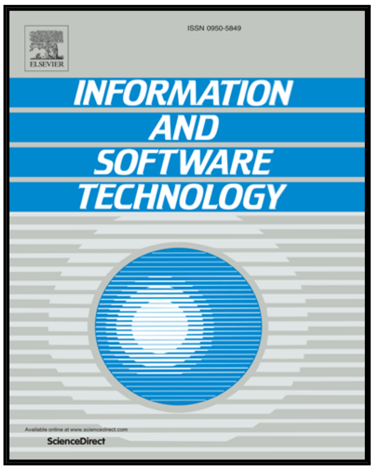
Accepted Manuscript



Research Patterns and Trends in Software Effort Estimation

Sumeet Kaur Sehra, Yadwinder Singh Brar, Navdeep Kaur, Sukhjit Singh Sehra

PII: S0950-5849(17)30431-7
DOI: 10.1016/j.infsof.2017.06.002
Reference: INFSOF 5836

To appear in: *Information and Software Technology*

Received date: 14 August 2016
Revised date: 23 March 2017
Accepted date: 7 June 2017

Please cite this article as: Sumeet Kaur Sehra, Yadwinder Singh Brar, Navdeep Kaur, Sukhjit Singh Sehra, Research Patterns and Trends in Software Effort Estimation, *Information and Software Technology* (2017), doi: 10.1016/j.infsof.2017.06.002

Highlights

- This study identified research trends prevailing in software effort estimation literature.
- Latent Dirichlet Allocation (LDA) was applied to the corpus of 1178 articles.
- This study established the semantic mapping between research patterns and trends.

Research Patterns and Trends in Software Effort Estimation

Sumeet Kaur Sehra*

I.K.G. Punjab Technical University, Jalandhar, Punjab, India

Guru Nanak Dev Engineering College, Ludhiana, Punjab, India

Yadwinder Singh Brar

I.K.G. Punjab Technical University, Jalandhar, Punjab, India

Navdeep Kaur

Sri Guru Granth Sahib World University, Fatehgarh Sahib, Punjab, India

Sukhjit Singh Sehra

Guru Nanak Dev Engineering College, Ludhiana, Punjab, India

Abstract

Context. Software effort estimation (SEE) is most crucial activity in the field of software engineering. Vast research has been conducted in SEE resulting into a tremendous increase in literature. Thus it is of utmost importance to identify the core research areas and trends in SEE which may lead the researchers to understand and discern the research patterns in large literature dataset.

Objective. To identify unobserved research patterns through natural language processing from a large set of research articles on SEE published during the period 1996 to 2016.

Method. A generative statistical method, called Latent Dirichlet Allocation (LDA), applied on a literature dataset of 1178 articles published on SEE.

Results. As many as twelve core research areas and sixty research trends have been revealed; and the identified research trends have been semantically mapped to associate core research areas.

Conclusion. This study summarises the research trends in SEE based upon a corpus of 1178 articles. The patterns and trends identified through this research can help in finding the potential research areas.

Keywords:
Software Effort Estimation, Latent Dirichlet Allocation, Research Trends

1. Introduction

Software Engineering (SE) discipline has evolved since 1960s and has garnered significant knowledge [1]. Over the years, there has been a strong criticism of SE research as it advocates more than it evaluates [2]. Many researchers have attempted to characterize software engineering research, but they failed to present a comprehensive picture [3, 4].

*corresponding author
Sumeet Kaur Sehra, is pursuing PhD from IKG, Punjab Technical University, Jalandhar, Punjab. She is working as Assistant Professor (CSE) at Guru Nanak Dev Engineering College, Ludhiana, Punjab, India.
Email address: sumeetksehra@gmail.com (Sumeet Kaur Sehra*)

SEE predicts the effort to accomplish development or maintenance tasks based on data which is generally incomplete, uncertain and noisy. Problems and issues in SEE have been addressed by researchers and practitioners from time to time. But much of the research has its focus on construction of formal SEE models [5]. The models designed by researchers have known advantages and disadvantages. The vast available literature on the subject posed a challenge before the researchers to review and identify the right path for their research.

The literature can be reviewed manually or algorithmically. The manual review provides an insight into the literature, but it is never free from biasness as researchers remain inclined towards more cited papers [6]. Natural language processing provides a powerful algorithm that extracts unobserved trends from a large collection of documents. Unlike manual tagging, which is effort intensive and requires expertise in the documents' subject-matter, algorithmic-based analysis is an automated process [7, 8, 9] called topic modelling. It takes a corpus, identifies the patterns and adds semantic meaning to the vocabulary. Both clustering and topic analysis approaches can be used with topic modelling. But as suggested by Evangelopoulos et al. [10], topic analysis is more appropriate relative to clustering for identification of research trends underlying the dataset. In topic analysis, a document is assigned to a mixture of topics, whereas in the case of clustering, each document is forced to join exactly one cluster. In this review, topic analysis and labelling have been incorporated to identify the latent patterns and trends in dataset. Two leading topic modelling techniques are Latent Semantic Indexing (LSI) [11] and Latent Dirichlet Allocation (LDA) [12]. In SE, LDA has been applied for mining software repositories [13], bug localization [14], defect prediction [15], software categorisation [16], classification of change messages [17] and software evolution [18].

Research patterns in SEE have been systematically identified and represented in this study by applying LDA to a corpus of 1178 articles published during the period 1996 to 2016. As many as twelve core SEE research areas and sixty research trends have emerged after the analysis of titles and abstracts of research articles. Semantic linking between sixty specific research trends and twelve core areas has been identified and presented. The review has been undertaken systematically keeping in view the guidelines proposed by [19, 20]. This review is intended to find the answer to following research questions:

- RQ1. Which research areas have been explored mostly by the researchers?
- RQ2. What research methods have been used for SEE?
- RQ3. Which research areas demand greater attention of researchers?

The paper has been divided into eight sections. The following section describes software effort estimation and related work in detail. The third section explains the process for data collection. The fourth section discusses the pre-processing and application of LDA to the corpus. The results and findings are presented in the fifth section. The sixth section provides the answers to research questions. The seventh section explains the threats to validity of the study. The final section concludes the paper.

2. Software effort estimation

SEE is one of the most challenging and important aspects in project management. Numerous estimation methods have been proposed by the researchers since the inception of SE as a research area [5, 21, 22]. Some of the studies on SEE have been reviewed in Table 1. There are several factors which influence the task of SEE. One of them is lack of information from the past including both experts' knowledge and experience. Another important problem, which is probably the origin of all other problems, is the nature of SE projects which are largely human-based, not repeatable, and affected by change. These aspects, in particular, make SEE difficult.

Other critical issue in SEE is project data, which is often incomplete, inconsistent, uncertain and unclear [23]. Since effort estimates act as base point for many project management activities including planning, budgeting and scheduling, it is very crucial to obtain almost accurate estimates. Moløkken-Østvold et al. [24] have reported in their study that there is overestimation by 30-40% on average in software projects. Simmons and Korrapati [25] have also supported this claim by explaining that 52.7 % of projects cost 189 % of their original estimates, which means that there is approximately 89% overrun in these projects. Estimation of effort in software systems is a challenging and substantial job for project managers. The challenge arises due to various factors, including the human factor, the complexity of the product, different development platforms, etc. [26]. Estimators use different methods for estimation. They employ a single technique (formal model or expert judgement) or both the techniques for estimation.

Numerous studies are available on software effort estimation. These have been listed and reviewed in Table 1. But none of the studies has discussed the patterns and trends in SEE by applying topic modelling technique. The review has been undertaken to identify different core research areas and trends prevailing in SEE.

3. Collection of data

The research data was collected from various online databases, journals, and conference proceedings. The data collection process comprises of the following steps:

Identifying information sources. The databases related to the research were identified. They included the esteemed software engineering journals and proceedings of various conferences.

Defining search criteria. The search keywords were decided based upon the research questions of the current study and adapted from the study of [37]. The search phrases identified were "software effort estimation", "software cost estimation", "project management" and "size metrics". The search string used for searching was "software effort estimation" OR "software cost estimation" OR "size metrics" OR "software project management". Search terms "software project management" and "size metrics" were included to broaden the search space for required articles on SEE. The search criteria conformed to relevancy and recency.

Search bibliographic databases. An automatic search was made through relevant sources of information using defined search criteria by open source tool JabRef [38] and search engines of specific publishers. The bibliographic databases of ScienceDirect, IEEExplore, Wiley and DBLP were searched; and identified articles were added to BibTeX database of JabRef. The bibliographic database search was meant for searching specific keywords in the publication title, abstract and keywords. As many as 1420 articles were collected in the BibTeX database. However, 1298 articles remained in the database after removal of duplicate entries.

Initial review. The literature dataset collected was reviewed by analysing the titles and abstracts using JabRef's search query for inclusion in the corpus. The papers were included in the corpus based upon inclusion and exclusion criteria given as here under:

Inclusion criteria. Under this criteria, the articles must be published in English. These must have their focus on software effort, cost or size estimation in any context from the year 1996 to 2016.

Exclusion criteria. The articles published before the year 1996 or not reporting on development related to the search keywords discussed in search criteria were excluded from the literature dataset. However, the papers describing the same study in more than one publication were not excluded.

Full review. The publications not conforming to the defined inclusion criteria were reviewed manually by analysing the metadata of their BibTeX entries in JabRef. After this step, 1178 publications considered relevant for the purpose of current research were identified.

The process followed for document collection is exhibited in Table 2. Figure 1 depicts the year-wise publications of collected SEE research literature dataset for the period 1996-2016. Further, Figures 2 and 3 highlight the necessary data about top ten authors and journals respectively.

4. Methodological analysis

4.1. Latent Dirichlet Allocation

LDA is applied to the literature dataset (corpus) to facilitate retrieving and querying a large corpus of data to identify latent ideas that describe the corpus as a whole [12]. In LDA, a document is considered as a mixture of a limited number of latent topics, and each keyword in the document is associated with one of these topics. Using latent clues, topic model connects similar meaning words and differentiates different meaning words [6, 39]. So, latent topics represent multiple observed entities having similar patterns identified from the corpus. Table 4 depicts the relevant entities represented through twelve latent topics. Loadings for

Table 1: Review of literature

Author	Findings of the study
Wen et al. [27]	The authors have investigated 84 primary studies of machine learning (ML) techniques in SEE for finding out different ML techniques, their estimation accuracy, the comparison between different models and estimation contexts. Based upon this study, they found out that in SEE, eight types of ML techniques have been applied; and concluded that ML models provide more accurate estimates as compared to non-ML models.
Jørgensen [28]	Jorgensen has reviewed 15 studies on expert estimation for validating the conformance to twelve expert estimation 'best practices' and found expert estimation as the dominant approach for SEE and no evidence in favour of model estimates over expert estimates.
Idri et al. [29]	The authors have performed a systematic review of 24 ensemble effort estimation (EEE) studies published between the period 2000 to 2016. They have identified two types of EEE, namely, homogeneous and heterogeneous. They also found that the estimation accuracy of EEE techniques is better than single models.
Trendowicz et al. [5]	This study has reviewed surveys for finding out the industrial objectives, the ability of software organizations to use estimation methods, and practically applied practices of SEE in organisations.
Idri et al. [30]	The authors have identified and reviewed 65 studies published on analogy-based Software Effort Estimation (ASEE) from the year 1990 to 2012; and revealed that the main focus of the research is on feature and case subset selection. They concluded that ASEE methods surmount when compared with eight techniques and can provide more accurate results in combination with fuzzy logic (FL) or genetic algorithms (GA).
Kitchenham et al. [31]	In this study, circumstances have been identified under which organisations should use cross-company dataset. The authors have reviewed 10 primary studies reporting comparative analysis of within and cross-company effort predictions. Three studies suggested that cross-company predictions perform equally good as within-company predictions, and four studies revealed the dominance of within-company predictions over cross-company predictions.
Sigweni and Shepperd [32]	The authors have conducted a review of published primary studies on feature weighting techniques (FWT) from the year 2000 to 2014 to determine whether FWTs lead to improved predictions based upon four parameters, namely, approach, strengths and weaknesses, performance and experimental evaluation. They have recommended the adoption of FWTs in the research.
Grimstad et al. [33]	The authors have argued that the missing software effort estimation terminology is a critical hindrance in estimation accuracy. They have reviewed software textbooks, SEE research papers and found limitations in use of estimation terminology. They also suggested guidelines to overcome this limitation.
Britto et al. [34]	This study has reviewed SEE in the context of global software development. The authors have used eight available studies; and concluded that there is a good scope for research in the global context.
Andrew and Selamat [35]	The authors have systematically analysed research works done on missing data imputation techniques from 2000-2012 to estimate software effort.They have presented the leading researchers, the current state of research and amount of work done in the area of missing data techniques in software effort estimation.
Usman et al. [36]	In this study, a review of effort estimation in agile software development (ASD) based on 20 papers has been presented. The study has concluded that most of the techniques are based on expert judgement, and use XP and Scrum only.

Table 2: Document collection process

Round #	Search criteria	Number of articles
1	Search phrases/Within databases/In Title OR Abstract OR Keywords	1420
2	Elimination of duplicate and irrelevant articles	242
Literature dataset		1178

terms and documents were generated by applying LDA to the corpus. The loading value for each topic indicates the extent of relation of the related term/document with certain topic solution. Core research areas and trends are represented by lower level and high level topic solutions respectively [40].

4.2. Pre-processing of data

Pre-processing phase involves the elimination of noisy words/characters from the dataset. The following steps have been incorporated for pre-processing the literature dataset:

1. Loading text
2. Lexical analysis of the text
3. Removing stop words
4. Stemming.

Loading text. The corpus in JabRef is imported into a .csv file; and required information from the bibliographic database is filtered by using JabRef export filter, designed exclusively for this purpose [41].

Lexical analysis. In lexical analysis, 1178 titles and abstracts of the articles were tokenised into 85,157 tokens. Generated tokens were converted into lowercase letters for each document. The elimination of punctuation characters, exclamation points, commas, apostrophe, question marks, quotation marks and hyphen was performed. Further, numeric values were removed to get only the textual tokens.

Stop-word removal. The common English words as given in nltk python package [42] and the phrases used to develop the literature dataset were removed.

Stemming. For preparing an effective literature dataset word forms are stemmed to their original root form. SnowballC stemmer algorithm [43] was used to stem the tokens for each document, and converted the inflected words to their base stem.

4.3. Applying LDA

LDA is applied to pre-processed corpus data as suggested by [12, 44, 45]. It produces topic models based on the three input parameters, namely, number of topics, the hyper-parameters α and β, and the number of the iterations needed for the model to converge. α is the magnitude of the Dirichlet prior over the topic distribution of a document. This parameter is considered as a number of "pseudo-words", divided evenly between all topics that are present in every document, no matter how the other words are allocated to topics. β is per-word-weight of Dirichlet prior over topic-word distributions. The magnitude of the distribution (the sum over all words) is ascertained by number of words in the vocabulary. For identifying two, five, eight, twelve and sixty topic solutions as suggested by [46], the number of iterations considered are 200. The hyper-parameters α and β are smoothing parameters that change the distribution over the topics and words respectively; and initialising these parameters correctly can result in high quality topic distribution. The value of α has been kept as 50/T, where T is number of topics; and β has been fixed as 0.01 for all topic solutions.

In unstructured document set, where number of relevant trends is not known in advance; and it is a tedious task to identify the optimal number of topics. Coarse LDA model is generated if number of topics is insufficient, whereas excessive number of topics can result in a complex model making interpretation difficult [47]. There is no established measure to defend the optimal number of solutions, however, heuristic parameters suggested by Cao et al. [48] and Arun et al. [46] were applied to find the optimal range of topic solutions, which lies in the range from 54-66 as both approaches converge at this range as shown in Figure 4. Based on these heuristics and findings of the study [49], the optimal number of topic solutions for identifying research trends is chosen as sixty. The twelve topic solution has been considered as optimal low level solution as discussed in study by Sidorova et al. [40]. Two,

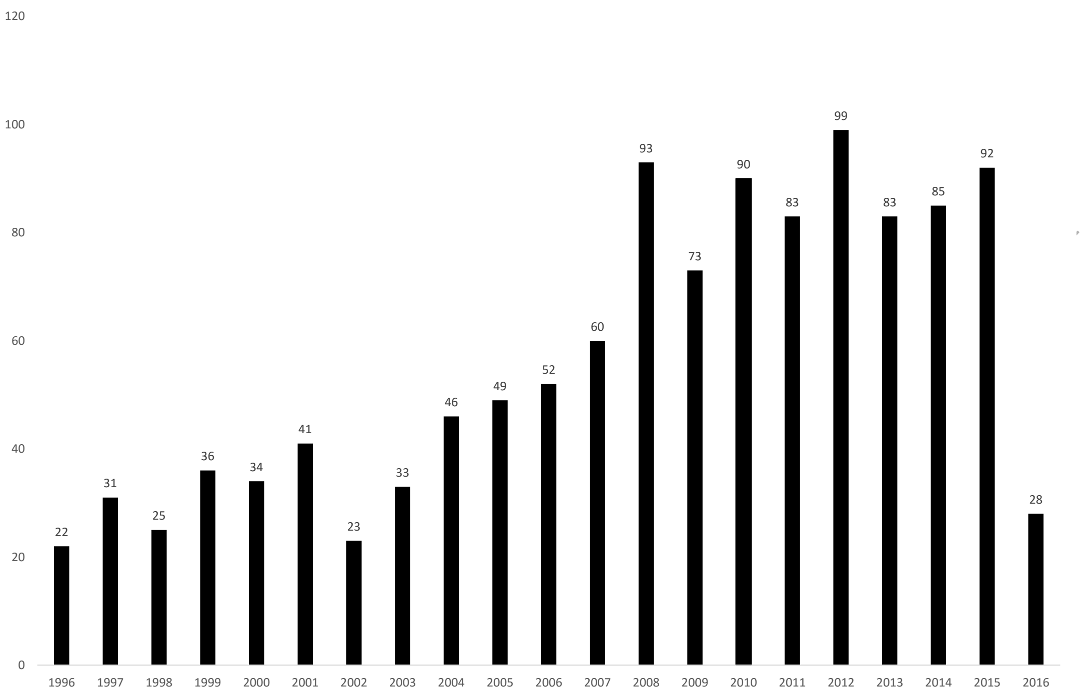


Figure 1: Paper publication year-wise

five and eight topic solutions are used to present the abstract view of large text corpus.

4.4. *Topic labelling*

For the purpose of current study, high-loading articles of all topic solutions have been reviewed. Further, the topics have been labelled individually to reach a conclusive topic label. Different topic solutions and count of high-loading articles are represented in a chronological order (Table 3 and 5).

5. Results and findings

Summary of topic solutions. The loadings for two, five, eight, twelve and sixty topics have been obtained through LDA. Table 3 summarises two, five and eight topic solutions as core research areas with corresponding labels and paper loadings. Twelve topic solution is considered to be the most appropriate for interpreting the core research areas. The number of articles loaded with each topic portrays the corresponding dominance of topic solution. Different topic solutions can correspond to the same research areas as "expert judgement" appears across (T5.5), (T8.6) and (T12.4), but there is a reduction in the number of high-loading articles.

5.1. *Core research areas*

The two topic solution presents abstract view of literature dataset and divides it into, "analysis of estimation process" (T2.1) and "estimation methods" (T2.2) as shown in Table 3. These areas cover analysis of different techniques, proposed models and methods to estimate effort.

In five topic solution, the emerged research areas are "size metrics" (T5.1), "estimation by analogy" (T5.2), "tools for estimation" (T5.3), "soft computing techniques" (T5.4) and "expert judgement" (T5.5).

The research areas have been further widened in eight topic solution with new areas emerging as

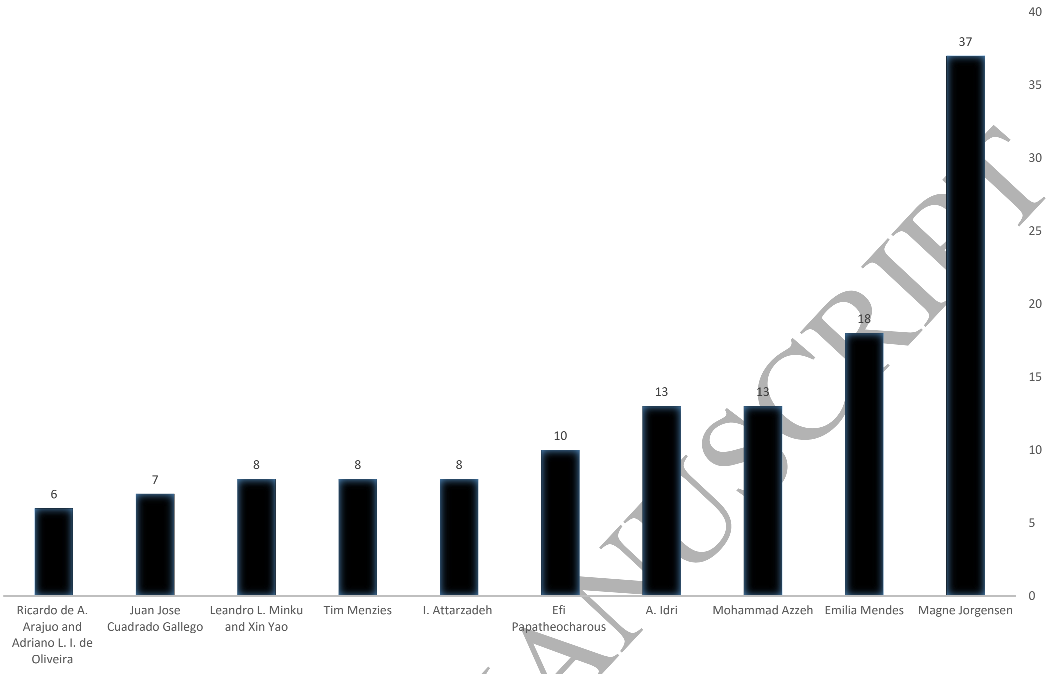


Figure 2: Top ten authors

"phase-wise effort distribution" (T8.3), "estimation for reusable components" (T8.4), "soft computing techniques" (T8.5), "neural networks" (T8.7), and "factors affecting cost" (T8.8). The remaining areas are the same as uncovered in five topic solution, including "estimation by analogy" (T8.1), "size metrics" (T8.2) and "expert judgement" (T8.6) with fewer loadings.

