EFFORT ESTIMATION FOR SOFTWARE PROJECTS USING FUNCTIONAL APPROXIMATION AND NON LINEAR REGRESSION IN NEURAL NETWORK

A Dissertation Submitted

By

Tanvi Arora

11108659

To

Department of Computer Science & Engineering

In Partial Fulfillment of the Requirement for the

Award of the Degree of

Master of Technology in Computer Science & Engineering

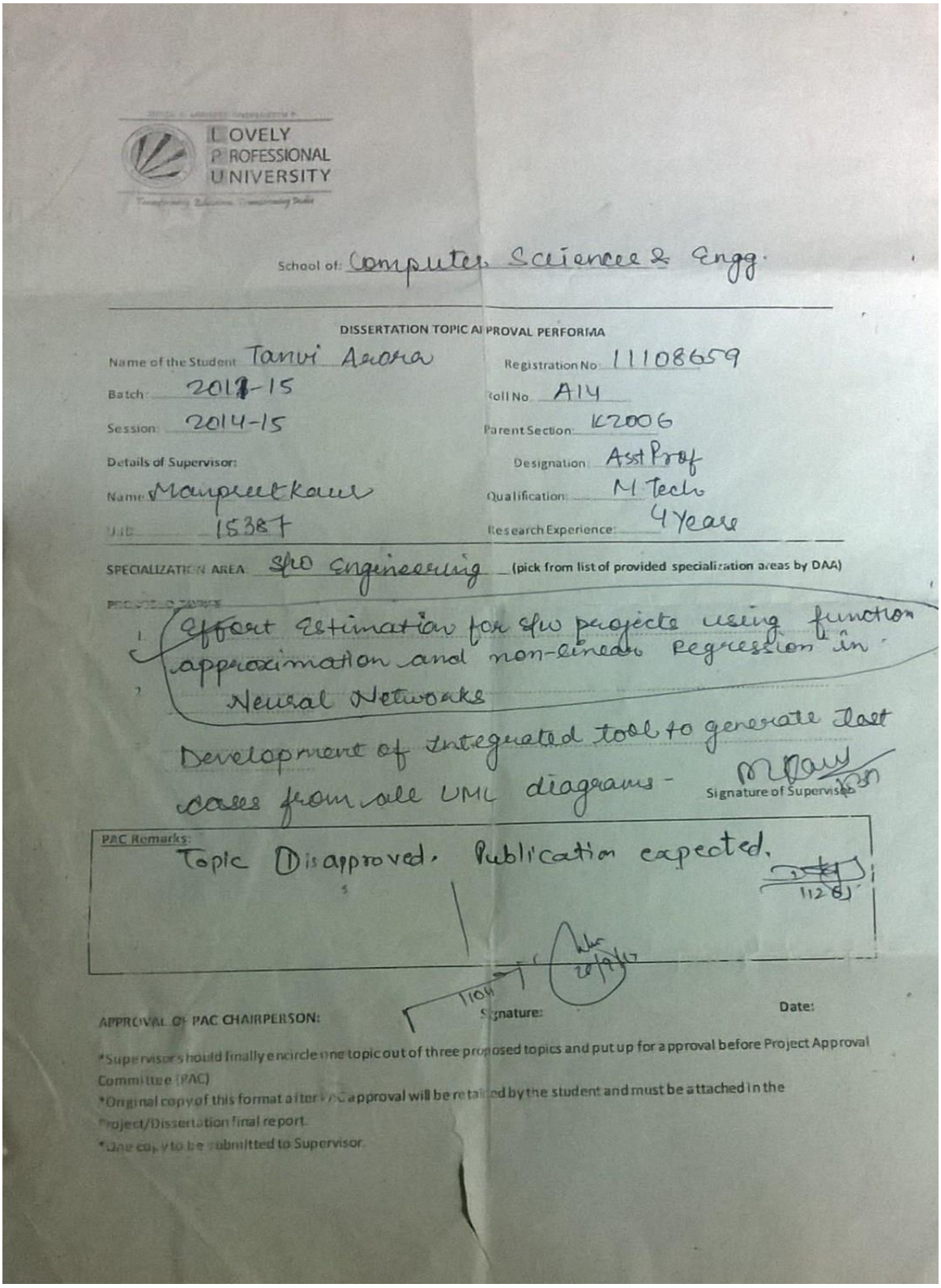
Under the Guidance of

Ms. Pooja Devi

(Assistant Professor)

(May, 2015)

APPROVED RESEARCH TOPIC



LOVELY PROFESSIONAL UNIVERSITY

School of: Computer Sciencee & Engg.

