Systematic Mapping Study of Ensemble Effort Estimation

Ali Idri[1], Mohamed Hosni[1] and Alain Abran[2]
[1]*Software Project Management Research Team, ENSIAS, Mohammed V University in Rabat, Rabat, Morocco*
[2]*Department of Software Engineering, ETS, Montréal H3C IK3, Canada*

Keywords: Systematic Mapping Study, Ensemble Effort Estimation, Software Development Effort Estimation.

Abstract: Ensemble methods have been used recently for prediction in data mining area in order to overcome the weaknesses of single estimation techniques. This approach consists on combining more than one single technique to predict a dependent variable and has attracted the attention of the software development effort estimation (SDEE) community. An ensemble effort estimation (EEE) technique combines several existing single/classical models. In this study, a systematic mapping study was carried out to identify the papers based on EEE techniques published in the period 2000-2015 and classified them according to five classification criteria: research type, research approach, EEE type, single models used to construct EEE techniques, and rule used the combine single estimates into an EEE technique. Publication channels and trends were also identified. Within the 16 studies selected, homogeneous EEE techniques were the most investigated. Furthermore, the machine learning single models were the most frequently employed to construct EEE techniques and two types of combiner (linear and non-linear) have been used to get the prediction value of an ensemble.

1 INTRODUCTION

Software development effort estimation (SDEE) is one of the most important challenges facing software project management (Wen et al. 2012). Over the past 35 years, software researchers have proposed a set of effort estimation techniques in order to produce an accurate estimation. In 2007, a systematic review (Jorgensen and Shepperd, 2007) identified 11 estimation methods that were used between 2000 and 2004: the dominant approach was the regression method with 49% in 304 selected studies. Recently the machine learning (ML) models has received increasing attention by software researchers in order to enhance the estimation accuracy (Elish et al., 2013). In 2012, the systematic review of ML based effort estimation techniques (Wen et al. 2012) identified eight ML techniques were identified, with case-based reasoning (CBR) and artificial neural networks (ANN) the most used techniques (investigated in 37% and 26% of 84 selected studies respectively).

Despite the large number of effort estimation techniques published since 1980s, none of them has been considered as the best model in all circumstances (Shepperd and Kadoda, 2001; Wen et al., 2012). The performance of these models varies from one dataset to another, which makes them unstable. Consequently, building an estimation model that provides a high and stable accuracy is needed. Within this context, a new approach namely Ensemble Effort Estimation (EEE) was proposed. It is defined as a combination of several single estimation techniques (called also base models) under a specific aggregation mechanism (Seni and Elder, 2010; Azzeh et al., 2015).

Figure 1 summarizes the EEE process. The estimation of an ensemble is given by the combination of the estimates of each base model that composes the ensemble. There are two types of EEE techniques (Elish et al., 2013):

(1) Homogeneous EEE: used to refer to an ensemble that combines one base model with at least two different configurations or a combination of one ensemble learning such as Bagging (Song et al., 2013), Negative Correlation or Random (Minku and Yao, 2013b) and one base model.

(2) Heterogeneous EEE: used to refer to an ensemble that combines at least two different base models.

In order to classify and analyze the state of art and provide an overview of the trends of EEE approaches,

Idri, A., Hosni, M. and Abran, A.
Systematic Mapping Study of Ensemble Effort Estimation.
In *Proceedings of the 11th International Conference on Evaluation of Novel Software Approaches to Software Engineering (ENASE 2016)*, pages 132-139
ISBN: 978-989-758-189-2

we conducted a systematic mapping of EEE techniques. A systematic mapping study is defined by Petersen et al. (2008) as a method in which a classification scheme is built and a field of interest structured. It provides a structure of the type of research reports and results that have been published by categorizing them. To the best of our knowledge, this paper is the first systematic mapping study that focuses on EEE techniques in SDEE, which motivates this work.

This systematic mapping study allowed us to discover which types of EEE techniques were most frequently used to predict software development effort, the single techniques used to construct the EEE techniques, and the combination rules most used to get the estimation of EEE technique. The research types and approaches that exist in literature were also identified. The results were analyzed, tabulated, and synthesized in order to provide a global picture of the trend of EEE techniques.