In twelve topic solution, more research areas appeared, namely, "application specific estimation" (T12.3), "estimation for web applications" (T12.5), "project data selection" (T12.7), "machine learning techniques" (T12.10) and "ensemble models" (T12.11). Other areas in twelve topic solution are "reviews and mapping studies" (T12.2), "expert judgement" (T12.4), "factors affecting estimation" (T12.6), "size metrics" (T12.8) and "estimation by analogy" (T12.9). High-loading terms and top five articles for twelve topic solution are depicted in Table 4. Figure 5 depicts research trends of twelve topic solution.

5.2. Research trends

The sixty topic solution resulted into detailed research trends in SEE as depicted in Table 5 with the count of published articles in chronological order. In sixty topic solution, some prominent research trends appeared, including "neural networks" (T60.31) with 60 articles and "fuzzy logic" (T60.45) with 41 articles. These relate to the "machine learning techniques" (T12.10) research area in twelve topic solution having substantial 158 high-loading papers. High loading articles include 'radial basis function neural network' [108], 'neuro-fuzzy model' [109], 'factors affecting fuzzy model' [110] and 'comparison of neural networks for SEE' [111]. "Size estimation" (T60.33), another important research trend with 40 articles, reports the articles on 'size estimation for object-oriented approach' [112], 'measurement of COSMIC' [87] and 'association of different function point methods' [113].

Another emerging trend in sixty topic solution

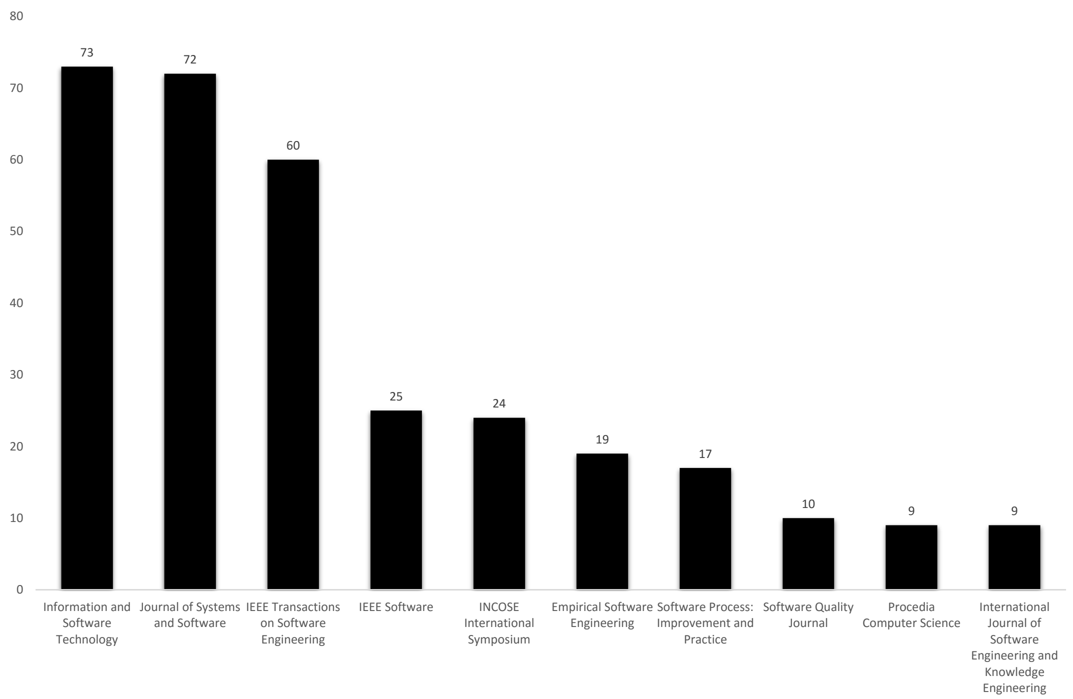


Figure 3: Top ten journals

is "ensemble models" (T60.58) which focuses on diverse approaches to estimate the effort rather than using a single approach. High loading papers focus on 'multi-objective ensemble generation for SEE' [102, 114] and 'review of ensemble models' [29].

Several other trends are also revealed across a number of articles in sixty topic solution including "missing data effects" (T60.10), "estimation by analogy" (T60.4), "factors affecting effort estimation" (T60.13), "estimation for web applications" (T60.37), "test effort estimation" (T60.42), "evaluation criteria and metrics" (T60.56), "nature inspired algorithms" (T60.38), "estimation tools" (T60.19) and "literature reviews" (T60.22).

6. Discussion

This study summarises the research trends in SEE based upon a corpus of 1178 articles. The present research includes articles from bibliographic databases from the period 1996-2016. Analysis of corpus for n topic solution has been performed by using LDA to find out the latent research patterns and trends. In SEE research, there have been only a few researchers who have long-term focus on SEE research. Figures 2 and 3 display top ten identified researchers and journals appearing respectively during the study period of 1996-2016. In this section, each research question has been discussed in view of the findings from the literature dataset and further research opportunities.

6.1. RQ1. Which research areas have been explored mostly by the researchers?

The brief answer is that various research areas have been explored by researchers since the inception of research in SEE. The focus of researchers in the SEE field changes over time as per the industry requirements. Figure 5 depicts that SEE research has gained momentum from the year 2004. The research areas can be explored by examining

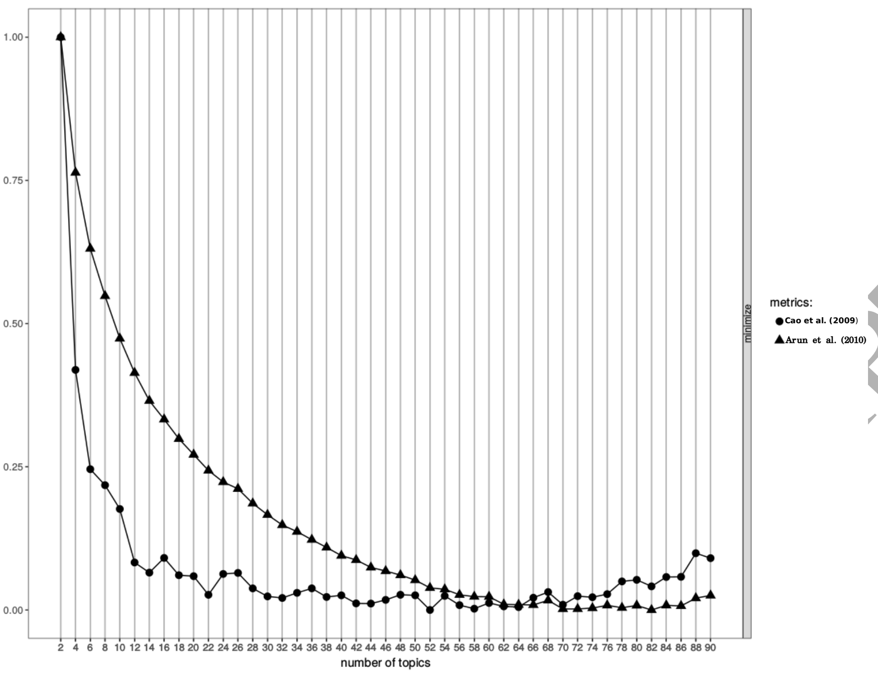


Figure 4: Selection of optimal number of topic solution

the dynamic changes in twelve core research areas and sixty research trends as identified in the last section. Further, the interesting results obtained from semantic mapping of core research areas and trends are presented in Table 7. The topics having loading value of 0.32 and above have been considered for semantic mapping [368]. But this does not validate the academic value of selected papers as loading values are the result of the number of occurrences of terms in the corpus. Also, the value of the threshold does not mean that the articles below threshold are not related to that factor.

The dynamics of twelve topic solution has brought out that "estimation for web applications" (T12.5), "size metrics" (T12.8), "ensemble models" (T12.11) and "dynamic effort estimation" (T12.12) have been considered for research more vigorously during the last two decades. The results of each research area help to find the dynamic characteristics. The results indicate that although many topics have remained stable over the time, yet "machine learning techniques" (T12.10), "expert judgement" (T12.4), and "estimation by analogy" (T12.9) have attracted the attention of more and more researchers. It has also been found that most of the research trends have shown an upward trend because the researchers have failed to reach consensus of developing and validating the generic model to predict effort for all types of projects. The researchers keep on experimenting with new methods, environments and metrics to evaluate and predict effort on the basis of review studies. Table 1 presents literature review at a glance on the sub-themes of SEE.

"Reviews and mapping studies" (T12.2) core research areas brings out "surveys" (T60.15) [55, 56, 176, 178] and "reviews" (T60.22) [21, 27, 33, 209] reporting on mapping studies and systematic literature reviews in SEE. The research trend "ensemble models" (T60.58) semantically mapped to this core area presents high-loaded paper that reports on systematic literature review on ensemble models [359]. These studies provide the road-map for naive researchers to identify the problem domains.

Since the inception of SEE research, size metrics (e.g. lines of code) has been a vital parameter for effort estimation, the researchers have proposed various new size metrics along with established size metrics and their suitability to predict in specific environment [84, 255, 354, 355]. "Size metrics" (T12.8) has uncovered two research trends "size estimation" (T60.33) and "function point analysis" (T60.57) in which research has been conducted since the year 1997 and is still an area

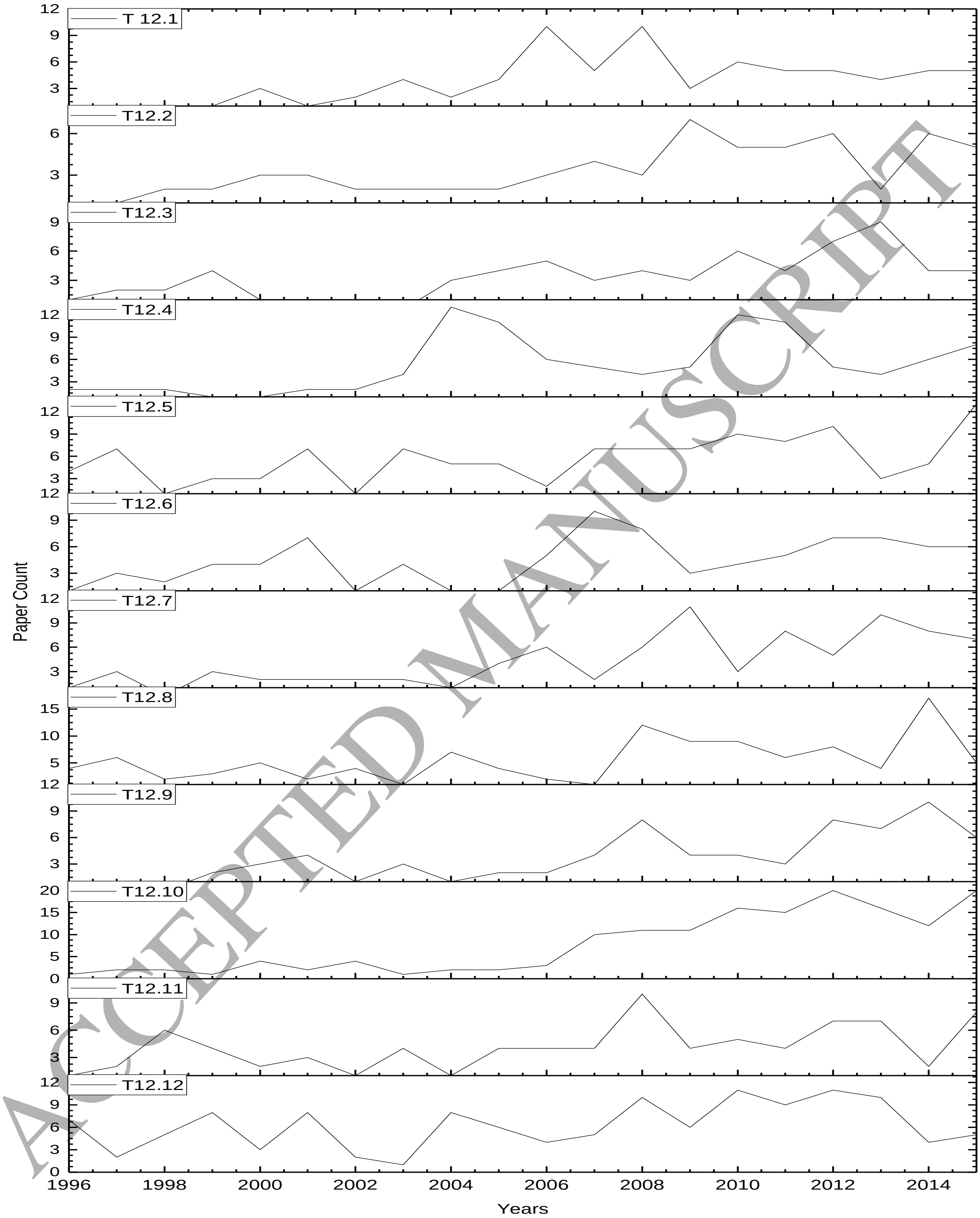


Figure 5: Research trends of twelve topic solution

Table 3: Year-wise paper count for two, five and eight topic solution

Topic id	Topic label	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	Total
T2.1	Analysis of estimation process	18	12	17	19	14	20	7	15	29	29	27	31	36	36	42	37	47	39	37	41	12	565
T2.2	Estimation methods	4	19	8	17	20	21	16	18	17	20	25	29	57	37	48	46	52	44	48	51	16	613
T5.1	Size metrics	2	5	2	6	5	5	6	6	14	3	6	5	18	10	12	12	17	6	19	14	5	178
T5.2	Estimation by analogy	1	4	2	6	7	12	4	6	1	7	10	17	26	18	18	19	24	30	23	23	7	265
T5.3	Tools for estimation	13	8	13	12	7	14	2	9	12	16	11	15	19	14	21	22	27	18	16	13	6	288
T5.4	Soft computing techniques	2	8	3	7	8	3	4	6	7	10	13	11	20	17	23	15	18	14	16	24	5	234
T5.5	Expert judgement	4	6	5	5	7	7	7	6	12	13	12	12	10	14	16	15	13	15	11	18	5	213
T8.1	Estimation by analogy	2	2	0	2	1	1	1	3	3	2	4	9	12	9	9	8	11	12	16	11	5	123
T8.2	Size metrics	4	6	4	8	8	8	3	4	9	5	2	2	15	13	12	12	12	7	16	4	2	156
T8.3	Phase-wise effort distribution	1	5	1	1	2	5	0	1	3	5	4	6	12	9	7	8	12	7	10	9	0	108
T8.4	Estimation for reusable components	9	4	7	6	0	4	2	5	5	6	8	9	9	9	12	10	9	10	11	10	6	151
T8.5	Soft computing techniques	0	3	2	2	1	2	4	1	3	3	7	1	8	9	19	14	14	7	6	17	5	128
T8.6	Expert judgement	3	6	2	5	5	5	2	12	13	10	8	6	11	8	10	10	12	8	10	11	3	160
T8.7	Neural networks	1	3	3	10	7	10	5	4	2	8	10	17	17	9	13	15	19	21	9	19	6	208
T 8.8	Factors affecting cost	2	2	6	2	10	6	6	3	8	10	9	10	9	7	8	6	10	11	7	11	1	144

of interest for researchers. The trends have been focused on the estimation of different size metrics [88, 255, 256, 257, 369], function point for SEE, and conversion of FP to COSMIC and other size metrics [84, 354, 355, 356, 357]. Different size metrics including function points and web objects have been proposed to predict effort of web applications [71, 73, 124, 370, 371, 372, 373]. The "estimation for web applications" (T12.5) core research area reveals trends, namely,"estimation for web applications" (T60.37) focusing on web objects metric and models [272, 273], "system engineering" (T60.36) focusing on estimation models in system engineering and "estimation tools" (T60.19) supporting the use of estimation tools [193, 194]. Researchers have also used machine learning techniques for estimating the effort in web applications [349, 374, 375].

The application of "machine learning techniques" (T12.10) to predict effort has gained significant momentum from the year 2006 and uncovered seven research trends. The most explored algorithmic approaches "fuzzy logic" (T60.45) [376, 377], "neural networks" (T60.31) [26, 108, 117, 378, 379, 380], genetic algorithms [277, 381, 382] have been consistently used in every aspect of software effort estimation. Other trends identified are "nature inspired algorithms" (T60.38) which focus on the use of algorithms based upon various natural phenomena [23, 247, 248, 249, 276, 277, 279]. "feature selection in problem domain" (T60.17) [101, 383], "support vector regression" (T60.55) [348, 349], and "case-based reasoning" (T60.35) [263]. "Morphological approach" (T60.12), a hybrid approach researched by de A. Araújo et al. [166], is a morphological-rank-linear approach to estimate SEE in a better way [93, 94, 166, 167] which has been revealed by "machine learning techniques" (T12.10) core research area.

Jørgensen et al. [3] advocated that formal models should be developed as support to expert judgement and proposed guidelines for "expert judgement" (T12.4) [209, 316, 330]. "Expert judgement" (T12.4) reveals the research trends including "factors influencing expert judgement" (T60.48) discussing various factors in expert judgement [64, 65, 318, 320], "strategy selection" (T60.51) reporting selection factors and techniques [28, 330]. "Fuzzy logic" (T60.45) with a focus on handling imprecision and uncertainty in expert judgement [305, 306, 307] has been emerged from this core area. Researchers explored the techniques such as pairwise-comparisons, baysian belief networks, neural networks and regression in conjunction with expert judgement [384, 385, 386, 387] for accurate predictions.

The high-loaded papers for research area "project data selection" (T12.7) focused on the selection of data for training the model using "windowing approach" (T60.60) to improve the software estimation [79, 80, 81, 82, 388]. The "data specific estimation" (T60.44) trend investigated the cross-company and within-company data to achieve more accurate results [50, 52, 389]. Different selection strategies for project selection have been reported in "estimation by analogy" (T60.4) [390, 391]. In "factors affecting estimation" (T12.6), the trend called "missing data effects" (T60.10) has

Table 4: High-loading research papers for twelve topic solution

Topic id	Key terms	Topic label	High-loading papers	Loading
T12.1	model predict base cocomo construct compani valid relationship build compar	Validation of estimation models	[50]	0.519
			[51]	0.5
			[52]	0.472
			[53]	0.458
			[54]	0.4
T12.2	research studi evalu framework exist organ type support includ analyz	Reviews and mapping studies	[55]	0.51
			[21]	0.486
			[56]	0.476
			[57]	0.421
			[58]	0.41
T12.3	approach base process complex practic integr decis provid develop resourc	Application specific estimation	[59]	0.509
			[60]	0.507
			[61]	0.367
			[62]	0.36
			[63]	0.357
T12.4	estim fuzzi experi task expert inform process assess accur uncertainti	Expert judgement	[64]	0.75
			[65]	0.634
			[66]	0.592
			[67]	0.588
			[68]	0.566
T12.5	system applic engin develop manag web methodologi appli risk reliabl	Estimation for web applications	[69]	0.568
			[70]	0.546
			[71]	0.533
			[72]	0.515
			[73]	0.5
T12.6	data project set factor analysi effect investig domain collect level	Factors affecting estimation	[74]	0.667
			[75]	0.609
			[76]	0.522
			[77]	0.495
			[78]	0.489
T12.7	project accuraci bas improv studi plan weight manag analogy combin	Project data selection	[79]	0.768
			[80]	0.717
			[81]	0.716
			[82]	0.64
			[83]	0.598
T12.8	function measur size requir tool propos analysi specif standard compon	Size metrics	[84]	0.634
			[85]	0.612
			[86]	0.607
			[87]	0.603
			[88]	0.577
T12.9	method techniqu propos select obtain analogi dataset appli attribut produc	Estimation by analogy	[89]	0.529
			[30]	0.518
			[90]	0.435
			[91]	0.391
			[92]	0.385
T12.10	regress algorithm network perform neural featur learn optim compar propos	Machine learning techniques	[93]	0.667
			[94]	0.627
			[95]	0.618
			[96]	0.568
			[97]	0.566
T12.11	test approach perform error object gener effect statist empir relat	Ensemble models	[98]	0.519
			[99]	0.485
			[100]	0.449
			[101]	0.44
			[102]	0.435
T12.12	product time cost metric process phase qualiti activ mainten requir	Dynamic effort estimation	[103]	0.554
			[104]	0.456
			[105]	0.442
			[106]	0.441
			[107]	0.434

