Available online at: https://ijact.in

Date of Submission	03/10/2019
Date of Acceptance	11/11/2019
Date of Publication	30/11/2019
Page numbers	3462-3471(10 Pages)

Cite This Paper: Manpreet Kaur, S.Sood. Web Effort estimation technique: A systematic literature review, 8(11), COMPUSOFT, An International Journal of Advanced Computer Technology. PP. 3462-3471.

ISSN:2320-0790

An International Journal of Advanced Computer Technology

WEB EFFORT ESTIMATION TECHNIQUES: A SYSTEMATIC LITERATURE REVIEW

Manpreet Kaur[1], Sumesh Sood[2]

[1]Research Scholar, IK Gujral Punjab Technical University, Kapurthala, Punjab, India.
manpreet161183@gmail.com
[2]Assistant Professor, IK Gujral Punjab Technical University, Dinanagar Campus, Punjab, India

Abstract: Web Effort Estimation is an important estimation measure for predicting the effort required to develop a web application. The completion of web projects within stipulated time and budget is not possible without accurate effort estimation. The numerous effort estimation models are present these days and they have achieved a pinnacle of success, but the uncertainty features are daunting its progress due to deviations in the data set collected, types of projects, and data set characteristics. The literature studied for this research task elaborated that this field still lacks in a significant direction for consolidated documentation, which guides the researchers to choose a specific technique in order to predict the effort required for web application development. The wide and versatile nature of this domain daunting the researchers to mine the literature in a more appropriate way and deploy ensemble techniques of effort prediction models in order to achieve better results for web application viz., schedule delays, budget overruns. The systematic literature review (SLR) in this research task has been done to inspect the various aspects affecting the prediction accuracy of web applications and these identified characteristics lead to a better effort estimation model. The literature review is conducted on a collection of 143 papers retrieved from online journals and conference proceedings. Only 53 relevant papers are selected for broad investigation. The study reveals that the expert judgment and algorithm-based models are very popular and used frequently for effort prediction, instead the machine learning (ML) based models are rare in use but cater comparatively better prediction accuracy. The authors suggest taking cognizance of this research domain for developing ensembles of early effort prediction models to overcome delays in schedule and budget.

Keywords: Effort Estimation, Machine Learning, Web applications, Algorithmic, Expert Opinion.

I. INTRODUCTION

Approximation of the development effort is an important management activity for planning and monitoring software development projects [1]. Effort estimation consists of anticipating how many hours of work and how many persons are required in a team to develop a project [2]. A successful software project management predicts the effort and its use to ascertain prices as well as allocation of resources effectively and lead projects to be handed over within budget and stipulated time [3]. A survey on Web-based projects, published by the Cutter Consortium in 2000, revealed that a large number of web-based projects are delivered with schedule delay (about 79%), budget overruns (approx. 63%), lack of needed functionality (almost 53%), lack of core requirements (over 84%) and suggested that software development is different from web application development in a certain way. The literature reveals that there are many reasons behind the completion of web projects with unexpected cost, backlog of schedule and budget overruns. Nowadays, the competition in the web application development market is growing day by day. During the web project development process, the customer's request for frequent changes in their requirements according to the market trends, which results in a delay in the completion of projects. Sometimes, that delayed web project serves no purpose to the market when compared to the growing technology of hardware/software. It is necessary to deliver the web projects within the stipulated time period together with an eye on upgrading technology otherwise this may lead to an obsolete web project. The productivity growth of web project development companies declines if their project got obsolete and affects their reputation.

The motivation for this research task has been grabbed from the above said challenges and reveals the necessity of methods for web application development effort estimation (WADEE). The WADEE is a striking field of investigation and paramount important chore in web project management. There are many methods present that have been employed to resolve the assessment problems, but the progress still lacks due to the unavailability of systematic documentation for WADEE.

The effort estimation models which are helpful for conventional software development are not extremely explicit for effort estimation of web project development. It is determined that there is no documentation that provides counsel to the analysts to use a specific and most appropriate model to predict effort. This work looks forward to explore research a gap in the literary works.The inquiry needs to be scrutinized further to develop innovative effort estimation models with the aim of warding off the cost and budget overruns of web projects.

Under the literature survey, this paper explores various effort estimation models proposed by different researchers. Mainly, effort estimation methods are categorized as Expert Judgment, Algorithmic, and Machine learning (ML) based methods. Expert Judgment has been widely used. However, the means of deriving an estimate are not explicit and therefore not repeatable. Algorithmic models to date the most popular in the literature, attempt to represent the relationship between effort and one or more project characteristics. Examples of algorithmic models are the COCOMO model by Boehm, 1981 and the SLIM model by Putnam, 1978.

It is seen in the literature that various algorithms-based effort estimation methods have been compared with non-algorithmic methods [4-6]. In recent years, machine learning-based method has received great attention in web application development effort estimation (WADEE) research. Various machine learning techniques based software effort estimation models have been proposed and compared with existing effort estimation models [5-8]. It is noticed that non-ML models lead to poor prediction accuracy.

Generally, literature reviews are divided into two categories: traditional literature review and systematic literature review (SLR) [5] [9]. A traditional review mainly covers the recent research trends, whereas the SLR's aim answers various research questions pertaining to web application development effort estimation. The objective of SLR is not just to aggregate all existing evidence on a research question. It is also intended to support the development of evidence-based suggestions for practitioners [10]. The intent of this paper is to review the literature in a well-ordered manner and find the facts in the literature to reinforce the proposed research questions specifically for web applications.

There are mainly three phases involved in a systematic review: planning the review, conducting the review, and reporting the review [15]. The SLR adopts the following sequence of steps:

1. Outline the research questions.
2. Perform a pilot study on the selected studies.
3. Explore the online repositories (IEEE, ACM, Google Scholar, CiteSeer) referring to search key terms.
5. Evaluation and selection of relevant studies.
6. Assessing and exhibiting the results.
7. Investigate the generalize ability of the conclusions and limitations of the review.
8. Get recommendations for exercise.