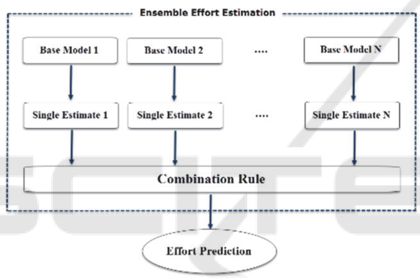

Figure 1: Ensemble Effort Estimation (EEE) process.

The remainder of this paper is organized as follows. Section 2 presents the research methodology. Section 3 presents the results obtained from the systematic mapping study. Section 4 discusses the main findings. Section 6 presents the conclusions and future work.

2 RESEARCH METHODOLOGY

This study has been organized as a systematic mapping study (SMS), based on the process suggested by Kitchenham and Charters (2007). According to Petersen et al. (2008), the main goal of a SMS is to provide an overview of a research area, and identify the quantity and type of research and results available within it. The mapping process involves five steps: (1) research questions, (2) search strategy, (3) study selection, (4) data extraction, and (5) data synthesis. The various steps of this review protocol are presented next.

2.1 Mapping Questions

Table 1 lists the seven mapping questions (MQs) that have been defined, along with their main motivations.

Table 1: Mapping questions (MQ).

ID	Mapping Questions	Main motivations
MQ1	Which publication channels are the main target for EEE techniques?	To identify where EEE papers can be found as well as the good targets for the publication of future papers.
MQ2	How has the frequency of EEE techniques changed over the time?	To identify the publication trends of EEE research in SDEE over time.
MQ3	In which research types are EEE techniques papers classified?	To explore the different types of research reported in EEE techniques in SDEE.
MQ4	What are the research approaches of the selected papers?	To discover the research approaches most investigated when evaluating EEE techniques.
MQ5	What are the most frequently investigated types of EEE techniques?	To discover the EEE techniques most investigated in SDEE.
MQ6	What are the most frequently single models used to construct EEE techniques?	To identify the most frequently single models used to construct EEE techniques.
MQ7	What are the combiner rules used to get the overall estimation of EEE techniques?	To gain knowledge about the combiner rules used to get the estimation effort of EEE techniques.

2.2 Search Strategy

The objective of the search strategy is to find the studies that will help us to address the MQs of Table 1. The primary studies were identified by performing a search using four digital libraries: (1) IEEE Xplore, (2) ACM Digital Library, (3) Science Direct, and (4) Google Scholar.

In order to establish the search string used to run the search in the four libraries, we derived major terms from the MQs of Table 1 and checked for their synonyms and alternative spellings (Idri et al., 2015; Wen et al., 2012). The complete set of search terms was formulated as follows:

Software AND (effort OR cost) AND (estimat* OR predict* OR assess*) AND (ensemble OR* *taxonomy OR multiple OR combin* OR cluster* OR classifiers) AND ("case based reasoning" OR "decision tree" OR "decision trees" OR "regression tree" OR "regression trees" OR "RTs" OR "RT" OR "classification tree" OR "classification trees" OR neural net* OR bayesian net* OR "linear regression" OR "support vector machine" OR "support vector machines" OR "support vector regression" OR "multilayer perceptron" OR "multilayer perceptrons" OR "MLPs" OR "MLP" OR "NN" OR nearest neighbors OR "Radial basis function" OR "RBF").*

The search process was carried out in two stages: (1) Run a separate search using the search string in each of the four databases and then gather a set of candidate papers. (2) the reference lists of the relevant papers (e.g. candidate papers that satisfy the inclusion criteria defined in Section 2.3) were examined in order to check if any papers related to EEE techniques were missed in stage 1, and add them (if found) to the set of candidate papers. The examination was based on title, abstract, and keywords. The full text of the papers was also examined when necessary. This stage ensured us that the search covered the maximum number of existing studies related to EEE techniques.

2.3 Study Selection

The aim of the selection process was to identify the articles that are the most relevant to the objective of this SMS. Each paper was assessed by two researchers independently, using the inclusion and the exclusion criteria. Each researcher categorised the papers as “included”, “excluded” or “uncertain”.

Inclusion criteria: (1) Studies using EEE techniques to estimate software development effort; (2) Studies that compare different EEE techniques or compare EEE techniques with other single techniques; (3) EEE studies using hybrid models to estimate development effort; (4) Duplicate publication of the same study, only the most complete and newest one will be included.

Exclusion criteria: (1) EEE studies for estimating maintenance or testing efforts; (2) EEE studies for estimating software size, schedule or duration only without estimating effort; (3) EEE studies addressing project control and management.

