Systematic Review Study of Decision Trees based Software Development Effort Estimation

Assia Najm[1], Abdelaziz Marzak[3]
Department of Mathematics and Computer Sciences
FSB M'sik, Hassan II University
Casablanca, Morocco

Abdelali Zakrani[2]
Department of Industrial
Engineering, ENSAM
Casablanca, Morocco

***Abstract*—The role of decision trees in software development effort estimation (SDEE) has received increased attention across several disciplines in recent years thanks to their power of predicting, their ease of use, and understanding. Furthermore, there are a large number of published studies that investigated the use of a decision tree (DT) techniques in SDEE. Nevertheless, in reviewing the literature, a systematic literature review (SLR) that assesses the evidence stated on DT techniques is still lacking. The main issues addressed in this paper have been divided into five parts: prediction accuracy, performance comparison, suitable conditions of prediction, the effect of the methods employed in association with DT techniques, and DT tools. To carry out this SLR, we performed an automatic search over five digital libraries for studies published between 1985 and 2019. In general, the results of this SLR revealed that most DT methods outperform many techniques and show an improvement in accuracy when combined with association rules (AR), fuzzy logic (FL), and bagging. Additionally, it has been observed a limited use of DT tools: it is therefore suggested for researchers to develop more DT tools to promote the industrial utilization of DT amongst professionals.**

Keywords*—Systematic literature review; decision tree; regression tree; software development effort estimation

I. Introduction

Much of the greater part of the literature on software project management pays particular attention to SDEE. According to [1] SDEE refers to the process of estimating the necessary effort needed for developing any software with regards to money, timeline, and staffing. The effort's unit is generally expressed in man-day/month/hour [2]. For instance, precise and accurate software cost prediction can result in successful control of the budget, time, and appropriate resource allocation. Unfortunately, overestimating is almost as strong a risk factor for software project failure as underestimating. Similarly, [3] found that inaccurate estimates of required resources are one of the most common reasons why software projects fail. Making correct estimation, therefore, helps in analyzing the practicability of any project regarding its cost-effectiveness [4] which ensures its success.

To date, there is a notable amount of studies investigating new models to perform accurate SDEE. In the SLR made by [5] over 304 candidate journal studies, they have outlined 11 prediction techniques which are grouped into two main groups: 1) algorithmic effort modeling which predict costs using a mathematical formula of project's attributes, 2) Machine learning techniques like (decision tree (DT), artificial neural networks (ANN), genetic programming (GA), and case-based reasoning (CBR)). Generally, machine learning techniques (MaL) have received considerable attention thanks to their power of modeling complex relations between software attributes and the target value (software cost), extremely where the form of the relationship cannot be straightforwardly determined. In the same vein, [6] has also conducted an SLR where they listed eight types of machine learning models. Overall, the results indicate that ANN, analogy based estimation (ABE), and DT are the most commonly employed SDEE techniques with (37%, 26%, and 17% respectively). A similar decreasing order is reported in [5]. Furthermore, DTs were adopted for SDEE mostly for their capability of predicting and interpreting results, unlike other MaL techniques as claimed by [7] in their systematic mapping study of decision tree-based SDEE where they identify 46 relevant papers. However, there exist some strong conditions and limitations that affect the ease use of DT techniques in a specific context (see Section III.C).

Also, results from earlier studies demonstrate a strong and inconsistent accuracy of DT, as compared to MaL and Non-MaL cost estimation techniques. According to some papers [8][9][10], DT outperforms regression models. This outcome is contrary to that of [11][12][13] who have highlighted the relevance of regression models in providing more accurate estimates than DT models. Moreover, DT show superior accuracy than RBFN models as reported in [14][15][16][17] differs from the findings presented in some published studies [17][18]. These existing inconsistent results have heightened the need for reviewing the evidence of the DT model, to better understand and enhance their application.

Furthermore, in reviewing the literature, it should be noted that there is no SLR of DTs for software effort estimation. Thereby, we follow the methodology presented by [19] in order to make a concise selection, deep examination, and synthesizing findings of all DT studies made from 1985 until 2019. This study examines the evidence of DT models concerning the following five perspectives: (1) the prediction accuracy of DT methods; (2) the comparison of prediction accuracy of DT techniques and other methods; (3) the suitable estimation contexts for employing DT techniques; (4) the effect of combining other methods with DT models; and (5) tools that implement DT methods.

The organization of this study is as follows. Section II outlines the methodology of research used to perform this SLR. Section III describes and analyzes distinct review results; Section IV summarizes the fundamental finding and suggests some recommendations for research and practice. Section V reports this review's limitations. Finally, Section VI presents conclusions and gives the perceptiveness of future work.

II. Methodology

The main steps of this SLR are: determining review questions, explicating the strategy of research, making a study selection, performing a quality assessment, extracting, and synthesizing data. All these steps will be detailed in the subsequent subsections.

A. Review Questions (ReQs)

This SLR attempts to assess the evidence of DT methods and to perform favorable recommendations based on the certainty of results. The five review questions are as follows:

ReQ1: How is globally the prediction accuracy of DT methods?

ReQ2: What is the performance of DT methods in comparison with other methods (MaL or Non-MaL)?

ReQ3: What are the suitable conditions for an accurate estimation of DT techniques?

ReQ4: How does the combination of other techniques with DT techniques affect the estimation accuracy?

ReQ5: What are the most commonly used DT tools?

B. Search Strategy

The search strategy encloses three phases that help at answering the ReQs, which are outlined precisely thereafter.

1) Search string: We construct the search string from words derived from ReQs and also by searching their homonyms, along with employing AND, OR, and NEAR operators to restrain the research results. We use the same search string conceived by Najm et al. [7].

2) Literature resources: To seek relevant studies, we use the next five electronic databases considering that they are largely employed in review studies: IEEE Xplore, Science Direct, ACM Digital Library, Springer, and DBLP.

3) Search process: The search process is handled out in two stages: in the first stage we search in digital databases for a query string to select relevant studies, the inspection takes into account the abstract, the document's title, keywords/Index as well as the whole text to not miss any suitable paper, after that the second stage consists of looking for additional papers by examining references of predetermined articles (selected in the 1st stage).

C. Study Selection

The study selection aims at identifying appropriate articles that address ReQs. So, to achieve this purpose, we use the inclusion/exclusion criteria to choose or discard the papers.

We notice that we employ the similar inclusion/exclusion criteria used by Najm et al. [7].

Fig. 1 shows the total of selected or remained papers after each phase, while phases are marked by letters from a to f.

D. Quality Assessment

Quality assessment (QuA) was conducted in this review to prevent any biased information that can affect the findings. For this purpose, the quality of 50 extracted papers was evaluated using the six following questions:

QuA1: Does the paper define explicitly the intended goals of the study?

QuA2: Does the study present properly the solution proposed?

