APPLICATION OF EVOLUTIONARY TECHNIQUES FOR SOFTWARE EFFORT ESTIMATION

A Dissertation submitted in the partial fulfilment for the award of Degree of

MASTER OF TECHNOLOGY

IN

SOFTWARE ENGINEERING

By

NEHA SAINI

(Roll No. 2K12/SWE/17)

Under the guidance of

DR. RUCHIKA MALHOTRA

Department of Software Engineering

Delhi Technological University, Delhi

Department of Computer Engineering

Delhi Technological University, Delhi

2012-2014

DELHI TECHNOLOGICAL UNIVERSITY

CERTIFICATE

This is to certify that the project report entitled **APPLICATION OF EVOLUTIONARY TECHNIQUES FOR SOFTWARE EFFORT ESTIMATION** is a bonafide record of work carried out by Neha Saini (2K12/SWE/17) under my guidance and supervision, during the academic session 2012-2014 in partial fulfilment of the requirement for the degree of Master of Technology in Software Engineering from Delhi Technological University, Delhi.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any Degree or Diploma.

Dr.RuchikaMalhotra

Asst. Professor

Department of Software Engineering

Delhi Technological University

Delhi

DELHI TECHNOLOGICAL UNIVERSITY

ACKNOWLEDGEMENT

With due regards, I hereby take this opportunity to acknowledge a lot of people who have supported me with their words and deeds in completion of my research work as part of this course of Master of Technology in Software Engineering.

To start with I would like to thank the almighty for being with me in each and every step of my life. Next, I thank my parents and family for their encouragement and persistent support.

I would like to express my deepest sense of gratitude and indebtedness to my guide and motivator, **Dr. Ruchika Malhotra**, Assistant Professor, Department of Software Engineering, Delhi Technological University for her valuable guidance and support in all the phases from conceptualization to final completion of the project.

I wish to convey my sincere gratitude to **Prof. Rajeev Kapoor,** Head of Department, and all the faculties and PhD. Scholars of Computer Engineering Department, Delhi Technological University who have enlightened me during my project.

I humbly extend my grateful appreciation to my friends whose moral support made this project possible.

Last but not the least, I would like to thank all the people directly and indirectly involved in successfully completion of this project.

Neha Saini

Roll No. 2K12/SWE/17

TABLE OF CONTENTS

Certificate ……………………………………………………………………………...…i

Acknowledgement ……………………………………………………………………ii

Table of contents ……………………………...…………………………………..….iv-vi

List of Tables ……………………………..…………………………………..……vii

List of Figures ………………..……………..…………………………….......…...viii-x

Abstract ……………………………………………………………...……………xi

Chapter 1: Introduction ……………………………………………...………………...1-7

1.1 Software Effort Prediction……………...…………………………………………1

1.2 Classification of software effort estimation techniques……..……………………2

1.3 Aim of Thesis ……………………………………………...………………...5

1.4 Motivation…………... ………………………………………………………6

1.5 Organization of thesis……………………………………………………...6

Chapter 2: Related Work ……………………………………………………..……8-14

2.1 Main research papers studied…………………………………………….…..8

2.2 Summary of Literature Survey………………………………………..…….13

Chapter 3: Research Methodology………………...…………………………..………15-28

3.1 FRSBM…. ……………………………………………………………… 15

3.2 GANN ………………………………………………………………….. 15

3.3 NNEP……………………………………..………………..………..…..16

3.4 GFS-GAP-Sym-R…………………………………..…………………………17

3.5 GFS-GSP-R……………………………………………………………………17

3.6 GFS-GP-R…………………………………………………………………..18

3.7 GFS-SAP-Sym-R……………………………………………...………………18

3.8 GFS-GPG-R……………………………………………………………...…18

3.9 MLP-BP-R……………………………………………………………….21

3.10 MLP-CG-R……………………………………………………..………22

3.11 Incr-RBFN-R…………………………………………….…..…………22

3.12 Ensemble-R……………………………………………………..……….23

3.13 iRProp+-R………………………………………………………………24

3.14 Linear LMS-R………………………………………………….……….25

3.15 PolQuadratic LMS-R………………………………………..……..25

3.16 M5 Rules……………………………………………………………..…26

3.17 CART-R……………………………………………………….……..26

3.18 Thrift…………………………………………………………………….27

3.19 NU_SVM_R…………………………………………………………….27

3.20 Epsilon_SVM_R……………………………………………………..27

3.21 Preprocessing Algorithm KNN –Missing values……………………………….28

Chapter 4: Research Background ……………………………………..……....29-39

4.1 Datasets Description…..……………………………………………...………29

4.1.1 Desharnais dataset……………………………………..…………29

4.1.2 Maxwell dataset………………………………...………………30

4.1.3 China dataset…………………………………………...…….32

4.1.4 NASA dataset……………….……………………………………..33

4.1.5 Heiat Heiat dataset………………………………………………………34

4.1.6 Miyazaki94 dataset………………………………………………..….35

4.2 Estimation accuracy measures…………………………………………………...36

4.3 Cross validation…………………………………………………………....…..37

4.4 Tool used for calculation……………………………………………………....37

Chapter 5: Results and discussion ……..……40-66

5.1 Analysis using Miyazaki94 dataset…………………………………….... 40

5.1.1 Network graph of analysis using Miyazaki94 dataset.............................40

5.1.2 Discussion of results with Miyazaki94 dataset..43

5.2 Analysis using NASA18 dataset……….....…………………………………...… 44

5.2.1 Network graph of analysis using NASA18 dataset..................................44

5.2.2 Discussion of results with NASA18 dataset...47

5.3 Analysis using Heiat Heiat dataset…………………………………………...… 48

5.3.1 Network graph of analysis using Heiat Heiat dataset...............................48

5.3.2 Discussion of results with Heiat Heiat dataset..51

5.4 Analysis using Desharnais dataset……………………………………………...… 53

5.4.1 Network graph of analysis using Desharnais dataset...............................53

5.4.2 Discussion of results with Desharnais dataset...55

5.5 Analysis using China dataset……………………………………...…......... 57

5.5.1 Network graph of analysis using China dataset.......................................57

5.5.2 Discussion of results with China dataset..60

5.6 Analysis using Maxwell dataset……………………………….....……...… 62

5.6.1 Network graph of analysis using Maxwell dataset...................................62

5.6.2 Discussion of results with Maxwell dataset..65

Chapter 6 Conclusion and Future Work ………………………………………..67-69

6.1 Applications...69

6.2 Future Work...69

References …………………………………………………………………..70-75

List of Tables

Table 1.1 Summary of software effort estimation techniques in literature of software effort estimation.........3

Table 1.2: Advantages and Disadvantages of different software effort estimation techniques.........4

Table 2.1 Summary of literature survey.........13

Table 4.1 Desharnais dataset description.........29

Table 4.2 Descriptive statistics of Desharnais dataset 30

Table 4.3 Descriptive statistics of Maxwell dataset 31

Table 4.4 Descriptive statistics of China dataset 32

Table 4.5 Descriptive Statistics of NASA18 dataset.........34

Table 4.6 Descriptive statistics of HeiatHeiat dataset.........34

Table 4.7 Attribute Information 35

Table 4.7 Descriptive Statistics of Miyazaki94 Dataset.........36

Table 5.1 Results with Miyazaki94 dataset.........42

Table 5.2 Results with NASA18 dataset.........46

Table 5.3 Results with Heiat Heiat dataset.........50

Table 5.4 Result with Desharnais dataset.........55

Table 5.5 Result with China dataset.........59

Table 5.6 Result with Maxwell dataset.........64

Table 6.1 Table describing best, good and worst methods for different datasets.........68

LIST OF FIGURES

Figure 2.1 Decision rule generation process……………………………………………………12

Figure 3.1 Principal structure of FRSBM..15

Figure 3.2 Principal structure of GANN System..16

Figure 3.3(a) Framework 1 for GFS-GPG-R..19

Figure 3.3(b) Framework after results of k-means clustering for GFS-GPG-R.......................20

Figure 3.4 An example of Feedforward Multilayer Perceptron Backpropagation algorithm...22

Figure 3.5 Network Graph of Incremental RBF……………………………………...23

Figure 3.6 Ensembling of two neural networks……………………………………...24

Figure 3.7 Linear LMS-R Method…………………………………………………...…25

Figure 3.8 PolQuadratic LMS-R..26

Figure 3.9 Support Vector Machine..27

Figure 4.1: Screen shot of user Interface 1 of KEEL Tool..39

Figure 4.2: Screenshot of user interface 2 of KEEL Tool...39

Figure 5.1(a) Miyazaki94 Network Graph 1...40

Figure 5.1(b) Miyazaki94 Network Graph 2...41

Figure 5.1(c) Miyazaki94 Network Graph 3...41

Figure 5.2 Line chart MMRE Vs Methods..43

Figure 5.3 Line chart PRED values Vs Methods..43

Figure 5.4(a) NASA18 Network Graph 1...44

Figure 5.4(b) NASA18 Network Graph 2...44

Figure 5.4(c) NASA18 Network Graph 3..45

Figure 5.4(d) NASA18 Network Graph 4..45

Figure 5.5 Line chart MMRE Vs Methods..47

Figure 5.6 Line chart PRED Values Vs Methods..47

Figure 5.7(a) Heiat Heiat Network Graph 1..48

Figure 5.7(b) Heiat Heiat Network Graph 2..49

Figure 5.7(c) Heiat Heiat Network Graph 3..49

Figure 5.7(d) Heiat Heiat Network Graph 4..50

Figure 5.8 Line Chart MMRE Vs Methods..52

Figure 5.9 Line chart PRED Values Vs Methods..52

Figure 5.10(a) Desharnais Network Graph 1..53

Figure 5.10(b) Desharnais Network Graph 2..54

Figure 5.10(c) Desharnais Network Graph 3..54

Figure 5.11 Line chart MMRE Vs Methods..56

Figure 5.12 Line Chart PRED Values Vs Methods..56

Figure 5.13(a) China Network Graph 1..57

Figure 5.13(b) China Network Graph 2..58

Figure 5.13(c) China Network Graph 3..58

Figure 5.13(d) China Network Graph 4..59

Figure 5.14 Line chart MMRE Vs Methods..61

Figure 5.15 Line chart PRED Values Vs Methods..61

Figure 5.16(a) Maxwell Network Graph 1..62

Figure 5.16(b) Maxwell Network Graph 2..63

Figure 5.16(c) Maxwell Network Graph 3..63

Figure 5.16(d) Maxwell Network Graph 4..64

Figure 5.17 Line chart MMRE Vs Methods..65

Figure 5.18 Line Chart PRED values Vs Methods...66

Abstract

Software effort estimation is a very difficult task carried out by software project managers as very little information is available in the early phases of software development. The information that we are collecting about various attributes of software needs to be subjective which otherwise can lead to uncertainity. Inaccurate software effort estimation can be disastrous. Both underestimation and over estimation can lead to schedule overruns and incorrect estimation of budget for software development. Software effort estimation is a very crucial activity for project control, quality control and success of any software project. Software effort estimation fall under the categories of expert judgement, algorithmic and machine learning techniques. We have tried to analyse the performance of evolutionary techniques for software effort estimation. For this purpose various datasets with different properties have been collected. After that various evolutionary algorithms like FRSBM-R, GFS-SAP-Sym-R, GFS-GAP-Sym-R, NNEP-R, GANN-R, GFS-GP-R, GFS-GSP-R, GFS-RB-MF-R, CART-R, Linear_LMS-R, NU_SVR-R, EPSILON_SVR-R etc have been used. Performance is measured in terms of various accuracy measures like MMRE, MRE, PRED(25), PRED(50) and PRED(75).

Results of our research have shown that evolutionary algorithms give more accurate results for software effort estimation as compared to traditional methods of software effort estimation. Moreover the comparison of different evolutionary algorithms is done to find which evolutionary learning algorithm is better for which situation.

Keywords : Machine learning,evolutionary algorithms,software effort estimation

Abstract

Software effort estimation is a very difficult task being carried out by software developers as very little information is available in the early phases of software development.The information that we are collecting about various attributes of software needs to be subjective which otherwise can lead to uncertainity. Inaccurate software effort estimation can be disastrous. Both underestimation and over estimation can lead to schedule overruns and incorrect estimation of budget for software development.Software effort estimation is very crucial activity for project control,quality control and success of any software project.Software effort estimation fall under the categories of expert judgement,algorithmic and machine learning techniques.Results of our research have shown that evolutionary algorithms give more accurate results for software effort estimation as compared to traditional methods of software effort estimation.Moreover the comparison of different evolutionary algorithms is done to find which evolutionary learning algorithm is better for which situation.

Keywords : Machine learning,evolutionary algorithms,software effort estimation

Chapter 1: Introduction

Software effort estimation is one of the most crucial activity that aids in software project planning and in prediction of amount of time and cost required for a particular software project.The use of software is growing continuously in our society.As a result software companies need to produce software of high quality with in time and budget to assure competitiveness. As a result software effort estimaton is one of the most important task of software project managers.There are so many models proposed in the past for predicting software effort .But there are so many problems because data related to software projects in the beginning is incomplete and inconsistent.

1.1.Software Effort Prediction

Software effort estimation predicts the amount of effort and time required to complete a particular project [1].The concept of software effort estimation first apperaed in 1950's and after that it has caught the attention of software project manager who wants to make accurate estimations for software effort [2][3].Since that time lot of models have been developed for software effort estimation.Diversity of models in the area of software effort estimation indicates that software effort estimation is a very complex problem.Till date there is no unique software effort estimation model which can give fast and accurate software effort predictions under all circumstances.The challenge that most software organisations faces is to deliver the software within budget and on time with desired functionalities.Both under estimation and over estimation have negative impact on software development.Under estimation can lead to delay in schedules and cost overruns which in turn reduces the quality of delivered products.Over estimation can lead to loss of customers and partners .Quality of software effort estimation is the determining factor for success of any software project. It helps in reducing the risks associated with any software project. According to [4], the error in estimation decreases with the progress of project as new information about project is available with its progress and hence chances of error reduces. In this manner as project reaches towards its completion, the accuracy increases.

1.2 Classification of Software Effort Estimation Techniques

The literature reports a wide variety of techniques for software effort estimation.The broad categories for software effort estimation techniques are

- **Expert Judgement** :They are based on experience of person and intuition
- **Analogy based** and machine learning methods:They predict effort on the basis of analysis of projects with similar attributes.
- **Algorithmic techniques** are mathematical formula based and formula depends on a number of variables.

  s

Bhatia et al. [1] has provided more detailed classification of software projects.The broad categorization is given below:

• **Empirical techniques:** They are mostly analogy based where result depends on the database of past historical projects available.

.• **Model/Theory based techniques** : They are algorithm based and include techniques like COCOMO, SLIM, Functional point analysis.

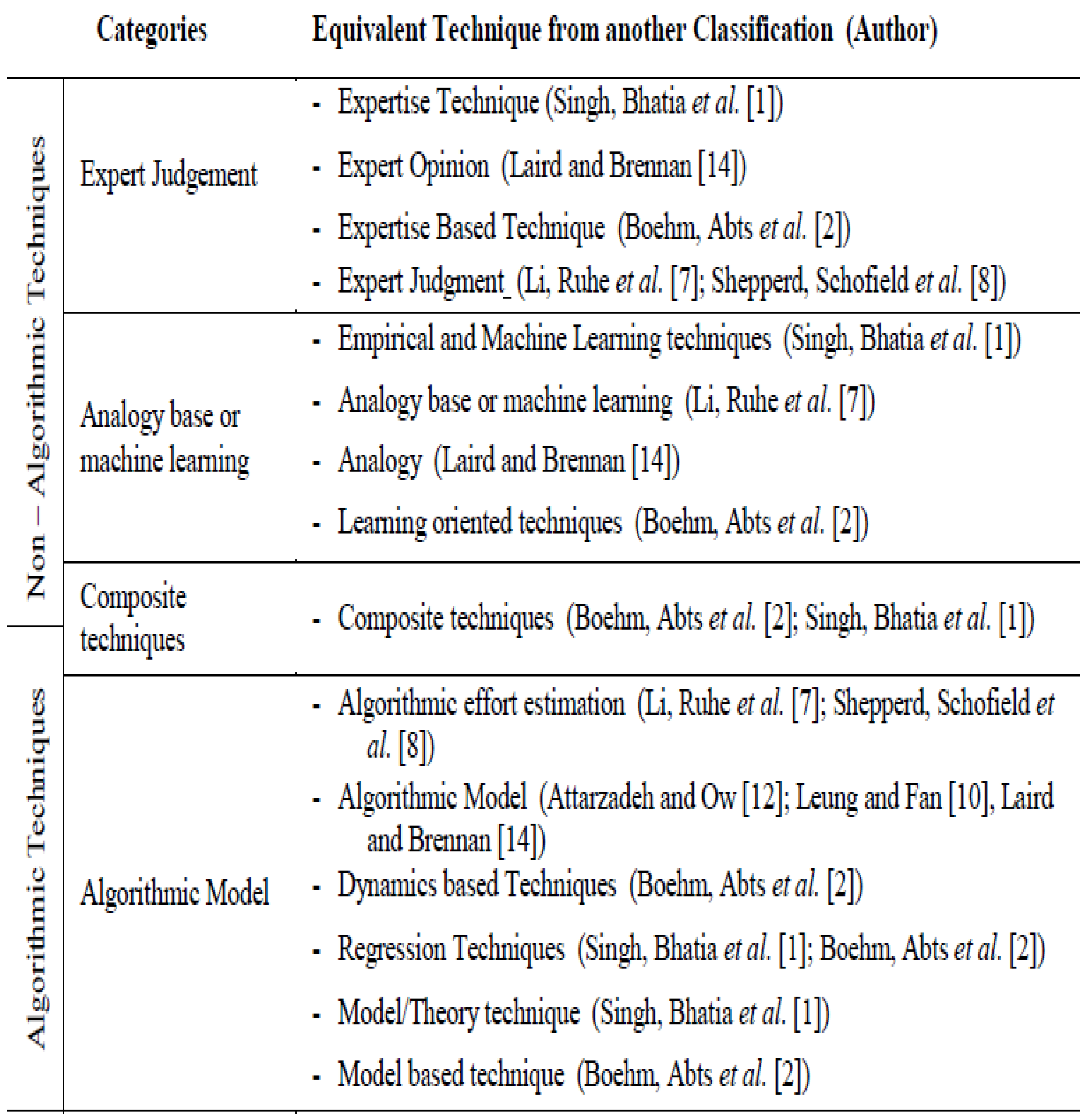
• **Expertise techniques:** They are based on reasoning power of the experts [1].

• **Regression based models** : They require past historical data to determine how variable y depends on variable x.

• **Composite techniques** Here expert judgement and past historical data are combined to give good estimates [4, 7].

Different techniques for software effort estimation by different authors in the history have been summarized in Table 1.1.

Table 1.1: Summary of software effort estimation techniques in literature of software effort estimation



	Categories	Equivalent Technique from another Classification (Author)
Non – Algorithmic Techniques	Expert Judgement	- Expertise Technique (Singh, Bhatia *et al.* [1]) - Expert Opinion (Laird and Brennan [14]) - Expertise Based Technique (Boehm, Abts *et al.* [2]) - Expert Judgment_(Li, Ruhe *et al.* [7]; Shepperd, Schofield *et al.* [8])
Non – Algorithmic Techniques	Analogy base or machine learning	- Empirical and Machine Learning techniques (Singh, Bhatia *et al.* [1]) - Analogy base or machine learning (Li, Ruhe *et al.* [7]) - Analogy (Laird and Brennan [14]) - Learning oriented techniques (Boehm, Abts *et al.* [2])
Non – Algorithmic Techniques / Algorithmic Techniques	Composite techniques	- Composite techniques (Boehm, Abts *et al.* [2]; Singh, Bhatia *et al.* [1])
Algorithmic Techniques	Algorithmic Model	- Algorithmic effort estimation (Li, Ruhe *et al.* [7]; Shepperd, Schofield *et al.* [8]) - Algorithmic Model (Attarzadeh and Ow [12]; Leung and Fan [10], Laird and Brennan [14]) - Dynamics based Techniques (Boehm, Abts *et al.* [2]) - Regression Techniques (Singh, Bhatia *et al.* [1]; Boehm, Abts *et al.* [2]) - Model/Theory technique (Singh, Bhatia *et al.* [1]) - Model based technique (Boehm, Abts *et al.* [2])

Table 1.2: Advantages and Disadvantages of different softwar effort estimation techniques

Techniques	Advantage	Disadvantage
Expert judgment	-These methods are very fast[13]. -They are very useful when historical data related to projects is not available[1]. - These methods provide estimations which are adjusted according to expert opinions - They are very good for estimating naïve projects [2].	- Here the estimations done by experts are sometimes questionable. - Factors that affect effort estimations are difficult to be documented. [1, 13, 2]. -Estimations provided by these methods may vary and is inconsistent. [2].
Analogy based	-They arevery accurate, simple and cheap [18].	- They have the capability to handle missing data -Quality of estimates depends on past historical data[8]; - Before making any estimations, database of relevant projects is required in advance [15]
Algorithmic effort estimation	- They are very fast, objective and easy to use methods [2]. - Accurate estimates are provided by these methods when similar historical project exists [8].	- We need to adjust them according to local conditions [8]. - They make use of size attribute which is very difficult to estimate in the beginning of software development lifecycle.