This review explores how the characteristics of datasets for web projects influence its effort. The effects of using different size measures and prediction accuracy measures have also been investigated. This paper focuses on identifying and summarizing the results to date of this research domain. It also presents recent trends in estimating the effort in the subject area. The SLR identifies research gaps in the surviving literature. The paper is organized into five sections. Section II presents the traditional literature review. Section III explains the methodology employed to conduct the SLR. Section IV details the results and discussion of the obtained results. Further, section V presents the conclusion, whereas section VI contains the future scope.

II. LITERATURE REVIEW

Many existing studies related to web effort estimation compare either ML-based models with non-ML models or ML models with other ML models. The literature from the last two decades, i.e. 2001-2019 is considered for investigation.

Mendes et al. in [4] compare the performance of the analogy method with two algorithmic models - linear regression and stepwise multiple regression to estimate the authoring effort of Web applications. Results suggest that estimation by analogy is a superior technique.

Mendes and Mosley in [6] prove that the use of simpler models such as median effort outperforms more complex models, such as Bayesian Networks.

Mendes, E., Mosley, N., & Watson, I. in [11] compare the prediction accuracy of three CBR techniques to estimate the effort to develop Web hypermedia applications. The prediction accuracy of the best CBR technique is then compared against other ML-based prediction models, namely multiple linear regression, stepwise regression, and regression trees. Mendes et al., in [12] compare the prediction accuracy of the best CBR technique, i.e. *Weighted Euclidean similarity measure*, against three prediction models, multiple linear regression, stepwise regression, and regression trees. The accuracy measures MMRE and MdMRE show better prediction accuracy for Multiple Regression, whereas box-plots show better prediction for CBR. This study reveals that the use of different prediction accuracy measures also affects the effort estimated.

Hooi et al., in [13] propose a parametric based web effort estimation model (WEBMO) particularly for IT industry of Klang Valley, Malaysia. Results suggest that WEBMO does not provide a significant contribution towards effort estimation, as it is not successful in delivering web projects within stipulated *time* and *cost* to the industry. Thus, it is shown from previous studies that ML-based effort estimation models outperform non-ML models [4-6].

Araujo et al. in [14] proposed a hybrid machine learning method called modified Genetic algorithm (MGA) to optimize the parameters and to select an optimal input feature subset of the database. MGA was compared with popular classical machine learning models and resulted in higher accuracy of the software development cost estimation. Bardsiri et al. in

[15]demonstrated that analogy based estimation (ABE) is unable to produce an accurate estimation when the importance level of project features is not the same or the relationship among features is difficult to determine. The authors proposed a hybrid estimation model with the combination of ABE and bio-inspired algorithm, particle swarm optimization (PSO). Developing hybrid models significantly improve the performance of existing estimation models. The authors in [16] highlighted the capability of neural network-based models for the purpose of software effort estimation. Neural network-based models are competitive to traditional regression and statistical models. 21 articles were taken under review.

Urbanek et al. in[17] employed different prediction accuracy measures as fitness functions and compared their performance. The results show that the Mean square error (MSE) accuracy metric performed best and recommended as a fitness function for machine learning algorithms. Sanchan et al. in [18] proposed a simplified genetic algorithm (SGA) to assess the prediction of software projects and compared its performance against an algorithm based Basic COCOMO model. The SGA gives better realistic estimates. Reza et al. in [19] employed different machine learning techniques such as Stochastic Gradient Boosting (SGB) and four SVR kernel techniques, on the ISBSG dataset to develop effort estimation models for web-based projects using IFPUG Function Point approach. Results show that SVR techniques exhibit better results than other machine learning techniques.

As per the current documentations, many researchers have proposed the improvements of machine learning algorithms. Satapathy and Rathin [8] employed variations of Stochastic Gradient Boosting (SGB)and four Support Vector Regression (SVR) kernels i.e. SVR Linear, SVR Polynomial, SVR RBF, SVR sigmoid kernels on ISBSG Release 12 dataset to predict the effort. It is evident that SVR RBF kernel exhibits better results than other ML techniques for both new and enhanced web projects. Also, the extension of this approach is suggested by the researcher. Zare et al. in [20] proposed an effort estimation model based on optimal control using genetic algorithm (GA) and Particle Swarm Optimization (PSO) for the COCOMO NASA database. Results show that effort estimation models based on genetic algorithm (GA) outperformed other methods.

Minku et al. in [21] investigated that SEE models can be used for decision-support by software managers to determine the effort required to develop a software project. The authors proposed an approach to choose the most useful past models among within-company and cross-company models in order to improve the effort of software projects. Resmi et al., in [22] proposed a hybrid process consisting of fuzzy analogy and the nature-inspired Firefly algorithm resulting in improved accuracy. Moosavi et al. in [23] presented a new model based on a combination of adaptive neuro-fuzzy inference system and satin-bower bird algorithm. The authors compared the proposed model with other bio-inspired algorithms, resulting in better prediction accuracy. Satapathy et al. in [24] compared decision tree, stochastic gradient boosting, and random forest using story points.

Pospieszny et al. in [25] proposed an ensembles averaging three machine learning algorithms support vector machine, neural networks, and generalized models and provided a decision support tool for effort and duration estimation. Usman et al. in [26] highlighted the capability of Expert Judgment based effort estimation technique by developing and using checklists to understand the important factors affecting the estimation process in a better way. Singh et al. in [27] employed an evolutionary algorithm named Environmental adaptation method thus giving better results. Usman et al. in [28] explored various factors affecting EE such as team maturity, distribution as well as requirements size and priorities. Abrahao et al. in [29] proposed a size measurement procedure OOHCFP and compared with existing size metric OOHFP to estimate the effort of web applications. In [30] , Two types of Support vector regression are employed to predict software enhancement effort and compared with other machine learning techniques. Polynomial SVR proved to be better.