QuA3: There exists a clear explanation of the estimate's context?

QuA4: There exist some supporting studies reported in the paper?

QuA5: Does the paper make any significant contribution to academia/industry?

QuA6: What is the quality of the publication channel where the articles were published?

Concerning questions 1-5, they can accept three answers as follows: ''Yes", ''Partially", and ''No", which have the corresponding scores: (+1), (+0.5), and (0).

While question 6 was scored based on the rates provided in Scimago Journal Rank (SJR) and Conference Rankings (CORE) [20]. It accepts these answers:

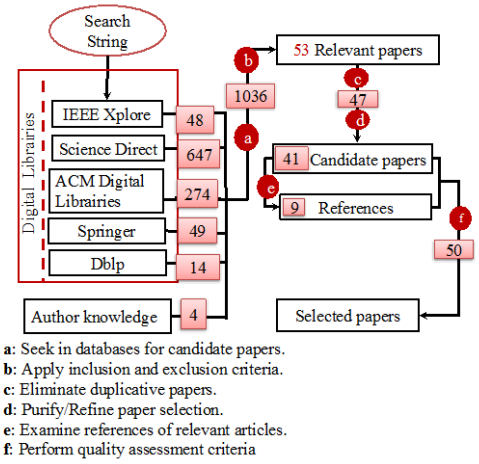

Fig. 1. Search, Selection and QuA Process.

Journals: (+1) for journals ranked Q1, (+0.5) for journals ranked Q2, and (0) for journals ranked Q3 or Q4.

Conferences/workshops/symposiums: (+1) for conferences/workshops/symposiums ranked CORE A, (+0.5) for the conferences/workshops/symposiums ranked CORE B, and (0) for conferences/workshops/symposiums ranked CORE C.

Although the QuA criteria, as well as their rates, might be nonobjective, they help us to compare the chosen studies. We note that the same criteria were employed in [6][21]. The quality assessment was conducted separately by two researchers who answer carefully the answers; any discord was discussed and finally, fixed by mutual agreement between the two researchers. We then selected only papers whose score rise above 3 (50% of the excellent quality of a paper: 6). All 50 relevant papers were then selected due to their suitable quality score of more than 3.

E. Data Extraction and Synthesis

The data extraction is used to extract all relevant data from selected papers to answer ReQs. Table I shows the form of data extraction.

To deal with the research question posed in this review study, two researchers read separately and synthesize carefully the selected papers, there were some disagreements concerning some review questions. Though, any discord was discussed and finally resolved by mutual agreement between the two researchers. It is worth noting, that for some review questions such as ReQ1, ReQ2, and ReQ4 the data was not obtained directly. We followed the same solution reported in [6]. Therefore, for the studies using multiple configurations, only the value relative to the best performance was extracted. While for studies using different database sampling, we used the mean of the accuracy value.

To address the review questions, the next step after the data extraction is the data synthesis, which aims to promote and enhance the generalization of the result. Yet, various methods were adopted:

- Narrative synthesis: It consists of enumerating the data and summarizing the finding of studies. We use tables, bar charts, and boxplots to strengthen the visualization of results.
- Vote counting: It intends to sum up the number of cases where a model outperforms or underperforms other models. It was used to address ReQs (ReQ2).
- Reciprocal translation: It consists of a translation of notions listed in the selected studies to determine similarities and recognize a difference between them. It was used to address the review question ReQ3.

TABLE I. DATA EXTRACTION FORM

Data extractor
Data checker
Study identifier
Name(s) of the author(s)
Article title
Author(s) purposes
ReQ1 – Estimation accuracy criteria and methods used to assess DT techniques
ReQ2 – Performance of DT techniques in comparison with other methods
ReQ3 – The suitable conditions for an accurate estimation of DT techniques
ReQ4 – Effects of combining DT techniques with other models
ReQ5 – The most commonly used DT tools

III. REVIEW RESULTS

In this section, we report and analyze the findings of all ReQs. A deep discussion and interpretation of the finding will be addressed in the following subsections.

A. Estimation Accuracy of DT Techniques (ReQ1)

The majority of studies are based on a history-based type, which means that the evaluation of DT techniques is based on historical software project datasets. Consequently, the accuracy of these DT estimation techniques may depend on certain categories of parameters which are organized into three different groups: the first concerns the dataset's characteristics like (dimension, outliers and missed data, etc); the second is about the DT's structure (Split rule, number of cases per node, depth of the tree, stopping criteria, effort calculation method, etc); the third concerns the employed techniques of evaluation and validation such as (assessment measures, k-fold, the leave-one-out method, etc.).

Additionally, it has been observed in the 50 selected studies that several datasets were applied to form and to assess the performance of DT models (see Fig. 2).

Table II shows the most commonly used databases mainly those employed in more than four studies, along with the proportion, the number of papers that employ each database, and the totality of projects per number of studies. What can be seen in Table II is the high rate of usage of the ISBSG dataset (20%), and then the COCOMO (13%) followed by Desharnais and NASA with (11% and 8% respectively).

Besides, several evaluation techniques were used to assess the prediction accuracy of DT models. The three techniques mostly used were holdout, leave-one-out (LOO), and k-fold (k>1). The Holdout was largely used about (72% or 28 of papers), followed by k-fold cross-validation (36% or 14 of studies), and LOOCV (21% or 8 of studies), we note that the total number of percentage exceeds 100% since some papers use more than one evaluation method.

Regarding the accuracy criteria, the selected studies use several measures; especially the MMRE is employed in 31 papers (63%), Pred(25) is employed in 29 papers (59%), and MdMRE is employed in 15 papers (31%). Therefore, these three measures were chosen to address the ReQ1.

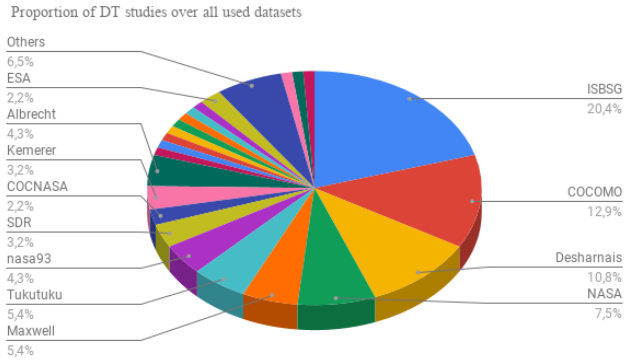

Fig. 2. The Proportion of DT Studies over all Datasets. (Others: for Databases without a well-known Name like Student Projects or Software House Projects).

TABLE II. DATASETS USED FOR DT EVALUATION.