Table 5: Year-wise paper count for sixty topic solution

Topic id	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	Total
T60.1	-	1	-	-	2	-	-	-	-	-	1	-	-	-	-	1	1	1	1	-	1	9
T60.2	-	2	-	1	1	1	1	1	1	-	-	1	2	1	1	-	1	-	1	-	-	15
T60.3	-	-	-	1	-	1	-	1	1	-	-	-	1	-	-	-	-	-	-	1	-	6
T60.4	-	-	-	1	-	-	-	1	1	-	1	-	5	2	-	1	1	1	1	2	-	17
T60.5	-	-	-	1	-	-	-	1	1	-	1	-	5	2	-	1	1	1	1	2	-	17
T60.6	-	-	-	1	1	1	-	2	-	-	-	1	-	-	2	-	-	1	-	-	-	9
760.7	2	-	1	-	-	-	-	-	-	-	1	3	1	1	-	2	1	1	-	1	1	15
T60.8	-	-	-	-	-	-	-	-	-	-	-	1	2	1	1	1	3	-	1	1	1	12
T60.9	-	-	-	-	-	1	-	1	-	-	1	-	1	1	2	-	2	1	-	-	-	10
T60.10	-	-	-	1	-	4	-	-	-	-	2	4	1	1	-	-	-	3	1	-	2	19
T60.11	1	1	1	1	2	3	1	-	1	-	1	1	-	1	-	3	2	1	3	1	-	24
T60.12	-	1	-	-	-	-	-	-	-	-	-	-	1	1	3	2	3	-	1	-	-	12
T60.13	-	-	-	-	-	-	-	-	-	-	-	-	1	-	3	1	2	2	1	3	-	13
T60.14	-	-	1	-	1	-	-	-	-	-	1	1	-	1	3	-	2	2	1	1	1	15
T60.15	-	-	-	-	-	1	-	-	-	2	-	-	-	1	-	1	2	1	2	2	-	12
T60.16	-	2	-	1	-	-	-	1	-	2	1	4	1	2	-	-	1	1	-	3	1	20
T60.17	-	-	-	-	-	-	-	1	-	-	-	1	1	1	1	2	1	-	1	-	1	10
T60.18	-	-	-	1	1	-	-	2	1	-	-	1	1	3	2	-	-	2	-	1	-	15
T60.19	1	-	1	1	1	-	-	-	1	-	1	1	1	1	3	3	3	-	1	1	1	21
T60.20	-	1	-	1	-	-	-	-	-	1	1	-	1	2	3	-	1	-	1	1	-	13
T60.21	-	1	-	-	-	-	-	1	-	1	1	-	1	-	-	2	-	-	2	1	-	10
T60.22	-	1	-	-	-	-	-	2	2	3	3	1	-	-	1	1	1	-	1	2	-	18
T60.23	-	-	1	2	-	1	-	1	-	2	-	1	-	-	2	1	1	1	-	-	1	14
T60.24	-	-	-	2	3	2	-	1	1	-	1	1	-	2	1	1	1	1	-	2	-	19
T60.25	-	-	-	-	-	2	1	-	-	-	-	-	-	1	-	3	4	1	3	2	1	18
T60.26	1	1	3	3	3	3	-	-	1	3	2	-	3	2	4	1	1	1	2	-	-	34
T60.27	-	-	1	1	2	2	3	1	3	2	-	1	-	2	-	1	4	2	1	2	1	29
T60.28	-	-	-	2	-	-	-	-	1	1	1	1	3	-	-	-	2	-	-	-	1	12
T60.29	-	-	-	-	-	-	-	-	1	-	-	-	-	-	2	-	1	1	2	1	-	8
T60.30	1	-	-	-	-	-	1	-	-	-	2	-	-	-	1	-	4	-	2	1	2	14
T60.31	1	2	1	1	1	1	2	-	-	1	1	4	5	4	8	4	9	5	5	5	-	60
T60.32	-	1	1	1	-	-	-	1	-	2	-	1	3	-	1	2	-	1	-	-	1	15
T60.33	-	2	1	-	2	2	2	-	2	1	-	-	4	4	5	3	2	2	6	2	-	40
T60.34	-	-	1	-	-	-	-	-	1	-	-	-	-	1	-	-	1	2	2	1	-	9
T60.35	-	-	1	2	1	2	1	-	-	3	1	1	4	-	2	-	1	1	-	-	-	20
T60.36	3	-	2	-	-	2	-	1	1	1	2	4	1	3	2	3	2	2	1	3	1	34
T60.37	1	-	1	-	3	1	1	6	2	1	-	-	3	1	1	-	2	1	1	3	-	28
T60.38	-	-	-	1	-	-	-	-	-	-	-	-	2	-	5	5	2	2	4	7	-	28
T60.39	-	2	-	-	-	1	1	-	-	1	1	1	1	2	2	1	1	4	4	1	-	23
T60.40	-	-	-	-	-	-	-	1	-	-	3	2	3	3	1	1	1	1	1	-	-	17
T60.41	1	-	1	-	-	2	-	1	-	2	3	2	1	4	-	3	1	1	3	2	-	27
T60.42	1	1	-	-	1	-	-	-	3	2	2	1	4	-	1	-	3	1	-	2	1	23
T60.43	-	-	-	-	-	-	1	-	2	2	-	-	2	-	1	-	1	1	-	2	-	12
T60.44	-	-	-	1	3	1	1	1	1	1	-	2	1	-	1	2	-	-	1	1	1	18
T60.45	-	2	1	-	-	-	1	1	3	2	4	1	2	3	5	7	1	1	3	4	-	41
T60.46	-	-	-	-	-	-	1	-	-	3	-	3	2	2	1	3	3	4	2	3	-	27
T60.47	2	-	-	1	-	-	-	2	-	1	-	1	1	1	1	2	3	3	3	1	-	22
T60.48	-	1	1	-	1	2	-	4	3	2	-	-	-	4	-	3	1	2	1	1	-	26
T60.49	-	1	-	-	-	-	1	-	-	-	1	3	1	4	3	1	1	5	4	-	-	25
T60.50	1	1	1	2	1	-	-	-	-	-	1	-	3	-	-	-	-	-	2	2	-	14
T60.51	-	-	1	-	1	-	-	-	1	1	-	-	-	-	5	2	1	1	-	-	-	13
T60.52	-	1	-	-	1	1	-	-	-	-	2	1	2	2	3	1	1	2	2	1	-	20
T60.53	2	1	-	-	-	-	-	-	3	1	-	1	1	1	1	-	3	-	-	-	1	15
T60.54	-	-	-	1	-	1	-	-	-	1	-	-	4	-	2	2	2	1	-	4	1	19
T60.55	-	-	-	-	1	-	-	-	-	-	1	2	2	2	1	2	2	4	-	5	-	22
T60.56	-	1	1	-	-	1	1	1	1	-	1	2	2	1	-	3	1	1	2	6	-	25
T60.57	1	2	-	3	1	1	1	-	1	1	-	-	3	2	1	2	5	1	5	-	1	31
T60.58	-	1	1	-	1	2	-	-	1	-	1	1	2	-	1	1	2	5	2	2	3	26
T60.59	1	-	-	1	-	-	-	-	1	1	3	3	6	-	1	1	-	1	1	-	1	21
T60.60	-	-	-	-	-	-	-	-	3	-	-	-	-	1	-	2	2	2	2	3	2	17

Table 6: Five high-loading papers for sixty topic solution

Topic id	Topic label	High-loading papers
T60.1	Early estimation	[115, 116, 117, 118, 119]
T60.2	Dynamic estimation	[120, 121, 72, 122, 123]
T60.3	Object oriented components estimation	[100, 124, 125, 126, 127]
T60.4	Estimation by analogy	[128, 129, 130, 131, 132]
T60.5	Productivity measurement	[133, 134, 135, 136, 137]
T60.6	Statistical filtering	[138, 139, 140, 141, 142]
T60.7	Project scheduling	[107, 143, 144, 145, 146]
T60.8	SRS based estimation	[147, 148, 149, 150, 151]
T60.9	Cost uncertainty	[152, 153, 154, 155, 156]
T60.10	Missing data effects	[157, 158, 159, 160, 161]
T60.11	Cost estimation models	[103, 162, 163, 164, 165]
T60.12	Morphological approach	[166, 167, 95, 94, 93]
T60.13	Factors affecting estimation	[168, 78, 169, 170, 171]
T60.14	Quality management	[172, 60, 173, 174, 175]
T60.15	Surveys	[176, 55, 177, 56, 178]
T60.16	Risk assessment	[179, 180, 181, 182, 183]
T60.17	Feature selection in problem domain	[184, 154, 185, 186, 187]
T60.18	Prediction interval approach	[188, 189, 190, 191, 192]
T60.19	Estimation tools	[193, 194, 195, 196, 197]
T60.20	Use case-based estimation	[198, 199, 200, 201, 202]
T60.21	Grey relation analysis	[203, 204, 205, 206, 207]
T60.22	Literature reviews	[208, 33, 209, 27, 28]
T60.23	Change impact analysis	[210, 211, 212, 213, 214]
T60.24	Software evolution	[106, 215, 216, 217, 218]
T60.25	Fuzzy analogy	[89, 30, 219, 220, 221]
T60.26	Object-oriented metrics	[222, 223, 224, 225, 226]
T60.27	Maintenance effort	[227, 228, 229, 230, 231]
T60.28	Country specific surveys	[232, 233, 234, 235, 236]
T60.29	Work breakdown structures	[237, 238, 239, 240, 241]
T60.30	Requirement elicitation impact	[242, 243, 244, 245, 246]
T60.31	Neural networks	[108, 247, 248, 249, 23]
T60.32	Reliability modelling	[250, 251, 252, 253, 254]
T60.33	Size estimation	[255, 256, 257, 87, 258]
T60.34	Estimation in cloud computing	[259, 260, 59, 261, 262]
T60.35	Case based reasoning	[263, 264, 265, 266, 267]
T60.36	System engineering	[268, 69, 269, 270, 271]
T60.37	Estimation for web applications	[272, 73, 273, 274, 275]
T60.38	Nature inspired algorithms	[276, 277, 278, 279, 280]
T60.39	Estimation for reusable components	[281, 282, 283, 284, 285]
T60.40	Parametric models	[286, 287, 288, 289, 290]
T60.41	Estimation in agile projects	[83, 291, 292, 293, 76]
T60.42	Test effort estimation	[294, 295, 296, 74, 297]
T60.43	Simulation techniques	[298, 299, 300, 301, 302]
T60.44	Data specific estimation	[50, 54, 303, 52, 304]
T60.45	Fuzzy logic	[305, 306, 307, 109, 308]
T60.46	Small datasets	[101, 309, 310, 311, 312]
T60.47	Evidence based judgement	[313, 314, 315, 316, 317]
T60.48	Factors influencing expert judgement	[64, 318, 65, 319, 320]
T60.49	Similarity measurement methods	[91, 321, 322, 323, 324]
T60.50	Empirical studies	[325, 326, 327, 328, 329]
T60.51	Strategy selection	[330, 331, 332, 333, 334]
T60.52	Global software development	[335, 62, 34, 336, 337]
T60.53	Software process improvement	[338, 339, 340, 341, 342]
T60.54	Regression techniques	[343, 344, 345, 346, 347]
T60.55	Support vector regression	[348, 349, 350, 96, 97]
T60.56	Evaluation criteria and metrics	[51, 53, 351, 352, 353]
T60.57	Function point analysis	[84, 354, 355, 356, 357]
T60.58	Ensemble models	[358, 359, 114, 360, 361]
T60.59	Impact of CMM levels	[362, 75, 363, 364, 365]
T60.60	Windowing approach	[81, 79, 80, 366, 367]

Table 7: Semantic mapping between twelve topic and sixty topic solution

Topic id	12 topic labels	Topic id	60 topic labels
T12.1	Validation of estimation models	T60.56	Evaluation criteria and metrics
		T60.40	Parametric models
		T60.44	Data specific estimation
		T60.39	Estimation for reusable components
		T60.54	Regression techniques
		T60.45	Fuzzy logic
T12.2	Reviews and mapping studies	T60.15	Surveys
		T60.22	Literature reviews
		T60.39	Estimation for reusable components
		T60.58	Ensemble models
T12.3	Application specific estimation	T60.34	Estimation in cloud computing
		T60.22	Literature reviews
		T60.47	Evidence based judgement
		T60.36	System engineering
T12.4	Expert judgement	T60.48	Factors influencing expert judgement
		T60.45	Fuzzy logic
		T60.60	Windowing approach
		T60.51	Strategy selection
		T60.47	Evidence based judgement
T12.5	Estimation for web applications	T60.37	Estimation for web applications
		T60.19	Estimation tools
		T60.36	System engineering
T12.6	Factors affecting estimation	T60.10	Missing data effects
		T60.59	Impact of CMM levels
		T60.54	Regression techniques
		T60.42	Test effort estimation
T12.7	Project data selection	T60.60	Windowing approach
		T60.44	Data specific estimation
		T60.41	Estimation in agile projects
		T60.4	Estimation by analogy
T12.8	Size metrics	T60.57	Function point analysis
		T60.33	Size estimation
T12.9	Estimation by analogy	T60.25	Fuzzy analogy
		T60.2	Dynamic estimation
		T60.10	Missing data effects
		T60.49	Similarity measurement methods
T12.10	Machine learning techniques	T60.12	Morphological approach
		T60.38	Nature inspired algorithms
		T60.31	Neural networks
		T60.35	Case based reasoning
		T60.17	Feature selection in problem domain
		T60.55	Support vector regression
		T60.45	Fuzzy logic
T12.11	Ensemble models	T60.58	Ensemble models
		T60.42	Test effort estimation
		T60.46	Small datasets
		T60.16	Risk assessment
T12.12	Dynamic effort estimation	T60.11	Cost estimation models
		T60.24	Software evolution
		T60.26	Object-oriented metrics
		T60.13	Factors affecting estimation
		T60.5	Productivity measurement
		T60.7	Project scheduling
		T60.27	Maintenance effort

emerged to depict handling and assessing of missing data [157, 158, 160, 161, 392]. "Impact of CMM levels" (T60.59) trend focuses on the effect of process maturity on SEE [75, 362, 364].

"Estimation by analogy" (T12.9) generates an effort estimate for software project based upon the data of similar past projects. There has been tremendous research in this area and it has uncovered "fuzzy analogy" (T60.25) trend that focuses on combining analogy and fuzzy logic [89, 219, 221]. The trend "dynamic estimation" (T60.2) emerged from this area reports on SEE models in dynamic environment [120, 123]. Various models have been developed by combining "estimation by analogy" (T60.4) with other techniques [201, 219, 220, 221, 310, 312, 359, 393, 394]. Various techniques have been suggested to improve the estimates by fuzzy analogy-based estimation including linguistic variables, learning adaptation, and k-modes algorithm [91, 221, 395]. The success and performance of all these techniques depends on availability of environment specific historic data. The research trend called "missing data effects" (T60.10) depicts the techniques for handling missing data in estimation by analogy [359, 396]. The "similarity measurement methods" (T60.49) provides leads on techniques to identify the similarity in different projects by way of using analogy [91, 321, 397].

In the field "application specific estimation" (T12.3), various trends have emerged after the year 2013. One such trend is called "estimation in cloud computing" (T60.34) which provides the model for estimation of cost in cloud environment [59, 60, 259, 260]. The other trends include "evidence based judgement" (T60.47) [314] and "system engineering" (T60.36) [268, 269, 270]. The "evidence based judgement" (T60.47) trend provides the model to estimate effort in web service composition-based projects estimation [314], while "system engineering" trend offers models to estimate system engineering cost [268, 269, 270] and heuristics for system engineering [398].

In "dynamic effort estimation" (T12.12), the trends identified are "cost estimation models" (T60.11), "software evolution" (T60.24), "object-oriented metrics" (T60.26) [222, 223], and "maintenance effort" (T60.27). The trend called "cost estimation models" (T60.11) helps in finding the cost of various components in SEE [103, 104, 163, 164]; the "software evolution" (T60.24) focuses on certain trends [106, 215, 399], while the "maintenance effort" (T60.27) trends provides the models for effort estimates in maintenance phase [227, 228, 229, 230].

The research carried out in the field of SEE, has proved that no single method performs consistently in all environments. So, it is advisable to generate estimates from ensembles of estimation methods. Multi-objective "ensemble models" (T12.11) have been the area of recent years [114, 361, 400]; and it consistently out-performs other techniques in various scenarios [358]. The core research area "ensemble models" (T12.11) has provided research trend "ensemble models" (T60.58) which focuses on ensemble generation [114, 309, 358]. "Object-oriented metrics" (T60.26) discuss metrics in object-oriented domain [222, 224, 226].

The core research area "validation of estimation models" (T12.1) has provided "evaluation criteria and metrics" (T60.56) which describe the metrics used for assessing the performance of estimation models and their modification [51] and comparative study of MMRE, MSE, Pred [351, 352, 353]. "Parametric models" (T60.40) having a focus on segmented models [286, 287, 289], "data specific estimation" (T60.44) focusing on cross-company, and within-company [50, 52, 31, 401], "estimation for reusable components" (T60.39) reporting on reuse framework [281, 283] have emerged from this research area. Certain others research trends have also been uncovered such as "regression techniques" (T60.54) which focus on calibration of models [344, 347] and "fuzzy logic" (T60.45) reporting on models based on fuzzy logic [26, 308, 402].

6.2. RQ2 What research methods have been used for SEE?

On the basis of analysis of high-loaded articles, it has been observed that diverse estimation approaches have been employed in SEE. The methods can be grouped into three groups, namely, formal, expert-based and hybrid. However, a few leading methods have been discovered though this study. Formal methods require large data sets for developing and validating, but the data set is generally sparse, incomplete, and inconsistent. Parametric approaches such as "case-based reasoning" [32, 185, 263, 264, 403, 404] and "support vector regression" [96, 349, 350, 405] assume data to be complete which is quite rare in the software domain. Non-parametric methods such as "neural networks" [108, 375, 248, 249, 378, 379, 406, 407], "genetic algorithms" [196, 277, 382, 408, 409], "fuzzy logic"

[91, 109, 110, 286, 305, 306, 378, 406, 410]. "Windowing approach" uses a window of recent projects as training data for building an effort estimation model [79, 80, 81, 83, 388]. Expert-based estimation is not based upon any project but it has limitation of being biased, time-consuming and dependent on estimator's experience. The methods such as "expert judgement" [28, 64, 313, 320, 411, 412], and "work breakdown structures" [237, 238] use the concept of expert-based estimation. Since no method performs consistently in all the environments, hybrid methods have been the area of concern. "Ensemble models" [102, 114, 201, 358, 359, 413, 414], "SRS based estimation" [148, 151] and "use case-based estimation" [415, 416] have been proposed for the purpose.

6.3. RQ3. Which research areas demand greater attention of researchers?

Although research in the field of SEE is being conducted for the last four decades, yet a complete study on effort estimation remains to be seen. The researchers have been unable to fully understand the requirements of the software industry in developing and validating the generic model to predict effort for all type of projects. The limitations which exist provide several opportunities for further research. In particular, Minku and Yao [361] realised that SEE is a multi-objective problem; and more attention is required to validate different models. It was observed that few longitudinal studies have been reported. To understand all the consequences of different aspects of SEE research, there is need to have more in-depth case studies. We can find the research areas which demand more attention through the interpretation of semantic mapping of twelve and sixty topic solution and loading of articles to topic solutions. The topics discussed below may form the basis for further research on SEE.

"Size metrics" (T12.8). SEE process begins with the prediction of the size of deliverables to be developed in a software. Many claim that the established size metrics are sufficient for measuring the complexities involved in SEE [354]. However, the study conducted by de Freitas Junior et al. [417] is one of the few studies with a longitudinal view emphasising on the need to improve size metrics. Thus, we need to focus on the following:

- Improvements in existing size metrics and their effectiveness in different environments.
- Standardization of conversion between different size metrics.

"Factors affecting estimation" (T12.6). Effort estimation is often influenced by several factors, both intrinsic and extrinsic. Very few studies have identified the factors which affect effort estimation [169, 418]. One of the biggest challenge is data imputation; and research area is available for improvements due to issues related to the application of existing methods [35, 160]. Further, these studies are not validated for sparse and high missing data imputation. This demands more rigorous research on the following:

- Longitudinal studies to identify the social, geographical and temporal factors which affect estimation.
- Developing an effective framework for handling data imputation based on the characteristics of effort data.

"Machine learning techniques" (T12.10). The study conducted by Wen et al. [27] provided an in-depth investigation of empirical evidences on machine learning models used in SEE. Machine learning models have been promising and most widely applied than any other approach in SEE. But the application of these machine learning models in the industry is still limited. Hence, more intensive research is required to identify the barriers in the adoption of machine learning models in the industry. Further, increased usage of performance metrics other than conventional metrics, application of machine learning in SEE projects, and increased collaboration between machine learning researchers, will create trust and promote machine learning usage in industry. Thus, there is a potential for further research on:

- Which of machine learning approach is suitable for specific environment based on characteristics of available data?
- Application of optimization techniques to tune parameters used in various machine learning approaches.
- Design and development of machine learning framework for solving multi-criteria effort estimation problems.

"Expert judgement" (T12.4). Jørgensen [28] advocated that expert judgement is leading estimation method adopted by organisations. To enable the organizations to get benefited from expert judgement, they must identify the human factors affecting the expert judgement [419] and apply practical guidelines for producing better estimates [316]. Jørgensen and Gruschke [65] concluded that it cannot be defined whether expert judgement is better or weaker than formal models. But practitioners need to identify the situations, when to use expert estimation and when to use formal models. Shepperd and Cartwright [420] tried to conceptualise framework of integration of computational approach and expert judgement, but lacked validation and generalization. This area is also open to research as the researchers have yet to focus on enhancing expert judgement in conjunction with computational models. The research community should pay more attention to the questions concerning:

- Longitudinal studies on identifying the motivational factors involved in intentional and unintentional distortions.
- Artificial intelligence based framework for integrating expert judgement with computational models.
- Elaborative studies on identification of steps to reduce the biasing and uncertainty.

"Ensemble models" (T12.11). The weakness of existing estimation models gave rise to ensemble models in SEE. A systematic review of ensemble estimation methods [29] explored the use of machine learning approaches for ensemble model generation and suggested to further investigate existing established approaches. Moreover, ensemble models have not been popular among practitioners in the real world. So research community is encouraged to conduct research on :

- Empirical studies to identify the suitable candidates for ensemble generation
- Design methodology to evaluate the performance of ensemble models based on different performance measures.

"Validation of estimation models" (T12.1). Findings of current study reveal that a large number of proposed estimation models lack validation and generalization of the results. Very few studies have reported on issues pertaining to model validation [50, 347, 421]. As suggested by Kocaguneli and Menzies [179], performance of a model can be assessed by dataset used and model evaluation criterion. Mean magnitude of relative error (MMRE) as evaluation criterion has been empirically studied and criticized by [53, 422], and suffers from rank reversal. Thus elaborative research can be conducted on :

- Comprehensive study of different performance measures to assess the accuracy of estimation models.

