Software Cost Estimation using Hill Climbing and Particle Swarm optimization

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR

THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

Suneeta Singh

Roll No. 21203028

SUPERVISED BY

Dr. Kuldeep Kumar
Assistant Professor

Dr. Avtar Singh
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DR. B. R. AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY
JALANDHAR – 144008, PUNJAB (INDIA)

JULY, 2023

CERTIFICATE

I hereby certify that the work, which is being presented in the dissertation, entitled "**Software Cost Estimation using Hill Climbing and Particle Swarm optimization**" by "**Suneeta Singh**" in partial fulfillment of requirements for the award of degree of M.Tech. (Computer Science and Engineering) submitted to the Department of Computer Science and Engineering of Dr. B R Ambedkar National Institute of Technology, Jalandhar, is an authentic record of my own work carried out during a period from August, 2022 to July, 2023 under the supervision of **Dr. Kuldeep Kumar**, Assistant Professor and **Dr. Avtar Singh**, Assistant Professor. The matter presented in this dissertation has not been submitted by me in any other University/Institute for the award of any degree.

Suneeta Singh
Roll No. : 21203028

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Dr. Kuldeep Kumar (Supervisor)
Assistant Professor
Department of Computer Engineering
NIT Kurukshetra

Dr. Avtar Singh(Supervisor)
Assistant Professor
Department of Computer Science & Engineering
Dr. B. R. Ambedkar NIT, Jalandhar

The M.Tech (Dissertation) Viva-Voce examination of Suneeta Singh, Roll No. 21203028, has been held on _____ and accepted.

(Signature)	(Signature)	(Signature)
External Examiner	**Supervisor(s)**	**Head of Department**

ACKNOWLEDGEMENT

Foremost, I would like to express my gratitude to my supervisor Dr. Kuldeep Kumar, Assistant Professor and Dr. Avtar Singh, Assistant Professor, Department of Computer Science and Engineering, DR B R Ambedkar National Institute of Technology, Jalandhar for the useful comments, remarks and engagement through the learning process of this master thesis. I cannot thank him enough for his tremendous support and help. He motivated and encouraged me throughout this work. Without his encouragement and guidance this project would not have materialized. I consider myself extremely fortunate to have a chance to work under his supervision. In spite of his busy schedule, he was always approachable and took his time off to guide me and gave appropriate advice.

I also wish to thank whole heartily all the faculty members of the Department of Computer Science and Engineering for the invaluable knowledge they have imparted on me and for teaching the principles in most exciting and enjoyable way. I also extend my thanks to the technical and administrative staff of the department for maintaining an excellent working facility.

I would like to thank my family for their continuous support and blessings throughout the entire process, both by keeping me harmonious and helping me putting me pieces together. I also like to thank all my batch mates for the useful discussions, constant support and encouragement during whole period of the work.

Last but not the least, I would like to thank almighty GOD for giving me enough strength and lifting me uphill this phase of life.

Suneeta Singh

ABSTRACT

Software cost estimation is a challenging and complex task during software development. It directs project managers and developers to analyze and predict costs at the beginning of the software development life cycle. The most crucial job for developing software projects is correctly estimating cost, time duration, and needed effort. It has become a significant focus in the field of Software Engineering. Software metrics are good indicators of the product's quality as they provide a quantification base for the software systems. The subjective natures of external quality attributes make estimating Software cost in early phases difficult. However, this limitation is mitigated by using internal quality attributes, which can measure software quality in early phases and act as good indicators of external quality attributes.

In the past decade, many object-oriented software metrics have been proposed and validated. Of those proposed so far, only a few have been validated using real-world applications inhibiting their use in the software industry. This thesis aims to review different models used for software cost estimation, including algorithmic, non-algorithmic, and learning-oriented models, which have been published over the last ten years. The comparison is done based on the methods, selected datasets, and metrics used in different techniques.

Further, a Hill Climbing and Particle Swarm Optimization based algorithm for Software Cost Estimation has been proposed. A scenario-based approach in combination with appropriate statistical techniques has been used in an attempt to capture the correlation between static and dynamic coupling metrics and, in turn, the structural and behavioral aspects of object oriented software systems. The dataset used for implementation are Desharnais, NASA93, Albrecht and ISBSG. We prepared a setup in which first analyses the dataset and then preprocess it using feature selection process. To select the features we used correlation coefficient to check out the relationship between dataset. Then selected most correlated feature of dataset then train the dataset on model. We evaluated many software metrics' to get the result finally. Used matrices are MSE, RMSE, MdMRE, MMRE and R squared. In the result we found that proposed algorithm are giving best result in comparison to others.

Keywords: Software Cost Estimation, Algorithmic Methods, Non-algorithmic Methods, Learning oriented Methods, Hill Climbing.

LIST OF PUBLICATIONS

1. Suneeta Singh and Kuldeep Kumar, "Software Cost Estimation: A Literature Review and Current Trends", International Conference on Secure Cyber Computing and Communications, IEEE.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Constructive Cost Model	COCOMO
Function point Analysis	FPA
Artificial Neural Network	ANN
Fuzzy Logic	FL
Mean Magnitude of Relative Error	MMRE
Mean Absolute Error	MAE
Root Mean Squared Error	RMSE
Relative Absolute Error	RAE
R Squared	R2
Principal component analysis	PCA
Mean magnitude of percentage error	MMPE
Median magnitude of relative error	Meme
Mean square error	MSE
Random Forest	RF
Hill Climbing	HC
Particle Swarm Optimization	PSO
Linear Regression	LR
K Nearest Neighbor	KNN
Simple Hill Climbing	SHC
Proportional Integral	PI
Proportional Integral Derivative	PID
Adjusted Function Points	AFP
Input Functions	INC
Output Functions	OUC
Enquiry Functions	EQC
Interface Functions	INF
Added Functions	ADD
Changed Functions	CHC
Categorical attribute Resource Level	RSL

Chapter 1

Introduction

1.1 Introduction

In the early phases of software development, it is difficult for the project managers to estimate the project's development cost accurately and precisely (G. Pavithra, 2021). It has been found that 66% of software projects suffer from their original estimates (Azzeh, M., 2013). Thus, cost estimation is the most challenging aspect of project management. It includes the size prediction of a software product to be developed, the effort needed, schedule estimation, and the overall cost estimation for the project. Software engineering could be enhances the standard procedures, from its initial phase to the final release maintenance that complete the different requirement of software project development. These standards will make sure to maintain the quality of projects at its highest level and projects should be completed timely and efficiently (Chirra, S.M.R. and Reza, H. 2019) .

The main aim of software engineering is to create systems and algorithms that grown up with fabulous presentations which are stable, liable and progressively steady. So at the time of progress procedure high authority and designers used some measurements to calculate and improve the quality of presentations(Azzeh, M., 2013). Cost estimation is the most challenging aspects of project management. It includes size prediction of a software product to be created, the effort needed, schedule estimation and finally, the overall cost estimation for the projects. So many problems are generated when the development team manages critical issues in cost estimation. So the first step is to understand and define the system to be needed for the software cost estimation (Chirra, S.M.R. and Reza, H. 2019). Software is continuously growing and changing. When hardware does not give proper result as we expect then the solution is achieved by making changes in software. These changes may create some other issues like delay in development and may be unanticipated software development(Azzeh, M., 2013).

Software quality models have the ability to perceive exact licenses of basic components to the use of different accreditation exercises starting from physical investigation to testing, manage suitable examination techniques and dynamic and static investigation. So far it helps to make steadfastness of conveying items. In software engineering different probabilistic and statistical techniques have been suggested to inclination of conjecture of coding models Butt.

The project manager defines and describes the target cost of the project (Figure 1.1). After that, model parameters are refined and adjusted until the target cost is justified (Slowik, A. 2011). For a better approach, it is crucial to analyze and estimate the risk factors associated with cost estimation and look for alternative solutions. It forces the project managers to predict and calculate risks associated with the target cost and explicate this helpful information to other stakeholders (S.A. et al., 2022). In most of the paper, machine learning techniques had been used to compute and compare the results of implementing different techniques on different datasets (G. Pavithra, 2021).

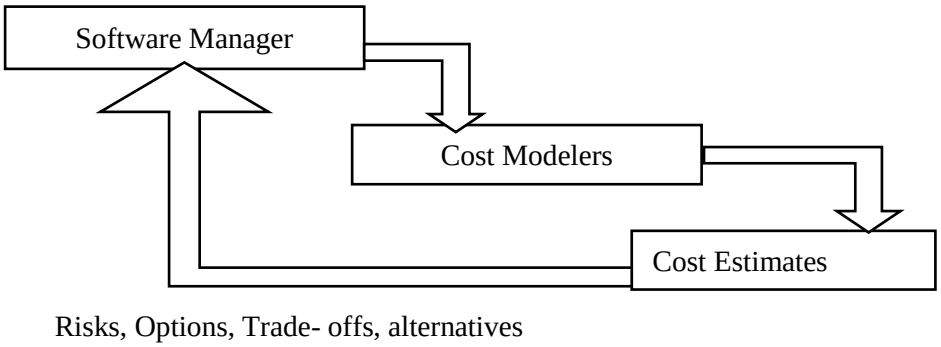

Figure 1.1 An Overview of the Cost Estimation Process

1.2 Background

Software Cost estimation has a very wide range and also have some more fields needs to explore. We tried to combine some of the methods, feature in this section.We collected the details on the basis of many factors like algorithmic, non-algorithmic and learning oriented methods. The work has to be done in this area is given here.

In the paper (Rashed Hama, A. et al. 2022) author proposed a web-based software system to estimate a user story's budget and effort. The Paper (Malathi, S. and Sridhar, S. 2011) (Bilgaiyan, S. et al. 2017) contains Cost Estimation at the beginning period of the development life cycle explained about the Fuzzy Model, Decision tree, Hybrid Techniques and compare the result with COCOMO model. These paper have common datasets Desharnais and NASA93 . The paper also contains Introduced about Fuzzy logic which is based on Soft Computing Techniques. The comparison has been done on the basis of different metrices like MRE.

In the paper (Chirra and Reza, 2019) (El Bajta *et al.*, 2015) (Anusha and Mukesh, 2013)Author given the deails about algorithmic and Non- algorithmic models. Predicted

effort using dataset attributes and performed comparison with the actual effort to measure the error based on different metrics. The discussion and result has been calculated on the basis of software metrices such as MMRE, MRE , RMSE, RAE and some paper has the other comparison criteria like complexity, duration , size and costs. Some paper like (Singal, Kumari and Sharma, 2020)(Singh, Singh and Mehta, 2021) has Discover the efficiency of DE algorithms to improve the parameter values .They proposed DE based COCOMO I and COCOMO II algorithms. Author in the paper (Hallmann *et al.*, 2018) proposed scrap rate implementation techniques on a case study and compared based on their required computational time and gets optimized results. (Kumari and Pushkar, 2018) this paper is discussed about some environmental and hybrid algorithms using COCOMO. Some paers have the datasets ISBSG, COCOMO, NASA collected from standard repository which is easily available at open source sites. The paper(Singh, Singh and Mishra, 2018) has proposed the different technique Optimizing Effort Parameter of COCOMO II Using Particle Swarm Optimization Method . PSO and tabu search implemented in this paper and gave the results on the basis of evaluation factore MMRE and MD. The paper (Kumari and Pushkar, 2018) has new approach Cuckoo Search algorithm-based approach for the prediction of effort of software development and discussed in details about COCOMO II hynrid algorithms.

In the example, like if 40 hours of work can be completed in 8 hours which is divided within 5 engineers of a project , on the other hand same work can take 40 hours if the dedication of engineer is less through out the whole project or suffering from any external issues. So, in both the cases effort is same and project has to be completed but the timeline is different. That's why make sure to manage both time and effort and count both factor while documented the projects. It is an important point in software cost esstimation.

1.3 Strengths and Weaknesses of various software cost estimation methods

From the literature, we found that various cost estimation methods have been proposed in the past. We compare these methods based on their strengths, weaknesses, and accuracy parameters. The choice of choosing a particular cost estimation method depends on the capability of the method and the nature of the project under consideration. Table 1.1 is showing a tabular comparison of the strengths and weaknesses of various cost estimation methods.

Table 1.1 Comparative analyses of different cost estimation methods

Method	Strengths	Weaknesses
COCOMO (Chirra and Reza, 2019)	Simple, takes less time and effort to estimate	Past project details are needed
FPA (Chirra and Reza, 2019)	Easy to estimate cost at first stage, Independent on the programming language	Development time, manpower has not considered
Expert judgment (Kumari and Pushkar, 2018)	Limited data needed; expert experience make estimation	Expert required of similar experience problem, Biased opinion
Neural Networks (Arslan, 2019)	Exceptionally good for non-linear relationships, more flexible	Tend to over fit, needed more computation
Fuzzy logic (Ch. and Singh, 2020)	Consider real-value instead of binary	Not well for complex Dataset
Regression Tree (Rashed Hama *et al.*, 2022)	Has better capability to manage noisy data	Suffers with high variance and low bias

1.4 Problem Statement and Contribution of Thesis

This thesis has been divided into two parts.

The first part of thesis applied Hill Climbing algorithm on datasets NASA93, DESHARNAIS, ALBRECHT, and ISBSG (R8) and compared the results with random forests, KNN and Linear regression in terms of root mean square error and mean absolute error. In the second part of thesis, applied Particle Swarm optimization on datasets NASA93, DESHARNAIS, ALBRECHT, and compared the results with same model applied on dataset of Turkish Software Industry which consists of twelve data instances.

1.5 Organization of Thesis

Thesis contains four more chapters which are as follows: Chapter 2 consists of Literature Survey. Previous works done in this research filed have been explained in detail in this chapter.

In Chapter 3, Problem Statement is explained in detail. Chapter 4 includes details about methodology used. In this chapter we can see the algorithms, uses and types of used methodology. Chapter 5 contains implementation setup and results; in this chapter we will see the work flow of implementation and about datasets and all the details of implementation step wise step. The whole work consisting of techniques and methodologies, results. Chapter 6 includes the conclusions and future scope part. The paper is concluded in this chapter.

