematic review and experimental results," in *Proceedings of the 2017 5th International Conference in Software Engineering Research and Innovation (CONISOFT)*, pp. 1–9, IEEE, Merida, Mexico, Octobar, 2017.

[83] B. Sharma and R. Purohit, "Review of current software estimation techniques," in *Proceedings of the International Conference on Recent Developments in Science, Engineering and Technology*, pp. 380–399, Bangalore, India, April, 2017.

[84] F. Walkerden and R. Jeffery, "Software cost estimation: a review of models, process, and practice," *Advances in Computers*, vol. 44, pp. 59–125, 1997.

[85] B. Nie and S. Sun, "Using text mining techniques to identify research trends: a case study of design research," *Applied Sciences*, vol. 7, no. 4, p. 401, 2017.

[86] p. Mongeon and A. Paul-Hus, "The journal coverage of web of science and scopus: a comparative analysis," *Scientometrics*, vol. 106, no. 1, pp. 213–228, 2016.

[87] L. S. Adriaanse and C. Rensleigh, "Web of science, scopus and Google scholar," *The Electronic Library*, vol. 31, no. 6, pp. 727–744, 2013.