Zakrani et al.in [31] investigated the use of Random Forest (RF) in software effort estimation. It is compared with that of classical regression trees (RT). RF outperformed RT.Lorko et al. in [32] explored how the duration of the project play a role in the estimation process.

COMPARISON OF PUBLICATIONS

Research Article	**Research Domain**	**Prediction method/ Measurement techniques**	**Datasets Employed (Projects #)**	**Obtained Results**
(Araujo et al., 2012)[14]	Software Development Effort Estimation	Hybrid model=modified genetic algorithm (MGA)+ classical machine learning models vs. (SVR-Linear, SVR-RBF, Bagging, GA-based with the SVR-Linear, GA-based with SVR-RBF and MRL	NASA Desharnais, COCOMO, Albrecht, Kemerer, Kotengray	MRLHID better and more consistent. Leads to improvement: 0.6% NASA, 11% Desharnais, 12% COCOMO,8% Albrecht, 12% Kemerer, 0.2 % KotenGray
(Bardsiri et al., 2013)[15]	Software Development Effort Estimation	Hybrid estimation model=PSO+ABE	IBM data processing services (24), Canadian Financial organization (21), ISBSG 2014(5052)	Hybrid methodology works better
(Dave et al., 2014)[16]	Software Development Effort Estimation	Neural network-based models	Explores the relationship between input parameters and the effort.	Effort varies with input parameters.
(Urbanek et al., 2015)	Software Development Effort	Analytical programming method +differential evolution algorithm to calibrate use case points	Poznan University of Technology dataset	-Analytical programming improves estimation accuracy. Mean Square Error(MSE) most suitable prediction

[17]Springer	Estimation	method.	(24)	accuracy measure as fitness function.
(Sanchan et al., 2016) [18]	Software Development Effort Estimation	Simplified Genetic Algorithm vs. Basic COCOMO	NASA (18)	GA tuned parameters give improved estimation
(Zare et al., 2016)[20]	Software Development Effort Estimation	A Bayesian Belief network based on COCOMO and controlled by Genetic Algorithm (GA) and Particle swarm optimization	NASA (20)	EE model using Genetic algorithm gives prediction accuracy.
(Minku et al., 2017)[21]	Software Effort Estimation	Developing and using checklists	Infoway –Brazil,Diyatech-Pakistan Tsoft –Norway	Little documentation, light-weight process.
(Resmi et al., 2017) [22]	Software Development Effort Estimation	Analogy + Firefly Algorithm vs. Multilayer perceptron v s. Analogy based estimation vs. Fuzzy analogy	NASA 93, NASA 60, COCOMO 81 and deshnaris datasets.	The proposed hybrid model performs best.
(Abrahao et al., 2018)	Web application	OOHCFP vs. OOHFP	Spanish web company (30)	OOHCFP better
(Moosavi et al.,2018) [23]	Software Development Effort Estimation	Hybrid model=Satin bowerbird optimization algorithm (SBO)+Adaptive neuro-fuzzyinference system (ANFIS) vs. ANN,CART,MLR,SWR	ISBSG R11, Kemerer datasets	Hybrid model
(Satapathy et al., 2017)[24]	Software Development Effort Estimation	Story Points using Decision tree vs. stochastic Gradient Boosting vs. Random Forest	21 software projects based on Story Points	Technique with story points using SGB shows better prediction.
(Popsiezny et al., 2018) [25]	Software Development Effort Estimation	Proposed model=Support Vector Machine (SVM) + Multi-Layer Perceptron (MLP) + Generalized linear models (GLM)	ISBSG R13 (6006)	Hybrid model improves prediction accuracy.
(Singh et al., 2018)[27]	Software Development Cost Estimation	Random Forest (RF) vs. Regression trees(RT)	ISBSG R8, Tukutuku, COCOMO	Random Forest
(Floriano et al., 2018)[30]	Software enhancement effort	Supervised learning used for parameter tuning	ISBSG R11	Better prediction accuracy in case of supervised learning algorithm.

III. RESEARCH METHODOLOGY

Systematic reviews suggest three phases: planning the review, conducting the review and reporting the review.

A. *Planning*

In parliamentary law to lead a systematic literature review, the research questions play a significant role in deciding the search scheme, data extraction, and analysis. The search process constructs the search string based on questions to be replied. The extraction of required information from the literature is also achieved according to the set research questions. The results are then assessed by analyzing the collected data.

1) Research Questions

The research questions were identified and structured with the help of the PIOC (Population, Intervention, Outcome, and Context) criteria [35].

TABLE 1: DESCRIPTION OF PIOC CRITERIA

Population	Web applications/Web hypermedia
Intervention	Methods/techniques for effort estimation
Final result	Effort estimation methods, prediction accuracy, the effectiveness of successful effort estimation methods/techniques
Context	Encloses academia as well as the software industry. All cases of empirical studies, including observation, interviews, questionnaires, experiments, case studies, and systematic reviews.

As a consequence, the research questions to be accosted in this systematic review are identified as follows:

RQ1: What does literature reveal about various effort estimation methods employed for web applications?

a) What techniques exist to predict effort specifically for web development projects?

b) What prediction techniques are described to be more efficient and successful than the other?

c) Which resource facets (cost/development effort, size, maintenance, or quality) of a web application have been employed for assessing the effort?

d) Which prediction accuracy measures have been used as evaluation criteria?

RQ2: What are the properties of data sets used in the research (either academic data collected by students or industrial data submitted by professionals).

RQ3: Does early effort estimation models exist to calculate the effort during the former stages of web application development?

B. *Conducting the Review Phase*

This phase explains how the SLR is being conducted. Search process constructs the search string in accordance with Kitchenham and Charters [35] guidelines. The search operation results in innumerable literature studies after mining prominent online repositories for the hunting terms. The relevant studies that meet the inclusion and exclusion criteria are chosen. A quality appraisal is then practiced on the selected studies.

1) Search Strategy

The main search terms employed in this systematic review are derived from the research questions. The alternative spellings and synonyms are explored using Thesaurus as shown in Table II. A search string is constructed by concatenating the search terms, alternative spellings, and synonyms using the Boolean OR and Boolean AND operator as given in Table III.

