Journal of Theoretical and Applied Information Technology
15th September 2020. Vol.98. No 17
© 2005 – ongoing JATIT & LLS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A SYSTEMATIC MAPPING STUDY ON SOFTWARE EFFORT ESTIMATION

AHLAM ALHADDAD [1], IBRAHIm Al-BALTAH [2], ABEDALLAH ABUALKISHIK[3], MAJDI ABDELLATIEF [4], ABDULMONEM ALI AL KHARUSI[5]

[1] Department of Computer Science, Faculty of Computer Science and Information Technology, Sana'a University, Sana'a, Yemen
[2] Department of Information Technology, Faculty of Computer Science and Information Technology, Sana'a University, Sana'a, Yemen
[3] College of Computer and Information Technology, American University in the Emirates, Dubai, UAE
[4] Department of Health Informatics, Inaya Medical Colleges, KSA
[5] Oman Research and Education Network, Muscat, Oman

E-mails:[1] ahlamahmed12@gmail.com ,[2] albalta2020@gmail.com ,[3] azasoft1@gmail.com ,
[4] khwaja24@yahoo.com, [5] Abdulmonem@omren.com.

ABSTRACT

Context: software effort estimation has been considered as one of the key drivers in software development success. A comprehensive understanding of state-of-the-art of software effort estimation techniques is very important.
Objective: the aim of this study is to identify and characterize the existing software effort estimation techniques and to points insights of this research field.
Method: a systematic mapping study on 136 primary studies was conducted to answer six research questions.
Results: the study revealed that most of the existing work have used MMRE, MRE, and PRED for measuring the accuracy of effort estimation, where NASA93 and COCOM81 were the most used dataset. Furthermore, most of the reviewed studies attempted to use machine learning methods, whereas other studies proposed hybrid models. With respect to size metrics, most of the reviewed studies used line of code (KLOC/ LOC/SLOC).
Conclusion: new research should be carried out and oriented towards studying the relationship between the various factors that increase or decrease software effort such as, project type, team member's expertise, required software reliability, and software complexity, which can be very useful to enhance effort estimation techniques.

Keywords: *Software Effort Estimation, cost estimation, Systematic mapping study.*

1. INTRODUCTION

Managing software projects is one of the key factors in project success. Barry Boehm(1981) pointed out in [1], "Poor management can increase software costs more rapidly than any other factor". The major aspect of poor software development management, which is typically caused the software to fail, is inaccurate effort estimation [2]. Therefore, software engineers should accurately estimate the amount of time and effort needed to complete any software project before they start planning for developing it. The accurate estimation is certainly help managing the software development more effectively, which is in turns reduce the likelihood of project failure [3].The major part of software cost is primarily the costs of effort involved. As a result of that, most of software cost estimation techniques rely on effort cost which calculate cost estimation in terms of a person-month [4], or in terms of software size. With this diversity, a considerable number of software effort estimation (SEE) techniques have been proposed in the literature. These techniques were divided into four categories which are algorithmic models (constructive cost model (COCOMO) and functional point analysis (FPA)), non-algorithmic models (expert judgment, price to win, Analogy based estimation), machine learning techniques (neural networks, fuzzy logic)[5], and hybrid method[6]. We conduct a Systematic Mapping Study (SMS) to understand the state of the art of software effort estimation technique. We aim to identify and analyze the different estimation techniques in the context of software effort estimation. Literature reviews aim to provide a brief background to the study, highlighting relevant work. Moreover, review articles describe a batch of articles that they have selected. However, the reliability of review articles is low not because they are unreliable, but because the writer has chosen articles and they might be biased. On the other hand, if the systematic mapping study is done correctly, it will greatly reduce possible bias. So it is will be the most reliable way to rich the body of knowledge of effort estimation. On the other hand, SMS focuses on providing an overview of a domain, identifying research activities on a research topic. One of the main differences between a literature review and SMS is that literature reviews focus on the results and discusses the findings, while SMSs have a broader scope to provide an overview of the research landscape[7].So in this end, we decide to do a systematic mapping study rather than a literature review. In this paper, we present a mapping study covered the papers that were published recently in software effort estimation up to January 2019. We performed automated searches on four famous scientific databases (ScienceDirect, Springer, ACM Digital Library, and IEEExplore). Then, we analyzed and classified 136 studies that were published from 2010 to 2019 based on six criteria namely: publication source, publication year, accuracy metrics, datasets, size metrics, and effort estimation methods. This study will serve as a high-level catalog of research in software effort estimation for both researchers and practitioners. In our study, we used the most two cited guidelines for conducting an SMS given by (Barbara Kitchenham [8], and Petersen et al[9, 10].

This systematic mapping study is structured as follows: related work is introduced in Section two. The methodology adopted to conduct this mapping study is presented in section three. The research results are illustrated in section Four. Section Five presents discussion and implications. Section Six describes threats to validity related to this mapping study, while Section Seven concludes this study and highlights some future works.

2. RELATED WORK

We have found five systematic mapping and systematic review studies which have investigated various facets on software cost/effort estimation [11], [12], [13], [14], [15].
Software effort estimation is not a recent topic, it has attracted researchers for more than a decads. As a result, the letirature is very rich of contributions that amied at enhancing the accuracy of SEE. One of the main reasons behinde that is, SEE was and still the predominant factor that has a direct impact on successul software project management. However, some existing studies focused on conducting a systimatic mapping study related to the topic of this study, which included studies publiched in different period of time.

A. Idri, M. Hosni, and A. Abran in [11] identified 24 studies that were published in the 2000–2016 periods. Their study aimed to analyze ensemble effort estimation (EEE) technique from six perspectives: individual models used to create ensembles, the accuracy of the ensemble estimate, rules for combining individual estimates, comparison of the precision of EEE methods with individual models, Comparison of precision between EEE techniques and methodology were used for creating ensemble methods. Authors concluded that EEE is more precise than individual models.

S. El Koutbi, A. Idri, and A. Abran in [12] focused on dealing with software effort estimation error. The authors selected 19 primary studies published between 1990 and 2015. They aimed to classify the identified primary studies according to research strategies, types of contributions, precise criteria, datasets, error approaches and methods for estimating effort. The results of their work indicated that in the past decades, the number of studies related to the estimate of software development effort error has increased. Despite the mapping study reported in [13], which was the most selected studies were conference papers, most of the studies published on journals. Moreover, authors noticed that accuracy metrics such as The Mean Magnitude of Relative Error, Median magnitude of relative error, and Percentage of the Prediction were widely used as performance metrics.

The study reported in 2015 [13] focused on summarizing studies on global software development cost estimation. A total of 16 primary

Journal of Theoretical and Applied Information Technology
15th September 2020. Vol.98. No 17
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

studies were selected and analyzed by the authors and categorized according to publishing source, year of publication, type of research, research strategy, and type of contribution, cost drivers, cost estimation activities and methods, and the performance of cost estimation for global software development. The reported results have shown that the number of studies in cost estimation for global software development has increased recently. The study, also, revealed that most of the studies (11 studies out of 16) were published on conferences. Another interesting finding is that, the dominant software cost estimate contribution types for global software development (GSD) studies are models.

Another study published in 2015 [14] conducted a mapping study on analogy-based software effort estimation (ASEE) techniques. Unlike [15], this study did not focus on all cost estimation techniques , rather it focused only on analogy-base method . The authors identified 65 studies from four scientific databases, which were published in the 1992–2012 periods. These selected studies were classified into four categories including the source of studies, research strategy, type of contribution techniques, and the steps of ASEE. Besides, this research examined ASEE techniques from different points of view such as estimation precision, comparative forecast precision, context estimation, the effect of methods utilized in combination with analogy-based software effort estimation techniques. This mapping study revealed that the use of fuzzy logic or genetic algorithms with an ASEE technique promises to produce more precise estimates.

M. Jorgensen and M. Shepperd In [15], presented a systematic mapping research on software cost estimation that published only in journals. Authors have selected 304 software cost estimation studies from 76 journals and classified them based on study topic, estimation method, research strategy, context of the study and the datasets used. They also presented some advices for future research on software cost estimation, such as expanding the domain of the search for relevant studies, use of the manual search, and raising awareness of the effect of dataset characteristics on the outcomes of estimation techniques.

It is clear that, most of the published systematic mapping studies share some similarities in the classification criteria such as estimation approach, dataset used, and accuracy metrics. Our work can be seen as an extended and updated to [15]. We adopted a more general search string than that presented in [14], which was more specific in analogy based estimation. Moreover, unlike [15], we included more recent publications, analyzed and classified 136 studies that were published recently which were not included in [11], [12], [13], [14], [15].