	- They can be iterated in many cycles to provide more accurate results [13].	- They have difficulty in modelling of complex set of attributes. - They do not support data that is divided in categorical way[12]. - Analysis of data using this method is very complex [16] - They have strong sensitivity towards outlierss [17].
Machine learning approaches	-They can model complex set of relationships between attributes very easily. -They are not sensitive to outliers	- They generally deals with domains that are understood very poorly. [8].

1.3.Aim of thesis

The main motive of research work is to apply evolutionary algorithms for software effort estimation.I have chosen evolutionary algorithms for empirical evaluation because there is not much research work using evolutionary algorithms in the area for software effort estimation.Evolutionary algorithms have shown outstanding performance in various industrial applications like image processing,speech recognition,signature verification and medical.So I have tried to analyse the performance of evolutionary algorithms for software effort estimaton.For this purpose publicly available datasets of different sizes have been taken.Range of sizes include very small,small,medium large and very large.Then different algorithms are applied on different datasets to check which algorithm is better for which dataset.This work is of utmost importance for software organisations because we need to find out best algorithms for different type of datasets so that prediction can be done in accurate manner.Accurate estimation can help software project managers in making accurate estimations for manpower and budget of projects.

1.4. Motivation

According to surveys conducted in the past approximately one third of the software projects run out of budget and two third of the projects exceed the deadline and original estimates.Its impossible for a project manager and system analyst to make accurate estimates of effort required to build a software product.Without accurate estimation,project managers can not determine how much time and man power is required for software project completion.This means the software portion of the project is out of control from beginning itself.In order to help the industry in development of quality products with in scheduled time accurate software effort estimation is required.There are many challenges in the area of software effort estimation .Some of the major ones are:

- There is no unique software effort estimation technique which can give accurate results under all circumstances.
- There exists strong dependency between type of data and software effort estimation technique being used.
- Estimates made by software effort estimation techniques till now are not so accurate.

1.5.Organization of Thesis

The research work carried out by me has been divided in to various chapters . The thesis is divided in to seven chapters.

Chapter 1 is the introductory part of thesis. It describes motivation of work, goals of thesis and the structure of thesis.

Chapter 2 Related work has been explained .Different effort estiamtion techniques used by different authors have been mentioned.

Chapter 3 describes the key concepts of the research work.The research work carried out by me uses key concepts of evolutionary techniques.Different evolutionary techniques with their advantages and disadvantages have been explained.Also data pre-processing technique KNN(K Nearest Neighbour) has been explained.

Chapter 4 It describes how research work was carried out.First of all six datasets used in research have been explained.Descriptive dtatistics of all the datasets have been given.After that estimation accuracy measures have been explained.They are the measures on the basis of which we judge the performance of our models.At the end,the tool used has been described very briefly.

In chapter 5 the results that came after applying evolutionary algorithms on different datasets have been given.The results are explained very clearly through tables and line charts.

Chapter 6 provides an overall conclusionof the work done.The results of the work are summarized in this chapter.Applications of the work are mentioned.Further,the future scope of the work is presented in this chapter.

Chapter 6 is followed by the references of various research papers(published in national and international journals and conferences) and books that have been gone through during the course of the thesis.

CHAPTER 2: Related work

There has been a lot of work done in the area of software effort estimation. Looking at the history of software effort estimation, we can infer that effort estimation can be done in three ways:

1. **Expert judgement**: It calculates effort by measuring the degree of similarity our project carries with the past historical project .It is not so accurate as accuracy depends on similarity .
2. **Algorithmic**: They are formula based. They are not so accurate as they are unable model all the sets of relationship between attributes on which our project depends.
3. **Machine learning methods:** Here effort estimation is based on applying various machine learning algorithms like neural networks, fuzzy, neuro-fuzzy, genetic algorithms. This is the best one as all set of relaionships between effort and independent variables can be modeled using machine learning techniques.

2.1. Main Research papers studied

Albrecht [25] in his paper has measured effort by taking functional points in to account. Effort calculation has been done in two steps:

1. Calculation of function points on the basis of various parameters like external input types, external output types, logic internal file types, external interface file types, external inquiry types.
2. These functional points are converted to lines of code and finally effort is calculated mathematical formula. Albrecht basically followed algorithmic approach.

Application of Machine Learning Methods for Software Effort Prediction by RuchikaMalhotra, Arvinderkaur and Yogesh Singhin 2010 [19] has proposed many models for estimating effort. Various methods used in the paper are least square regression, linear regression, MSP, M5 Rules, RBF,SVM, Pace regression and REP Tree. The performance measures used in the project are relative absolute error and root relative squared error. MSP and M5 rules outperformed all other models.

Software Effort Prediction using Statistical and Machine Learning Methods by RuchikaMalhotra and Ankita Jain in 2011 [20] has compared various methods like Linear Regression, Artificial Neural Network, Decision Tree, Support Vector Machine and Bagging on renowned software project dataset. China dataset consisting of 499 projects was used. Performance measures being used in the project are MMRE(Mean magnitude of relative error), RMSE(Root mean squared error), RAE(Relative absolute error), RRSE, PRED(25).Best performance was shown by decision trees amongst all methods.

The paper by Finnie and Wittig [21] They have examined the power of artificial neural networks and case based reasoning for effort estimation using dataset based on Australian Software Metrics Association. Also the same authors have compared the performance of artificial neural network models and case based reasoning models against linear regression. They came out with conclusion that the result vary from dataset to dataset.

Improved software effort estimation using MART: The paper by **Elish**[22] has proposed a multiple additive regression trees (MART) model and compared its result with well known models like RBF,Linear regression and Support vector regression.The results of MART model were better than RBF,Linear regression and support vector regression models.

C.J. Burgess and M.Lefley —"Can genetic programming improve software effort estimation? A comparative evaluation": The paper by Burgess and Lefley [23],the performance of genetic algorithms have been analysed for software effort estimation and comparison is don ewith well known models like Linear LSR and ANN.For this purpose Desharnais dataset consisting of 81 projects have been used.

Improving the Accuracy in Software Effort Estimation Using Artificial Neural Network Model Based on Particle Swarm Optimization : The paper by Zhang Dan[25] proposes an artificial neural network (ANN) prediction model that incorporates with Constructive Cost Model (COCOMO) which is improved by applying particle swarm optimization (PSO), PSO-ANN-COCOMO II, to provide a method which can estimate the software develop effort accurately. The modified model increases the convergence speed of artificial neural network and solves the problem of artificial neural network's learning ability that has a high dependency of the network initial weights. This model improves the learning ability of the original model and keeps the advantages of COCOMO model. Using two data sets (COCOMO I and NASA93) to verify the modified model, the result comes out that PSO-ANN-COCOMO II has an improvement of 3.27% in software effort estimation accuracy than the original artificial neural network.

Software Effort Estimation Using Neuro-Fuzzy Approach: The paper by Urvashi Rahul Saxena and S.P Singh [26] has used neurofuzzy model to gauge its performance for software effort estimation.The results of neurofuzzy models have been compared with traditional models like Halstead, Bailey Basili and doty models.MMRE and RMSE have been used as accuracy measures.

Optimization of COCOMO II Effort Estimation using Genetic Algorithm: This paper by Astha Dhiman and Chander Diwaker [27] aims to tune the coefficients used for effort calculation in COCOMO II post architecture model. The results are really very impressive.

Search-Based Approaches for Software Development Effort Estimation : This paper by Federica Sarro[29] makes use of search based techniques for software effort estimation. These approaches include a variety of meta-heuristics, such as evolutionary algorithms and local search. They search for suitable solutions to problems characterized by large search space, using an objective function that gives an indication of how a solution is suitable for the problem under investigation. AMSE, MMRE ,BMMRE has been used as performance measures.

A Comparison Between Decision Trees and Decision Tree Forest Models for Software Development Effort Estimation: In this paper by Ali BouNassif and Mohammad [28] a decision tree forest (DTF) model is compared to a traditional decision tree (DT) model, as well as a multiple linear regression model (MLR). The evaluation was conducted using ISBSG and Desharnais industrial datasets. Results show that the DTF model is competitive and can be used as an alternative in software effort prediction.

Improving the Accuracy of Software Effort Estimation based on Multiple Least Square Regression Models by Estimation Error-based Data Partitioning: In this paper by Yeong-SeokSeo [29] ,the clusters are built on the basis of MRE values. Then we build multiple LSR models on each of these clusters. After this effort is calculated using each of these models for their respective clusters. Finally, we have the cumulative effort.

Software Effort Estimation using Artificial Neural Networks: A Survey of the Current Practices: This survey paper by Dr. Haitham Hamza and Dr. Amr Kamel Khaled Shams provides[30] an overview on the use of Artificial Neural Networks methods to estimate the development effort for software development projects. In this survey an explanation, on why those methods are used and how accurate they are. Various methods being studied in the survey paper are Feed-forward neural network, Recurrent neural networks, Radial basis function (RBF) network and Neuro-fuzzy networks.

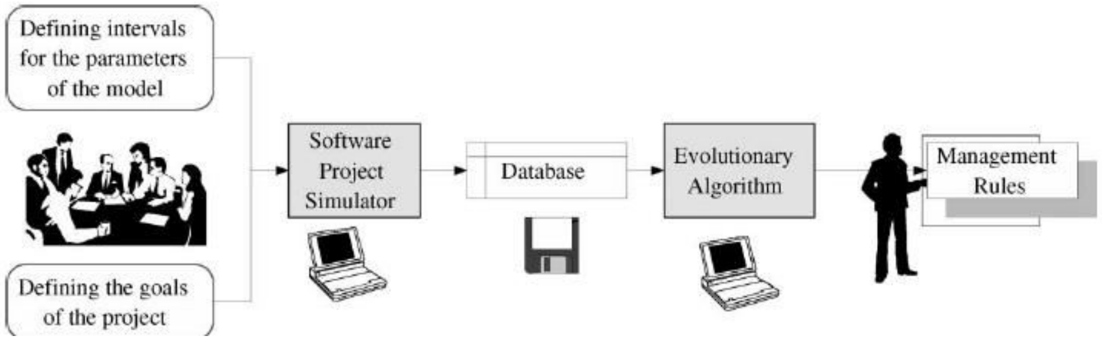
Prediction of Software Effort Using Artificial Neural Network and Support Vector Machine: This paper by Prabhakar and Maitreyee Dutta[31] have used artificial neural networks and support vector machine for predicting effort on china data set.Various performance measures used are Sum-Square-Error (SSE), Mean-Square-Error (MSE), Root-Mean-Square-Error (RMSE), Mean-Magnitude-Relative-Error (MMRE), Relative-Absolute-Error (RAE), Relative-Root-Square-Error (RRSE), Mean-Absolute-Error (MAE), Correlation Coefficient and Pred(25).

Probabilistic estimation of software size and effort : In this paper by Parag C. Pendharkar, **[32]** three machine learning methods such as Artificial Neural Networks (ANNs), Case-Based Reasoning (CBR) and Rule Induction (RI) have been used to estimate effort. Three factors have been used to compare software effort prediction systems .The factors are accuracy, explanatory value and configurability.

Software Development Effort Estimation using Fuzzy Logic: A Case Study: The paper by Martin [33] has used enhanced fuzzy model for software effort estimation.Performance is analyzed in terms of MMRE value.

An evolutionary approach to predicting software development effort : In this paper by Anguilar- Ruiz [34],a new approach based on the combination of Software Project simulator(SPS) and Evolutionary computation have been used.The main motive is to provide accurate decision rules so that manager can take accurate decisions at ant point of time.SPS generates a database when a project is input to it and that database is provided as input to evolutionary algorithm to produce accurate decision rules.The process for generation of management decision rules has been explained in Figure 2.1[34].

Figure 2.1 Decision rule generation process


Defining intervals for the parameters of the model
Defining the goals of the project
Software Project Simulator
Database
Evolutionary Algorithm
Management Rules

Software Development Effort Estimation Using Soft Computing: In this paper by Sandeep Kad and Vinay Chopra[35] soft computing technique have been used to improve the quality of software effort estimation. Various parameters used in COCOMO model have been fuzzified for more accurate effort estimation.MRE value of fuzzified COCOMO model comes out to be much better than algorithmic models.Gaussian membership function have shown much better results than triangular and trapezoidal memebership functions.

Software Effort Estimation Using Machine Learning Methods: This paper by Bilge baskeles [36] have applied various machine learning techniques like MLP,SVM,RBF etc on public datasets and data obtained from software organizations in turkey.Results of techniques vary from dataset to dataset.

2.2.Summary of Literature Survey

The literature survey has been summarized in table 1. The table has two columns, the author of the paper and the technique used.

Table 2.1 Summary of literature survey

Author name	**Technique used**
A. Albert and J.E. Gaffney	Measurement of effort by taking functional points in to account.
Ruchika Malhotra, Arvinder kaur and Yogesh Singh	MSP,M5 Rules, RBF,SVM, Pace regression and REP Tree.
Ruchika Malhotra and Ankita Jain	Linear Regression, Artificial Neural Network, Decision Tree, Support Vector Machine and Bagging.
Finnie and Wittig	Artificial neural networks and case based reasoning.
M.O.Elish	Multiple additive regression trees are being used for software effort estimation.
C.J. Burgess and M.Lefley	Genetic Programming
Author	Technique used
Zhang Dan	Artificial neural network model using particle swarm optimization
Urvashi Rahul Saxena and S.P Singh	Neurofuzzy techniques for software effort estimation.
AsthaDhiman and ChanderDiwaker	Optimization of COCOMO II using genetic algorithm

Federica Sarro	Hill Climbing, Tabu Search, Simulated Annealing) or Evolutionary Algorithms (e.g., Genetic Algorithms, Genetic Programming
Prabhakar and Maitreyee Dutta	Artificial Neural Network and support vector machine
Ali BouNassif and Mohammad	Decision tree and multiple linear regression model.
Yeong-SeokSeo	Multiple least square regression models
Dr.HaithamHamzaand Dr.AmrKamelKhaled Shams	Artificial neural network techniques.
Parag C.Pendharkar	Artificial Neural Networks,Case Based Reasoning and Rule Induction
Martin	Fuzzy logic
Anguilar-Ruiz	Combination of Software project Simulator and evolutionary computation
Sandeep Kadd and Vinay Chopra	Fuzzy logic
Bilge Baskeles	Multilayer perceptron,support vector machine and radial basis function

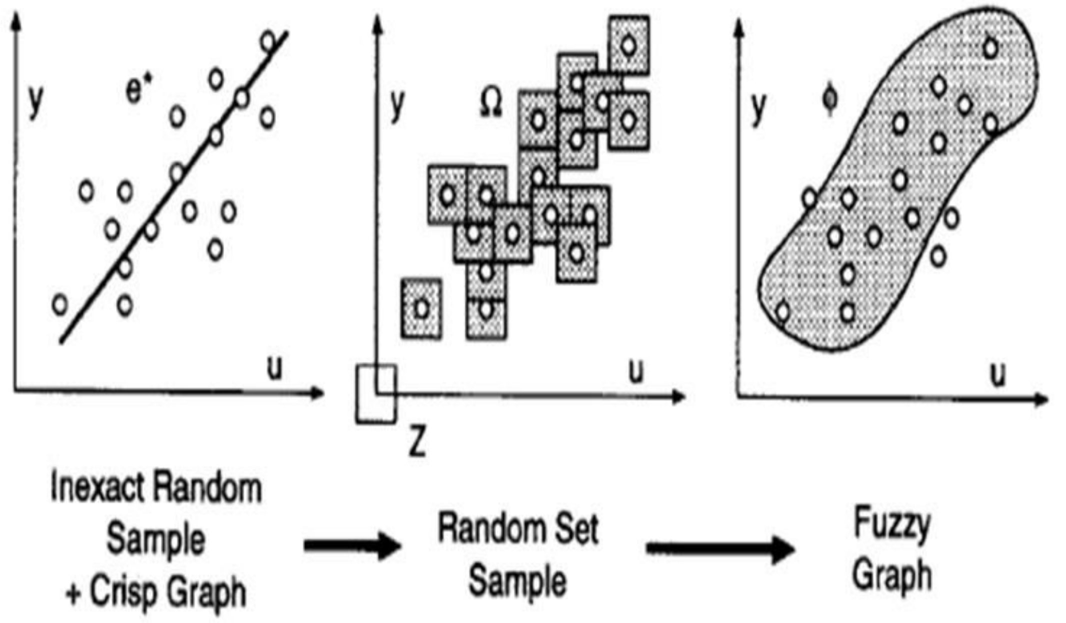
CHAPTER 3: Research Methodology

In this research work evolutionary algorithms have been used for software effort estimation .A brief introduction of each of these techniques is given below.

3.1.Fuzzy and Random sets based modeling(FRSBM)

The main problem that arises in signal processing is the nature of signals that causes interference. The interfering signals are often treated as random processes. Previous researches have shown that its possible to modify the models in favourable way if uncertainty processes that causes interference are treated as fuzzy experiments. So, keeping all these things in mind the sole purpose of FRSBM is to build a mathematical model which can justify the set of imprecise measurements taken over a system. Here the impact occurred to the system due to unmodelled inputs is treated as random process. Relationships between probabilistic,random-set –based and fuzzy set based modelling vave been shown in Figure 3.1[37].

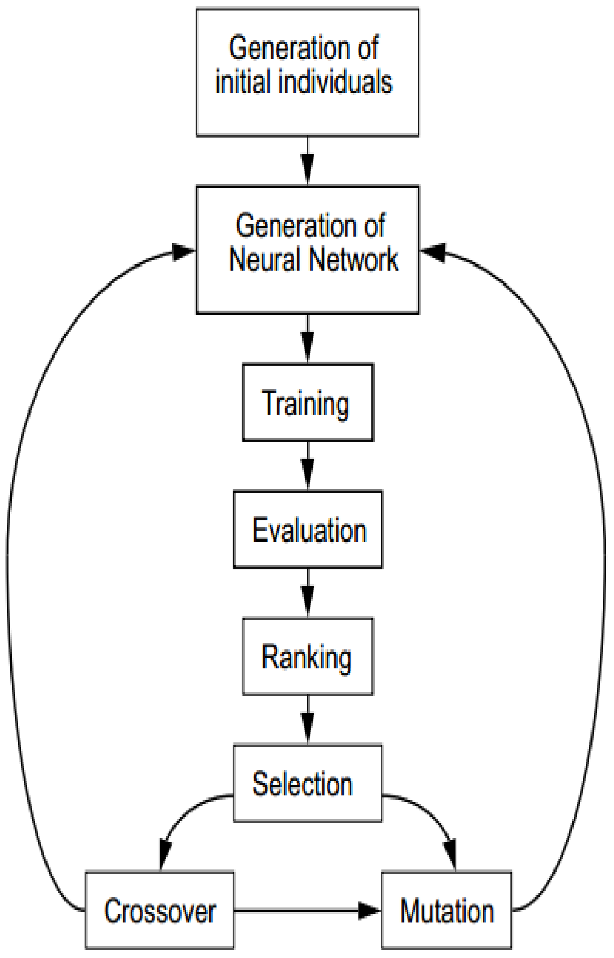
Figure 3.1 : Principal structure of FRSBM



3.2.Genetic Algorithm with Neural Network(GANN)

Genetic algorithms and neural networks have powerful problem solving capability. Both of them are based on simple principles. Neural networks using the concept of back propagation learning shows best results by searching various functions and finding best out of them. The success of training process depends on the selection of basic parameters like learning rate, initial weights etc. Genetic algorithms are basically global search algorithms that are based on various concepts like crossover, mutation, selection etc. In GANN genetic algorithms are used to optimize the network of neural networks. Figure 2 [38] shows how genetic algorithm and neural networks are used together inorder to obtain optimal solutions to a particular problem.

Figure 3.2: Principal structure of GANN System


Generation of initial individuals
Generation of Neural Network
Training
Evaluation
Ranking
Selection
Crossover
Mutation

3.3.Neural Network Evolutionary Programming(NNEP)

Basically neural networks are parallel processing structures and have the capability to recognize patterns. Although there are diverse areas where neural networks can be applied but they do have various limitations like avoidance of local convergence, to find an appropriate network architecture and processing capabilities for each individual neuron as well as adjusting the learning procedure[39].Evolutionary programming is used to address all these issues. Evolutionary programming can yield faster, efficient and robust training procedures. Arbitrary interconnections and neurons possessing additional processing capabilities can be accommodated[39].By using an evolutionary algorithm the structure and weights of static neural networks can be simultaneously acquired[39].Using evolutionary programming delayed links are introduced in to neural networks to form recurrent neural networks.