Other areas. Besides research topics discussed above, some recent areas have emerged from the study including "morphological approach", "use case-based estimation", "feature selection", "windowing approach", "grey relation analysis" but high loading articles count for these areas is quite low. These areas are potential topics for further exploration. Designing of a formal design methodology and validation of "morphological approach" on large datasets is still an open area in morphological approach. Future work in "windowing approach" should focus on optimum window size and steepness of weighted function. Since these areas have not been widely investigated, research community should focus on exploring these areas to find the challenges ahead.

7. Threats to validity

Threats to the validity of our study and their corresponding mitigation strategies have been described and detailed as follows:

Selection of the search string. This threat refers to the effectiveness of the applied search string to find a sufficiently large number of relevant articles. Limitation of selecting search terms and their synonyms, search string design, and search engines may result in incomplete literature dataset. Therefore, due to our choice of keywords and search string, there is a risk that relevant articles may be omitted. In this study, to ensure the relevancy of collected articles in the literature dataset to SEE, they were reviewed thoroughly by two step review process.

Subjectivity in topic labelling. The labelling of topics is a great concern due to the subjectivity and biasing involved in it. To overcome this limitation, two authors of the paper did topic labelling individually and later on it was combined to generate conclusive label.

8. Conclusion

This research is based on mathematical foundations and discovers the research trends in SEE literature. It uncovers the research trends by analysing 1178 documents published by the researchers. The approach generated n-topic solution, and corresponding term and document loadings. These loadings explain the proximity to a given topic. Only highly-loaded terms and documents above the threshold were considered relevant to the topic.

Researchers can understand not only the trends prevailing in SEE research, but the potential research areas can also be identified. Moreover, researchers can analyse any one core research area out of twelve research areas identified in the current study for further exploration. Further, references have been provided to a large number of success stories which not only educate practitioners, but also make them feel more confident about adopting SEE. Although some limitations have been identified during research on SEE in organisations, yet there is enough scope for future research. The results of current study can enable the researchers to face new research challenges and align their work with contemporary research. They may also use other topic modelling techniques to identify the hidden patterns and trends from a large bibliographic dataset.

References

[1] M. V. Zelkowitz, R. T. Yeh, R. G. Hamlet, J. D. Gannon, V. R. Basili, Software engineering practices in the us and japan., Computer 17 (6) (1984) 57–70. doi:10.1109/MC.1984.1659162.

[2] R. L. Glass, I. Vessey, V. Ramesh, Research in software engineering: an analysis of the literature, Information and Software technology 44 (8) (2002) 491–506. doi:10.1016/S0950-5849(02)00049-6.

[3] M. Jorgensen, B. Boehm, S. Rifkin, Software Development Effort Estimation: Formal Models or Expert Judgment?, IEEE Software 26 (2) (2009) 14–19. doi:10.1109/MS.2009.47.

[4] M. Shaw, What makes good research in software engineering?, International Journal on Software Tools for Technology Transfer 4 (1) (2002) 1–7. doi:10.1002/0471028959.sof550.

[5] A. Trendowicz, J. Münch, R. Jeffery, Software Engineering Techniques, Springer, 2008, Ch. State of the practice in software effort estimation: a survey and literature review, pp. 232–245. doi:10.1007/978-3-642-22386-0_18.

[6] M. Yalcinkaya, V. Singh, Patterns and trends in building information modeling (bim) research: A latent semantic analysis, Automation in Construction 59 (2015) 68 – 80. doi:10.1016/j.autcon.2015.07.012.

[7] J. C. Campbell, A. Hindle, E. Stroulia, The Art and Science of Analyzing Software Data, Morgan Kaufmann, 2015, Ch. Latent Dirichlet Allocation: Extracting Topics from Software Engineering Data, pp. 139 – 159. doi:10.1016/B978-0-12-411519-4.00006-9.

[8] K. R. Canini, L. Shi, T. L. Griffiths, Online inference of topics with latent dirichlet allocation., Journal of Machine Learning Research 5 (2009) 65–72. URL http://jmlr.org/proceedings/papers/v5/canini09a/canini09a.pdf

[9] S. Saini, B. Kasliwal, S. Bhatia, Language identification using g-lda, International Journal of Research in Engineering and Technology 2 (11) (2013) 42–45. URL https://issuu.com/ijret/docs/language_identification_using_g-lda

[10] N. Evangelopoulos, X. Zhang, V. R. Prybutok, Latent semantic analysis: five methodological recommendations, European Journal of Information Systems 21 (1) (2012) 70–86. doi:10.1057/ejis.2010.61.

[11] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, R. Harshman, Indexing by latent semantic analysis, Journal of the American society for information science 41 (6) (1990) 391–407. doi:10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.

[12] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, Journal of Machine Learning Research 3 (2003) 993–1022. URL http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf

[13] S. W. Thomas, Mining software repositories using topic models, in: 33rd International Conference on Software Engineering, Waikiki, USA, 2011, pp. 1138–1139. doi:10.1145/1985793.1986020.

[14] S. K. Lukins, N. A. Kraft, L. H. Etzkorn, Source code retrieval for bug localization using latent dirichlet allocation, in: 15th Working Conference on Reverse Engineering, IEEE, Antwerp, Belgium, 2008, pp. 155–164. doi:10.1109/WCRE.2008.33.

[15] B. Clark, D. Zubrow, How good is the software: a review of defect prediction techniques, in: Software Engineering Symposium, Pittsburgh,Pennsylvania, 2001. URL http://www.sei.cmu.edu/library/assets/defect-prediction-techniques.pdf

[16] K. Tian, M. Revelle, D. Poshyvanyk, Using latent dirichlet allocation for automatic categorization of software, in: 6th IEEE International Working Conference on Mining Software Repositories, IEEE, Vancouver, Canada, 2009, pp. 163–166. doi:10.1109/MSR.2009.5069496.

[17] Y. Fu, M. Yan, X. Zhang, L. Xu, D. Yang, J. D. Kymer, Automated classification of software change messages by semi-supervised latent dirichlet allocation, Information and Software Technology 57 (2015) 369–377. doi:10.1016/j.infsof.2014.05.017.

[18] S. Banitaan, M. Alenezi, Software evolution via topic modeling: An analytic study, International Journal of Software Engineering and Its Applications 9 (5) (2015)

43–52. doi:10.14257/ijseia.2015.9.5.05.

[19] B. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering, Tech. rep., Keele University (2007). URL https://www.cs.auckland.ac.nz/~mria007/Sulayman/Systematic_reviews_5_8.pdf

[20] K. Petersen, S. Vakkalanka, L. Kuzniarz, Guidelines for conducting systematic mapping studies in software engineering: An update, Information and Software Technology 64 (2015) 1–18. doi:10.1016/j.infsof.2015.03.007.

[21] M. Jørgensen, M. J. Shepperd, A systematic review of software development cost estimation studies, IEEE Transactions on Software Engineering 33 (1) (2007) 33–53. doi:10.1109/TSE.2007.256943.

[22] H. Rastogi, S. Dhankhar, M. Kakkar, A survey on software effort estimation techniques, in: 5th International Conference - The Next Generation Information Technology Summit (Confluence), Noida, India, 2014, pp. 826–830. doi:10.1109/CONFLUENCE.2014.6949367.

[23] P. Reddy, K. Sudha, P. R. Sree, S. Ramesh, Software effort estimation using radial basis and generalized regression neural networks, Journal of Computing 2 (5) (2010) 87–92. URL https://arxiv.org/vc/arxiv/papers/1005/1005.4021v1.pdf

[24] K. Moløkken-Østvold, M. Jørgensen, S. S. Tanilkan, H. Gallis, A. C. Lien, S. W. Hove, A survey on software estimation in the norwegian industry, in: 10th International Symposium on Software Metrics, Chicago, USA, 2004, pp. 208–219. doi:10.1109/METRIC.2004.1357904.

[25] K. O. Simmons, R. Korrapati, A study to examine factors that contribute software project success, in: Allied Academies International Conference, Vol. 8, Jordan Whitney Enterprises, Inc, Maui, USA, 2004, pp. 47–50.

[26] I. Attarzadeh, S. H. Ow, Software development effort estimation based on a new fuzzy logic model, International Journal of Computer Theory and Engineering 1 (4) (2009) 473. doi:10.7763/ijcte.2009.v1.77.

[27] J. Wen, S. Li, Z. Lin, Y. Hu, C. Huang, Systematic literature review of machine learning based software development effort estimation models, Information and Software Technology 54 (1) (2012) 41–59. doi:10.1016/j.infsof.2011.09.002.

[28] M. Jørgensen, A review of studies on expert estimation of software development effort, Journal of Systems and Software 70 (1) (2004) 37–60. doi:10.1016/S0164-1212(02)00156-5.

[29] A. Idri, M. Hosni, A. Abran, Systematic literature review of ensemble effort estimation, Journal of Systems and Software 118 (2016) 151 – 175. doi:10.1016/j.jss.2016.05.016.

[30] A. Idri, F. A. Amazal, A. Abran, Analogy-based software development effort estimation: A systematic mapping and review, Journal of Information and Software Technology 58 (2015) 206–230. doi:10.1016/j.infsof.2014.07.013.

[31] B. Kitchenham, E. Mendes, G. H. Travassos, A systematic review of cross-vs. within-company cost estimation studies, in: 10th international conference on Evaluation and Assessment in Software Engineering, British Computer Society, Swinton, UK,, 2006, pp. 81–90. URL http://www.bcs.org/upload/pdf/ewic_ea06_paper10.pdf

[32] B. Sigweni, M. Shepperd, Feature weighting techniques for cbr in software effort estimation studies: a review and empirical evaluation, in: 10th International Conference on Predictive Models in Software Engineering, ACM, Torino, Italy, 2014, pp. 32–41. doi:10.1145/2639490.2639508.

[33] S. Grimstad, M. Jørgensen, K. Moløkken-Østvold, Software effort estimation terminology: The tower of babel, Information and Software Technology 48 (4) (2006) 302–310. doi:10.1016/j.infsof.2005.04.004.

[34] R. Britto, V. Freitas, E. Mendes, M. Usman, Effort estimation in global software development: A systematic literature review, in: 9th International Conference on Global Software Engineering, IEEE, Shnaghai, China, 2014, pp. 135–144. doi:10.1109/ICGSE.2014.11.

[35] B. Andrew, A. Selamat, Systematic literature review of missing data imputation techniques for effort prediction, in: International Conference on Information and Knowledge Management, Singapore, 2012, pp. 222–226. URL http://ipcsit.com/vol45/043-ICIKM2012-M0087.pdf

[36] M. Usman, E. Mendes, F. Weidt, R. Britto, Effort estimation in agile software development: a systematic literature review, in: 10th International Conference on Predictive Models in Software Engineering, ACM, Torino, Italy, 2014, pp. 82–91. doi:10.1145/2639490.2639503.

[37] M. Jørgensen, Forecasting of software development work effort: Evidence on expert judgement and formal models, International Journal of Forecasting 23 (3) (2007) 449 – 462. doi:10.1016/j.ijforecast.2007.05.008.

[38] JabRef Development Team, JabRef Version 3.3, http://www.jabref.org [Accessed on 15 May 2016] (2016). URL http://www.jabref.org

[39] M. Steyvers, T. Griffiths, Probabilistic Topic Models, Laurence Erlbaum, 2007, Ch. Latent Semantic Analysis: A Road to Meaning., pp. 1–13. URL https://cocosci.berkeley.edu/tom/papers/SteyversGriffiths.pdf

[40] N. Sidorova, A.nd Evangelopoulos, J. Valacich, T. Ramakrishnan, Uncovering the intellectual core of the information systems discipline, MIS Quarterly: Management Information Systemsdoi:10.1007/978-1-4614-7158-5_2.

[41] S. S. Sehra, Custom .csv exporter for jabref, https://bitbucket.org/sukhjitsehra/topic-modelling, [Accessed on 15 May 2016] (2016).

[42] S. Bird, Natural language toolkit (nltk), version 3.2, [Online; accessed on July, 2016] (2016). URL https://pypi.python.org/pypi/nltk

[43] R. Boulton, Pystemmer, https://github.com/snowballstem/pystemmer, [Online; accessed on July, 2016] (2006).

[44] T. Mavridis, A. L. Symeonidis, Semantic analysis of web documents for the generation of optimal content, Engineering Applications of Artificial Intelligence 35 (2014) 114–130. doi:10.1016/j.engappai.2014.06.008.

[45] R. Alghamdi, K. Alfalqi, A survey of topic modeling in text mining, International Journal of Advanced Com-

puter Science and Applications 6 (1) (2015) 147–153. doi:10.14569/IJACSA.2015.060121.

[46] R. Arun, V. Suresh, C. V. Madhavan, M. N. Murthy, On finding the natural number of topics with latent dirichlet allocation: Some observations, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2010, pp. 391–402. doi:10.1007/978-3-642-13657-3_43.

[47] W. Zhao, J. J. Chen, R. Perkins, Z. Liu, W. Ge, Y. Ding, W. Zou, A heuristic approach to determine an appropriate number of topics in topic modeling, BMC Bioinformatics 16 (13) (2015) S8. doi:10.1186/1471-2105-16-S13-S8.

[48] J. Cao, T. Xia, J. Li, Y. Zhang, S. Tang, A density-based method for adaptive {LDA} model selection, Neurocomputing 72 (7–9) (2009) 1775 – 1781. doi:10.1016/j.neucom.2008.06.011.

[49] R. B. Bradford, An empirical study of required dimensionality for large-scale latent semantic indexing applications, in: 17th ACM Conference on Information and Knowledge Management, CIKM '08, ACM, New York, NY, USA, 2008, pp. 153–162. doi:10.1145/1458082.1458105.

[50] E. Mendes, C. Lokan, R. Harrison, C. Triggs, A replicated comparison of cross-company and within-company effort estimation models using the isbsg database, in: 11th IEEE International Software Metrics Symposium, Como, Italy, 2005, pp. 10 pp.–36. doi:10.1109/METRICS.2005.4.

[51] V. S. Dave, K. Dutta, Neural network based software effort estimation amp; evaluation criterion mmre, in: 2nd International Conference on Computer and Communication Technology, Allahabad,India, 2011, pp. 347–351. doi:10.1109/ICCCT.2011.6075192.

[52] E. Mendes, B. Kitchenham, Further comparison of cross-company and within-company effort estimation models for web applications, in: 10th International Symposium on Software Metrics, Chicago, USA, 2004, pp. 348–357. doi:10.1109/METRIC.2004.1357920.

[53] T. Foss, E. Stensrud, B. Kitchenham, I. Myrtveit, A simulation study of the model evaluation criterion mmre, IEEE Transactions on Software Engineering 29 (11) (2003) 985–995. doi:10.1109/TSE.2003.1245300.

[54] R. Premraj, T. Zimmermann, Building software cost estimation models using homogenous data, in: First International Symposium on Empirical Software Engineering and Measurement, Madrid, Spain, 2007, pp. 393–400. doi:10.1109/ESEM.2007.34.

[55] J. Zhi, V. Garousi-Yusifoğlu, B. Sun, G. Garousi, S. Shahnewaz, G. Ruhe, Cost, benefits and quality of software development documentation: A systematic mapping, Journal of Systems and Software 99 (2015) 175 – 198. doi:10.1016/j.jss.2014.09.042.

[56] M. E. Bajta, A. Idri, J. L. F. Alemán, J. N. Ros, A. Toval, Software cost estimation for global software development - a systematic map and review study, in: 10th International Conference on Evaluation of Novel Approaches to Software Engineering, SciTePress, Barcelona, Spain, 2015, pp. 197–206. doi:10.5220/0005371501970206.

[57] V. Nguyen, Improved size and effort estimation models for software maintenance, in: 26th IEEE International Conference on Software Maintenance, Timioara, Romania, 2010, pp. 1–2. doi:10.1109/ICSM.2010.5609554.

[58] L. O'Brien, Keynote talk: Scope, cost and effort estimation for soa projects, in: 13th Enterprise Distributed Object Computing Conference Workshops, Auckland, New Zealand, 2009, pp. 254–254. doi:10.1109/EDOCW.2009.5331977.

[59] D. E. Ajeh, J. Ellman, S. Keogh, A cost modelling system for cloud computing, in: 14th International Conference on Computational Science and Its Applications, Guimaraes, Portugal, 2014, pp. 74–84. doi:10.1109/ICCSA.2014.24.

[60] T. Kaur, D. Kaur, A. Aggarwal, Cost model for software as a service, in: 5th International Conference Confluence 2014 The Next Generation Information Technology Summit, Uttar Pradesh, India, 2014, pp. 736–741. doi:10.1109/CONFLUENCE.2014.6949281.

[61] I. Buchmann, S. Frischbier, D. Pütz, Towards an estimation model for software maintenance costs, in: 15th European Conference on Software Maintenance and Reengineering, Oldenburg, Germany, 2011, pp. 313–316. doi:10.1109/CSMR.2011.45.

[62] N. C. Narendra, K. Ponnalagu, N. Zhou, W. M. Gifford, Towards a formal model for optimal task-site allocation and effort estimation in global software development, in: Annual SRII Global Conference, San Jose, USA, 2012, pp. 470–477. doi:10.1109/SRII.2012.58.

[63] D. Kang, J. Jung, D.-H. Bae, Constraint-based human resource allocation in software projects, Journal of Software: Practice and Experience 41 (5) (2011) 551–577. doi:10.1002/spe.1030.

[64] M. Jørgensen, D. Sjøberg, Expert Estimation of Software Development Cost: Learning through Feedback, John Wiley & Sons, Ltd, 2006, Ch. Software Evolution and Feedback: Theory and Practice, pp. 489–506. doi:10.1002/0470871822.ch25.

[65] M. Jørgensen, T. M. Gruschke, The impact of lessons-learned sessions on effort estimation and uncertainty assessments, IEEE Transanctions on Software Engineering 35 (3) (2009) 368–383. doi:10.1109/TSE.2009.2.

[66] S. Grimstad, M. Jorgensen, Preliminary study of sequence effects in judgment-based software development work-effort estimation, IET Software 3 (5) (2009) 435–441. doi:10.1049/iet-sen.2008.0110.

[67] S. Grimstad, M. Jørgensen, Inconsistency of expert judgment-based estimates of software development effort, Journal of Systems and Software 80 (11) (2007) 1770 – 1777. doi:10.1016/j.jss.2007.03.001.

[68] M. Jørgensen, K. Moløkken-Østvold, How large are software cost overruns? a review of the 1994 {CHAOS} report, Information and Software Technology 48 (4) (2006) 297 – 301. doi:10.1016/j.infsof.2005.07.002.

[69] G. Wang, G. J. Roedler, R. Valerdi, A. Ankrum, J. E. Gaffney, Harmonizing systems and software cost estimation, in: INCOSE International Symposium, Vol. 19, Singapore, 2009, pp. 232–252. doi:10.1002/j.2334-5837.2009.tb00947.x.

[70] T. Nanri, H. Sato, M. Shimasaki, High Performance Computing, Springer, 1997, Ch. Cost Estimation of Coherence Protocols of Software Managed Cache on Distributed Shared Memory System, pp. 335–342. doi:10.1007/BFb0024228.

[71] E. Mendes, N. Mosley, S. Counsell, Web metrics - estimating design and authoring effort, IEEE MultiMedia

8 (1) (2001) 50–57. doi:10.1109/93.923953.

[72] A. Vickers, Cbse: can we count the cost?, in: Fifth International Symposium on Assessment of Software Tools and Technologies, Pittsburgh, Pennsylvania, 1997, pp. 95–97. doi:10.1109/AST.1997.599917.

[73] E. Corona, G. Concas, M. Marchesi, G. Barabino, D. Grechi, Effort estimation of web applications through web cmf objects, in: Joint Conference of the 22nd International Workshop on Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement, Assisi, Italy, 2012, pp. 15–22. doi:10.1109/IWSM-MENSURA.2012.12.

[74] R. T. Mittermeir, Facets of Software Evolution, John Wiley & Sons, Ltd, 2006, Ch. Software Evolution and Feedback: Theory and Practice, pp. 71–93. doi:10.1002/0470871822.ch4.

[75] M. A. Yahya, R. B. Ahmad, S. Lee, Impact of CMMI based software process maturity on COCOMO ii's effort estimation, International Arab Journal of Information Technology 7 (2) (2010) 129–137. URL http://www.ccis2k.org/iajit/PDF/vol.7,no.2/725.pdf

[76] R. Popli, N. Chauhan, Estimation in agile environment using resistance factors, in: International Conference on Information Systems and Computer Networks, Mathura, India, 2014, pp. 60–65. doi:10.1109/ICISCON.2014.6965219.

[77] H. Wang, H. Wang, H. Zhang, Software productivity analysis with csbsg data set, in: International Conference on Computer Science and Software Engineering, Vol. 2, Wuhan, China, 2008, pp. 587–593. doi:10.1109/CSSE.2008.1178.

[78] Z. Mansor, S. Yahya, N. H. H. Arshad, Software Engineering and Computer Systems, Springer, 2011, Ch. Success Factors in Cost Estimation for Software Development Project, pp. 210–216. doi:10.1007/978-3-642-22170-5_19.

[79] S. Amasaki, C. Lokan, The Evaluation of Weighted Moving Windows for Software Effort Estimation, Springer, 2013, Ch. Product-Focused Software Process Improvement, pp. 214–228. doi:10.1007/978-3-642-39259-7_18.

[80] S. Amasaki, C. Lokan, On the effectiveness of weighted moving windows: Experiment on linear regression based software effort estimation, Journal of Software: Evolution and Process 27 (7) (2015) 488–507. doi:10.1002/smr.1672.

[81] S. Amasaki, C. Lokan, The Effects of Gradual Weighting on Duration-Based Moving Windows for Software Effort Estimationt, Vol. 8892 of Lecture Notes in Computer Science, Springer, 2014, Ch. Product-Focused Software Process Improvemen, pp. 63–77. doi:10.1007/978-3-319-13835-0_5.