Table 1: Related work summary

References studies	Number primary studies	Database	Time period	Main focus	Classified based on					
					publication source	publication year	countries	accuracy criteria	datasets used	methods used
[11]	24	IEEE – ACM - Science Direct- Google Scholar	2000–2016	ensemble effort estimation	applicable	Applicable	not applicable	applicable	applicable	Applicable
[12]	19	IEEE – ACM - Science Direct- Google Scholar	1990 -2015	software effort estimation error	applicable	Applicable	not applicable	applicable	applicable	Applicable

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

studies References	Number primary studies	Database	Time period	Main focus	Classified based on					
[13]	16	IEEE – ACM - Science Direct- Google Scholar	2001-2014	global software development cost estimation	applicable	Applicable	not applicable	not applicable	not applicable	Applicable
[14]	65	IEEE – ACM - Science Direct- Google Scholar	1992–2012	analogy-base method	applicable	Applicable	not applicable	applicable	applicable	Applicable
[15]	304	76 journal s	1989 -April 2004	software cost estimation	not applicable	not applicable	not applicable	not applicable	applicable	applicable
This study	136	IEEE- Springe r- Science Direct.- ACM	2010 -Jan 2019	software effort estimation	applicable	applicable	applicable	applicable	applicable	applicable

3. METHOD

This mapping study was carried out according to the guidelines given by Petersen et al[9, 10]. Fig. 1 shows the process of systematic mapping research that used in this study with the actual number of obtained studies. Besides, the figure illustrates the outcomes from each step. As it is common in conducting mapping study, the first step is the formulation of research questions. This step is important for defining the systematic mapping scope. While, the main task of the second step is to search for studies that are potentially related to software effort estimation by applying the search strings on well-known databases. The result of this step is a large number of retrieved studies which are screened in third step. In this step only relevant studies are included by implementing the exclusion and inclusion criteria. Finally, a classification scheme is established using keywords, abstracts and full text that will be used to map the primary studies.

Journal of Theoretical and Applied Information Technology
15th September 2020. Vol.98. No 17
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ID | Research Questions | Main motivation

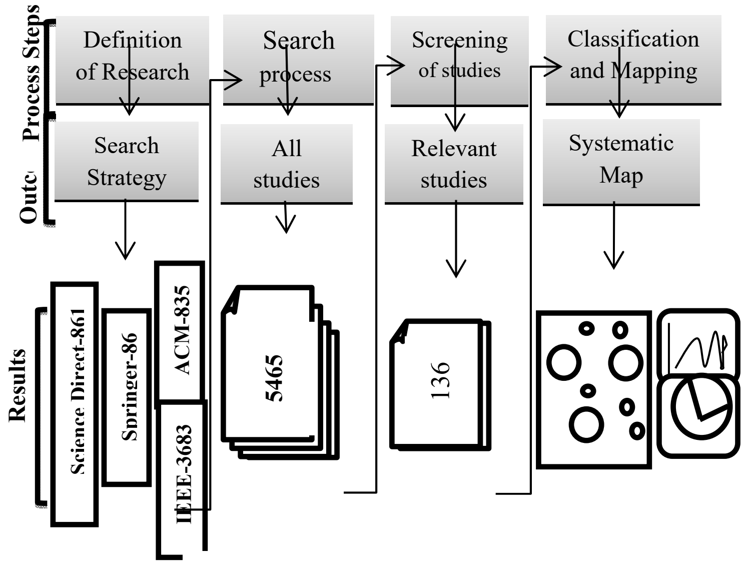

Fig. 1: Systematic mapping process

3.1 Research Questions

This study formulates six research questions which are aligned with the objectives and context of this study. These research questions are presented in Table2.

Table 2: Research questions

ID	Research Questions	Main motivation
RQ1	Which sources of publishing and venues are the primary destinations for studies on software effort estimation?	To provide researchers with a list of different publishing sources and venues for software effort estimation studies.
RQ2	How many effort estimation researches are published annually?	To investigate how the frequency of effort estimation research has changed over time.
RQ3	What metrics were used to evaluate the accuracy of effort estimation techniques? In the past, many metrics and number of methods in effort estimation have been proposed. Unfortunately, most of them involve experts' judgment which could affect the accuracy of the results.	To identify the different accuracy metrics that used to evaluated software effort estimation techniques and to identify the most widely used estimation performance criteria.

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

RQ4:	Which datasets were used to evaluate software effort estimation models? Although the set of metrics proposed for effort estimation are indeed useful for the characterization of software effort , unfortunately, most of the definition have lacked one or both of the following problems • Sound conceptual, theoretical bases • Statistically significant experimental validation	To identify the most used datasets.
RQ5	What are the methods, and techniques used for performing software effort estimation?	Many of the software metrics practitioners claim that the current state of software effort estimation methods is not very satisfying. Therefore, the goal of this question is to identify effort estimation methods in order to recognize the current state-of-the-art, identify gaps, and limitations in the recent research and to highlight the potential upcoming research.
RQ6	What size metrics have been used to estimate software effort?	To identify the most used size metrics.

3.2 Search Strategy

To gather the primary studies relevant to our research questions, a search strategy of two stages was performed. The focus of the first stage was mainly on the determination of the search terms in order to formulate the search strings. In order to conduct the second step, we first selected four online databases based on their availability and reputation of publishing a wide range of software engineering publications. The selected online databases are ScienceDirect, Springer, ACM Digital Library, and IEEExplore.

3.2 .1 Search terms

Our searchterms have been combined using the AN D operator to retrieve any studies containing all search terms. The search strings were formulated as follows:
(Software AND cost AND estimation) or (software AND effort AND estimation).

3.2.2. Literature resources

For the search process, we conducted an online search on four famous databases (IEEExplore, Springer, ScienceDirect, and ACM) based on the pre-constructed search string, All searches were restricted to studies published from January 2010 to January 2019. In IEEE, ACM, and ScienceDirect databases we include the studies that contain the search string in their titles, abstract, or keywords. In Springer, we include only the studies that contain the search string in their title; this is due to the restriction imposed by Springer system.

3.3 Screening The Studies For Exclusion And Inclusion

The screening process has three distinct stages during and after the search process to select the relevant studies.

First, we just read title and abstract of **5465** studies looking for information to exclude studies that do not relate to software effort estimation. As a result, **3979** studies were excluded from the study, and **1486** studies have left for the second stage. Column “**stage 1**“in Table4, displays the number of studies selected from each database.

In stage 2, we analyzed the studies, for further data beyond the title and abstract. The Criteria for exclusion and inclusion were used to exclude irrelevant studies, and to include only relevant

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

conclusion for each selected study in order to ensure that the focus of the study was within the scope of effort estimation .However, in some cases, we had to read the whole study to ensure their relevancy.

As a result of the third stage, **308** studies were excluded from the **444** studies which were resulted from second stage. As a result of carrying out these three stages, a total of 136 studies were chosen to perform this mapping study as presented in Table 4 (under stage 3 column).

studies for our mapping research. The Criteria for inclusion and exclusion were presented in Table 3. At this point, we also excluded studies that don't provide any contribution to software effort estimation. As a result of the second stage, **1042** more studies were excluded from the **1486** studies in the first stage leaving **only 444** studies for the third phase as presented in Table 4 (column stage 2).

In the last stage, duplicated studies were excluded. Moreover, in this stage we read the introduction and

Table3: Inclusion and exclusion criteria

Criteria of inclusion
- Published **studies** with the period of January 2010 to January 2019. - Published **studies** that focused on the estimation of software effort or software costs. - Peer reviewed journals, conferences and book chapter.
Criteria for exclusion
- **Studies** written in not English language. **- Studies that were not focused on** the estimation of software effort or software costs. -Review **studies**.

Table4: Number of studies before and after filtration phase

Databases	Results of search		Stage 1		Stage 2		Stage3
	query1	query2	query1	query2	query1	query2	
IEEE	2,511	1,172	566	414	153	102	80
ACM	451	384	101	112	24	26	9
ScienceDirect	507	354	94	117	41	45	25
Springer	42	44	40	42	31	22	22
The total	5465		1486		444		136

3.4 Keywording Of Abstracts

Keywording is a systematic way to identify and classify a set of existing studies and make sure that the scheme takes all existing studies into account[9]. We use Keywording techniques to reduce the effort required to create a classification scheme. Keywording techniques are made up of two steps .In step 1, we read the abstract of each study looking for any statements or keywords that represent the context, or the contribution of the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

study. As a result of that, our understanding of the nature and contribution of the studies is significantly improved. Also, it helps us to come up with a set of classifications. For some studies which their abstract were unclearly written, we had to read also the introduction and conclusion sections. In the second step, we built the categories for our mapping study based on the final set of keywords produced from **step1**.

3.4 Mapping And Data Extraction

To answer the aforementioned questions, we followed the guidelines provided by Kitchenham [8] .We extract the needed information from the primary studies based on the developed form for extracting data as presented in Table5. The data extraction form is organized into three columns, which are data items, their values, and our research questions. On the other hand, to enhance the presentation of our results, we used some visualization tools such as scatter plots, charts, and bars, etc.

Table5: Data extraction form

Data Items	**Values**	**Research Questions**
Study number	Integer	
Title of study	Study Name	
Author Name	Set of authors Names	
Publishing year	Year	RQ2
Country	Name of Country	RQ1
Venue	Publishing venue name	RQ1
Effort estimation method used	Hybrid, Machine Learning, algorithmic, Non-Algorithmic, Meta-heuristic, Optimization COCOMO	RQ5
Dataset	Name of the datasets	RQ4
Metric used for accuracy evolution	MMRE, MdMRE, PRED (x), etc.	RQ3
Metric used for size estimation	Size metrics : LOC, Function Points	RQ6

4. RESULTS

In general, the total number of the included studies in this mapping study is 136, which were retrieved and distributed differently in databases. The distribution of the selected studies is as follow: 80(58.8%) were selected from IEEExplore, 9(6.62%) were selected from ACM 25(18.4%) were selected from ScienceDirect and 22(16.%) were selected from Springer. Fig 2 illustrates the annual distribution of the publications included in this mapping research.