3.4. GFS-GAP-Sym-R(Symbolic Fuzzy Learning based on Genetic Programming Grammar Operators)

GA-P methods have the capability to form a generalized law given a set of samples.GA-P methods are easy as compared to others as they are very flexible and takes in to account the maximum complexity of a particular expression, maximum number of parameters and some random set of operations. In this method GA-P algorithm have been modified to produce a fuzzy model that is arithmetic based. This method can produce fuzzy estimations for outputs and parameters used.Additionaly symbolic information can be used here provided that information is encoded using fuzzy sets.

3.5 GFS-GSP-R(Symbolic Fuzzy Learning based on Genetic Programming Grammar Operators and Simulated Annealing)

Here genetic programming operators are combined with simulated annealing search to evolve fuzzy rule based classifiers. The genotype-phenotype encoding of fuzzy rule bases in GA, along with their corresponding crossover and mutation operators, can be used by other search schemes ,improving the behavior of these last ones[40]. A simulated annealing based method for inducing both parameters and structure of a fuzzy classifier has been developed. The adjacency operator used in simulated annealing (SA) has been replaced with a macromutation which is taken from tree-shaped genotype GA's. Results of simulated annealing are similar to genetic programming in terms of learning ability and linguistic interpretability. In addition to it,memory consumption of learning process is comparatively lower.

3.6. GFS-GP-R(Symbolic Fuzzy Learning based on Genetic Programming)

Genetic programming and genetic algorithms are combined in GA-P algorithms to solve symbolic regression problems. A fuzzy arithmetic based GA-P procedure is used to search an analytic expression which can relate input and output variables. The algorithm have been successfully tested on electrical engineering problems. Here in our research, we are using this algorithm for software effort estimation.

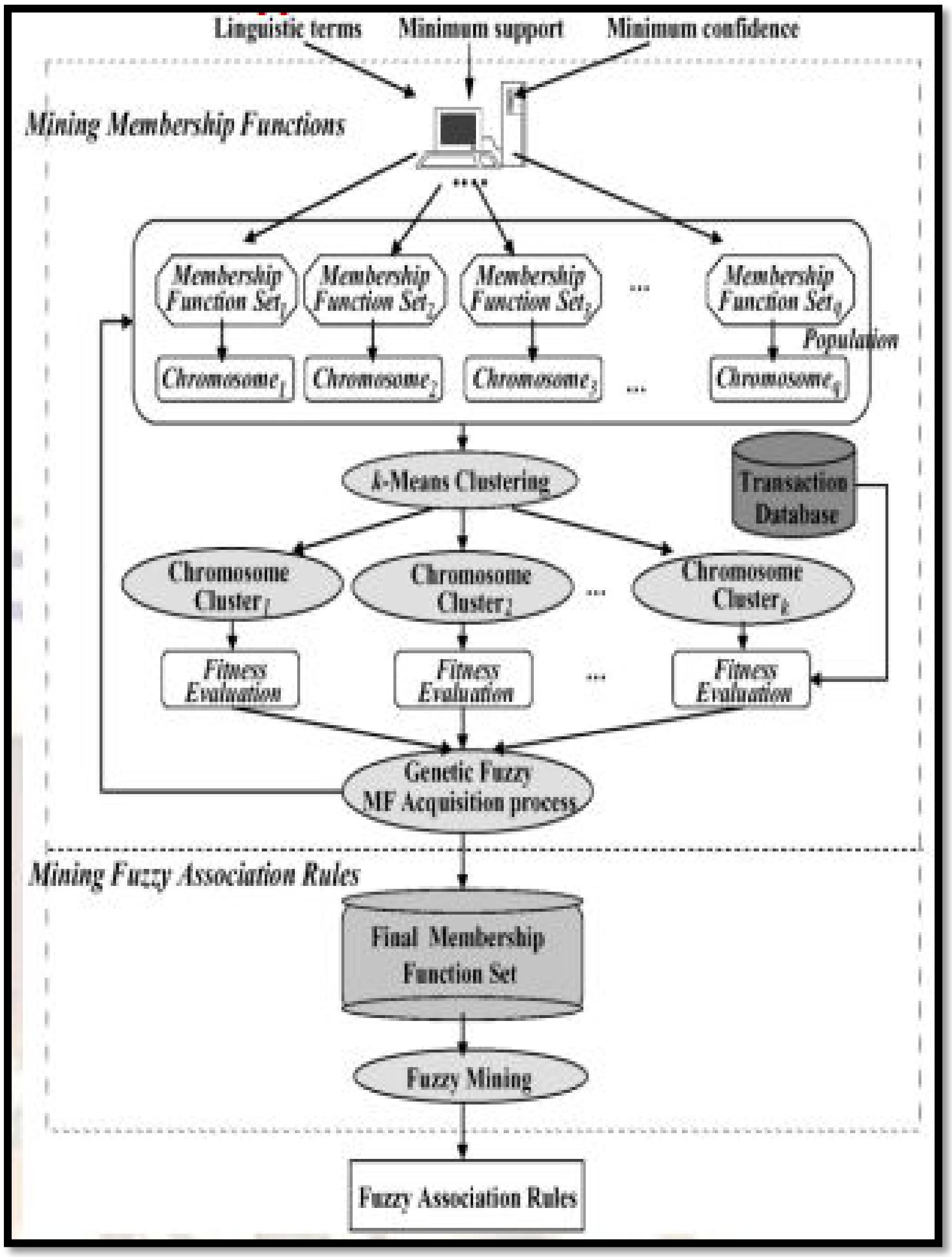
3.7. GFS-SAP-Sym-R(Symbolic Fuzzy-Valued Data Learning based on Genetic Programming Grammar Operators and Simulated Annealing)

In this approach research is devoted to new representations of fuzzy rule bases and application dependent crossover operator can be applied to different search schemes allowing them to be applied to new fields[41].This method has the capability to perform automatic feature subselection. As per the practical usefulness of this method , SA was more difficult to adjust than GP. Memory consumption is comparatively lesser using simulated annealing.

3.8. GFS-GPG-R(Fuzzy Learning based on Genetic Programming Grammar Operators)

Here genetic programming grammar operators and fuzzy rule inference engine is combined to form fuzzy rules that are very efficient and powerful to use.The basic framework o this method has been shown in Figure 3.3[49].

Figure 3.3 Framework for GFS-GPG-R



3.9. MLP-BP-R(Multilayer Perceptron with Back propagation training)

Its basically a feed forward model whose main function is to map the set of inputs to output.It makes use of non linear activation function and consists of three or more layers of neurons.Its advantage is that it can easily distinguish between data even if the data is not linearly separable.Basically in back propagation trainin of MLP, gradient of loss function is calculated.Loss function determines the error.Gradient of loss function in turn is used to optimize the weights .There are some general rules for building the architecture of multilayer perceptron with back propagation training .Some of the general rules are:

- The complexity of relationship input and output and number of neurons in hidden layer are directly proportional to each other.
- The amount of training data available decides the number of neurons in each layer.

The algorithm is divided in to two phases 1)Propagation 2)Weight update

First Phase: Propagation

There are two steps in propagation:

- Training patterns are propagated in forward direction so that propagation's output activations are generated.
- Using backward propagation output activations travels and deltas of all hidden and output layer neurons are generated.

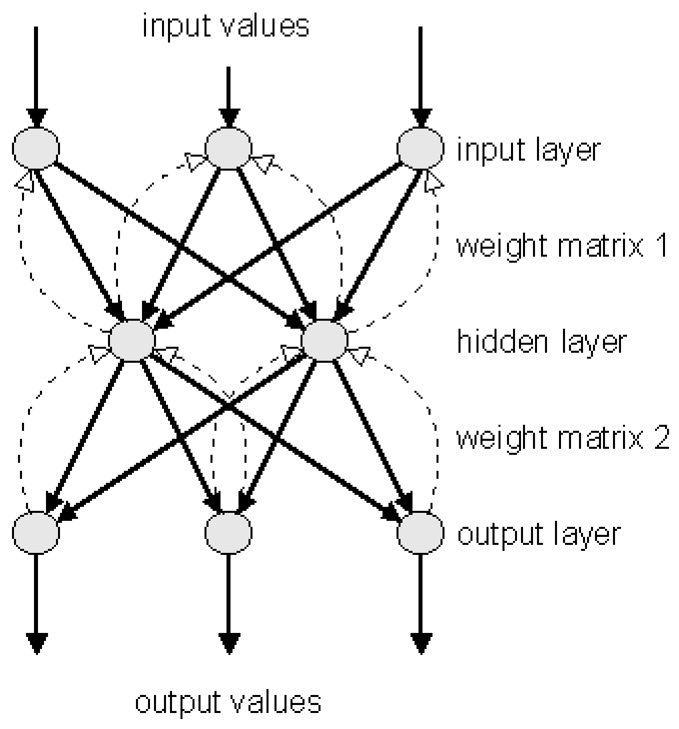
Phase 2: Weight update

For each weight-synapse following two steps are followed:

- the gradient of the weight is calculated by multilying output delta and input activation.
- After that a particular percentage or ratio of gradient is subtracted from weight.

Repeat phase 1 and phase 2 untill you get satisfactory results.An example of feed forward back propagation algorithm is given in Figure 3.4[50].

Figure 3.4 An example of Feedforward Multilayer Perceptron Backpropagation algorithm



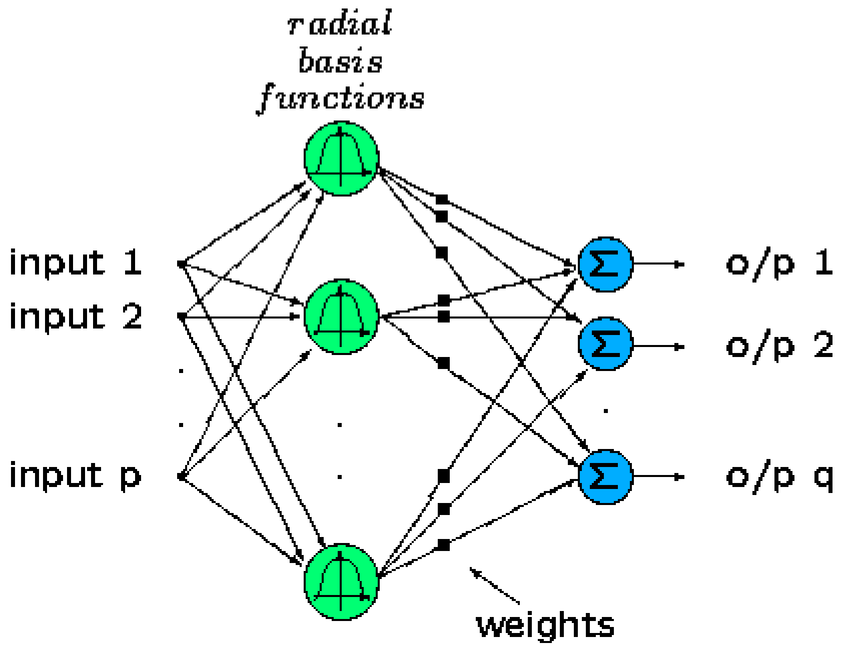
3.10. MLP-CG-R(Multilayer Perceptron with Conjugate Gradient Based Training)

An optimization technique i.e. scaled conjugate gradient (SCG) is introduced in this method in order to make it better than standard back propagation algorithms.It does not contain any parameters that are dependent on users.It does not include any time consuming operations like line search per learning iteration.As a result its faster than other second order algorithms proposed recently.

3.11. Incr-RBFN-R (Incremental Radial Basis Function Neural Network for Regression Problems)

By using this technique new data can be added very easily and invalid data can be removed efficiently.The method has been used successfully in various fields like image processing,pattern recognition,speech recognition,medical field.This method is also very fast due to low computation complexity.They are very powerful as they enable interpolation of scattered points in K-dimensional space.It has capability to handle large set of data points. The network graph of incremental RBF is given in Figure 3.5[51].

Figure 3.5 Network Graph of Incremental RBF



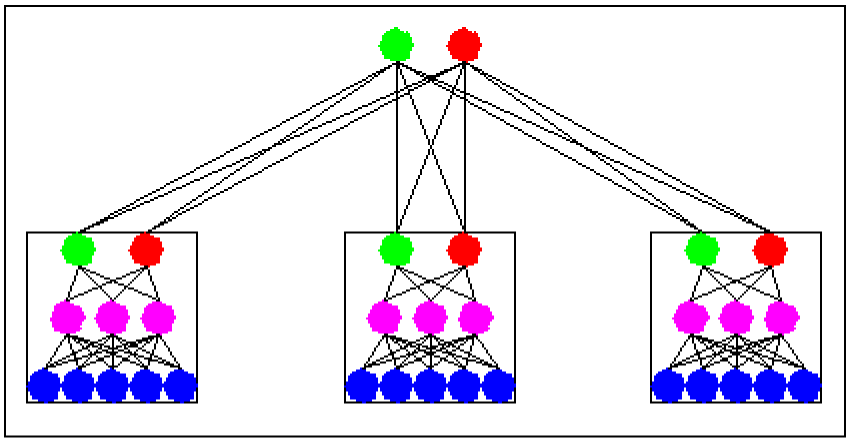
3.12. Ensemble-R(Ensemble Neural Network for Regression problems)

In ensemble averaging,first of all multiple neural network models are created.After that those models are combined to produce desired output.Generally error is reduced by combining several models as errors due to multiple models average out.Generally neural network ensembling consists of following steps:

- Generate N experts and set initial values for each of the experts.
- Train each and every expert.
- Combine the experts and after that average their values.

Ensembling of three neural network models is shown in Figure 3.6.

Figure 3.6 Ensembling of three neural networks



3.13.iRProp+-R (Improved Resilient backpropagation Plus)

This algorithm was proposed by Reidmiller and Braune.Its one of the first order learning algorithm giving best performance.As compared to other algorithms like Quick prop and conjugate gradient, its performance is better.

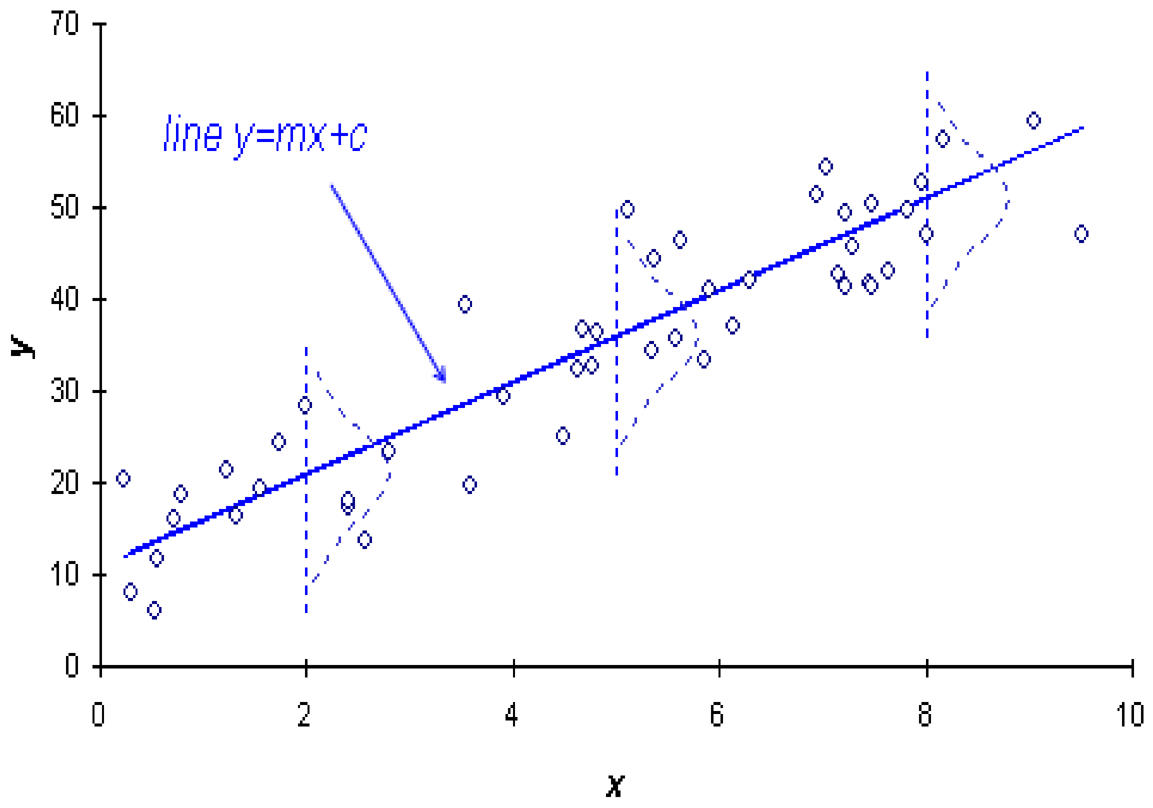
Advantages of this method are:

- The method gives very fast and accurate results.
- Its robust with repect to internal parameters and parameters are easily tunable.
- Its easy to implement.
- Its very suitable for the situations where the error is noisy.
- They are general methods and does not depend on particular network topology

3.14. Linear LMS-R

In this method our motive is to create a line from where the vertical distance of data points is minimum.Here we have independent variable say x and dependent variable ,say y.For example we need to create a line y=mx+c,where y is dependent variable and x is independent variable.Here m is the slope of line and c is the intercept made on y axis.The created line with respect to datapoints is shown in Figure 3[42].

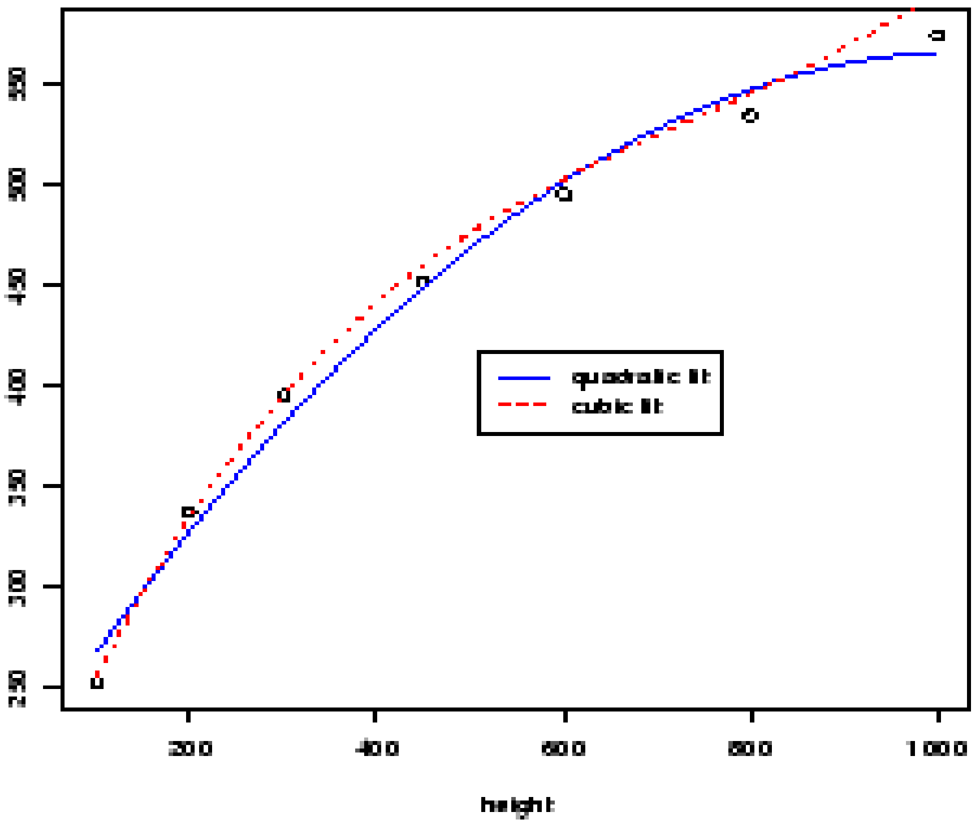
Figure 3.7 Linear LMS-R Method



3.15.Pol Quadratic LMS-R

It is non linear regression.Here the relation between dependent and independent variable is in the form of a quadratic equation.The advantage of Pol Quadratic LMS-R as compared to linear regression is the broad range of functions that they can cover.Disadvantage being strong sensitivity to outliers.Results of non linear analysis are affected in a serious way even if one or two outliers are present.Pol Quadratic fit has been shown in Figure 4[42].

Figure 3.8 PolQuadratic LMS-R



3.16.M5 Rules

It is used to form decision rules for regression problems using divide and conquer strategy.In every iteration,M5 is used to make model trees and best leaf is taken as the rule.