TABLE II: LIST OF SEARCH TERMS

Web	Web application development, Web project, Web hypermedia application, Web metrics, Web size measures, Web Engineering
Effort	The effort, early effort, cost, development
Estimation	Estimation, early estimation, prediction, assessment
Method	Process, techniques, model, system, measurement

Based on Table II, a complete search string used to explore relevant literature is presented in Table III.

TABLE III: CONCATENATION OF SEARCH TERMS AND SYNONYMS USING BOOLEAN "OR" AND BOOLEAN"AND" OPERATOR

(Web OR hypermedia) AND (application OR method OR process OR system OR technique OR methodology OR procedure) AND (cost OR effort OR early effort OR development) AND (estimation OR early estimation OR prediction OR assessment)

The search based on the derived search string is performed on online databases, namely IEEE Xplore, Springer Link, Science Direct, and ACM Digital Library. Many prominent journal articles, workshop papers, and conference papers between the time period 1999-2019 are extracted. The search is applied to full-text to avoid exclusion of papers that do not include the keywords in the title or abstract but are even relevant to the review.

2) Search Process

The search process begins once the search terms have been identified. The main source of the online database used in this search process is from those available online databases and studies published from 1999-2019.Table IV presents the summarized search result of each online database:

TABLE IV: SEARCH RESULT OF EACH ONLINE DATABASE

Online database	Total Search
ACM Digital Library	10
IEEE Xplore	61
Elsevier Science Direct	30
Springer Link	19
Others(Cite seer, Research gate, Google Scholar)	23
Total	143

3) Study Selection

The investigation of database repositories gives rise to 143 candidate papers. The appropriate studies are chosen on some conditions described in inclusion and exclusion criteria.

Inclusion Criteria

RQ1a. Studies related to effort estimation techniques specifically for web applications are included.

RQ1b. Studies related to comparison among effort estimation techniques are accepted.

RQ1c. Studies related to the estimation of any resource facets i.e. cost, effort, size, maintenance or quality is included.

RQ1d. Studies that evaluate the performance of EE models using prediction accuracy measures.

RQ2.Studies that validates the EE models using some relevant dataset either Academic or Industrial.

RQ3. Studies that propose models for early effort estimates in web applications.

Only peer-reviewed papers described in the English language are preferred.

Exclusion Criteria

RQ1a. Works related to effort estimation techniques are excluded, which feature effort estimation techniques for traditional software and web hypermedia applications.

RQ1b. Subjects related to web metrics, literature reviews, web application design techniques or requirements methodology is left out.

RQ1c. Works that do not include the appraisal of development effort are kept out.

RQ1d. Subject areas that do not measure the performance of EE models using prediction accuracy measures are omitted.

RQ2. Works that do not include validation of EE models using some relevant dataset are excluded.

RQ3. Subject areas that do not propose models for early effort estimates are taken out.

The screening for selection of literature is composed of two stages: primary and secondary. Initially, the title and abstract of all papers are examined in the main stage of the search. The specific study is accepted for further investigation if found in conformity with the inclusion and exclusion criteria. The subject areas that do not meet the inclusion-exclusion criteria are rejected. As a consequence, 89 papers are outlined in Table V.

TABLE V: STUDIES AFTER READING TITLE AND ABSTRACT

Paper ID of Selected Articles	*Excluded Articles*
P10-11,P13-17,P19-30,P32-40, P42-49,P52-53,P55,P57,P61,P63-67,P69,P71-85,P91-94,P95,P99,P101-103,P105-107,P109,P111-112,P114-118,P120-122,P125,P128,P130,P132,P141.	P1-9,P12,P18,P31,P41,P46, P50,P51,P54,P56,P58-60,P62,P68,P70,P86-90,P92,P96-98,P100, P104,P108,P110,P113,P119,P123-124,P126-127,P129,P131,P133,P134-140,P142-144.
Total: 89	Total: 55

In the second stage, the papers are examined in full-text form and are critically evaluated based on the inclusion and exclusion criteria. A list of 83 out of 89 papers is shortlisted as shown in Table VI.

TABLE VI: STUDY SELECTION AFTER READING FULL-TEXT

Included Publications	*Excluded Publications*
P10,P13-16,P19-30,P32-40,P42-49,P52-55,P57,P61,P63-67,P69,P71-85,P91,P93-95,P99,P102-103,P105-107,P109,P111-112,P114-115,P117-118,P120--122,P125,P130,P141.	P11, P17, P101, P116, P128, P132
Total:83	**Total:6**

4) Study Quality Assessment

Once related literature has been selected, the authors needed to identify whether the paper was of quality or not to answer the research questions. To assess the quality of shortlisted papers, the guidelines defined by Kitchenham and Charters [35] were followed and developed a quality checklist of 6 questions to be answered for each shortlisted paper. The scale of 0-1 was used to calculate the literature quality: Y=1, N=0, and P=0.5. The higher the score, the greater its quality and any study which scored lower than 5.0 was removed. The following criteria were used in the checklist:

TABLE VII: QUALITY ASSESSMENT CHECKLIST

No	Question	Answer
1	Are the research objectives clearly indicated?	Y/N/P
2	Are the estimation techniques used clearly described?	Y/N/P
3	Are the data collected fully defined?	Y/N/P
4	Are the statistical techniques employed for the analysis of the date fully defined and their use justified?	Y/N/P
5	Is the study finding credible?	Y/N/P
6	Is there a discussion of any problems with the validity/reliability of their results?	Y/N/P

After performing the quality assessment, 30 papers were removed because of their low-quality score. Table IX shows the final study selection of a total of 53studies.