DISSERTATION TOPIC APPROVAL PERFORMA

Name of the Student: Tanvi Arora — Registration No: 11108659

Batch: 2013-15 — Roll No: A14

Session: 2014-15 — Parent Section: K2006

Details of Supervisor:

Name: Manpreet Kaur — Designation: Asst Prof

UID: 15387 — Qualification: M Tech

Research Experience: 4 Years

SPECIALIZATION AREA: SDO Engineering (pick from list of provided specialization areas by DAA)

PROJECT TOPICS

1. Effort Estimation for SDO projects using function approximation and non-linear regression in Neural Networks

Development of integrated tool to generate test cases from all UML diagrams -

Signature of Supervisor

PAC Remarks: Topic 1 is approved. Publication expected.

APPROVAL OF PAC CHAIRPERSON: Signature: Date: 20/9/14

*Supervisor should finally encircle one topic out of three proposed topics and put up for approval before Project Approval Committee (PAC)

*Original copy of this format after PAC approval will be retained by the student and must be attached in the Project/Dissertation final report.

*One copy to be submitted to Supervisor.

ABSTRACT

Effort estimation is the process of estimating the most realistic amount of effort required to develop the software which is based on incomplete, uncertain or noisy input. Here in this report, a literature survey is being done related to effort estimation. With this, knowledge is being gained regarding estimations i.e. the models in which estimation was already performed; there analysis was done so that comparisons can be done accordingly. For assessing the accuracy, evaluation is done by using MMRE and PRED. Various estimation techniques are being discussed. Among those techniques, neural network is selected as one in which estimating the effort will be performed by using functional approximation. The aim of this whole process is to achieve nearby actual value of estimations. It will then be helpful for project managers and experts to spend less time on this task and more time to some more important issues.

CERTIFICATE BY THE ADVISOR

This is to certify that Tanvi Arora Reg. Number **11108659** has completed M Tech. dissertation **"Effort Estimation for Software Projects using Functional Approximation and Non Linear Regression in Neural Network"** under my guidance and supervision. To the best of my knowledge, the present work is the result of her original investigation and study. No part of the dissertation proposal has ever been submitted for any other degree or diploma.

The Dissertation is fit for the submission and the partial fulfillment of the conditions for the award of M. Tech Computer Science and Engineering.

Date: 3rd May, 2015

Name: Pooja Devi
Assistant Professor
UID: 17778

ACKNOWLEDGEMENT

First and foremost I would like to thank Almighty for giving me courage to bring up this dissertation. Before getting into thick and thin of this dissertation I would like to show my gratitude to some of the people who have helped me in this project. Firstly I would like to propose a word thanks to my mentor MS. POOJA DEVI who has encouraged me to get through this dissertation. Secondly I would like to thanks my friends who gave me unending support and helped me in numerous ways from the stage when the idea of the thesis was conceived. I am very thankful to all of them for making my work complete successfully under their guidance.

Tanvi Arora ……

DECLARATION

I hereby declare that the dissertation entitled, **"Effort Estimation for Software Projects using Functional Approximation and Non Linear Regression in Neural Network"** submitted for the M.Tech Degree is entirely my original work and all ideas and references have been duly acknowledged. It does not contain any work for the award of any other degree or diploma.

Date: **3rd May, 2015**

Tanvi Arora

11108659

TABLE OF CONTENTS

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

In today's time, Software is called to be the most extortionate component in many projects of computer system because its major part of cost is due to human effort. Software development life cycle contains effort estimation of software as one of the process in planning stage which is used for predicting the software effort so that the required costs are being estimated. For software project management, accuracy in software project cost is important. Proper software cost estimates are essential for both among the development team and customers. Underestimation in the cost may exceed the budgets and further results in failure for completing the task on time. Overestimation may result in commitment of excess resources in the project, if which can't be fulfilled will result in loss of several jobs. So, accuracy in each estimation is important. [9]