Chapter 2

Literature Survey

2.1 Related Work

In this chapter we will see the details about how much work has to be done in Software cost estimation field till the days. So many papers have been published in this area in past. We prepared a tabular comparison of method, datasets, and approaches used in this field. These sections can give idea about the algorithms which is mostly used and current trend about the area. So that we can analyze better to get a perfect result. We divided this section into two parts first one is on the basis of algorithmic and non-algorithmic methods and second one is learning oriented methods and others.

2.1.1 Based on Algorithmic and Non Algorithmic Methods

In 2022,(Butt, S.A. et al. 2022) Author Aziz Butt et al. proposed techniques to fulfill the requirements of customer's by the help of Scrum. There is some special cost estimation technique for scrum based projects which have some boundations that makes it less useful to overcome the problems of different challenging filters using scrum. In this paper, they surveyed in many software industries to know the findings. After the survey prepared around 30 observations that focused on how much given estimation technique was valuable for accuracy purpose. They set the age criteria to know the review of most experienced developers. To overcome these challenges they design a framework to get control on cost and time of these techniques. They also proposed these techniques in different software industries for case studies and got the results(G. Pavithra 2021). This software based estimation techniques collected information as an input from members who were working and estimated the time and cost.

In the results, they got their approach for estimation to reduce the problems which came during scrum based projects development and useful for projects estimation. This estimation techniques given the new theory of estimation that was beneficial for the clients ,industries ,programmers and also fulfill the need of the customer's complete variation during the development and sent the projects within given time and cost(Aunsri and Rattarom, 2022).

The Paper "Estimation of Software Development effort: A Differential Evolution Approach "in 2020, Author Singala et. al (Singal, Kumari and Sharma, 2020) introduced the

efficient algorithm named differential evolution algorithms. This approach used in order to improve and enhance the parameter values, metrics like COCOMO do. Then we could get parameter values by different mutation strategies in differential evolution. Proposed methodology had tested on two datasets from repository. After testing results of each test were compared and analyzed with COCOMOI and COCOMOII on the base of MMRE. Finally under consideration, concluded that proposed differential evolution approach calculated the estimate effort more precisely than the original methods for two given datasets.

In 2019, the Author Reddy Chirra et. Al summarized a systematic literature review of different models including algorithmic, non- algorithmic and ML oriented methods used for software cost estimation. And they provided the strengths, weakness, accuracy, amount of data needed, duration of projects and validation techniques. According to this survey, it concluded that neural network based models prominent than other cost estimation techniques. This survey is good for beginners in research area. Parallely they proposed a future work which include the research on application LSTM recurrent neural networks for time series forecasting for any project management(Chirra and Reza, 2019).

Author Shailendra Pratap Singh et sl.(Singh, Singh and Mehta, 2021) ,in May 2018 , explained that differential evolution is very famous and effectual algorithm to solve nature related problems and this algorithm is also motivated by nature used to resolve the critical problems in different areas. The computational cost is increases when we will use DE algorithm at higher trend. The new type of DE algorithm is presented by incorporating the homeostasis adaption based mutation operator (HABMO), which manage the distinction while facing any problems. In software development this operator is used with DE algorithm to estimate the software cost, where optimization technique is used with algorithmic cost constructive model (COCOMO) in order to maintain the parametric attributes. Mainly this model is used to predict accurate efficiency and minimize the error MMRE, MSE, MMER, and RMSE with less experiment for COCOMO model. And author also wanted to show the different versions of DE, it's comparison and proposed HABDE approach(Singal, Kumari and Sharma, 2020).

In DE algorithm after applying the homeostasis adaption, we concluded that the approach was verified on COCO platform for real values and also globally analyzed with CEC'2015 for all 24 identified functions. They performed comparison between suggested approach HABDE variant and DE algorithm in order to all different factors. Result was homeostasis adaption theory could enhance on DE algorithm. Apart from this HABDE

algorithm has been able to give better result and predictions. In future we would enhance its control parameter by selected whole population dynamically(Butt *et al.*, 2022).

In 2018, the Author Sweta Kumari et. al proposed an approach based on cuckoo algorithm to predict accurate effort in software development. Cuckoo search algorithm used to discover best COCOMO II method parameter and hybridized with artificial neural network in order to get better effort prediction of software development. Then hybrid method had tested on different datasets and had been seen that proposed hybrid models give more precise and efficient results compare to other existing models in experiment. Results had been evaluated with MMRE and shown the capability and accuracy of proposed hybrid method in percentage. A comparison of computational complexity had been also done with existing approaches which proves the prominent performance of proposed approach to existing models(Singal, Kumari and Sharma, 2020).

In the paper entitled with "Optimizing Effort Parameter of COCOMO II Using Particle Swarm Optimization Method", 2018, Author Kholed Langsari et. al stated that COCOMO II model is the best model to evaluate effort parameter. In this paper Author used Particle Swarm Optimization (PSO) to optimize the parameters in a model of COCOMO II. Author implemented proposed method on Turkish Software Industry dataset which has 12 data items. They show that this method could efficiently manage uncertainty in inputs and improve the reliability of software effort. In the result , they found that accuracy is high and significantly minimize the error MMRE(Langsari, Sarno and Sholiq, 2018).

In 2015, the Author Bajta et. el said about a systematic mapping study (SMS) to summarize the software cost estimation in the terms of Global Software Development (GSD) research in this paper they answered the nine mapping questions. Researchers selected some 16 articles and further classified them according to 9 criteria: source of publication, year of publication, type of research, research approach, type of contribution, software cost estimation techniques, activity, cost drivers and cost estimation performances for GSD projects. In the result they found that, interest for software cost estimation for GSD projects had grown up in earlier years. Mostly they focused on concept. Models were the most outstanding and dominant in the software cost estimation for GSD research and software development cost was being known as predominant. Selecting empirical solutions was more trustable path for researchers in software cost estimation for GSD(El Bajta *et al.*, 2015).

In 2014, Author Yansi Keim et. al summarized the cost estimation techniques COCOMOII, COCOMO, Putnam, Steer and Estimacs based on different parameter implementation ability, extensibility, flexibility and traceability for software cost estimation.

Most researchers focused and concerned about the inability of estimating accurate cost in software development. So this concern was become more pressing when development cost were going to increase. In result, researcher consider towards using a good thought of implementing software cost estimating tools. According to this analysis, it was difficult to say which model was outstanding on the basis criteria size, time and other factors. But COCOMO- II models were largely used and also had broad prospect(Heemstra, 1990)

2.1.2 Based on Learning Oriented Methods and Others

By the study of paper in 2022(Rashed Hama *et al.*, 2022), researcher Ako Rashed Hamma et al. stated a suitable model that combines genetic ,inheritate theory and topographical information system which was used to evaluate minimum operational , construction and usability flow cost (Fmin). The Fmin model was created to evaluate operational costs of water treatment units which was wasted , also for its building cost and can be reuse this water for agricultural purpose in Sulaimani city(Ch* and Singh, 2020). For the optimization purpose they used the techniques that consider on datasets by doing 31 small sized DTU and their reusability in 827 different greenery areas within the city. A transportation matrix was also designed to calculate Fmin flow from DTUs to GRs. For this process we need required length of pipes and path that conveying the flow using GIS network analysis Origin Destination (OD) tools. In the results, this proposed model was giving reliable solution and this algorithm is enough for considering all the options.

Because of this study they invent the optimized GA matrix and GIS tools to create best DTU sizes for agricultural purposes in green areas in Sulaimani city. This model was able to covers whole city in large quantity of data. These combined algorithms performed optimize and reliable solution and gave best successful results by taking different scenarios and comparisons. Genetic algorithm gives best and absolute results as ignoring local optum and considering global optum(Singal, Kumari and Sharma, 2020).

In 2022(Zhou, Etemadi and Mardon, 2022), Author Gordon Zhou et al. in paper " Machine learning –based cost predictive model for better operating expenditure estimations of U.S. light rail transit projects " explained that when they were planning to predict inaccurate OPEX (operating expenditures) that time new LRT(Light Rail Transit) projects were underrated by 45% of costs. When OPEX got increases projects level then local agencies turned down their public agencies to control within the annual budget. Then this research focused to release more accurate LRT OPEX model for prediction of project

planning stage. Between 2008 to 2018 they used near about 22 LRT system to use some traditional statistics and ML based algorithms in US from different governmental public databases(Farsi *et al.*, 2021). In this study they generally practiced on summation of unit cost estimations which usually unsuccessful due to not proper work on planning phase. Some methodologies which already existed are regression based; uses system based elements but did not enhanced the predictive results from unit cost based. By the research practices, it was improved and released more accurately. It was easy to use and reliable ML based predictive model by using environmental and LRT related attributes under consideration. They compared results by ANN, random forest, K nearest neighbor, SVM and regression methods(Chirra and Reza, 2019).

In result they analyzed that ML based algorithms gave better results in comparison to MLR based model in terms of best fit and errors. The Accurate method was KNN in different factors as high R2 and low MAE ,RMSE and MAPE .After comparing results, the unit cost based method came in industry and earlier research. Using regression it was signified and clear that accuracy was improved which will quite easy to decide public transit agencies were useful in real time to predict new LRT projects for future OPEX across the world(Aunsri and Rattarom, 2022).

In September 2021, Author G. Pavithra, in the paper "The Software Cost Estimation and Cost-Efficient Fees Structure Format for Educational Institutions", wanted to discuss the application and use of learning management system. Author prepared to implement software and cost comparison with existing methods and also suggests whether to create or buy the software for educational institutions with efficient fees format. They introduce all types of cost estimation methods algorithmic and non-algorithmic with their pros and cons. And after the review of several studies, author finds many reasons for software projects failures in last few years. According to this analysis concluded that the main causes of software project failures are not correct prediction in starting stage of the projects. So after this estimation gets correct and reliable calculations(Rattarom, 2021).

It is very important to know all the aspects of estimation methods for choosing best methods. As we know evolution of each estimation methods depends on each factor like beginning and deadline of project, complexity, expertise of developer and methods of projects. Author had been represented in this paper for getting brief performance of some evolution metrics and actual estimation method. And presented fee format for educational institution with efficiency.

In 2020 paper, the Author Shiva Suryanarayana Ch. et. al proposed a machine learning based model fuzzy model in their paper and let some characteristics of datasets which was utilized for analysis and kind of intelligence. In this paper author found that COCOMO dataset is prominent than others. They comprised the different evaluation parameter MMRE and MRE with fuzzy method and their proposed method. They got that neural network techniques were continuously used followed by hybrid techniques and then fuzzy logic. This paper may help to beginners for research purposes as they described(Ch. and Singh, 2020).

In 2019, the Author Farrukh Arslan described about to build a new software cost estimation approach using ML techniques. In this paper 13 ML techniques analyzed on two datasets. The evaluation criteria were R2, MAE, RMAE, RAE and RSE. The main focus of proposed approach is to estimate effort with dataset and compare those with actual effort to calculate error using different evaluation criteria. And final conclusion is that the random forest has given good result using first dataset and another approach regression, KStar gave good results on other datasets (Ch. and Singh, 2020). In future work author proposed to include more datasets to have wider direction and verity in inputs to get better and accurate estimation results.

In the paper entitled with "Software Development Effort Estimation Using Random Forests: An Empirical Study and Evaluation", Author Abdelali Zakran et. al, 2019, seen that among all the ML techniques Regression tree based model gained best result due to their generalized ability and understandability. But in this paper author did few studies and investigations of potential of Random forests in software cost estimation. Author designed that RF model adjusted empirically by changing its key parameters. They analyzed the impact of RF model accuracy with efficient tuning at training stage. Author evaluated the results and performance of RF model and compares it with Regression Tree. Their comparison was on the basis of holdout validation using five datasets named ISBSG R8, Tukutuku, COCOMO, Desharnais and Albrecht. Author found that adjusted Random forests performed better than the Regression tree on the all evaluation criteria. They used three broadly known accuracy parameters Pred, MMRE and MdMRE. Author also found that proposed model was giving better results than some recent techniques used in literature review for software effort estimation(Zakrani, Hain and Namir, 2018).

In the paper named "Comparison of different methods for scrap rate estimation in sampling –based tolerance – cost – optimization" 2018, Author Martin Hallmann et. al wanted to present some methods for scrapping methods estimation in sampling based tolerance cost optimization. In this paper proposed method implemented on case study and

compared based on their required computational time and get optimized results. The understanding of research was to compare different scrap rate estimation for this optimization and give better suggestion to future researchers and practitioners(Hallmann *et al.*, 2018).

In 2018, Author Tribhuvan singha et. al stated that IEAM-RP (Improved Environmental Adaptation Method with Real Parameter) was used to predict the required effort in software product development. This experiment was also analyzed by NASHA software project dataset to check the efficiency of IEAM-RP. The overall finally result and performance on the basis of MMRE and PRED values described that IEAM-RP was better in effort prediction which was required for software development(Singh, Singh and Mishra, 2018). In this paper author proposed this new approach for cost estimation process.

In 2017, the Author Singh Bedi et. al introduced about fuzzy logic which was based on soft computing techniques and gave outstanding performance such as accurate estimation. The earlier research represented a new approach with expected results and after comparing with COCOMO techniques it had been seen that accuracy rate was improved by proposed approach. This proposed method was simple and effective and implemented using MATLAB's fuzzy logic toolbox. The results got by proposed method that encourage to represent the feature of proposed techniques in software companies for software development efforts estimation(Bedi and Singh, 2017).

In 2017, the Author Bilgaiyan et. al tried to represents the systematic survey in agile software development of cost estimation. This paper might help for agile users to get out through the current trends. Author compared the results with different existing approaches. For the future work they introduce some software effort and cost estimation using soft computing techniques. They reviewed through the answering of research questions and comparing the results with existing agile software development approach to proposed approach. They also showed the results for water fall model and agile model(Bilgaiyan *et al.*, 2017).