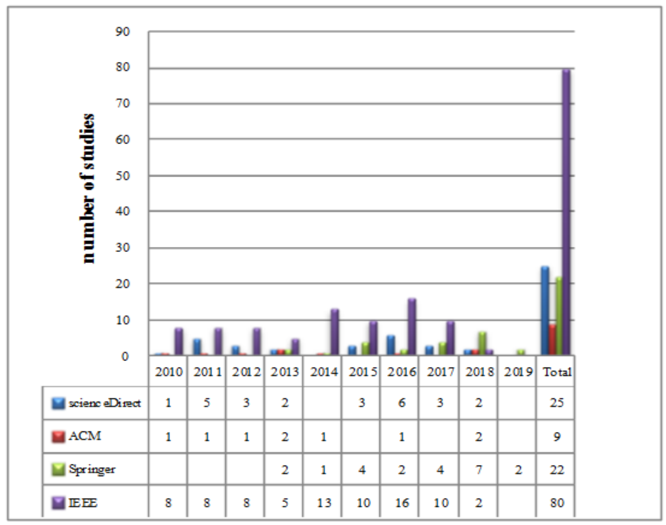

Fig2: Publication distribution per year

RQ1: Which sources of publishing and venues are the main destinations for studies on software effort estimation?

The selected studies were published in different venues as shown in table 6. A total of 136 selected studies were distributed over four venue types including journal, conferences, workshops, and symposium. However, journal and conference are the popular venue types, which account for 25.7% (35 studies out of 136) and 69.9% (95 studies out of 136) respectively.

Table 6: Number and proportion of the selected studies over publication venues

Venue types	Study ID	Number of studies	Percentage (%)
Journal	S1,S2,S5,S6,S10,S11,S12,S13,S14,S15, S18,S19,S20,S21,S29,S30,S31,S32,S42, S55,S75,S76, S96,S98,S99, S100,S103,S104,S106 ,S107,S108,S110,S111, S113 ,S115	35	25.7%
Conference	S3,S4,S7,S8,S9,S16, S17,S22,S23,S24,S25,S26,S27,S28,S33, S34,S35,S36,S37,S39,S41,S43,S44,S45, S46,S47,S48,S49,S50,S51,S52,S53,S54, S56,S57,S58,S61,S62,S63,S64,S65,S66, S67,S68,S69,S70,S71,S72,S73,S74,S77, S78,S79,S80,S81,S82,S83,S84,S85,S86, S87,S88,S89,S90,S91,S92,S93,S94,S95, S97,S101, S102,S105,S109,S112,S114,S116,S117, S118,S119,S120,S121,S122,S123,S126, S127,S128,S129,S130,S131,S132,S133, S134,S135 ,S136	95	69.9%
Workshop	S60	1	0.7%
Symposium	S38,S40 ,S59,S124,S125	5	3.7%

Fig 3 presents the proportion of the selected studies over countries. The leading country for publishing is India which account for 41% (56 studies out of 136) .

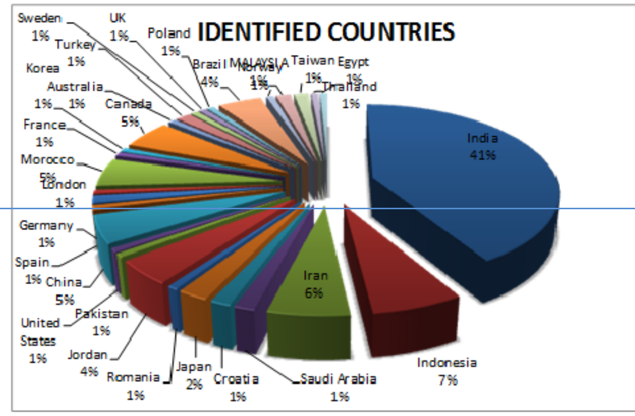

Fig3. Proportion of the selected studies over countries

RQ2: How many effort estimation researches are published annually?

Fig4 illustrates the distribution of the selected studies over years. Software effort estimation has received a noticeable attention from researchers and reached its peak at 2016. Same number of studies (17 study) were published in 2015 and 2017. From 2018 the number of published studies is gradually decreased. As it can be seen from Fig 4, there are a few studies published in 2019. Researchers continue their works in this area when we stop searching the literature.

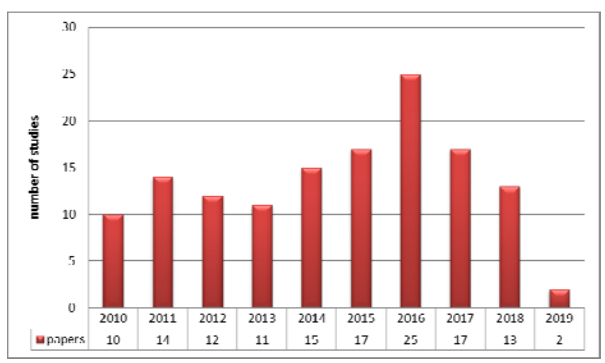

Fig4: The annual publication frequency

RQ3: What metrics were used to evaluate the accuracy of effort estimation techniques?

The results of this question are illustrated in Fig5. This figure shows the variety of accuracy

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

measurements that were used in the selected studies. As it can be seen from Fig 5, more than half of the selected studies used MMRE to calculate the mean relative error magnitude of the proposed work. MRE and PRED are considered as a common metrics which were used in 51 and 62 studies respectively. MRE is used to measure the absolute difference between actual and estimated efforts, while PRED is used to calculate the percentage of estimation that fall within X% of the real value. Unfortunately, the studies S23, S38, S45, S57, S59, S62 ,S83,S84,S90,and S113 did not use accuracy metrics to evaluate their proposed effort estimation methods. It is significant to notice that, the 136 selected studies used different measurements of accuracy. This makes it extremely difficult to compare the performance of different techniques. However, it is interesting to notice that MMRE ,PRED (X), and MRE are the most frequently used accuracy measure in our primary studies. Moreover, another interesting notice is that, some metrics where used only in one study such as MD, RMS, AF etc., which means there were proposed and used in the same study.

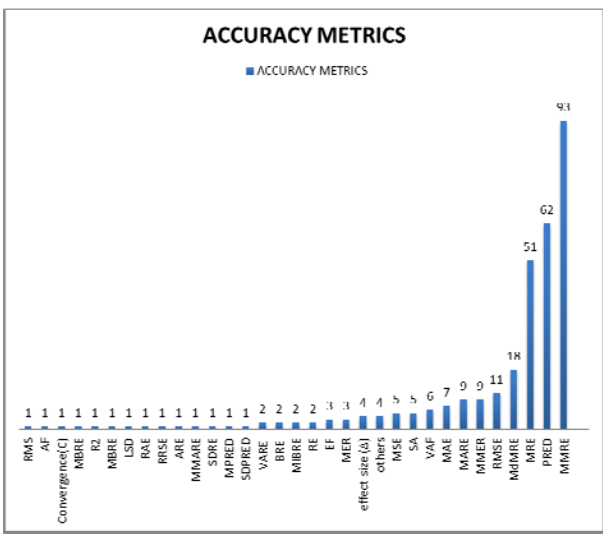

Fig5: Accuracy measurements that were used in the selected studies

RQ4: Which datasets were used to evaluate software effort estimation models?

Figure 6 presents the multiple datasets that were used in the selected studies to evaluate the efficiency of the propose work that focus on software **effort** estimation. As it can be seen from the figure 5, the most used dataset is NASA93 (25.7%). This dataset includes 93 flight or ground system software projects developed for NASA in seven different development centers in the 1970s and 1980s[16]. Moreover, nearly (22 %) used the COCOMO81 dataset, which consists of 63 software projects. We noticed that, 51 of the selected studies (37.5%) used more than one datasets. However, some studies including S30, S41, S45, S49, S57, S70,S80,S100,S113,S127,S128, and S132 did not state what dataset being used.

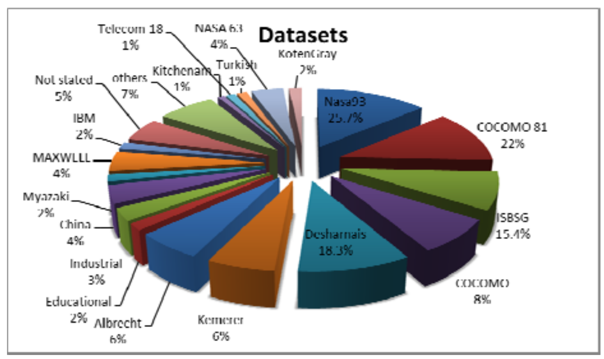

Fig6: The proportion of the found datasets

RQ5: What are the methods and techniques used for performing software effort estimation?

In order to answer Question 5, we extracted and classified data related to the methods that have been used to estimate software **effort**. The result of this process is represented in Table 7, which consists of six classes namely: hybrid approach (which is a combination of two or more techniques), machine learning approach, algorithmic approach, non-algorithmic approach, optimization COCOMO model, and Meta-heuristic approach.

As it can be seen from the table7, almost 48 studies out of 136 (35.3%) used machine learning. On the other hand, some researchers preferred to combine more than one technique to get more accurate effort estimation. This hybrid approach was used by 37 studies (27.2%), while a proportion of (26.5%) were attempted to optimize COCOMO model.