Datasets	Total Number of studies	Proportion	Number of instances /number of studies
ISBSG	19	20,43	>1000/15, 789/1, 69/1, 500/1,501/1
COCOMO	12	12,9	63/9, 252/1
Desharnais	10	10,75	81/7, 77/3
NASA	7	7,53	60/3, 18/2, N/2
Maxwell	5	5,38	62
Tukutuku	5	5,38	53/3, 87/1, 150/1

N: The total number of database projects not specified

Though where it is not obvious to report directly the values of accuracy measures, we used the following logic: if there were various configuration models, we extracted the values of the best configuration, but if there were different database sampling, we calculate the means of the accuracy values. We take advantage of boxplots to have a clear interpretation of each accuracy criteria based on the values reported in articles.

Fig. 3 shows that the medians of accuracy values are as follows: median of MMRE is around 29%, the median of MdMRE 23%, and a median of Pred(25) is around 51%. It is known that contrary to Pred(25), lower values of MMRE and MdMRE show better estimation accuracy. From the data in Fig. 3 it is apparent that MMRE's distribution as well as that of MdMRE and Pred(25) present a positive dissymmetry because the medians are nearer to the inferior quartile.

What stands out in this figure by considering the distance between the lower and upper quartiles is the fewer variations of the values of MMRE. Therefore the box of MMRE is shorter than the boxes of MdMRE and Pred(25), elsewhere there is a possible explanation for this result: the values used for boxplots stem from various DT models that used different datasets specifications and several evaluation methods.

Typically, all databases apart from Tukutuku and COCOMO, have a mean of MMRE ranging between 17% and 68%, that of MdMRE between 11% and 44%, and that of Pred(25) between 36% and 89%. Therefore, it is awkward to report any conclusion because of the modest number of studies and experiments.

B. Accuracy Comparison between DT Models and MaL/Non-MaL Techniques (ReQ2)

This section set out to compare the preciseness of DT models with eleven MaL and Non-MaL methods: Regression (Reg), Radial Basis Function Neural Networks (RBFNN), COCOMO model (CCM), Use Case Point (UCP), Stepwise Anova (SA), Support-Vector Machines (SVM), Multilayer Perception (MLP), Case-Based Reasoning (CBR), Analogy Based Estimation (ABE), k-Nearest Neighbors (KNN), Association Rules (AR). To achieve this purpose, we had counted the amount of evaluations where DT models perform better (or less) than the eleven methods in terms of a particular estimation accuracy measure. Fig. 4 to 6, provide the results obtained from this comparison analysis (the "+" sign in front of MMRE/MdMRE/Pred signifies that DT methods perform better while the "-" sign signifies that DT methods perform less than the other models), we mention that the blue colors show the total examinations where DT methods perform better, while the red colors present the total examinations where DT methods perform less. Concerning Non-MaL methods, the majority of papers compare DT models with Reg models (87 examinations). From the data in Fig. 4 to 6, it is apparent that DT models perform better than Reg according to the MMRE measure. Similarly, regarding the MaL methods, the major part of DT papers makes a comparison with MLP (41 examinations), SVM (35 examinations), then RBFN and CBR (21 and 20 examinations respectively). According to the MMRE, MdMRE, and pred(25) values, we found that DT models perform better than MLP, RBFN, and CBR. Moreover, SVM outperforms DT methods in terms of the aforementioned three accuracy measures. However, for the remaining techniques, it is hard to report any inspection because of the few numbers of evaluations (less than 10 evaluations).

Additionally, all previously mentioned results are gathered from DT studies, so they might be subjective.

C. Prediction Context of DT Methods (ReQ3)

Given that, the investigated software effort estimation techniques provide various results, it is crucial to give closer attention to the favorable context of prediction more than looking for the perfect estimation model.

Mendes [22] has investigated numerous effort estimation techniques and asked a question: what technique to employ? The answer is "it depends". The main explanation is that the estimation depends on the context of prediction, which is related to database characteristics (dimension, outliers, attributes' types, missed data, and amount of collinearity) and different model designs.

Our review study intends to investigate these issues; therefore, we have retrieved and listed the advantages and limitations of DT techniques which were especially reported in the selected papers, see Table III. The main finding is that DT approaches have a greater sensitivity to the type and quality of historical datasets, which have a considerable effect on their estimation accuracy.

Some studies have examined the impact of dataset size on the estimation accuracy, it is found that DT techniques perform better with smaller datasets like in [23][9][24][25][26][27]. However opposed results were found, for instance [25] found that DT techniques can perform well when large datasets were employed. Nevertheless, it is challenging to confirm that DT techniques should be favorable in small datasets considering that satisfactory results were achieved in large datasets.

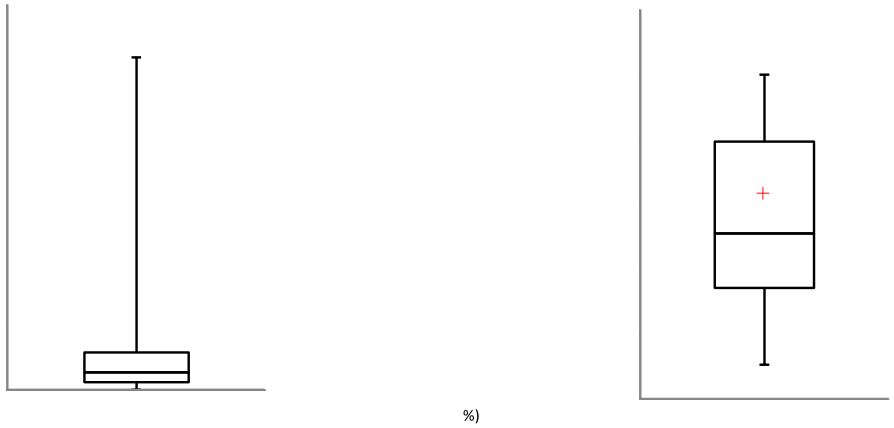

Fig. 3. Boxplots of MMRE(%), MdMRE(%), Pred(25%).

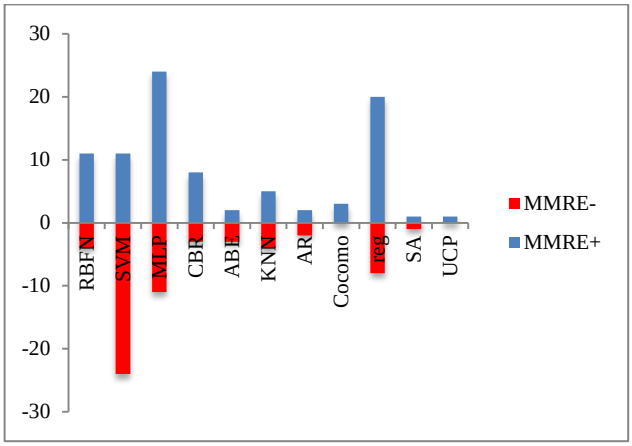

Fig. 4. Comparison of DT Methods with other Techniques in Terms of MMRE.