[82] S. Amasaki, C. Lokan, A replication study on the effects of weighted moving windows for software effort estimation, in: 20th International Conference on Evaluation and Assessment in Software Engineering, ACM, Limerick, Ireland, 2016, p. 40. doi:10.1145/2915970.2915983.

[83] C. Lokan, E. Mendes, Investigating the use of chronological split for software effort estimation, IET Software 3 (5) (2009) 422–434. doi:10.1049/iet-sen.2008.0107.

[84] H. Diab, M. Frappier, R. S. Denis, Formalizing cosmic-ffp using room, in: ACS/IEEE International Conference on Computer Systems and Applications, Beirut, Lebanon, 2001, pp. 312–318. doi:10.1109/AICCSA.2001.934002.

[85] T. Uemura, S. Kusumoto, K. Inoue, Function point measurement tool for uml design specification, in: Sixth International Software Metrics Symposium, 1999, pp. 62–69. doi:10.1109/METRIC.1999.809727.

[86] S. Kusumoto, K. Inoue, T. Kasimoto, A. Suzuki, K. Yuura, M. Tsuda, Function point measurement for object-oriented requirements specification, in: 24th Annual International Computer Software and Applications Conference, Taipei , Taiwan, 2000, pp. 543–548. doi:10.1109/CMPSAC.2000.884779.

[87] A. A. Akca, A. Tarhan, Run-time measurement of cosmic functional size for java business applications: Initial results, in: Joint Conference of the 22nd International Workshop on Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement, Assisi, Italy, 2012, pp. 226–231. doi:10.1109/IWSM-MENSURA.2012.40.

[88] L. Lavazza, V. del Bianco, G. Liu, Analytical convertibility of functional size measures: A tool-based approach, in: 2012 Joint Conference of the 22nd International Workshop on Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement, Assisi, Italy, 2012, pp. 160–169. doi:10.1109/IWSM-MENSURA.2012.32.

[89] A. Idri, F. A. Amazal, A. Abran, Accuracy comparison of analogy-based software development effort estimation techniques, International Journal of Intelligent Systems 31 (2) (2016) 128–152. doi:10.1002/int.21748.

[90] S. Amasaki, T. Yokogawa, The effects of variable selection methods on linear regression-based effort estimation models, in: Joint Conference of the 23nd International Workshop on Software Measurement (IWSM) and the Eighth International Conference on Software Process and Product Measurement (Mensura), Ankara, Turkey, 2013, pp. 98–103. doi:10.1109/IWSM-Mensura.2013.24.

[91] S. Ezghari, A. Zahi, A. Idri, A learning adaptation cases technique for fuzzy analogy-based software development effort estimation, in: Second World Conference on Complex Systems (WCCS), 2014, pp. 492–497. doi:10.1109/ICoCS.2014.7060958.

[92] F. González-Ladrón-de-Guevara, M. Fernández-Diego, C. Lokan, The usage of ISBSG data fields in software effort estimation: A systematic mapping study, Journal of Systems and Software 113 (2016) 188–215. doi:10.1016/j.jss.2015.11.040.

[93] R. de A. Araújo, A. L. I. Oliveira, S. C. B. Soares, S. R. L. Meira, Gradient-based morphological approach for software development cost estimation, in: The 2011 International Joint Conference on Neural Networks, San Jose, USA, 2011, pp. 588–594. doi:10.1109/ijcnn.2011.6033274.

[94] R. d. A., A. L. I. de Oliveira, S. Soares, Hybrid intelligent design of morphological-rank-linear perceptrons for software development cost estimation, in: 22nd IEEE International Conference on Tools with Artificial Intelligence, Vol. 1, Arras, France, 2010, pp. 160–167. doi:10.1109/ICTAI.2010.30.

[95] R. de A. Araújo, A. L. I. de Oliveira, S. Soares, Hybrid intelligent design of morphological-rank-linear percep-

trons for software development cost estimation, in: 22nd IEEE International Conference on Tools with Artificial Intelligence, Vol. 1, IEEE Computer Society, Arras, 2010, pp. 160–167. doi:10.1109/ICTAI.2010.30.

[96] A. L. I. Oliveira, Estimation of software project effort with support vector regression, Neurocomputing 69 (13-15) (2006) 1749–1753. doi:10.1016/j.neucom.2005.12.119.

[97] L. Song, L. L. Minku, X. Yao, The impact of parameter tuning on software effort estimation using learning machines, in: 9th International Conference on Predictive Models in Software Engineering, Baltimore, USA, 2013, pp. 9:1–9:10. doi:10.1145/2499393.2499394.

[98] D. S. Rosenblum, E. J. Weyuker, Using coverage information to predict the cost-effectiveness of regression testing strategies, IEEE Transactions on Software Engineering 23 (3) (1997) 146–156. doi:10.1109/32.585502.

[99] P. C. Pendharkar, Ensemble based point and confidence interval forecasting in software engineering, Expert Systems with Applications 42 (24) (2015) 9441 – 9448. doi:10.1016/j.eswa.2015.08.002.

[100] B. A. Malloy, P. J. Clarke, E. L. Lloyd, A parameterized cost model to order classes for class-based testing of c++ applications, in: 14th International Symposium on Software Reliability Engineering, Denver,USA, 2003, pp. 353–364. doi:10.1109/ISSRE.2003.1251057.

[101] E. Kocaguneli, T. Menzies, E. Mendes, Transfer learning in effort estimation, Empirical Software Engineering 20 (3) (2015) 813–843. doi:10.1007/s10664-014-9300-5.

[102] L. L. Minku, X. Yao, An analysis of multi-objective evolutionary algorithms for training ensemble models based on different performance measures in software effort estimation, in: 9th International Conference on Predictive Models in Software Engineering, Baltimore, USA, 2013, pp. 8:1–8:10. doi:10.1145/2499393.2499396.

[103] R. Davidrajuh, Extended reachability graph of petri net for cost estimation, in: 8th EUROSIM Congress on Modelling and Simulation, Wales, United Kingdom, 2013, pp. 378–383. doi:10.1109/EUROSIM.2013.72.

[104] M. Grottke, C. Graf, Modeling and predicting software failure costs, in: 33rd Annual IEEE International Computer Software and Applications Conference, Vol. 2, Washington, USA, 2009, pp. 180–189. doi:10.1109/COMPSAC.2009.195.

[105] C. Jones, Activity based software costing, Computer 29 (5) (1996) 103–104. doi:10.1109/2.494092.

[106] J. Ruohonen, S. Hyrynsalmi, V. Leppänen, Time series trends in software evolution, Journal of Software: Evolution and Process 27 (12) (2015) 990–1015. doi:10.1002/smr.1755.

[107] H. Xia, M. Dawande, V. Mookerjee, Optimal coordination in distributed software development, Production and Operations Management 25 (1) (2016) 56–76. doi:10.1111/poms.12408.

[108] A. Idri, A. Abran, S. Mbarki, An experiment on the design of radial basis function neural networks for software cost estimation, in: International Conference on Information & Communication Technologies: From Theory to Applications, Vol. 2, Damascus, Syria, 2006, pp. 1612–1617. doi:10.1109/ICTTA.2006.1684625.

[109] W. L. Du, D. Ho, L. F. Capretz, Improving software effort estimation using neuro-fuzzy model with SEER-SEM, Global Journal of Computer Science and Technology abs/1507.06917.
URL https://arxiv.org/ftp/arxiv/papers/1507/1507.06917.pdf

[110] Z. Muzaffar, M. A. Ahmed, Software development effort prediction: A study on the factors impacting the accuracy of fuzzy logic systems, Information and Software Technology 52 (1) (2010) 92 – 109. doi:10.1016/j.infsof.2009.08.001.

[111] M. K. Ghose, R. Bhatnagar, V. Bhattacharjee, Comparing some neural network models for software development effort prediction, in: 2nd National Conference on Emerging Trends and Applications in Computer Science, Meghalaya, India, 2011, pp. 1–4. doi:10.1109/NCETACS.2011.5751391.

[112] Z. Yang, An improved software size estimation method based on object-oriented approach, in: IEEE Symposium on Electrical Electronics Engineering, 2012, pp. 615–617. doi:10.1109/EEESym.2012.6258733.

[113] O. Demirors, C. Gencel, Conceptual association of functional size measurement methods, IEEE Software 26 (3) (2009) 71–78. doi:10.1109/MS.2009.60.

[114] S. Gu, R. Cheng, Y. Jin, Multi-objective ensemble generation, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 5 (5) (2015) 234–245. doi:10.1002/widm.1158.

[115] M. B. M. B. A. Ferchichi, J. Bourey, Design systems engineering of software products: implementation of a software estimation model, in: Multiconference on Computational Engineering in Systems Applications, Beijing, China, 2006, pp. 1181–1188. doi:10.1109/CESA.2006.313501.

[116] A. Heiat, N. Heiat, A model for estimating efforts required for developing small-scale business applications, Journal of Systems and Software 39 (1) (1997) 7 – 14. doi:10.1016/S0164-1212(96)00159-8.

[117] M. Azzeh, A. B. Nassif, A hybrid model for estimating software project effort from use case points, Applied Soft Computing (2016) 981–989doi:10.1016/j.asoc.2016.05.008.

[118] P. Mork, W. Melo, S. Dutcher, C. Curtis, M. Scroggs, Cost estimation for model-driven interoperability: A canonical data modeling approach, in: 14th International Conference on Quality Software, Texas,USA, 2014, pp. 145–153. doi:10.1109/QSIC.2014.51.

[119] A. B. Nassif, D. Ho, L. F. Capretz, Towards an early software estimation using log-linear regression and a multilayer perceptron model, Journal of Systems and Software 86 (1) (2013) 144 – 160. doi:10.1016/j.jss.2012.07.050.

[120] Q. Zhu, Y. Sun, S. Motheramgari, Developing cost models with qualitative variables for dynamic multi-database environments, in: 16th International Conference on Data Engineering, San Diego, USA, 2000, pp. 413–424. doi:10.1109/ICDE.2000.839441.

[121] H. Rahmandad, D. M. Weiss, Dynamics of concurrent software development, System Dynamics Review 25 (3) (2009) 224–249. doi:10.1002/sdr.425.

[122] Y. Tao, D. Papadias, J. Zhang, Cost models for overlapping and multi-version b-trees, in: 18th International Conference on Data Engineering, San Jose, USA, 2002, pp. 191–200. doi:10.1109/ICDE.2002.994709.

[123] M. T. Baldassarre, D. Caivano, G. Visaggio, Software renewal projects estimation using dynamic calibration, in: International Conference on Software Maintenance, Amsterdam, the netherlands, 2003, pp. 105–115. doi:10.1109/ICSM.2003.1235411.

[124] B. Kitchenham, E. Mendes, Software productivity measurement using multiple size measures, IEEE Transactions on Software Engineering 30 (12) (2004) 1023–1035. doi:10.1109/TSE.2004.104.

[125] L. C. Briand, J. Wust, The impact of design properties on development cost in object-oriented systems, in: Seventh International Software Metrics Symposium, London, UK, 2001, pp. 260–271. doi:10.1109/METRIC.2001.915534.

[126] M. Badri, F. Toure, L. Lamontagne, Predicting unit testing effort levels of classes: An exploratory study based on multinomial logistic regression modeling, Procedia Computer Science 62 (2015) 529 – 538. doi:10.1016/j.procs.2015.08.528.

[127] A. Abran, J. P. Jacquet, A structured analysis of the new iso standard on functional size measurement-definition of concepts, in: Fourth IEEE International Symposium and Forum on Software Engineering Standards, Curitiba, Brazil, 1999, pp. 230–241. doi:10.1109/SESS.1999.766599.

[128] J. Keung, Empirical evaluation of analogy-x for software cost estimation, in: Second ACM-IEEE international symposium on Empirical software engineering and measurement, Kaiserslautern, Germany, 2008, pp. 294–296. doi:10.1145/1414004.1414057.

[129] T. Menzies, S. Williams, O. Elrawas, D. Baker, B. Boehm, J. Hihn, K. Lum, R. Madachy, Accurate estimates without local data?, Software Process: Improvement and Practice 14 (4) (2009) 213–225. doi:10.1002/spip.414.

[130] M. Azzeh, A replicated assessment and comparison of adaptation techniques for analogy-based effort estimation, Empirical Software Engineering 17 (1-2) (2012) 90–127. doi:10.1007/s10664-011-9176-6.

[131] Y. Kamei, J. W. Keung, A. Monden, K. Matsumoto, An over-sampling method for analogy-based software effort estimation, in: Second ACM-IEEE international symposium on Empirical software engineering and measurement, ACM, Kaiserslautern, Germany, 2008, pp. 312–314. doi:10.1145/1414004.1414064.

[132] J. Keung, B. A. Kitchenham, Experiments with analogy-x for software cost estimation, in: Australian Software Engineering Conference, Perth, Australia, 2008, pp. 229–238. doi:10.1109/ASWEC.2008.4483211.

[133] Q. Hu, Evaluating alternative software production functions, IEEE Transactions on Software Engineering 23 (6) (1997) 379–387. doi:10.1109/32.601078.

[134] L. Pickard, B. Kitchenham, P. Jones, Q. Hu, Comments on: evaluating alternative software production functions, IEEE Transactions on Software Engineering 25 (2) (1999) 282–285. doi:10.1109/32.761451.

[135] A. Dagnino, H. Srikanth, M. Naedele, D. Brantly, A model to evaluate the economic benefits of software components development, in: IEEE International Conference on Systems, Man and Cybernetics, Vol. 4, Washington,USA, 2003, pp. 3792–3797 vol.4. doi:10.1109/ICSMC.2003.1244479.

[136] T. Bruckhaus, N. H. Madhavii, I. Janssen, J. Henshaw, The impact of tools on software productivity, IEEE Software 13 (5) (1996) 29–38. doi:10.1109/52.536456.

[137] M. A. Mahmood, K. J. Pettingell, A. Shaskevich, Measuring productivity of software projects: A data envelopment analysis approach, Decision Sciences 27 (1) (1996) 57–80. doi:10.1111/j.1540-5915.1996.tb00843.x.

[138] M. Jørgensen, U. Indahl, D. Sjøberg, Software effort estimation by analogy and “regression toward the mean”, Journal of Systems and Software 68 (3) (2003) 253 – 262. doi:10.1016/S0164-1212(03)00066-9.

[139] N. Chiu, S. Huang, The adjusted analogy-based software effort estimation based on similarity distances, Journal of Systems and Software 80 (4) (2007) 628–640. doi:10.1016/j.jss.2006.06.006.

[140] C. Lokan, An empirical analysis of function point adjustment factors, Information and Software Technology 42 (9) (2000) 649 – 659. doi:10.1016/S0950-5849(00)00108-7.

[141] C. J. Burgess, M. Lefley, Can genetic programming improve software effort estimation? A comparative evaluation, Information and Software Technology 43 (14) (2001) 863–873. doi:10.1016/S0950-5849(01)00192-6.

[142] T. K. Le-Do, K. Yoon, Y. Seo, D. Bae, Filtering of inconsistent software project data for analogy-based effort estimation, in: IEEE 34th Annual Computer Software and Applications Conference, Seoul, Korea, 2010, pp. 503–508. doi:10.1109/COMPSAC.2010.56.

[143] T. Chan, S. L. Chung, T. H. Ho, An economic model to estimate software rewriting and replacement times, IEEE Transactions on Software Engineering 22 (8) (1996) 580–598. doi:10.1109/32.536958.

[144] C. L. Martín, A. Chavoya, M. E. Meda-Campaña, Pattern Recognition, Springer, 2011, Ch. Software Development Effort Estimation in Academic Environments Applying a General Regression Neural Network Involving Size and People Factors, pp. 269–277. doi:10.1007/978-3-642-21587-2_29.

[145] N. Nan, D. E. Harter, Impact of budget and schedule pressure on software development cycle time and effort, IEEE Transactions on Software Engineering 35 (5) (2009) 624–637. doi:10.1109/TSE.2009.18.

[146] R. Hughes, Estimating and Segmenting Projects, Morgan Kaufmann, Boston, 2013, Ch. Agile Data Warehousing Project Management, pp. 207 – 248. doi:10.1016/B978-0-12-396463-2.00007-7.

[147] Ö. Ö. Top, B. Özkan, M. Nabi, O. Demirörs, Internal and external software benchmark repository utilization for effort estimation, in: Joint Conference of the 21st International Workshop on Software Measurement and the 6th International Conference on Software Process and product measurement, Nara, Japan, 2011, pp. 302–307. doi:10.1109/IWSM-MENSURA.2011.41.

[148] A. Sharma, D. S. Kushwaha, Estimation of software development effort from requirements based complexity, Procedia Technology 4 (2012) 716 – 722. doi:10.1016/j.protcy.2012.05.116.

[149] A. Tripathi, B. Kumar, A. Sharma, D. S. Kushwaha, Srs based estimation of software maintenance effort, in: Third International Conference on Computer and Communication Technology, Allahabad,India, 2012, pp. 154–155. doi:10.1109/ICCCT.2012.38.

[150] L. F. Capretz, V. Marza, Improving effort estimation by voting software estimation models, Advances in

Software Engineering 2009 (2009) 829725:1–829725:8. doi:10.1155/2009/829725.

[151] A. Sharma, M. Vardhan, D. S. Kushwaha, A versatile approach for the estimation of software development effort based on SRS document, International Journal of Software Engineering and Knowledge Engineering 24 (1) (2014) 1–42. doi:10.1142/S0218194014500016.

[152] K. Ponnalagu, N. C. Narendra, Automated trendline generation for accurate software effort estimation, in: 3rd annual conference on Systems, programming, and applications: software for humanity, ACM, Tucson, USA, 2012, pp. 203–212. doi:10.1145/2384716.2384774.

[153] N. Ramasubbu, R. K. Balan, Overcoming the challenges in cost estimation for distributed software projects, in: 34th International Conference on Software Engineering, IEEE Computer Society, Zurich, Switzerland, 2012, pp. 91–101. doi:10.1109/ICSE.2012.6227203.

[154] J. Liu, X. Li, Factors contribute to high costs of software projects, in: Advanced Software Engineering and Its Applications, Hainan Island, China, 2008, pp. 202–205. doi:10.1109/ASEA.2008.38.

[155] I. Stamelos, L. Angelis, Managing uncertainty in project portfolio cost estimation, Information and Software Technology 43 (13) (2001) 759–768. doi:10.1016/S0950-5849(01)00183-5.

[156] S. Koch, J. Mitlöhner, Software project effort estimation with voting rules, Decision Support Systems 46 (4) (2009) 895–901. doi:10.1016/j.dss.2008.12.002.

[157] I. Myrtveit, E. Stensrud, U. H. Olsson, Analyzing data sets with missing data: an empirical evaluation of imputation methods and likelihood-based methods, IEEE Transactions on Software Engineering 27 (11) (2001) 999–1013. doi:10.1109/32.965340.

[158] W. Zhang, Y. Yang, Q. Wang, Using bayesian regression and {EM} algorithm with missing handling for software effort prediction, Information and Software Technology 58 (2015) 58 – 70. doi:10.1016/j.infsof.2014.10.005.

[159] L. F. Johnson, M. L. Smith, A. M. Stevens, Validation of software effort models, in: IEEE Canadian Conference on Electrical and Computer Engineering, Vol. 1, Edmenton,Canada, 1999, pp. 273–276. doi:10.1109/CCECE.1999.807208.

[160] X. Jing, F. Qi, F. Wu, B. Xu, Missing data imputation based on low-rank recovery and semi-supervised regression for software effort estimation, in: 38th International Conference on Software Engineering, Newyork,USA, 2016, pp. 607–618. doi:10.1145/2884781.2884827.

[161] P. Sentas, L. Angelis, Categorical missing data imputation for software cost estimation by multinomial logistic regression, Journal of Systems and Software 79 (3) (2006) 404–414. doi:10.1016/j.jss.2005.02.026.

[162] E.-Y. Chung, L. Benini, G. D. Micheli, Source code transformation based on software cost analysis, in: The 14th International Symposium on System Synthesis, Montreal,Canada, 2001, pp. 153–158. doi:10.1109/ISSS.2001.156549.

[163] V. Hilderman, Understanding do-178c software certification: Benefits versus costs, in: IEEE 25th International Symposium on Software Reliability Engineering Workshops, Naples, Italy, 2014, pp. 114–114. doi:10.1109/ISSREW.2014.118.

[164] N. R. Mandala, G. S. Walia, J. C. Carver, N. Nagappan, Application of kusumoto cost-metric to evaluate the cost effectiveness of software inspections, in: ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, Lund, Sweden, 2012, pp. 221–230. doi:10.1145/2372251.2372291.

[165] B. Weitzel, Understanding deployment costs of enterprise systems: Towards architecture support in deployment decisions, in: 28th IEEE International Conference on Software Maintenance, Trento, Italy, 2012, pp. 677–680. doi:10.1109/ICSM.2012.6405352.

[166] R. de A. Araújo, A. L. I. de Oliveira, S. C. B. Soares, A morphological-rank-linear approach for software development cost estimation, in: 21st IEEE International Conference on Tools with Artificial Intelligence, Newyork, USA, 2009, pp. 630–636. doi:10.1109/ictai.2009.39.

[167] R. de A. Araújo, A. L. I. de Oliveira, S. C. B. Soares, A shift-invariant morphological system for software development cost estimation, Expert System and Applications 38 (4) (2011) 4162–4168. doi:10.1016/j.eswa.2010.09.078.

[168] X. Li, C. Guo, H. Wang, A runtime software monitoring cost analysis method, in: 2nd IEEE International Conference onInformation Management and Engineering, 2010, pp. 97–101. doi:10.1109/ICIME.2010.5477511.

[169] V. Lenarduzzi, Could social factors influence the effort software estimation?, in: 7th International Workshop on Social Software Engineering, ACM, Bergamo, Italy, 2015, pp. 21–24. doi:10.1145/2804381.2804385.