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 7: Effort estimation techniques

Estimation technique	Study ID	Number of studies	Percentage (%)
Hybrid model	S1, S11, S12, S16, S24, S25, S29, S30, S32, S33, S36, S40, S42, S47, S52, S53, S54,S55, S62, S63,S78 ,S81,S83,S86,S90,S91,S101, S103,S104,S110,S111,S112, S115, S121 ,S124,S127,S135	37	27.2%
Machine Learning	S2, S3, S6, S7, S13, S17, S20, S22, S28, S34, S35, S44, S51, S56, S59, S66, S67, S68, S70,S75,S76 ,S77,S85,S88,S89,S92,S93,S94,S97,S98,S99,S102,S105, S107,S108,S109,S114,S116, S117,S120,S123,S126,S128, S129,S130,S131,S132,S133	48	35.3%
Optimizing COCOMO	S4, S5, S8, S9, S10, S18, S19, S26, S31, S37, S38, S39, S41, S43, S46, S50, S58, S61, S69, S71, S72, S73, S74,S79, S84,S87,S95,S96,S100, S106,S118,S119,S122,S125, S134,S136	36	26.5%
Non-Algorithmic	S14, S23, S60,S82,S113	5	3.7%
Algorithmic method	S15, S45, S48, S57	4	2.9%
Meta-heuristic algorithm	S21, S27, S49, S64, S65,S80	6	4.4%

RQ6: What size metrics have been used to estimate effort?

Table8 shows that the size measures used in the 136 primary studies indicate that the lines of code (KLOC / LOC / SLOC) are the most frequently used size measures in **effort** estimation. In 7 of the studies, the use case was used as size measures, and a function point is used in 7 studies. No size measurements were shown in 53 studies because they focused only on cost drivers.

Journal of Theoretical and Applied Information Technology
15th September 2020. Vol.98. No 17
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 8: Size metrics

Size metric	Study ID	Number of studies	Percentage (%)
COSMIC method	S1	1	0.7%
Use Case Point	S2,S76,S93,S104,S116,S126, S127	7	5.1%
KLOC/ LOC/ SLOC	S3, S4, S5,S8,S9,S10, S16 ,S18,S19,S21, S22,S24,S25 ,S26,S28, S29, S31, S34, S37 ,S38,S39,S41,S43,S46,S49 ,S50,S52,S58,S59,S61,S64,S69,S71, S72,S73,S74,S78,S79,S83 ,S84,S87,S88,S91, S96,S98, S100,S102,S105,S108,S110, S118, S92, S119,S121, S122,S123,S124,S125,S129, S131, S132,S133, S134,S136	64	47.1 %
Adjusted Function Point	S6,S60	2	1.5%
function points	S14,S15,S27,S57,S80,S81,S128	7	5.1%
Not stated	S7, S11, S12, S13, S17, S20 , S23, S30,S32,S35,S36,S33, S40,S44,S45,S47,S48,S51,S53,S54,S55,S56,S62,S63,S65,S66,S67,S68,S70,S77,S82, S85,S86,S89,S90,S94,S95,S97,S99,S101,S103, S106,S107,S109,S111, S112,S113,S114,S115, S117 ,S120,S130,S135	53	39%
International Function Point	S42,S75	2	1.5%

5. DISCUSSION AND IMPLICATIONS

In this systematic mapping study, we reviewed 136 primary studies focused on the topic of software effort estimation, which were published between 2010 till Jan 2019. The results of our systematic mapping study have implications for researchers who are planning new studies for improving the accuracy of software effort estimation effectively. In this section, we summarize our findings and our recommendations for researchers as follows:

Accuracy metrics: according to the results of this study, MMRE, MRE, and PRED (X) are the most used accuracy measures in our primary studies that confirm the results obtained by the previous mapping studies [11], [12], [14] . Moreover, the selected studies used various accuracy metrics to evaluate the proposed effort estimation methods. This variation makes accuracy comparison between different software effort estimation methods very difficult.

Although MMRE and PRED are still widely used for effort estimation accuracy measurement which is based on magnitude of relative error (MRE), basically, they do not measure the real accuracy but the distribution. Kitchenham et al. [17] reported that MMRE and PRED (25) are, respectively, spread measurements and kurtosis of factor Y where Y = estimate observation / actual observation. Accordingly, Y is known as a measure of accuracy, and MMRE and PRED (25) are a measure of the properties of the distribution of Y [17]. On the other hand, Idri, Ali et al.[18] Stated that PRED (x) is less biased towards underestimations than MMRE and usually selects the same best methods as the Standardized Accuracy (SA), thus PRED (x) can be considered a reliable measure of accuracy.

The Standardized Accuracy (SA) is a new accuracy measure proposed by Shepperd and MacDonell [19] , which based on the mean absolute residual (MAR) . SA measures accuracy as the MAR of prediction technique p_i relative to random guessing p_0 [19] to make sure that the obtained models are performing significantly better than random and constant models.

Therefore, we recommend using the Standardized Accuracy (SA) measures that have been extensively used recently. For example, Abualkishik and Lavazza [20] highlighted that: relative absolute residuals (MMRE, MAR) can produce somewhat different evaluation and thus different interpretation for accuracy results. Therefore, they used absolute residuals in their evaluation to alleviate this problem.

Datasets: The 136 selected studies used various historical datasets to evaluate the performance of estimation methods.
The NASA93 dataset, for instance, was published in 2006 and is publicly available for research purpose. NASA93 was collected from various centers in the period between 1971- 1987. In other words, it is too antiquated, and it might not be able to cope with the modern software development practices. In spite of that, NASA93 dataset has been used in (25.7%) studies as a primary dataset for evaluating the accuracy of the proposed effort estimation methods. Moreover, we found that almost none of the studies discussed deeply the properties of the dataset or the relationship between dataset's attributes and how they affect the software effort estimation. Rather, they only address the availability of the dataset.

On the other hand, the ISBSG dataset is a large public database that contains data from different countries, organizations, application types, and development types[21]. The ISBSG dataset is large, verified, recent, and representative of current technologies in compared to other public datasets available to researchers, such as NASA93 and COCOMO 81 datasets. However, the ISBSG dataset has been found only in (15.4%) of our chosen studies as a primary dataset for evaluating the accuracy of the proposed effort estimation methods.

Therefore, we recommend that researchers pay more attention to understand the relationship between (effort multipliers) in the datasets and how representative they are for modern software rather than focusing on just the availability of the datasets. Moreover, we encourage researchers to use the most maintained and annually renewed commercial dataset such as the ISBSG dataset rather than using the old dataset.

Estimation methods: our systematic mapping has identified six different approaches to estimate software effort. Almost 48 studies out of 136 (35.3%) used machine learning to get a more accurate estimation . Also, we found that 37 studies (27.2%) used hybrid methods, which are the mixture of two or more approaches such as (meta-heuristic + machine learning), following by 36 studies (26.5%) that present various techniques/methods for optimizing the COCOMO model. It is difficult to determine which method is the best for estimating the software development effort since they are dissimilar in types and using un-comparative techniques. Moreover, the validation of the six methods has been using different accuracy metrics and datasets.
According to our sample findings, there is no standard technique to estimate software effort. However, it seems that recently, machine learning method was the most commonly used model by researchers.

Consequently, to provide a clear comparison between various estimation methods we suggest that researcher must study the different estimation methods deeply and apply them in real life to decide which method is more accurate and suitable to accurately estimate the effort of software. Not only this, we recommend that each organization that wish to use the effort estimation models to build their own model based on their own in-house data and calibrate the models from time to time to adapt to any estimation discrepancies. They are also encouraged to test several estimation methods to select the best-fits method that best-fit their data and their needs.
Size measures: according to the results of this study, lines of code (KLOC / LOC / SLOC) were the most frequently used size measures in effort estimation studies. Indeed, LOC has been criticized a lot as inadequate measure due to inherent reasons such as subjectivity in coding style, programming language maturity, availability at later stages in the software development life cycle, and many other reasons [22]. To overcome the limitations of SLOC, Albrecht[23]proposed a new size measurement method called Function Point Analysis (FPA). Most of the recent studies have shifted to use Function Point since it covers the limitation of LOC. Thus, we encourage researchers/ practitioners to use function points instead of using LOC to avoid the limitations of SLOC.

6. THREATS TO VALIDITY

The validity of our mapping study on software **effort** estimation has some threats. .In this section, we discuss the main threats and highlight the strategies that have been taken to minimize their eff ects:

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Above all, exclusion of relevant studies was the main threat we faced in our mapping study. To avoid this issue, an extensive search is being done into four well-known and relevant databases (ScienceDirect, Springer, IEEExplore, and ACM digital library) by using two search strings listed in Table 3 to retrieve as many relevant studies as possible. Moreover, we only chose studies that match our criteria for inclusion and exclusion. Nevertheless, we perceived that the search terms we used will not return all relevant studies. So, it is impossible to guarantee that we include all the relevant studies .

Internal validity is another threat which could lead to irrelevant or inaccurate conclusion. The primary cause of the threat of the internal validity of our mapping study is inaccuracy of data extraction, and improper interpretation of the information in the selected studies. Extraction data from primary studies is a manual process which might lead to inaccuracies. To reduce this threat, we have used a predefined data extraction form which is presented in Table 4 to decide which information must be extracted from the selected studies to answer each mapping question. This will help us to classify the extracted data, and draw conclusions easily. However, data extraction bias may still occur despite the use of a data extraction form.
Moreover , the software effort estimation techniques that we present in table 7 were based on the selected studies. Therefore, it might not cover all methods if there some techniques presented in papers that not satisfied our inclusion criteria.

