AN ENSEMBLE MODEL TO IMPROVE EFFORT ESTIMATION ACCURACY FOR SOFTWARE DEVELOPMENT

YASIR MAHMOOD

A thesis submitted in partial fulfilment of the
requirements for the award of the degree of
Doctor of Philosophy

Razak Faculty of Technology and Informatics
Universiti Teknologi Malaysia

JANUARY 2022

DEDICATION

I am honoured to dedicate this thesis to my beloved parents, family, Dr. Ahmad Salman Khan and siblings for their encouragement, endless support and sacrifices throughout my journey towards the successful completion of the thesis.

ACKNOWLEDGEMENT

I am grateful to ALMIGHTY ALLAH for blessing me with the opportunity and strength to make the best possible contribution to the society.

I would like to express my heartfelt and sincere gratitude to my Main Supervisor Azri Azmi (Ts. Dr.) and Co-Supervisor Prof. Mohd Nazri Kama (Ts. Dr.) for their continuous guidance, encouragement and inspiration throughout this study. This research would not have been possible without their help and advice. I am thankful to Universiti Teknologi Malaysia (UTM) for providing me with all the necessary facilities required in carrying out this research.

There are no words to express gratitude to my loving parents for their support, and love. I also appreciate the sacrifices and patience of my wife (Mamoona), children (Muhammad Nafay, Mahnoor Yasir, Ahmad Hassan) and siblings. I would also like to give my deepest gratitude to Dr. Ahmad Salman Khan as it would not have been possible to successfully complete this research without his memorable guidance and assistance on various occasions.

In the end, my sincere appreciation also extends to all my colleagues, friends and everyone who has been a source of support and inspiration at various occasions for successful completion of this thesis.

Universiti Teknologi Malaysia, January 2022

Yasir Mahmood

ABSTRACT

In recent years, due to significant evolution in adopting new technologies and development methodologies in the field of software engineering, there is an increased requirement to have an accurate effort estimation model that can cater for the needs of the continually growing software industry. Accurate effort estimation model is an essential feature of software engineering for effective planning, controlling and on-time delivering quality software projects within budget. In the last few decades, several models and practices of estimating the software effort have evolved, but it is still an essentially unresolved problem. One of the main reasons for inaccuracy is due to ineffective use of estimation models. Nevertheless, there is no proven software estimation model that can be used continuously in various situations to accurately estimate the software effort. In software development, it is difficult to accurately estimate the amount of work required to develop a software system of which suitable estimation model is a major concern. The over-estimation may result in a lost bid while under-estimation may fail the project. Consequently, the inaccuracy in estimating the software effort may result in serious consequences for developers and customers; resulting in disappointment, inaccurate estimation and hence, contribute to either low-quality project, team frustration or cost overrun. The main aim of this research is to optimize the estimation accuracy performance of software development effort using an ensemble technique. In this research, a novel software effort predictive model is proposed in which it incorporates techniques such as 1) Use Case Points (UCP), 2) Expert Judgement, and 3) Case-Based Reasoning as base models to create an ensemble. In this model, a feature importance selection technique (Extra Tree Classifier) and K-Nearest Neighbour machine learning algorithm are applied to identify the most relevant features from the UCP benchmark dataset and to assess project similarity respectively. Finally, the effort of the individual base models is ensembled using linear combination methods. This research is conducted through primary (a multi-case study involving software companies and university students' projects), and secondary case studies to make an ensemble model. To show the accuracy, reliability and applicability of the proposed model, the software projects from primary studies as case selections are selected by applying a quantitative approach through experiments, industrial experts, archival data about estimates and evaluation metrics. The results of this research revealed that in comparison to UCP, expert judgement, and CBR techniques, the ensemble technique produced 15.9%, 14.6 %, and 14.6 % Mean Magnitude of Relative Error; 20.6 %, 14 %, and 1% Mean Magnitude of Error Relative; 10.94 %, 14.53 %, and 1.1 % PRED (25) accuracy improvement. The proposed ensemble model can be used by software development firms and practitioners as an instrument to accurately estimate the effort required to develop new software projects at an earlier stage.

ABSTRAK

Dalam beberapa tahun terakhir, kerana evolusi yang penting dalam menerima pakai teknologi baharu dan metodologi pengembangan di bidang kejuruteraan perisian, terdapat peningkatan keperluan untuk memiliki model anggaran usaha yang tepat yang dapat memenuhi keperluan industri perisian yang terus berkembang. Model anggaran usaha yang tepat adalah ciri penting kejuruteraan perisian untuk merancang, mengawal dan menyerahkan projek perisian yang berkualiti tepat pada waktunya dalam lingkungan belanjawan. Dalam beberapa dekad terakhir, beberapa model dan praktik menganggar usaha perisian telah berkembang, tetapi masih merupakan masalah yang tidak dapat diselesaikan. Salah satu sebab utama ketidaktepatan adalah disebabkan penggunaan model anggaran yang tidak berkesan. Walaupun begitu, tidak ada model anggaran perisian yang terbukti dapat digunakan secara berterusan dalam pelbagai situasi untuk menganggarkan usaha perisian dengan tepat. Dalam pembangunan perisian, sukar untuk menganggarkan secara tepat jumlah kerja yang diperlukan untuk membangunkan sistem perisian yang mana model anggaran yang sesuai menjadi perhatian utama. Anggaran yang berlebihan boleh mengakibatkan hilang tawaran manakala anggaran yang terkurang mungkin akan menggagalkan projek. Akibatnya, ketidaktepatan dalam menganggarkan usaha perisian boleh menghasilkan akibat yang serius bagi pemaju dan pelanggan; mengakibatkan kekecewaan, anggaran yang tidak tepat dan oleh itu, menyumbang kepada projek berkualiti rendah, kekecewaan pasukan atau kos yang berlebihan. Tujuan utama penyelidikan ini adalah untuk mengoptimumkan prestasi ketepatan anggaran usaha perisian dengan menggunakan teknik *ensemble*. Dalam penyelidikan ini, model ramalan usaha perisian baharu dicadangkan di mana ia menggabungkan teknik seperti 1) Gunakan Titik Kes (UCP), 2) Penghakiman Pakar dan 3) Penaakulan Berasaskan Kes sebagai model asas untuk mencipta *ensemble*. Dalam model ini, teknik pemilihan kepentingan (*Extra Tree Classifier*) dan algoritma pembelajaran mesin *K-Nearest Neighbor* digunakan untuk mengenal pasti ciri yang paling berkaitan daripada set data penanda aras UCP dan untuk menilai persamaan projek masing-masing. Akhirnya, usaha model asas individu digabungkan menggunakan kaedah gabungan linear. Penyelidikan ini dilakukan melalui kajian utama (kajian pelbagai kes yang melibatkan syarikat perisian dan projek pelajar universiti), dan kajian menengah untuk membuat model *ensemble*. Untuk menunjukkan ketepatan, kebolehpercayaan dan kebolehgunaan model yang dicadangkan, projek perisian daripada kajian utama sebagai pemilihan kes dipilih dalam menggunakan pendekatan kuantitatif melalui eksperimen, pakar industri, data arkib mengenai anggaran dan metrik penilaian. Hasil penyelidikan ini mendedahkan bahawa berbanding dengan UCP, pertimbangan pakar, dan teknik CBR, teknik *ensemble* menghasilkan 15.9%, 14.6 %, dan 14.6 % Purata Magnitud Ralat Relatif; 20.6 %, 14 % dan 1% Purata Magnitud Ralat Relatif; 10.94 %, 14.53 % dan 1.1 % PRED (25) peningkatan ketepatan. Model *ensemble* yang dicadangkan boleh digunakan oleh syarikat pembangunan perisian dan pengamal sebagai instrumen untuk menganggarkan dengan tepat usaha yang diperlukan untuk membangunkan projek-projek perisian baharu pada peringkat awal.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	-	Artificial Intelligence
ANN	-	Artificial Neural Networks
CA	-	Closet Analogue
CBR	-	Case Based Reasoning
COCOMO	-	COst COnstructive MOdel
DT	-	Decision Tree
EEE	-	Ensemble Effort Estimation
EF	-	Environmental Factor
GA	-	Genetic Algorithm
IRWM	-	Inverse Ranked Weighted Mean
MER	-	Magnitude of Error Relative
MMRE	-	Mean Magnitude of Relative Error
MRE	-	Magnitude of Relative Error
MSE	-	Mean Squared Error
PRED	-	Predict
RMSE	-	Root Mean Squared Error
SDLC	-	Software Development Life Cycle
SDP	-	Software Development Phase
SEE	-	Software Effort Estimation
SLIM	-	Putnam Software LIfe cycle Model
SLOC	-	Source Line of Code
SLR	-	Systematic Literature Review
SPM	-	Software Project Management
SVR	-	Support Vector Regression
SLR	-	Systematic Literature Review
TCF	-	Technical Complexity Factor
UAW	-	Unadjusted Actor Weight
UCP	-	Use Case Points
UUCP	-	Unadjusted Use Case Point
UUCW	-	Unadjusted Use Case Weight

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Overview

Software Effort Estimation (SEE) studies have started since the 1960s and continuous research has been conducted due to numerous claims on attaining accurate estimation results (Bardsiri et al., 2013; Lehtinen et al., 2014). In the planning phase of project management, SEE is an essential feature to deliver a successful software system. Software effort estimation is defined as a process of predicting the amount of work and hours required to develop software systems. It is typically measured in man-hours or man-months unit (Wen et al., 2012). Today, developing software systems are expensive and difficult. The software engineering presents several ways to quantify a project. One of the most important steps in software engineering process is to accurately estimate the cost, effort and time which has an important role in determining the success or failure of the project. The software development cost and effort estimation are important in development process and customer requirements. The reports on conducting projects show that there is almost no control over software projects and usually, the scale of the accomplished work is more than what has been estimated before. Therefore, usually projects terminate later than planned time (Jain et al., 2014).