3.17.CART-R

Classification and regression trees are machine learning methods.They are prediction based models and make predictions from given set of data points.Here data space is partioned recursively and we try to fit simple prediction model with in each partition and in this way models are obtained.Decision trees are used to represent partioning graphically. For dependent variables,effort in our case, classification trees are designed which takes some finite values as input and error in predicting values is measured in term of misclassification cost.Regression trees are made for dependent variables which take continuous values as input and prediction error is taken as squared difference between actual and predicted values

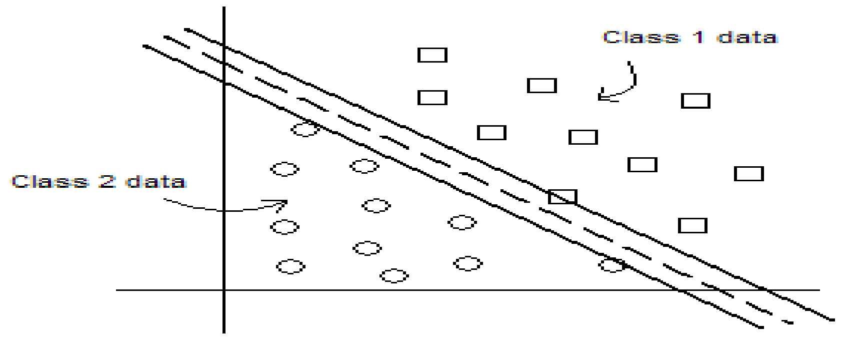
3.18.Thrift

It was proposed by thrift[43] and is based on Pittsburgh approach.It consists of various steps.First step comprises of learning fuzzy rules with a fixed set of fuzzy MFs set by hand.Relational matrix is used to represent the rule base.The created rules are then encoded in to chromosomes and membership function value is kept fixed.

3.19.NU_SVM-R

They are mainly used for classification and regression problems [44][45][47]. They tend to minimimize structural risk as well as error together[47]. This method have shown impressive results for software effort estimation.They were first used by oliveira.Their tendency to minimize risk has lead to their popularity.

Figure 3.9 Support Vector Machine



3.19 Epsilon SVM-R

SVM is a supervised learning method which is used to analyze data and recognize data after analyzing it.Its used for classification and regression problems.SVM takes a set of inputs and predicts to which of the two classes input belongs to.It makes SVM a non-probabilistic binary linear classifier.SVM have proved to be very successful for pattern classification problems.SVM is a classification and regression tool and uses machine learning to maximize the accuracy in prediction.

3.20 Preprocessing Algorithm KNN Missing value(K-Nearest Neighbour algorithm)

This algorithm is used in prediction of variables that are continuous.There are many variants of this algorithm.The algorithm used by us used weighted average of K-nearest neighbours weighted by the inverse of distance[53].

- First of al we need to compute the euclidean distance of the point from the set of trained data.
- Then we order the trained examples in increasing order of euclidean distance.
- Then we find the optimal number of K-nearest neighbours from trained data according to RMSE value.
- Then weighted average is calculated according to inverse of distance.

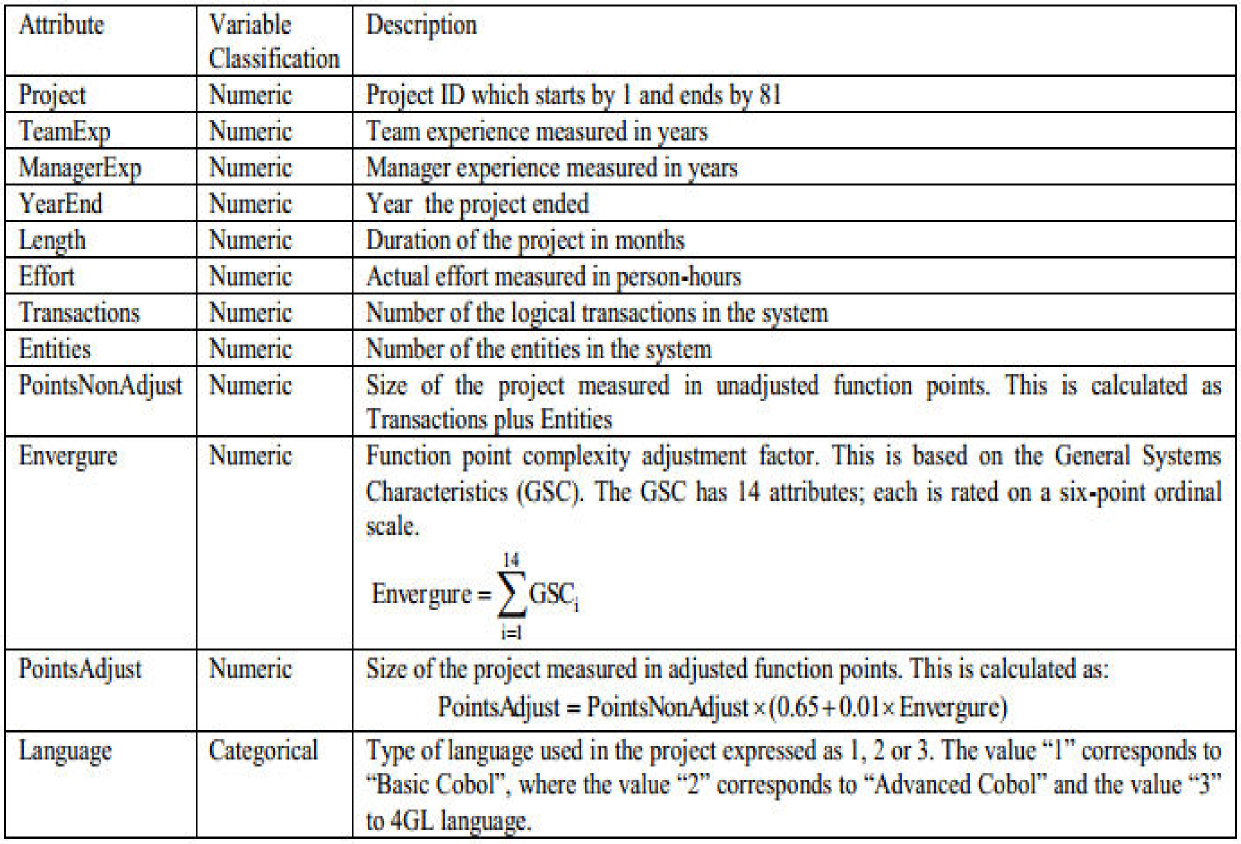
Chapter 4: Research Background

4.1. Datasets Description

For our research 6 datasets have been used. Description of datasets is as follows:-

4.1.1 Desharnais Dataset from Promise software repository have been used[48].This dataset is available publically to encourage research in the field of repeatable and verifiable software effort estimation models. The Desharnais dataset [48] is composed of a total of 81 projects developed by a Canadian software house in 1989. Each project has twelve attributes which are described in table I. The projects 38, 44, 65 and 75 contain missing attributes, so only 77 complete projects are used. The attributes of dataset have been explained below in Table 4.1[54]:-

Table 4.1:Desharnais dataset description



Attribute	Variable Classification	Description
Project	Numeric	Project ID which starts by 1 and ends by 81
TeamExp	Numeric	Team experience measured in years
ManagerExp	Numeric	Manager experience measured in years
YearEnd	Numeric	Year the project ended
Length	Numeric	Duration of the project in months
Effort	Numeric	Actual effort measured in person-hours
Transactions	Numeric	Number of the logical transactions in the system
Entities	Numeric	Number of the entities in the system
PointsNonAdjust	Numeric	Size of the project measured in unadjusted function points. This is calculated as Transactions plus Entities
Envergure	Numeric	Function point complexity adjustment factor. This is based on the General Systems Characteristics (GSC). The GSC has 14 attributes; each is rated on a six-point ordinal scale. $\text{Envergure} = \sum_{i=1}^{14} GSC_i$
PointsAdjust	Numeric	Size of the project measured in adjusted function points. This is calculated as: $\text{PointsAdjust} = \text{PointsNonAdjust} \times (0.65 + 0.01 \times \text{Envergure})$
Language	Categorical	Type of language used in the project expressed as 1, 2 or 3. The value "1" corresponds to "Basic Cobol", where the value "2" corresponds to "Advanced Cobol" and the value "3" to 4GL language.

Table 4.2: Descriptive statistics of Desharnais dataset :

Descriptive Statistics

	N	Minimum	Maximum	Mean		Std. Deviation	Variance
	Statistic	Statistic	Statistic	Statistic	Std. Error	Statistic	Statistic
TeamExp	81	0	4	2.21	.152	1.367	1.868
Mngrexp	81	0	7	2.57	.175	1.573	2.473
Yearend	81	83	88	85.79	.128	1.148	1.318
Length	81	1	39	11.72	.822	7.400	54.756
Transactions	81	9	886	179.90	15.924	143.315	20539.165
Entities	81	7	387	122.33	9.431	84.882	7204.975
Pointsadjust	81	73	1127	302.23	19.964	179.677	32283.757
Evergure	81	5	52	27.63	1.177	10.592	112.186
Pointsnonadjust	81	62	1116	287.05	20.568	185.108	34265.023
Language	81	1	3	1.56	.079	.707	.500
EFFORT	81	546	23940	5046.31	490.974	4418.767	19525503.816
Valid N (listwise)	81						

4.1.2. MAXWELL DATASET

Maxwell dataset have been used because this dataset is relatively new and contains 62 enteries which are enough for software effort prediction using evolutionary algorithms. Each entry in turn is described by 26 features. Except attributes numbered 1,24,25 and 26 all attributes are nummerical.

Table 4.3: Descriptive statistics of Maxwell dataset :

Descriptive Statistics

	N	Minimum	Maximum	Mean		Std. Deviation	Variance
	Statistic	Statistic	Statistic	Statistic	Std. Error	Statistic	Statistic
SYEAR	62	85	93	89.58	.271	2.131	4.543
APP	62	1	5	2.35	.126	.993	.987
HAR	62	1	5	2.61	.127	.998	.995
DBA	62	0	4	1.03	.056	.442	.196
IFC	62	1	2	1.94	.031	.248	.061
SOURCE	62	1	2	1.87	.043	.338	.114
TLONUSE	62	0	1	.24	.055	.432	.186
NLAN	62	1	4	2.55	.129	1.019	1.039
T1	62	1	5	3.05	.127	.999	.998
T2	62	1	5	3.05	.090	.711	.506
T3	62	2	5	3.03	.113	.886	.786
T4	62	2	5	3.19	.089	.698	.487
T5	62	1	5	3.05	.090	.711	.506
T6	62	1	4	2.90	.088	.694	.482
T7	62	1	5	3.24	.114	.900	.809
T8	62	2	5	3.81	.121	.955	.913
T9	62	2	5	4.06	.094	.744	.553
T10	62	2	5	3.61	.113	.894	.799

T11	62	2	5	3.42	.125	.984	.969
T12	62	2	5	3.82	.088	.690	.476
T13	62	1	5	3.06	.121	.956	.914
T14	62	1	5	3.26	.128	1.007	1.014
T15	62	1	5	3.34	.095	.745	.556
duration	62	4	54	17.21	1.353	10.651	113.447
Size	62	48	3643	673.31	99.579	784.085	614788.511
Time	62	1	9	5.58	.271	2.131	4.543
Effort	62	583	63694	8223.21	1333.489	10499.903	110247966.529
Valid N (listwise)	62						

4.1.3.CHINA DATASET

The dataset consists of 19 drivers for predicting software effort.Total number of project instances are 499.Descriptive statistics of the dataset is given in Table 4.4.

Table 4.4: Descriptive statistics of China dataset

Descriptive Statistics

	N	Minimum	Maximum	Mean	Std. Deviation	Variance
ID	499	1	499	250.00	144.193	20791.667
AFP	499	9	17518	486.86	1059.171	1121844.130
Input	499	0	9404	167.10	486.339	236525.209
Output	499	0	2455	113.60	221.274	48962.349
Enquiry	499	0	952	61.60	105.423	11113.975
File	499	0	2955	91.23	210.271	44213.887

Interface	499	0	1572	24.23	85.041	7231.971
Added	499	0	13580	360.35	829.842	688638.298
Changed	499	0	5193	85.06	290.857	84597.817
Deleted	499	0	2657	12.35	124.224	15431.634
PDR_AFP	499	.3	83.8	11.771	12.1056	146.547
PDR_UFP	499	.3	96.6	12.080	12.8187	164.319
NPDR_AFP	499	.4	101.0	13.270	14.0098	196.276
NPDU_UFP	499	.4	108.3	13.626	14.8434	220.327
Resource	499	1	4	1.46	.824	.679
Dev_Type	499	0	0	0.00	0.000	0.000
Duration	499	1.0	84.0	8.719	7.3471	53.979
N_Effort	499	31	54620	4277.64	7071.248	50002548.781
Effort	499	26	54620	3921.05	6480.856	42001489.307
Valid N (listwise)	499					

4.1.4. NASA Dataset

NASA dataset have been used to evaluate the performance of evolutionary algorithms. This dataset was donated by Bailey and Basiliin 1981.It wasfirst used by Shin and Goel in 2000 and after that Oliveira in 2006.The dataset consists of 18 project instances.There are two independent attributes i.e. M (methodology used) and DL (Nnumber of developed lines of source code with comments). Effort is the dependent attribute which is measured in terms of number of man months required for project completion

Table 4.5: Descriptive Statistics of NASA18 dataset

Descriptive Statistics

	N	Minimum	Maximum	Mean		Std. Deviation	Variance
	Statistic	Statistic	Statistic	Statistic	Std. Error	Statistic	Statistic
DL	18	2.1	100.8	33.589	7.6827	32.5950	1062.434
M	18	19	35	27.78	1.269	5.386	29.007
E	18	5.0	138.3	49.472	10.7777	45.7259	2090.855
Valid N (listwise)	18						

4.1.5Heiat Heiat Dataset:

It had 33 instances of software projects.The dataset consists of one independent variable size and one dependent variable effort.Size is measured as number of lines of code of project.Effort is calculated as the number of hours required for the completion of a particular project**.**

Table 4.6: Descriptive statistics of HeiatHeiat dataset

Descriptive Statistics

	N	Minimum	Maximum	Mean		Std. Deviation	Variance
	Statistic	Statistic	Statistic	Statistic	Std. Error	Statistic	Statistic
SIZE	33	148.0	368.0	235.061	9.9162	56.9643	3244.934
EFFORT	33	14.2	47.8	24.785	1.4578	8.3746	70.134
Valid N (listwise)	33						

4.1.6.Miyazaki94 Dataset:

It was given by Miyazaki in 2010.This dataset consists of 48 instances of software projects.There are 9 attributes in total.Out of 9 attributes 1 is identifier, 7 are condition attributes and 1 is decision attribute.

Table 4.7 :Attribute Information

Attribute name	Description
ID	It is the unique identifier associated with each project.
KSLOC	Number of sourcle lines of code in COBOL Project excluding comment lines.
SCRN	The total number of input and output screens of project.
FORM	The total number of forms in project.
File	It I sthe count of total number of files in the project
ESCRN	It is the count of total number of data elements in all the screens.
EFORM	It is the count of total number of data elements in all the forms.
EFILE	It is the count of total number of data elements in all the files.
MM	Total number of man months required to complete the project.

Table 4.7 Descriptive Statistics of Miyazaki94 Dataset

Descriptive Statistics

	N	Minimum	Maximum	Mean		Std. Deviation	Variance
	Statistic	Statistic	Statistic	Statistic	Std. Error	Statistic	Statistic
Kloc	48	6.9	417.6	70.792	12.6393	87.5678	7668.119
Scrn	48	0	281	33.69	6.818	47.236	2231.283
Form	48	0	91	22.38	2.966	20.548	422.239
File	48	2	370	34.81	7.703	53.365	2847.773
Escrn	48	0	3000	525.60	90.364	626.058	391948.670
Eform	48	0	1566	460.67	57.275	396.816	157462.993
Efile	48	57	45000	1854.58	923.559	6398.605	40942141.993
Effort	48	5.6	1586.0	87.475	33.0186	228.7597	52331.020
Valid N (listwise)	48						

4.2.Estimation accuracy measures

In thesis work standard and renowned software effort estimation measures have been taken. Most of the studies done in this area have used these measures for software effort estimation. MMRE value can be calculated for any daataset consisting of m observatons in following manner [22]:

$$\text{MMRE}=1/\text{m}\sum_{\text{i}=1}^{\text{m}} \text{MREi}$$

Where MREi is difference between actual and predicted values divided by the actual value.. It is calculated as follows[22]:

$$\text{MREi}=\frac{X_i-\overline{X\iota}}{X_i}$$

PRED(25)=m/n

Here m is the count of observations with MMRE value less than or equal to 0.25 and n is the count of datapoints in our dataset [22].

PRED(50)=m/n

Here m is the count of observations with MMRE value less than or equal to 0.25 and n is the count of datapoints in our dataset [22].

PRED(75)=m/n

Here m is the count of observations with MMRE value less than or equal to 0.25 and n is the count of datapoints in our dataset [22].

4.3. Cross validation

10-cross validation method

In 10 cross validation the data sample is partioned in a random way in to 10 subsamples of equal size.Out of these 10 subsamples,one subsample is retained as the testing sample and 9 subsamples are used to train our model.This process is repeated again and again untill evry subsample is used as testing sample.After that we take the average of all 10 results available.Cross validation can be k0cross validation but generally k is taken as 10.

4.4.Tool used for result calculation

KEEL tool has been used in our research work. KEEL tool is an open source java tool.Its used to assess evolutionary algorithms for data mining problems.Its mainly used to solve regression,classification and clustering mining problems.Different approaches related to genetic fuzzy algorithms like Pittsburgh,michigan etc have been collected together at one place.KEEL tool can be used by different users with different expectations.Three main functionalities provided by KEEL tool are:

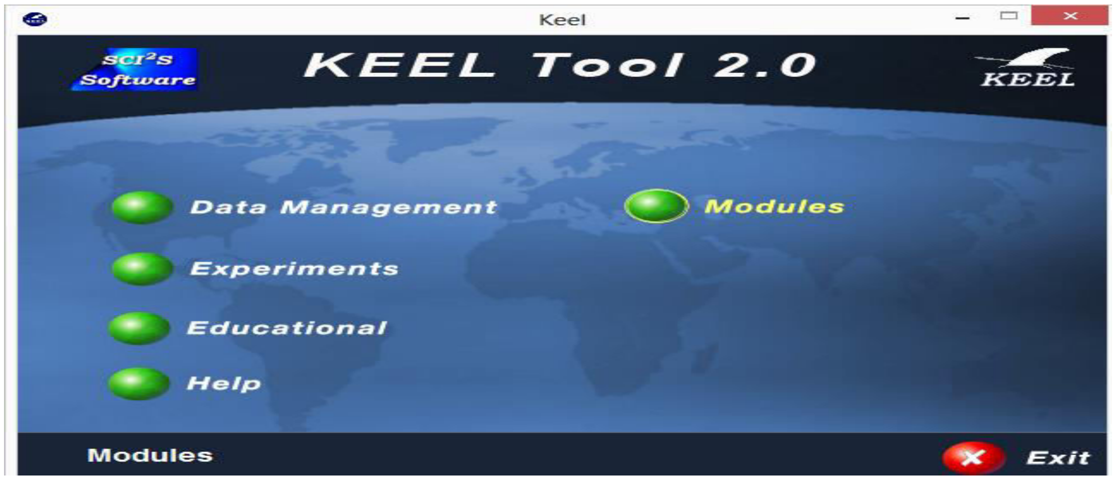
- **Data Management**: It provides features to add our own new data,export and import data from other formats to our format,data visualization and data partioning.
- **Experiments**:Its used to design our own experiments by selecting the data and setting various parameters.
- **Educational**:Its used to create experiments and run it step by step in order to show the learning process.

Main features that KEEL tool provides are:

It provides various preprocessing algorithms like discretization, instance selection,missing values supplement,data transformation etc.Post processing algorithms can be used to improve the results obtained from various knowledge extraction algorithms. Post processing algorithms ensure ensure membership function tuning, fuzzy rule weighing and selection[].It also contains statistical library so that results of various algorithms can be analyzed and compared with each other.Here offline expeiments can be set up for research purpose and online experiments for education purpose.It provides user friendly graphical user interface where multiple functional nodes can be combined to form a single process.