7. CONCLUSION AND FUTURE WORK

A systematic mapping study in software **effort** estimation context was presented in this study. This study will serve researchers through providing a library of software **effort** estimation studies. We summarized and classified primary studies according into six criteria namely: publication source, publication year, accuracy metrics, datasets, size metrics and effort estimation methods. We performed automated searches in four famous databases (IEEE, Springer, Direct Science and ACM), to identify the primary studies for the predefined research questions. Our systematic mapping study covered studies published between 2010 and Jan 2019. .Overall, we identified 136 primary studies. From the mapping study that has been carried out, it is possible to conclude the major findings as follows:

One of the more significant findings emerge from this study is that the interest in using machine learning methods has increased in recent years. The second major finding was that the most targeted publications for software **effort** estimation are conferences. Moreover, the MMRE, MRE, and PRED(X) are the most used accuracy measure in most primary studies despite all the criticisms for these accuracy measures. Therefore, we recommend using the Standardized Accuracy (SA) or absolute residues instead.

To train and test the proposed **effort** estimation methods, NASA93 dataset is the most used dataset by researchers, followed by the COCOMO81 dataset. In addition, it is important to notice that, some studies used more than one dataset to evaluate the accuracy of their optimization techniques.
It is obvious that, new research should be carried out and oriented towards studying the relationship between the various factors that increase or decrease software effort such as, project type, team member's expertise, required software reliability, and software complexity, which can be very useful to enhance effort estimation techniques.
In our future research, we plan to conduct a detailed SLR based on the findings of this systematic mapping.

REFERENCES

[1] B. W. Boehm, Software engineering economics vol. 197: Prentice-hall Englewood Cliffs (NJ), 1981.
[2] A. Trendowicz, "Why Software Effort Estimation?," in Software Cost Estimation, Benchmarking, and Risk Assessment: The Software Decision-Makers' Guide to Predictable Software Development, ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 3-7.
[3] N. Dewan and S. Sehra, "Ant Colony Optimization Based Software Effort Estimation," International Journal of Computer Science And Technology, vol. 5 ,pp. 53-56, 2014.
[4] K. Choudhary, "Parametric Estimation of Software Systems," International Journal of Soft Computing and Engineering (IJSCE) ISSN, pp. 2231-2307, 2011.
[5] S. Gupta, S. Tiwari, H. Singh, A. Shukla, and H. Raghuvanshi, "A Comparison between Various Software Cost Estimation

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Models," Int. J. Emerg. Trends Sci. Technol., vol. 3, pp. 4771-4776, 2016.

[6] M. T. Bingamawa and M. Kamalrudin, "A Review of Software Cost Estimation: Tools, Methods, and Techniques," ed, 2016.

[7] I. D. Cooper, "What is a “mapping study?”," Journal of the Medical Library Association: JMLA, vol. 104, p. 76, 2016.

[8] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," 2007.

[9] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic Mapping Studies in Software Engineering," in EASE, 2008, pp. 68-77.

[10] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," Information and Software Technology, vol. 64, pp. 1-18, 2015.

[11] A. Idri, M. Hosni, and A. Abran, "Systematic literature review of ensemble effort estimation," Journal of Systems and Software, vol. 118, pp. 151-175, 2016.

[12] S. El Koutbi, A. Idri, and A. Abran , "Systematic mapping study of dealing with error in software development effort estimation," in 2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2016, pp. 140-147.

[13] 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 Evaluation of Novel Approaches to Software Engineering (ENASE), 2015 International Conference on, 2015, pp. 197-206.

[14] A. Idri, F .azzahra Amazal, and A. Abran, "Analogy-based software development effort estimation: A systematic mapping and review," Information and Software Technology, vol. 58, pp. 206-230, 2015.

[15] M. Jorgensen and M. Shepperd, "A systematic review of software development cost estimation studies," IEEE Transactions on software engineering, vol. 33, 2007.

[16] F. Nayebi, A. Abran, and J.-M. Desharnais, "Automated selection of a software effort estimation model based on accuracy and uncertainty," Artificial Intelligence Research, vol. 4, p. 45, 2015.

[17] B. A. Kitchenham, L. M. Pickard, S. G. MacDonell, and M. J. Shepperd, "What accuracy statistics really measure," IEE Proceedings-Software, vol. 148, pp. 81-85, 2001.

[18] A. Idri, I. Abnane, and A. Abran, "Evaluating Pred (p) and standardized accuracy criteria in software development effort estimation," Journal of Software: Evolution and Process, vol. 30, p. e1925, 2018.

[19] M. Shepperd and S. MacDonell, "Evaluating prediction systems in software project estimation," Information and Software Technology, vol. 54, pp. 820-827, 2012.

[20] A. Z. Abualkishik and L. Lavazza, "IFPUG Function Points to COSMIC Function Points convertibility: A fine-grained statistical approach," Information and Software Technology, vol ,97 .pp. 179-191, 2018.

[21] F. González-Ladrón-de-Guevara, M. Fernández-Diego, and C. Lokan, "The usage of ISBSG data fields in software effort estimation: A systematic mapping study," Journal of Systems and Software, vol. 113, pp. 188-215, 2016.

[22] A .Z. Abualkishik, J.-M. Desharnais, A. Khelifi, A. A. A. Ghani, R. Atan, and M. H. Selamat, "An exploratory study on the accuracy of FPA to COSMIC measurement method conversion types," Information and Software Technology, vol. 54, pp. 1250-1264, 2012/11/01.2012 /

[23] A. J. Albrecht, "Measuring application development productivity," in Proc. Joint Share, Guide, and IBM Application Development Symposium, 1979, 1979.

[24] R. K. Sachan, A. Nigam, A. Singh, S. Singh, M. Choudhary, A. Tiwari, et al., "Optimizing basic COCOMO model using simplified genetic algorithm," Procedia Computer Science, vol. 89, pp. 492-498, 2016.

[25] A. Panda, S. M. Satapathy, and S. K. Rath, "Empirical validation of neural network models for agile software effort estimation based on story points," Procedia Computer Science, vol. 57, pp. 772-781, 2015.

[26] S. Laqrichi, F. Marmier, D. Gourc, and J. Nevoux, "Integrating uncertainty in software effort estimation using Bootstrap

based Neural Networks," IFAC-PapersOnLine, vol. 48, pp. 954-9.2015 ,59
[27] S. P. Singh, V. P. Singh, and A. K. Mehta, "Differential Evolution Using Homeostasis Adaption Based Mutation Operator and its Application for Software Cost Estimation," Journal of King Saud University-Computer and Information Sciences, 2018.
[28] A. Puspaningrum and R. Sarno, "A hybrid cuckoo optimization and harmony search algorithm for software cost estimation," Procedia Computer Science, vol. 124, pp. 461-469, 2017.
[29] P. R. Sree and R. SNSVSC, "Improving efficiency of fuzzy models for effort estimation by cascading & clustering techniques," Procedia Computer Science, vol. 85, pp. 278-285, 2016.
[30] 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.
[31] D. Čeke and B. Milašinović, "Early effort estimation in web application development," Journal of Systems and Software, vol. 103, pp. 219-237, 2015.
[32] B. Tanveer, A. M. Vollmer, and S. Braun, "A hybrid methodology for effort estimation in Agile development: an industrial evaluation," in Proceedings of the 2018 International Conference on Software and System Process, 2018, pp. 21-30.
[33] F. Sarro and A. Petrozziello, "Linear Programming as a Baseline for Software Effort Estimation," ACM Transactions on Software Engineering and Methodology (TOSEM), vol. 27, p. 12, 2018.
[34] S. Ezghari and A. Zahi, "Uncertainty management in software effort estimation using a consistent fuzzy analogy-based method," Applied Soft Computing, vol. 67 ,pp. 540-557, 2018.
[35] S. Dragicevic, S. Celar, and M. Turic, "Bayesian network model for task effort estimation in agile software development," Journal of Systems and Software, vol. 127, pp. 109-119, 2017.
[36] A. Idri, M. Hosni, and A. Abran, "Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles," Applied Soft Computing, vol. 49, pp. 990-1019, 2016.
[37] S. H. S. Moosavi and V. K. Bardsiri, "Satin bowerbird optimizer: A new optimization algorithm to optimize ANFIS for software development effort estimation," Engineering Applications of Artificial Intelligence, vol. 60, pp. 1-15, 2017.
[38] F. Zare, H. K. Zare, and M. S. Fallahnezhad, "Software effort estimation based on the optimal Bayesian belief network," Applied Soft Computing, vol. 49, pp. 968-980, 2016.
[39] P. Rijwani and S. Jain, "Enhanced software effort estimation using multi layered feed forward artificial neural network technique," Procedia Computer Science, vol. 89, pp. 307-312, 2016.
[40] S. Kumari and S. Pushkar, "A genetic algorithm approach for multi-criteria project selection for analogy-based software cost estimation," in Computational Intelligence in Data Mining-Volume 3, Smart Innovation, Systems and Technologies 33,, L. C. J. e. al., Ed ,.ed India: Springer, 2015, pp. 13-24.
[41] M. Lusky, C. Powilat, and S. Böhm, "Software cost estimation for user-centered mobile app development in large enterprises," in International Conference on Applied Human Factors and Ergonomics, 2017, pp. 51-62.
[42] V. Venkataiah, R. Mohanty, and M. Nagaratna, "Prediction of Software Cost Estimation Using Spiking Neural Networks," in Smart Intelligent Computing and Applications, ed: Springer, 2019, pp. 101-112.
[43] S. P. Singh and A. Kumar, "Multiobjective differential evolution using homeostasis based mutation for application in software cost estimation," Applied Intelligence, vol. 48, pp. 628-650, 2018.
[44] P. S. Bishnu and V. Bhattacherjee, "Software cost estimation based on modified K-Modes clustering Algorithm," Natural Computing, vol. 15, pp. 415-422, 2016.
[45] I. Kaur, G. S. Narula, R. Wason, V. Jain, and A. Baliyan, "Neuro fuzzy—COCOMO II model for software cost estimation," International Journal of Information Technology, vol. 10, pp. 181-187, 2018.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[46] A. Kaushik, S. Verma, H. J. Singh, and G. Chhabra, "Software cost optimization integrating fuzzy system and COA-Cuckoo optimization algorithm," International Journal of System Assurance Engineering and Management, vol. 8, pp. 1461-1471, 2017.