In software engineering, managers would be able to estimate, forecast, and properly quote the needs for schedule, budget, and personnel to effectively finish software projects using effort estimation techniques. Delivering high-quality software to end users on time and on budget is still a big challenge for software project teams (Kerzner, 2018). The importance of the software project manager's involvement in the success or failure of a project has been underlined in several studies (Gupta and Kalia, 2017; Medina and Francis, 2015). According to Lehtinen et al. (2014), a failure of software project indicates recognizable cost, scope, effort, schedule, or quality failure.

According to the CHAOS report (2015) of The Standish Group International ("Standish Group International," CHAOS Report 2015), 60% of IT projects were not on their scheduled time and 56% were not on budget. The International Society of Parametric Analysis (ISPA) studied that inaccurately estimating the staff's skills level, underestimating software size and lack of requirement's understandings are some of the core reasons behind project failures (Eck *et al.*, 2009). To date, researchers have therefore introduced different types of SEE techniques. On the other hand, the majority of the techniques, were proposed at the start of the software development process, based on pre-defined requirements.

Software effort estimation is broadly divided into three main categories: 1) algorithmic, 2) expert estimation and 3) machine learning (Wen *et al.*, 2012). Considering these estimation techniques, the experts and practitioners proposed to develop numerous estimation methods for accomplishing high effort estimation accuracy and afterwards chosen just a single best method to utilize. However, there is no consensus between the research communities that concludes the best solo method. A new endeavours on ensemble estimation methods have been proposed (E. Kocaguneli *et al.*, 2012; Minku and Yao, 2013; Pai *et al.*, 2013). An ensemble effort estimation technique is defined as the combinations of more than one single technique to estimate software development effort of a new project using a combination rule i.e. mean, median, Inverse Rank Weighted Mean (IRWM) etc., (Seni and Elder, 2010). The estimation of each base model is combined that produced the estimation of an ensemble.

In this research, an ensemble model is proposed to improve the estimation accuracy of software development effort. It integrates approaches such as Use Case Points (UCP), expert judgement, and Case-Based Reasoning (CBR). In addition, this research examined a systematic review of studies on use case points and expert judgment-based software development effort estimation. Furthermore, a comparative study is conducted concerning effort estimation accuracy in solo and ensemble techniques. The estimation accuracy of the proposed model is evaluated by using projects from software development organizations and student projects as case studies in a quantitative manner that includes experiments, industrial experts, archival data on estimates, and evaluation criteria. Software development firms and practitioners will use the proposed model at the end of this research as an instrument to estimate the software effort.

The rest of this chapter is organized as follows: The problem background is presented in Section 1.2, which includes a brief explanation of the primary software engineering domains investigated in this study. Section 1.3 explains the problem statement. The research questions are presented in Section 1.4. Section 1.5 summarizes research objectives. The scope of the research and its significance are briefly described in Section 1.6 and 1.7 respectively. The operational definitions are presented in Section 1.8. Section 1.9 organizes the remaining chapters of this thesis.

1.2 Problem Background

Any software project's success depends primarily on its accuracy in estimating effort. To date, a lot of research has been conducted to estimate the accuracy of software effort using distinctive techniques. In any case, researchers and specialists are striving to recognize which estimation technique gives increasingly accurate outcomes on the given datasets and the other applicable attributes. The number of software projects fails due to incomplete requirements and inaccuracy in software estimation (Kaur and Sengupta, 2013). The Project Management Institute (PMI) conducted a survey in 2017, investigated that 69% of software successfully achieved the project's original goals and business priorities, 43% were not finished within their initial budgets, 48% were delivered late and 32% failed due to budget lost (PMI's., 2017).

The factors that influence effort and cost during the conception and design phases have been extensively researched, mostly using cost-estimating techniques. According to Doloi (2013), proper cost and effort estimation is the key to avoiding project cost overruns, regardless of management skill or financial strength of the contractor. Cost and effort estimation is a technical technique for predicting expenditures, and its success is dependent on the resources and project execution.

According to widely published initial estimates, project complexity, technology needs, and project team requirements are among the factors influencing cost performance.

A study of information systems managers and other information systems professionals of 596 software development team member’s data on various organizations to investigate the reasons behind in accuracy as well as suggestions and improvements to avoid these factors confirmed that information systems software cost and effort estimating is an important concern (Lamba, 2020). Table 1.1 shows the top ten causes of inaccuracy estimates indicated by the respondents.

Table 1.1 Top ten causes of inaccuracy estimates (Lamba, 2020)

No.	Causes of Inaccuracy	Mean
1	Less Training	4.9
2	Survival pressure in market	4.6
3	Different working methods within team	4.5
4	Complex environment	4.3
5	Unexpected maintenance work	3.9
6	Problems with development tools	3.6
7	Overlooked tasks	3.59
8	Predefined project cost requires	3.4
9	Issues with acceptance testing	3.3
10	Slow continuous integration feedback	3.1

An enterprise International Project Management Association ("International Project Management Association," 2019) conducted a survey of 100 software businesses across a broad cross section of industries. The results of the study show that implementing consistent governance supervision, focusing on managing benefits, and managing change throughout the project lifecycle is difficult. According to the survey, 70% of firms had at least one project failure in the preceding year. Moreover, half of the respondents said their initiative didn't always accomplish what it set out to do. Projects are expected to be delivered on time and on budget in 30 % and 36% of organisations, respectively. IBM-PMO ("IBM," 2019) consultants conducted a survey of 1,500 change management executives and investigated that 40% of projects reached their schedule, budget, and quality goals, with underestimating project complexity also listed as a challenge factor in 35% of projects.

In SEE literature, the researchers have proposed different models and techniques for accomplishing high effort estimation accuracy. Tronto *et al.* (2007), conducted a comparison of artificial neural networks (ANN) and regression models. ANN and regression analysis were applied on COCOMO dataset for estimating effort from size. The performance of both methods was compared and results revealed that ANN was effective in effort estimation. Wen *et al.* (2012), compared two machine learning techniques and found that it is more accurate than non-machine learning. They investigated that the mean of PRED (25) was 46% and MMRE was 51% for case-based reasoning (CBR) as compared to mean of PRED (25) = 64% and MMRE = 37% for ANN. Kamal and Ahmed (2011), performed a comparison of several UCP metrics and proposed a use case-based model using fuzzy logic. The results showed that the UCP method in machine learning techniques estimation approaches may bring significant impact on estimation accuracy. Azzeh and Nassif (2016), presented a hybrid model to predict UCP and productivity using the Radial Basis Neural Network (RBNN) and Support Vector Machine (SVM) which included prediction and classification stages. In this model, the historical productivity was clustered into fine-grain productivity using bisecting K-medoids algorithm clustering technique and then classified based on environmental factors. The results found that the use of RBNN shows significant improvement for effort estimation. They also investigated that the environmental complexity factor (ECF) may be removed from the estimation and the productivity factor should be more focused. Nagar and Dixit (2012), combined the UCP and COCOMO and divided four software projects into sub-modules to estimate the KLOC with the help of use cases. It was found that dividing the project into smaller sub-modules gets the estimated effort closer to the actual effort relative to the entire project.