In 2017 the paper "A Cost Estimation Approach for IoT Modular Architectures implementations in Legacy Systems" Author Tedeschi et. Al proposed about base to create an innovative conceptual cost estimation model to implement trendy infrastructure for legacy systems. In this paper the proposed model aspects some effects on the cost of different Internet of things architectures like data collection, complexity, protocols, duration, security etc. So the researchers want extra implementation on cost models to follow the

organizations in order to get more cost effective architectures for legacy systems in this modern era(Tedeschi *et al.*, 2018).

In 2016, Author Dheeraj Kapoor et. al, the paper entitled with "Software Cost Estimation Techniques – A review of Literature" explained about all the cost estimation techniques like algorithmic, non-algorithmic and learning oriented methods and their impact on the next papers. They also described about latest trends about cost estimation. Researchers also compared the results on the basis of different criteria like complexity, accuracy, precision and duration of projects. They reviewed over last 20 years papers and concluded that all the papers include existing and latest trends techniques. For the future aspects they guided in the directions to choose best measurement parameters for cost estimation with special computational intelligence techniques. The main concern was to find the existence of different features with frequencies(R.K., 2014).

In 2013, the Author Anusha M.E. et. al in this paper dealt with research and progress to find out best software cost estimation models to achieve better accuracy on software cost estimation. We all know the performance of each method depends on some of the different parameters, factors like complexity, duration, and expertise and development method of the projects. According to some analysis author presented some evaluation metrics and actual estimation for explaining the production of estimation method. For the future perspective they tried to improve the results of existing approaches and also giving new estimation methods based on current requirements of software projects(E and E, 2013).

In 2011, the Author S. Malathi et. al developed new software effort estimation methods for projects which used categorical and numerical data caused by analogy, linguistic quantifier and fuzzy approach. So they analyzed that existing historical datasets produced better and precise outputs in comparison with dataset analyzed by previous methodologies. The best thing is, this method can take care the imprecision and uncertainty quite smoothly at the time of explaining software project. Final conclusion was observed that proposed methods was effective and gave average effort for the datasets of software projects(Malathi and Sridhar, 2011).

Table 2.1 Comparative analysis of selected published works on software cost estimation

Author Detail	**Approach**	**Datasets**	**Techniques**	**Targeted Techniques / Metrics**
(Butt *et al.*, 2022)	A web-based software system is implemented to estimate a user story's budget and effort.	-	Scrum method	Compared with Algorithmic and Non-algorithmic

(Ch. and Singh, 2020)	Cost Estimation at the beginning period of the development life cycle	Desharnais Dataset, NASA 93 Dataset	Fuzzy Model, Decision tree, Hybrid Techniques	COCOMO
(Singal, Kumari and Sharma, 2020)	Discover the efficiency of DE algorithms to improve the parameter values	COCOMO 81, NASA 93 Dataset	DE-based COCOMO I and COCOMO II based on MMRE	Original COCOMO
(Chirra and Reza, 2019)	Performed comparison of software effort, cost, and size estimation based on all techniques	-	Algorithmic Non-Algorithmic Learning oriented methods	Based on accuracy
(Arslan, 2019)	Predicted effort using dataset attributes and performed comparison with the actual effort to measure the error based on different metrics.	USP05-FT, USP05	Algorithmic Non-algorithmic Learning oriented methods	Based on MAE, RMAE, RAE, RRSE
(Singh, Singh and Mehta, 2021)	Introduce the DE algorithm to estimate the cost of software	-	Homeostasis adaption-based mutation operator differential evolution (HABDE) algorithm	COCOMO based on MMRE, MSE, RMSE
(Sai Mohan Reddy Chirra and Reza, H 2019)	Performed comparison of software effort ,cost and size estimation based on all techniques		Algorithmic Non-algorithmic Learning oriented methods	Based on accuracy
(Kumari and Pushkar, 2018)	Cuckoo Search algorithm-based approach for the prediction of effort of software development.	COCOMO 81,COCOMO NASA	Hybrid algorithm, CS COCOMO II	Original COCOMO II based on MMRE
(Abdelali Zakrani *et. al*., 2018)	Software Development Effort Estimation Using Random Forests	ISBSG, COCOMO, TUKUTUKU, DESHARNAIS, ALBRECHT	Regression trees, Random Forests	MMRE, MdMRE
(Kholed Langsari et. at., 2018)	Optimizing Effort Parameter of COCOMO II Using Particle Swarm Optimization Method		Particle Swarm Optimization , Tabu search	MMRE, MD
(Hallmann *et al.*, 2018)	Implemented on a case study and compared based on their required computational time and gets optimized results	-	Scrap rate estimation techniques	
(Singh, Singh and Mishra, 2018)	IEAM-RP is used to predict the required effort in software product development	-	IEAM-RP (Improved Environmental Adaptation Method with Real Parameter)	Based on MMRE, MSE
(Bedi and Singh, 2017)	Introduced about Fuzzy logic which is based on Soft Computing Techniques	-	Fuzzy logic	Compared with COCOMO based on MRE

(Bilgaiyan *et. al.*, 2017)	A systematic survey and review in Agile Software Development of cost Estimation.		Water fall model and Agile Model	Algorithmic Non-algorithmic Learning oriented methods based on MRE, MMRE
(Tedeschi *et al.*, 2018)	Create an innovative conceptual cost estimation model to implement new IoT architectures for legacy systems	-	Cost Assessment Model for Smart Manufacturing Implementation in Legacy Systems (CAM-SMILS)	
(El Bajta *et al.*, 2015)	This paper summarizes software cost estimation in the context of GSD research and answers the nine mapping questions	-	Algorithmic and Non-Algorithmic	Based on MRE, MMRE, RMSE
(Keim *et al.*, 2014)	Summarized the cost Estimation techniques Based on different criteria	-	COCOMO-II, COCOMO, Putnam, Steer and Estimate based	Based on implementation ability, extensibility, flexibility, and traceability for software cost
(Anusha and Mukesh, 2013)	It deals with the effort and progress of finding out the best method for Cost Estimation to give better accuracy on Software Cost Estimation	COCOMO Dataset	Algorithmic Non-algorithmic Learning oriented methods	Based on the complexity, Duration, size, ability, and development method of projects
(Malathi and Sridhar, 2011)	Estimate software Effort on categorical, numerical data using by analogy and fuzzy approach.	Desharnais Dataset, NASA 93, NASA 60	Analogy, Linguistic quantifier, and Fuzzy approach	COCOMO based on effort

2.2 Types of Cost Estimation Methods

Three types of methods are available to estimate the software cost:

2.2.1 Algorithmic Methods

In this method, mathematical models are used to estimate the software cost. These mathematical models are designed on the basis of research, previous data, inputs such as numbers of functions to be executed, lines of source code, and other functionalities such as design, methods, skills, etc. The algorithmic models are very popular and widely used. Some of the commonly used algorithmic methods are:

a. Constructive Cost Model (COCOMO): COCOMO is a very basic and widely used cost estimation model. COCOMO analysis is based on previously selected projects. It is a regression model that considers lines of code for cost estimation. It further considers factors such as skills, experience, qualification of the team, and development environment(Ch. and Singh, 2020)(Chirra and Reza, 2019)**.**

We are adding the point in the definition; the Constructive Cost Model uses metrics that guide about its operation. The two metrics which help in the calculations guidance are Function Points (FP) and Object Point (OP) and parallelly they ensured the evaluations are in the line with codes relating to KLOC and KDSI(Chirra and Reza, 2019).

We know, there are some different equations used with COCOMO. Below, we are giving the two equations that used with COCMO to calculate scheduled time and the effort. The two equations that is 2.1 and 2.2 are (Chirra and Reza, 2019) given below :

$$TDEV = c(PM)d \quad (2.1)$$

$$PM = a(KDSI) * EAF \quad (2.2)$$

All abbreviation given in equation stands for different purpose that affects the software cost. Scheduled time is measured in months.

PM is Person-Months

EAF is Effort Adjustment Factor

TDEV is Scheduled time

KDSI is Number of lines of code (that is denoted in thousands)

Initially a, b, c and d are constants which result from the mode that is used in estimating cost.

b. Function point Analysis (FPA): It is a method to calculate software size based on its functional view. It considers five factors—numbers of input, output, queries, internal logical files, and external logical files to compute the size of the software(Dewi, Subriadi and Sholiq, 2017). FPA is a set of rules of functional size measurement. This software is used to relocate production application at project implementation.

To analyze the functionality given by software, we used FPA and unadjusted function point is the basic unit of measurement.

These are the some objectives of FPA:

- It measures the functionality of the user requests and receives.
- It measures software development and maintenance separately and independently of the techniques that is used for implementation.
- To reduce the overhead of measurement process, FPA should be sufficient.

- Among different projects and organizations, FPA measurement should be consistent.

c. Putnam Model: The model equations by investigating the many projects and calculate the manpower distribution. It is a mathematical model to estimate the time and effort required to complete a project of a given size. According to this model, as the software development time is decreased, the effort required to build the software increases to the fourth power of development time reduction (Butt *et al.*, 2022). In the functional equations, relation between the size and effort is non- linear and sensitive to delivery time(Chirra and Reza, 2019).

In the combination of this model, SLIM is a tool that is used to estimate the cost and allowing workforce scheduling. There are some set of aspects that is need to be consider relating to software development life cycle. The uncertainty may lead to the inaccurate cost estimation in the software size. In all this process, the advantage of this model is that, it is based on two factors that are unit to cost estimation Time and Size. This unit needs less parameters in comparison with COCMO81 and COCOMOII(Chirra and Reza, 2019).

2.2.2 Non-Algorithmic Methods

a. Expert Judgment: It is the one of the traditional techniques which is used in software cost estimation in starting phase of software development. In this method, domain experts or a group of senior members are consulted for the cost estimation. The project cost is estimated based on their experience and the project specifications. The expert judgment also used to estimate cost of given project even on limited expertise of estimator to any particular field. Experts who have knowledge and experience on a particular field for cost estimation are put their skills in given projects at a better position. These techniques come in picture when there are limitations on effectiveness and data collection. Delphi technique is the example that follows same approach. In this technique, an expert would be able to give an honest and experienced opinion with their best knowledge and understanding when it comes to an impact of system to be incorporated. There are also some of the setbacks in the software projects, so that documenting them with best of expertise is tedious and difficult. There is no validation for this method because estimation depends on current domain knowledge and experiences of the expert(Chirra and Reza, 2019).

Some of the basic disadvantage is biased and optimistic while obtaining cost estimation. The experts who give judgments have human emotion that may be their emotions influence the judgment process. So that the possibility of pessimism, optimism and bias judgment process might be influenced.

b. Top-Down Estimation Method: In this method, the initial overall cost for the project is estimated at the abstract/top level based on the complete properties and characteristics of the software project. Then, the cost is estimated and refined at lower sub-component levels. This method is used for estimating cost when global characteristics of the software project are known. This method can be used when less or limited historical data to be known about similar projects. When the projects are in early stage then this technique is more beneficial, because at this stage we do not need detailed information related to project. Top Down estimation technique is used for taking high level decisions where planning phase is long. Unlike other cost estimation techniques , this method concentrates on management and integration activities .The ballpark estimate is very inaccurate, it is the main drawback of this technique(Butt *et al.*, 2022).

c. Bottom-Up Estimation Method: This is the exact opposite method of Top Down estimation. In this method, initially, the cost of each sub-component of the software system is evaluated. The cost estimates of sub-components are then used to estimate the cost of the whole project(Butt *et al.*, 2022). In this technique we obtain a proper estimate that is the accumulation of estimates of smaller software components. Depend on the variety of projects both techniques are useful. For small projects, this method is best fitted to estimate the cost.

In bottom – up estimation, we take each small work package in a schedule and assign a cost to that and then combine them to get the overall cost of the projects. This process can give very accurate and create detailed estimation. Moreover, there are some disadvantages of bottom – up estimation.

The accuracy is very high, but this all process will take too much time to reach this level of estimation, that means it will be expensive to create. So that to comeback with this disadvantage preferable to do high – level estimation which is top – down estimate for the whole projects and then can go for detailed information that are completed in short term(Ch. and Singh, 2020).

d. Estimation by Analogy: The cost of the project is estimated based on the cost of existing similar projects which is completed earlier, whose information and development details are known(Chirra and Reza, 2019).

e. Price –to – win estimation: In this method, budget of customer is more important than functional behavior of the software and projects cost. It Means software project is directly

proportional to the budget of customer. This method is not so much suggested because it focuses on the customer's budget and capacity rather than focusing on software functionality. Because of customer's budget, accuracy effected very rapidly and resultant as low accuracy. We do not need previous data at all in this method as customer will fulfill the requirements and independent on historical data. This may cause the delay in all process like development and delivery time and it may pressurize the team to work overtime, so that it is not a good practice. We will validate this approach based on customer's budget and person- month factor. The big advantage of Price-to win method is that the estimation will not require extra money than the customer's budget. And Software quality is also compromised because functionality of software is not more important than customer's budget(Ch. and Singh, 2020).

2.2.3 Learning Oriented Methods

Various learning-oriented models have been used for software cost estimations. Some of the commonly used learning-oriented methods are:

a. Neural Network based methods: These methods are based on using machine learning models for cost estimation. Various neural network based models such as Feed-forward neural networks, Recurrent neural networks, Artificial neural network(ANN), Convolutional Neural Network (CNN), Radial basis function (RBF) networks, and Neuro-fuzzy networks have been used in the past for cost estimation(Chirra and Reza, 2019).

The ANN approach is used for the software cost estimation when the pattern of data needs to understand before the estimation of project cost. For the cost estimation, it is used to differentiate data into clustering, predefined classes and prediction. For example, in some projects like stock exchange market, this method is used for predictions using nature of the previous data. The unique advantage of this model is that it is best to capture non-linear relationships. The model contains a deep neural net that requires lesser features. This neural net has the capability to generate its own feature which is one of the advantages when training dataset has limited features.