[47] S .Porru, A. Murgia, S. Demeyer, M. Marchesi, and R. Tonelli, "Estimating story points from issue reports," in Proceedings of the The 12th International Conference on Predictive Models and Data Analytics in Software Engineering, 2016, p. 2.

[48] 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, pp. 5299-5310, 2018.

[49] S. Kumari and S. Pushkar, "Cuckoo search based hybrid models for improving the accuracy of software effort estimation," Microsystem Technologies, vol. 24, pp. 4767-4774, 2018.

[50] J. F. Vijay, "Enrichment of accurate software effort estimation using fuzzy-based function point analysis in business data analytics," Neural Computing and Applications, pp. 1-7, 2018.

[51] S. K. Sehra, Y. S. Brar, N. Kaur, and S. S. Sehra, "Software effort estimation using FAHP and weighted kernel LSSVM machine," Soft Computing, pp. 1-20, 2018.

[52] S. Arora and N. Mishra, "Software Cost Estimation Using Artificial Neural Network," in Soft Computing: Theories and Applications, ed: Springer, 2018, pp. 51-58.

[53] V. Venkataiah, R. Mohanty, J. Pahariya, and M. Nagaratna, "Application of Ant Colony Optimization Techniques to Predict Software Cost Estimation," in Computer Communication, Networking and Internet Security, ed: Springer, 2017, pp. 315-325.

[54] R. Puri and I. Kaur, "Human Opinion Dynamics for Software Cost Estimation," in Proceedings of Fifth International Conference on Soft Computing for Problem Solving, 2016, pp. 375-386.

[55] Z. H. Wani and S. Quadri, "Artificial Bee Colony-Trained Functional Link Artificial Neural Network Model for Software Cost Estimation," in Proceedings of Fifth International Conference on Soft Computing for Problem Solving, 2016, pp. 729-741.

[56] A. Idri and S. Cherradi, "Improving effort estimation of Fuzzy Analogy using feature subset selection," in Computational Intelligence (SSCI), 2016 IEEE Symposium Series on, 2016, pp. 1-8.

[57] M. Baiquni and R. Sarno, "Improving the accuracy of COCOMO II using fuzzy logic and local calibration method," in Science in Information Technology (ICSITech), 2017 3rd International Conference on, 2017, pp. 284-289.

[58] M. Algabri, F. Saeed, H. Mathkour, and N. Tagoug, "Optimization of soft cost estimation using genetic algorithm for NASA software projects," in Information Technology: Towards New Smart World (NSITNSW), 2015 5th National Symposium on, 2015, pp. 1-4.

[59] S. Girotra and K. Sharma, "Tuning of software cost drivers using BAT algorithm," in Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on, 2016, pp. 1051-1056.

[60] T. R. Benala, R. Mall, S. Dehuri, and P. Swetha, "Software effort estimation using functional link neural networks tuned with active learning and optimized with particle swarm optimization," in International Conference on Swarm, Evolutionary, and Memetic Computing, 2014, pp. 223-238.

[61] P. S. Rao and R. K. Kumar, "Software Effort Estimation through a Generalized Regression Neural Network," in Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India (CSI) Volume 1, 2015, pp. 19-30.

[62] M. Padmaja and D. Haritha, "Software Effort Estimation Using Grey Relational Analysis with K-Means Clustering," in Information Systems Design and Intelligent Applications, ed: Springer, 2018, pp. 924-933.

[63] F. Fellir, K. Nafil, R. Touahni, and L. Chung, "Improving Case Based Software Effort Estimation Using a Multi-criteria Decision Technique," in Computer Science On-line Conference, 2018, pp. 438-451.

[64] S. A. Shahwaiz, A. A. Malik, and N. Sabahat, "A parametric effort estimation

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

model for mobile apps," in Multi-Topic Conference (INMIC), 2016 19th International, 2016, pp.6-1 .

[65] P. S. Rao, K. K. Reddi, and R. U. Rani, "Optimization of neural network for software effort estimation," in Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET), 2017 International Conference on, 2017, pp. 1-7.

[66] I. M. M. Parwita, R. Sarno, and A. Puspaningrum, "Optimization of COCOMO II coefficients using Cuckoo optimization algorithm to improve the accuracy of effort estimation," in Information & Communication Technology and System (ICTS), 2017 11th International Conference on, 2017, pp. 99-104.

[67] M. Owais and R. Ramakishore, "Effort, duration and cost estimation in agile software development," in Contemporary Computing (IC3), 2016 Ninth International Conference on, 2016, pp. 1-5.

[68] K. Moharreri, A. V .Sapre, J. Ramanathan, and R. Ramnath, "Cost-effective supervised learning models for software effort estimation in agile environments," in Computer Software and Applications Conference (COMPSAC), 2016 IEEE 40th Annual, 2016, pp. 135-140.

[69] K. Langsari and R. Sarno, "Optimizing effort and time parameters of COCOMO II estimation using fuzzy multi-objective PSO," in Electrical Engineering, Computer Science and Informatics (EECSI), 2017 4th International Conference on, 2017, pp. 1-6.

[70] E. Khatibi and V .K. Bardsiri, "Model to estimate the software development effort based on in-depth analysis of project attributes," IET software, vol. 9, pp. 109-118, 2015.

[71] R. Jain, V. K. Sharma, and S. Hiranwal, "Reduce mean magnitude relative error in software cost estimation by HOD-COCOMO algorithm," in Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2016 International Conference on, 2016, pp. 708-712.

[72] A. BaniMustafa, "Predicting Software Effort Estimation Using Machine Learning Techniques," in 2018 8th International Conference on Computer Science and Information Technology (CSIT), 2018, pp. 249-256.

[73] M. Azzeh and A. B. Nassif, "Analogy-based effort estimation: a new method to discover set of analogies from dataset characteristics," IET Software, vol. 9, pp. 39-50, 2015.

[74] J. H. Wu and J. W. Keung, "Utilizing cluster quality in hierarchical clustering for analogy-based software effort estimation," in Software Engineering and Service Science (ICSESS), 2017 8th IEEE International Conference on, 2017, pp. 1-4.

[75] A. Tiwari and A. Chaturvedi, "Class partition approach for software effort estimation using support vector machine," in Electrical, Computer and Electronics Engineering (UPCON), 2016 IEEE Uttar Pradesh Section International Conference on, 2016, pp. 653-659.

[76] S. Nanda and B. Soewito, "Modeling software effort estimation using hybrid PSO-ANFIS," in Intelligent Technology and Its Applications (ISITIA), 2016 International Seminar on, 2016, pp. 219-224.

[77] F .Soltanveis and S. H. Alizadeh, "Using parametric regression and KNN algorithm with missing handling for software effort prediction," in Artificial Intelligence and Robotics (IRANOPEN), 2016, 2016, pp. 77-84.

[78] D. Sivakumar and C. Sureshkumar, "Effort estimation of software projects with optimized coefficients using soft computing technique," in Emerging Devices and Smart Systems (ICEDSS), 2017 Conference on, 2017, pp. 84-89.

[79] H. K. Sharma, R. Tomar, J. Patni, and A. Dumka, "E-COCOMO: An effort estimation model for cleanroom software development approach," in Next Generation Computing Technologies (NGCT), 2016 2nd International Conference on, 2016, pp. 131-136.

[80] F. S. Gharehchopogh, R. Rezaii, and B. Arasteh, "A new approach by using tabu search and genetic algorithms in software cost estimation," in Application of Information and Communication Technologies (AICT), 2015 9th International Conference on, 2015, pp. 113-117.