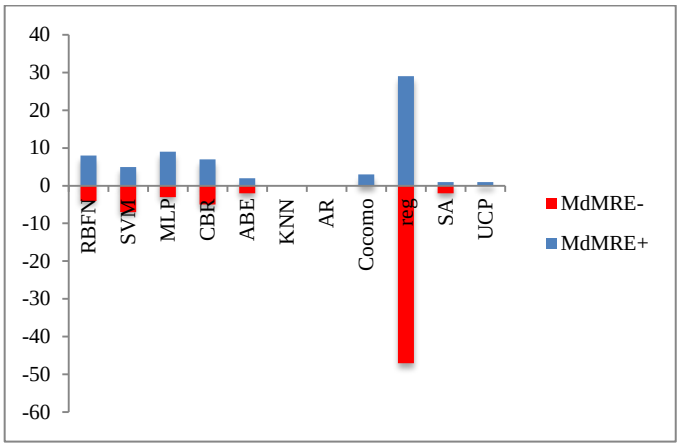

Fig. 5. Comparison of DT Methods with other Techniques in Terms of MdMRE.

Besides, DT techniques have a great challenge to extrapolate beyond the data on which it was trained, for example, these studies [8][28][27][29][30] confirm that DT techniques are typically unable to give accurate estimates for a project not similar to those available in the training set.

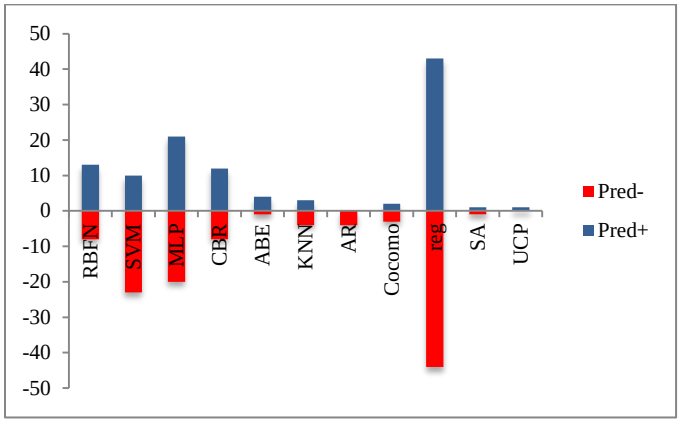

Fig. 6. Comparison of DT Methods with other Techniques in Terms of Pred(25).

Furthermore, classical DT methods cannot deal with imprecision and uncertainty [23][31][27]. As a result, numerous techniques have been suggested to handle imprecise data and therefore obtain more accurate estimates. In particular, [32][33][34] have suggested an improved technique which uses the concepts of fuzzy logic (FL). Their methods improve the performance of the traditional DT methods by incorporating the concept of FL theory.

One of the great advantages of using DT techniques is their resistance to outliers as reported in [14][25][35][27][10] and robustness to any multi-collinearity problems such as [36][37][38][27]. This is because DT methods perform an automatic feature selection as argued in these papers [14][39][36][40] which means that they just select the relevant features which have an important impact on the effort. Moreover, these studies [14][15][35] have shown that DT methods provide accurate effort estimation without performing a variable selection which strengthens the idea of resistance to multi-collinearity problems.

Other influencing factors must be taken into consideration along with dataset characteristics. They are all listed in Table III. For instance, The DT methods are suggested when it is difficult to describe the complex relationships that exist

among project attributes and effort. This is because DT techniques guide the practitioners to know which effort factors have a potential effect on the prediction and how the model derives the results; this is why DT methods produce more interpretable and comprehensible results; also they can perform well at an early stage of project development with just early available attributes; which help practitioners at taking good decisions; but they require the use of historical dataset to generate an estimate.

TABLE III. ADVANTAGES AND LIMITATIONS OF A DT TECHNIQUE

Advantages	**Supporting studies**
A DT method can handle categorical variables.	[41], [34]
RTs could handle different scales of attributes.	[18], [34], [31]
DT is designed for exploratory data analysis.	[42]
Automatic handling of missing values.	[14], [27]
Resistance to outliers.	[14], [25], [35], [10], [27]
Automatic variable subset selection.	[14], [36], [39], [40]
DT methods are among the best even without feature selection.	[14], [15], [35]
Perform better with early available attributes' scheme.	[43], [41], [34]
DT methods are recommended for relevantly homogenous databases.	[42]
Can deal with imprecision and uncertainty.	[32], [33], [34]
The ability to learn from past finished projects and have predictive power.	[27], [30]
High comprehensibility and producing more interpretable results.	[8], [12], [22]–[24], [29], [31], [34], [35], [42], [44]–[46]
High applicability.	[23], [27], [29]
DTs are not used only for classification but also for regression issues.	[12], [22], [27], [29], [30], [37], [38], [46], [47]
Limitations	**Supporting studies**
It cannot extrapolate beyond the data on which it was trained.	[8], [27]–[30]
The sensitivity of DT approaches to the nature and quality of historical data.	[11], [12], [17], [18], [28], [30], [35], [40], [41], [46], [48], [49]
Not perform better on large datasets.	[9], [26], [27], [29]
Depends on the training set size.	[12], [46]
Classical DT methods Cannot deal with imprecision and uncertainty, medium uncertainty.	[23], [27], [31]
Low accuracy.	[8], [23], [30], [50]
Medium causality and medium sensitivity to parameters changes.	[23], [28], [39], [51]
Not provide any meta information to guide the project manager in the budgeting process.	[50]
Accuracy Not significantly sensitive to the company-specific data or multi-organizational data.	[9], [22], [26], [31], [31]
Some DT methods act as a black box.	[10]
Need completed and historic databases.	[22]

In sum, DT techniques have many advantages however; they suffer from some limitations, which can be bypassed by ensembles methods like in [52] or by the integration of other techniques.

There exist some studies such as [10][32][25][24][36][40][43][44][53][45][51] that recommend the use of hybrid models that incorporate other techniques along with DT methods to enhance the prediction accuracy.

Note that in the next section we will discuss concisely the impact of the combination of other techniques on the performance of DT models.

D. Effect of Combining a DT with Other Method (ReQ4)

The present subsection investigates the effect of combining a DT with another technique especially the effect of each technique on the estimation accuracy. Table IV gives the MMRE improvement along with MdMRE improvement and Pred(25) improvement for each method employed in association with DT approaches. We note that the accuracy improvement was made only for studies that report the accuracy of DT combination compared to the accuracy related to DT alone.

Table V provides more details about each associated method: the total number of articles dealing with each method, the number of articles comparing the accuracy, and the total examinations done in these papers. For instance, from the 8 papers, which combines Fuzzy Logic with DT (FL-DT) methods, just 2 papers made an estimation accuracy comparison with that of a DT method alone, and only 3 evaluations were made to assess the accuracy of the estimation. Meanwhile, for a certain number of methods, which are associated with DT models, the number of examinations was considerably greater than the total papers including those techniques. For example, grid search combined with generic backward input selection (GS+GBIS), there was only 1 study investigating the comparison of DT techniques with that of a GS+GBIS+DT technique, yet 9 evaluations were performed.