[170] H. Wagner, O. Pankratz, W. Mellis, D. Basten, Effort of eai projects: A repertory grid investigation of influencing factors, Project Management 46 (5) (2015) 62–80. doi:10.1002/pmj.21523.

[171] J. Alwidian, W. Hadi, Enhancing the results of ucp in cost estimation using new external environmental factors, in: International Conference on Information Technology and e-Services, Sousse, Tunisia, 2012, pp. 1–11. doi:10.1109/ICITeS.2012.6216623.

[172] N. Hurtado, M. Ruiz, E. Orta, J. Torres, Using simulation to aid decision making in managing the usability evaluation process, Information and Software Technology 57 (2015) 509 – 526. doi:10.1016/j.infsof.2014.06.001.

[173] Y. Yokoyama, M. Kodaira, Software cost and quality analysis by statistical approaches, in: International Conference on Software Engineering, Kyoto,Japan, 1998, pp. 465–467. doi:10.1109/ICSE.1998.671607.

[174] K. Paithankar, M. Ingle, Characterization of software projects by restructuring parameters for usability evaluation, in: Second International Conference on Computer and Electrical Engineering, Vol. 2, Dubai, 2009, pp. 436–441. doi:10.1109/ICCEE.2009.216.

[175] M. Hoppe, A. Engel, S. Shachar, Systest: Improving the verification, validation, and testing process— assessing six industrial pilot projects, Systems Engineering 10 (4) (2007) 323–347. doi:10.1002/sys.20082.

[176] M. Fernández-Diego, F. G.-L. de Guevara, Potential and limitations of the {ISBSG} dataset in enhancing software engineering research: A mapping review, Information and Software Technology 56 (6) (2014) 527 – 544. doi:10.1016/j.infsof.2014.01.003.

[177] C. Mair, M. Shepperd, The consistency of empirical comparisons of regression and analogy-based software project cost prediction, in: International Symposium on Empirical Software Engineering, Queensland, Australia, 2005, pp. 509–518. doi:10.1109/ISESE.2005.1541858.

[178] F. González-Ladrón-de-Guevara, M. Fernández-Diego, ISBSG variables most frequently used for software effort estimation: a mapping review, in: 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, Torino, Italy, 2014, pp. 42:1–42:4. doi:10.1145/2652524.2652550.

[179] E. Kocaguneli, T. Menzies, Software effort models should be assessed via leave-one-out validation, Journal of Systems and Software 86 (7) (2013) 1879 – 1890. doi:10.1016/j.jss.2013.02.053.

[180] M. Li, W. Zhang, Q. Lu, Y. Xia, X. Wang, The estimated methodology for military software support cost, in: IEEE Symposium on Robotics and Applications, Kuala Lumpur, Malaysia, 2012, pp. 443–446. doi:10.1109/ISRA.2012.6219220.

[181] O. Benediktsson, D. Dalcher, K. Reed, M. Woodman, Cocomo-based effort estimation for iterative and incremental software development, Software Quality Journal 11 (4) (2003) 265–281. doi:10.1023/A:1025809010217.

[182] H. Cannon, K. Gundy-Burlet, Software cost estimation for the ladee mission, in: IEEE Aerospace Conference, Big Sky,USA, 2015, pp. 1–8. doi:10.1109/AERO.2015.7119109.

[183] J. Gao, M. Shah, M. Shah, D. Vyas, P. Pattabhiraman, K. Dandapani, E. Bari, Systematic risk assessment and cost estimation for software problems, in: 21st International Conference on Software Engineering & Knowledge Engineering, Massachusetts, USA, 2009, pp. 103–109.
URL https://www.researchgate.net/publication/221390589_Systematic_Risk_Assessment_and_Cost_Estimation_for_Software_Problems

[184] E. Kocaguneli, T. Menzies, J. Hihn, B. H. Kang, Size doesn't matter?: on the value of software size features for effort estimation, in: 8th International Conference on Predictive Models in Software Engineering, ACM, Lund,Sweden, 2012, pp. 89–98. doi:10.1145/2365324.2365336.

[185] Y. F. Li, M. Xie, T. N. Goh, A study of mutual information based feature selection for case based reasoning in software cost estimation, Expert Systems with Applications 36 (3) (2009) 5921–5931. doi:10.1016/j.eswa.2008.07.062.

[186] J. Moses, M. Farrow, Assessing variation in development effort consistency using a data source with missing data, Software Quality Journal 11 (4) (2003) 283–300. doi:10.1007/s11219-004-5261-z.

[187] J. Kothari, T. Denton, A. Shokoufandeh, S. Mancoridis, Reducing program comprehension effort in evolving software by recognizing feature implementation convergence, in: 15th IEEE International Conference on Program Comprehension, Banff, Canada, 2007, pp. 17–26. doi:10.1109/ICPC.2007.33.

[188] S. Inoue, S. Yamada, Bootstrap interval estimation methods for cost-optimal software release planning, in: IEEE International Conference on Systems, Man, and Cybernetics, IEEE, Manchestar,UK, 2013, pp. 621–626. doi:10.1109/SMC.2013.111.

[189] S. M. Nadgeri, V. P. Hulsure, A. D. Gawande, Comparative study of various regression methods for software effort estimation, in: Third International Conference on Emerging Trends in Engineering and Technology, Goa, India, 2010, pp. 642–645. doi:10.1109/ICETET.2010.22.

[190] N. Mittas, L. Angelis, Bootstrap confidence intervals for regression error characteristic curves evaluating the prediction error of software cost estimation models, in: Workshops of the 5th IFIP Conference on Artificial Intelligence Applications & Innovations, Vol. 475 of CEUR Workshop Proceedings, Thessaloniki, Greece, 2009, pp. 221–230.
URL http://ceur-ws.org/Vol-475/AISEW2009/23-pp-221-230-210.pdf

[191] P. L. Braga, A. L. I. Oliveira, S. R. L. Meira, Software effort estimation using machine learning techniques with robust confidence intervals, in: 19th IEEE International Conference on Tools with Artificial Intelligence, Patras,Greece, 2007, pp. 352–357. doi:10.1109/ICTAI.2007.172.

[192] M. Jørgensen, D. Sjøberg, An effort prediction interval approach based on the empirical distribution of previous estimation accuracy, Information and Software Technology 45 (3) (2003) 123 – 136. doi:10.1016/S0950-5849(02)00188-X.

[193] Z. Jie, Z. Rongcai, Y. Yuan, A nested loop fusion algorithm based on cost analysis, in: The 14th IEEE International Conference On High Performance Computing And Communications The 9th IEEE International Conference On Embedded Software And Systems, Liverpool, United Kingdom, 2012, pp. 1096–1101. doi:10.1109/HPCC.2012.160.

[194] A. J. Nolan, A. C. Pickard, The 7 +/- 2 uses for an estimation tool, in: INCOSE International Symposium, Vol. 22, Rome, Italy, 2012, pp. 816–828. doi:10.1002/j.2334-5837.2012.tb01373.x.

[195] A. J. Nolan, S. Abrahão, Software Product Lines: Going Beyond, Springer, 2010, Ch. Dealing with Cost Estimation in Software Product Lines: Experiences and Future Directions, pp. 121–135. doi:10.1007/978-3-642-15579-6_9.

[196] A. Chavoya, C. Lopez-Martin, M. E. Meda-Campa, Applying genetic programming for estimating software development effort of short-scale projects, in: Eighth International Conference on Information Technology: New Generations, Las Vegas,USA, 2011, pp. 174–179. doi:10.1109/ITNG.2011.37.

[197] S. Kusumoto, F. Matukawa, K. Inoue, S. Hanabusa, Y. Maegawa, Estimating effort by use case points: method, tool and case study, in: 10th International Symposium on Software Metrics, Chicago, Illinois, 2004, pp. 292–299. doi:10.1109/METRIC.2004.1357913.

[198] T. Yamaura, T. Kikuno, A framework for top-down cost estimation of software development, in: The Twenty Third Annual International Computer Software and Applications Conference, IEEE Computer Society, Phoenix, Arizona, 1999, pp. 322–323. doi:10.1109/CMPSAC.1999.812729.

[199] Y. F. Li, M. Xie, T. N. Goh, Bayesian inference approach for probabilistic analogy based software maintenance effort estimation, in: The 14th IEEE Pacific Rim International Symposium on Dependable Computing, Taipei, Taiwan, 2008, pp. 176–183. doi:

10.1109/PRDC.2008.21.

[200] V. Bozhikova, M. Stoeva, An approach for software cost estimation, in: 11th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing on International Conference on Computer Systems and Technologies, Sofia, Bulgaria, 2010, pp. 119–124. doi:10.1145/1839379.1839401.

[201] M. Azzeh, A. B. Nassif, L. L. Minku, An empirical evaluation of ensemble adjustment methods for analogy-based effort estimation, Journal of Systems and Software 103 (2015) 36–52. doi:10.1016/j.jss.2015.01.028.

[202] A. Henrich, Repository Based Software Cost Estimation, Springer, 1997, Ch. Database and Expert Systems Applications, pp. 653–662. doi:10.1007/BFb0022073.

[203] P. Shoval, O. Feldman, A combination of the mk-ii function points software estimation method with the {ADISSA} methodology for systems analysis and design, Information and Software Technology 39 (13) (1997) 855 – 865. doi:10.1016/S0950-5849(97)00009-8.

[204] J. Liu, J. Qiao, A grey rough set model for evaluation and selection of software cost estimation methods, Grey Systems: Theory and Application 4 (1) (2014) 3–12. doi:10.1108/GS-08-2013-0016.

[205] P. Dhavachelvan, N. Sivakumar, N. S. N. Cailassame, S. S. Kumar, V. S. K. Venkatachalapathy, Software complexity assessment using statistical technique: Towards software testing effort estimation, in: 2nd Indian International Conference on Artificial Intelligence, Pune, India, 2005, pp. 2045–2055. doi:10.1109/ICISIP.2005.1529476.

[206] S. Huang, N. Chiu, L. Chen, Integration of the grey relational analysis with genetic algorithm for software effort estimation, European Journal of Operational Research 188 (3) (2008) 898–909. doi:10.1016/j.ejor.2007.07.002.

[207] C. j. Hsu, C. y. Huang, Comparison and assessment of improved grey relation analysis for software development effort estimation, in: IEEE International Conference on Management of Innovation and Technology, Vol. 2, 2006, pp. 663–667. doi:10.1109/ICMIT.2006.262302.

[208] S. Dighe, A. Joshi, Human-Centered Software Engineering, Springer, 2014, Ch. An Autoethnographic Study of HCI Effort Estimation in Outsourced Software Development, pp. 19–35. doi:10.1007/978-3-662-44811-3_2.

[209] M. Jorgensen, K. Molokken, A preliminary checklist for software cost management, in: Third International Conference on Quality Software, 2003, pp. 134–140. doi:10.1109/QSIC.2003.1319095.

[210] R. Lagerström, P. Johnson, M. Ekstedt, Architecture analysis of enterprise systems modifiability: a metamodel for software change cost estimation, Software Quality Journal 18 (4) (2010) 437–468. doi:10.1007/s11219-010-9100-0.

[211] W.-T. Lee, W.-Y. Deng, J. Lee, S.-J. Lee, Change impact analysis with a goal-driven traceability-based approach, International Journal of Intelligent Systems 25 (8) (2010) 878–908. doi:10.1002/int.20443.

[212] B. Tanveer, Hybrid Effort Estimation of Changes in Agile Software Development, Springer, 2016, Ch. Agile Processes, in Software Engineering, and Extreme Programming, pp. 316–320. doi:10.1007/978-3-319-33515-5_33.

[213] B. Boehm, R. Valerdi, Impact of software resource estimation research on practice: a preliminary report on achievements, synergies, and challenges, in: 33rd International Conference on Software Engineering, Honolulu, USA, 2011, pp. 1057–1065. doi:10.1145/1985793.1985994.

[214] J. Law, G. Rothermel, Whole program path-based dynamic impact analysis, in: 25th International Conference on Software Engineering, Portland,USA, 2003, pp. 308–318. doi:10.1109/ICSE.2003.1201210.

[215] A. Israeli, D. G. Feitelson, The linux kernel as a case study in software evolution, Journal of Systems and Software 83 (3) (2010) 485 – 501. doi:10.1016/j.jss.2009.09.042.

[216] T. Kurtz, Project planning and estimation with ask pete, in: 26th Annual NASA Goddard Software Engineering Workshop, Greenbelt, Maryland, 2001, pp. 55–61. doi:10.1109/SEW.2001.992655.

[217] A. Rainer, M. Shepperd, Re-planning for a successful project schedule, in: Sixth International Software Metrics Symposium, Boca Raton,USA, 1999, pp. 72–81. doi:10.1109/METRIC.1999.809728.

[218] J. K. Lee, N. Lee, Least modification principle for case-based reasoning: a software project planning experience, Expert Systems with Applications 30 (2) (2006) 190 – 202. doi:10.1016/j.eswa.2005.06.021.

[219] F. A. Amazal, A. Idri, A. Abran, An analogy-based approach to estimation of software development effort using categorical data, in: Joint conference of the 24th International Workshop on Software Measurement (IWSM) and the 9th International Conference on Software Process and Product Measurement (Mensura), Rotterdam, the Netherlands, 2014, pp. 252–262. doi:10.1109/IWSM.Mensura.2014.31.

[220] A. Idri, A. Abran, T. M. Khoshgoftaar, Estimating software project effort by analogy based on linguistic values, in: Eighth IEEE Symposium on Software Metrics, Ottawa, Canada, 2002, pp. 21–30. doi:10.1109/METRIC.2002.1011322.

[221] F. A. Amazal, A. Idri, A. Abran, Improving fuzzy analogy based software development effort estimation, in: 21st Asia-Pacific Software Engineering Conference, Jeju, South Korea, 2014, pp. 247–254. doi:10.1109/apsec.2014.46.

[222] V. D. Bianco, L. Lavazza, An empirical assessment of function point-like object-oriented metrics, in: 11th IEEE International Software Metrics Symposium, Como, Italy, 2005, pp. 10 –40. doi:10.1109/METRICS.2005.9.

[223] K. Akingbehin, A structured framework for software metrics based on three primary metrics, in: 9th IEEE/ACIS International Conference on Computer and Information Science, Yamagata, Japan, 2010, pp. 749–752. doi:10.1109/ICIS.2010.158.

[224] V. B. Misic, D. N. Tesic, Estimation of effort and complexity: An object-oriented case study, Journal of Systems and Software 41 (2) (1998) 133–143. doi:10.1016/S0164-1212(97)10014-0.

[225] P. Nesi, T. Querci, Effort estimation and prediction of object-oriented systems, Journal of Systems and Software 42 (1) (1998) 89–102. doi:10.1016/S0164-1212(97)10021-8.

[226] Y. Zhou, Y. Yang, B. Xu, H. Leung, X. Zhou, Source code size estimation approaches for object-oriented systems from {UML} class diagrams: A comparative study, Information and Software Technology 56 (2) (2014) 220 – 237. doi:10.1016/j.infsof.2013.09.003.

[227] H. B. Yadav, D. K. Yadav, A fuzzy logic based approach for phase-wise software defects prediction using software metrics, Information and Software Technology 63 (2015) 44 – 57. doi:10.1016/j.infsof.2015.03.001.

[228] M. Grottke, B. Schleich, Cost optimality in testing and rejuvenation, in: 23rd IEEE International Symposium on Software Reliability Engineering Supplemental Proceedings, Texas, USA, 2012, pp. 259–264. doi:10.1109/ISSREW.2012.84.

[229] Y. Ahn, J. Suh, S. Kim, H. Kim, The software maintenance project effort estimation model based on function points, Journal of Software Maintenance 15 (2) (2003) 71–85. doi:10.1002/smr.269.

[230] R. G. Kula, K. Fushida, N. Yoshida, H. Iida, Experimental study of quantitative analysis of maintenance effort using program slicing-based metrics, in: 19th Asia-Pacific Software Engineering Conference, Vol. 2, Hong Kong, 2012, pp. 50–57. doi:10.1109/APSEC.2012.105.

[231] M. Tsunoda, K. Toda, K. Fushida, Y. Kamei, M. Nagappan, N. Ubayashi, Revisiting software development effort estimation based on early phase development activities, in: Tenth Working Conference on Mining Software Repositories, San Francisco,USA, 2013, pp. 429–438. doi:10.1109/MSR.2013.6624059.

[232] S. I. K. Wu, The quality of design team factors on software effort estimation, in: IEEE International Conference on Service Operations and Logistics, and Informatics, Shanghai,China, 2006, pp. 6–11. doi:10.1109/SOLI.2006.328973.

[233] R. Litoriya, N. Sharma, A. Kothari, Incorporating cost driver substitution to improve the effort using agile cocomo ii, in: Sixth International Conference on Software Engineering, Indore,India, 2012, pp. 1–7. doi:10.1109/CONSEG.2012.6349494.

[234] D. Yang, Q. Wang, M. Li, Y. Yang, K. Ye, J. Du, A survey on software cost estimation in the chinese software industry, in: Second ACM-IEEE international symposium on Empirical software engineering and measurement, ACM, Kaiserslautern, Germany, 2008, pp. 253–262. doi:10.1145/1414004.1414045.

[235] R. Valerdi, T. R. Ryan, Total Cost of Ownership (TOC): An Approach for Estimating UMAS Costs, John Wiley & Sons, Ltd, 2016, Ch. Operations Research for Unmanned Systems, pp. 207–232. doi:10.1002/9781118918937.ch11.

[236] M. Jorgensen, S. Grimstad, Over-optimism in software development projects: "the winner 's curse", in: 15th International Conference on Electronics, Communications and Computers, Pubela, Mexico, 2005, pp. 280–285. doi:10.1109/CONIEL.2005.58.

[237] W. He, A framework of combining case-based reasoning with a work breakdown structure for estimating the cost of online course production projects, British Journal of Educational Technology 45 (4) (2014) 595–605. doi:10.1111/bjet.12043.

[238] Z. Li, J. Keung, Software cost estimation framework for service-oriented architecture systems using divide-and-conquer approach, in: The Fifth IEEE International Symposium on Service Oriented System Engineering, IEEE Computer Society, Nanjing, China, 2010, pp. 47–54. doi:10.1109/SOSE.2010.29.

[239] R. Nilchiani, M. Efatmaneshnik, K. Dalili, Measuring the value of adaptability of fractionated spacecrafts, in: INCOSE International Symposium, Vol. 22, Rome,Italy, 2012, pp. 1165–1181. doi:10.1002/j.2334-5837.2012.tb01395.x.

[240] N. J. Pizzi, A fuzzy classifier approach to estimating software quality, Information Sciences 241 (2013) 1 – 11. doi:10.1016/j.ins.2013.04.027.

[241] L. Antonelli, G. Rossi, J. C. S. do Prado Leite, A. Oliveros, Language extended lexicon points: Estimating the size of an application using its language, in: IEEE 22nd International Requirements Engineering Conference, Karlskrona, Sweden, 2014, pp. 263–272. doi:10.1109/RE.2014.6912268.

[242] K. Børte, S. R. Ludvigsen, A. I. Mørch, The role of social interaction in software effort estimation: Unpacking the "magic step" between reasoning and decision-making, Information and Software Technology 54 (9) (2012) 985 – 996. doi:10.1016/j.infsof.2012.03.002.

[243] M. W. Janette, R. P. Will, J. Blanton, Enhancing knowledge elicitation using the cognitive interview, Expert Systems with Applications 10 (1) (1996) 127 – 133. doi:10.1016/0957-4174(95)00039-9.

[244] E. Bjarnason, K. Wnuk, B. Regnell, Are you biting off more than you can chew? a case study on causes and effects of overscoping in large-scale software engineering, Information and Software Technology 54 (10) (2012) 1107 – 1124. doi:10.1016/j.infsof.2012.04.006.

[245] A. Abran, Software estimation: Transforming dust into pots of gold?, in: Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA), Rotterdam,the Netherlands, 2014, pp. 64–65. doi:10.1109/IWSM.Mensura.2014.48.

[246] O. Shmueli, N. Pliskin, L. Fink, Can the outside-view approach improve planning decisions in software development projects?, Information Systems Journal 26 (4) (2015) 395–418. doi:10.1111/isj.12091.

[247] A. Idri, A. Zahi, E. Mendes, A. Zakrani, Software Cost Estimation Models Using Radial Basis Function Neural Networks, Vol. 4895 of Lecture Notes in Computer Science, Springer, 2007, Ch. Software Process and Product Measurements, pp. 21–31. doi:10.1007/978-3-540-85553-8_2.

[248] M. Madheswaran, D. Sivakumar, Enhancement of prediction accuracy in cocomo model for software project using neural network, in: International Conference on Computing, Communication and Networking Technologies, Hefei, China, 2014, pp. 1–5. doi:10.1109/ICCCNT.2014.6963021.

[249] V. S. Dave, K. Dutta, Comparison of regression model, feed-forward neural network and radial basis neural network for software development effort estimation, ACM SIGSOFT Software Engineering Notes 36 (5) (2011) 1–5. doi:10.1145/2020976.2020982.

[250] M. Daneva, A. Herrmann, Requirements prioritization based on benefit and cost prediction: A method classification framework, in: 34th Euromicro Con-

ference Software Engineering and Advanced Applications, Parma,Italy, 2008, pp. 240–247. doi:10.1109/SEAA.2008.46.

[251] C.-T. Lin, C.-Y. Huang, J.-R. Chang, Integrating generalized weibull-type testing-effort function and multiple change-points into software reliability growth models, in: 12th Asia-Pacific Software Engineering Conference, Taipei, Taiwan, 2005, pp. 431–438. doi:10.1109/APSEC.2005.74.

[252] C. Huang, J. Lo, S. Kuo, M. R. Lyu, Software reliability modeling and cost estimation incorporating testing-effort and efficiency, in: l0th International Symposium on Software Reliability Engineering, IEEE Computer Society, Boca Raton, USA, 1999, pp. 62–72. doi:10.1109/ISSRE.1999.809311.