The paper was retained or rejected if was categorized as “Included” or “Excluded” respectively by both researchers. Papers that were judged differently were discussed by the two researchers until an agreement was found.

Figure 2 shows the number of papers retrieved in each step. First, the search in four electronic databases gave 358 candidate papers. In addition, 5 papers were added according to authors’ knowledge; those 5 papers weren’t retrieved by the automated search. This gave us 363 papers in total, including 36 duplicated papers. Second, we applied on the candidate papers the inclusion and exclusion criteria which provided us 14 relevant papers. There was no disagreement between the researchers in this stage. Third, we scanned the references list of the relevant papers and two extra papers were found (Wu et al., 2013; Vinaykumar, M.C.K, Ravi 2009). After that, we checked the references list of these two extra papers, but no additional relevant paper was found. Finally, 16 papers were selected. They are indicated by an (*) at the end of their citations in the References Section.

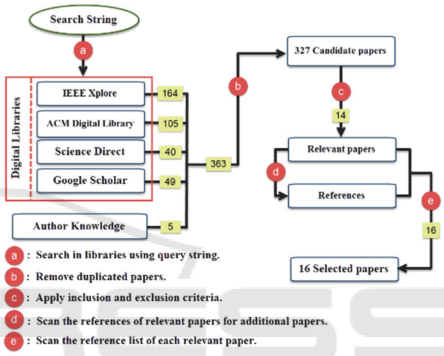

Figure 2: Search, Selection and QA process.

2.4 Data Extraction Strategy and Synthesis Method

The purpose of data extraction step is to extract all data that would address the MQs raised in this study. Table 2 presents the data extraction form used to collect all the information from the selected studies. The narrative synthesis was adopted in order to synthesize and to summarize the data relating to MQs. It consists on tabulating the data in a consistent

Table 2: Data extraction form.

Data extractor
Data checker
Study identifier
Author(s) name(s)
Article title
(MQ1) Publication source and channel
(MQ2) Publication year
(MQ3) Research type
(MQ4) Research approach
(MQ5) Type of EEE techniques
(MQ6) Single models used to construct EEE technique
(MQ7) Rule used to get the estimation of EEE technique

manner with the mapping questions. In order to improve the presentation of these findings we used some visualization tools such as bar charts and bubble plots.

3 RESULTS AND DISCUSSION

This section presents and discusses the findings related to the mapping questions (MQs) of Table 1.

3.1 Publications Channels (MQ1)

Table 3 lists all the resources, the different publication channels and the number of papers per publication source. Three publication channels were identified: Journal, Conference and Book. Among the 16 selected studies, 44% (7 papers) were published in journals, 50% (8 papers) were presented at conferences, and 6% (one paper) came from a chapter book. Table 3 shows the distribution of the selected studies across the publication sources. Note that, except for International Conference on Predictive Models in Software Engineering, no source (conference or journal) was used more than once to publish studies on EEE.

3.2 Publications Trends (MQ2)

Figure 3 shows the distribution of papers published per year from 2000 and 2015. Ensemble techniques haven't been investigated early in SDEE. In fact, the first paper was published in 2007 (Braga et al., 2007). Moreover, when analysing the papers' distribution over time (see Fig. 3), we found that the trends of EEE publications are characterized by discontinuity. Indeed, not a singler paper was published in 2008, 2011, and 2014. In 2013, the research topic has gained an increased attention by the publication of 7 papers (around 44%), but it decreased afterwards.

3.3 Research Types (MQ3) and Research Approaches (MQ4)

For the research approach, Figure 4 shows that all studies belong to the Solution Proposal approach: all papers investigated different EEE with different configurations and different experimental designs. Also, all papers were included in the Evaluation approach since they evaluated the solution they presented. Note that this study did not find any opinion study. Figure 4 shows also that all selected papers fall into the history-based type, since they all used historical datasets to evaluate their proposed EEE techniques.

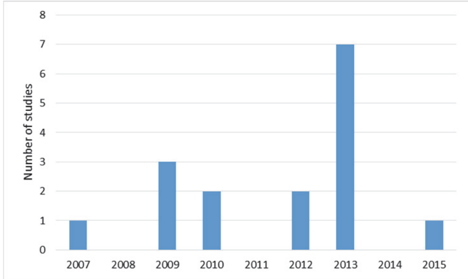

Figure 3: Publication per year.