Silhavy *et al.* (2018), evaluated Gaussian Mixture Model Clustering, Moving Window, K-means clustering, and Spectral Clustering techniques as a subset selection method for UCP estimation. The prediction error of linear regression methods was shown to be significantly decreased when clustering approaches were used. When compared to UCP, the SC reduced prediction error by up to 98 %. The moving window produced inconsistent results due to its sensitivity of data. Toka and Turetken (2013), presented an empirical assessment on parametric software estimation models (SLIM, COCOMO II, True Planning, and SEER-SEM) based on their prediction accuracy. The results suggested that COCOMO-II model showed significant results than the other three models on MMRE metrics. Patil *et al.* (2014), showed the improved accuracy of component-based software development (CBSD) effort estimation using fuzzy logic technique and found component point to be the best method accurately estimate the size. Wu *et al.* (2018), introduced a combined method integrating CBR and PSO for software effort estimation. The optimised weight derived from the PSO approach is proposed for three extensively used CBR methods in SEE (Euclidean distance, Manhattan distance, and grey relational grade). The suggested models are evaluated using two well used datasets (the Desharnais and Maxwell datasets), and the results are compared to other widely used methods, such as MICBR and GABE, using the MMRE, PRED (25), and MdMRE criteria. The experimental results showed that the combination technique incorporating PSO and CBR increased estimation performance for the three performance metrics at both the training and test stages. Ardiansyah *et al.* (2018), proposed an analogy effort estimation model by adjusting three distance measurements, namely Euclidean, Manhattan and Minkowski distance. Manhattan distance yields the best results, with a % MMRE, a 28 % MdMRE, and a 48% PRED (25). The analogy method has a mean accuracy of 49.9%, MdMRE 29.37 %, and PRED (25) 51.23 %. An empirical study is conducted using five popular datasets and the 30% hold-out validation approach to evaluate and compare the performance of optimal tree ensemble. In terms of MMRE, MdMRE, and PRED (25), the proposed ensemble outperforms regression trees and random forest models (Abdelali *et al.*, 2019). Alhazmi (2020) conducted a comparison study of twelve ensemble approaches for estimating effort. With an MMRE value of 10% and PRED (25) of 97%, the Genetic Algorithm feature selection for the bagging M5 rule was shown to be the best method for forecasting efforts. In expert estimation, the experts use their prior experiences and knowledge to analyze numerous factors for effort estimation. To empower the organizations to get profited by expert judgment, they should distinguish the human elements influencing the expert judgment (Magazinius *et al.*, 2012) and apply useful guidelines for delivering better estimates. According to Basri *et al.*

(2016), due to its simplicity and versatility, software development teams prefer to use expert judgement rather than other estimation models.

Based on the evidence mentioned, the fewer endeavours have been managed to accurately estimate the effort of the software systems. Hence, this research aims to propose an ensemble model to improve the estimation accuracy of software development effort. The proposed model has incorporated Use Case Points (UCP), expert judgement and Case-Based Reasoning (CBR) techniques to optimize the effort accuracy.

1.3 Problem Statement

In recent years, due to significant adoption of new technologies and development processes in the field of software engineering, developers and researchers have been working on improving the accuracy of software effort estimation. Development practitioners and researchers have long been frustrated by the inaccuracy of software effort estimates. Despite significant efforts to improve this key activity, estimation accuracy remains low (Usman *et al.*, 2018). Earlier researchers (Azzeh and Nassif, 2016; Wu *et al.*, 2018) highlighted that the software industry lacks in accurately estimating the effort of software applications. According to a review of surveys on effort estimation conducted by Keshta (2017), schedule and budget overruns occurred in 60%–80% of the projects examined. Inaccurate effort estimates can result in unrealistic schedules and budgets, which can be a significant business risk. According to a report by The Standish Group International ("Standish Group International," CHAOS Report 2015), over 25,000 projects highlight the consequences of not using the estimation method to enable accurate software estimation. These consequences range from a lack of competitiveness and underestimation to project failures and, ultimately, corporate loss. In the last few decades, several models and practices of estimating the software effort have evolved, but it is still an essentially unresolved problem (Sehra *et al.*, 2017). The software industry is striving to optimize accuracy as there is no proven software estimation model that can be used continuously in various situations to estimate the software effort (Gautam and Singh, 2018). Consequently, underestimation and overestimation may result in serious consequences for developers and customers towards disappointment, inaccurate estimation and hence, contribute to either low-quality project, team frustration or cost overrun. This research aims to propose an ensemble model to improve the accuracy of software development effort. The proposed model has incorporated Use Case Points (UCP), expert judgement and Case-Based Reasoning (CBR) techniques to optimize the effort accuracy. It is evaluated using projects from software development organisations and student projects as case studies, with a quantitative approach involving experiments, industry experts, archival data on estimations, and evaluation criteria.

1.4 Research Questions

The main research question of this research is:

"How to improve the estimation accuracy of software development effort?"

Four research questions are stated to provide an effective solution to the main research question:

i) **RQ1:** What does the existing studies investigate about effort estimation models and accuracy improvement in ensemble and solo techniques?
ii) **RQ2:** How to develop an ensemble effort estimation model using Use Case Points (UCP), expert judgement and Case-Based Reasoning (CBR) techniques?
iii) **RQ3:** Does the ensemble model improve estimation accuracy of software development effort compared to the existing solo techniques?
iv) **RQ4:** How can the ensemble effort estimation model be accurate and applicable in software development?

1.5 Research Objectives

This research aims to propose an ensemble model to improve the estimation accuracy of software development effort. The ensemble model is incorporated with Use Case Point (algorithmic), expert judgement (non-algorithmic) and Case-Based Reasoning (machine learning) techniques to make an ensemble. Hence, the objectives are as follows to achieve this aim:

i) **RO1:** To investigate the effort estimation models and the accuracy improvement of ensemble and solo techniques.
ii) **RO2:** To develop an ensemble effort estimation model using Use Case Points (UCP), expert judgement and Case-Based Reasoning (CBR) techniques.
iii) **RO3:** To evaluate the improvement of estimation accuracy of an ensemble model by comparing existing solo models.
iv) **RO4:** To validate the applicability of the proposed model with diverse evaluation metrics in software development.

1.6 Scope of Research

The main purpose in defining a research scope is to concentrate on how far the research area has been explored in terms of research limits and constraints. The limitations of the research scope are as following:

1.6.1 Research Context

The objective of this research is to develop an ensemble estimation model to optimize the accuracy of software development effort. While most of the ensemble models are developed using machine learning techniques only. However, this research focusses on combining Use Case Point (algorithmic), expert judgement (non-algorithmic) and Case-Based Reasoning (machine learning) techniques to make an ensemble for improving effort estimation accuracy.

1.6.2 Research Challenges

Since this study focused on the software development phase, the challenges of gathering actual industry data in real software projects due to the COVID-19 pandemic were faced. The software industry therefore has limitations with real software development; for example, confidentiality, approaching technical experts, complex organization structures and politics. These factors affected the milestones of the research and collection of data. This study, therefore, summaries the following challenges:

i) *Find estimation experts:* This research needs contribution from industrial software experts to estimate software effort and gather related data. To find estimation experts based on experiences and competency in solving the task with highly related domain background and good estimation records is a challenge.

ii) *Selection of real software projects:* This research aims to investigate effort estimation (phenomenon) in the real context. This study, therefore, selected a case study methodology to perform this investigation in its natural context. The selection of real software projects requires participation from the industry. This research might not be able to benefit from them directly due to commercial obligations, confidentiality, privacy policies and complex organizational structure. Such challenges obstructed to capture and collection of meaningful data for this research.

iii) *Sufficient documentation:* The documentation or information required to estimate software effort includes Software Requirement Specification Document (SRS), progress report (used for the actual amount of effort), software design document (if available), case selection and opinion from experts using UCP size and checklist. In Agile methodology, the documentation includes the product backlogs and sprint backlogs. The availability and integrity of the aforementioned documents are found missed in few software industries due to less focusing on the documentation. Moreover, in the UCP method, the UML were not well-written and structured for accurate estimation of software effort that caused difficulties.

1.7 Significance of Research

The main effect of this research is significant in opinions shown in Figure 1.1. In the first opinion, the proposed ensemble effort estimation model will provide significant information while estimating the amount of work and hours needed to develop software. Previous studies conclude that the key factor in project failure is inaccurate effort estimation, hence, this research will help practitioners and developers in making good management decisions, project planning and controlling activities. The proposed model developed at the end of this research will be useful to software development companies and practitioners as a tool for accurately estimating the time and effort required to develop new software systems.