TABLE VIII: FINAL STUDY SELECTION AFTER PERFORMING THE QUALITY ASSESSMENT

Included Publications	**Excluded Publications**
P10,P14-16,P19-21,P23-24,P33-37,P45,P47-49,P53,P55,P57,P61,P63-65,P67,P69,P71-73,P75-77,P80-85,93-95,P99,P102,P105-107,P109,P112,P114,P118,P120,P122,P125,P130	P13,P22,P25-32,P38-40,P42-44,P46,P52,P66,P74,P78-79,P83,P91,P103,P111,P115, P117,P121,P141
Total:53	Total:30

5) Data Extraction

The data extraction phase retrieves the data from the selected studies using the format presented in Table IX format.

TABLE IX: EXTRACTION FORM

Data Item	*Value*
1. Study information data	Web application/Web hypermedia application/software project
2. Paper ID	P1-P144
3. Title	Title of the research paper
4. Author (s)	Names of all the authors
5. Year of Publication	The year in which paper has been published
6. Publication Type	Journal/Conference/Article
7. Publisher	A paper published in IEEE, Elsevier, ScienceDirect, etc.
8. Data relevant to answering Research questions	Tukutuku dataset/ISBSG dataset/dataset collected by students or researchers
9. Data Characteristics	Industry/Academia
10. What methods/techniques used for Effort estimation	CBR, SWR, LR, BN, SVR, CART, OLSR, etc.
11.Which prediction accuracy techniques have been used	MRE, MdMRE, Pred (25), Boxplots of residuals, Boxplots of Z,
12. Which size measures have been used?	Function points, web objects, COSMIC, COSMIC-FFP,etc.
13. What are the limitations of web effort estimation methods used?	Limitations, if any.

IV. RESULTS AND DISCUSSION

The third and last phase of a systematic literature review is about reporting the results attained after analyzing the data extracted from the survey publications.

C. Results Reporting

The information is extracted from all the selected publications to answer the relevant research questions in this third phase of results reporting.

Research Question 1a:

Data extraction for RQ1a mines the methods/techniques used to estimate the early effort in web application development project. Figure 3 shows the effort estimation methods/techniques that have been identified in this study, where the methods/techniques are classified into three approaches which are: expert-based approach, algorithm-based approach, and machine learning-based approach.

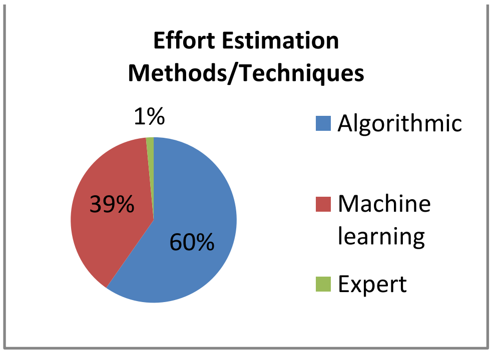

Fig.1: Effort Estimation Methods/Techniques

Fig. 1shows that the most common approach used is the algorithmic method which is 60%, followed by machine learning with 39% and expert-based with 1%. Commonly used Algorithm-based effort estimation methods for web applications are COCOMO, WebMO, mean and median-based. Mendes & Mosley [6] prove that the use of simpler models, such as median effort, outperforms more complex models, such as Bayesian Networks. It can be seen that the use of algorithmic methods to estimate the effort in developing web applications, is more frequent as compared to the other approaches.

Hooi & Yusoff [13] proposed a parametric based web effort estimation model (WEBMO) particularly for IT industry of Klang Valley, Malaysia. Results suggest that WEBMO does not provide a significant contribution towards effort estimation, as it is not successful in delivering web projects within stipulated *time* and *cost* to the industry. It is observed that machine learning-based effort estimation models outperform non-ML models [4-6].So, more research requires to be performed by using machine learning models in the research domain WADEE.

Research Question 1.b:

Data extraction for RQ1b examined the most used methods/techniques to estimate early effort for web applications. Table X shows the most used techniques in estimating the early effort from the study field.

TABLE X: MOST USED EFFORT ESTIMATION TECHNIQUES IN WEB PROJECTS

Estimation Methods/Techniques	*Paper ID*	*%age of usage*
Case-Based Reasoning(CBR)	P23,P34,P36,P44,P47,P65,P69,P71-73,P75,P82,P93,P95,P130.	28.3%
Stepwise Regression (SWR)	P14-16,P19,P21,P24,P33,P44,P47,P65,P69,P71-73,P75,P82, P95, P102.	33.9%
Linear Regression	P14-16,P19,P21,P24, P120, P130.	15%
Bayesian network(BN)	P63, P65,P72, P73,P82.	9.4%
Support Vector Regression SVR	P82,P95,P125	5.6%
REGRESSION TREE	P65	1.8%
Analogy-based Estimation	P15	1.8%
COSMIC method	P24,P61,P80	5.6%
Web-COBRA	P37,P81,P93	5.6%
Classification and regression Tree(CART)	P65	1.8%
Expert Opinion	P118	1.8%
OLSR	P37,P77,P80,P81,P93,P99,P102, P107	15.9%
Others: Allette system Informal Methods(S37), WebMO (S79), Simple function point (S49), Mean Estimation (S63, S73),Stochastic Gradient Boosting (S125), Median Estimation(S63,S73), Fuzzy Radial basis function neural network(S95), COSMIC +OOH (S83),WEBMO+ (S84), FHSWebEE (S107), OOmFPWeb (S107), OOHFP (S85)		

From the table, the most practiced techniques are Stepwise Regression (SWR) with 33.9%, followed by Case-based reasoning (CBR) with 28.3%.The rest of the methods such as linear regression, support vector regression, Web-COBRA, account for less than 15%.Machine learning-based effort estimation models significantly outperform non-ML models in the field of web-based projects specifically [4-6].

It can be seen from the literature [12] that no single machine learning-based technique can be considered as the most suitable effort estimation technique. Satapathy & Rath [8] employed different ML techniques such as Stochastic Gradient Boosting (SGB) and four Support Vector Regression (SVR) kernels. It is evident that SVR RBF kernel exhibits better results than other ML techniques for both new and enhanced web projects. However, it is seen that prediction accuracy varies widely depending on the type of project whether new or enhanced. Results also show that the ML-based approach is a more promising approach for effort estimation of web applications.