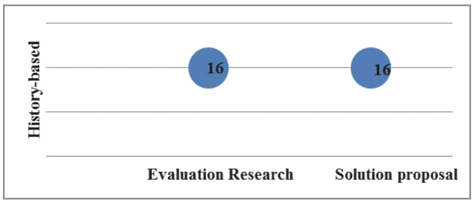

Figure 4: Research types and research approaches.

3.4 EEE Types (MQ5)

MQ5 reports the distribution of EEE types used in SDEE. Homogenous EEE techniques are the most frequently used: 12 of 16 selected studies (75%) presented homogenous EEE techniques with 15

Table 3: Publication venues.

P.Ch*	Publication venue (Number of studies)
Journal	ACM Transactions on Software Engineering and Methodology (1)
	Expert systems with applications (1)
	IEEE Transactions on Software Engineering (1)
	Mathematical Problems in Engineering (1)
	Information and Software Technology (1)
	The Journal of Systems and Software (1)
	The Journal of Supercomputing (1)
	Knowledge based systems (1)
Conference	IEEE International Joint Conference on Neural Networks (1)
	International Symposium on Empirical Software Engineering and Measurement (1)
	International Symposium on Software Reliability Engineering (1)
	International Computer Software and Applications Conference (1)
	IEEE Symposium on Computational Intelligence and Data Mining (1)
	International Conference on Predictive Models in Software Engineering (2)
Book	Handbook Of Research On Machine Learning Applications and Trends (1)

**Publication Channel*

homogenous combination types. Precisely, as shown in Table 4, three combinations of homogeneous EEE based on the combination of different configurations of a single model were proposed. Further, 12 combinations of homogeneous EEE based on a combination of ensemble machine learning and single model were proposed in the selected studies. In these 12 homogenous EEE, the bagging ensemble was the most used. As for the heterogeneous EEE, they were discussed in 7 papers with 9 heterogeneous combination types (see Table 5). Note that 3 of the selected studies (Elish et al. 2013; Kocaguneli et al. 2012; Azhar et al. 2013) discussed both types of EEE techniques.

3.5 Single Models (MQ6)

To count the frequency of single models used to construct the EEE techniques, we proceed as follow: (1) if it is a heterogeneous EEE technique, we count each single technique once. For example, if an ensemble is based on ANN and CBR, we count ANN once and CBR once; (2) If it is a homogeneous EEE technique, we count the single technique only once. In order to make the analysis of the frequency of single models used to construct EEE techniques clear, the base models of each type of ensembles were discussed separately (e.g. Homogeneous (HM) and heterogeneous (HT)). Table 6 shows that 12 single models have been used to construct EEE techniques (8 and 4 machine learning and non-machine learning models respectively).

3.5.1 Homogeneous EEE (HM)

As it can be seen from column 3 (HM) of Table 6, the ANNs (Minku & Yao 2013a) and Decision Trees (DTs) (Elish 2009) are the two single models most frequently used to construct HM EEE techniques: they are adopted by 57% (9 papers), and 37% (6 papers) of selected studies respectively. In fact, ANNs were used 14 times to construct the HM EEE. In particular, the MLP is the most investigated ANN: it was adopted by 9 out of 11 studies. DTs were used 10 times in order to build HM EEE: specifically, DTs construction-based on CART was the most adopted as single models with 5 out of 10 times. The CBR and SVR were adopted by 2 studies each, and were used 45 and 2 times respectively to construct HM ensembles. Note that the CBR as a single technique was investigated by (Azzeh et al. 2015) with 40 different configurations, and used to construct 44 HM EEE techniques. As for Regression and NF (Neuro-Fuzzy), they were used only once to construct HM ensembles and supported by one study. Note that there is no parametric model that has been used to construct HM EEE.

Table 4: Homogenous EEE (HM).

Homogeneous EEE	References
Bagging + M5P/Regression Trees (RT)	S1
Bagging + M5P/Model Trees (MT)	S1
Bagging + Multilayer perceptron (MLP)	S1, S8, S11, S12
Boostrapping+ MLP	S3
Bagging + Linear Regression (L.R)	S1
Bagging + Support Vector regression (SVR)	S1,S8
Bagging + Radial Basic Function (RBF)	S11
Bagging + RT	S11, S12
Negative correlation learning (NCL) + MLP	S11
Random + MLP	S11
Bagging +Adaptive neuro fuzzy inference system (ANFIS)	S8
Case based-reasoning (CBR, EBA)	S4, S15
Multiple additive regression trees (MART)	S7
Classification and Regression trees (CART)	S9, S10
MLP	S13, S14

3.5.2 Heterogeneous EEE (HT)

For the Heterogeneous EEE (see column 4 (HT) in Table 6), the CBR and ANN were the most adopted techniques (Elish 2013). In fact, they were adopted by

Table 5: Heterogeneous EEE (HT).