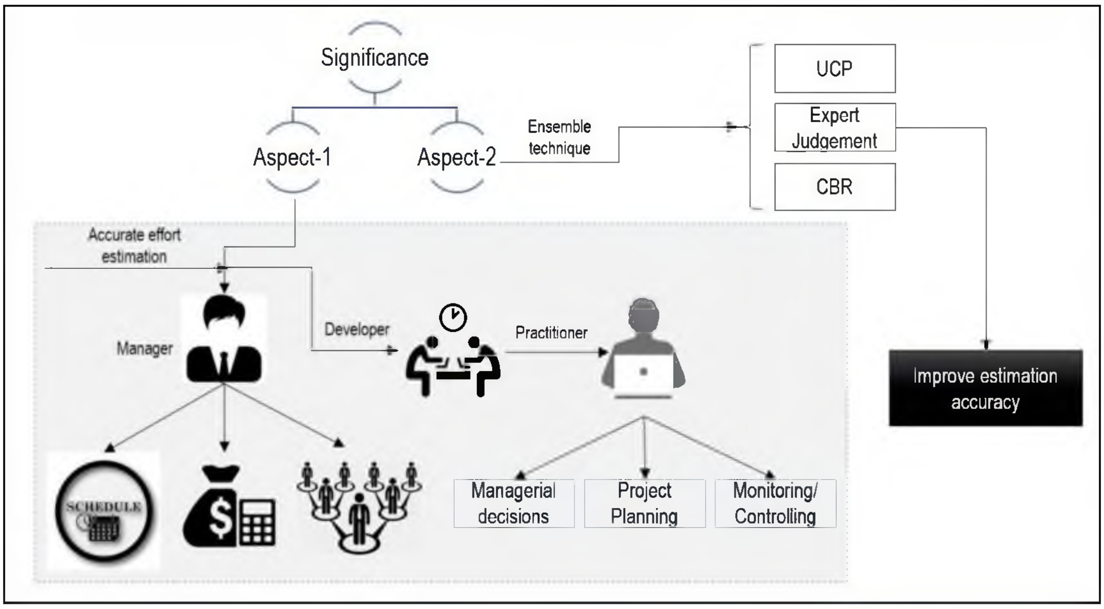

Figure 1.1 Significance of research

In other opinion, by realizing the significance of an ensemble technique, incorporating algorithmic, expert estimation and machine learning approaches using combination rules will need to improve the accuracy of the development effort.

1.8 Research Gaps

The research gaps investigated in this thesis are described below:

Gap 1: The gap was identified during the initial literature review of thesis topic. The different types of reviews of studies have been conducted on effort estimation in software development. However, to the best of the knowledge regardless of the number of review of studies in this perspective, algorithmic, non-algorithmic and machine learning based effort have not been accumulated so far in their studies in a solitary review. It is also found that since 2016 none of the systematic literature reviews has been studied.

Gap 2: This gap was identified during execution and analysis phases of the SLR. To the best of knowledge, comparative study of ensemble and solo machine learning techniques on effort estimation accuracy have not been addressed in the literature.

Gap 3: The different types of studies have been conducted on effort estimation categories. However, algorithmic, expert estimation and machine learning techniques have not been ensembled so far in the research literature. It is investigated that the accuracy improvement using ensemble technique gaining researchers' attention towards further exploring this technique for achieving accurate effort estimation results. It is also found that an ensemble technique produced better estimation results than a solo technique. This is because each solo estimation technique has merits and demerits which leads to somehow inaccurate estimation results.

1.9 Operational Definitions

The operational terminologies used in this research are briefly stated below:

Software development	It is the process of gathering requirements, designing, coding, testing and fixing bugs involved in creating and maintaining software applications.
Algorithmic models	It uses statistical and mathematical formulation that take set of inputs, manipulate and produce output to derive estimation results.
Expert estimation	The experts involved in this technique analysed a variety of factors using their knowledge and previous experience with similar projects.
Software Effort estimation	It is the process of estimating how much time and effort will be required to develop a software system; it is typically measured in man-days, man-months, and man-hours.
Magnitude of Relative Error (MRE)	It is defined as the ratio of actual effort to estimated effort.
Mean Magnitude of Relative Error (MMRE)	The amount of estimated effort to know the under-estimation or over-estimation attributes in comparison to the actual estimation.
Magnitude of Error Relative (MER)	It is defined as the ratio of estimated effort to actual effort.
Mean Magnitude of Error Relative (MMER)	The amount of actual effort to know the under-estimation or over-estimation attributes in comparison to the estimated estimation.
PRED (25)	It is the percentage of estimation within 25% of the actual efforts.
Applicability	The degree of significance of the proposed model in software development.
Accuracy	The degree of precision of the estimated effort compared to the actual effort

Ensemble Effort Estimation	It consists of combining more than one technique by means of a combination rule.
Case-Based Reasoning (CBR)	It identifies one or more past projects that are similar to the target project and extracts the effort estimation from those projects.
Use Case Point (UCP)	In object-oriented software environment, the use case diagrams are converted into size metrics.
MSE	The average squared difference between the actual and estimated values is measured by the Mean Squared Error (MSE).

1.10 Organization of the Thesis

The mapping of the research questions with chapters and research objectives with the research questions addressed in remainder chapters are shown in Table 1.2.

Table 1.2 Mapping of research questions with chapters

<table>
<tr><th colspan="8">Chapters</th></tr>
<tr><th rowspan="2">RQs</th><th colspan="2">2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th></tr>
<tr><th>Study[1]</th><th>Study[2]</th><th></th><th></th><th></th><th></th><th></th></tr>
<tr><td rowspan="2">1</td><td>RQ1/RO1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>RQ1/RO1</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>2</td><td></td><td></td><td></td><td>RQ2/RO2</td><td></td><td></td><td></td></tr>
<tr><td>3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>4</td><td></td><td></td><td></td><td></td><td></td><td>RQ3/RO3; RQ4/RO4</td><td></td></tr>
</table>

There are seven chapters in this thesis. Introduction to research, research background, problem statement, research questions (RQ), research objectives (RO), scope, significance of research, and operational definitions are all covered in the first chapter. The overview of the remainder chapter of thesis is shown in Figure. 1.2.

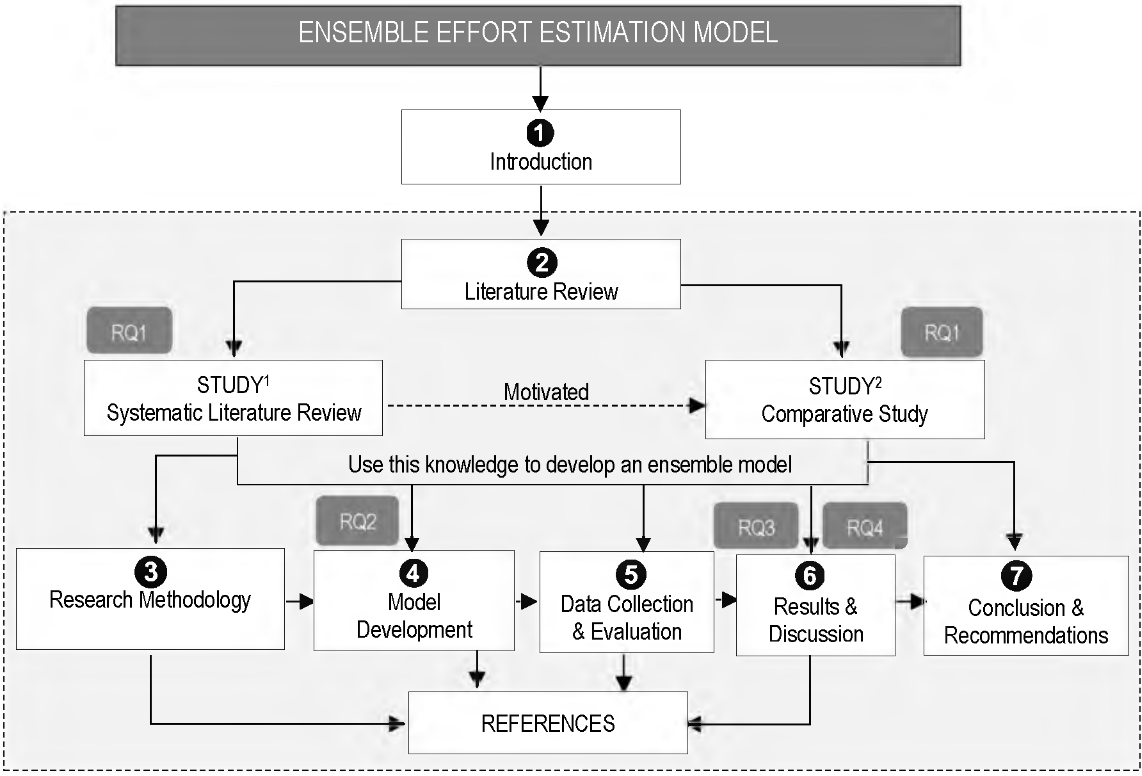

Figure 1.2 Overview of thesis

REFERENCES

A. Abran, J. M. D., S. Oligny, D. St. Pierre, C. Symons. (2001). COSMIC-FFP measurement manual, the COSMIC implementation guide for ISO/IEC 19761, Version 2.1. . *The Common Software Measurement International Consortium*.

Abdelali, Z., Hicham, M., and Abdelwahed, N. (2019). *An ensemble of optimal trees for software development effort estimation*. Paper presented at the International Conference on Advanced Information Technology, Services and Systems.

Albrecht, A. J. (1979). Measuring Application Development Productivity. *Proceedings of the Joint SHARE/GUIDE/IBM Application Development Symposium*, 83-92.

Alhazmi, O. a. K., M. (2020). Software Effort Prediction Using Ensemble Learning Methods. *Journal of Software Engineering and Applications, 13*, 143-160. doi:10.4236/jsea.2020.137010.

alves. (2013). An Empirical Study on the estimation of sw dev effort with UCP.

Alves, L. M., Sousa, A., Ribeiro, P., and Machado, R. J. (2013, 23-26 Oct. 2013). *An empirical study on the estimation of software development effort with use case points*. Paper presented at the 2013 IEEE Frontiers in Education Conference (FIE).