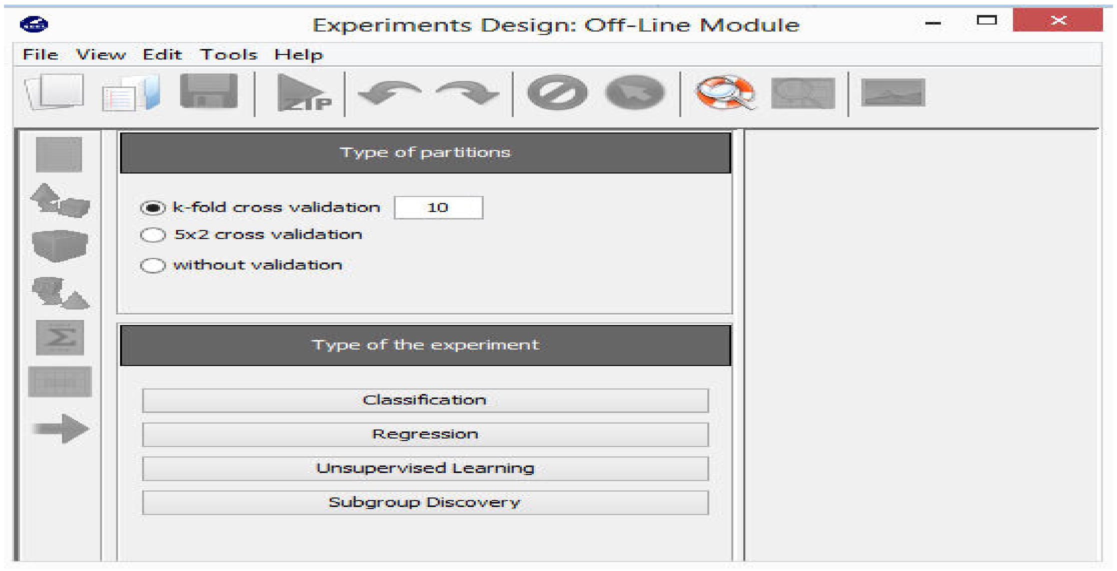
The screenshot of KEEL tool user interface is given in Fire 4.1.We need to click on experiments to proceed further for experiments of regression type.

Figure 4.1: Screen shot of user Interface 1 of KEEL Tool



After clicking on experiments we can choose experiments of different types like regression ,classification,suoervised,unsupervised according to our problem and apply different pre processing and post processing algorithms by creating network graph.Screenshot of screen after selecting experiments is given in Figure 4.2.

Figure 4.2: Screenshot of user interface 2 of KEEL Tool



Type of partitions

(● k-fold cross validation) 10
○ 5x2 cross validation
○ without validation

Type of the experiment

Chapter 5: Results and discussion

The results that came after the application of different evolutionary algorithms to different datasets have been explained below with the help of tables and after that a detailed discussion about the results has been done.

5.1. Analysis using Miyazaki94 dataset

5.1.1 Network Graph of analysis using Miyazaki94 dataset

Figure 5.1(a) Miyazaki94 Network Graph 1

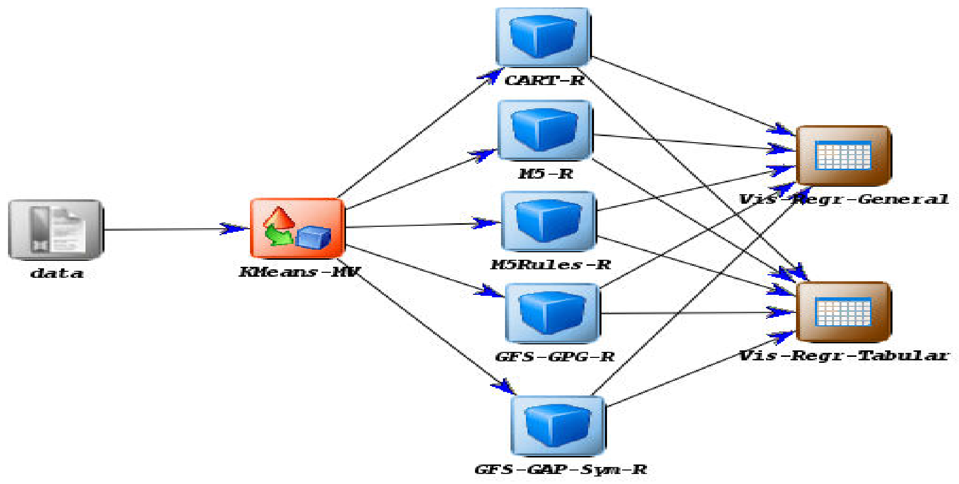


Figure 5.1(b) Miyazaki94 Network Graph 2

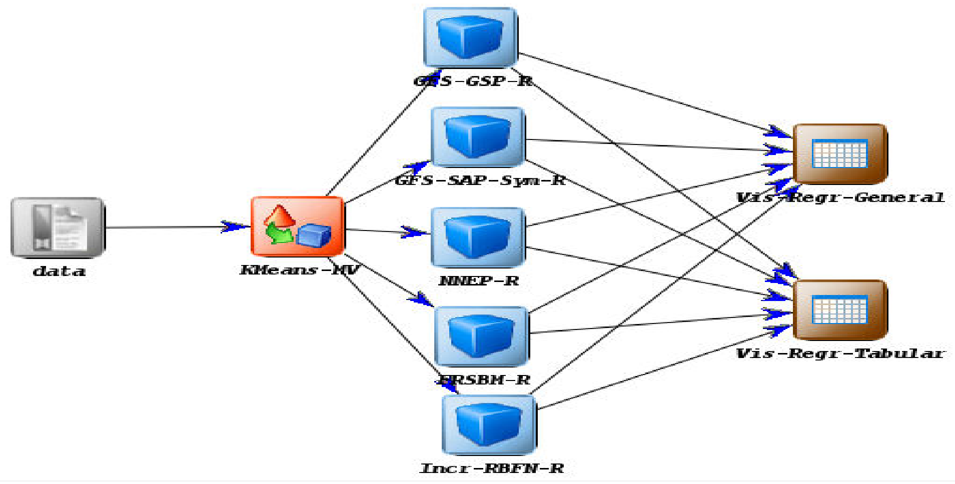


Figure 5.1(c) Miyazaki94 Network Graph 3

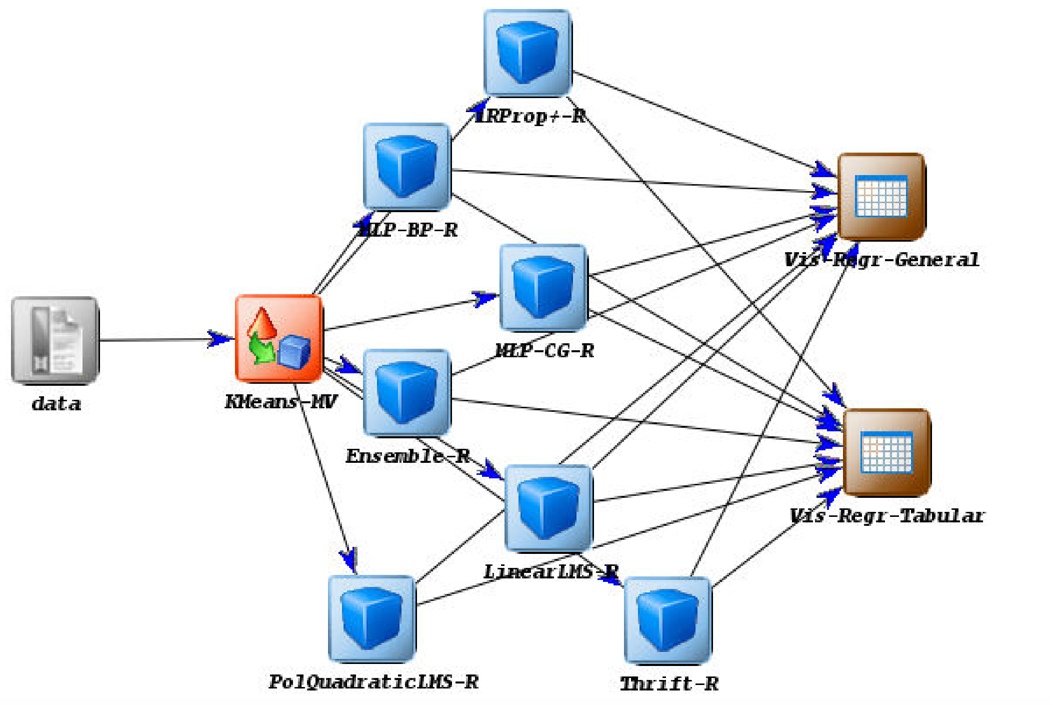


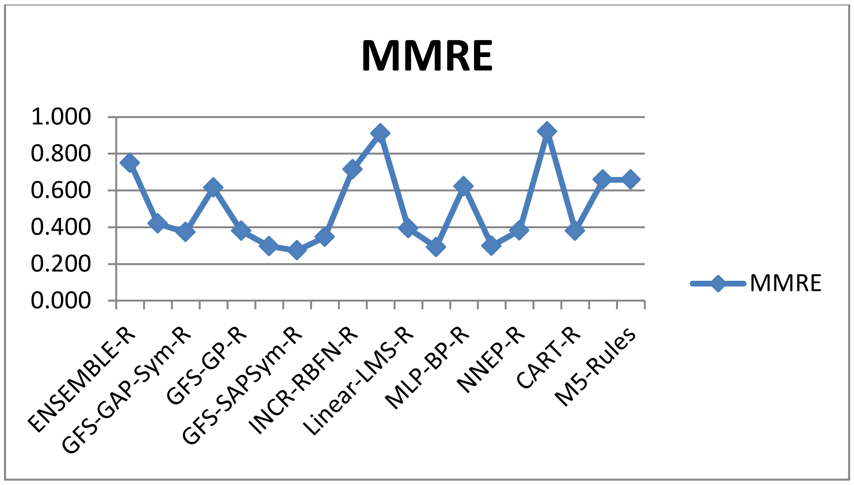
Table 5.1: Results with Miyazaki94 dataset

TECHNIQUE	MRE	MMRE	PRED(25)	PRED(50)	PRED(75)
ENSEMBLE-R	0.143	0.750	0.125	0.333	0.500
FRSBM	0.067	0.419	0.208	0.396	0.563
GFS-GAP-Sym-R	-0.012	0.373	0.229	0.458	0.604
GFS-GPG-R	0.552	0.615	0.167	0.333	0.438
GFS-GP-R	-0.014	0.379	0.229	0.458	0.583
GFS-GSP-R	-0.035	0.296	0.313	0.500	0.604
GFS-SAPSym-R	-0.006	0.273	0.333	0.521	0.625
GFS-SP-R	0.096	0.346	0.542	0.542	0.729
INCR-RBFN-R	-0.084	0.714	0.125	0.208	0.250
iRProp+-R	-0.269	0.910	0.167	0.313	0.396
Linear-LMS-R	0.030	0.393	0.188	0.354	0.500
PolQuadratic-LMS-R	-0.110	0.292	0.208	0.458	0.542
MLP-BP-R	-0.143	0.623	0.146	0.417	0.563
MLP-CG-R	-0.007	0.298	0.354	0.583	0.708
NNEP-R	0.066	0.382	0.292	0.500	0.771
Thrift	-0.780	0.920	0.021	0.021	0.042
CART-R	0.131	0.380	0.313	0.563	0.688
M5-R	0.176	0.658	0.208	0.354	0.563
M5-Rules	0.176	0.658	0.208	0.354	0.563

5.1.2.Discussion of results with Miyazaki94 dataset

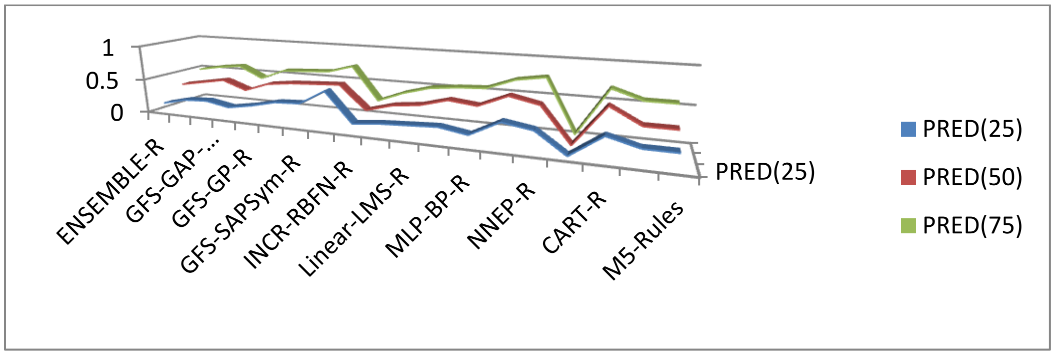
As can be seen from Figure 5.2 GFS-SAP-Sym-R outperforms all other models in term of MMRE value. It is having the lowest MMRE value of 0.27. Its MRE value is also lowest i.e. 0.005. After this MLP-CG-R, GFS-GSP-R and PolQuadratic LMS-R have also shown good performance having MMRE values of 0.298, 0.296 and 0.291 respectively. The worst performance has been shown by the Thrift method with highest MMRE value of 0.92.

Figure 5.2 Line chart MMRE Vs Methods



It is evident from Figure 5.3 that GFS-SP-R method is the best in terms of pred(25). The value of pred(25) for this method is 0.54.MLP-CG-R has shown good performance in terms of pred(50) value having pred(50) of 0.58.NNEP-R is best method in terms of pred(75) value.Incr-RBFN-R is worst method in terms of pred(25), pred(50) and pred(75) values.The values are 0.12,0.20 and 0.25 respectively.

Figure 5.3 Line chart PRED values Vs Methods



5.2.Analysis of results with NASA Dataset

5.2.1 Network Graph for NASA18 Dataset

Figure 5.4(a) NASA18 Network Graph 1

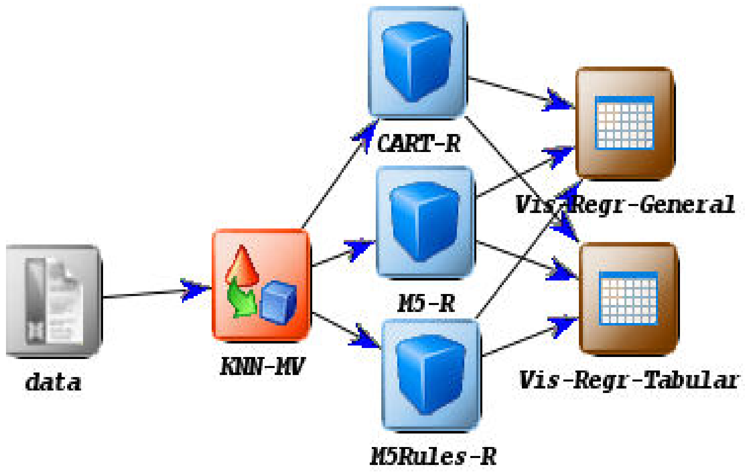


Figure 5.4(b)NASA18 Network Graph 2

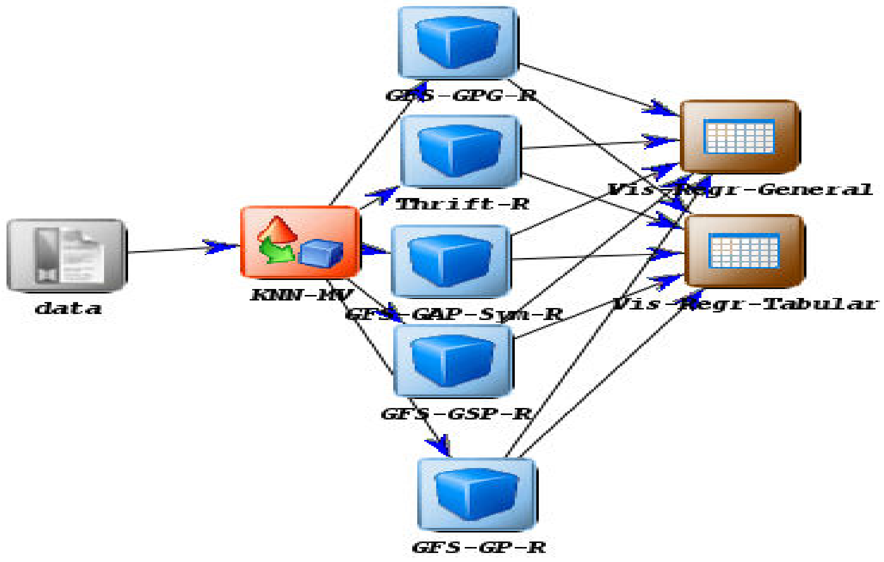


Figure 5.4(c) NASA18 Network Graph 3

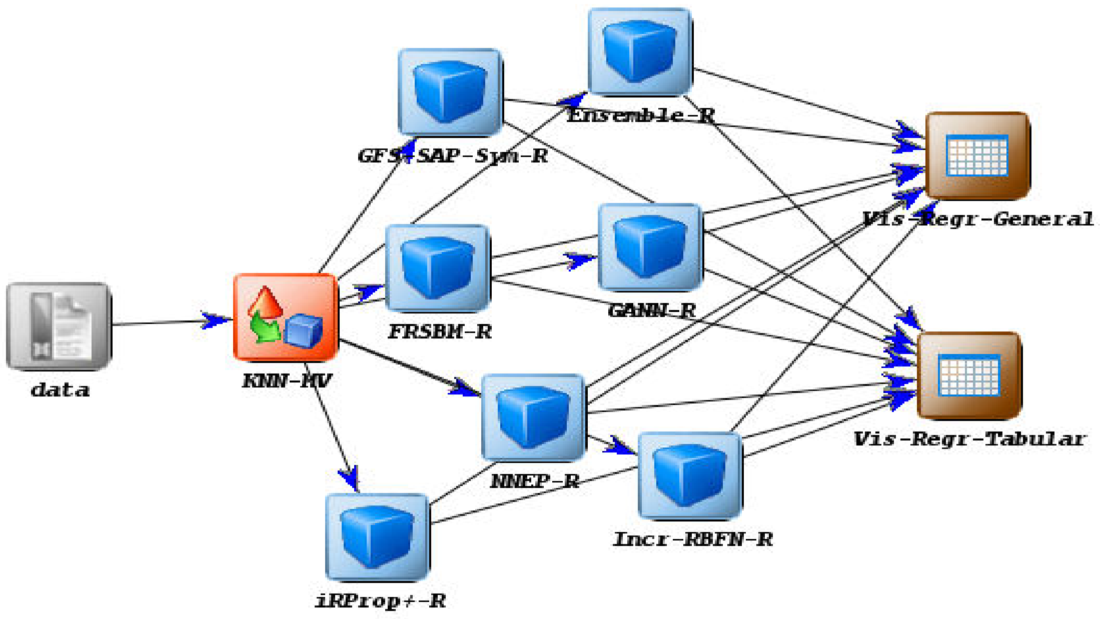


Figure 5.4(d)NASA18 Network Graph 4

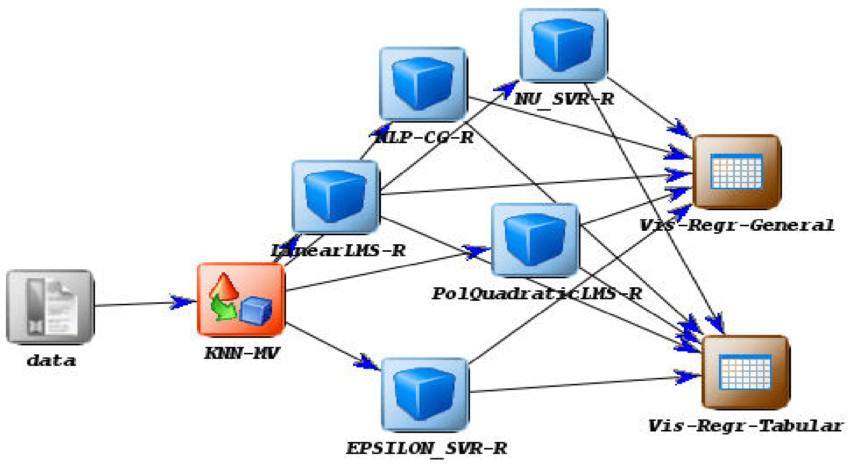


Table 5.2 Results with NASA dataset

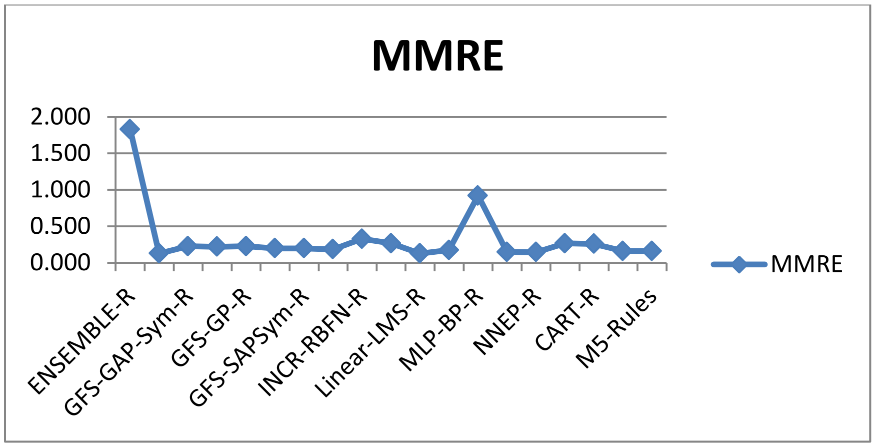
TECHNIQUE	MRE	MMRE	PRED(25)	PRED(50)	PRED(75)