Heterogeneous EEE	References
Gaussian Process (GaP) + MLP + RBF + SVR + k-nearest neighbors (K-NN) + locally weighted learning (LWL) + Bagging (fast decision tree) + Additive regression with decision stump (ARwDS) + Random sub space (RSS) + Decision Stump (DS) + M5P + Conjunctive Rule (CR) + Decision table	S2
MLP + SVR + K-NN + RT + RBF	S5
COCOMO + L.R + CBR + artificial neural network (ANN*) + Grey Relational Analysis(GRA)	S6
L.R + CBR + ANN* + GRA	S6
Linear Regression + ANN* + GRA	S6
L.R + ANN*	S6
MLP + ANFIS + SVR	S8
CART + CBR	S9, S10
Multi linear regression (MLR) + Back-Propagation Neural Networks (BPNN) + RBF + dynamic evolving neural-fuzzy inference system (DENFIS) + Threshold-Acceptance-based Neural Network (TANN) + SVR	S16

(*) (Hsu et al. 2010) did not provide any information about the model architecture.

31% (5 studies) of selected studies each. They were used to construct 28 and 12 ensembles respectively, followed by DTs and SVR with 25% (4 studies) each; they were investigated 50 and 5 times respectively to build heterogeneous EEE. As for the Regression and NF, they were adopted by 2 studies each, and they were used 5 and 2 times respectively to build heterogeneous EEE. The remaining models were adopted only by one study (Hsu et al. 2010; Kocaguneli et al. 2009), and were used one time to construct heterogeneous EEE, except for GRA which was used 3 times.

3.6 Combinations Rules (MQ7)

The combination rule allows to get the estimation of an EEE technique by combining the single estimate of each of its base models (see Figure 1). From the selected studies, we identified 18 rules that have been used to get the prediction values of an ensemble. They fall into two categories of rules: linear and non-linear (Elish et al. 2013). Table 7 presents the type, the name of combination and the number of selected studies that use each rule.

Table 6: Distribution of single models used to construct EEE techniques.

Model		**#Papers**	**HM**	**HT**
ANN	MLP	9	13	3
	RBF	3	1	3
	ANN*	1	-	4
	TANN	1	-	1
	BPNN	1	-	1
	Total	**11**	**14**	**12**
DT	M5P/ RT	1	1	-
	M5P/ MT	1	1	-
	RT	3	2	1
	MART	1	1	-
	CART	2	5	44
	M5P	1	-	1
	Fast DT	1	-	1
	RSS	1	-	1
	DS	1	-	1
	ARwDS	1	-	1
	Total	**8**	**10**	**50**
CBR		**7**	**45**	**28**
SVR		**5**	**2**	**5**
Reg.*	L R	2	1	4
	MLR	1	-	1
	Total	**3**	**1**	**5**
NF**	ANFIS	1	1	1
	DENFIS	1	-	1
	Total	**2**	**1**	**2**
GRA		1	-	3
Decision table		1	-	1
Conjunctive Rule		1	-	1
Locally Weighted		1	-	1
Gaussian process		1	-	1
COCOMO		1	-	1

** Regression, **Neuro-Fuzzy.*

Table 7 shows that the linear rules are the most used ones. In fact, they were adopted by most of the selected studies. Indeed, the mean rule (i.e. Average) was the most frequently used with 81% of selected studies (13 papers), followed by the median rule with 25% of selected studies (4 papers). Whereas, the non-linear rules were adopted by three studies (Kultur et al. 2009; Vinaykumar et al 2009; Elish et al. 2013). In fact, they were used once, except for MLP and SVR rules which were used twice. Note that 6 studies use more than one combination rule. Indeed, Elish and al. (2013) use eight combination rules (2 of them were linear and 6 were non-linear), and 10 of the selected studies used only one combination rule.