[253] R. Burnett, A trade-off method between cost and reliability, in: International Conference of the Chilean Computer Science Society, Valparaiso, Chile, 1997, pp. 21–28. doi:10.1109/SCCC.1997.636852.

[254] M. Kassab, M. Daneva, O. Ormandjieva, A metamodel for the assessment of non-functional requirement size, in: 34th Euromicro Conference Software Engineering and Advanced Applications, Parma,Italy, 2008, pp. 411–418. doi:10.1109/SEAA.2008.58.

[255] L. Lavazza, R. Meli, An evaluation of simple function point as a replacement of ifpug function point, in: Joint conference of the 24th International Workshop on Software Measurement (IWSM) and the 9th International Conference on Software Process and Product Measurement(Mensura), Rotterdam, the Netherlands, 2014, pp. 196–206. doi:10.1109/IWSM.Mensura.2014.28.

[256] L. Buglione, Ç. Gencel, Impact of Base Functional Component Types on Software Functional Size Based Effort Estimation, Springer, 2008, Ch. Product-Focused Software Process Improvement, pp. 75–89. doi:10.1007/978-3-540-69566-0_9.

[257] J. J. Cuadrado-Gallego, P. Rodriguez-Soria, A. Gonzalez, D. Castelo, S. Hakimuddin, Early functional size estimation with ifpug unit modified, in: 9th IEEE/ACIS International Conference on Computer and Information Science, Yamagata, Japan, 2010, pp. 729–733. doi:10.1109/ICIS.2010.12.

[258] B. Özkan, O. Türetken, O. Demirörs, Software Functional Size: For Cost Estimation and More, Springer, 2008, Ch. Software Process Improvement, pp. 59–69. doi:10.1007/978-3-540-85936-9_6.

[259] B. Palanisamy, A. Singh, L. Liu, B. Langston, Cura: A cost-optimized model for mapreduce in a cloud, in: IEEE 27th International Symposium on Parallel Distributed Processing, Boston, USA, 2013, pp. 1275–1286. doi:10.1109/IPDPS.2013.20.

[260] S. H. Liew, Y. Y. Su, Cloudguide: Helping users estimate cloud deployment cost and performance for legacy web applications, in: IEEE 4th International Conference on Cloud Computing Technology and Science, Taipei, Taiwan, 2012, pp. 90–98. doi:10.1109/CloudCom.2012.6427577.

[261] O. Mazhelis, P. Tyrväinen, L. Frank, Vertical software industry evolution: The impact of software costs and limited customer base, Information and Software Technology 55 (4) (2013) 690 – 698. doi:10.1016/j.infsof.2012.10.006.

[262] J. Gaisler, Cost-cutting through standardization, in: 10th IEEE Pacific Rim International Symposium on Dependable Computing, 2004, pp. 337–. doi:10.1109/PRDC.2004.1276586.

[263] E. Mendes, I. Watson, C. Triggs, N. Mosley, S. Counsell, A comparison of development effort estimation techniques for web hypermedia applications, in: Eighth IEEE Symposium on Software Metrics, Ottawa, Canada, 2002, pp. 131–140. doi:10.1109/METRIC.2002.1011332.

[264] Y.-F. Li, M. Xie, T.-N. Goh, Adaptive ridge regression system for software cost estimating on multi-collinear datasets, Journal of Systems and Software 83 (11) (2010) 2332 – 2343, interplay between Usability Evaluation and Software Development. doi:10.1016/j.jss.2010.07.032.

[265] M. Fernández-Diego, S. Elmouaden, J. Torralba-Martínez, Software effort estimation using NBC and SWR: A comparison based on ISBSG projects, in: Joint Conference of the 22nd International Workshop on Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement, IEEE Computer Society, Assisi,Italy, 2012, pp. 132–136. doi:10.1109/IWSM-MENSURA.2012.28.

[266] E. Mendes, N. Mosley, Comparing effort prediction models for web design and authoring using boxplots, in: 24th Australasian Computer Science Conference, Gold Coast, Australia, 2001, pp. 125–133. doi:10.1109/ACSC.2001.906632.

[267] C. van Koten, A. Gray, Bayesian statistical effort prediction models for data-centred 4gl software development, Information and Software Technology 48 (11) (2006) 1056 – 1067. doi:10.1016/j.infsof.2006.01.001.

[268] G. Wang, R. Valerdi, G. J. Roedler, A. Ankrum, J. E. G. Jr., Harmonising software engineering and systems engineering cost estimation, International Journal of Computer Integrated Manufacturing 25 (4-5) (2012) 432–443. doi:10.1080/0951192X.2010.542182.

[269] R. Valerdi, B. W. Boehm, D. J. Reifer, Cosysmo: A constructive systems engineering cost model coming of age, in: INCOSE International Symposium, Vol. 13, Washington, USA, 2003, pp. 70–82. doi:10.1002/j.2334-5837.2003.tb02601.x.

[270] M. A. Bone, R. Cloutier, Applying systems engineering modeling language (sysml) to system effort estimation utilizing use case points, in: INCOSE International Symposium, Vol. 21, Denver, USA, 2011, pp. 114–127. doi:10.1002/j.2334-5837.2011.tb01189.x.

[271] Y. Wang, Y. Yuan, The formal economic model of software engineering, in: 2006 Canadian Conference on Electrical and Computer Engineering, Ottawa,Canada, 2006, pp. 2385–2388. doi:10.1109/CCECE.2006.277682.

[272] T. C. Hooi, Y. Yusoff, Z. Hassan, Comparative study on applicability of webmo in web application cost estimation within klang valley in malaysia, in: .8th IEEE International Conference on Computer and Information Technology Workshops, Sydney,Australia, 2008, pp. 116–121. doi:10.1109/CIT.2008.Workshops.48.

[273] G. Barabino, G. Concas, E. Corona, D. Grechi, M. Marchesi, D. Tigano, Web framework points: an effort estimation methodology for web application development using a content management framework, Journal of Software: Evolution and Process 27 (9) (2015) 603–624. doi:10.1002/smr.1715.

[274] M. Ruhe, R. Jeffery, I. Wieczorek, Cost estimation for web applications, in: 25th International Conference on Software Engineering, Portland, USA, 2003, pp. 285–294. doi:10.1109/ICSE.2003.1201208.

[275] S. Escolar, J. Carretero, An open framework for translating portable applications into operating system-specific wireless sensor networks applications, Software: Practice and Experience 43 (3) (2013) 333–357. doi:10.1002/spe.2114.

[276] S. Chalotra, S. K. Sehra, S. S. Sehra, An analytical review of nature inspired optimization algorithms, International Journal of Science Technology & Engineering 2 (2) (2015) 123–126. doi:10.17148/ijarcce.2015.4363.

[277] F. S. Gharehchopogh, R. Rezaii, B. Arasteh, A new approach by using tabu search and genetic algorithms in software cost estimation, in: 9th International Conference on Application of Information and Communication Technologies, Rostov-on-Don, Russia, 2015, pp. 113–117. doi:10.1109/ICAICT.2015.7338528.

[278] R. Gupta, N. Chaudhary, S. K. Pal, Hybrid model to improve bat algorithm performance, in: International Conference on Advances in Computing, Communications and Informatics, New Delhi, India, 2014, pp. 1967–1970. doi:10.1109/ICACCI.2014.6968649.

[279] S. Chalotra, S. Sehra, Y. Brar, N. Kaur, Tuning of cocomo model parameters by using bee colony optimization, Indian Journal of Science and Technology 8 (14) (2015) 1–5. doi:10.17485/ijst/2015/v8i14/70010.

[280] T. K. Sharma, M. Pant, Halton based initial distribution in artificial bee colony algorithm and its application in software effort estimation, International Journal of Natural Computing Research 3 (2) (2012) 86–106. doi:10.4018/jncr.2012040105.

[281] G. Wang, G. J. Roedler, R. Valerdi, M. Pena, Quantifying systems engineering reuse – a generalized reuse framework in cosysmo, in: INCOSE International Symposium, Vol. 23, Philadelphia, Pennsylvania, 2013, pp. 1215–1233. doi:10.1002/j.2334-5837.2013.tb03082.x.

[282] H. Suelmann, Putnam's effort-duration trade-off law: Is the software estimation problem really solved?, in: Joint conference of the 24th International Workshop on Software Measurement (IWSM) and the 9th International Conference on Software Process and Product Measurement (Mensura), Rotterdam, the Netherlands, 2014, pp. 79–84. doi:10.1109/IWSM.Mensura.2014.25.

[283] G. Wang, G. J. Roedler, M. Pena, R. Valerdi, A generalized systems engineering reuse framework and its cost estimating relationship, in: INCOSE International Symposium, Vol. 24, Las Vegas, USA, 2014, pp. 274–297. doi:10.1002/j.2334-5837.2014.tb03149.x.

[284] J. C. C. P. Mascena, E. S. de Almeida, S. R. de Lemos Meira, A comparative study on software reuse metrics and economic models from a traceability perspective, in: IEEE International Conference on Information Reuse and Integration, Las Vegas,USA, 2005, pp. 72–77. doi:10.1109/IRI-05.2005.1506452.

[285] L. H. Putnam, W. Myers, How solved is the cost estimation problem?, IEEE Software 14 (6) (1997) 105–107. doi:10.1109/52.636696.

[286] J. Arobaa, J. J. Cuadrado-Gallegob, M. Ángel Siciliab, I. Ramosc, E. García-Barriocanalb, Segmented software cost estimation models based on fuzzy clustering, Journal of Systems and Software 81 (11) (2008) 1944 – 1950. doi:10.1016/j.jss.2008.01.016.

[287] M. Garre, M. Sicilia, J. J. Cuadrado, M. Charro, Regression Analisys of Segmented Parametric Software Cost Estimation Models Using Recursive Clustering Tool, Vol. 4224 of Lecture Notes in Computer Science, Springer, 2006, Ch. Intelligent Data Engineering and Automated Learning, pp. 849–858. doi:10.1007/11875581_102.

[288] J. J. Cuadrado-Gallego, M. Ángel Sicilia, M. Garre, D. Rodríguez, An empirical study of process-related attributes in segmented software cost-estimation relationships, Journal of Systems and Software 79 (3) (2006) 353 – 361. doi:10.1016/j.jss.2005.04.040.

[289] J. J. Cuadrado-Gallego, L. F. Sanz, M. Sicilia, Enhancing input value selection in parametric software cost estimation models through second level cost drivers, Software Quality Journal 14 (4) (2006) 339–357. doi:10.1007/s11219-006-0039-0.

[290] S. Malathi, S. Sridhar, Effort Estimation in Software Cost Using Team Characteristics Based on Fuzzy Analogy Method - A Diverse Approach, Springer, 2012, Ch. Signal Processing and Information Technology, pp. 1–8. doi:10.1007/978-3-319-11629-7_1.

[291] O. Mizuno, T. Adachi, T. Kikuno, Y. Takagi, On prediction of cost and duration for risky software projects based on risk questionnaire, in: Second Asia-Pacific Conference on Quality Software, Hong Kong, 2001, pp. 120–128. doi:10.1109/APAQS.2001.990010.

[292] B. Seetharaman, Z. Mansor, The development of agile cost management tool, in: International Conference on Electrical Engineering and Informatics, Bali,Indonesia, 2015, pp. 400–404. doi:10.1109/ICEEI.2015.7352534.

[293] R. Popli, N. Chauhan, Agile estimation using people and project related factors, in: International Conference on Computing for Sustainable Global Development, New Delhi,India, 2014, pp. 564–569. doi:10.1109/IndiaCom.2014.6828023.

[294] X. Zhu, B. Zhou, L. Hou, J. Chen, L. Chen, An experience-based approach for test execution effort estimation, in: The 9th International Conference for Young Computer Scientists, Hunan, China, 2008, pp. 1193–1198. doi:10.1109/ICYCS.2008.53.

[295] D. S. Kushwaha, A. K. Misra, Software test effort estimation, ACM SIGSOFT Software Engineering Notes 33 (3). doi:10.1145/1360602.1361211.

[296] E. Aranha, P. Borba, Test effort estimation models based on test specifications, in: Testing: Academic & Industrial Conference Practice And Research Techniques, Windsor, UK, 2007, pp. 67–71. doi:10.1109/TAIC.PART.2007.29.

[297] Q. Yi, Z. Bo, Z. Xiaochun, Early estimate the size of test suites from use cases, in: 5th Asia-Pacific Software Engineering Conference, Beijing,China, 2008, pp. 487–492. doi:10.1109/APSEC.2008.62.

[298] K. Molokken-Ostvold, M. Jorgensen, A comparison of software project overruns - flexible versus sequential development models, IEEE Transactions on Software Engineering 31 (9) (2005) 754–766. doi:10.1109/TSE.2005.96.

[299] L. Chwif, J. Banks, M. R. P. Barretto, Estimating the implementation time for discrete-event simulation model building, in: Winter Simulation Conference, Baltimore,USA, 2010, pp. 1774–1785. doi:

10.1109/WSC.2010.5678891.

[300] D. Luzeaux, Conclusion: What Return on Investment Can We Expect from Simulation?, John Wiley & Sons, Inc., 2013, Ch. Simulation and Modeling of Systems of Systems, pp. 355–372. doi:10.1002/9781118616727.ch9.

[301] G. K. Mislick, D. A. Nussbaum, Epilogue: The Field of Cost Estimating and Analysis, John Wiley & Sons, Inc, 2015, Ch. Cost Estimation: Methods and Tools, pp. 291–294. doi:10.1002/9781118802342.ch17.

[302] P. Clarke, R. V. O'Connor, The situational factors that affect the software development process: Towards a comprehensive reference framework, Information and Software Technology 54 (5) (2012) 433 – 447. doi:10.1016/j.infsof.2011.12.003.

[303] R. Jeffery, M. Ruhe, I. Wieczorek, A comparative study of two software development cost modeling techniques using multi-organizational and company-specific data, Information and Software Technology 42 (14) (2000) 1009 – 1016. doi:10.1016/S0950-5849(00)00153-1.

[304] B. A. Kitchenham, E. Mendes, G. H. Travassos, Cross versus within-company cost estimation studies: A systematic review, IEEE Transactions in Software Engineering 33 (5) (2007) 316–329. doi:10.1109/TSE.2007.1001.

[305] H. K. Verma, V. Sharma, Handling imprecision in inputs using fuzzy logic to predict effort in software development, in: IEEE 2nd International on Advance Computing Conference, Patiala, India, 2010, pp. 436–442. doi:10.1109/IADCC.2010.5422889.

[306] V. Sharma, H. K. Verma, Optimized fuzzy logic based framework for effort estimation in software development, in: Software Development, Computer Science Issues, Vol. 7, 2010, pp. 30–38.
URL https://arxiv.org/ftp/arxiv/papers/1004/1004.3270.pdf

[307] M. A. Ahmed, Z. Muzaffar, Handling imprecision and uncertainty in software development effort prediction: A type-2 fuzzy logic based framework, Information and Software Technology 51 (3) (2009) 640 – 654. doi:10.1016/j.infsof.2008.09.004.

[308] M. O. Saliu, M. Ahmed, J. AlGhamdi, Towards adaptive soft computing based software effort prediction, in: IEEE Annual Meeting of the Fuzzy Information Procesing, Vol. 1, Banff, Canada, 2004, pp. 16–21. doi:10.1109/NAFIPS.2004.1336241.

[309] O. Jalali, T. Menzies, D. Baker, J. Hihn, Column pruning beats stratification in effort estimation, in: Third International Workshop on Predictor Models in Software Engineering, Minneapolis, Minnesota, 2007, pp. 7–7. doi:10.1109/PROMISE.2007.3.

[310] J. Li, G. Ruhe, Analysis of attribute weighting heuristics for analogy-based software effort estimation method aqua+, Empirical Software Engineering 13 (1) (2008) 63–96. doi:10.1007/s10664-007-9054-4.

[311] E. Kocaguneli, T. Menzies, How to find relevant data for effort estimation?, in: International Symposium on Empirical Software Engineering and Measurement, Banff, Canada, 2011, pp. 255–264. doi:10.1109/ESEM.2011.34.

[312] J. Li, G. Ruhe, Software effort estimation by analogy using attribute selection based on rough set analysis, International Journal of Software Engineering and Knowledge Engineering 18 (1) (2008) 1–23. doi:10.1142/S0218194008003532.

[313] K. Moløkken-Østvold, N. C. Haugen, H. C. Benestad, Using planning poker for combining expert estimates in software projects, Journal of Systems and Software 81 (12) (2008) 2106 – 2117. doi:10.1016/j.jss.2008.03.058.

[314] Z. Li, L. O'Brien, H. Zhang, Circumstantial-evidence-based judgment for software effort estimation, in: 1st International Workshop on Evidential Assessment of Software Technologies, Beijing, China, 2011, pp. 18–27.
URL https://arxiv.org/ftp/arxiv/papers/1302/1302.2193.pdf

[315] K. Moløkken, Software Effort Estimation: Planning XP Guidelines Compared to Research on Traditional Software Development, Springer, 2003, Ch. Extreme Programming and Agile Processes in Software Engineering, pp. 441–442. doi:10.1007/3-540-44870-5_77.

[316] M. Jørgensen, Practical guidelines for expert-judgment-based software effort estimation, IEEE Software 22 (3) (2005) 57–63. doi:10.1109/MS.2005.73.

[317] R. T. Hughes, Expert judgement as an estimating method, Information and Software Technology 38 (2) (1996) 67 – 75. doi:10.1016/0950-5849(95)01045-9.

[318] M. Jørgensen, The effect of the time unit on software development effort estimates, in: 9th International Conference on Software, Knowledge, Information Management and Applications, Kathmandu, Nepal, 2015, pp. 1–5. doi:10.1109/SKIMA.2015.7399992.

[319] D. Basten, W. Mellis, A current assessment of software development effort estimation, in: International Symposium on Empirical Software Engineering and Measurement, Banff, Canada, 2011, pp. 235–244. doi:10.1109/ESEM.2011.32.

[320] M. Jørgensen, Contrasting ideal and realistic conditions as a means to improve judgment-based software development effort estimation, Information and Software Technology 53 (12) (2011) 1382–1390. doi:10.1016/j.infsof.2011.07.001.

[321] E. Kocaguneli, T. Menzies, J. W. Keung, Kernel methods for software effort estimation - effects of different kernel functions and bandwidths on estimation accuracy, Empirical Software Engineering 18 (1) (2013) 1–24. doi:10.1007/s10664-011-9189-1.

[322] C. Hsu, C. Huang, Improving effort estimation accuracy by weighted grey relational analysis during software development, in: 14th Asia-Pacific Software Engineering Conference, Nagoya,Japan, 2007, pp. 534–541. doi:10.1109/ASPEC.2007.62.

[323] C. Hsu, C. Huang, Comparison of weighted grey relational analysis for software effort estimation, Software Quality Journal 19 (1) (2011) 165–200. doi:10.1007/s11219-010-9110-y.

[324] J. Wen, S. Li, L. Tang, Improve analogy-based software effort estimation using principal components analysis and correlation weighting, in: 16th Asia-Pacific Software Engineering Conference, Penang, Malaysia, 2009, pp. 179–186. doi:10.1109/APSEC.2009.40.

[325] M. C. Ohlsson, C. Wohlin, B. Regnell, A project effort estimation study, Information and Software Technology 40 (14) (1998) 831–839. doi:10.1016/S0950-5849(98)00097-4.

[326] J. J. Ahonen, P. Savolainen, H. Merikoski, J. Nevalainen, Reported project management effort, project size, and contract type, Journal of Systems and Software 109 (2015) 205 – 213. doi:10.1016/j.jss.2015.08.008.

[327] E. Capra, C. Francalanci, F. Merlo, An empirical study on the relationship between software design quality, development effort and governance in open source projects, IEEE Transactions on Software Engineering 34 (6) (2008) 765–782. doi:10.1109/TSE.2008.68.

[328] L. M. Alves, S. Oliveira, P. Ribeiro, R. J. Machado, An empirical study on the estimation of size and complexity of software applications with function points analysis, in: 14th International Conference on Computational Science and Its Applications, Guimaraes, Portugal, 2014, pp. 27–34. doi:10.1109/ICCSA.2014.17.

[329] E. H. Nathanael, B. Hendradjaya, W. D. Sunindyo, Study of algorithmic method and model for effort estimation in big data software development case study: Geodatabase, in: International Conference on Electrical Engineering and Informatics, Bali,Indonesia, 2015, pp. 427–432. doi:10.1109/ICEEI.2015.7352539.

[330] M. Jørgensen, Selection of strategies in judgment-based effort estimation, Journal of Systems and Software 83 (6) (2010) 1039–1050. doi:10.1016/j.jss.2009.12.028.

[331] M. Jorgensen, T. Halkjelsvik, The effects of request formats on judgment-based effort estimation, Journal of Systems and Software 83 (1) (2010) 29 – 36. doi:10.1016/j.jss.2009.03.076.

[332] M. Jørgensen, G. J. Carelius, An empirical study of software project bidding, IEEE Transactions on Software Engineering 30 (12) (2004) 953–969. doi:10.1109/TSE.2004.92.

[333] B. J. Osteen, K. Jegannathan, S. Ramanan, Optimizing the cost of software quality - a path to delivery excellence, in: Tenth International Conference on Information Technology: New Generations, 2013, pp. 754–756. doi:10.1109/ITNG.2013.118.

[334] C. Smartt, S. Ferreira, Advancing systems engineering in support of the bid and proposal process, Systems Engineering 14 (3) (2011) 255–266. doi:10.1002/sys.20177.

[335] N. Zhou, Q. Ma, K. Ratakonda, Quantitative modeling of communication cost for global service delivery, in: IEEE International Conference on Services Computing, Banglore, India, 2009, pp. 388–395. doi:10.1109/SCC.2009.79.

[336] A. Lamersdorf, J. Munch, A. F. d. V. Torre, C. R. Sanchez, D. Rombach, Estimating the effort overhead in global software development, in: 5th IEEE International Conference on Global Software Engineering, Princeton, USA, 2010, pp. 267–276. doi:10.1109/ICGSE.2010.38.