ENSEMBLE-R	-1.585	1.829	0.278	0.333	0.333
FRSBM	0.014	0.127	0.778	0.944	0.944
GFS-GAP-Sym-R	-0.088	0.226	0.611	0.778	0.889
GFS-GPG-R	-0.142	0.219	0.667	0.833	0.944
GFS-GP-R	-0.088	0.226	0.611	0.778	0.889
GFS-GSP-R	-0.076	0.198	0.778	0.833	0.889
GFS-SAPSym-R	-0.076	0.198	0.778	0.833	0.889
GFS-SP-R	-0.065	0.185	0.778	0.900	0.889
INCR-RBFN-R	-0.009	0.328	0.444	0.667	0.944
iRProp+-R	-0.115	0.265	0.500	0.778	0.833
Linear-LMS-R	0.014	0.127	0.778	0.944	0.944
PolQuadratic-LMS-R	0.023	0.173	0.667	0.944	0.944
MLP-BP-R	-0.292	0.920	0.278	0.333	0.494
MLP-CG-R	0.013	0.147	0.722	0.944	0.944
NNEP-R	-0.010	0.144	0.778	0.944	0.944
Thrift	-0.169	0.267	0.611	0.722	0.778
CART-R	-0.024	0.259	0.556	0.889	1.000
M5-R	-0.026	0.160	0.778	0.889	1.000
M5-Rules	-0.026	0.160	0.778	0.889	1.000

5.2.2.Discusssion of results with NASA Dataset

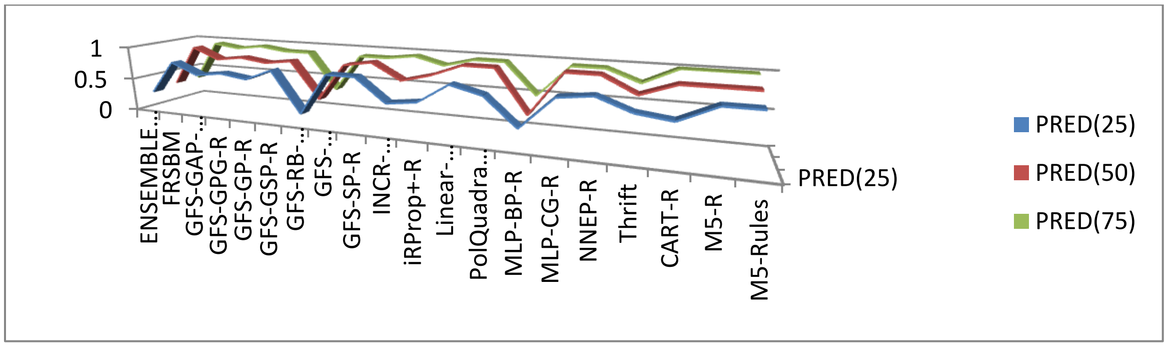
FRSBM and Linear LMS-R have comparable performance in terms of MMRE value.MMRE value is almost same for two .The value is 0.1266.After this NNEP-R and MLP-CG-R have also shown comparable performance having MMRE values of 0.144 and 0.147 respectively.Incr-RBFN-R has lowest MRE value of -0.008.Ensemble-R is worst method in terms of MMRE value having MMRE value of 1.82.

Figure 5.5 Line chart MMRE Vs Methods



FRSBM,GFS-GSP-R,GFS-SAP-Sym-R,GFS-SP-R,Linear LMS-R,NNEP-R and M5 Rules have shown best performance in terms of PRED(25) value with PRED(25) value of 0.778. FRSBM ,Linear LMS-R, PolQuadratic LMS-R,MLP-CG-R and NNEP-R are best in terms of PRED(50) value having value of 0.9444.CART-R and M5 Rules are best method in terms of PRED(75) value,value being 1 for both of them.

Figure 5.6 Line chart PRED Values Vs Methods



So overall for NASA18 dataset FRSBM and Linear LMS-R are best methods showing comparable performance.

5.3.Analysis of results with HeiatHeiat dataset

5.3.1 Network Graph of analysis using HeiatHeiat dataset

Figure 5.7(a)Heiat Heiat Network Graph 1

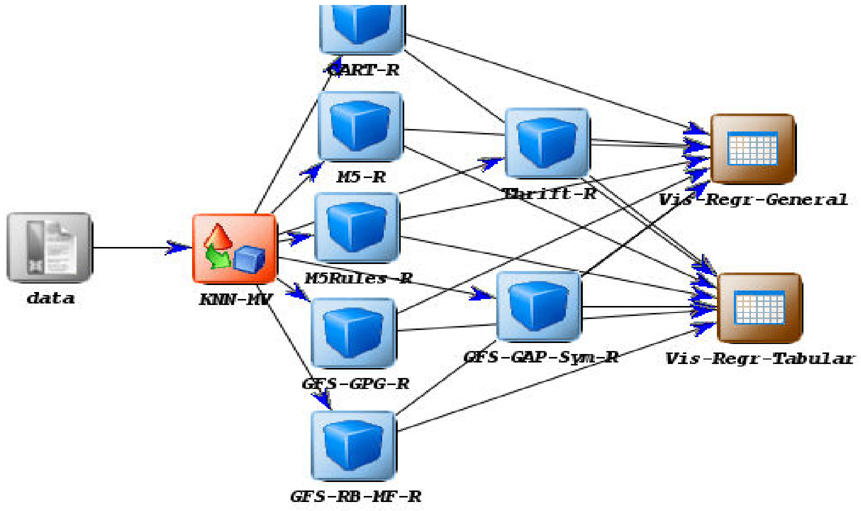


Figure 5.7(b)Heiat Heiat Network Graph 2

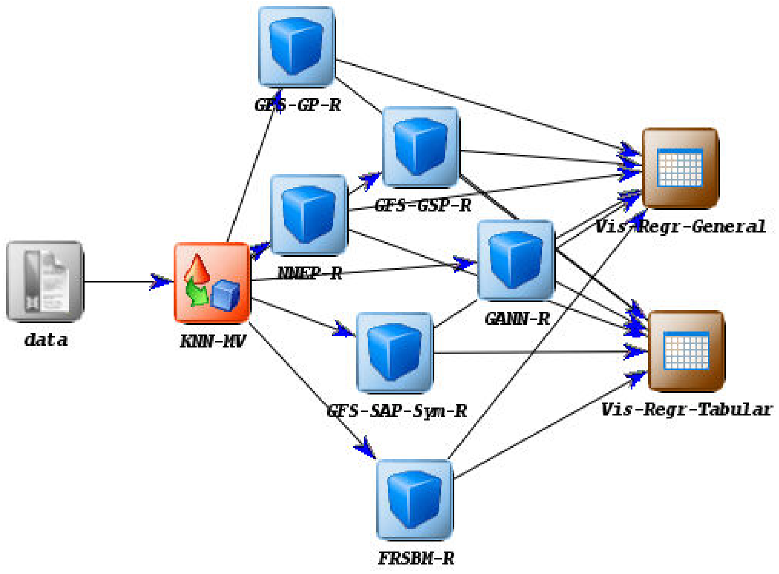


Figure 5.7(c) HeiatHeiat Network Graph 3

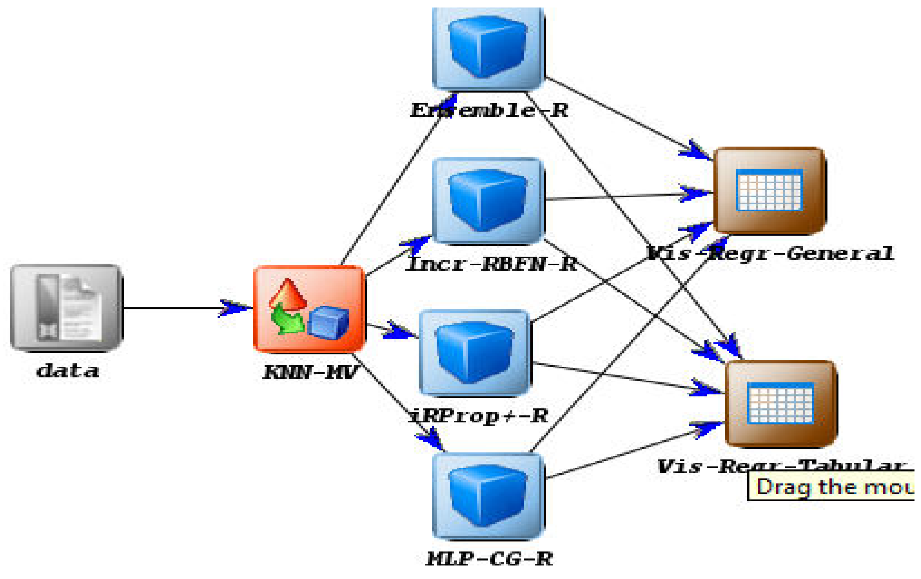


Figure 5.7(d) Heiat Heiat Network Graph 4

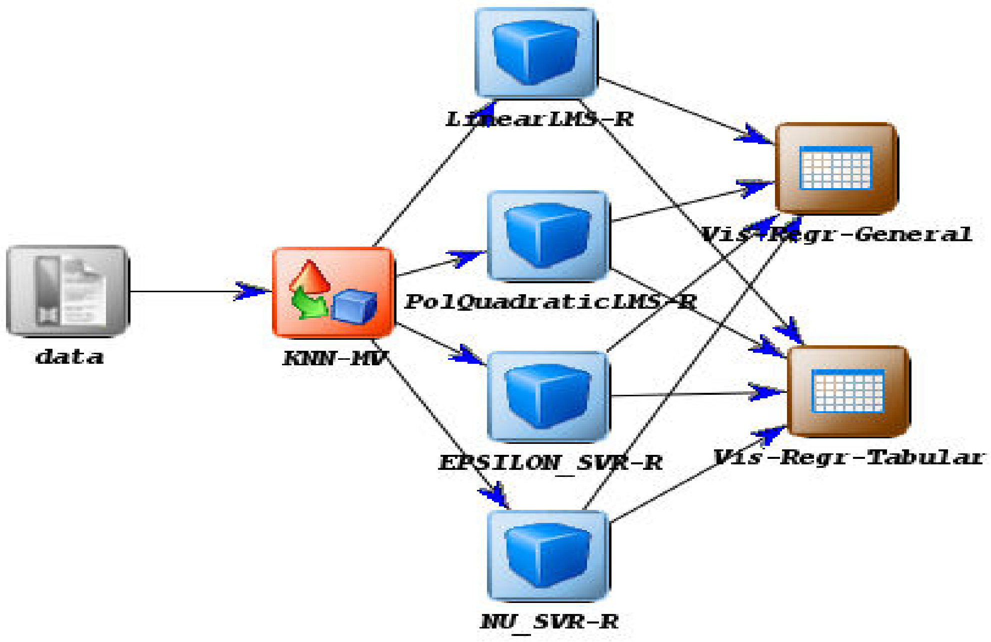


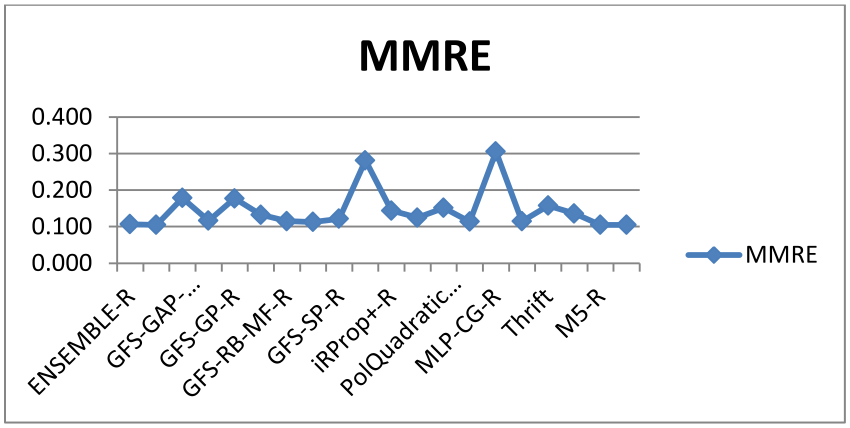
Table 5.3 Results with HeiatHeiat dataset

TECHNIQUE	MRE	MMRE	PRED(25)	PRED(50)	PRED(75)
ENSEMBLE-R	0.039	0.107	0.727	0.758	0.758
FRSBM	-0.031	0.105	0.909	0.970	0.970
GFS-GAP-Sym-R	-0.070	0.178	0.667	0.939	0.939
GFS-GPG-R	-0.024	0.116	0.879	0.939	0.939
GFS-GP-R	-0.061	0.177	0.606	0.939	0.939
GFS-GSP-R	-0.037	0.132	0.818	0.939	0.939
GFS-RB-MF-R	-0.006	0.115	0.818	0.939	0.939
GFS-SAPSym-R	-0.038	0.113	0.879	0.939	0.939
GFS-SP-R	-0.001	0.121	0.879	0.939	0.939
INCR-RBFN-R	-0.183	0.281	0.344	0.636	0.758
iRProp+-R	0.036	0.144	0.636	0.758	0.758
Linear-LMS-R	-0.053	0.124	0.848	0.970	0.970
PolQuadratic-LMS-R	-0.072	0.151	0.788	0.909	0.970
MLP-BP-R	0.032	0.114	0.727	0.758	0.758
MLP-CG-R	-0.189	0.305	0.303	0.576	0.758
NNEP-R	-0.024	0.115	0.939	0.970	0.970
Thrift	-0.105	0.157	0.758	0.939	0.939
CART-R	-0.022	0.136	0.848	0.970	0.970
M5-R	0.031	0.105	0.909	0.970	0.970
M5-Rules	0.308	0.105	0.909	0.970	0.970

5.3.1.Discussion of results with Heiat Heiat dataset

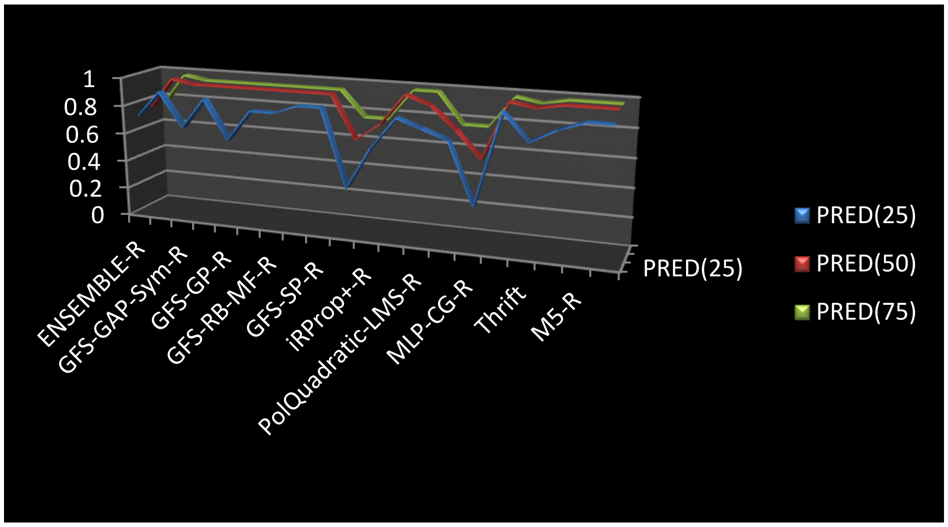
It can be seen from figure that ENSEMBLE-R, FRSBM and M5-R have shown comparable performance for MMRE value having MMRE values of 0.106,0.104 and 0.104 respectively.GFS-SAP-Sym-R is also having low MMRE value of 0.112.MLP-CG-R is worst method with MMRE value of 0.30.

Figure 5.8 Line Chart MMRE Vs Methods



NNEP-R has highest PRED(25) value of 0.939. After that FRSBM and M5 Rules have PRED(25) value of 0.909.In terms of PRED(50) and PRED(75) FRSBM,Linear LMS-R,CART-R,M5Rules outperforms all other methods with value of 0.969.

Figure 5.9 Line chart PRED Values Vs Methods



So overall FRSBM and M5 Rules are best methods for Heiat Heiat dataset in terms of MMRE value as well as PRED(25),PRED(50) and PRED(75) values.

5.4.Analysis of results with Desharnais dataset

5.4.1 Network Graph of analysis using Miyazaki94 dataset

Figure 5.10(a) Desharnais Network Graph 1

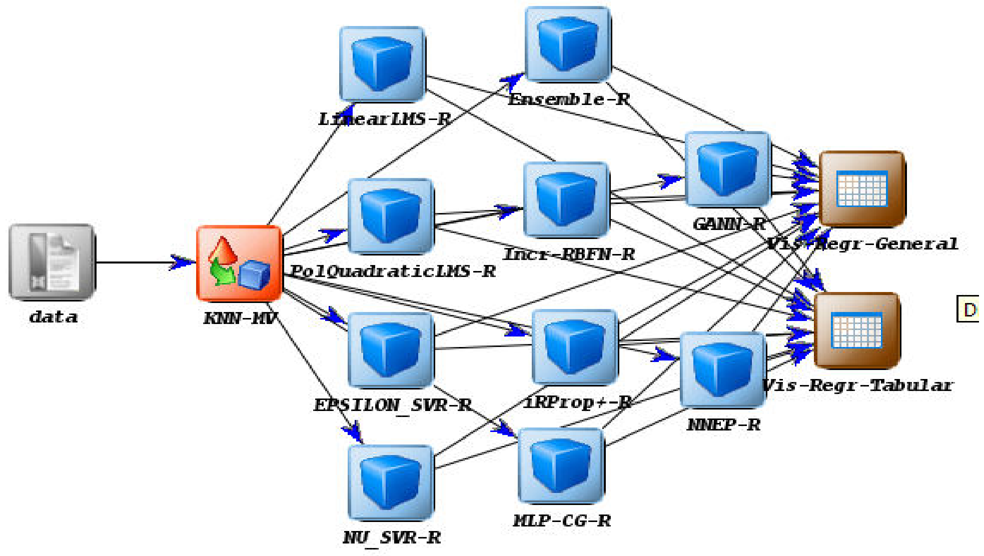


Figure 5.10(b)Desharnais Network Graph 2

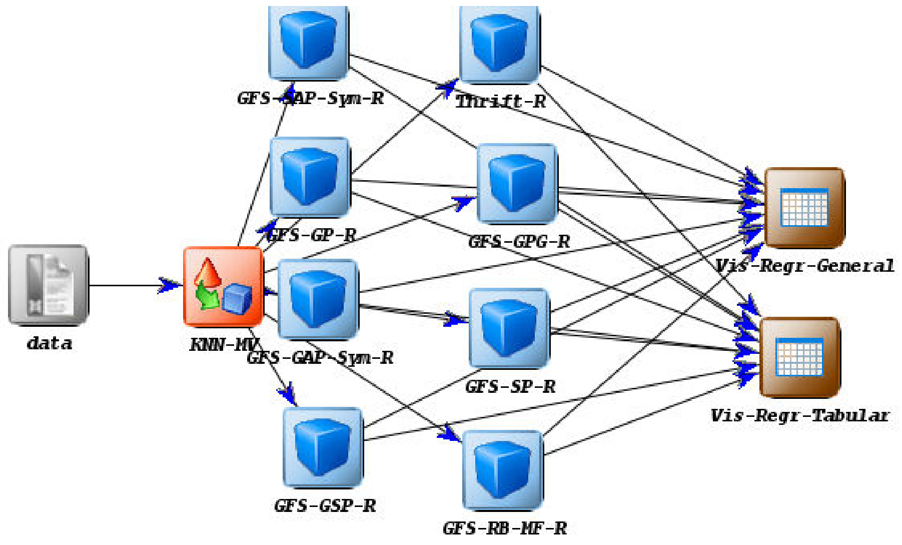


Figure 5.10(c)Desharnais Network Graph 3

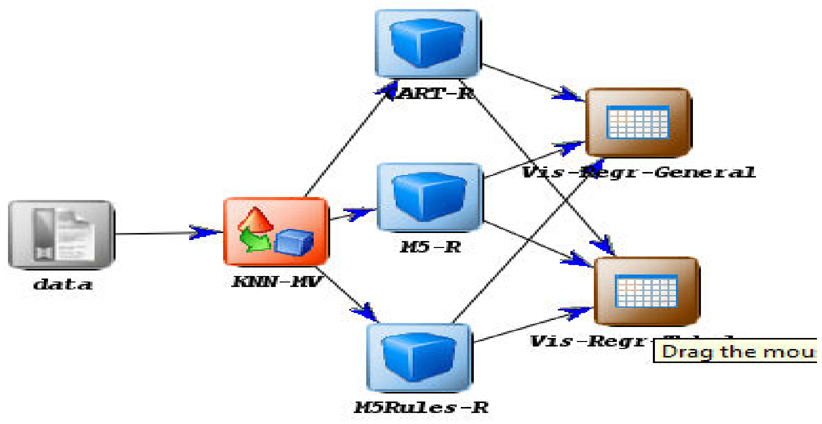


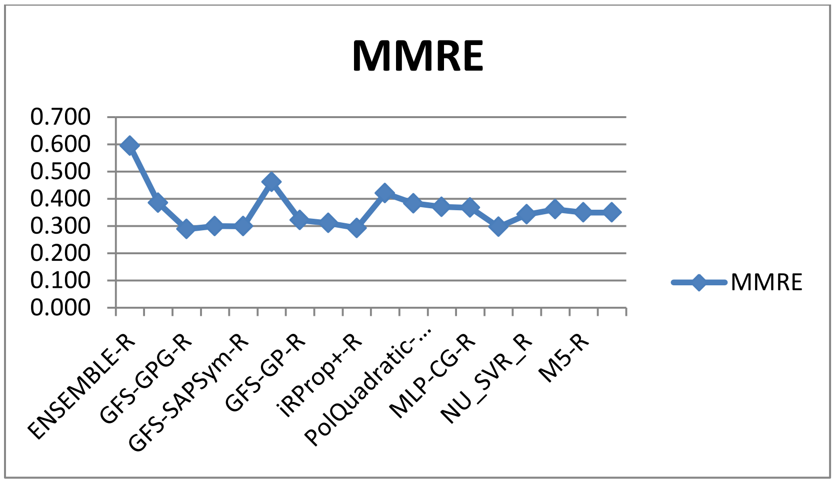
Table 5.4 Result with Desharnais dataset