We mention that the number of combined methods may be (>=1) such as ABE line in Table V shows the accuracy values when combining (ABE) alone with DT techniques while (Boost+PCA+Poisson) line presents the values of accuracy when combining Bootstrapping, Principal Component Analysis (PCA) and Poisson Regression. In sum, note that the Bagging, Regression, Fixed Size Window Policy (FSWP), and (Boost+PCA+Poisson) were less incorporated with DT techniques (one examination by one paper).

Closer inspection of Table V, considering the number of evaluations and MMRE's median, FL is the best method, which strengthens the accuracy of DT techniques (92,56% improvement), followed by Boost+PCA+Poisson (88,33 %) and AR (78,22%). On the basis of the MdMRE's median, Boost+PCA combined with Poisson Regression has the most improvement (71.42%), followed by GS combined with GBIS (5.99%). According to the median of Pred(25), AR has the greatest effect (84.99% improvement), followed by FL (18.45%).

To prevent the bias coming from the evaluations made on the same study, we investigate the impact of combining other techniques with DT methods, by taking advantage of the totality of articles, instead of the totality of examinations. Table V shows that Reg, FL as well as ABE are the three methods frequently combined with DT techniques. Table IV indicates that according to Reg, FL lines as well as that of ABE, only FL technique has the greatest improvement based on the MMRE and Pred(25) accuracy measures.

To summarize the findings, we realize that not all presented methods in Table IV, contribute necessarily to the accuracy improvement of DT techniques mainly, FL, Bagging and AR are the only ones that improve both MMRE and Pred(25) criteria, which are supported by 2,1 and 1 studies respectively. We figured out that, Bagging contributes to a small improvement in accuracy when combined with DT techniques. Due to the fact that Bagging gives good results with good basic learners otherwise if the basic learner is bad, bagging may contribute to the degradation of the accuracy of estimates.

Moreover, AR appears to be a more promising technique than FL when combined with DT techniques since it improves significantly both accuracy measures MMRE and Pred(25).

Nevertheless, all these results require more evaluations in more search studies due to the restricted amount of papers that analyzed the effect of incorporating other techniques with DT.

E. DT Tools (ReQ5)

In this SLR we identify seven tools, which are listed in Table VI. Weka presents the mostly employed tool, then Matlab, SPSS AnswerTree version, and Fispro.

Weka is an application developed by researchers, it is open-source software based on Java. It contains a set of machine learning algorithms, in particular data preprocessing, clustering, classification, and AR extraction.

MATLAB is a numeric-computing environment that was developed by MathWorks but it was created by Cleve Moler in the 1970s. Also, there exist statistical tools built on MATLAB, which offer a set of unsupervised and supervised MAL algorithms including decision trees with boosting and bagging techniques.

TABLE IV. PERCENTAGE OF MMRE, MDMRE, AND PRED(25) IMPROVEMENT OF EACH ASSOCIATED METHOD WITH DT

Incorporated methods with DT	Reference	Dataset	MMRE improvement (%)	MDMRE improvement (%)	Pred(25) improvement (%)	Incorporated methods with DT	Reference	Dataset	MMRE improvement (%)	MDMRE improvement (%)	Pred(25) improvement (%)
FL	[32]	COCOMO'81	98	N	18,45	GS +GBIS	[48]	ISBSG	N	-1,25	0,44
	[32]	Tukutuku	92,41	N	2,01			Experience	N	4,2	5,22
	[1]	Tukutuku	92,56	N	40,15			ESA	N	2,55	3,4
Bagging	[45]	NASA	0,73	N	6,67			ISP05	N	9,11	2,54
ABE	[9]	House project	-16,21	-39,13	8,16			Euroclea	N	9,38	21,32
	[31]	Laturi	-89,16	-17,37	1,58			COCNASA	N	4,25	23,89
Reg	[9]	House project	2,7	-30,43	-5,44			coc81	N	14,52	14,41
PCA	[18]	NASA	-75,17	-107,72	-14,02			Desharnais	N	8,98	3,46
		USC	-144,06	-129,89	-15,36			Maxwell	N	5,99	-0,65
		SDR	-43,3	-73,16	-11,42	Boost+PCA +Poisson	[51]	Software House	88,33	71,42	N
FSWP	[40]	ISBSG	22,85	N	N	Voting Ensemble	[49]	Coco81	-36,47	N	N
GA	[39]	Desharnais	3,09	N	9,98			nasa93	-43,01	N	N
		NASA	-3,37	N	0			cocomonasa_v1	-113,52	N	N
AR	[42]	ISBSG	76,18	N	90,15						
		STTF	80,27	N	79,84						

TABLE V. THE TOTALITY OF PAPERS COMPARING THE ACCURACY AND THE NUMBER OF EXAMINATIONS MADE FOR EACH ASSOCIATED METHOD WITH DT MODELS

	FL	Bag	ABE	Reg	PCA	FSWP	GA	GS+GBIS	AR	Boost+PCA +Poisson	Voting Ensemble	Bees algorithm
No. Of studies	**8**	2	**3**	**4**	2	1	1	1	1	1	2	1
Total papers comparing the accuracy	**2**	1	**2**	**1**	1	1	1	1	1	1	1	0
No. Evaluations	**3**	1	**2**	**1**	3	1	2	9	2	1	3	8

TABLE VI. DT TOOLS

Tool	Authors	Year	Studies using the tool	References
Weka	Hall et al.	2008	[45]	[54]
	Witten and Frank	2000	[49]	[55]
	Hall et al.	2009	[25][1], [35][1]	[56]
	N	N	[57][1], [58][7]	N
Matlab	N	N	[16], [24], [59], [34][2], [60][8]	N
SPSS I*nc*	N	N	[43][3], [34], [41][4], [12], [61][5], [47][6]	N
Fispro	Guillaume et al.	2002	[1],[32]	[62]
CART software	Dan and Colla	1995	[31], [26]	[63]
MART software (TreeNet)	Salford-Systems	1997	[14]	[64]

[1] – RepTree, [2]- v.7.5.0, [3]- v.15.0, [4]- v.17.0, [5] - AnswerTree v 2.1.1, [6] - AnswerTree v 3.1, [7] - v 3.8 REPTree, and M5P, [8] - v.7.1.

Over the whole selected papers, only a few studies have employed DT tools to obtain or generate software effort estimates. Moreover, the majority of the existing tools implement the traditional DT methods, which didn't integrate other techniques, for example, FL, GS (Grid Search) to enhance the estimates.