Research Question 1c:

Data collected for the RQ1c will provide information as to which areas of the Web resource estimation domain have been studied.

TABLE XI: RESOURCE FACET INVESTIGATED

Resource Facet	*Paper ID*	*%age of usage*
Design	P14,P16,P19,P20,P21,P24, P67	13.2%
Quality	P84	1.8%
Maintenance	NIL	0%
Size	P24,P33,P45,P49,P53,P61,P68,P77,P80,P81,P85,P93,P94,P99,P112,P120,P130	32%
Cost/Effort	All remaining studies	62%
Not Specified	P17	1.8%

It is observed from the Table XI that the majority of research in the field of Web resource estimation has focused on development Cost/Effort estimation, with 62% of the primary studies selected by this SLR. Design effort estimation is related to cost/effort estimation with just 13.2%, whereas 32% of studies related to size estimation show that size is regarded as the key determinant in the development of web applications. Size measure is the most prominent predictor, as almost every study related to WEE uses size measure in some sort.

Different Web size measures have been explored in literature, namely Source lines of Code (SLOC), Function Points (FPA), Web Objects by Reifer [34]. Prediction accuracy is not possible to evaluate in early phases of developing Web projects, using LOC as size measure, whereas FPA is able to predict effort on the basis of requirements only, collected from the user/client during the initial stage of development. It is seen that FPA can fail to capture some specific features of Web applications (Reifer, 2002 [34]. However, Web Objects introduced by Reifer [34] are confirmed as indicator of Web application development effort [36][37]. Certain effort estimation models OLSR, Web-COBRA and CBR use Web Objects as a size measure and provide statistically superior results as compared to the FPA method [37][38]. Rosmina & Suharjito [39] combined Functional size measurement of object-oriented web application named OOmFPWeb and Web metrics for web application size measurement. The evaluation results show that effort estimation for web application with the combined model is better than just using OOFPWebor web metrics.

It has been observed through this SLR that most research has mainly focused on development effort or cost, neglecting other resource facets like size, quality, and maintenance. Still, no work has been done in literature where effort estimation has encompassed more than one resource facet at a time.

Research Question 1d:

Data extraction for question 1d addresses the prediction accuracy measures used to evaluate Web effort estimation techniques as shown in Table XII. The absolute residual (a residual is a difference between the estimate and the actual value) forms the basis for all the numerical measures of accuracy. The MRE (magnitude of relative error) is calculated considering the absolute residual relative to the actual value. The mean and median MRE (MMRE and MdMRE), along with Pred (25) (the percentage of estimates with an MRE of 25% or less) are the most frequently used measures of accuracy, being seen in 56.6%, 32% and 49% of the primary studies respectively. Boxplots are a graphical representation of accuracy. Boxplots enable a visual comparison of different estimation techniques and may also help explain the values obtained by numerical accuracy measures.

TABLE XII: ACCURACY MEASURES USED IN EFFORT ESTIMATION

Accuracy Measure	Paper ID	%age of usage
MMRE	P14-15,P23,P33-34,P36-37,P44,P47-48, P63PS65,P69,P71-73,P75-77,P80-82,P84,P85,P93-95,P99,P112,P125	56.6%
MdMRE	P15,P23,P33,P47,P48,P63,P65,P69,P71-73,P75,P82,P93,P95,P99,P125.	32%
Pred(25)	P33-34,P36,P37,P44,P47-48,P63,P65,P69,P71-72,P75,P77,P79-80,P81-82,P84-85,P93-95,P99,P112,P125.	49%
Boxplots of Residuals	P14,P16,P19,P21,P23-24,P36-37,P44,P47,P69,P72,P75-77,P81-82,P85,P102,P112,P130.	40%
Boxplots of z	P72	18.8%
MEMRE	P73	18.8%
MdEMRE	P73	18.8%
Others	Standard Deviation(P84), Correlation (P84), Boxplots of MRE(P80,P130), Sum of absolute residuals(P73), Mean Absolute Residuals (P47,P48), Median Absolute Residuals (P47,P48,P107)	--

Research Question 2

Data collection for RQ2a examined the available dataset used to train the early effort estimation method. The dataset can be defined as data obtained from industry data or academic data. The purpose to review this question is to know and understand the relationship of the attributes that were being within the dataset. Table XIV summarizes the currently available dataset that was used.

From Table XIII, it can be seen that the most used datasets in the existing literature are Tukutuku dataset and Industry dataset. Both datasets differ in usage by only 1.5%. The academic dataset provided by students of different universities can be seen in 18.8% of papers, whereas 3.8% of the studies do not specify the dataset used.

TABLE XIII: DOMAIN OF THE DATASET USED

Dataset Type	*Paper ID*	*%age of usage*
Industry: Tukutuku	P33-34,P44,P47-48,P61,P63, P65,P72-73,P75-76,P82,P85, P95,P106,P114,P118.	34%
Industry: Web services provided company	P23,P49,P69,P71,P77,P79,P80-81,P84,P93-94,P99,P102, P109,P112,P120,P122,P125, P130,	35.8%
Academia-student projects	P14-16, P19, P21, P24, P36, P37, P84, P107.	18.8%
Not specified	P10, P17	3.8%

It is seen that empirical research has favored the use of industry datasets. Of these industry datasets, single-company datasets seem to produce superior estimates than cross-company datasets.

Research Question3:

The data extraction for RQ3 as exhibited in Table XIV reveals that 11.3% of studies measure the effort estimation on web applications in the early-stage development life cycle. Prediction accuracy also depends on using different Web size measures introduced in the literature, namely Source lines of Code (SLOC), Function Points (FPA), and Web Objects [40]. Prediction accuracy is not possible to pass judgment in early forms of developing Web projects, using SLOC as size measure, whereas FPA is able to predict effort on the basis of requirements only, collected from the user/client during the initial phase of development [40] [41].