1.1 Effort Estimation

An effort is simply the involvement of work which is being performed to complete some small or large tasks. Effort estimation is a process used for analyzing or approximating an amount of effort that will be required for the development and maintenance of a product. We can then use these estimates in several ways i.e. For giving input to some project plans, For analyzing investments being done earlier or some kind of bidding processes. These estimates are also useful for the development of well organized software. Many techniques and models are being used for estimating the effort in software [3]

1.1.1 Models used for effort estimation in Software Development

- COCOMO model - COCOMO i.e. Constructive Cost Estimation Model was being proposed by Boehm. The basic COCOMO model is stated by the following equation:

  Effort = a_1 (KLOC)$^{a\ 2}$PersonMonths

  Here KLOC is the size of any software product which is being expanded as Kilo Lines of Code,a1, a2 stated above are the constants for every category of software products, Effort is computed as total effort required for developing the software

product, stated in units of person months Boehm has derived this expression through the analysis of historical data collected from enormous number of real time projects. As there are three classes of software products so the value of constants is different and the formulas for calculating the effort will be:

$$\text{Organic: Effort} = 2.4(\text{KLOC})^{1.05}\ \text{PM}$$

$$\text{Semi-detached: Effort} = 3.0(\text{KLOC})^{1.12}\ \text{PM}$$

$$\text{Embedded: Effort} = 3.6(\text{KLOC})^{1.20}\ \text{PM}$$

- Doty, Bailey Basili and Halstead Models are three of the models which are being used for estimating the effort.

The equations for calculating Effort in them are:

Effort in Doty Model = $5.288(\text{KLOC})^{1.047}$ Person- months

Effort in Bailey Basili Model = $5.5+0.73\ (\text{KLOC})^{1.16}$ Person- months

Effort in Halstead Model = $5.2(\text{KLOC})^{1.50}$ Person- months

These models are being derived by studying a huge amount of complete software projects which are gathered from various organizations and applications so that it is easy to know that how several project sizes are mapped into project effort [10]

1.1.2 Techniques used for Effort Estimation

Many techniques have also been introduced for the purpose of effort estimation. Those various techniques are:

- **Regression analysis**- Regression analysis is one of the statistical tools which are being used for investigating the relationships between several variables. Regression analysis makes us understand that how a value of the dependent variable will change when any independent variable is being altered, and some other independent variables are kept as having fixed values. The main applications of Regression analysis are for predicting and forecasting the values. [1]
- **Statistical model**–Statistical model covers some Statistical modeling approaches. These approaches are being examined for the purpose of fitting the effort

estimation data and the results are being compared in the form of Root mean square error values: Linear Model contains those models which includes constant and the terms of first order only. Pure-Quadratic Models are those which include some constant, linear and squared terms as well. [12]

- **Genetic algorithm**- Genetic Algorithms are called to be an adaptive and heuristic search algorithm because they imitate the overall process consisting natural selection. Innovation of that particular selection makes the process as the complete family of data processing model. For the purpose of preserving the important data various operators are combined and are then applied to the prescribed structure, which contains simple chromosome-like data structure and the defined algorithms will then derive a proper solution to a specific problem occurring on these structures. The solutions will be developed for optimizing the disturbances by using techniques such as inheritance, mutation, crossover and selection of data. In simple way, we can say that a genetic algorithm is a type of population based model which have used selection and various recombination operators so that new sample points are being generated in a defined search space. [5]
- **Artificial Neural Network** - Artificial neural networks are those models which are being inspired through biological neural networks. These are used for estimating or approximating those functions which depends upon a large number of inputs and are mostly unknown. Artificial Neural network is being presented as a system of interconnected neurons in which values can be computed with the help of inputs. [6]

Among these techniques, neural network is being used as a proposal for estimating the effort. Neural networks have their own adaptive nature so it is good to use them here.

1.2 Artificial Neural Network

It is a system being motivated through architectural composition, Method used for processing and learning skills of a biological brain.