IV. SUMMARY AND IMPLICATIONS FOR SEARCH AND USE

Our suggestions concerning the use of DT models in SDEE concern are listed below:

The estimates' accuracy of DT methods: Due to the modest number of studies we were unable to draw any conclusion. Furthermore, the majority of studies use historic databases, so we suggest carrying out more works with the help of concrete and practical experience in the industrial sectors.

The accuracy of DT compared to that of MaL and Non-MaL methods: the DT techniques outperform some models including MaL and Non-MaL. Typically, RBFN, for which there were enough evaluations. Nevertheless, to report a definitive result is a challenging issue, because of the insufficient number of studies investigating accuracy comparison. It is therefore interesting for researchers to conduct more experiments to deal with this issue.

The suitable conditions for an accurate estimation of DT techniques: It should be noted, that it is difficult to make a conclusion concerning the use of DT techniques. Consequently, practitioners have to figure out which techniques had to be in combination with DT methods towards overcoming limitations relative to (missing values, categorical data, features selection, etc.) and accommodate DT to their context.

Effect of combination of other methods with DT methods: the accuracy of estimates of DT models was not usually enhanced. The results show that using bagging techniques doesn't improve greatly the accuracy of DT techniques in comparison with the AR and FL techniques. This indicates that MaL techniques are more desirable to be incorporated with DT methods rather than Non-MaL techniques.

DT tools: We have recognized in this review study, seven tools to estimate software effort using DT methods. Especially, WEKA and MATLAB are the tools most often used. Moreover, the majority of tools implement classical DT methods. It is therefore suggested for researchers to investigate the implementation of other techniques along with DT models that enhance significantly the estimates' performance like AR, FL, and Bagging and hence encouraging industrial utilization of DT amongst professionals.

V. LIMITATIONS OF THIS REVIEW

The three accuracy metrics used in this review are MMRE, MdMRE, and Pred(25).

However, these indicators don't take into account the quality of databases so implicitly they suppose that the estimation method may give estimates with a maximum precision of 100% for a particular database [65]. Additionally, the MMRE has been subject to criticism for being not balanced in several evaluation contexts in addition to its penalization character of overestimated values further than underestimated ones [66], [67]. Even though, in this review study, we are based on these three criteria, since they were widely employed in relevant articles.

In addition, it is challenging to define the circumstances of all estimates because they were obtained from the selected studies based on various DT techniques and using several experimental designs, which include design decisions (feature selection, project selection, split rule, stopping criteria, pruning, etc.) and validation methods (holdout, LOOCV, k-fold cross-validation, etc.).

Moreover, in this review, we consider only studies about DT techniques. For that reason, the mentioned performance of DT techniques would be overestimated, besides that, the advantages and limitations of each DT technique may be subjective. Therefore, the reader must also take into consideration the potential effect of the authors' concern and viewpoint on these results.

VI. CONCLUSION AND FUTURE WORK

This systematic review synthesizes the results of DT studies in conformity with software effort estimation. Moreover, the selected papers were examined according to the five perspectives: prediction accuracy, the performance of DT

techniques in comparison with other methods, contexts of the estimates, and effect of the combination on DT's performance, and DT tools.

In sum, we identified 50 relevant papers, especially between the years 1985 and 2019. The important results found in this review study are as follows:

What is the overall performance of DT techniques? The overall picture suggests that no conclusive affirmation can be made since the mean accuracy values are around 52,5% for MMRE, 26,1% for MdMRE, and 56,1% for Pred(25).

What is the performance of DT techniques in comparison with other methods (MaL or Non-MaL)? In general, DT techniques outperform RBFN, MLP, and CBR techniques. Especially, they outperform also Regression models according to MMRE.

What are the suitable conditions for an accurate estimation of DT techniques? Many studies confirm that DT methods can describe the complex relationships that exist among project attributes and effort, and can produce more interpretable and comprehensible results. In addition to their resistance to outliers and robustness to any multi-collinearity problems. However, classical DT methods cannot deal with imprecision and uncertainty. Furthermore, several papers propose the use of hybrid models to overcome the existing DT limitations.

How the combination of other techniques with DT techniques does affect estimation accuracy? The techniques the most commonly used in combination with DT studies are fuzzy logic followed by regressions. However, not all combined techniques improve the accuracy estimation of DT techniques. Typically Association rules, fuzzy logic, and bagging are the techniques that improve the prediction accuracy of DT based on the MMRE and Pred(25) measures.

What are the most commonly used DT tools? WEKA, created by researchers at the University of Waikato is the most widely used tool to estimate effort using DT techniques.

In terms of future work, it would be interesting to perform a comparative study and repeat the experiment using unbiased evaluation criteria like the standard accuracy (SA), and the effect size rather than the biased MMRE.

REFERENCES

[1] Idri and S. Elyassami, “A Fuzzy Decision Tree to Estimate Development Effort for Web Applications,” Int. J. Adv. Comput. Sci. Appl., vol. 1, no. 3, 2011, doi: 10/gfrgk2.

[2] J. Brooks Frederick, The Mythical Man-Month: Essays on Software Engineering. 1995.

[3] R. N. Charette, “Why software fails [software failure],” IEEE Spectr., vol. 42, no. 9, pp. 42–49, Sep. 2005, doi: 10/cwspfc.

[4] S. M. Satapathy, M. Kumar, and S. K. Rath, “Fuzzy-class point approach for software effort estimation using various adaptive regression methods,” CSI Trans. ICT, vol. 1, no. 4, pp. 367–380, Dec. 2013, doi: 10/gf7784.

[5] M. Jørgensen and M. J. Shepperd, “A Systematic Review of Software Development Cost Estimation Studies,” IEEE Trans. Softw. Eng., vol. 33, 2007, doi: 10/cmwgb8.

[6] J. Wen, S. Li, Z. Lin, Y. Hu, and C. Huang, “Systematic literature review of machine learning based software development effort estimation models,” Inf. Softw. Technol., vol. 54, pp. 41–59, 2012, doi: 10/b6cd53.

[7] A. Najm, A. Zakrani, and A. Marzak, “Decision Trees Based Software Development Effort Estimation: A Systematic Mapping Study,” in 2019 International Conference of Computer Science and Renewable Energies (ICCSRE), Jul. 2019, pp. 1–6, doi: 10/gf7785.

[8] A. B. Nassif, M. Azzeh, L. F. Capretz, and D. Ho, “A comparison between decision trees and decision tree forest models for software development effort estimation,” in 2013 Third International Conference on Communications and Information Technology (ICCIT), Beirut, Lebanon, Jun. 2013, pp. 220–224, doi: 10.1109/ICCITechnology.2013.6579553.

[9] S. D. M. G Costagliola, “Effort Estimation Modeling Technique A Case Study For Web Application,” in Proceedings of the 6th International Conference on Web Engineering, ICWE 2006, Palo Alto, California, USA, July 11-14, 2006, 2006, p. 8.