4 IMPLICATION FOR RESEARCH AND PRACTICE

The findings of this systematic mapping study have implications for researchers and practitioners working in the SDEE area. It allows them to find out the existing EEE techniques as well as the base model used to construct them. This study found that the trends of EEE publications are characterized by discontinuity; therefore, researchers are encouraged to conduct more empirical studies on the EEE approaches since they are more likely to produce reliable results (Hastie et al. 2009).

Homogenous EEE are the most investigated, since they are the easiest to construct and evaluate. Heterogeneous EEE are more complex to elaborate since they use different base models. Consequently, researchers are encouraged to perform more experiments on Heterogeneous EEE. This mapping study concluded that a few number of single models (12 models) have been used to construct ensembles techniques, especially the parametric ones such as SLIM and COCOMO. Also there are some machine learning models such as those based on genetic programming and genetic algorithm that have not been used. The researchers' community is encouraged to investigate these single models in EEE to widen the possibility of using all single models of SDEE. Moreover, there are some models that showed a high performance singly, such as RT and CBR, but have not been sufficiently investigated (Wu et al. 2013; Minku and Yao 2013c). For example, CBR that incorporates Fuzzy Logic to measure the similarity between projects (Idri and Abran 2001) has shown a high performance accuracy when used to predict software effort (Idri et al. 2002; Idri et al. 2006). Even so, it is interesting that the researchers conducted more empirical studies in order to check the performance of ensemble techniques based on RT and CBR.

Concerning the combination rules, it was found that the non-linear rules were only used by three studies to get the estimation of EEE techniques. Moreover, only 6 studies used more than one combiner. Therefore, researchers are encouraged to investigate more combination rules.

5 CONCLUSION AND FUTURE WORK

This paper has presented a systematic mapping study that summarizes the existing EEE studies. This SMS examined and classified the ensemble techniques according to five classification criteria: research type, research approach, EEE type, single models used to construct EEE techniques, and rule used the combine single estimates in an EEE technique. Publication channels and trends were also identified. In total, 16 selected studies were identified. The findings of this SMS are summarized as follow.

(MQ1): EEE approaches have not been massively investigated in SDEE, as observed by the small number of publications in conferences/symposiums, journals and books.

(MQ2): The timescale of selected articles extends from 2000 to 2015 and the trends of EEE publications in SDEE are characterized by discontinuity since there is no publication in 2008, 2011 and 2014.

(MQ3): All selected papers belong to the Solution Proposal research type.

(MQ4): All selected papers belong to the history-based evaluation research approach.

(MQ5): The homogeneous EEE were the most investigated type; they were investigated by 12 of 16 selected studies.

(MQ6): 12 single models have been used to construct EEE techniques and among them, 8 machine learning models.

(MQ7): Two types of combiners were used to get the prediction effort of EEE techniques: linear and non-linear. The linear ones were the most frequently used.

A systematic literature review is ongoing to assess the research on EEE techniques by taking into consideration the results found in this systematic mapping study.

Table 7: Distribution of combination rules.

Type	Combination rule	#Papers
Linear Combination	Mean	13
	Median	4
	Inverse ranked weighted mean	2
	Mean weighted	2
	Equally weighted	1
	Median weighted	1
	Weighted adjustment based on criterion	1
	Outperformance combination	1
Non-linear Combination	MLP	2
	SVR	2
	Adaptive Resonance Theory	1
	Fuzzy inference system using fuzzy c-means (FIS-FCM)	1
	Fuzzy inference system using subtractive clustering	1
	ANFIS-FCM	1
	ANFIS-SC	1
	MLR	1
	RBF	1
	DENFIS	1

REFERENCES

Azhar, D. et al., 2013. Using Ensembles for Web Effort Estimation. In *2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement*. pp. 173–182. S10*

Azzeh, M., Nassif, A.B. & Minku, L.L., 2015. An empirical evaluation of ensemble adjustment methods for analogy-based effort estimation. *The Journal of Systems and Software*, 103, pp.36–52. S15*

Braga, P. et al., 2007. Bagging Predictors for Estimation of Software Project Effort. In *Proceedings of International Joint Conference on Neural Networks*. pp. 14–19. S1*

Elish, M.O., 2013. Assessment of voting ensemble for estimating software development effort. In *IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*. Singapore, pp. 316–321. S5*

Elish, M.O., 2009. Improved estimation of software project effort using multiple additive regression trees. *Expert Systems with Applications*, 36(7), pp.10774–10778. S7*

Elish, M.O., Helmy, T. & Hussain, M.I., 2013. Empirical Study of Homogeneous and Heterogeneous Ensemble Models for Software Development Effort Estimation. *Mathematical Problems in Engineering*, 2013. S8*

Hastie, T., Friedman, J. & Tibshirani, R., 2009. *The Elements of Statistical Learning: Data Mining , Inference and Prediction* Second Edi., Springer New York.