Anda, B., Angelvik, E., and Ribu, K. (2002). *Improving estimation practices by applying use case models*. Paper presented at the International Conference on Product Focused Software Process Improvement.

Anda, B., Benestad, H. C., and Hove, S. E. (2005). *A multiple-case study of software effort estimation based on use case points*. Paper presented at the 2005 International Symposium on Empirical Software Engineering, 2005.

Angelis, L., and Stamelos, I. (2000). A Simulation Tool for Efficient Analogy Based Cost Estimation. *Empirical Software Engineering, 5*(1), 35-68. doi:10.1023/a:1009897800559

Anupama, K., Devendra Kumar, T., and Kalpana, Y. (2020). The Role of Neural Networks and Metaheuristics in Agile Software Development Effort Estimation. *International Journal of Information Technology Project Management (IJITPM), 11*(2), 50-71.

Ardiansyah, Mardhia, M. M., and Handayaningsih, S. (2018). Analogy-based model for software project effort estimation. *International Journal of Advances in Intelligent Informatics, 4*(3), 251-260. doi:10.26555/ijain.v4i3.266

Arifin, H. H., Daengdej, J., Khanh, N. T., and Ieee. (2017). An Empirical Study of Effort-Size and Effort-Time in Expert-Based Estimations. In *2017 8th Ieee International Workshop on Empirical Software Engineering in Practice* (pp. 35-40). New York: Ieee.

Arnuphaptrairong, T. (2018). Early Stage Software Effort Estimation Using Function Point Analysis: Empirical Evidence. *Lecture Notes in Engineering and Computer Science, 2203*, 730-735.

Attarzadeh, I., Mehranzadeh, A., and Barati, A. (2012). *Proposing an Enhanced Artificial Neural Network Prediction Model to Improve the Accuracy in Software Effort Estimation*. Paper presented at the Proceedings of the 2012 Fourth International Conference on Computational Intelligence, Communication Systems and Networks.

Ayyildiz, T. E., & Koçyigit, A. (2014). . (2014). An Early Software Effort Estimation Method Based on Use Cases and Conceptual Classes. *Journal of Software, JSW, 9*(8).

Azzeh, M., and Nassif, A. B. (2016). A hybrid model for estimating software project effort from Use Case Points. *Applied Soft Computing, 49*, 981-989. doi:10.1016/j.asoc.2016.05.008

Azzeh, M., Nassif, A. B., and Minku, L. L. (2015). S13- An empirical evaluation of ensemble adjustment methods for analogy-based effort estimation. *Journal of Systems and Software, 103*, 36-52.

Azzeh M., N. D., Cowling P. . (2008). Software Project Similarity Measurement Based on Fuzzy C-Means. *In: Wang Q., Pfahl D., Raffo D.M. (eds) Making Globally Distributed Software Development a Success Story. ICSP 2008. Lecture Notes in Computer Science, vol 5007. Springer, Berlin, Heidelberg.* . doi:https://doi.org/10.1007/978-3-540-79588-9_12

BaniMustafa, A. (2018, 11-12 July 2018). *Predicting Software Effort Estimation Using Machine Learning Techniques.* Paper presented at the 2018 8th International Conference on Computer Science and Information Technology (CSIT).

Bardsiri, V. K., Jawawi, D. N. A., Bardsiri, A. K., and Khatibi, E. (2013). LMES: A localized multi-estimator model to estimate software development effort. *Engineering Applications of Artificial Intelligence, 26*(10), 2624-2640. doi:10.1016/j.engappai.2013.08.005

Baskeles, B., Turhan, B., and Bener, A. (2007). *Software effort estimation using machine learning methods.* Paper presented at the 2007 22nd international symposium on computer and information sciences.

Basri, S., Kama, N., Haneem, F., and Adli, S. (2016). Using static and dynamic impact analysis for effort estimation. *IET Software, 10*(4), 89-95. doi:10.1049/iet-sen.2015.0043

Benala, T. R., and Mall, R. (2018). DABE: Differential evolution in analogy-based software development effort estimation. *Swarm and Evolutionary Computation, 38*, 158-172. doi:https://doi.org/10.1016/j.swevo.2017.07.009

Borandag, E., Yucalar, F., and Erdogan, S. Z. (2016). A case study for the software size estimation through MK II FPA and FP methods. *International Journal of Computer Applications in Technology, 53*(4), 309-314. doi:10.1504/IJCAT.2016.076777

Braga, P. L., Oliveira, A. L. I., Ribeiro, G. H. T., and Meira, S. R. L. (2007, 12-17 Aug. 2007). *acceptable range for PRED and MMRE Bagging Predictors for Estimation of Software Project Effort.* Paper presented at the 2007 International Joint Conference on Neural Networks.

Braz, M. R., and Vergilio, S. R. (2006). *Software Effort Estimation Based on Use Cases.* Paper presented at the Proceedings of the 30th Annual International Computer Software and Applications Conference - Volume 01.

Britto, R., Freitas, V., Mendes, E., and Usman, M. (2014, 18-21 Aug. 2014). *Effort Estimation in Global Software Development: A Systematic Literature Review.* Paper presented at the 2014 IEEE 9th International Conference on Global Software Engineering.

Britto, R., Mendes, E., and Börstler, J. (2015). An Empirical Investigation on Effort Estimation in Agile Global Software Development. *2015 IEEE 10th International Conference on Global Software Engineering*, 38-45.

Britto, R., Usman, M., and Mendes, E. (2014). *Effort Estimation in Agile Global Software Development Context.* Paper presented at the XP Workshops.

Burgess, C. J., and Lefley, M. (2001). Can genetic programming improve software effort estimation? A comparative evaluation. *Information and Software Technology, 43 (14)*, 863-873.

Carroll, E. R. (2015). *Estimating software based on use case points*. Paper presented at the Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, San Diego, CA, USA.

Chandrasekaran, R., and Kumar, R. V. (2016). *On the Estimationof the Software Effort and Schedule using Constructive Cost Model - II and Functional Point Analysis*.

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

Dan, Z. (2013, 28-30 July 2013). *Improving the accuracy in software effort estimation: Using artificial neural network model based on particle swarm optimization.* Paper presented at the Proceedings of 2013 IEEE International Conference on Service Operations and Logistics, and Informatics.

Demirors, O., and Gencel, C. (2009). Conceptual Association of Functional Size Measurement Methods. *IEEE Software, 26*(3), 71-78. doi:10.1109/MS.2009.60

Doloi, H. (2013). Cost Overruns and Failure in Project Management: Understanding the Roles of Key Stakeholders in Construction Projects. *Journal of Construction Engineering and Management, 139*(3), 267-279. doi:10.1061/(ASCE)CO.1943-7862.0000621

Eck, B. Brundick, T. Fettig, and Ugljesa, J. D. a. J. (2009). *Parametric estimating handbook. The International Society of Parametric Analysis, Fourth Edition.* In.

Elish, M. O. (2013, 16-19 April 2013). *Assessment of voting ensemble for estimating software development effort.* Paper presented at the 2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM).

Elish, M. O., Helmy, T., and Hussain, M. I. (2013). S7- Empirical study of homogeneous and heterogeneous ensemble models for software development effort estimation. *Mathematical Problems in Engineering, 2013*.

Finnie, G. R., Wittig, G. E., and Desharnais, J. M. (1997). A comparison of software effort estimation techniques: Using function points with neural networks, case-based reasoning and regression models. *Journal of Systems and Software, 39*(3), 281-289.

Friedman, M. (1940). A Comparison of Alternative Tests of Significance for the Problem of m Rankings. *Ann. Math. Statist., 11*(1), 86-92. doi:10.1214/aoms/1177731944

Frohnhoff, S. (2014). *Revised Use Case Point Method-Effort Estimation in Development Projects for Business Applications*.

Furulund, K. M., and Molkken-stvold, K. (2007, 11-12 Oct. 2007). *Increasing Software Effort Estimation Accuracy Using Experience Data, Estimation Models and Checklists.* Paper presented at the Seventh International Conference on Quality Software (QSIC 2007).

G. Schneider, and Winters, J. P. (2001). *Applying Use Cases: A Practical Guide*. In.

Gabrani, G., & Saini, N. (2016). Effort estimation models using evolutionary learning algorithms for software development. Paper presented at the 2016 Symposium on Colossal Data Analysis and Networking, CDAN 2016. doi:doi:10.1109/CDAN.2016.7570916

Gautam, S. S., and Singh, V. (2018). The state-of-the-art in software development effort estimation. *Journal of Software: Evolution and Process, 30*(12). doi:10.1002/smr.1983

Grimstad, S., and Jorgensen, M. (2008). *A preliminary study of sequence effects in judgment-based software development work-effort estimation*. Paper presented at the Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering, Italy.

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

Gupta, M., and Kalia, A. (2017). Empirical Study of Software Metrics. *Research Journal of Science and Technology, 9*(1), 17-24.