This model also has some limitations as they try to over fit the data in software cost estimation process. The Model also requires huge amount of power for computation which may not be present at all times. If the model perform good as like as in efficiently operation position when sometimes there is less computation power could generates high computation power by increasing the cost then it was better(Bedi and Singh, 2017).

b. Fuzzy Logic based models: Fuzzy logic has been used to resolve many critical problems in cost estimation that were difficult to manage by algorithmic and non-algorithmic methods. Fuzzy logic is very useful in improving the performance of existing cost-estimating models. They enhance the accuracy by enabling semantic representation of incoming data (input) and outgoing data (output) of the model(Ch. and Singh, 2020)(Bedi and Singh, 2017).

This is very good method and also used to decrease the complexity of completed project as well as can increase the access ability of control theory. Some of the places, the fuzzy logic model are used for software effort estimations. There are more applications of fuzzy logic that is able to overcome to the existed inherent problems of software effort estimation. Fuzzy logic is also very essential to improve the feature of existed estimating models rather than only predicting efforts. Fuzzy logic enables semantic representation of incoming data (input) and outgoing data (output) of a model to get better accuracy. Many software data are calculated on nominal or ordinal scale type as like it is specially fitted for effort estimation that is mainly a case of sematic values. We have to follow these three steps while using FL(Chirra and Reza, 2019):

- Fuzzification that converts a Crisp into a fuzzy set.
- Fuzzy Rule- based System, in this step after taking all crisp input has been fuzzified into their linguistic values the inference engine then it derives their linguistic values respectively.
- Defuzzification is the last step that involves the conversion of fuzzy output into crisp output.

Chapter 3

Problem Statement

3.1 Problem Definition

Software cost estimation is a challenging and complex task during software development. It directs project managers and developers to analyze and predict costs at the beginning of the software development life cycle. The most important job for developing software projects is correctly estimating cost, time duration, and needed effort.

As this work is based on new algorithm introduced for Software Cost Estimation that is Hill Climbing and Particle Swarm Optimization. A systematic approach which has features of statistical techniques has been given in order to get proper correlation coefficient between static and dynamic behavior of metrics. This will output the structural and behavioral features of objects of software systems. So that the problem of cost estimation has been defined.

3.2 Objectives

The objective of this thesis is to address the cost estimation problem for software using Hill Climbing and Particle Swarm Optimization to reduce the error and achieve better accuracy. To get this aim, we used hill climbing choosing random starting point. To evaluate the result we use regression metrics RMSE, MSE, Mean magnitude of relative error, Median magnitude of relative error (MdMRE), Mean magnitude of percentage error and Correlation coefficient R square (R ^2). These are the regression matrix, so that we will use these to get the output as will give regression types input for our model.

We use Hill climbing for the purpose of getting high accuracy as it is an optimization algorithm and on the other hand in the next part of this thesis we applied Particle Swarm Optimization algorithms on some of the datasets.

3.3 Contribution

We have divided the thesis into three parts:

- First we reviewed last ten years papers and gave the summarize details which is included in chapter 2 that is literature survey. Provided there tabular comparison of methods, datasets, approaches used in last ten years in the field of software cost estimation.
- In the next part of thesis applied Hill Climbing algorithm on datasets NASA93, DESHARNAIS, ALBRECHT, and ISBSG (R8) and compared the results with random forests, KNN and Linear regression in terms of root mean square error and mean absolute error.

  In this part of implementation first try to find the characteristics of dataset by applying correlation coefficient and then applied algorithm. For the result analysis we use regression metrics. In the result we get that for some of the metrics Hill climbing perform best using thee datasets. Result is showed in tabular form.
- In the last part of thesis, applied Particle Swarm optimization on datasets NASA93, DESHARNAIS, ALBRECHT, and compared the results with same model applied on dataset of Turkish Software Industry which consists of twelve data instances.

  In the analysis part we compared the result with these three datasets on the basis of some metrics like MSE, RMSE, MMRE and R^2 etc.

Chapter 4

Methodology Used

4.1 Hill climbing

4.1.1 Introduction

Hill climbing is an optimization algorithm that is historically calling used for maximum task. One of the previous algorithm gradient decent used for local extreme but we replace it by hill climbing. Hill Climbing uses random local search to choose direction and size of each Step.

Hill climbing is an evolutionary type algorithm who use two size of mutation operator for implementation. It uses feedback informtion for generating solutions. The used algorithm is describes about climbing a mountain. It includes each steps while climbing. So, initially the location is selected randomly to climb a mountain according to algorithm and it is going up toward higher side of mountain in each step untill it reaches on highest point of mountain. As shown in Figure 4.1 it starts from initial node, moving towards neighbor nodes and comparing the values with nieghbouring nodes. If the value of current node is higher than other neighbouring value then it is treated as maximum value , otherwise it is replaced with highest neighbour node value in order to complete the mountain climbing process. The iteration is repeated again and again till highest point. And finally optimal solution is selected as current solution in each step till optimal solution gives a good performance relatively.

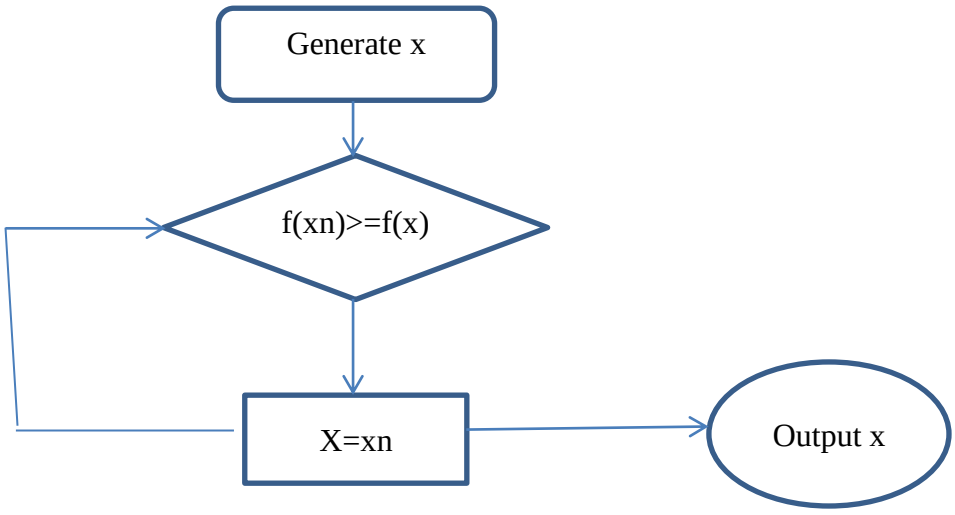

Figure 4.1 Flow chart of algorithm

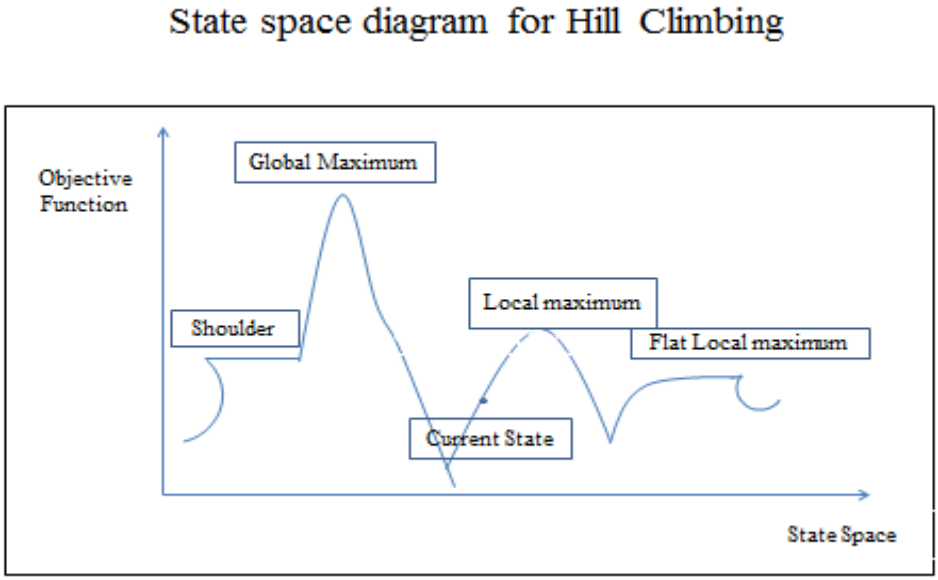

Figure 4.2 State space diagram for Hill Climbing

The main advantage of hill climbing algorithm is, we do not need traversal process to get the optimal solution. It pick out higher value nodes through a heuristic method , that highly improves the efficiency. In this process we do not need to remember earlier steps, that gives a memory saving to getting optimal solution from big number of parameter space.

Hill climbing also have some disadvantages , like in Figure 4.2 , if a node has higher value than it's just neighbor , then this node value will treated as optimal result but it may not be the highest point for whole function, so in this case we can not get satisfactory result while calculating.

About the diagram :

x- axis : It denotes the state space , where our algorithm have to reach

y- axis : It denotes objective function value with respect to a particular table

In fig. 4.2 , the nature of following points are given below from graph:

Local maximum : This is one of the state in graph which is better than it's neighbor value but it may be possible to exist a state which is better than local maximum.

Global maximum: This state has best posslible value from all states because objective function at this stage has maximum value.

Flat / Plateau local maximum: This is a flat region of function value where all neighbor states have the same value. Function value remains unchange through the certain range.

Current state : This is the region where we are currently present in this search.

Shoulder : It is an unfill edge, plateau type.

In Hill climbing , we used to follow 3 random number distribution:

- First we generate initial weight matrix W.
- We have to select a random number who determines the step wise variables.
- Then we have to calculate step matrix dNW which have all new random number components.

4.1.2 Applications of Hill Climbing

- The hill climbing is a very basic optimization algorithm that is very easy to start. We can improve the performance of initial structure using hill climbing.
- Lin et al. designed a 1-D photonic cavity based on hill climbing and deterministic method. It gives the initial structure of crystal features.

  Initial phase of one- dimensional photonic crystal is quadratically tapered and filling fractions are also same. It has been very effective in earlier work.
- The nanophotonic optimization devices have complex problems. And, it is very necessary to get out the optimal solutions for full parameter and objective functions also have complicated form. Therefore hill climbing may not be good method to design nanophotonic devices contain complex functions. But initial steps have been given that we can improve the further performance of device and also creates effective functions.

4.1.3 Advantages and Disadvantages of Hill climbing

Advantages

- Hill Climbing is simple, convenient algorithm which is easy to understand and implement.
- It is very widely usable for optimization problem with huge parameters and complex problems.
- It is very good in finding local optima; it will give a good solution very recently.
- Hill Climbing can be modify very easily and extended to additional constraints.

Disadvantages

- Hill Climbing can be stop at local optima and that’s why it may not reach to global optima of problem.
- This algorithm is very sensitive to choose in the initial stage and poor selection in initial stage

may result poor in final stage.

- It is very bad to exploring search space; because of this it's unable to reach the better solution.
- It has also some demerits and does not give optimized results than other optimization algorithm like genetic and simulated annealing for some other types of problems.

4.1.4 Types of Hill Climbing

1. **Simple Hill Climbing**

   SHC evaluates the neighboring node value one by one and choose the node which has maximum value from all the neighbor nodes and optimizes the current cost. Then current cost will be like next node.

2. **Steepest – Ascent Hill Climbing**

   This method first evaluates all the neighboring nodes and then chooses the node that is close to the solution step as a next node. All the types of HC have only minor differences.

3. **Stochastic Hill Climbing**

   This method is little different from others. It just chooses a neighbor node randomly and based on efficiency and improvement of that neighbor node decides whether to proceed with that node or move to next node by leaving that one. In short we can say, it does not analyze the entire neighbor node before deciding which node to choose in starting phase.

4.2 Particle Swarm Optimization

4.2.1 Introduction

PSO has become one of the most widely used algorithms based on its intelligence and due to (Slowik, A. 2011) its flexibility and simplicity. It was introduced by Kennedy and Eberhart in 1995.Particle swarm optimization refer to swarm behavior, like birds and fishes called swarm intelligence.

PSO applies randomness rather than using mutations algorithm. The PSO algorithm covers the area of objective function by updating the details like path of the particle, name of particle, this type of connection path made by quasi-stochastic vector. The movement of particle consists of two components , first one is stochastic component and other is deterministic component(Slowik, A. 2011).