Hsu, C.-J. et al., 2010. A Study of Improving the Accuracy of Software Effort Estimation Using Linearly Weighted

Combinations. In *Proceedings of the 34th IEEE Annual Computer Software and Applications Conference Workshops*. Seoul, pp. 98–103. S6*

Idri, A. & Abran, A., 2001. A Fuzzy Logic Based Set of Measures for Software Project Similarity: Validation and Possible Improvements. In *Proceedings of the Seventh International Software Metrics Symposium*. London, pp. 85 – 96.

Idri, A., Amazal, F.A. & Abran, A., 2015. Analogy-based software development effort estimation: A systematic mapping and review. *Information and Software Technology*, 58, pp.206–230.

Idri, A., Khoshgoftaar, T.M. & Abran, A., 2002. Investigating soft computing in case-based reasoning for software cost estimation. *Engineering Intelligent Systems for Electrical Engineering and Communications*, 10(3), pp.147–157.

Idri, A., Zahi, A. & Abran, A., 2006. Software Cost Estimation by Fuzzy Analogy for Web Hypermedia Applications. In *Proceedings of International Conference on Software Process and Product Measurement*. Cadiz, Spain, pp. 53–62.

Jorgensen, M. & Shepperd, M., 2007. A Systematic Review of Software Development Cost Estimation Studies. *IEEE Transactions on Software Engineering*, 33(1), pp.33–53.

Kitchenham, B. & Charters, S., 2007. Guidelines for performing Systematic Literature Reviews in Software Engineering. *Engineering*, 2, p.1051.

Kocaguneli, E., Kultur, Y. & Bener, A.B., 2009. Combining Multiple Learners Induced on Multiple Datasets for Software Effort Prediction. In *Proceedings of International Symposium on Software Reliability Engineering*. S2*

Kocaguneli, E., Menzies, T. & Keung, J.W., 2012. On the Value of Ensemble Effort Estimation. *IEEE Transactions on Software Engineering*, 38(6), pp.1403–1416. S9*

Kultur, Y., Turhan, B. & Bener, A., 2009. Ensemble of neural networks with associative memory (ENNA) for estimating software development costs. *Knowledge-Based Systems*, 22(6), pp.395–402. S3*

Minku, L.L. & Yao, X., 2013a. An analysis of multi-objective evolutionary algorithms for training ensemble models based on different performance measures in software effort estimation. In *Proceedings of the 9th International Conference on Predictive Models in Software Engineering - PROMISE '13*. pp. 1–10. S13*

Minku, L.L. & Yao, X., 2013b. Ensembles and locality: Insight on improving software effort estimation. *Information and Software Technology*, 55(8), pp.1512–1528. S11*

Minku, L.L. & Yao, X., 2013c. Software Effort Estimation As a Multiobjective Learning Problem. *ACM Trans.Softw.Eng.Methodol.*, 22(4), pp.1–32. S14*

Petersen, K. et al., 2008. Systematic mapping studies in software engineering. *EASE'08 Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering*, pp.68–77.

Seni, G. & Elder, J.F., 2010. *Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions*,

Shepperd, M.J. & Kadoda, G., 2001. Comparing software prediction techniques using simulation. *IEEE Transactions on Software Engineering*, 27(11), pp.1014–1022.

Song, L., Minku, L.L. & Yao, X., 2013. The impact of parameter tuning on software effort estimation using learning machines. In *Proceedings of the 9th International Conference on Predictive Models in Software Engineering*. S12*

Vinaykumar, M.C.K, Ravi, V., 2009. Software cost estimation using soft computing approaches. In Handbook of Research on Machine Learning Applications and Trends, ed. *Handbook of Research on Machine Learning Applications and Trends*. IGI-global, pp. 499–518. S16*

Wen, J. et al., 2012. Systematic literature review of machine learning based software development effort estimation models. *Information and Software Technology*, 54(1), pp.41–59.

Wu, D., Li, J. & Liang, Y., 2013. Linear combination of multiple case-based reasoning with optimized weight for software effort estimation. *The Journal of Supercomputing*, 64(3), pp.898–918. S4*