Hasselbring, W., and Giesecke, S. (2006). Research Methods in Software Engineering. GITO-Verl.

Haugen, N. C. (2006). *An Empirical Study of Using Planning Poker for User Story Estimation*. Paper presented at the Proceedings of the conference on AGILE 2006.

Hidmi, O., and Sakar, B. E. (2017). Software development effort estimation using ensemble machine learning. *Int J Comput Commun Instrum Eng, 4*(1), 143-147.

Hira, A., and Boehm, B. (2016). *Function Point Analysis for Software Maintenance*. Paper presented at the Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, Ciudad Real, Spain.

Hosni, M., Idri, A., Nassif, A. B., and Abran, A. (2016). *Heterogeneous ensembles for software development effort estimation*. Paper presented at the Soft Computing & Machine Intelligence (ISCMI), 2016 3rd International Conference on.

Hsu, C., Rodas, N. U., Huang, C., and Peng, K. (2010, 19-23 July 2010). *A Study of Improving the Accuracy of Software Effort Estimation Using Linearly Weighted Combinations*. Paper presented at the 2010 IEEE 34th Annual Computer Software and Applications Conference Workshops.

Huang, S.-J., and Chiu, N.-H. (2006). Optimization of analogy weights by genetic algorithm for software effort estimation. *Information and Software Technology, 48*(11), 1034-1045. doi:https://doi.org/10.1016/j.infsof.2005.12.020

Huang, S.-J., Chiu, N.-H., and Chen, L.-W. (2008). Integration of the grey relational analysis with genetic algorithm for software effort estimation. *European Journal of Operational Research, 188*(3), 898-909. doi:https://doi.org/10.1016/j.ejor.2007.07.002

IBM. (2019). Retrieved 4 December 2021 <https://www.ibm.com/services>

Idri, Hosni, and Abran. (2016). Systematic literature review of ensemble effort estimation. *Journal of Systems and Software, 118*, 151-175. doi:10.1016/j.jss.2016.05.016

Idri, A., Amazal, F. a., and Abran, A. (2015). Analogy-based software development effort estimation: A systematic mapping and review. *Information and Software Technology, 58*, 206-230. doi:https://doi.org/10.1016/j.infsof.2014.07.013

Idri, A., Amazal, F. A., and Abran, A. (2016). Accuracy comparison of analogy-based software development effort estimation techniques. *International Journal of Intelligent Systems, 31*(2), 128-152.

Idri, A., Hosni, M., and Abran, A. (2016). Improved estimation of software development effort using Classical and Fuzzy Analogy ensembles. *Applied Soft Computing, 49*, 990-1019. doi:https://doi.org/10.1016/j.asoc.2016.08.012

International Project Management Association. (2019). Retrieved 4 December 2021 <https://www.ipma.world/assets/PM-Survey-FullReport-2019-FINAL.pdf>

Jacobson, I. (1992). *Object-Oriented Software Engineering - a Use Case Driven Approach*. In.

Jain, S., Yadav, V., and Singh, R. (2014, 5-7 March 2014). *An approach for OO software size estimation using Predictive Object Point Metrics.* Paper presented at the 2014 International Conference on Computing for Sustainable Global Development (INDIACom).

Jodpimai, P., Sophatsathit, P., and Lursinsap, C. (2018). Ensemble effort estimation using selection and genetic algorithms. *International Journal of Computer Applications in Technology, 58*(1), 17-28.

Jørgensen, M. (2004). A review of studies on expert estimation of software development effort. *Journal of Systems and Software, 70*(1-2), 37-60. doi:10.1016/s0164-1212(02)00156-5

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

Jørgensen, M. (2016). Unit effects in software project effort estimation: Work-hours gives lower effort estimates than workdays. *Journal of Systems and Software, 117*, 274-281. doi:10.1016/j.jss.2016.03.048

Jørgensen, M., and Gruschke, T. M. (2009). The Impact of Lessons-Learned Sessions on Effort Estimation and Uncertainty Assessments. *IEEE Transactions on Software Engineering, 35*(3), 368-383. doi:10.1109/TSE.2009.2

Jørgensen, M., and Halkjelsvik, T. (2010). The effects of request formats on judgment-based effort estimation. *Journal of Systems and Software, 83*(1), 29-36. doi:https://doi.org/10.1016/j.jss.2009.03.076

Jorgensen, M., and Moløkken-Østvold, K. (2003). *A preliminary checklist for software cost management*.

Jorgensen, M., and Shepperd, M. (2007). A Systematic Review of Software Development Cost Estimation Studies. *IEEE Transactions on Software Engineering, 33*(1), 33-53. doi:10.1109/TSE.2007.256943

Kamal, M. W., and Ahmed, M. A. (2011). A proposed framework for use case based effort estimation using fuzzy logic: building upon the outcomes of a systematic literature review. *International Journal of New Computer Architectures and their Applications (IJNCAA), 1*(4), 953-976.

Karner, G. (1993). *Resource Estimation for Objectory Projects*.

Kaur, M., and Sehra, S. K. (2014, 7-8 Feb. 2014). *Particle swarm optimization based effort estimation using Function Point analysis.* Paper presented at the Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on.

Kaur, R., and Sengupta, J. (2013). Software Process Models and Analysis on Failure of Software Development Projects. *CoRR, abs/1306.1068*.

Kchaou, D., Bouassida, N., and Ben-Abdallah, H. (2015). *Change effort estimation based on UML diagrams application in UCP and COCOMOII.* Paper

presented at the 2015 10th International Joint Conference on Software Technologies (ICSOFT).
Kemerer, C. F. (1987). An empirical validation of software cost estimation models. *Commun. ACM, 30*(5), 416-429. doi:10.1145/22899.22906
Kerzner, H. (2018). *Project management best practices: Achieving global excellence*: John Wiley & Sons.
Keshta, I. M. (2017). S-5 Software Cost Estimation Approaches: A Survey. *Journal of Software Engineering and Applications, 10*(10), 824-842. doi:10.4236/jsea.2017.1010046
Khan, M. Z. (2020). Particle Swarm Optimisation Based Feature Selection for Software Effort Prediction Using Supervised Machine Learning and Ensemble Methods: A Comparative Study. *Invertis Journal of Science & Technology, 13*, 33.
Kirmani, M. M., and Wahid, A. (2015). Impact of Modification Made in Re-UCP on Software Effort Estimation. *Journal of Software Engineering and Applications, 08*(06), 276-289. doi:10.4236/jsea.2015.86028
Kitchenham, and Charters. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. *2*.
Kitchenham, B., and Mendes, E. (2004). Software productivity measurement using multiple size measures. *IEEE Transactions on Software Engineering, 30*(12), 1023-1035. doi:10.1109/TSE.2004.104
Kocaguneli, E., Menzies, T., Bener, A., and Keung, J. W. (2012). Exploiting the Essential Assumptions of Analogy-Based Effort Estimation. *IEEE Transactions on Software Engineering, 38*(2), 425-438. doi:10.1109/TSE.2011.27
Kocaguneli, E., Menzies, T., and Keung, J. W. (2012). S8- On the value of ensemble effort estimation. *IEEE Transactions on Software Engineering, 38*(6), 1403-1416.
Kocaguneli, E., Menzies, T., and Keung, J. W. (2013). Kernel methods for software effort estimation. *Empirical Software Engineering, 18*(1), 1-24. doi:10.1007/s10664-011-9189-1
Kultur, Y., Turhan, B., and Bener, A. (2009). Ensemble of neural networks with associative memory (ENNA) for estimating software development costs. *Knowledge-Based Systems, 22*(6), 395-402. doi:https://doi.org/10.1016/j.knosys.2009.05.001
Kumar, A., and Datta, U. (2015). An Effort Estimation Model for Software Development using Ensemble Learning. *International Journal of Computer Applications, 115*(21).
Kusumoto, S., Matukawa, F., Inoue, K., Hanabusa, S., and Maegawa, Y. (2004). *Estimating effort by use case points: method, tool and case study*. Paper presented at the 10th International Symposium on Software Metrics, 2004. Proceedings.
Lamba, V. (2020). A Survey: To Explore the Factors Behind Inaccuracy in Cost Effort Estimation and Providing Improvements and Suggestions.
Lehtinen, T. O. A., M, M. V., Vanhanen, J., Itkonen, J., and Lassenius, C. (2014). Perceived causes of software project failures - An analysis of their relationships. *Inf Softw. Technol., 56*(6), 623-643. doi:10.1016/j.infsof.2014.01.015
Li, Q., Wang, Q., Yang, Y., and Li, M. (2008). *Reducing biases in individual software effort estimations: a combining approach*. Paper presented at the Proceedings

of the Second ACM-IEEE international symposium on Empirical software engineering and measurement, Kaiserslautern, Germany.