TABLE XIV: STAGE AT WHICH EFFORT ESTIMATION DONE

Stage	*Paper ID*	*%age of usage*
Early	P23, P33, P35, P53, P112, P122.	11.3%
Late	P37,P49,P55,P57,P61, P63,P72,P73,P81,P82, P84,P93,P107,P114, P125, P130	30.2%
Not determined	All remaining studies	58.4%

It shows that there is an urgent need for models that perform effort estimation at the early stages of the development cycle.

V. CONCLUSION

The systematic review for effort estimation of web application development based on the techniques Viz., expert judgment (approx. 01%), algorithmic methods (approx. 60%) and machine learning (ML)(approx. 39%) has been conducted. In spite of several methods for effort estimation, the algorithmic models are deploying frequently but sometimes lack in predictions, whereas machine learning methods, although deploying rarely but cater better prediction accuracy. Still, every model struggles with certain restrictions such as attributes of the dataset (single-company / cross-company data, less messy / very messy, industrial / academic), different size measurement standards (SLOC, function points, COSMIC-FFP, etc.), and varying prediction accuracy measures. Due to this, it still lacks to contribute the combination of different resource facet (size, quality, and maintenance) along with the crossover of ML techniques together with algorithmic models for web effort estimation. Moreover, if the ML deployment together with fitness functions viz., MRE, MdMRE, and Boxplots of residuals will be done, it may lead to the improvement in the prediction accuracy of development effort.

Thus, researchers need to put more focus on ML in order to develop predictive models for evaluating the early effort of web applications.

VI. FUTURE WORK

Currently, a study to investigate the effectiveness of estimation methods web application development through ML Viz., Neural Networks, Random Forest, and Fuzzy logic are under process. ML deployment together with fitness functions viz., MRE, MdMRE, and Boxplots of residuals will be done in the future and its performance will be compared with the traditional effort estimation methods.

The future work is thus completely focused on the evaluation of improved machine learning-based early effort, estimation model.

ACKNOWLEDGEMENT

Authors are highly thankful to IK Gujral Punjab Technical University, Kapurthala, Punjab, India for giving the chance to conduct this research work.

REFERENCES

[1] Martino, S. D., Ferrucci, F., Gravino, C., & Mendes, E. (2007). Comparing Size Measures for Predicting Web Application Development Effort: A Case Study. *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)* (pp. 324-333). Madrid, Spain: IEEE.

[2] Moayed, M. J., Ghani, A. A., & Seyedzadegan, M. (2007). Comparing Between Web Application Effort Estimation Methods. *Fifth International Conference on Computational Science and its Applications (ICCSA 2007)* (pp. 153-160). Kuala Lumpur, Malaysia: IEEE.

[3] Mendes, E., Watson, I., Triggs, C., Mosley, N., & Counsell, S. (2003). A Comparative Study of Cost Estimation models for Web Hypermedia Applications. (L. Briand, Ed.) *Empirical Software Engineering*, 163-196.

[4] Mendes, E., N., M., & Counsell, S. (2001). Using an Engineering Approach to Understanding and Predicting Web authoring and Design. *In Proc. 34th Hawaii Int'l Conference on System Sciences.* IEEE.

[5] Wen, J., Li, S., Lin, Z., Hu, Y., & Huang, C. (2012). Systematic literature review of machine learning-based software development effort estimation models. *Information and Software Technology, 54*, 41-59.

[6] Mendes, E., & Mosley, N. (2008). Bayesian Network Models for Web Effort Prediction: A Comparative Study. *IEEE Transactions on Software Engineering, 34*(6), 723-737.

[7] Corazza, A., Martino, D. S., Ferrucci, F., Gravino, C., & Mendes, E. (2009). Applying Support Vector Regression for Web Effort Estimation using a Cross-Company Dataset. *Third International Symposium on Empirical Software and measurement* (pp. 191-202). IEEE.

[8] Satapathy, S. M., & Rath, S. K. (2016). Effort estimation of web-based applications using machine learning techniques. *Int'l Conference on Advances in Computing, Communications, and Informatics(ICACCI)* (pp. 973-980). Jaipur, India: IEEE.

[9] Kitchenham, B., Brereton, O. P., Budgon, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in Software Engineering- A systematic literature review. *Information and Software Technology*, 7-15.

[10] Kitchenham, B. (2004). *Procedures for performing systematic reviews.* NICTA Technical Report.

[11] Mendes, E., Mosley, N., & Watson, I. (2002). A Comparison of Case-Based reasoning Approaches to Web Hypermedia Project Cost Estimation. *Proceedings of the 11th international conference on World Wide Web* (pp. 272-280). Honolulu, Hawaii, USA: ACM.

[12] Mendes, E., Watson, I., Triggs, C., Mosley, N., & Counsell, S. (2002). A Comparison of Development Effort estimation Techniques for Web Hypermedia Applications. *Proceedings Eighth IEEE Symposium on Software Metrics.* Ottawa, Ontario, Canada, IEEE.

[13] Hooi, T. C., & Yusoff, Y. H. (2008). Comparative Study on Applicability of WEBMO in Web Application Cost Estimation within Klang Valley in Malaysia. *8th International Conference on Computer and Information Technology Workshops* (pp. 116-121). Sydney, QLD, Australia: IEEE.

[14] Araujo, Ricardo de A., Sergio Soares and L.I. Adriano Oliveira. "Hybrid morphological methodology for software development cost estimation." *Expert Systems with Applications* 39 (2012): 6129-6139.

[15] Bardsiri, Vahid Khatibi, et al. "A PSO-based model to increase the accuracy of software development effort estimation." *Software Quality* (2013): 501-526.

[16] Dave, V. S. and K. Dutta. "Neural network based models for software effort estimation: a review." *Artificial Intelligence* (2014): 295-307.

[17] Urbanek, T., et al. "Prediction accuracy measurements as a fitness function for software effort estimation." *SpringerPlus* (2015).