[81] S. Garg and D. Gupta, "PCA based cost estimation model for agile software development projects," in Industrial Engineering and Operations Management (IEOM), 2015 International Conference on, 2015, pp. 1-7.
[82] R. S. Dewi, G. F. Prassida, and A. P. Subriadi, "UCPabc as an integration model for software cost estimation," in Science in Information Technology (ICSITech), 2016 2nd International Conference on, 2016, pp. 187-192.
[83] S. Chhabra and H. Singh, "Simulink based fuzzified COCOMO," in Contemporary Computing and Informatics (IC3I), 2016 2nd International Conference on, 2016, pp. 847-851.
[84] M. El Bajta, "Analogy-based software development effort estimation in global software development," in Global Software Engineering Workshops (ICGSEW), 2015 IEEE 10th International Conference on, 2015, pp. 51-54.
[85] P. Agrawal and S. Kumar, "Early phase software effort estimation model," in Colossal Data Analysis and Networking (CDAN), Symposium on, 2016, pp. 1-8.
[86] N. Gupta and K. Sharma, "Optimizing intermediate COCOMO model using BAT algorithm," in Computing for Sustainable Global Development (INDIACom), 2015 2nd International Conference on, 2015, pp. 1649-1653.
[87] S. Bagheri and A. Shameli-Sendi, "Software Project Estimation Using Improved Use Case Point," in 2018 IEEE 16th International Conference on Software Engineering Research, Management and Applications (SERA), 2018, pp. 143-150.
[88] R. Y. Pratama and R. Sarno, "Optimizing COCOMO II parameters using artificial bee colony method," in Information & Communication Technology and System (ICTS), 2017 11th International Conference on, 2017, pp. 125-130.
[89] K. Langsari and R. Sarno, "Optimizing COCOMO II parameters using particle swarm method," in Science in Information Technology (ICSITech), 2017 3rd International Conference on, 2017, pp. 29-34.
[90] H. Karna and S .Gotovac, "Estimating software development effort using Bayesian networks," in Software, Telecommunications and Computer Networks (SoftCOM), 2015 23rd International Conference on, 2015, pp. 229-233.
[91] S. S. Jafari and F. Ziaaddini, "Optimization of software cost estimation using harmony search algorithm," in Swarm Intelligence and Evolutionary Computation (CSIEC), 2016 1st Conference on, 2016, pp. 131-135.
[92] K. Iwata, T. Nakashima, Y. Anan, and N. Ishii, "Effort estimation for embedded software development projects by combining machine learning with classification," in Applied Computing and Information Technology/3rd Intl Conf on Computational Science/Intelligence and Applied Informatics/1st Intl Conf on Big Data, Cloud Computing, Data Science & Engineering (ACIT-CSII-BCD), 2016 4th Intl Conf on, 2016, pp. 265-270.
[93] V.-S. Ionescu, "An approach to software development effort estimation using machine learning," in Intelligent Computer Communication and Processing (ICCP), 2017 13th IEEE International Conference on, 2017, pp. 197-203.
[94] M. Hosni, A. Idri, A. B. Nassif, and A. Abran, "Heterogeneous ensembles for software development effort estimation," in Soft Computing & Machine Intelligence (ISCMI), 2016 3rd International Conference on, 2016, pp.178-174 .
[95] F. S. Gharehchopogh, I. Maleki, and A. Talebi, "Using hybrid model of artificial bee colony and genetic algorithms in software cost estimation," in Application of Information and Communication Technologies (AICT), 2015 9th International Conference on, 2015, pp. 102-106.
[96] M. Kaur and S. K. Sehra, "Particle swarm optimization based effort estimation using Function Point analysis," in 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 ,pp. 140-145.
[97] R. Gupta, N. Chaudhary, and S. K. Pal, "Hybrid model to improve bat algorithm performance," in 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2014, pp. 1967-1970.
[98] Z. Dan, "Improving the accuracy in software effort estimation: Using artificial neural network model based on particle

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

swarm optimization," in Proceedings of 2013 IEEE International Conference on Service Operations and Logistics, and Informatics, 2013, pp. 180-185.

[99] F .A. Amazal, A. Idri, and A. Abran, "Improving fuzzy analogy based software development effort estimation," in 2014 21st Asia-Pacific Software Engineering Conference, 2014, pp. 247-254.

[100] 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, pp. 144-160, 2013.

[101] Y.-S. Seo, D.-H. Bae, and R. Jeffery, "AREION: Software effort estimation based on multiple regressions with adaptive recursive data partitioning," Information and Software technology, vol. 55, pp. 1710-1725, 2013.

[102] S. P. Singh and A. Kumar, "Software cost estimation using homeostasis mutation based differential evolution," in Intelligent Systems and Control) ISCO), 2017 11th International Conference on, 2017, pp. 173-181.

[103] R. Sarno and J. Sidabutar, "Improving the accuracy of COCOMO's effort estimation based on neural networks and fuzzy logic model," in Information & Communication Technology and Systems) ICTS), 2015 International Conference on, 2015, pp. 197-202.

[104] S. Ezghari, A. Zahi, and A. Idri, "A learning adaptation cases technique for fuzzy analogy-based software development effort estimation," in 2014 Second World Conference on Complex Systems (WCCS), 2014, pp. 492-497.

[105] N. Kushwaha, "Software cost estimation using the improved fuzzy logic framework," in 2014 Conference on IT in Business, Industry and Government (CSIBIG), 2014, pp. 1-5.

[106] Q. Liu, X. Chu, J. Xiao, and H. Zhu, "Optimizing non-orthogonal space distance using pso in software cost estimation," in 2014 IEEE 38th Annual Computer Software and Applications Conference, 2014, pp. 21-26.

[107] Q. Liu, S. Shi, H. Zhu, and J. Xiao, "A mutual information-based hybrid feature selection method for software cost estimation using feature clustering," in 2014 IEEE 38th Annual Computer Software and Applications Conference, 2014, pp. 27-32.

[108] M. Madheswaran and D. Sivakumar, "Enhancement of prediction accuracy in COCOMO model for software project using neural network," in Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2014, pp. 1-5.

[109] D. Manikavelan and R. Ponnusamy, "Software cost estimation by analogy using feed forward neural network," in International Conference on Information Communication and Embedded Systems (ICICES2014), 2014, pp. 1-5.

[110] P. Phannachitta, J. Keung, A. Monden, and K.-i. Matsumoto, "Improving analogy-based software cost estimation through probabilistic-based similarity measures," in 2013 20th Asia-Pacific Software Engineering Conference (APSEC), 2013, pp. 541-546.

[111] L. V. Patil, R. M. Waghmode, S. Joshi, and V. Khanna, "Generic model of software cost estimation: A hybrid approach," in 2014 IEEE International Advance Computing Conference (IACC), 2014, pp. 1379-1384.

[112] A. Sinhal and B. Verma, "A novel fuzzy based approach for effort estimation in software development," ACM SIGSOFT Software Engineering Notes, vol. 38, pp. 1-6, 2013.

[113] D. Kashyap and A. K. Misra, "Software development cost estimation using similarity difference between software attributes," in Proceedings of the 2013 International Conference on Information Systems and Design of Communication, 2013, pp. 1-6.

[114] F.-A. Amazal, A .Idri, and A. Abran, "An analogy-based approach to estimation of software development effort using categorical data," in 2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement, 2014, pp. 252-262.

[115] M. Azzeh and A. B. Nassif, "Fuzzy model tree for early effort estimation," in 2013 12th International Conference on Machine Learning and Applications, 2013, pp. 117-121.

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[116] S. Mukherjee and R. K. Malu, "Optimization of project effort estimate using neural network," in 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, 2014, pp. 406-410.
[117] Ö. F. Saraç and N. Duru, "A novel method for software effort estimation: Estimating with boundaries," in 2013 IEEE INISTA, 2013, pp. 1-5.
[118] S. Saraswathi and N. Kannan, "Soft computing based estimation of software development effort," in International Conference on Information Communication and Embedded Systems (ICICES2 ,2014 ,(014pp. 1-6.
[119] S. M. Satapathy, M. Kumar, and S. K. Rath, "Class point approach for software effort estimation using soft computing techniques," in 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI ,2013 ,(pp. 178-183.
[120] M. Ochodek, J. Nawrocki, and K. Kwarciak, "Simplifying effort estimation based on Use Case Points," Information and Software Technology, vol. 53, pp. 200-213, 2011.
[121] M. Azzeh, D. Neagu, and P. I. Cowling, "Analogy-based software effort estimation using Fuzzy numbers," Journal of Systems and Software, vol. 84, pp. 270-284, 2011.
[122] K. Ponnalagu and N. Narendra, "Automated trendline generation for accurate software effort estimation," in Proceedings of the 3rd annual conference on Systems, programming, and applications: software for humanity, 2012, pp. 203-212.
[123] M. Azzeh, "Software effort estimation based on optimized model tree," in Proceedings of the 7th International Conference on Predictive Models in Software Engineering, 2011, pp. 1-8.
[124] A. Mittal, K. Parkash, and H. Mittal, "Software cost estimation using fuzzy logic," ACM SIGSOFT Software Engineering Notes, vol. 35, pp. 1-7, 2010.
[125] S. M. Satapathy, M. Kumar, and S. K. Rath, "Fuzzy-class point approach for software effort estimation using various adaptive regression methods," CSI transactions on ICT, vol. 1, pp. 367-380, 2013.
[126] A. Kaushik, A. Soni, and R. Soni, "An improved functional link artificial neural networks with intuitionistic fuzzy clustering for software cost estimation," International Journal of System Assurance Engineering and Management, vol. 7, pp. 50-61, 2016.
[127] J. Shivhare and S. K. Rath, "Software effort estimation using machine learning techniques," in Proceedings of the 7th India Software Engineering Conference, 2014, pp. 1-6.
[128] M. Tsunoda, A. Monden, J. Keung, and K. Matsumoto, "Incorporating expert judgment into regression models of software effort estimation," in 2012 19th Asia-Pacific Software Engineering Conference ,2012 ,pp. 374-379.
[129] A. L. Oliveira, P. L. Braga, R. M. 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, pp. 1155-1166, 2010.
[130] R. d. A. Araújo, S. Soares, and A. L. Oliveira, "Hybrid morphological methodology for software development cost estimation," Expert Systems with Applications, vol. 39, pp. 6129-6139, 2012.
[131] T. R. Benala, R .Mall, S. Dehuri, and K. C. Babu, "Software effort prediction using unsupervised learning (clustering) and functional link artificial neural networks," in 2012 World Congress on Information and Communication Technologies, 2012, pp. 115-120.
[132] R. d. A .Araújo, A. L. Oliveira, and S. Soares, "A shift-invariant morphological system for software development cost estimation," Expert Systems with Applications, vol. 38, pp. 4162-4168, 2011.
[133] R. d. A. Araújo, A. L. Oliveira, S. Soares, and S. Meira, "An evolutionary morphological approach for software development cost estimation," Neural Networks, vol. 32, pp. 285-291, 2012.
[134] M. Kazemifard, A. Zaeri, N. Ghasem-Aghaee, M. A. Nematbakhsh, and F. Mardukhi, "Fuzzy emotional COCOMO II software cost estimation (FECSCE) using multi-agent systems," Applied Soft Computing, vol. 11, pp. 2260-2270, 2011.