López-Martín, C., and Abran, A. (2012). Applying expert judgment to improve an individual's ability to predict software development effort. *International Journal of Software Engineering and Knowledge Engineering, 22*(4), 467-483. doi:10.1142/S0218194012500118

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

Mahmood, Y., Kama, N., and Azmi, A. (2020). A systematic review of studies on use case points and expert-based estimation of software development effort. *Journal of Software: Evolution and Process, 32*(7), e2245. doi:10.1002/smr.2245

Mahnič, V., and Hovelja, T. (2012). On using planning poker for estimating user stories. *Journal of Systems and Software, 85*(9), 2086-2095.

Malgonde, O., and Chari, K. (2018). An ensemble-based model for predicting agile software development effort. *Empirical Software Engineering*. doi:10.1007/s10664-018-9647-0

Malhotra, R., and Jain, A. (2011). Software effort prediction using statistical and machine learning methods. *International Journal of Advanced Computer Science and Applications, 2*(1), 145-152.

Medina, A., and Francis, A. J. (2015). What are the characteristics that software development project team members associate with a good project manager? *Project Management Journal, 46*(5), 81-93.

Mendes, E., Counsell, S., and Mosley, N. (2003). Web hypermedia cost estimation: further assessment and comparison of cost estimation modelling techniques. *New Rev. Hypermedia Multimedia, 8*(1), 199-229. doi:10.1080/13614560208914741

Mendes, E., Mosley, N., and Counsell, S. (2006). Web Effort Estimation. In E. Mendes & N. Mosley (Eds.), *Web Engineering* (pp. 29-73). Berlin, Heidelberg: Springer Berlin Heidelberg.

Mendes, E., Watson, I., Triggs, C., Mosley, N., and Counsell, S. (2003). A Comparative Study of Cost Estimation Models for Web Hypermedia Applications. *Empirical Software Engineering, 8*(2), 163-196. doi:10.1023/a:1023062629183

Minku, L. L., and Yao, X. (2011). *A principled evaluation of ensembles of learning machines for software effort estimation.* Paper presented at the Proceedings of the 7th International Conference on Predictive Models in Software Engineering.

Minku, L. L., and Yao, X. (2013). S10- Ensembles and locality: Insight on improving software effort estimation. *Information and Software Technology, 55*(8), 1512-1528. doi:https://doi.org/10.1016/j.infsof.2012.09.012

Mohagheghi, P., Anda, B., and Conradi, R. (2005). *Effort estimation of use cases for incremental large-scale software development.* Paper presented at the Proceedings of the 27th international conference on Software engineering.

Molokken-Ostvold, K., and Haugen, N. C. (2007). *Combining Estimates with Planning Poker--An Empirical Study*. Paper presented at the Proceedings of the 2007 Australian Software Engineering Conference.

Moløkken-Østvold, K., and Jørgensen, M. (2004). Group Processes in Software Effort Estimation. *Empirical Software Engineering, 9*(4), 315-334. doi:10.1023/B:EMSE.0000039882.39206.5a

Nagar, C., and Dixit, A. (2012). Efforts estimation by combining the use case point and COCOMO. *International Journal of Computer Applications, 52*(7).

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

Nassif, A. B., Capretz, L. F., and Ho, D. (2011). *Estimating Software Effort Based on Use Case Point Model Using Sugeno Fuzzy Inference System*. Paper presented at the 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence.

Nassif, A. B., Capretz, L. F., and Ho, D. (2012). *Estimating software effort using an ANN model based on use case points*. Paper presented at the 2012 11th International Conference on Machine Learning and Applications.

Nassif, A. B., Capretz, L. F., and Ho, D. (2016). Enhancing use case points estimation method using soft computing techniques. *arXiv preprint arXiv:1612.01078*.

Nassif, A. B., Capretz, L. F., Ho, D., and Azzeh, M. (2012). *S16- A Treeboost Model for Software Effort Estimation Based on Use Case Points*. Paper presented at the 2012 11th International Conference on Machine Learning and Applications.

Ochodek, M. (2016). Functional size approximation based on use-case names. *Information and Software Technology, 80*, 73-88. doi:https://doi.org/10.1016/j.infsof.2016.08.007

Ochodek, M., Nawrocki, J., and Kwarciak, K. (2011). Simplifying effort estimation based on Use Case Points. *Information and Software Technology, 53*(3), 200-213. doi:10.1016/j.infsof.2010.10.005

P. Runeson, M. H¨ost, and A. Rainer, a. B. R. (2012). *Case Study Research in Software Engineering: Guidelines and Examples*. In.

Pai, D. R., McFall, K. S., and Subramanian, G. H. (2013). Software Effort Estimation Using a Neural Network Ensemble. *Journal of Computer Information Systems, 53*(4), 49-58. doi:10.1080/08874417.2013.11645650

Pandey, M., Litoriya, R., and Pandey, P. (2020). Validation of existing software effort estimation techniques in context with mobile software applications. *Wireless Personal Communications, 110*(4), 1659-1677.

Parvez, A. W. M. M. (2013). Efficiency factor and risk factor based user case point test effort estimation model compatible with agile software development. *2013 International Conference on Information Technology and Electrical Engineering (ICITEE)*, 113-118.

Passing, U., and Shepperd, M. (2003, 30 Sept.-1 Oct. 2003). *An experiment on software project size and effort estimation.* Paper presented at the 2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003. Proceedings.

Patil, L. V., Shivale, N. M., Joshi, S. D., and Khanna, V. (2014, 21-22 Feb. 2014). *Improving the accuracy of CBSD effort estimation using fuzzy logic.* Paper presented at the 2014 IEEE International Advance Computing Conference (IACC).

PMI's. (2017). *P. M Institute. Success rates rise: Transforming the high cost of low performance. PMI's PULSE of the PROFESSION*. Retrieved from

Popli, R., and Chauhan, N. S. (2014). Cost and effort estimation in agile software development. *2014 International Conference on Reliability Optimization and Information Technology (ICROIT)*, 57-61.

Popovic, J., Bojic, D., and Korolija, N. (2015). Analysis of task effort estimation accuracy based on use case point size. *IET Software, 9*(6), 166-173. doi:10.1049/iet-sen.2014.0254

Port, D., and Korte, M. (2008). *Comparative studies of the model evaluation criterions mmre and pred in software cost estimation research*. Paper presented at the Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement, Kaiserslautern, Germany.

Pospieszny, P., Czarnacka-Chrobot, B., and Kobylinski, A. (2018). An effective approach for software project effort and duration estimation with machine learning algorithms. *Journal of Systems and Software, 137*, 184-196. doi:https://doi.org/10.1016/j.jss.2017.11.066

Rastogi, H., Dhankhar, S., and Kakkar, M. (2014, 25-26 Sept. 2014). *A survey on software effort estimation techniques*. Paper presented at the Confluence The Next Generation Information Technology Summit (Confluence), 2014 5th International Conference -.

Rijwani, P., and Jain, S. (2016). Enhanced software effort estimation using multi layered feed forward artificial neural network technique. *Procedia Computer Science, 89*, 307-312.

Robiolo, G., and Orosco, R. (2008). Employing use cases to early estimate effort with simpler metrics. *Innovations in Systems and Software Engineering, 4*(1), 31-43.

Runeson, P., and H, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Softw. Engg., 14*(2), 131-164. doi:10.1007/s10664-008-9102-8

Sabrjoo, S., Khalili, M., and Nazari, M. (2015, 5-6 Nov. 2015). *Comparison of the accuracy of effort estimation methods*. Paper presented at the 2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI).

Saroha, M., and Sahu, S. (2015). *Software effort estimation using enhanced use case point model*. Paper presented at the International Conference on Computing, Communication & Automation.

Satapathy, S., Acharya, B. P., and Rath, S. K. (2014). Class point approach for software effort estimation using stochastic gradient boosting technique. *ACM SIGSOFT Softw. Eng. Notes, 39*, 1-6.

Satapathy, S. M., Acharya, B. P., and Rath, S. K. (2016). S17- Early stage software effort estimation using random forest technique based on use case points. *IET Software, 10*(1), 10-17. doi:10.1049/iet-sen.2014.0122

Sehra, S. K., Brar, Y. S., Kaur, N., and Sehra, S. S. (2017). Research patterns and trends in software effort estimation. *Information and Software Technology, 91*, 1-21. doi:10.1016/j.infsof.2017.06.002

Seni, G., and Elder, J. F. (2010). Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions. *Synthesis Lectures on Data Mining and Knowledge Discovery, 2*(1), 1-126. doi:10.2200/S00240ED1V01Y200912DMK002

Seo, Y.-S., Yoon, K.-A., and Bae, D.-H. (2008). *An empirical analysis of software effort estimation with outlier elimination*. Paper presented at the Proceedings of the 4th international workshop on Predictor models in software engineering, Leipzig, Germany. https://doi.org/10.1145/1370788.1370796

Sharma, P., and Singh, J. (2017, 11-12 Dec. 2017). *Systematic Literature Review on Software Effort Estimation Using Machine Learning Approaches.* Paper presented at the 2017 International Conference on Next Generation Computing and Information Systems (ICNGCIS).