[18] Sachan, R. K., et al. "Optimizing Basic COCOMO Model using simplified Genetic Algorithm." *Twelfth International multi-conference on Information processing*. Elsevier ScienceDirect, 2016. 492-498.

[19] Reza, S. M., Rahman, M. M., Parvez, M. H., Kaiser, M. S., & Mamun, S. A. (2015). Innovative approach in Web Application Effort & Cost Estimation using Functional Measurement Type. *2nd Int'l Conference on Electrical Engineering and Information & Communication Technology(ICEEICT).* Dhaka, Bangladesh: IEEE.

[20] Zare, F., H. K. Zare and M. S. Fallahnezhad. "Software effort estimation based on the optimal Bayesian belief network." *Applied Soft Computing* (2016): 968-980

[21] Minku, L. L. and X. Yao. "Which models of the past are relevant to the present? A software effort estimation approach to exploiting useful past models." *Auton Software Engineering* (2017): 499-542.

[22] Resmi, V., Vijayalakshmi, S., Chandrabose, R. S. "An effective software project effort estimation system using optimal firefly algorithm". *Cluster Computing Springer* (2017)

[23] Moosavi, S. H. and V. K. Bardsiri. "Satin Bowerbird optimizer: A new optimization algorithm to optimize ANFIS for software development effort estimation." *Engineering applications of Artificial Intelligence* 60 (2017): 1-15.

[24] Satapathy, S. M. and S. K. Rath. "Empirical assessment of machine learning moedls for agile software development effort estimation using story points." *Innovations System Software* (2017): 191-200.

[25] Pospieszny, P., B. C. Chrobot and A. Kobylinski. "An effective approach for software project effort and duration estimation with machine learning algorithms." *The Journal of Systems and Software* 137 (2018): 184-196.

[26] Usman, M., Petersen, K.., Borstler, J., Neto, P.S. "Developing and using checklists to improve software effort estimation: A multi-case study." *The Journal of Systems and Software* (2018): 286-309.

[27] Singh, T., R. Singh and K K Mishra. "Software cost estimation using Environmental Adaptation method." *8th International conference on advances in computing and communication*. Elsevier ScienceDirect, 2018. 325-332.

[28] Usman, M.,Britto, R., Damm, L., Borstler, J. "Effort estimation in large-scale software development: An industrial case study." *Information and Software technology* (2018): 21-40.

[29] Abrahao, S., Marco, L. D., Ferrucci, F., Gomez, J., Gravino, C. "Definition and evaluation of a COSMIC measurement procedure for sizing web applications in model-driven development environment." *Information and Software technology* (2018) 144-161.

[30] Floriano, a. G., et al. "Support Vector regression for predicting software enhancement effort." *Information and Software Technology* (2018): 99-109.

[31] Abdelali, Z., H. Mustapha and N. Abdelwahed. "Investigating the use of random forest in software effort estimation." *Second International Conference on Intelligent Computing in data Sciences*. Els ScienceDirectevier, 2019. 343-352.

[32] Lorko, M., M. Servatka and Le Zhang. "Anchoring in project duration estimation." *Journal of Economic behavior and Organization* (2019): 49-65.

[33] Minku, L. Leandro. "A novel online supervised hyperparameter tuningn procedure applied to cross-company software effort estimation." *Empirical Software Engineering* (2019): 3153-3204.

[34] Reifer, D. (2002). Estimating Web Development Costs: There are differences. *Crosstalk*, pp. 13-17.

[35] Kitchenham, B., & Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering (version 2.3).* Software Engineering Group, School of Computer Science and Mathematics, Keele University and Department of Computer Science, University of Durham.

[36] Morisio, M., Stamelos, I., Spahose, V., & Romano, D. (1999). Measuring functionality and productivity in Web-based applications : A Case Study. *In: Proceeding of International Software metrics Symposium* (pp. 111-118). Los Alanitos: IEEE press.

[37] Martino, D. S., Ferrucci, F., Gravino, C., & Sarro, F. (2011). Using Web objects for development effort estimation of web applications. *Springer-Verlag Berlin Heidelberg*, 186-201.

[38] Ruhe, M., Wieczorek, I., & Jeffery, R. (2003). Cost estimation for Web applications. *Proceedings - International Conference on Software Engineering.* Research Gate.

[39] Rosmina, & Suharjito. (2012). FHSWebEE: An Effort Estimation Model for Web Application. *International Conference on Advances Science and Contemporary Engineering.50*, pp. 613-622. Elsevier, ScienceDirect.

[40] Reifer, D. (2000). Web Development: Estimating Quick time to Market Software. IEEE Software. *17*(6), 57-64.

[41] Marco, L. D., Ferrucci, F., Gravino, C., Sarro, F., Abrahao, S., & Gomez, J. (2012). Functional versus Design Measures for Model-Driven Web Applications:A Case Study in the Context of Web Effort Estimation. *WETSoM* (pp. 21-27). Zurich, Switzerland: IEEE.

[42] Oliveira, A., Braga, P. L., Lima, R. M., & Cornelio, M. L. (2010). GA-based method for feature selection and parameter optimization for machine learning regression applied to software effort estimation. *Information and Software Technology, 52*, 1155-1166.

[43] Lin, J. C., Chang, C. T., & Huang, S. Y. (2011). Research on Software Effort Estimation Combined with Genetic Algorithm and Support Vector Regression. *International Symposium on Computer Science and Society* (pp. 349-352). IEEE.

[44] Corazza, A., Martino, S. D., Ferrucci, F., Gravino, C., Sarro, F., & Mendes, E. (2011). Using Tabu Search to configure support vector regression for effort estimation. In T. Menzies, & G. Koru (Ed.), *Empirical Software Engineering.* Springer.

[45] T. K. Lee, K. T. Wei and A. A. A. Ghani, "Systematic literature review on effort estimation for Open Sources (OSS) web application development," *2016 Future Technologies Conference (FTC)*, San Francisco, CA, 2016, pp. 1158-1167. doi: 10.1109/FTC.2016.7821748.