TECHNIQUE	MRE	MMRE	PRED(25)	PRED(50)	PRED(75)
ENSEMBLE-R	-0.397	0.594	0.160	0.358	0.457
FRSBM	-0.051	0.385	0.284	0.593	0.741
GFS-GPG-R	0.037	0.288	0.346	0.617	0.753
GFS-GSP-R	0.023	0.299	0.333	0.617	0.753
GFS-SAPSym-R	0.023	0.299	0.333	0.617	0.753
GFS-SP-R	0.054	0.461	0.231	0.432	0.564
GFS-GP-R	-0.0431	0.321	0.2451	0.422	0.541
INCR-RBFN-R	0.020	0.311	0.259	0.543	0.667
iRProp+-R	-0.014	0.293	0.309	0.580	0.667
Linear-LMS-R	0.020	0.420	0.210	0.358	0.568
PolQuadratic-LMS-R	0.060	0.383	0.198	0.346	0.568
MLP-BP-R	-0.005	0.370	0.210	0.358	0.494
MLP-CG-R	0.096	0.367	0.321	0.654	0.556
NNEP-R	0.070	0.296	0.198	0.198	0.778
NU_SVR_R	0.157	0.343	0.210	0.358	0.593
CART-R	0.082	0.361	0.259	0.580	0.765
M5-R	-0.040	0.349	0.383	0.642	0.753
M5-Rules	-0.040	0.349	0.383	0.642	0.753

5.4.1.Analysis of results with Desharnais dataset

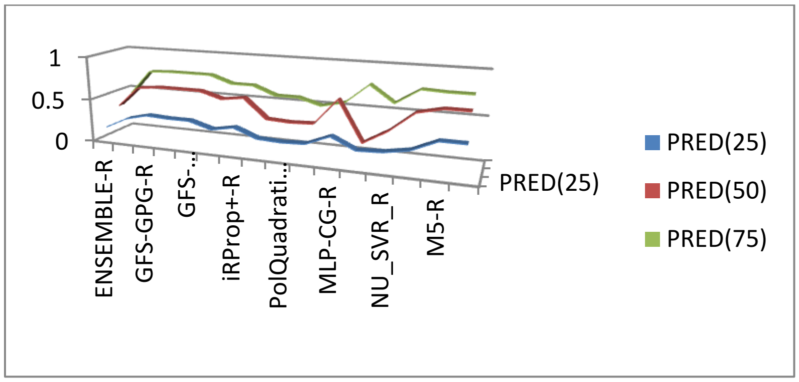
GFS-GPG-R has shown best performance in terms of MMRE value with MMRE value of 0.288.After that GFS-GSP-R,GFS-SAP-Sym-R,iRProp+-R and NNEP-R have almost same MMRE value of 0.299,0.298,0.292 and 0.296 respectively.ENSEMBLE-R is worst method in terms of MMRE with value of 0.593.

Figure 5.11 Line chart MMRE Vs Methods



M5-Rules are best in terms of PRED(25) value with value of 0.38.MLP-CG-R is best in terms of PRED(50) value with 0.654 value.In terms of PRED(75) , NNEP-R outperforms all other methods with value of 0.777.

Figure 5.12 Line Chart PRED Values Vs Methods



5.5.Analysis of results with China dataset

5.5.1 Network Graph of analysis using Miyazaki94 dataset

Figure 5.13(a) China Network Graph 1

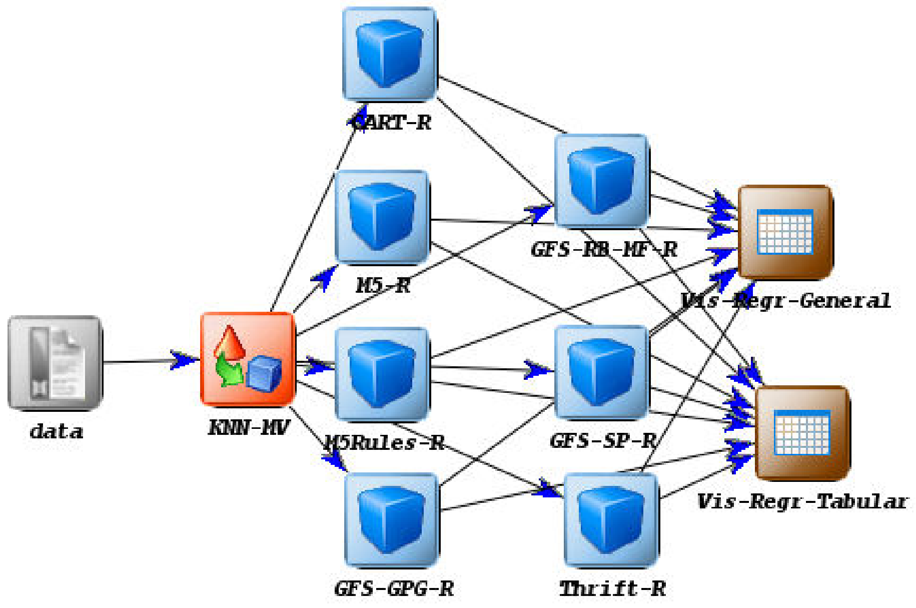


Figure 5.13(b)China Network Graph 2

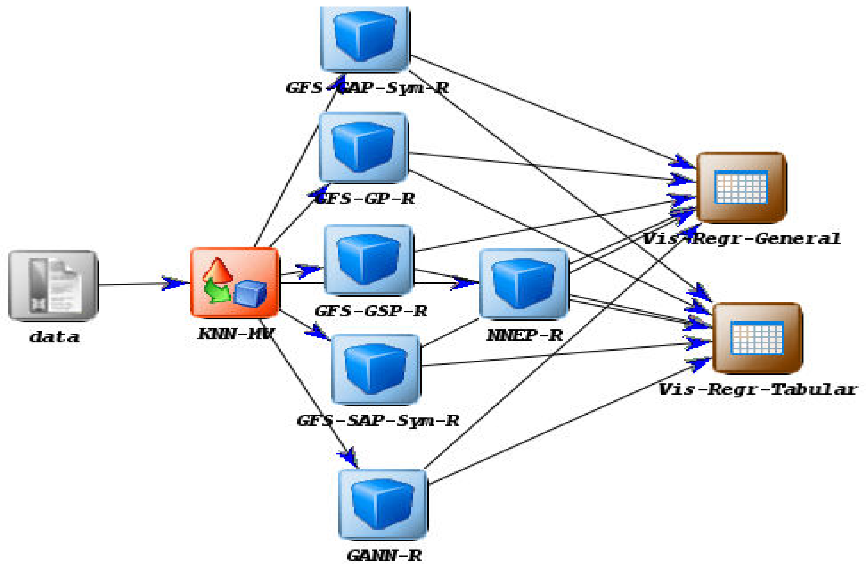


Figure 5.13(c)China Network Graph 3

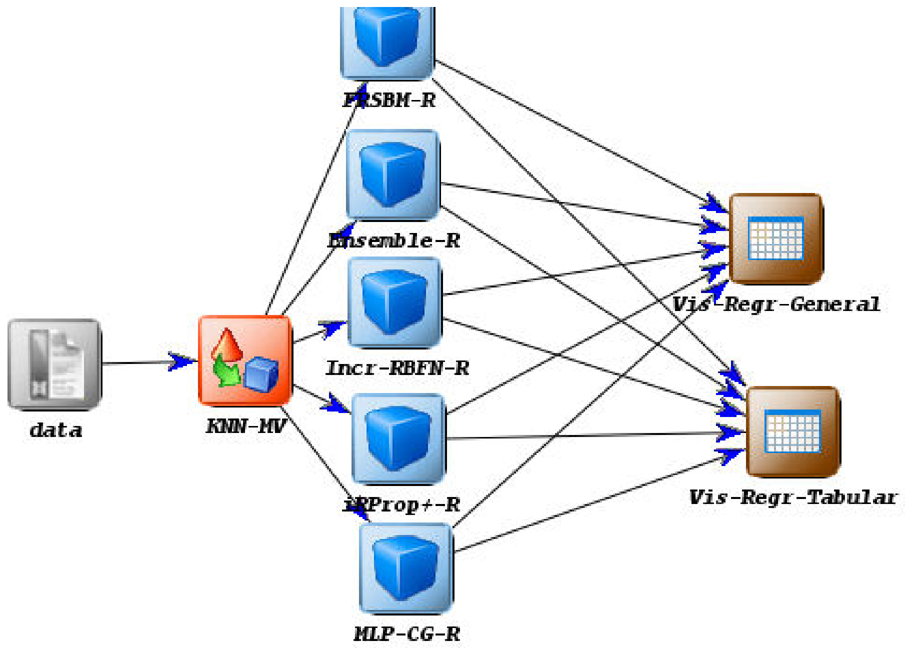


Figure 5.13(d)China Network Graph 4

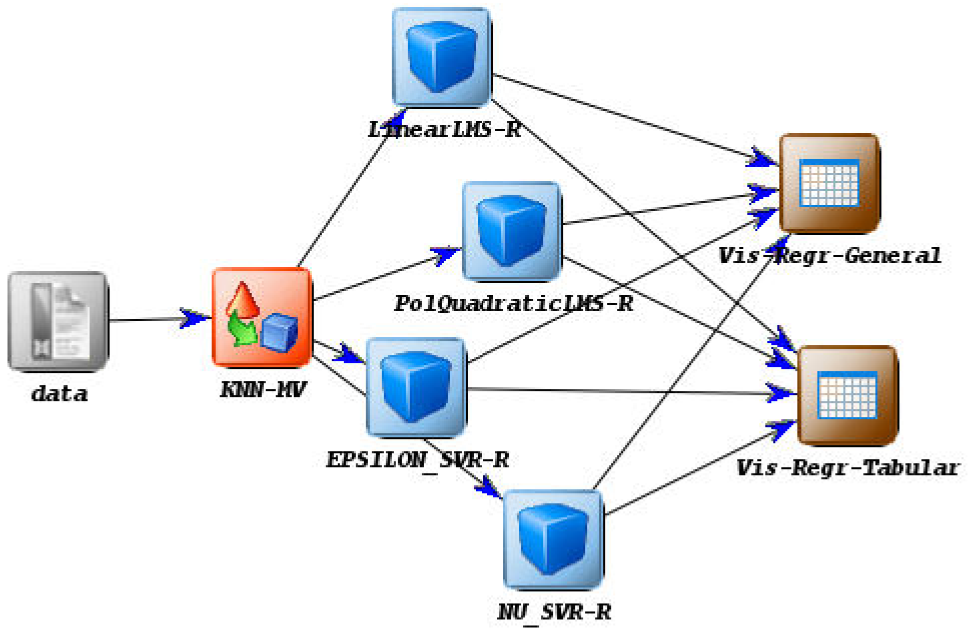


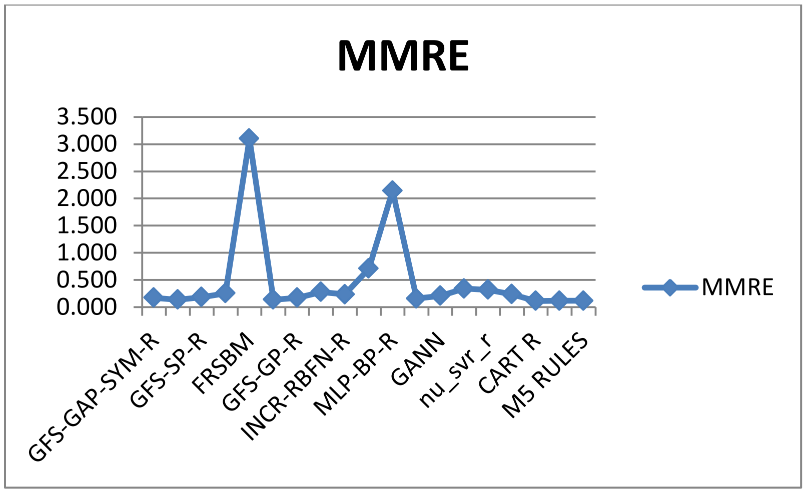
Table 5.5 Result with China dataset

TECHNIQUE	MRE	MMRE	PRED(25)	PRED(50)	PRED(75)
GFS-GAP-SYM-R	0.059	0.173	0.828	0.927	0.962
GFS-SAP-SYM-R	-0.045	0.139	0.898	0.963	0.973
GFS-SP-R	-0.083	0.185	0.885	0.939	0.958
ENSEMBLE-R	0.093	0.257	0.563	0.789	0.887
FRSBM	-2.855	3.100	0.152	0.274	0.423
GFS-GSP-R	-0.043	0.137	0.901	0.965	0.975
GFS-GP-R	0.059	0.173	0.827	0.927	0.962
GFS-GPG-R	-0.181	0.279	0.801	0.909	0.945
INCR-RBFN-R	0.029	0.235	0.710	0.886	0.946
IRPROP+-R	-0.064	0.710	0.276	0.531	0.722
MLP-BP-R	0.740	2.143	0.116	0.206	0.347
MLP-CG-R	-0.086	0.160	0.897	0.952	0.971
GANN	0.065	0.209	0.341	0.536	0.878
Thrift	0.054	0.341	0.434	0.617	0.823
nu_svr_r	-0.021	0.321	0.511	0.758	0.900
NNEP	-0.023	0.238	0.758	0.885	0.951
CART R	-0.024	0.115	0.956	0.978	0.992
M5R	-0.021	0.118	0.946	0.982	0.986
M5 RULES	-0.021	0.118	0.946	0.982	0.986

5.5.1.Discussion of results with China dataset

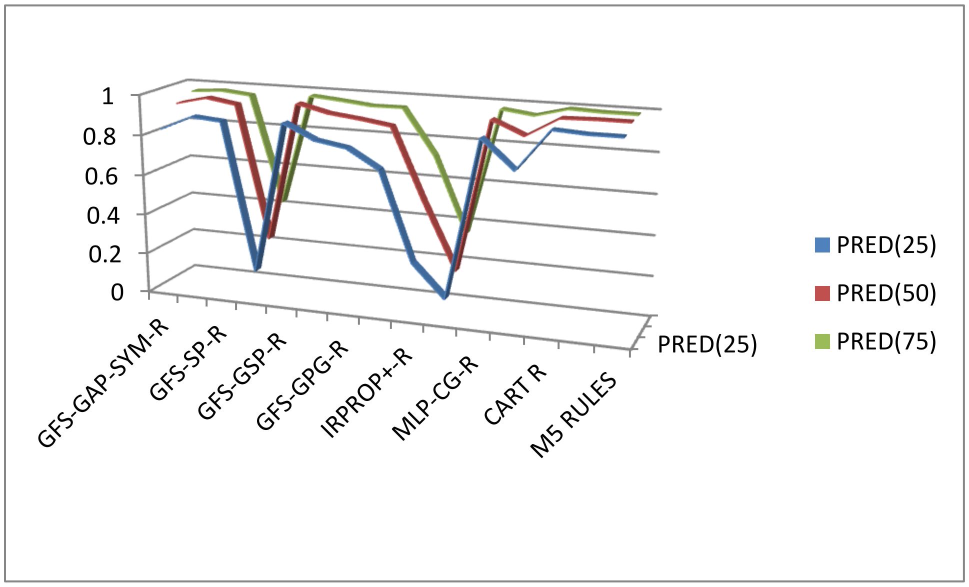
CART-R is best method in terms of MMRE value.MMRE value for CART-R is 0.114.After that M5-R ,GFS-GSP-R,GFS-SAP-Sym-R have also shown good performance with MMRE values of 0.117,0.137 and 0.139 respectively.FRSBM is worst method with highest MMRE value.

Figure 5.14 Line chart MMRE Vs Methods



CART-R is best method in terms of PRED(25),PRED(50) and PRED(75) value .The values are 0.95,0.97 and 0.99 for PRED(25),PRED(50) and PRED(75) respectively.M5-R and M5 Rules also have good PRED(25),PRED(50) and PRED(75) value.The respective values for these two are 0.94,0.981 and 0.985 respectively.