4.2.2 Flow Chart

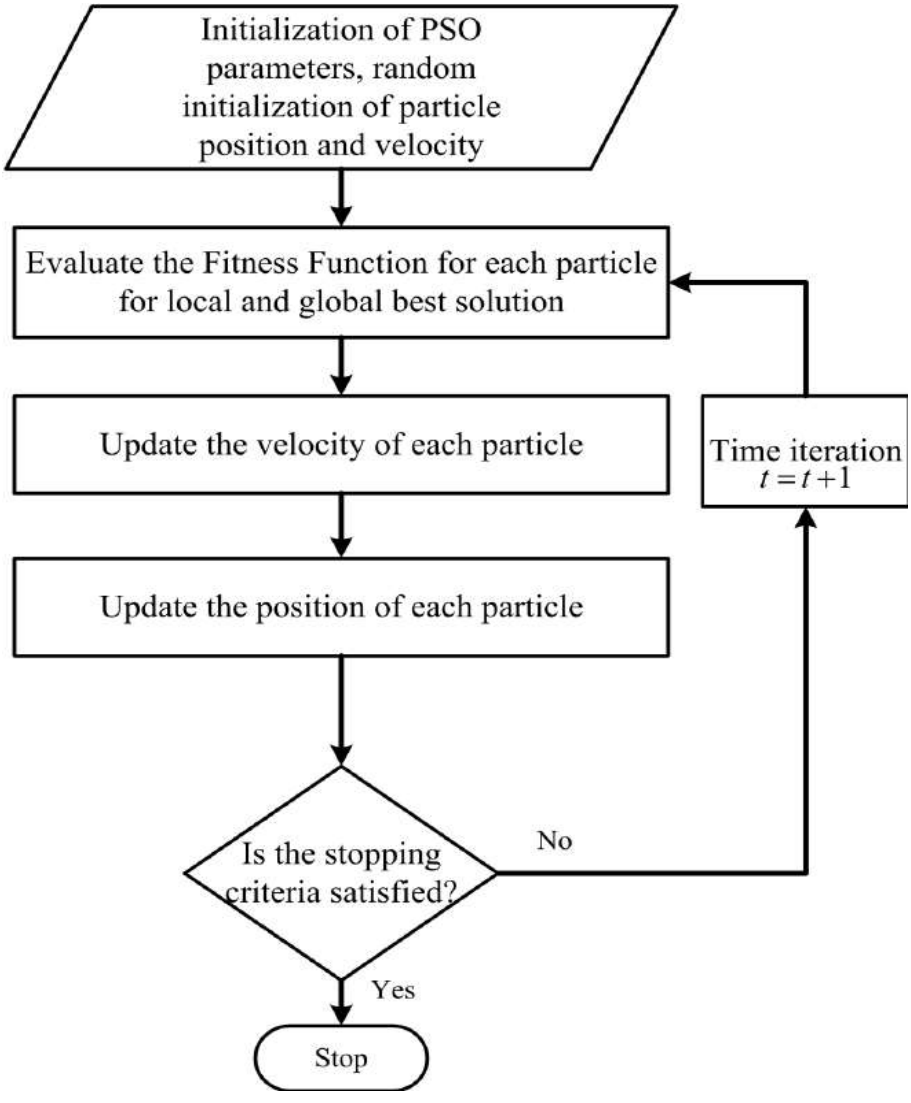

Figure 4.3 Flow Chart of PSO

The concept behind the algorithm is to create particles swarm that move in the problem space to search their goal or a better place that fulfills their requirements generated from fitness function. This algorithm follows the nature analogy with birds: Like a bird flies in the nature to search a best place to take rest. The place could be combination of features such as searching of food, searching of water, place to live or any other related characteristics. There are two main idea come about this optimization properties based on this concept (Figure 4.3)(Tedeschi, S. *et al.* 2018).

- Based on the present position of a single particle, the goodness of single particle can be determined. This will be useful to collect and share knowledge from other particles and also to explore knowledge from its problem space.
- The velocity in each particle that made them to move in an unknown problem space is because of stochastic nature in each particle. The combined property gives good initial classification of swarm that enables the details information of problem space and also makes a huge possibility of finding best and efficient solutions.

4.2.3 Applications of PSO

PSO algorithms increase exponentially in its initial development. This algorithm is focusing on the main area of artificial neural network training, Control strategy and ingredient Mix Optimization.

After few years, in a survey some author filed the exponential growth is an application of PSO and also verified near about 700 documentation in PSO. Author also classified the research area of application in 26 categories that is more searched (Tedeschi, S. *et al.* 2018):

- **Antennas -** It is taken as optimal control and design of array. Apart from this, it has so many drawbacks improvement and miniaturization. It optimizes the design system and control structure.
- **Control –** It used in especially PI and PID controllers. The terms explain in abbreviation part. Based on its present output and its expected value, previous errors and future error predicted, the application system can control all the processes.
- **Distribution Networks –** It is basically used in design and dispatches the loads in electricity networks.
- **Electronics and Electromagnetics –** This application is very clean and clear, some of the main devices and applications that are used from this application in PSO are: fuel cells,

optimization in semi-conductors, on- chip inductors and generic design. This gives us to a real life usable application.

- **Image and Video –** This is very wide and have details documented works in this area. So that as the name defined it has very wide range application and uses like detection of face, recognition of face, image segmentation and image retrieval, image fusion, contrast enhancement. This application will be also used in daily life.
- **Power Systems and Plants –** This application has main focused on power optimization and control. Have some other applications that are useful in electric system and performance like: photovoltaic systems control, load forecasting and minimization of power loss.
- **Scheduling –** Scheduling is defined for the purpose of management where we will manage the jobs and tasks. The especial concerns of this application is to focus on flow that is flow of shop scheduling, flow of task management, job and shop scheduling, task scheduling in distributed systems. Apart from this some other scheduling problems related to same application are given as train and project, production management and assembly.

Chapter 5

Implementation Setup and Results

5.1 Proposed Cost Estimation Model

We are going to follow these steps for the further implementations work. First we will discuss about the datasets, preprocessing, Methodology used, evolution criteria and then results. And we will finalize our results by comparing it with some existing standard methodologies as in Figure 5.1.

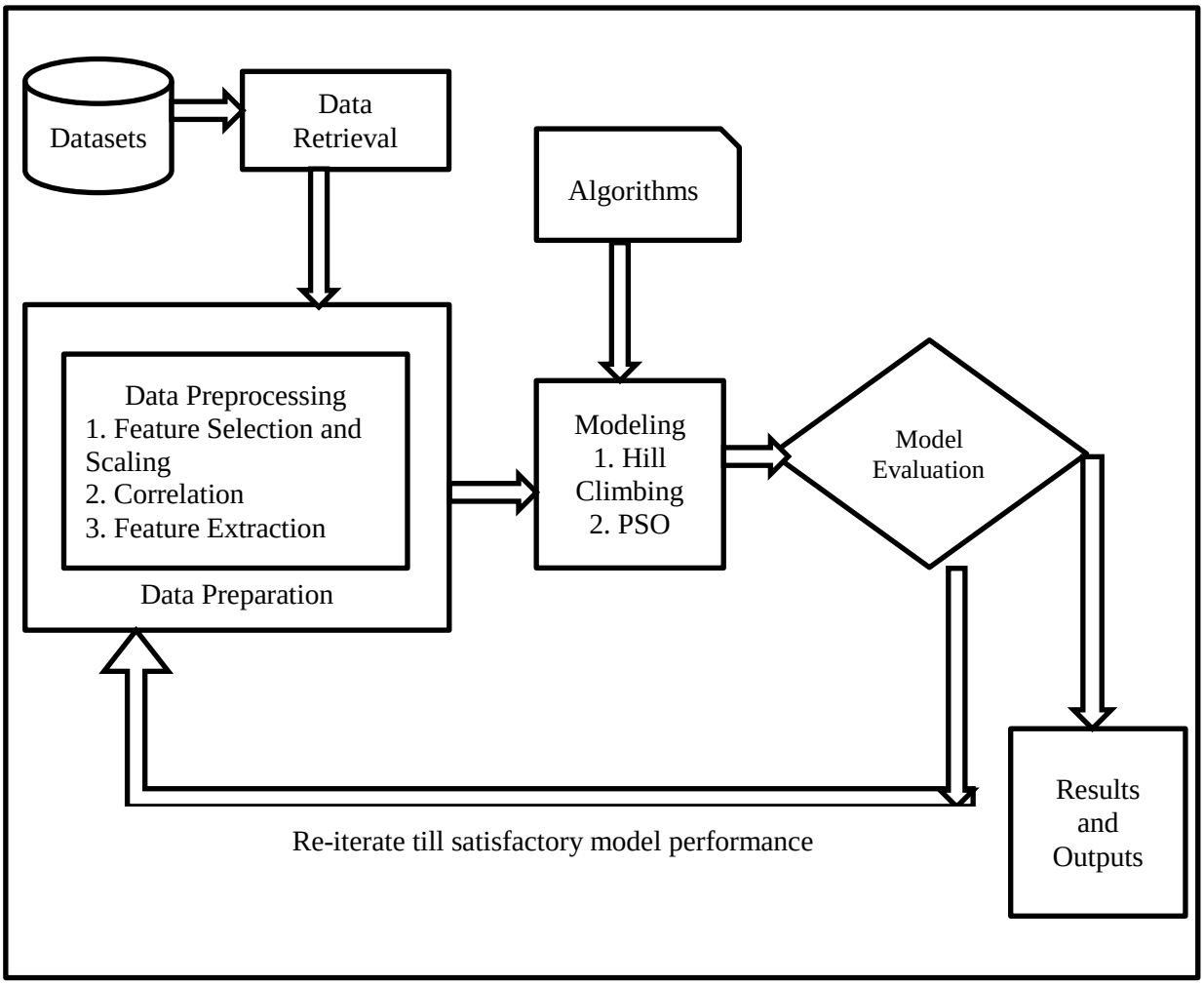

Figure 5.1 Proposed Cost Estimation Model

Now we will discuss about each step in detail and will go for results.

5.2 Step - 1 about Datasets

5.2.1 Desharnais dataset: This dataset(Figure 5.2) contains 81 software projects and easily available on PROMISE repository. Dataset can be collected from one more source named Canadian Software Houses. The dataset contains 10 attributes, in which two are dependent attributes and eight are independent. The Dependent attributes are duration and effort that is measured in person per hours. From all the 81 software projects, 4 projects were having missing values therefore we picked out those one because they may mislead the estimation process. So, after preprocessing stage collected 77 complete software projects.

The dataset contains attributes named: TeamExp, ManagerExp, YearEnd, Length, Transactions, Entities, id, Project, Effort, PointsNonAdjust, Adjustment, and PointsAdjust. Then in the first stage we selected some of the attributes on the basis of similarity matrices that measures the correlation with target variable, for this data we took effort as a target and this metrics have statistical significance. In result we got some are strongly correlated attributes and some are not. We took effort as a target variable and then select out maximum correlated attribute named Length, Transactions, Entities, PointsNonAdjust, Points Adjust for further implementation(Malathi, S. and Sridhar, S. 2011) .

	id	Project	TeamExp	ManagerExp	YearEnd	Length	Effort	Transactions	Entities	PointsNonAdjust	Adjustment	PointsAjust	Language
0	1	1	1	4	85	12	5152	253	52	305	34	302	1
1	2	2	0	0	86	4	5635	197	124	321	33	315	1
2	3	3	4	4	85	1	805	40	60	100	18	83	1
3	4	4	0	0	86	5	3829	200	119	319	30	303	1
4	5	5	0	0	86	4	2149	140	94	234	24	208	1
5	6	6	0	0	86	4	2821	97	89	186	38	192	1
6	7	7	2	1	85	9	2589	119	42	161	25	145	2
7	8	8	1	2	83	13	3913	186	52	238	25	214	1
8	9	9	3	1	85	12	7854	172	88	260	30	247	1
9	10	10	3	4	83	4	2422	78	38	116	24	103	1

Figure 5.2 Desharnais Dataset

5.2.2 Albrecht dataset: This dataset Figure 5.3 includes 24 software projects that are generated by using third generation languages like Perl and COBOL. Albrecht dataset contains 8 attributes named: Input, Output, Inquiry, File, FPAdj, RawFPcounts, AdjFP, and Effort. Then after applying correlation on attributes with respect to Effort, we select out some of the attributes like Input, Output, Inquiry, File, RawFPcounts, AdjFP and Effort is taken as target attribute. 'Work hours' is one dependent attribute that describe the dataset which shows respective effort in 1000 hours, six attributes are independent and numeric type , mostly

projects were written in COBOL , in which four were written in PL1 and rest other projects were written in management type languages. In these projects two projects have effort values near about infinity that is two times larger than third largest projects(Malathi, S. and Sridhar, S. 2011) . These extreme projects are considered as negligible impact on prediction of cost estimation but we used to keep them in progress.

	Input	Output	Inquiry	File	FPAdj	RawFPcounts	AdjFP	Effort
0	25	150	75	60	1.00	1750.00	1750	102.4
1	193	98	70	36	1.00	1902.00	1902	105.2
2	70	27	0	12	0.80	535.00	428	11.1
3	40	60	20	12	1.15	660.00	759	21.1
4	10	69	1	9	0.90	478.89	431	28.8
5	13	19	0	23	0.75	377.33	283	10.0
6	34	14	0	5	0.80	256.25	205	8.0
7	17	17	15	5	1.10	262.73	289	4.9
8	45	64	14	16	0.95	715.79	680	12.9
9	40	60	20	15	1.15	690.43	794	19.0

Figure 5.3 Albrecht Dataset

5.2.3 ISBSG dataset: This dataset has repository named ISBSG Repository (Release 10) (Figure 5.4) that contains more than 4000 projects collected from different worldwide software development companies. All the projects of ISBSG repository, contains attributes of numerical and categorical types. In all the projects many have missing values, only some of the 575 projects can say with quality rating “A” are taken: total 9 attributes were selected as useful from them and 8 from them have numerical features and 1 has categorical feature. The selected attributes are given as: Adjusted Function Points, Input Functions, Output Functions, EQC, Count of Files, Interface Functions, ADD Functions, Changed Functions, and categorical attribute Resource Level.