[337] M. Uzzafer, A pitfall of estimated software cost, in: The 2nd IEEE International Conference on Information Management and Engineering, Chengdu, China, 2010, pp. 578–582. doi:10.1109/ICIME.2010.5478225.

[338] C. Jones, The economics of software process improvement, Computer 29 (1) (1996) 95–97. doi:10.1109/2.481498.

[339] B. K. Clark, Cost modeling process maturity-cocomo 2.0, in: IEEE Aerospace Applications Conference, Vol. 3, Manhattan,USA, 1996, pp. 347–369 vol.3. doi:10.1109/AERO.1996.496074.

[340] H. Erdogmus, A cost effectiveness indicator for software development, in: First International Symposium on Empirical Software Engineering and Measurement, Madrid, Spain, 2007, pp. 446–448. doi:10.1109/ESEM.2007.47.

[341] J. Keung, D. R. Jeffery, B. Kitchenham, The challenge of introducing a new software cost estimation technology into a small software organisation, in: Australian Software Engineering Conference, Melbourne, Australia, 2004, pp. 52–59. doi:10.1109/ASWEC.2004.1290457.

[342] H. C. Yeoh, J. Miller, Cots acquisition process: incorporating business factors in cots vendor evaluation taxonomy, in: 10th International Symposium on Software Metrics, 2004, pp. 84–95. doi:10.1109/METRIC.2004.1357893.

[343] L. Lavazza, S. Morasca, Software effort estimation with a generalized robust linear regression technique, in: 16th International Conference on Evaluation and Assessment in Software Engineering, 2012, pp. 206–215. doi:10.1049/ic.2012.0027.

[344] N. Mittas, L. Angelis, Combining regression and estimation by analogy in a semi-parametric model for software cost estimation, in: Second ACM-IEEE international symposium on Empirical software engineering and measurement, Kaiserslautern, Germany, 2008, pp. 70–79. doi:10.1145/1414004.1414017.

[345] N. Mittas, L. Angelis, A permutation test based on regression error characteristic curves for software cost estimation models, Empirical Software Engineering 17 (1-2) (2012) 34–61. doi:10.1007/s10664-011-9177-5.

[346] W. M. Evanco, Prediction models for software fault correction effort, in: Fifth European Conference on Software Maintenance and Reengineering, Lisbon, Portugal, 2001, pp. 114–120. doi:10.1109/.2001.914975.

[347] Y. Yang, Z. He, K. Mao, Q. Li, V. Nguyen, B. W. Boehm, R. Valerdi, Analyzing and handling local bias for calibrating parametric cost estimation models, Information and Software Technology 55 (8) (2013) 1496–1511. doi:10.1016/j.infsof.2013.03.002.

[348] P. L. Braga, A. L. I. Oliveira, G. H. T. Ribeiro, S. R. L. Meira, Bagging predictors for estimation of software project effort, in: International Joint Conference on Neural Networks, Florida, USA, 2007, pp. 1595–1600. doi:10.1109/IJCNN.2007.4371196.

[349] A. Corazza, S. D. Martino, F. Ferrucci, C. Gravino, E. Mendes, Investigating the use of support vector regression for web effort estimation, Empirical Software Engineering 16 (2) (2011) 211–243. doi:10.1007/s10664-010-9138-4.

[350] A. Corazza, S. D. Martino, F. Ferrucci, C. Gravino, F. Sarro, E. Mendes, Using tabu search to configure support vector regression for effort estimation, Empirical Software Engineering 18 (3) (2013) 506–546. doi:10.1007/s10664-011-9187-3.

[351] D. Port, M. Korte, Comparative studies of the model evaluation criterions mmre and pred in software cost estimation research, in: Second ACM-IEEE international symposium on Empirical software engineering and measurement, Kaiserslautern, Germany, 2008, pp. 51–60. doi:10.1145/1414004.1414015.

[352] V. Anandhi, R. M. Chezian, Regression techniques

in software effort estimation using cocomo dataset, in: International Conference on Intelligent Computing Applications, Coimbatore, India, 2014, pp. 353–357. doi:10.1109/ICICA.2014.79.

[353] Y. Jin, J. Li, J. Lin, Q. Chen, Software project cost estimation based on groupware, in: WRI World Congress on Software Engineering, Vol. 2, Xiamen,China, 2009, pp. 437–441. doi:10.1109/WCSE.2009.268.

[354] N. A. S. Abdullah, N. I. A. Rusli, M. F. Ibrahim, Mobile game size estimation: Cosmic fsm rules, uml mapping model and unity3d game engine, in: 2014 IEEE Conference on Open Systems, Subang Jaya, Malaysia, 2014, pp. 42–47. doi:10.1109/ICOS.2014.7042407.

[355] A. Z. Abualkishik, M. H. Selamat, A. A. A. Ghani, R. Atan, J. M. Desharnais, A. Khelifi, Theoretical and probabilistic conversion model between fpa and cosmic measurement method, in: Joint Conference of the 22nd International Workshop on Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement, Assisi, Italy, 2012, pp. 150–159. doi:10.1109/IWSM-MENSURA.2012.31.

[356] S. Patel, Function point distribution using maximum entropy principle, in: IEEE Second International Conference on Image Information Processing, 2013, pp. 684–689. doi:10.1109/ICIIP.2013.6707682.

[357] T. Fetcke, A. Abran, T.-H. Nguyen, Mapping the oo-jacobson approach into function point analysis, in: Technology of Object-Oriented Languages and Systems TOOLS 23, Santa Barbara, California, 1997, pp. 192–202. doi:10.1109/TOOLS.1997.654721.

[358] D. Azhar, P. Riddle, E. Mendes, N. Mittas, L. Angelis, Using ensembles for web effort estimation, in: ACM / IEEE International Symposium on Empirical Software Engineering and Measurement, Baltimore,USA, 2013, pp. 173–182. doi:10.1109/ESEM.2013.25.

[359] A. Idri, I. Abnane, A. Abran, Missing data techniques in analogy-based software development effort estimation, Journal of Systems and Software 117 (2016) 595 – 611. doi:10.1016/j.jss.2016.04.058.

[360] B. W. Boehm, C. Abts, S. Chulani, Software development cost estimation approaches - A survey, Annals of Software Engineering 10 (2000) 177–205. doi:10.1023/A:1018991717352.

[361] L. L. Minku, X. Yao, Software effort estimation as a multiobjective learning problem, ACM Transactions on Software Engineering and Methodology 22 (4) (2013) 35. doi:10.1145/2522920.2522928.

[362] M. A. Yahya, R. Ahmad, S. P. Lee, Effects of software process maturity on cocomo ii effort estimation from cmmi perspective, in: IEEE International Conference on Research, Innovation and Vision for the Future, Ho Chi Minh City, Vietnam, 2008, pp. 255–262. doi:10.1109/RIVF.2008.4586364.

[363] J. F. Peters, S. Ramanna, Application of the choquet integral in software cost estimation, in: Fifth IEEE International Conference on Fuzzy Systems, Vol. 2, New Orleans, USA, 1996, pp. 862–866. doi:10.1109/FUZZY.1996.552292.

[364] M. Agrawal, K. Chari, Software effort, quality, and cycle time: A study of cmm level 5 projects, IEEE Transactions on Software Engineering 33 (3) (2007) 145–156. doi:10.1109/TSE.2007.29.

[365] M. Ruiz, I. Ramos, M. Toro, An integrated framework for simulation-based software process improvement, Software Process: Improvement and Practice 9 (2) (2004) 81–93. doi:10.1002/spip.198.

[366] S. Amasaki, C. Lokan, The effect of moving windows on software effort estimation: Comparative study with cart, in: 6th International Workshop on Empirical Software Engineering in Practice, Osaka,Japan, 2014, pp. 1–6. doi:10.1109/IWESEP.2014.10.

[367] E. Lohre, M. Jorgensen, Numerical anchors and their strong effects on software development effort estimates, Journal of Systems and Software 116 (2016) 49 – 56. doi:10.1016/j.jss.2015.03.015.

[368] B. G. Tabachnick, L. S. Fidell, Principal Compnents and Factor Analysis, 5th Edition, Allyn & Bacon, 2006, Ch. Using Multivariate Statistics, pp. 612–680. URL http://baunne.unne.edu.ar/documentos/EstadisticaMultivariable.pdf

[369] L. Lavazza, Automated function points: Critical evaluation and discussion, in: Sixth International Workshop on Emerging Trends in Software Metrics, Florence, Italy, 2015, pp. 35–43. doi:10.1109/WETSoM.2015.13.

[370] D. J. Reifer, Web development: estimating quick-to-market software, IEEE Software 17 (6) (2000) 57–64. doi:10.1109/52.895169.

[371] M. Ruhe, R. Jeffery, I. Wieczorek, Using web objects for estimating software development effort for web applications, in: Ninth International Software Metrics Symposium, Sydney, Australia, 2003, pp. 30–37. doi:10.1109/METRIC.2003.1232453.

[372] E. Mendes, N. Mosley, S. Counsell, Investigating web size metrics for early web cost estimation, Journal of Systems and Software 77 (2) (2005) 157–172. doi:10.1016/j.jss.2004.08.034.

[373] S. M. Reza, M. M. Rahman, M. H. Parvez, M. S. Kaiser, S. A. Mamun, Innovative approach in web application effort amp; cost estimation using functional measurement type, in: International Conference on Electrical Engineering and Information Communication Technology, Dhaka, Bangladesh, 2015, pp. 1–7. doi:10.1109/ICEEICT.2015.7307462.

[374] E. Mendes, N. Mosley, S. Counsell, The application of case-based reasoning to early web project cost estimation, in: 26th Annual International Computer Software and Applications Conference, Oxford, UK, 2002, pp. 393–398. doi:10.1109/CMPSAC.2002.1045034.

[375] A. Idri, A. Zakrani, M. Elkoutbi, A. Abran, Fuzzy radial basis function neural networks for web applications cost estimation, in: 4th International Conference on Innovations in Information Technology, Dubai, 2007, pp. 576–580. doi:10.1109/IIT.2007.4430367.

[376] J. Wong, D. Ho, L. F. Capretz, An investigation of using neuro-fuzzy with software size estimation, in: ICSE Workshop on Software Quality, Vancouver, Canada, 2009, pp. 51–58. doi:10.1109/WOSQ.2009.5071557.

[377] J. Ryder, Fuzzy modeling of software effort prediction, in: Information Technology Conference, Syracuse, USA, 1998, pp. 53–56. doi:10.1109/IT.1998.713380.

[378] X. Dasheng, H. Shenglan, Estimation of project costs based on fuzzy neural network, in: World Congress on Information and Communication Technologies, Trivandrum, India, 2012, pp. 1177–1181. doi:10.1109/WICT.2012.6409253.

[379] A. Idri, T. M. Khoshgoftaar, A. Abran, Can neural networks be easily interpreted in software cost esti-

mation?, in: IEEE International Conference on Fuzzy Systems, Vol. 2, Honolulu, Hawaii, 2002, pp. 1162–1167. doi:10.1109/FUZZ.2002.1006668.

[380] A. F. Sheta, D. Rine, S. Kassaymeh, Software effort and function points estimation models based radial basis function and feedforward artificial neural networks, International Journal of Next Generation Computing 6 (3). URL https://www.researchgate.net/publication/285232770_Software_Effort_and_Function_Points_Estimation_Models_Based_Radial_Basis_Function_and_Feedforward_Artificial_Neural_Networks

[381] A. L. Oliveira, P. L. Braga, R. M. Lima, 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 52 (11) (2010) 1155 – 1166, special Section on Best Papers {PROMISE} 2009. doi:10.1016/j.infsof.2010.05.009.

[382] D. Milios, I. Stamelos, C. Chatzibagias, A genetic algorithm approach to global optimization of software cost estimation by analogy, Intelligent Decision Technologies 7 (1) (2013) 45–58. doi:10.3233/idt-120150.

[383] Q. Liu, S. Shi, H. Zhu, J. Xiao, A mutual information-based hybrid feature selection method for software cost estimation using feature clustering, in: 38th Annual IEEE Computer Software and Applications Conference, VÄSTERÅS, SWEDEN, 2014, pp. 27–32. doi:10.1109/COMPSAC.2014.99.

[384] N. Tadayon, Neural network approach for software cost estimation, in: International Conference on Information Technology: Coding and Computing, Las Vegas, USA, 2005, pp. 815–818. doi:10.1109/itcc.2005.210.

[385] I. Stamelos, L. Angelis, P. Dimou, E. Sakellaris, On the use of bayesian belief networks for the prediction of software productivity, Information and Software Technology 45 (1) (2003) 51 – 60. doi:10.1016/S0950-5849(02)00163-5.

[386] J. Hihn, K. T. Lum, Improving software size estimates by using probabilistic pairwise comparison matrices, in: 10th International Symposium on Software Metrics, Chicago, USA, 2004, pp. 140–150. doi:10.1109/METRIC.2004.1357898.

[387] M. Tsunoda, A. Monden, J. W. Keung, K. Matsumoto, Incorporating expert judgment into regression models of software effort estimation, in: 19th Asia-Pacific Software Engineering Conference, Hong Kong, 2012, pp. 374–379. doi:10.1109/apsec.2012.58.

[388] C. Lokan, E. Mendes, Investigating the use of duration-based moving windows to improve software effort prediction, in: 19th Asia-Pacific Software Engineering Conference, Vol. 1, Hong Kong, 2012, pp. 818–827. doi:10.1109/APSEC.2012.74.

[389] I. Wieczorek, M. Ruhe, How valuable is company-specific data compared to multi-company data for software cost estimation?, in: Eighth IEEE Symposium on Software Metrics, Ottawa, Canada, 2002, pp. 237–246. doi:10.1109/METRIC.2002.1011342.

[390] M. Auer, S. Biffl, Increasing the accuracy and reliability of analogy-based cost estimation with extensive project feature dimension weighting, in: International Symposium on Empirical Software Engineering, Redondo Beach, USA, 2004, pp. 147–155. doi:10.1109/ISESE.2004.1334902.

[391] V. K. Bardsiri, E. Khatibi, Insightful analogy-based software development effort estimation through selective classification and localization, Innovations in Systems and Software Engineering 11 (1) (2015) 25–38. doi:10.1007/s11334-014-0242-2.

[392] J. Li, A. Al-Emran, G. Ruhe, Impact analysis of missing values on the prediction accuracy of analogy-based software effort estimation method AQUA, in: First International Symposium on Empirical Software Engineering and Measurement, Madrid, Spain, 2007, pp. 126–135. doi:10.1109/esem.2007.10.

[393] J. Li, G. Ruhe, Decision support analysis for software effort estimation by analogy, in: International Workshop on Predictor Models in Software Engineering, Minneapolis, 2007, pp. 6–6. doi:10.1109/PROMISE.2007.5.

[394] A. Idri, A. Zahi, Software cost estimation by classical and fuzzy analogy for web hypermedia applications: A replicated study, in: IEEE Symposium on Computational Intelligence and Data Mining, IEEE, Singapore, 2013, pp. 207–213. doi:10.1109/cidm.2013.6597238.

[395] S. Malathi, S. Sridhar, Optimization of fuzzy analogy in software cost estimation using linguistic variables, Procedia Engineeringdoi:10.1016/j.proeng.2012.06.025.

[396] A. A.-E. M. M. R. Jingzhou Li, Guenther Ruhe, A flexible method for software effort estimation by analogy, Empirical Software Engineering 12 (1) (2007) 65–106. doi:10.1007/s10664-006-7552-4.

[397] J. Keung, B. Kitchenham, Optimising project feature weights for analogy-based software cost estimation using the mantel correlation, in: 14th Asia-Pacific Software Engineering Conference, Nagoya, Japan, 2007, pp. 222–229. doi:10.1109/apsec.2007.73.

[398] R. Valerdi, Heuristics for systems engineering cost estimation, IEEE Systems Journal 5 (1) (2011) 91–98. doi:10.1109/JSYST.2010.2065131.

[399] U. Raja, D. P. Hale, J. E. Hale, Modeling software evolution defects: a time series approach, Journal of Software Maintenance and Evolution: Research and Practice 21 (1) (2009) 49–71. doi:10.1002/smr.398.

[400] E. Kocaguneli, T. Menzies, J. W. Keung, On the value of ensemble effort estimation, IEEE Transactions on Software Engineering 38 (6) (2012) 1403–1416. doi:10.1109/tse.2011.111.

[401] L. Minku, F. Sarro, E. Mendes, F. Ferrucci, How to make best use of cross-company data for web effort estimation?, in: 2015 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), 2015, pp. 1–10. doi:10.1109/ESEM.2015.7321199.

[402] X. Huang, D. Ho, J. Ren, L. F. Capretz, Improving the {COCOMO} model using a neuro-fuzzy approach, Applied Soft Computing 7 (1) (2007) 29 – 40. doi:10.1016/j.asoc.2005.06.007.

[403] D. Wu, J. Li, Y. Liang, Linear combination of multiple case-based reasoning with optimized weight for software effort estimation, The Journal of Supercomputing 64 (3) (2013) 898–918. doi:10.1007/s11227-010-0525-9.

[404] E. Mendes, N. Mosley, I. Watson, A comparison of case-based reasoning approaches, in: 11th international conference on World Wide Web, Hawai, USA, 2002, pp. 272–280. doi:10.1145/511479.511482.

[405] P. L. Braga, A. L. I. Oliveira, S. R. L. Meira, A

ga-based feature selection and parameters optimization for support vector regression applied to software effort estimation, in: ACM symposium on Applied computing, Ceara, Brazil, 2008, pp. 1788–1792. doi:10.1145/1363686.1364116.

[406] A. Kaushik, A. K. Soni, R. Soni, A hybrid approach for software cost estimation using polynomial neural networks and intuitionistic fuzzy sets, International Journal of Computer Applications in Technology 52 (4) (2015) 292–304. doi:10.1504/ijcat.2015.073596.

[407] A. Idri, A. Zakrani, A. Abran, Functional equivalence between radial basis function neural networks and fuzzy analogy in software cost estimation, in: 3rd International Conference on Information and Communication Technologies: From Theory to Applications, Damascus, Syria, 2008, pp. 1–5. doi:10.1109/ICTTA.2008.4530015.

[408] M. Algabri, F. Saeed, H. Mathkour, N. Tagoug, Optimization of soft cost estimation using genetic algorithm for nasa software projects, in: 5th National Symposium on Information Technology: Towards New Smart World, Riyadh, Saudi Arabia, 2015, pp. 1–4. doi:10.1109/NSITNSW.2015.7176416.

[409] S. Huang, N. Chiu, Optimization of analogy weights by genetic algorithm for software effort estimation, Information and Software Technology 48 (11) (2006) 1034–1045. doi:10.1016/j.infsof.2005.12.020.

[410] M. A. Ahmed, M. O. Saliu, J. AlGhamdi, Adaptive fuzzy logic-based framework for software development effort prediction, Information and Software Technology 47 (1) (2005) 31 – 48. doi:10.1016/j.infsof.2004.05.004.

[411] M. Jørgensen, Selection of strategies in judgment-based effort estimation, Journal of Systems and Software 83 (6) (2010) 1039 – 1050, software Architecture and Mobility. doi:10.1016/j.jss.2009.12.028.

[412] M. Jorgensen, E. Lohre, First impressions in software development effort estimation: Easy to create and difficult to neutralize, in: 16th International Conference on Evaluation & Assessment in Software Engineering, Spain, 2012, pp. 216–222. doi:10.1049/ic.2012.0028.

[413] Y. Kultur, B. Turhan, A. Bener, Ensemble of neural networks with associative memory (enna) for estimating software development costs, Knowledge-Based Systems 22 (6) (2009) 395 – 402. doi:10.1016/j.knosys.2009.05.001.

[414] L. L. Minku, X. Yao, A principled evaluation of ensembles of learning machines for software effort estimation, in: 7th International Conference on Predictive Models in Software Engineering, ACM, Banff, Canada, 2011, p. 9. doi:10.1145/2020390.2020399.

[415] A. Issa, M. Odeh, D. Coward, Software cost estimation using use-case models: a critical evaluation, in: 2006 2nd International Conference on Information Communication Technologies, Vol. 2, 2006, pp. 2766–2771. doi:10.1109/ICTTA.2006.1684849.

[416] F. Wang, X. Yang, X. Zhu, L. Chen, Extended use case points method for software cost estimation, in: International Conference on Computational Intelligence and Software Engineering, Wuhan, China, 2009, pp. 1–5. doi:10.1109/CISE.2009.5364706.

[417] M. de Freitas Junior, M. Fantinato, V. Sun, Improvements to the function point analysis method: A systematic literature review, IEEE Transactions on Engineering Management 62 (4) (2015) 495–506. doi:10.1109/tem.2015.2453354.

[418] O. Morgenshtern, T. Raz, D. Dvir, Factors affecting duration and effort estimation errors in software development projects, Information & Software Technology 49 (8) (2007) 827–837. doi:10.1016/j.infsof.2006.09.006.

[419] A. Magazinius, S. Börjesson, R. Feldt, Investigating intentional distortions in software cost estimation - an exploratory study, Journal of Systems and Software 85 (8) (2012) 1770–1781. doi:10.1016/j.jss.2012.03.026.

[420] M. Shepperd, M. Cartwright, Predicting with sparse data, IEEE Transactions on Software Engineering 27 (11) (2001) 987–998. doi:10.1109/32.965339.

[421] F. A. Amazal, A. Idri, A. Abran, Software development effort estimation using classical and fuzzy analogy: a cross-validation comparative study, International Journal of Computational Intelligence and Applications 13 (3) (2014) 1–19. doi:10.1142/S1469026814500138.

[422] I. Myrtveit, E. Stensrud, M. Shepperd, Reliability and validity in comparative studies of software prediction models, IEEE Transactions on Software Engineering 31 (5) (2005) 380–391. doi:10.1109/TSE.2005.58.