Sharma, P., and Singh, J. (2018, 28-29 Sept. 2018). *Machine Learning Based Effort Estimation Using Standardization.* Paper presented at the 2018 International Conference on Computing, Power and Communication Technologies (GUCON).

Shepperd, M., Schofield, C., and Kitchenham, B. (1996). *Effort estimation using analogy.* Paper presented at the Proceedings of the 18th international conference on Software engineering, Berlin, Germany.

Shepperd, M. J., and Schofield, C. (1997). Estimating Software Project Effort Using Analogies. *IEEE Trans. Software Eng., 23*, 736-743.

Shull, F., Singer, J., and berg, D. I. K. S. (2007). *Guide to Advanced Empirical Software Engineering*: Springer-Verlag.

Silhavy, Silhavy, and Prokopova. (2019). Outliners Detection Method for Software Effort Estimation Models. *Software Engineering Methods in Intelligent Algorithms*, 444–455.

Silhavy, R. (2017). Use Case Points Benchmark Dataset. doi:10.17632/2rfkjhx3cn.1

Silhavy, R., Silhavy, P., and Prokopova, Z. (2015a). Algorithmic Optimisation Method for Improving Use Case Points Estimation. *PLOS ONE, 10*(11), e0141887. doi:10.1371/journal.pone.0141887

Silhavy, R., Silhavy, P., and Prokopova, Z. (2015b). Applied Least Square Regression in Use Case Estimation Precision Tuning. In *Software Engineering in Intelligent Systems* (pp. 11-17).

Silhavy, R., Silhavy, P., and Prokopova, Z. (2017). Analysis and selection of a regression model for the Use Case Points method using a stepwise approach. *J. Syst. Softw., 125*(C), 1-14. doi:10.1016/j.jss.2016.11.029

Silhavy, R., Silhavy, P., and Prokopova, Z. (2018). Evaluating subset selection methods for use case points estimation. *Information and Software Technology, 97*, 1-9. doi:10.1016/j.infsof.2017.12.009

Singal, P., Kumari, A. C., and Sharma, P. (2020). Estimation of software development effort: a differential evolution approach. *Procedia Computer Science, 167*, 2643-2652.

Standish Group International. (CHAOS Report 2015). Retrieved 5 December 2021 <https://www.infoq.com/articles/standish-chaos-2015/>

Toka, D., and Turetken, O. (2013, 4-6 Sept. 2013). *Accuracy of Contemporary Parametric Software Estimation Models: A Comparative Analysis.* Paper presented at the 2013 39th Euromicro Conference on Software Engineering and Advanced Applications.

Trendowicz, A., and Jeffery, R. (2014a). Software project effort estimation. *Foundations and Best Practice Guidelines for Success, Constructive Cost Model–COCOMO pags, 12*, 277-293.

Trendowicz, A., and Jeffery, R. (2014b). *Software Project Effort Estimation: Foundations and Best Practice Guidelines for Success*: Springer Publishing Company, Incorporated.

Tronto, I. F. d. B., Silva, J. D. S. d., and Sant'Anna, N. (2007, 12-17 Aug. 2007). *Comparison of Artificial Neural Network and Regression Models in Software*

Effort Estimation. Paper presented at the 2007 International Joint Conference on Neural Networks.
Tsunoda, M., Monden, A., Keung, J., and Matsumoto, K. (2012). *Incorporating Expert Judgment into Regression Models of Software Effort Estimation.* Paper presented at the Proceedings of the 2012 19th Asia-Pacific Software Engineering Conference - Volume 01.
Urbanek, T., Prokopova, Z., and Silhavy, R. (2015). On the value of parameters of use case points method. In *Artificial Intelligence Perspectives and Applications* (pp. 309-319): Springer.
Usman, M., Börstler, J., and Petersen, K. (2017). An Effort Estimation Taxonomy for Agile Software Development. *International Journal of Software Engineering and Knowledge Engineering, 27*(04), 641-674. doi:10.1142/s0218194017500243
Usman, M., Mendes, E., and Börstler, J. (2015). *Effort estimation in agile software development.* Paper presented at the Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering - EASE '15.
Usman, M., Petersen, K., Börstler, J., and Santos Neto, P. (2018). Developing and using checklists to improve software effort estimation: A multi-case study. *Journal of Systems and Software, 146*, 286-309. doi:10.1016/j.jss.2018.09.054
Wen, J., Li, S., Lin, Z., Hu, Y., and Huang, C. (2012). Systematic literature review of machine learning based software development effort estimation models. *Information and Software Technology, 54*(1), 41-59. doi:10.1016/j.infsof.2011.09.002
Whigham, P. A., Owen, C. A., and Macdonell, S. G. (2015). A Baseline Model for Software Effort Estimation. *ACM Trans. Softw. Eng. Methodol., 24*(3), 1-11. doi:10.1145/2738037
Wilcoxon, F. (1945). Individual Comparisons by Ranking Methods. *Biometrics Bulletin, 1*(6), 80-83. doi:10.2307/3001968
Wohlin, C., and Aurum, A. (2015). Towards a decision-making structure for selecting a research design in empirical software engineering. *Empirical Software Engineering, 20*(6), 1427-1455. doi:10.1007/s10664-014-9319-7
Wu, D., Li, J., and Bao, C. (2018). Case-based reasoning with optimized weight derived by particle swarm optimization for software effort estimation. *Soft Computing, 22*(16), 5299-5310. doi:10.1007/s00500-017-2985-9
Wu, D., Li, J., and Liang, Y. (2013a). Linear combination of multiple case-based reasoning with optimized weight for software effort estimation. *The Journal of Supercomputing, 64*(3), 898-918. doi:10.1007/s11227-010-0525-9
Wu, D., Li, J., and Liang, Y. (2013b). S3- Linear combination of multiple case-based reasoning with optimized weight for software effort estimation. *The Journal of Supercomputing, 64*(3), 898-918. doi:10.1007/s11227-010-0525-9
Yurdakurban, V., and ErdoĞan, N. (2018, 2-5 May 2018). *Comparison of machine learning methods for software project effort estimation.* Paper presented at the 2018 26th Signal Processing and Communications Applications Conference (SIU).
Zaidah, Z. (2007). Case Study as a Research Method. *Juornal Kemanusiaan,, 9, 1-6.*

LIST OF PUBLICATIONS

Journal with Imact Factor

1. **Mahmood, Y,** Kama, N, Azmi, A, Khan, AS, Ali, M. Software effort estimation accuracy prediction of machine learning techniques: A systematic performance evaluation. Softw: Pract Exper. 2022; 52(1): 39– 65. https://doi.org/10.1002/spe.3009. **(Q2, IF: 2.02)**

2. **Mahmood, Y.,** Kama, N., and Azmi, A. (2020). A systematic review of studies on use case points and expert-based estimation of software development effort. Journal of Software: Evolution and Process, 32(7), e2245. https://doi.org/10.1002/smr.22452. **(Q3, IF: 1.97)**

Indexed Journal

1. **Yasir Mahmood**, Nazri Kama, Azri Azmi and Suraya Ya'acob, "An IoT Based Home Automation Integrated Approach: Impact on Society in Sustainable Development Perspective". International Journal of Advanced Computer Science and Applications (IJACSA), 11(1), 2020. http://dx.doi.org/10.14569/IJACSA.2020.0110131. **(Q4, Cite Score: 0.17, Indexed by WoS & SCOPUS)**

2. Faizura Haneem, Hussin Abu Bakar, Nazri Kama, Nik Zalbiha Nik Mat, Razatulshima Ghazali and **Yasir Mahmood**, "Recent Progress of Blockchain Initiatives in Government" International Journal of Advanced Computer Science and Applications (IJACSA), 11(11), 2020. http://dx.doi.org/10.14569/IJACSA.2020.0111144. **(Q4, Cite Score: 0.17, Indexed by WoS & SCOPUS)**

Indexed Conference Proceedings

3. **Y. Mahmood,** N. Kama, A. Azmi and M. Ali, "Improving Estimation Accuracy Prediction of Software Development Effort: A Proposed Ensemble Model," 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), Istanbul, Turkey, 2020, pp. 1-6, June 2020, Institute of Electrical & Electronics Engineers (IEEE) doi: 10.1109/ICECCE49384.2020.9179279. https://doi.org/10.1109/ICECCE49384.2020.9179279. **(Indexed by SCOPUS)**

BOOK PUBLICATION

Kama, M. N., Basri, S., and **Mahmood, Y. (2020).** Software Requirement Change Effort Estimation: An Algorithmic Approach

ISBN: 978-8194951773

The book is globally available on Amazon Kindle, Kobo.com, Google Books and Google Play.

https://www.amazon.com/Software-Requirement-Change-Effort-Estimation/dp/8194951771
https://www.kobo.com/in/en/ebook/
https://play.google.com/store/books/
https://books.google.co.in/books/