To identify best combination of attributes, all the different attributes are transferred to the coming order of process. At the first stage of formulation, we identified 6 different

correlation similarity matrix(Rashed Hama, A. *et al.* 2022) . I applied **OrdinalEncoder()** like Figure 5.5 to convert text into numerical format.

index	text	edge	intent
0	champion product approv stock split champion product inc board director approv two for stock split common share for sharehold record april compani board vote recommend sharehold annual meet april increas author capit stock mln mln share reuter	champion product approv stock split champion product inc board director approv two for stock split common share for sharehold record april compani board vote recommend sharehold annual meet april increas author capit stock mln mln share reuter	earn
1	comput termin system cpml complet sale comput termin system inc complet sale share common stock and warrant acquir addit mln share sedio lugano switzerland for dlr compani warrant exercis for year purchas price dlr per share comput termin sedio bui addit share and increas total hold pct comput termin outstand common stock circumst involv chang control compani compani condit occur warrant exercis price equal pct common stock market price time not exce dlr per share comput termin sold technolgi right dot matrix impact technolog includ futur improv woodco inc houston tex for dlr continu exclus worldwid license technolog for woodco compani move part reorgan plan and pai current oper cost and ensur product deliveri comput termin make comput gener label form tag and ticket printer and termin reuter	comput termin system cpml complet sale comput termin system inc complet sale share common stock and warrant acquir addit mln share sedio lugano switzerland for dlr compani warrant exercis for year purchas price dlr per share comput termin sedio bui addit share and increas total hold pct comput termin outstand common stock circumst involv chang control compani compani condit occur warrant exercis price equal pct common stock market price time not exce dlr per share comput termin sold technolgi right dot matrix impact technolog includ futur improv woodco inc houston tex for dlr continu exclus worldwid license technolog for woodco compani move part reorgan plan and pai current oper cost and ensur product deliveri comput termin make comput gener label form tag and ticket printer and termin reuter	acq
2	cobanco inc cbco year net shr ct dlr net asset mln mln deposit mln mln loan mln mln note qtr not year includ extraordinari gain tax carri forward dlr ct per shr reuter	cobanco inc cbco year net shr ct dlr net asset mln mln deposit mln mln loan mln mln note qtr not year includ extraordinari gain tax carri forward dlr ct per shr reuter	earn
3	intern inc qtr jan oper shr loss two ct profit ct oper shr profit profit rev mln mln avg shr mln mln six mth oper shr profit nil profit ct oper net profit profit rev mln mln avg shr mln mln note per shr calcul payment prefer dividend result exclud credit ct and ct for qtr and six mth six ct and ct for prior period oper loss carryforward reuter	intern inc qtr jan oper shr loss two ct profit ct oper shr profit profit rev mln mln avg shr mln mln six mth oper shr profit nil profit ct oper net profit profit rev mln mln avg shr mln mln note per shr calcul payment prefer dividend result exclud credit ct and ct for qtr and six mth six ct and ct for prior period oper loss carryforward reuter	earn
4	brown forman inc bfd qtr net shr dlr ct net mln mln rev mln mln mth shr dlr dlr net mln mln rev billion mln reuter	brown forman inc bfd qtr net shr dlr ct net mln mln rev mln mln mth shr dlr dlr net mln mln rev billion mln reuter	earn

Figure 5.4 ISBSG Dataset

After applying ordinal encoder datasets were in this format given in Figure. 1.5

index	text	edge	intent
0	883.0	883.0	2.0
1	1098.0	1098.0	0.0
2	1023.0	1023.0	2.0
3	2365.0	2365.0	2.0
4	687.0	687.0	2.0

Figure 5.5 Encoded ISBSG Dataset

5.2.4 NASA93 dataset: The NASA93 dataset contains 93 software projects and comprised all 93 projects. In which 17 are independent variables and from which 15 are having categorical attributes(Figure 5.6). Cost estimation value for the given dataset is calculated based on methodology used(Zhou, G., Etemadi, A. and Mardon, A. 2022). The datasets were collected from Promise repository, chosen to validate the model for research(Tedeschi, S. *et al.* 2018) . The datasets are easily available on various open source platform.

	rely	data	cplx	time	stor	virt	turn	acap	aexp	pcap	vexp	lexp	modp	tool	sced	equivphyskloc	act_effort
0	h	l	h	n	n	l	l	n	n	n	n	h	h	n	l	25.9	117.6
1	h	l	h	n	n	l	l	n	n	n	n	h	h	n	l	24.6	117.6
2	h	l	h	n	n	l	l	n	n	n	n	h	h	n	l	7.7	31.2
3	h	l	h	n	n	l	l	n	n	n	n	h	h	n	l	8.2	36.0
4	h	l	h	n	n	l	l	n	n	n	n	h	h	n	l	9.7	25.2
5	h	l	h	n	n	l	l	n	n	n	n	h	h	n	l	2.2	8.4
6	h	l	h	n	n	l	l	n	n	n	n	h	h	n	l	3.5	10.8
7	h	l	h	n	n	l	l	n	n	n	n	h	h	n	l	66.6	352.8
8	h	l	h	xh	xh	l	h	h	h	h	n	h	h	h	n	7.5	72.0
9	n	l	h	n	n	l	l	h	vh	vh	n	h	n	n	n	20.0	72.0

Figure 5.6 NASA93 Dataset

On this dataset, we also applied **OrdinalEncoder()** to convert text into numerical format (Figure 5.7).

And after applying formula dataset converted like this in numerical format.

	rely	data	cplx	time	stor	virt	turn	acap	aexp	pcap	vexp	lexp	modp	tool	sced	equivphyskloc	act_effort
0	0.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	2.0	1.0	2.0	0.0	0.0	2.0	1.0	31.0	22.0
1	0.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	2.0	1.0	2.0	0.0	0.0	2.0	1.0	30.0	22.0
2	0.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	2.0	1.0	2.0	0.0	0.0	2.0	1.0	10.0	6.0
3	0.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	2.0	1.0	2.0	0.0	0.0	2.0	1.0	12.0	7.0
4	0.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	2.0	1.0	2.0	0.0	0.0	2.0	1.0	13.0	5.0
5	0.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	2.0	1.0	2.0	0.0	0.0	2.0	1.0	1.0	0.0
6	0.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	2.0	1.0	2.0	0.0	0.0	2.0	1.0	3.0	1.0
7	0.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	2.0	1.0	2.0	0.0	0.0	2.0	1.0	48.0	37.0
8	0.0	1.0	0.0	3.0	3.0	1.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	2.0	9.0	15.0
9	2.0	1.0	0.0	1.0	1.0	1.0	1.0	0.0	3.0	2.0	2.0	0.0	2.0	2.0	2.0	27.0	15.0

Figure 5.7 Encoded NASA93 dataset

5.3 Step – 2 Data Retrieval

Data retrieval is generated from data collection and data extraction. When we download the Dataset from any sites or open source, it comes in all forms including structured, unstructured missing or noisy data. So we need to do some special decoding or encoding to get data in normal forms. For example we can see the ISBSG dataset in 5.2.3 section and NASA93 dataset in 5.2.4 section. Here we applied some ordinal encoding to get proper data in normal form.

On the other hand , if some data are already present in structured or unstructured forms , then some of them needs additional approval to extract from site either they are public or private accessible datasets.

5.4 Step - 3 Data Preparation and Preprocessing

We followed some steps of preprocessing to make datasets in a concise and usable way.

5.4.1 Feature Selection

In this process, we identify and find out most relevant features subsets from all the datasets to the use as an input for the model. The main motive of feature selection is to improve data quality and remove redundant and irrelevant features from the datasets. Because these redundancy might create noise into the model.

By reducing the dimensionality, we can improve the accuracy and efficiency of the model. We will select only important features, the model could be focus on only those key factors which have high impact on the results and ignore useless and redundant characters that only create noise in the data. By doing this, training data gives faster results, accuracy is improved, error gets reduced, and model is fitted perfectly without over fitting.

Feature selection is introduced by many types; one of the types is Correlation.

5.4.2 Correlation: Correlation gives strength relationship between features and target variables. These relationships may be positive, negative and neutral. Positive relationship shows that variables are highly correlated in same direction, negative relationship shows variables are highly correlated in opposite direction and neutral shows no correlation at all.

In this thesis , first select target variable and then we calculate correlation between all variables and then select out most correlated variables in positive direction with respect to target variable. Target variable is Y- factor and maximum correlated variable is X- factor for the further implementation.

We can see the example of preprocessing on one of the dataset like nasa93 that is Figure 5.8 and Figure 5.9.

Original dataset

	recordnumber	projectname	cat2	forg	center	year	mode	rely	data	cplx	...	acap	aexp	pcap	vexp	lexp	modp	tool	sce
0	1	de	avionicsmonitoring	g	2	1979	semidetached	h	l	h	...	n	n	n	n	h	h	n	
1	2	de	avionicsmonitoring	g	2	1979	semidetached	h	l	h	...	n	n	n	n	h	h	n	
2	3	de	avionicsmonitoring	g	2	1979	semidetached	h	l	h	...	n	n	n	n	h	h	n	
3	4	de	avionicsmonitoring	g	2	1979	semidetached	h	l	h	...	n	n	n	n	h	h	n	
4	5	de	avionicsmonitoring	g	2	1979	semidetached	h	l	h	...	n	n	n	n	h	h	n	
5	6	de	avionicsmonitoring	g	2	1979	semidetached	h	l	h	...	n	n	n	n	h	h	n	
6	7	de	avionicsmonitoring	g	2	1979	semidetached	h	l	h	...	n	n	n	n	h	h	n	
7	8	erb	avionicsmonitoring	g	2	1982	semidetached	h	l	h	...	n	n	n	n	h	h	n	
8	9	gal	missionplanning	g	1	1980	semidetached	h	l	h	...	h	h	h	n	h	h	h	
9	10	gal	missionplanning	g	1	1980	semidetached	n	l	h	...	h	vh	vh	n	h	n	n	

Figure 5.8 Dataset without Pre-processing

After preprocessing dataset

	rely	data	cplx	time	stor	virt	turn	acap	aexp	pcap	vexp	lexp	modp	tool	sced	equivphyskloc	act_effort
0	h	l	h	n	n	l	l	n	n	n	n	h	h	n	l	25.9	117.6
1	h	l	h	n	n	l	l	n	n	n	n	h	h	n	l	24.6	117.6
2	h	l	h	n	n	l	l	n	n	n	n	h	h	n	l	7.7	31.2
3	h	l	h	n	n	l	l	n	n	n	n	h	h	n	l	8.2	36.0
4	h	l	h	n	n	l	l	n	n	n	n	h	h	n	l	9.7	25.2
5	h	l	h	n	n	l	l	n	n	n	n	h	h	n	l	2.2	8.4
6	h	l	h	n	n	l	l	n	n	n	n	h	h	n	l	3.5	10.8
7	h	l	h	n	n	l	l	n	n	n	n	h	h	n	l	66.6	352.8
8	h	l	h	xh	xh	l	h	h	h	h	n	h	h	h	n	7.5	72.0
9	n	l	h	n	n	l	l	h	vh	vh	n	h	n	n	n	20.0	72.0

Figure 5.9 Dataset after Pre-Processing

5.4.3 Feature Extraction: The main role of feature extraction is to synchronize the data that may not affect the relevant information. It is about extracting features from original dataset to create new features. By applying these techniques, some numbers of features are also reduced from original feature set that causes better model complexity and over fitting. It will reduce error and increase computational efficiency. The process of transforming raw data can come into this category.

There are so many types of feature extraction techniques in which one is PCA (Principal component analysis).

5.5 Step – 4 Modeling

5.5.1 Hill Climbing Experimental Setup

Before applying the algorithm on the datasets, first we calculated the correlation between features of datasets and then we select out the feature to be taken for further process or not. After this the algorithm Hill Climbing applied on datasets which we are getting from correlation such that the correlation matrix (Figure 5.10) given below for the desharanais datasets. In this dataset we took Effort as a target variable and with respect to target variable we will select other features. In the Fig. the yellow color represent correlation coefficient is one and green color is showing higher correlation coefficient than other blue and sky blue color values.

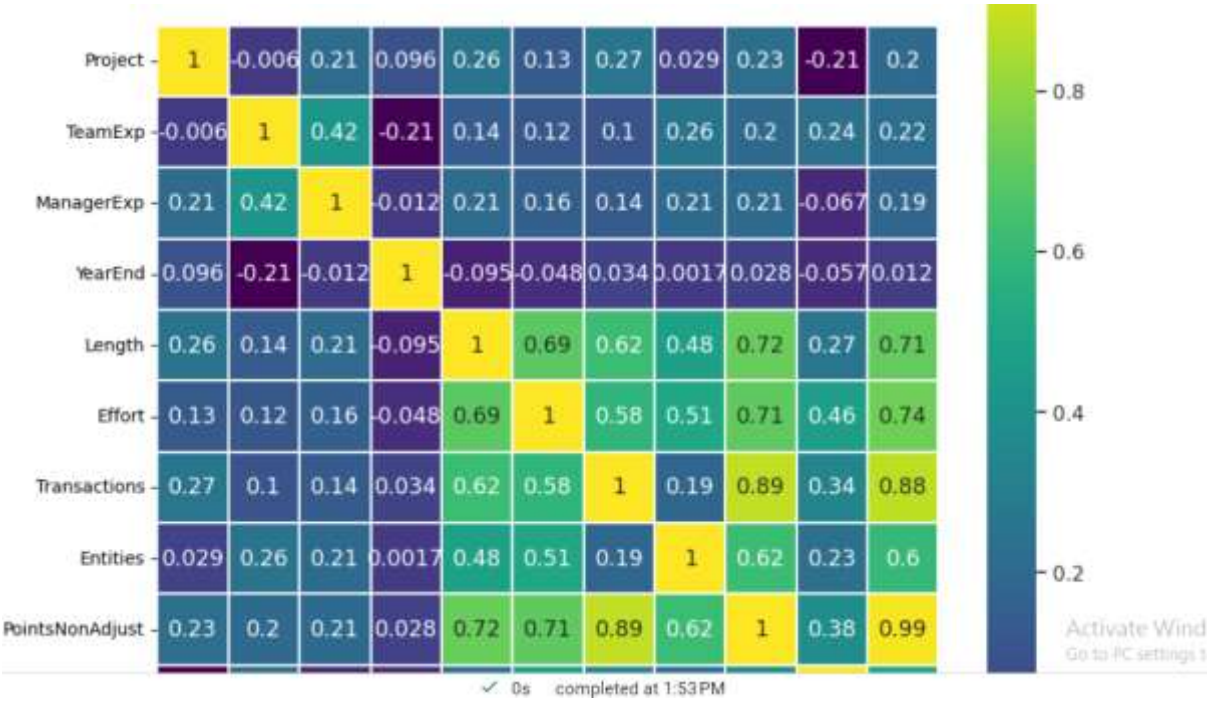

Figure 5.10 Correlation matrix

In this fig we can see that 'Length', 'Transactions', 'Entities', 'PointsNonAdjust', 'PointsAjust' have maximum correlation with respect to effort , that's why we will take these variable opposite to target variable for further calculations. We also normalize the target using **MinMaxScaler()** if needed.