Figure 5.15 Line chart PRED Values Vs Methods



5.6.Analysis of results with Maxwell dataset

5.6.1 Network Graph of analysis using Maxwell dataset

Figure 5.16(a) Maxwell Network Graph 1

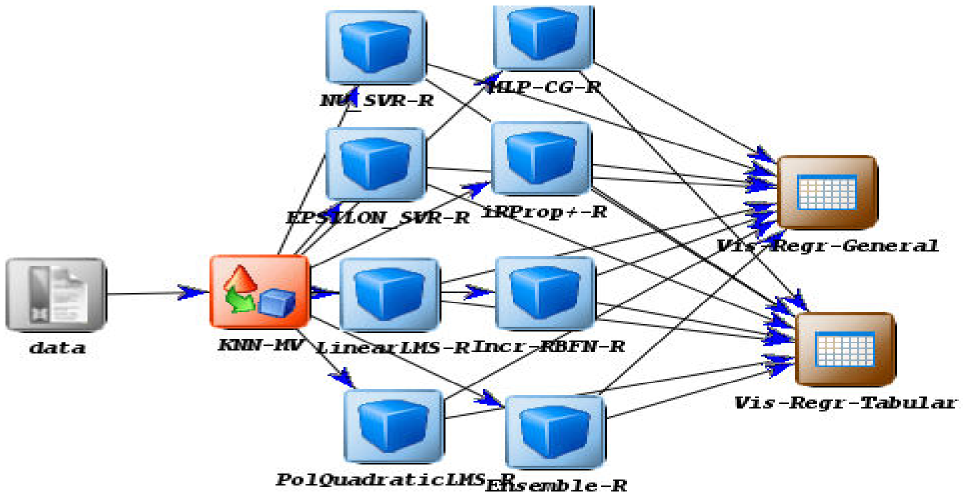


Figure 5.16(b)Maxwell Network Graph 2

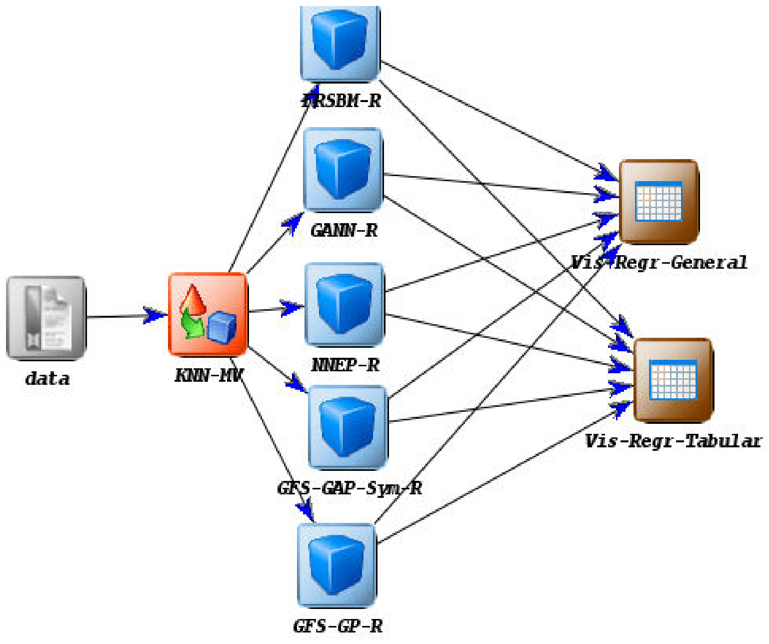


Figure 5.16(c) Maxwell Network Graph 3

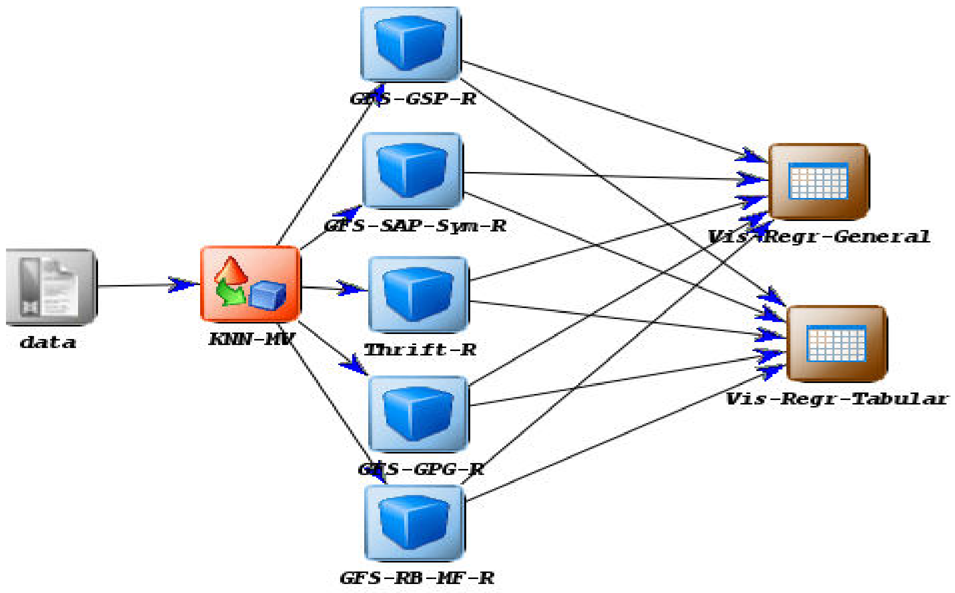


Figure 5.16(d)Maxwell Network Graph 4

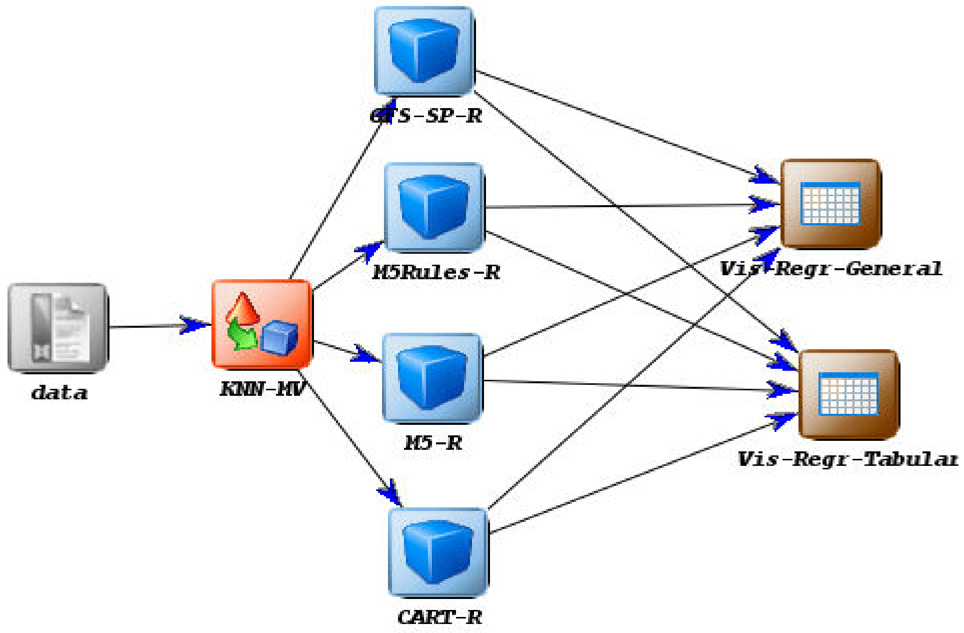


Table 5.6 Result with Maxwell dataset

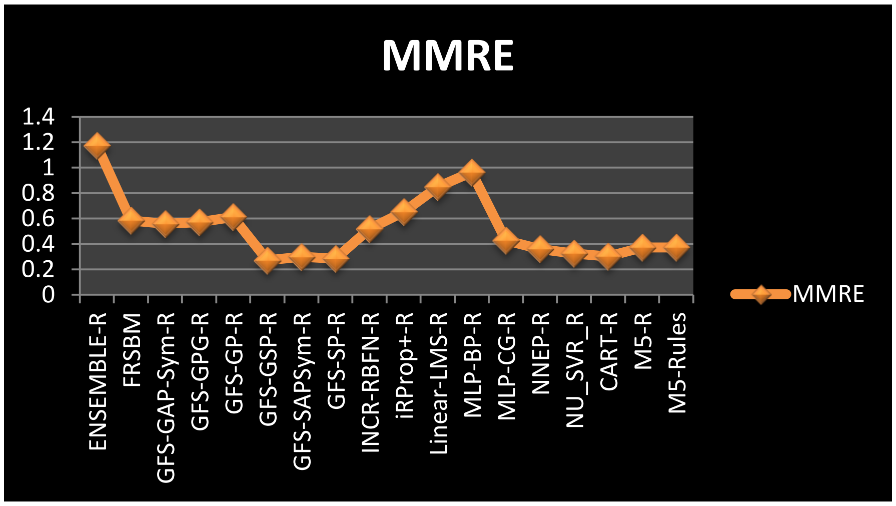
TECHNIQUE	MRE	MMRE	PRED(25)	PRED(50)	PRED(75)

ENSEMBLE-R	-0.939	1.174	0.129	0.194	0.274
FRSBM	-0.256	0.583	0.258	0.387	0.484
GFS-GAP-Sym-R	-0.179	0.558	0.242	0.371	0.581
GFS-GPG-R	0.379	0.573	0.113	0.242	0.452
GFS-GP-R	-0.267	0.611	0.242	0.371	0.581
GFS-GSP-R	0.135	0.273	0.306	0.629	0.677
GFS-SAPSym-R	0.130	0.299	0.306	0.629	0.677
GFS-SP-R	0.099	0.285	0.306	0.532	0.694
INCR-RBFN-R	-0.025	0.520	0.177	0.371	0.661
iRProp+-R	0.029	0.653	0.258	0.371	0.516
Linear-LMS-R	0.221	0.849	0.226	0.355	0.452
MLP-BP-R	-0.626	0.966	0.113	0.226	0.339
MLP-CG-R	-0.022	0.427	0.161	0.500	0.629
NNEP-R	0.021	0.361	0.226	0.355	0.532
NU_SVR_R	0.108	0.325	0.242	0.484	0.581
Thrift	0.231	0.313	0.257	0.506	0.623
CART-R	0.140	0.302	0.274	0.468	0.710
M5-R	0.084	0.372	0.323	0.484	0.661
M5-Rules	0.084	0.372	0.323	0.484	0.661

5.6.1.Discussion of results with Maxwell dataset

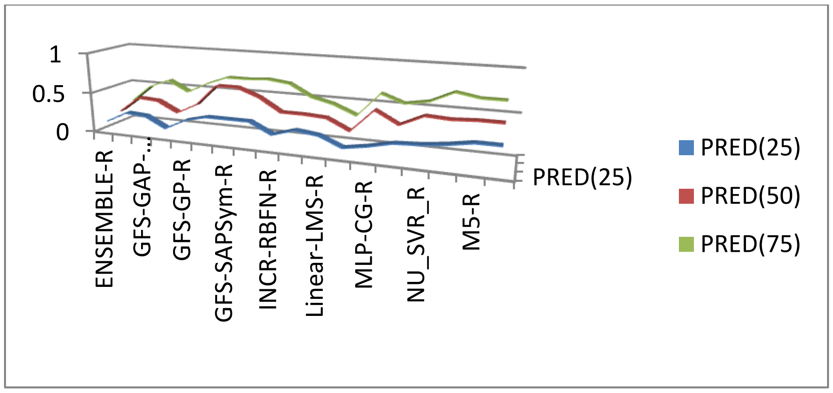
GFS-GSP-R is best method in terms of MMRE value having lowest MMRE value of 0.272.After that GFS-SP-R and GFS-SAP-Sym-R have also shown good performance with MMRE values of 0.284 and 0.299 respectively.ENSEMBLE-R is the worst method in terms of MMRE value having highest MMRE.

Figure 5.17 Line chart MMRE Vs Methods



GFS-GSP-R and GFS-SAP-Sym-R have comparable performance in terms of PRED(25),PRED(50) and PRED(75). The values of PRED(25),PRED(50) and PRED(75) for both methods are 0.306,0.629 and 0.677 respectively.GFS-SP-R also have good performance in terms of PRED(25),PRED(50) and PRED(75) ,values being 0.302,0.532 and 0.693 respectively.

Figure 5.18 Line Chart PRED values Vs Methods



Chapter 6: Conclusion and Future Work

On the basis of our entire study, we can conclude that evolutionary algorithms give better results as compared to traditional methods of software effort estimation.There are so many factors which affect the results of software effort estimation like software complexity, computer platform,fragmentation etc.As there are many evolutionary algorithms for software effort prediction,its difficult to say which evolutionary technique is better than the other one.Thresults of our research have been summarized in Table 6.1.

Table 6.1 Table describing best,good and worst methods for different datasets

Dataset	Size	Best Method	Good Method	Worst Method
Desharnais	Large	GFS-GPG-R	GFS-GSP-R,GFS-SAP-Sym-R,iRProp+-R and NNEP-R	ENSEMBLE-R
Maxwell	Medium	GFS-GSP-R	GFS-SP-R,GFS-SAP-Sym-R,	ENSEMBLE-R
China	Very Large	CART-R	M5-R,GFS-GSP-R,GFS-SAP-Sym-R	FRSBM
NASA18	Very Small	FRSBM,Linear-LMS-R	NNEP,MLP-CG-R	ENSEMBLE-R
Heiat Heiat	Small	ENSEMBLE-R	FRSBM,M5-R	MLP-CG-R
Miyazaki94	Small	GFS-SAP-Sym-R	MLP-CG-R,GFS-GSP-R,PolQuadratic LMS-R	Thrift

After analyzing the results with different datasets and using different evolutionary algorithms on the datasets we can conclude different datasets show different results with different algorithms. The performance of a particular evolutionary algorithm depends on the type of data on which our evolutionary algorithm is trained.As per our studies, GFS-SAP-Sym-R is a good method for dataset of any size(very small,small,medium,large,very large).Out of 6 datasets,ENSEMBLE-R has shown worst performance for 3 datasets.

6.1 Applications of software effort estimation using evolutionary algorithms

- We can make correct estimations for the budget of project if we know which evolutionary technique should be applied under various circumstaces.So according to our research if the dataset is of very small,small,medium,large or very large size ,selection of evolutionary technique can be done accordingly.

- We can estimate the manpower required for project completion and higher correct number of people.

- Selection of correct software effort estimation technique can ensure timely completion of software projects.

- Quality of software projects improve if software estimation technique is selected wiselyss

6.2 Future Work

In the future, this empirical study can be done for some industrial software.Also we can apply other evolutionary algorithms like Bacterial foraging and particle swarm optimization on the same datasets and see if there is any performance in results or not.This study can be repeated again and again for other new techniques untill we find a unique technique which can give correct estimations for all types of datasets.

References

[1]Y. Singh, P.K. Bhatia, A. Kaur, and O. Sangwan, “A Review of Studies on Effort Estimation Techniques of Software Development,” *in Proc. Conference Mathematical Techniques: Emerging Paradigms for Electronics and IT Industries*, New Delhi, 2008, pp. 188-196.

[2]B. Boehm, C. Abts and S. Chulani, Software development cost estimation approaches - A survey, *Annal of Software Engineering*, vol. 10, pp. 177-205, 2000.

[3]C. Jones, Estimating Software Costs: Bringing Realism to Estimating. McGraw-Hill, 2007.

[4]S. McConnell, Software Estimation: Demystifying the Black Art. Redmond,Washington: Microsoft Press, 2006.

[5]S. Amasaki, “A Study on Performance Inconsistency between Estimation by Analogy and Linear Regression,” no. 25.

[6]U. Keel, M. Graczyk, T. Lasota, and B. Trawiński, “Comparative Analysis of Premises Valuation Models.”

[7] L.C. Briand and I. Wieczorek,” Resource Estimation in Software Engineering,”*Encyclopedia of Software Eng*., J.J. Marcinak, ed., pp. 1160-1196, John Wiley & Sons, 2002.

[8] J. Li, G. Ruhe, A. Al-Emran and M. M. Richter,” A Flexible Method for Software Effort Estimation by Analogy,” *Empirical Software Engineering*, vol. 12, pp. 65-106, 2006.

[9] L. M. Laird and M. C. Brennan, Eds., Software Measurement and Estimation: A Practical Approach. Hoboken, New Jersey: Jonh Wiley & Sons, Inc., 2006.

[10] M. Shepperd, C. Schofield and B. Kitchenham, Effort Estimation Using Analogy, in Proc. International Conference on Software Engineering Berlin, 1996, pp. 170-178.

[11] H. Leung and Z. Fan, Software Cost Estimation, in Handbook of Software Engineering and Knowledge Engineering, Hong Kong Polytechnic University, 2002.

[12] I. Attarzadeh and S. H. Ow, Software Development Effort Estimation Based on a New Fuzzy Logic Model, International Journal of Computer Theory and Engineering, vol. 1, pp. 1793-8201, 2009.

[13] J. Kaur, S. Singh and K. S. Kahlon, Comparative Analysis of the Software Effort Estimation Models, in Proc. World Academy of Science, Engineering and Technology, Auckland, New Zealand, 2008, pp. 485-487.

[14] L. M. Laird and M. C. Brennan, "Software Measurement and Estimation: A Practical Approach," Hoboken, New Jersey: Jonh Wiley & Sons, Inc., 2006.

[15]R. Schoedel," Proxy Based Estimation (PROBE) for Structured Query Language (SQL), "Carnegie Mellon University, 2006.

[16]M. Jorgensen, "A review of studies on expert estimation of software development effort, "The Journal of Systems and Software, vol. 70, pp. 37-60, 2004.

[17] M. Jordensen, G. Kirkeboen, D. I. K. Sjoberg, B. Anda and L. Bratthall, "Human Judgement in Effort Estimation of Software Projects, in Proc. International Conference on Software Engineering, Limerick, Ireland. 2000.

[18] J. Hill, L. C. Thomas and D. E. Allen, "Experts estimates of task durations in software development projects," International Journal of Project Management, vol. 18, pp. 13-21, 2000.

[19] R.Malhotra, A.Kaur and Y.singh, "Application of Machine Learning Methods for Software Effort Prediction,"in Newsletter ACM SIGSOFT Software Engineering Notes , vol. 35, May 2010.

[20] R. Malhotra, A.Jain , "Software Effort Prediction using Statistical and Machine Learning Methods," vol. 2, no. 1, pp. 145–152, 2011.

[21] G. R. Finnie, G. E. Wittig, and J.-M. Desharnais, "A comparison of software effort estimation techniques: Using function points with neural networks, case-based reasoning and regression models," *J. Syst. Softw.*, vol. 39, no. 3, pp. 281–289, Dec. 1997.

[22] M. O. Elish, "Improved estimation of software project effort using multiple additive regression trees," *Expert Syst. Appl.*, vol. 36, no. 7, pp. 10774–10778, Sep. 2009.

[23] C. J. Burgess and M. Le, "Can genetic programming improve software effort estimation ? A comparative evaluation," vol. 43, 2001.

[24] Albrecht and J. E. Gaffney, "Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation," *IEEE Trans. Softw. Eng.*, vol. SE-9, no. 6, pp. 639–648, Nov. 1983.

[25] Z. Dan, "Improving the accuracy in software effort estimation: Using artificial neural network model based on particle swarm optimization," *Proc. 2013 IEEE Int. Conf. Serv. Oper. Logist. Informatics*, pp. 180–185, Jul. 2013.

[26] U.Saxena, S.P.Singh, "Software Effort Estimation Using Neuro-Fuzzy Approach", *CSI 6th International Conference*, pp.1-6, Sept.2012.

[27] A. Dhiman and C. Diwaker, "Optimization of COCOMO II Effort Estimation using Genetic Algorithm," pp. 208–212, 2013.

[28] A. B. Nassif, M. Azzeh, L. F. Capretz, and D. Ho, "A comparison between decision trees and decision tree forest models for software development effort estimation," *2013 Third Int. Conf. Commun. Inf. Technol.*, pp. 220–224, Jun. 2013.

[29] Y.-S. Seo, K.-A. Yoon, and D.-H. Bae, "Improving the Accuracy of Software Effort Estimation Based on Multiple Least Square Regression Models by Estimation Error-Based Data Partitioning," *2009 16th Asia-Pacific Softw. Eng. Conf.*, pp. 3–10, Dec. 2009.

[30] H. Hamza, A. Kamel, and K. Shams, “Software Effort Estimation Using Artificial Neural Networks: A Survey of the Current Practices,” *2013 10th Int. Conf. Inf. Technol. New Gener.*, pp. 731–733, Apr. 2013.

[31] C. Science and S. Engineering, “Prediction of Software Effort Using Artificial Neural Network and Support Vector Machine,” vol. 3, no. 3, pp. 40–46, 2013.

[32] P. C. Pendharkar, G. H. Subramanian, and J. A. Rodger, “A Probabilistic Model for Predicting Software Development Effort,” vol. 31, no. 7, pp. 615–624, 2005.

[33] C. L. Martín, “Software Development Effort Estimation Using Fuzzy Logic : A Case Study,” 2005.

[34] Â. C. Riquelme, M. Toro, Â. S. Aguilar-ruiz, and I. Ramos, “An evolutionary approach to estimating software development projects q,” vol. 43, pp. 875–882, 2001.

[35] A. Sheta,”Software effort estimation and stock market prediction using takagi-sugeno fuzzy models," *in Fuzzy Systems, 2006 IEEE International Conference* , pp. 171-178, IEEE, 2006.

[36] B.Baskeles,” Software Effort Estimation Using Machine Learning Methods,” *IEEE 22nd Symposium on Computer and Information Sciences*, pp. 1-6,2007.

[37] L. Sánchez, “A random sets-based method for identifying fuzzy models,” *Fuzzy Sets Syst.*, vol. 98, no. 3, pp. 343–354, Sep. 1998.

[38] A. Thesis, “Combining Genetic Algorithms and Neural Networks : The Encoding Problem,” no. December, pp. 1–67, 1994.

[39] D. B. Fogel and L. J. Fogel, “Evolutionary Programming for Training Neural Networks David B. Fogel Lawrence J. Fogel,” pp. 601–605.

[40] S. Garc, “Evolving Fuzzy Rule Based Classifiers with GA-P : A Grammatical Approach,” pp. 203–210, 1999.

[41] S. Luciano, “Combining GP operators with SA search to evolve fuzzy rule based classifiers,” vol. 136, 2001.

[42] http://en.wikipedia.org/wiki/Least_squares

[43] P. Thrift, “Fuzzy logic synthesis with genetic algorithms”, *Proceedings of 4th International Conference on Genetic Algorithms (ICGA'91)*, pp 509–513.

[44] G.W. Flake, S. Lawrence, “Efficient SVM regression training with SMO”. *Machine Learning*, 46 (1–3) (2002) 271–290.

[45] A. J. Smola, B. Scholkopf, “A tutorial on support vector regression”. *Statistics and Computing*, vol. 14, no. 3, pp.199–222, 2004.

[46] A. L. I. Oliveira, “Estimation of software projects effort with support vector regression”. *Neurocomputing*, vol. 69, no. 13-15, pp. 1749-1753, August 2006.

[47] S. Haykin, *Neural Networks: a Comprehensive Foundation*. Prentice Hall, New Jersey, USA, 1999.

[48] Promise. Available: http://promisedata.org/repository/.

[49] S. N. Fatma, “Genetic Approach for Fuzzy Mining Using Modified K-Means Clustering,” vol. 2, no. 6, pp. 261–267, 2012.

[50] http://www.nnwj.de/uploads/pics/1_2-backpropagation-net.gif

[51] http://annet.eeng.nuim.ie/intro/course/images/radial_1.gif

[52]http://cms.horus.be/files/99936/MediaArchive/pictures/Artificial_Neural_Networks_Ensemble.jpg

[53] http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm

[54] A. B. Nassif, L. F. Capretz, and R. Hill, "Analyzing the Non-Functional Requirements in the Desharnais Dataset for Software Effort Estimation," no. August, 2012.

[55] K. Maxwell, "Applied statistics for software managers," Englewood Cliffs, NJ: Prentice-Hall, 2002.

[56]KEEL.Available at http://www.keel.es/

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

[58] K. Srinivasan and D. Fisher, "Estimating Software Development Effort," vol. 21, no. 2, 1995.