[135] S. Malathi and S. Sridhar, "Optimization of Fuzzy Analogy in Software cost estimation using linguistic Variables," Procedia Engineering, vol. 38, pp. 17-7 .2012 ,190
[136] M. Azzeh, "Model tree based adaption strategy for software effort estimation by analogy," in 2011 IEEE 11th International Conference on Computer and Information Technology, 2011, pp. 328-335.
[137] I. Attarzadeh, A. Mehranzadeh, and A . Barati, "Proposing an enhanced artificial neural network prediction model to improve the accuracy in software effort estimation," in 2012 Fourth International Conference on Computational Intelligence, Communication Systems and Networks, 2012, pp. 167-172.
[138] I. Attarzadeh and S. H. Ow, "Improving estimation accuracy of the COCOMO II using an adaptive fuzzy logic model," in 2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011), 2011, pp. 2458-2464.
[139] R. d. A. Araújo, A. L. Oliveira, S .Soares, and S. Meira, "Gradient-based morphological approach for software development cost estimation," in The 2011 International Joint Conference on Neural Networks, 2011, pp. 588-594.
[140] A. B. Nassif, L. F. Capretz, and D. Ho, "Estimating software effort using an ANN model based on use case points," in 2012 11th International Conference on Machine Learning and Applications, 2012, pp. 42-47.
[141] D. Wu, J. Li, and Y. Liang, "Linear combination of multiple case-based reasoning with optimized weight for software effort estimation," The Journal of Supercomputing, vol. 64, pp. 898-918, 2013.
[142] E. Kocaguneli, A. Tosun, and A. Bener, "Ai-based models for software effort estimation," in 2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications, 2010, pp. 323-326.
[143] M. Jørgensen, "Contrasting ideal and realistic conditions as a means to improve judgment-based software development effort estimation," Information and Software Technology, vol. 53, pp. 1382-1390, 2011.
[144] M. Sadiq, F. Mariyam, A. Alil, S. Khan, and P. Tripathi, "Prediction of software project effort using fuzzy logic," in 2011 3rd International Conference on Electronics Computer Technology, 2011, pp. 353-358.
[145] A. B. Nassif, L. F. Capretz, and D. Ho, "Estimating software effort based on use case point model using sugeno fuzzy inference system," in 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence, 2011, pp. 393-398.
[146] A. B. Nassif, L. F. Capretz, and D. Ho, "Software effort estimation in the early stages of the software life cycle using a cascade correlation neural network model," in 2012 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2012 ,pp. 589-594.
[147] J.-C. Lin and H.-Y. Tzeng, "Applying particle swarm optimization to estimate software effort by multiple factors software project clustering," in 2010 International Computer Symposium (ICS2010), 2010, pp. 1039-1044.
[148] J.-C. Lin, C-.T. Chang, and S.-Y. Huang, "Research on software effort estimation combined with genetic algorithm and support vector regression," in 2011 International Symposium on Computer Science and Society, 2011, pp. 349-352.
[149] J. S. Kumar, T. GovindaRao, Y. NagaBabu, S. Chaitanya, and K. Subrahmanyam, "A novel method for software effort estimation using inverse regression as firing interval in fuzzy logic," in 2011 3rd International Conference on Electronics Computer Technology, 2011, pp. 177-182.
[150] A. Kaushik, A. Soni, and R. Soni, "An adaptive learning approach to software cost estimation," in 2012 National Conference on Computing and Communication Systems, 2012, pp. 1-6.
[151] C. V. Hari, T. S. Sethi, B. Kaushal, and A. Sharma, "CPN-a hybrid model for software cost estimation," in 2011 IEEE Recent Advances in Intelligent Computational Systems, 2011, pp. 902-906.
[152] I. Attarzadeh and S. H. Ow, "A novel soft computing model to increase the accuracy of software development cost estimation,"

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

in 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE), 2010, pp. 603-607.

[153] A. L. de Oliveira and S. Soares, "Hybrid intelligent design of morphological-rank-linear perceptrons for software development cost estimation," in 2010 22nd IEEE International Conference on Tools with Artificial Intelligence, 2010, pp. 160-167.

[154] S. Aljahdali and A. F. Sheta, "Software effort estimation by tuning COOCMO model parameters using differential evolution," in ACS/IEEE International Conference on Computer Systems and Applications-AICCSA 2010, 2010, pp. 1-6.

[155] F. S. Alaa and A. Al-Afeef, "A GP effort estimation model utilizing line of code and methodology for NASA software projects," in 2010 10th International Conference on Intelligent Systems Design and Applications, 2010, pp. 290-295.

[156] C. V. Hari, P. R. PVGD, M. Jagadeesh, and G. S. Ganesh, "Interval type-2 fuzzy logic for software cost estimation using TSFC with mean and standard deviation," in 2010 International Conference on Advances in Recent Technologies in Communication and Computing, 2010, pp. 40-44.

[157] A. B. Nassif, L. F. Capretz, D. Ho, and M. Azzeh, "A treeboost model for software effort estimation based on use case points," in 2012 11th International Conference on Machine Learning and Applications, 2012, pp. 314-319.

[158] P. Jodpimai, P. Sophatsathit, and C. Lursinsap, "Estimating software effort with minimum features using neural functional approximation," in 2010 International Conference on Computational Science and Its Applications, 2010, pp. 266-273.

[159] U. R. Saxena and S. Singh, "Software effort estimation using neuro-fuzzy approach," in 2012 CSI Sixth International Conference on Software Engineering (CONSEG), 2012, pp. 1-6.

© 2005 – ongoing JATIT & LLS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Appendix A. Studies table

Study ID	S1	S2	S3	S4	S5	S6	S7	S8
References	[31]	[30]	[29]	[28]	[27]	[26]	[25]	**[24]**
Study ID	S9	S10	S11	S12	S13	S14	S15	**S16**
References	[93] [31] (93) (93)	[38]	[37]	[36]	[35]	[34]	[33]	**[32]**
Study ID	S17	S18	S19	S20	S21	S22	23	**24**
References	[47]	[46]	[45]	[44]	[34] [43] (34) (34)	[42]	[41]	**[40]**
Study ID	25	S26	S27	S28	S29	S30	S31	**S32**
References	[55]	[54]	[53]	[52]	[51]	[50]	[49]	**[48]**
Study ID	S33	S34	S35	S36	S37	S38	S39	**S40**
References	[63]	[62]	[61]	[60]	[59]	[58]	[57]	**[56]**
Study ID	S41	S42	S43	S44	S45	S46	S47	**S48**
References	[71]	[70]	[69]	[68]	[67]	[66]	[65]	**[64]**
Study ID	S49	S50	S51	S52	S53	S54	S55	**S56**
References	[79]	[78]	[77]	[76]	[75]	[74]	[73]	**[72]**
Study ID	S57	S58	S59	S60	S61	S62	S63	**S64**
References	[87]	[86]	[85]	[84]	[83]	[82]	[81]	**[80]**
Study ID	S65	S66	S67	S68	S69	S70	S71	**S72**
References	[95]	[94]	[93]	[92]	[91]	[90]	[89]	**[88]**
Study ID	S73	S74	S75	S76	S77	S78	S79	**S80**
References	[103]	[102]	[101]	[100]	[99]	[98]	[97]	**[96]**
Study ID	S81	S82	S83	S84	S85	S86	S87	**S88**
References	[111]	[110]	[109]	[108]	[107]	[106]	[105]	**[104]**
Study	S89	S90	S91	S92	S93	S94	S95	**S96**

ID								
References	[119]	[118]	[117]	[116]	[115]	[114]	[113]	**[112]**
Study ID	S97	S98	S99	S100	S101	S102	S103	**S104**
References	[127]	[126]	[125]	[124]	[123]	[122]	[121]	**[120]**
Study ID	S105	S106	S107	S108	S109	S110	S111	**S112**
References	[135]	[134]	[133]	[132]	[131]	[130]	[129]	**[128]**
Study ID	S113	114	S115	S116	S117	S118	S119	**S120**
References	[143]	[142]	[141]	[140]	[139]	[138]	[137]	**[136]**
Study ID	S121	S122	S123	S124	S125	S126	S127	**S128**
References	[151]	[150]	[149]	[148]	[147]	[146]	[145]	**[144]**
Study ID	S129	S130	S131	S132	S133	S134	S135	**S136**
References	[159]	[158]	[157]	[156]	[155]	[154]	[153]	**[152]**