We did same for the all datasets nasa93, ISBSG and Albrecht.

So now these are the matrices calculated for all the datasets given in below Figure 5.11.

Code output for Desharnais Dataset

```
Mean Magnitude of Relative Error: 0.14895899541469168
Median Magnitude of Relative Error: 0.11512537365963235
R squared :  99.98668011976115
Mean square error:  0.00036767437499344907
Root mean square error:  0.19174837026516003
Mean magnitude of percentage error : 7.190336565401914
```

Figure 5.11 Code Output

5.5.2 Particle Swarm Optimization Experimental Setup

Before apply the algorithm we have to first preprocess the data to get the proper and accurate results. We applied correlation to get maximum correlation features and then applied the PSO algorithm.

After correlation (Figure 5.12) we get this result and then we will find out maximum correlated variables with respect to target variable. The target variable is act_effort and other variables are rely', 'data', 'cplx', 'time', 'virt', 'turn', 'acap', 'aexp', 'pcap', 'vexp', 'modp', 'tool', 'sced', 'equivphyskloc'. We took the example of NASA93 Dataset.

	rely	data	cplx	time	stor	virt	turn	acap	aexp	pcap	vexp	lexp	modp
rely	1.000000	-0.146089	0.060362	0.351873	0.258990	0.340290	0.119315	-0.203907	-0.189960	-0.141314	-0.086655	-0.078659	0.309948
data	-0.146089	1.000000	-0.056043	-0.262069	-0.128951	-0.166059	-0.099424	0.368133	-0.083468	0.138749	-0.090828	0.057205	0.483661
cplx	0.060362	-0.056043	1.000000	0.192573	0.203319	-0.044516	0.238752	-0.210561	-0.020683	-0.008092	-0.339143	0.107158	-0.119984
time	0.351873	-0.262069	0.192573	1.000000	0.739577	0.119730	-0.228141	0.125143	-0.089825	-0.112993	-0.103215	-0.126674	0.041985
stor	0.258990	-0.128951	0.203319	0.739577	1.000000	-0.022821	0.029780	0.061174	-0.191286	-0.104578	-0.217339	-0.073956	0.094930
virt	0.340290	-0.166059	-0.044516	0.119730	-0.022821	1.000000	0.326939	0.012019	-0.107804	-0.118346	0.062601	0.045628	-0.152005
turn	0.119315	-0.099424	0.238752	-0.228141	0.029780	0.326939	1.000000	-0.231538	0.063717	-0.062744	-0.209591	0.117474	-0.298778
acap	-0.203907	0.368133	-0.210561	0.125143	0.061174	0.012019	-0.231538	1.000000	0.180393	0.439825	0.250747	0.216065	0.161230
aexp	-0.189960	-0.083468	-0.020683	-0.089825	-0.191286	-0.107804	0.063717	0.180393	1.000000	0.559101	0.329040	0.082674	-0.151988
pcap	-0.141314	0.138749	-0.008092	-0.112993	-0.104578	-0.118346	-0.062744	0.439825	0.559101	1.000000	0.259302	0.206884	-0.103270
vexp	-0.086655	-0.090828	-0.339143	-0.103215	-0.217339	0.062601	-0.209591	0.250747	0.329040	0.259302	1.000000	0.329701	0.112023
lexp	-0.078659	0.057205	0.107158	-0.126674	-0.073956	0.045628	0.117474	0.216065	0.082674	0.206884	0.329701	1.000000	0.081473
modp	0.309948	0.483661	-0.119984	0.041985	0.094930	-0.152005	-0.298778	0.161230	-0.151988	-0.103270	0.112023	0.081473	1.000000
tool	-0.099621	0.242717	-0.210878	-0.018876	-0.147524	-0.082581	-0.263057	0.170274	0.058144	-0.009935	0.023943	-0.047694	0.342505
sced	-0.073003	0.084601	-0.345378	-0.493431	-0.252271	-0.085242	0.056005	0.067511	0.100087	0.064731	0.348734	0.369549	0.193356

Figure 5.12 Correlation of NASA93 Dataset

After getting feasible features then applied the algorithm and getting this results given below in Figure 5.13.

Result for nasa93 dataset

```
Iteration: 99
Best (PSO): 9.007771868112338
Iteration: 100
Best (PSO): 9.007771868112338
Run Time --- 3.6783924102783203 seconds ---
RMSE: 9.092446331426604
Mean Magnitude of Relative Error: 6.765188210515973
Median Magnitude of Relative Error: 5.227912897561069
R squared :  0.8300180617002559
Mean square error:  82.67258028987312
Root mean square error:  9.092446331426604
Mean magnitude of percentage error : 299898585051938.56
Feature Size: 10
```

Figure 5.13 Results for NASA93 Dataset

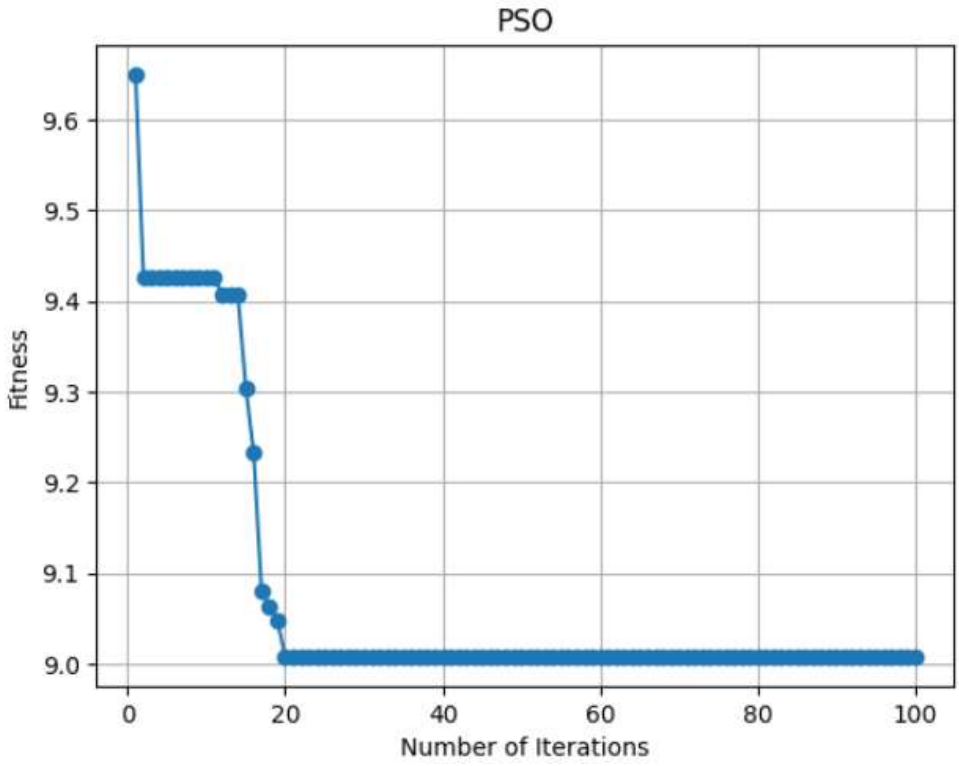

Figure 5.14 Relationship between number of Iterations and Fitness for NASA93 Dataset using PSO Algorithm

The graph in Figure 5.14 showed the relationship between number of iterations and fitness of features with respect to target variables.

5.6 Step – 5 Model Evaluations

5.6.1 Measurements available for evaluating Software Cost Estimation techniques

Various metrics are available for performance measurement for different cost estimation techniques (Kumari, S. and Pushkar, S. 2018) .

- **Magnitude of Relative Error (MRE):** It is the most famous measurement parameter for cost estimation techniques. It is evaluated as the difference between the actual effort and the estimated effort. It can be measure by given formula in equation 5.1:

$$MRE = |(act_i - est_i)| \, / act_i \qquad (5.1)$$

Where, act is actual effort

est is estimated effort

- **Mean Absolute Error (MAE):** This parameter is calculated by dividing the absolute sum of the given error by the number of predictions for that error. Mean absolute error is also called as Mean Magnitude of Relative error. The difference of the actual effort and predicted effort is termed as given error, which we can calculate by given formula in equation 5.2 as:

$$MMRE = \frac{1}{N}\sum_i^N MREi \qquad (5.2)$$

N is Number of measured errors

- **Root Mean Squared Error:** It is calculated by doing the square root of the sum of the square of the error divided by the number of predictions. The error is calculated as the difference between actual effort and predicted effort. The formula is given in equation 5.3:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(act_i - est_i)^2} \qquad (5.3)$$

- **Relative Absolute Error (RAE):** We can calculate this parameter by dividing the summation of the absolute error by the summation of absolute relative error. Here, the difference between actual effort and predicted effort is termed as absolute error. The difference between the actual effort and the mean of estimated effort is known as absolute relative error. The formula is given in equation 5.4 as:

$$RAE = \frac{\sum_1^N |(act_i - \widehat{est_i})|}{\sum_1^N |(act_i - \overline{est})|} \qquad (5.4)$$

- **Mean Magnitude of Percentage Error (MMPR):** MMPR is the difference between actual effort value and estimated effort value and compared with actual value. Results are

expressed in percentage. In short we can say that relative error multiplied by 100. Formula is given as below in equation 5.5:

$$MMPR = [|(act_i - est_i)| / act_i] \times 100 \quad (5.5)$$

- **Correlation Coefficient (R^2):** It is used to measure the relation between actual and estimated effort.

5.7 Step – 6 Results and Outputs

The details about the results of all the metrics given below in tabular form and also showed the comparison with existing methods KNN, Linear Regression and Random forests.

5.7.1 Hill Climbing Results

5.7.1.1 Results of Mean Squared Error(MSE)

Dataset	Random Forest	Linear Regression	KNN	Hill Climbing
	MSE	**MSE**	**MSE**	**MSE**
DESHARNAIS	1.65	1.43	3.02	**0.000252**
ALBRECHT	0.82	2.72	0.53	3.56
ISBSG (R8)	5.30	6.43	5.23	**1.83**
NASA93	2.86	3.68	4.15	**0.00075**

Table 5.1

MSE measures error in statistical methods by using the average squared difference between observed and predicted values. It measures average of the square of the error. In above Table 5.1, we can see that the error is reduced for DESHARANAIS, NASA93, and ISBSG dataset in comparison to other standard techniques Random Forest, Linear Regression and KNN. The reduced error is 0.02 % for Desharnais, 0.07% for NASA93 dataset and 1.83 for ISBSG dataset.

5.7.1.2 Results of Root Mean Square error (RMSE)

Dataset	Random Forest	Linear Regression	KNN	Hill Climbing
	RMSE	**RMSE**	**RMSE**	**RMSE**
DESHARNAIS	12.85	11.95	17.37	**0.16**
ALBRECHT	9.05	16.51	7.28	**0.059**
ISBSG (R8)	23.02	25.36	22.87	**13.52**
NASA93	16.90	19.19	20.38	**0.274**

Table 5.2

Root mean square error is calculated by doing the square root of the sum of the square of the error divided by the number of predictions. In the above Table 5.2 the result for Hill Climbing method is better than other standard regression methods. The RMSE vale for Desharnais is 0.16, for Albrecht is 0.059, for ISBSG is 13.52 and for NASA93 is 0.274. In the result we can clearly see that error is reduced than other algorithms.

5.7.1.3 Results of Mean Absolute Error or Mean Magnitude of Relative Error (MMRE)

Dataset	Random Forest	Linear Regression	KNN	Hill Climbing
	MMRE	**MMRE**	**MMRE**	**MMRE**
DESHARNAIS	10.40	8.93	12.70	**0.15**
ALBRECHT	6.99	13.07	5.28	**0.041**
ISBSG (R8)	15.16	16.68	15.80	**9.04**
NASA93	12.01	14.10	13.77	**0.18**

Table 5.3

Mean Absolute Error (MAE): This parameter is calculated by dividing the absolute sum of the error by the number of predictions. The error is difference between actual and predicted effort. In the Table 5.3, we can see that Hill climbing more reduced the error in comparison to other standard techniques. Like for Desharnais dataset the error value is 0.15 and for NASA93 dataset the value is 0.18 while other methods give very high error values.

5.7.1.4 Results of Mean Magnitude of percentage Error (MMPR)

Dataset	Random Forest	Linear Regression	KNN	Hill Climbing
	MMPR	**MMPR**	**MMPR**	**MMPR**
DESHARNAIS	578.16	259.28	615.60	**0.18**
ALBRECHT	75.92	130.99	54.08	**0.79**
ISBSG (R8)	2.48	3.47	2.60	**1.73**
NASA93	38.18	47.74	41.52	**1.013**

Table 5.4

MMPR is the difference between estimated value and actual value and compared with actual value. Results are expressed in percentage. In short we can say that relative error multiplied by 100. In the table 5.4 we can see that results for all the datasets using Hill Climbing are giving appropriate and reduced error than others. For Desharnais datasets value is 79% which is lesser than other methods.

5.7.1.5 Results of Median Magnitude of Relative Error (MdMRE)

MdMRE is the ANOVA that is analysis of variance. In the Table 5.5, we can see that using Hill climbing the results for Albrecht, NASA93 and ISBSG are better in comparison with other techniques. The error is reduced on these three datasets using Hill Climbing algorithm.