[10] A. B. Nassif, L. F. Capretz, D. Ho, and M. Azzeh, “A Treeboost Model for Software Effort Estimation Based on Use Case Points,” in 2012 11th International Conference on Machine Learning and Applications, Boca Raton, FL, USA, Dec. 2012, pp. 314–319, doi: 10.1109/ICMLA.2012.155.

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

[12] E. Mendes, “A Comparative Study of Cost Estimation Models for Web Hypermedia Applications,” Empir. Softw. Eng., p. 34, 2003.

[13] R. Jeffery, M. Ruhe, and I. Wieczorek, “Using public domain metrics to estimate software development effort,” in Proceedings Seventh International Software Metrics Symposium, London, UK, 2001, pp. 16–27, doi: 10.1109/METRIC.2001.915512.

[14] M. O. Elish, “Improved estimation of software project effort using multiple additive regression trees,” Expert Syst. Appl., vol. 36, no. 7, pp. 10774–10778, Sep. 2009, doi: 10/d449c3.

[15] P. L. Braga, A. L. I. Oliveira, G. H. T. Ribeiro, and S. R. L. Meira, “Bagging Predictors for Estimation of Software Project Effort,” in 2007 International Joint Conference on Neural Networks, Aug. 2007, pp. 1595–1600, doi: 10.1109/IJCNN.2007.4371196.

[16] S. M. Satapathy, B. P. Acharya, and S. K. Rath, “Class point approach for software effort estimation using stochastic gradient boosting technique,” ACM SIGSOFT Softw. Eng. Notes, vol. 39, no. 3, pp. 1–6, Jun. 2014, doi: 10/gfrgkt.

[17] M. O. Elish, “Assessment of voting ensemble for estimating software development effort,” in 2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), Singapore, Singapore, Apr. 2013, pp. 316–321, doi: 10.1109/CIDM.2013.6597253.

[18] B. Baskeles, B. Turhan, and A. Bener, “Software effort estimation using machine learning methods,” in 2007 22nd international symposium on computer and information sciences, Ankara, Turkey, Nov. 2007, pp. 1–6, doi: 10.1109/ISCIS.2007.4456863.

[19] B. Kitchenham and S. Charters, Guidelines for performing Systematic Literature Reviews in Software Engineering. 2007.

[20] “Computer Science Conference Rankings CORE.” http://portal.core.edu.au/conf-ranks/.

[21] A. Idri, M. Hosni, and A. Abran, “Systematic literature review of ensemble effort estimation,” J. Syst. Softw., vol. 118, pp. 151–175, Aug. 2016, doi: 10/gf8kg7.

[22] E. Mendes, “Introduction to Effort Estimation,” in Practitioner’s Knowledge Representation, Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 27–53.

[23] S. Bibi and I. Stamelos, “Selecting the Appropriate Machine Learning Techniques for the Prediction of Software Development Costs,” in Artificial Intelligence Applications and Innovations, vol. 204, I. Maglogiannis, K. Karpouzis, and M. Bramer, Eds. Springer US, 2006, pp. 533–540.

[24] S. M. Satapathy and S. K. Rath, “Empirical assessment of machine learning models for agile software development effort estimation using story points,” Innov. Syst. Softw. Eng., vol. 13, no. 2–3, pp. 191–200, Sep. 2017, doi: 10/gfrgkv.

[25] L. L. Minku and X. Yao, “Ensembles and locality: Insight on improving software effort estimation,” Inf. Softw. Technol., vol. 55, no. 8, pp. 1512–1528, Aug. 2013, doi: 10/f43mtj.

[26] L. C. Briand, T. Langley, and I. Wieczorek, “A replicated Assessment and Comparison of Common Software Cost Modeling Techniques,” in Proceedings of the 2000 International Conference on Software Engineering. ICSE 2000 the New Millennium, 2000, p. 10.

[27] A. Trendowicz and R. Jeffery, “Classification and Regression Trees,” in Software Project Effort Estimation, Cham: Springer International Publishing, 2014, pp. 295–304.

[28] K. Srinivasan and D. Fisher, “Machine learning approaches to estimating software development effort,” IEEE Trans. Softw. Eng., vol. 21, no. 2, pp. 126–137, Feb. 1995, doi: 10/dgd32s.

[29] L. C. Briand and I. Wieczorek, “Resource Estimation in Software Engineering,” in Encyclopedia of Software Engineering, J. J. Marciniak, Ed. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2002.

[30] S. Bibi, I. Stamelos, and L. Angelis, “Software cost prediction with predefined interval estimates,” in Proceedings 1st Software Measurement European Forum (SMEF’2004), 2004, p. 10.

[31] L. C. Briand, K. E. Emam, D. Surmann, I. Wieczorek, and K. D. Maxwell, “An Assessment and Comparison of Common Software Cost Estimation Modeling Techniques,” in Proceedings of the 1999 International Conference on Software Engineering (IEEE Cat. No.99CB37002), 1999, p. 10.

[32] A. Idri and S. Elyassami, “Applying Fuzzy ID3 Decision Tree for Software Effort Estimation,” IJCSI Int. J. Comput. Sci. Issues Vol 8 Issue 4 No 1 July 2011, vol. 8, no. 4, p. 8, 2011.

[33] S. Elyassami, “Investigating Effort Prediction of Software Projects on the ISBSG Dataset,” Int. J. Artif. Intell. Appl., vol. 3, no. 2, pp. 121–132, Mar. 2012, doi: 10/gfrgk3.

[34] E. Papatheocharous and A. S. Andreou, “A HYBRID SOFTWARE COST ESTIMATION APPROACH UTILIZING DECISION TREES AND FUZZY LOGIC,” Int. J. Softw. Eng. Knowl. Eng., vol. 22, no. 03, pp. 435–465, May 2012, doi: 10/gfrgkw.

[35] L. L. Minku and X. Yao, “A principled evaluation of ensembles of learning machines for software effort estimation,” in Proceedings of the 7th International Conference on Predictive Models in Software Engineering - Promise ’11, Banff, Alberta, Canada, 2011, pp. 1–10, doi: 10.1145/2020390.2020399.

[36] S. Amasaki and C. Lokan, “Evaluation of Moving Window Policies with CART,” in 2016 7th International Workshop on Empirical Software Engineering in Practice (IWESEP), Osaka, Japan, Mar. 2016, pp. 24–29, doi: 10.1109/IWESEP.2016.10.

[37] E. Mendes, N. Mosley, and S. Counsell, “Web Effort Estimation,” in Web Engineering, 2006, pp. 29–73.

[38] E. Mendes, “Web Cost Estimation and Productivity Benchmarking,” in Software Engineering, vol. 5413, A. De Lucia and F. Ferrucci, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 194–222.