Dataset	Random Forest	Linear Regression	KNN	Hill Climbing
	MdMRE	MdMRE	MdMRE	MdMRE
DESHARNAIS	0.08	0.08	0.08	3.28
ALBRECHT	0.051	0.051	0.051	**0.050**
ISBSG (R8)	5.97	5.97	5.97	**5.96**
NASA93	0.097	0.097	0.097	**0.096**

Table 5.5

5.7.1.6 Results of R Squared(R^2)

Dataset	Random Forest	Linear Regression	KNN	Hill Climbing
	R^2	R^2	R^2	R^2
DESHARNAIS	40.19	48.30	-9.31	**99.99**
ALBRECHT	32.62	-123.93	56.46	**99.99**
ISBSG (R8)	17.46	-0.204	18.54	**71.55**
NASA93	65.50	55.55	49.88	**99.99**

Table 5.6

The R squared is the Pearson coefficient. It is correlation coefficient that gives the strength between actual and estimated effort. It is used to measure the relation between actual and estimated effort. Higher value shows the strong relation between efforts. In the Table 5.6 Hill Climbing gives the highest correlation among all the standard methods for all the datasets. NASA93, Desharnais and Albrecht datasets are showing very good relation that is 99.99% and for ISBSG dataset the coefficient value is 71.55 which are also higher than other methods given in Table 5.6.

5.7.2 Particle Swarm Optimization Results

5.7.2.1 Result for Mean Square Error (MSE)

Dataset	Particle swarm optimization
	MSE
DESHARNAIS	2.23
ALBRECHT	3.16
NASA93	12.90
Dataset of Turkish Software Industry	-

Table 5.7

In the Table 5.7 we can see that the Desharnais dataset is showing lesser error than other dataset. It means this dataset is more flexible and reliable than Albrecht and NASA93 dataset.

5.7.2.2 Result for Root Mean Square Error (RMSE)

Dataset	Particle swarm optimization
	RMSE
DESHARNAIS	1.49
ALBRECHT	1.78
NASA93	9.10
Dataset of Turkish Software Industry	-

Table 5.8

Root Mean square error is very less for Desharanais Dataset in comparison to other dataset and gives a very high accuracy. So that, this dataset will give best result using Particle swarm optimization algorithm.

5.7.2.3 Results of Mean Absolute Error or Mean Magnitude of Relative Error (MMRE)

Dataset	Particle swarm optimization
	MMRE
DESHARNAIS	1.00
ALBRECHT	1.33
NASA93	6.78
Dataset of Turkish Software Industry	34.19

Table 5.9

In the Table 5.9 again we can see that Desharnais, Albrecht, NASA93 give better performance and their error rate is very low in comparison to Dataset of Turkish Software industry. The result is showed in percentile. Desharnais has 1.0 error value, Albrecht has 1.33 error values, NASA93 has 6.78 error values and apart from this Dataset of Turkish Software Industry has very high error rate that is 34.19.

5.7.2.4 Results of Mean Magnitude of percentage Error (MMPR)

Dataset	Particle swarm optimization
	MMPR
DESHARNAIS	2.79
ALBRECHT	1.96
NASA93	0.51
Dataset of Turkish Software Industry	-

Table 5.10

This is referred as relative error multiplied by 100. For some applications, percentage error is expressed as positive value. In the Table 5.10 the result for NASA93 metrics is good for this metrics. This result is different from other above results.

5.7.2.5 Results of R Squared(R^2)

Dataset	Particle swarm optimization
	R^2
DESHARNAIS	1.00
ALBRECHT	1.00
NASA93	0.71
Dataset of Turkish Software Industry	-

Table 5.11

The Table 5.11 is showing the result of correlation coefficient. Higher the values means estimated and predicted effort are highly correlated. In the given table Desharnais and Albrecht dataset has high correlation between actual and predicted efforts.

There is represented so many results using Hill climbing and Particle Swarm Optimization algorithm. These algorithms are applied on different datasets and compared the results through different matrices.

Chapter 6

Conclusion and Future Scope

In this thesis, we represented an overview and introduction of software cost estimation methods. In the literature, various methods are available based on algorithmic methods, non-algorithmic methods, and learning-oriented methods. We did tabular comparison of these methods based on their strengths and weaknesses. We surveyed existing published papers and provided a comparative analysis of them. From the literature, we found that learning-oriented methods generally outperform other cost estimation methods. However, none of the methods is better than the other. All methods have their pros and cons. So, we must choose them according to need and research.

In the part of implementation we can see the details of each result and discussed about the results. So as in many of the cases Hill climbing gave the better result as compared to other standard algorithms on the basis of given metrics MSE, RMSE, MMRE, MdMRE, MMPE and R squared. For the Particle Swarm Optimization, we can see this method was giving some better results on some of the datasets and compared to other. The datasets used Desharnais, NASA93, ISBSG and Albrecht have their own features and characteristics. Like some has different in types, category, so that after performing some types of preprocessing and encoder for textual datasets, we got the results and comparison with existing models.

For the future, we recommend the researchers to explore the further application of machine learning models to get more precise cost estimation. We have tried to cover most of the cost estimation techniques in this thesis, but there are also some remaining methods. We can suggest researcher to go in details and try to compare the results with optimization to optimization technique. For the proper management of any project, the appropriate measurement is also required. So a deep study and analysis of various software cost estimation methods are essential to achieve the target and completion of the project on time and under budget.

References

Anusha, M.E. and Mukesh, R. (2013) 'A Study On Software Cost Estimation', *International Journal of Emerging Trends & Technology in Computer Science*, 2013.

Arslan, F. (2019) 'A Review of Machine Learning Models for Software Cost Estimation', *Review of Computer Engineering Research*, 6(2), pp. 64–75. Available at: https://doi.org/10.18488/journal.76.2019.62.64.75.

Aunsri, N. and Rattarom, S. (2022) 'Novel eye-based features for head pose-free gaze estimation with web camera: New model and low-cost device', *Ain Shams Engineering Journal*, 13(5), p. 101731. Available at: https://doi.org/10.1016/j.asej.2022.101731.

Azzeh, M. (2013) 'Dataset Quality Assessment: An extension for analogy based effort estimation', *International Journal of Computer Science & Engineering Survey*, 4(1), pp. 1–21. Available at: https://doi.org/10.5121/ijcses.2013.4101.

El Bajta, M. *et al.* (2015) 'Software cost estimation for global software development: A systematic map and review study', *ENASE 2015 - Proceedings of the 10th International Conference on Evaluation of Novel Approaches to Software Engineering*, pp. 197–206. Available at: https://doi.org/10.5220/0005371501970206.

Bedi, R.P.S. and Singh, A. (2017) 'Software Cost Estimation using Fuzzy Logic Technique', *Indian Journal of Science and Technology*, 10(3). Available at: https://doi.org/10.17485/ijst/2017/v10i3/109997.

Bilgaiyan, S. *et al.* (2017) 'A systematic review on software cost estimation in Agile Software Development', *Journal of Engineering Science and Technology Review*, 10(4), pp. 51–64. Available at: https://doi.org/10.25103/jestr.104.08.

Bloch, M., Blumberg, S. and Laartz, J. (2012) *Delivering large-scale IT projects on time, on budget, and on value*.

Butt, S.A. *et al.* (2022) 'A software-based cost estimation technique in scrum using a developer's expertise', *Advances in Engineering Software*, 171(May). Available at: https://doi.org/10.1016/j.advengsoft.2022.103159.

Ch*, S.S. and Singh, S.P. (2020) 'A Novel Fuzzy Model for Software Cost Estimation', *International Journal of Innovative Technology and Exploring Engineering*, 9(4), pp. 111–119. Available at: https://doi.org/10.35940/ijitee.d1141.029420.

Chirra, S.M.R. and Reza, H. (2019) 'A Survey on Software Cost Estimation Techniques', *Journal of Software Engineering and Applications*, 12(06), pp. 226–248. Available at: https://doi.org/10.4236/jsea.2019.126014.

Dewi, R.S., Subriadi, A.P. and Sholiq (2017) 'A Modification Complexity Factor in Function

Points Method for Software Cost Estimation Towards Public Service Application', *Procedia Computer Science*, 124, pp. 415–422. Available at: https://doi.org/10.1016/j.procs.2017.12.172.

E, J.M.B.A.M. and E, R.M.M. (2013) 'A Study On Software Cost Estimation', 2013.

Farsi, M. *et al.* (2021) 'A digital twin architecture for effective product lifecycle cost estimation', *Procedia CIRP*, 100, pp. 506–511. Available at: https://doi.org/10.1016/j.procir.2021.05.111.

Format, F. (2021) 'The Software Cost Estimation and Cost-Efficient Fees Structure Format for Educational Institutions', 9(9), pp. 39–49.

G. Pavithra (2021) 'The Software Cost Estimation and Cost-Efficient Fees Structure Format for Educational Institutions', *International Journal of Creative Research Thoughts (IJCRT)*, 9(9), pp. 39–49.

Hallmann, M. *et al.* (2018) 'Comparison of different methods for scrap rate estimation in sampling-based tolerance-cost-optimization', *Procedia CIRP*, 75, pp. 51–56. Available at: https://doi.org/10.1016/j.procir.2018.01.005.

Heemstra, F.J. (1990) 'Software cost estimation models', *Proceedings of the Jerusalem Conference on Information Technology*, 3(2), pp. 286–297. Available at: https://doi.org/10.1016/b978-0-7506-0813-8.50035-6.

Heemstra, F.J. (1992) 'Software cost estimation', *Information and Software Technology*, 34(10), pp. 627–639. Available at: https://doi.org/10.1016/0950-5849(92)90068-Z.

Keim, Y. *et al.* (2014) 'Software cost estimation models and techniques: A survey', *International Journal of Engineering Research and Technology*, 3(2), pp. 1763–1768.

Kumari, S. and Pushkar, S. (2018) 'Cuckoo search based hybrid models for improving the accuracy of software effort estimation', *Microsystem Technologies*, 24(12), pp. 4767–4774. Available at: https://doi.org/10.1007/s00542-018-3871-9.

Langsari, K., Sarno, R. and Sholiq (2018) 'Optimizing effort parameter of COCOMO II using Particle Swarm Optimization method', *Telkomnika (Telecommunication Computing Electronics and Control)*, 16(5), pp. 2208–2216. Available at: https://doi.org/10.12928/TELKOMNIKA.v16i5.9703.

Malathi, S. and Sridhar, S. (2011) 'A Classical Fuzzy Approach for Software Effort Estimation on Machine Learning Technique', *International Journal of Computer Science Issues*, 8(6), pp. 249–253.

R.K., K.D. ;Gupt. (2014) 'Software Cost Estimation Techniques– A Review of Literature', *International Journal of Research and Development in Applied Science and Engineering (IJRDASE)*, 9359(4), pp. 104–108.

Rashed Hama, A. *et al.* (2022) 'Optimization model for cost estimation of decentralized wastewater treatment units', *Ain Shams Engineering Journal*, 13(6), p. 101751. Available at: https://doi.org/10.1016/j.asej.2022.101751.

Singal, P., Kumari, A.C. and Sharma, P. (2020) 'Estimation of Software Development Effort: A Differential Evolution Approach', *Procedia Computer Science*, 167(2019), pp. 2643–2652. Available at: https://doi.org/10.1016/j.procs.2020.03.343.

Singh, S.P., Singh, V.P. and Mehta, A.K. (2021) 'Differential evolution using homeostasis adaption based mutation operator and its application for software cost estimation', *Journal of King Saud University - Computer and Information Sciences*, 33(6), pp. 740–752. Available at: https://doi.org/10.1016/j.jksuci.2018.05.009.

Singh, T., Singh, R. and Mishra, K.K. (2018) 'Software cost estimation using environmental adaptation method', *Procedia Computer Science*, 143, pp. 325–332. Available at: https://doi.org/10.1016/j.procs.2018.10.403.

Slowik, A. (2011) 'Particle Swarm Optimization', *The Industrial Electronics Handbook - Five Volume Set* [Preprint], (July). Available at: https://doi.org/10.1007/978-3-319-46173-1_2.

Tedeschi, S. *et al.* (2018) 'A cost estimation approach for IoT modular architectures implementation in legacy systems', *Procedia Manufacturing*, 19, pp. 103–110. Available at: https://doi.org/10.1016/j.promfg.2018.01.015.

Zakrani, A., Hain, M. and Namir, A. (2018) 'Software development effort estimation using random forests: An empirical study and evaluation', *International Journal of Intelligent Engineering and Systems*, 11(6), pp. 300–311. Available at: https://doi.org/10.22266/IJIES2018.1231.30.

Zhou, G., Etemadi, A. and Mardon, A. (2022) 'Machine learning-based cost predictive model for better operating expenditure estimations of U.S. light rail transit projects', *Journal of Public Transportation*, 24, p. 100031. Available at: https://doi.org/10.1016/j.jpubtr.2022.100031.

Suneetha 2

ORIGINALITY REPORT

8%	4%	5%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Cuicui Lu, Hongyi Yuan, Nianen Zhang. "Nanophotonic devices based on optimization algorithms", Elsevier BV, 2023 Publication	1%
2	Submitted to University of Mumbai Student Paper	1%
3	Geeta Nagpal, Moin Uddin, Arvinder Kaur. "Analyzing software effort estimation using k means clustered regression approach", ACM SIGSOFT Software Engineering Notes, 2013 Publication	1%
4	academic.oup.com Internet Source	1%
5	www.geeksforgeeks.org Internet Source	1%
6	Abbas Heiat. "Comparison of artificial neural network and regression models for estimating software development effort", Information and Software Technology, 2002 Publication	<1%

Suneetha 1

ORIGINALITY REPORT

17%	14%	12%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	commons.und.edu Internet Source	6%
2	www.scirp.org Internet Source	2%
3	www.researchgate.net Internet Source	1%
4	journal.uad.ac.id Internet Source	1%
5	Shailendra Pratap Singh, Vibhav Prakash Singh, Ashok Kumar Mehta. "Differential Evolution Using Homeostasis Adaption Based Mutation Operator and its Application for Software Cost Estimation", Journal of King Saud University - Computer and Information Sciences, 2018 Publication	1%
6	www.semanticscholar.org Internet Source	1%
7	Submitted to University of Science and Technology, Yemen	1%