[39] A. L. I. Oliveira, P. L. Braga, R. M. F. Lima, and M. L. Cornélio, “GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation,” Inf. Softw. Technol., vol. 52, no. 11, pp. 1155–1166, Nov. 2010, doi: 10/dw8fh3.

[40] S. Amasaki and C. Lokan, “The Effect of Moving Windows on Software Effort Estimation: Comparative Study with CART,” in 2014 6th International Workshop on Empirical Software Engineering in Practice, Osaka, Japan, Nov. 2014, pp. 1–6, doi: 10.1109/IWESEP.2014.10.

[41] E. Papatheocharous and A. S. Andreou, “Classification and Prediction of Software Cost through Fuzzy Decision Trees,” in Enterprise Information Systems, 11th International Conference, ICEIS 2009, Milan, Italy, May 6-10, 2009. Proceedings, Berlin, Heidelberg, 2009, vol. 24, pp. 234–247, doi: 10.1007/978-3-642-01347-8_20.

[42] S. Bibi, I. Stamelos, and L. Angelis, “Combining probabilistic models for explanatory productivity estimation,” Inf. Softw. Technol., vol. 50, no. 7–8, pp. 656–669, Jun. 2008, doi: 10/cw2x8c.

[43] A. S. Andreou and E. Papatheocharous, “Software Cost Estimation using Fuzzy Decision Trees,” in 2008 23rd IEEE/ACM International Conference on Automated Software Engineering, L’Aquila, Italy, Sep. 2008, pp. 371–374, doi: 10.1109/ASE.2008.51.

[44] S. Elyassami and A. Idri, “Evaluating software cost estimation models using fuzzy decision trees,” Recent Adv. Knowl. Eng. Syst. Sci. WSEAS Press, p. 6, 2013.

[45] P. L. Braga, A. L. I. Oliveira, and 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(ICTAI 2007), Patras, Greece, Oct. 2007, pp. 181–185, doi: 10.1109/ICTAI.2007.172.

[46] E. Mendes, “An Overview of Web Effort Estimation,” in Advances in Computers, vol. 78, Elsevier, 2010, pp. 223–270.

[47] E. Mendes, Cost Estimation Techniques for Web Projects. IGI Global, 2008.

[48] K. Dejaeger, W. Verbeke, D. Martens, and B. Baesens, “Data Mining Techniques for Software Effort Estimation: A Comparative Study,” IEEE Trans. Softw. Eng., vol. 38, no. 2, pp. 375–397, Mar. 2012, doi: 10/bb2pkm.

[49] E. Kocaguneli, “Combining Multiple Learners Induced on Multiple Datasets for Software Effort Prediction,” in Proceedings of International Symposium on Software Reliability Engineering (ISSRE), 2009, p. 7.

[50] I. Myrtveit and E. Stensrud, “Do arbitrary function approximators make sense as software prediction models?,” in 12 International Workshop on Software Technology and Engineering Practice (STEP’04), Chicaco, IL, USA, 2004, pp. 3–9, doi: 10.1109/STEP.2004.9.

[51] L. C. Briand and J. Wust, “Modeling development effort in object-oriented systems using design properties,” IEEE Trans. Softw. Eng., vol. 27, no. 11, pp. 963–986, Nov. 2001, doi: 10/dmfwsn.

[52] Z. Abdelali, M. Hicham, and N. Abdelwahed, “An Ensemble of Optimal Trees for Software Development Effort Estimation,” in Smart Data and Computational Intelligence, vol. 66, F. Khoukhi, M. Bahaj, and M. Ezziyyani, Eds. Cham: Springer International Publishing, 2019, pp. 55–68.

[53] S. Elyassami and A. Idri, “Fuzzy model for an early estimation of software development effort,” Int. J. Appl. Math. Inform., vol. 7, no. 3, p. 9, 2013.

[54] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten, “The WEKA data mining software: An update,” SIGKDD Explor Newsl, vol. 11, pp. 10–18, 2008, doi: 10/cnvc45.

[55] I. H. Witten and E. Frank, Data mining: practical machine learning tools and techniques with Java implementations. San Francisco, Calif: Morgan Kaufmann, 2000.

[56] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA Data Mining Software: An Update,” SIGKDD Explor Newsl, vol. 11, no. 1, pp. 10–18, Nov. 2009, doi: 10/cnvc45.

[57] L. Song, L. L. Minku, and X. Yao, “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, New York, NY, USA, 2013, pp. 9:1–9:10, doi: 10.1145/2499393.2499394.

[58] M. M. Al Asheeri and M. Hammad, “Machine Learning Models for Software Cost Estimation,” in 2019 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Sakhier, Bahrain, Sep. 2019, pp. 1–6, doi: 10/ggjgkb.

[59] S. M. Satapathy and S. K. Rath, “Empirical Assessment of Machine Learning Models for Effort Estimation of Web-based Applications,” in Proceedings of the 10th Innovations in Software Engineering Conference on - ISEC ’17, Jaipur, India, 2017, pp. 74–84, doi: 10.1145/3021460.3021468.

[60] M. Azzeh, “Software effort estimation based on optimized model tree,” in Proceedings of the 7th International Conference on Predictive Models in Software Engineering, Banff, Alberta, Canada, Sep. 2011, pp. 1–8, doi: 10/dxfq6p.

[61] E. Mendes, “A Comparison of Techniques for Web Effort Estimation,” in First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007), Madrid, Spain, Sep. 2007, pp. 334–343, doi: 10.1109/ESEM.2007.14.

[62] S. Guillaume, B. Charnomordic, and J.-L. Lablee, “FisPro:Logiciel open source pour les systemes d’inference floue.” INRA-Cemagref, 2002, [Online]. Available: http://www.inra.fr/bia/M/fispro.

[63] S. Dan and P. Colla, “CART: Tree-Structured NonParametric Data Analysis.” San Diego, CA: Salford Systems, 1995.

[64] Salford-Systems, “Frequently Asked Questions and Answers about TreeNet.” [Online]. Available: http:// www.salfordsystems.com/doc/TreeNetFAQ.pdf.

[65] J. W. Keung, “Theoretical Maximum Prediction Accuracy for Analogy-Based Software Cost Estimation,” in 2008 15th Asia-Pacific Software Engineering Conference, Dec. 2008, pp. 495–502, doi: 10.1109/APSEC.2008.43.

[66] M. J. Shepperd and C. Schofield, “Estimating Software Project Effort Using Analogies,” IEEE Trans Softw. Eng, vol. 23, pp. 736–743, 1997, doi: 10/bmqpcq.

[67] T. Foss, E. Stensrud, B. Kitchenham, and I. Myrtveit, “A simulation study of the model evaluation criterion MMRE,” IEEE Trans. Softw. Eng., vol. 29, no. 11, pp. 985–995, Nov. 2003, doi: 10/dx5mx6.