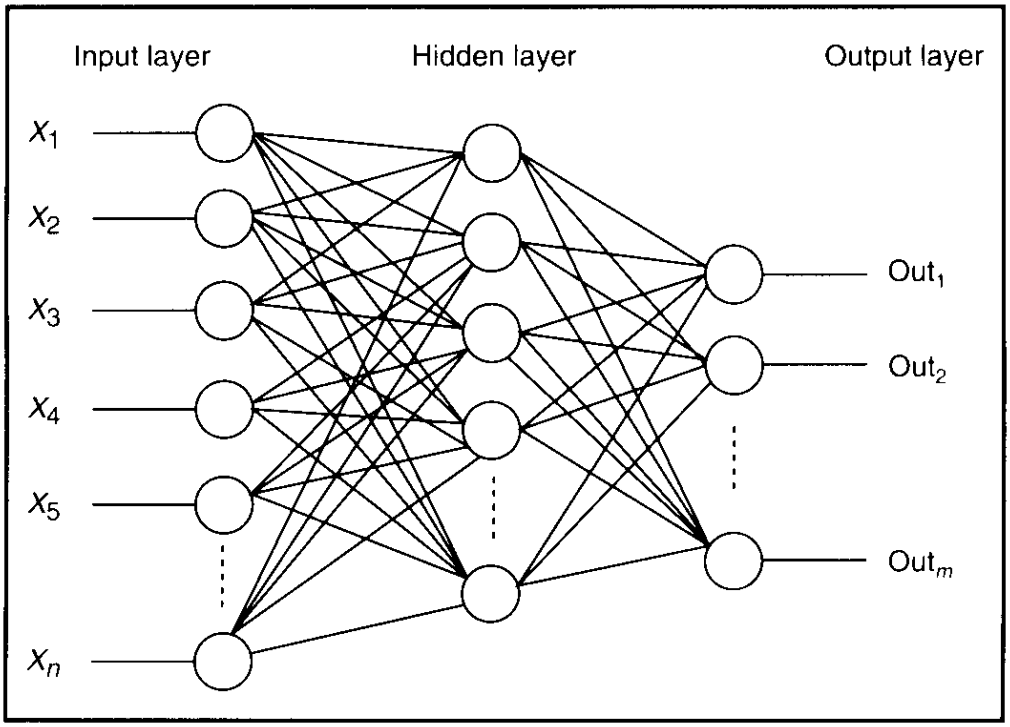


Fig 1.1: A Simple Neural network [21]

1.2.1 Characteristics of Artificial Neural Networks

- **The Structure of its network**: - The Structure of network in ANN should not be very complicated. It should be described in an easiest way. Structures are of two types i.e. recursive structures and non recursive structures. The Recursive Structure can also be calledas self associative or A Feedback Network and the Non Recursive Structure can also be calledas Associative network or A Feed forward Network.

- **Concurrent Processing Capability**: - Artificial Neural Network is only known for enlargement in the concept of processing concurrently in the field of computers. This processing is being performed through the neurons in human body and are also complicated in nature but by implementing some simple processing techniques we can apply it in Artificial Neural Network like Matrices and its various computations.

- **Distributed Memory:** - As we know Artificial Neural Network is a vast system so an isolated place or a single central memory cannot achieve the needs of it therefore in that case weight matrix i.e. a kind of long term memory is needed for

storing information because it is good to store information in some form of patterns throughout the network structure.

- **Fault Tolerance Ability**: - Artificial Neural Networks are complex to be understood by a layman so one of its requirement is that it should be reliable or having an ability to tolerate the fault Because if any single part is failed it won't influence the system much but if all of its fragments are failed simultaneously the overall system will fail automatically.
- **Collective Solution**: - ANN is an interdependent system where the outcome of a system is derived from collection of various inputs therefore the result is sum of each and every output that comes after processing of those multiple input values. The incomplete answer is treated as null for any kind of ANN system user.
- **Learning Ability**: - In Artificial Neural Networks maximum numbers of learning rules are being utilized for developing various models of processes, while the adoption of the networks in the fluctuated environment and finding the required and beneficial knowledge. The various learning methods are supervised learning, unsupervised learning and Reinforcement Learning. Large amounts of processing neurons like processing elements are present to be provided into the network. [11]

1.2.2 Applications of Neural Networks

- **Character Recognition** – Neural networks are used for recognizing handwritten characters as various handheld devices are becoming popular day by day.
- **Image Compression** – As a large amount of information can be received by neural network in a single instant so it makes image compression useful in it.
- **Stock Market Prediction** - High fluctuations in stock market in a regular manner makes the business more and more complicated As neural networks can analyze a large amount of information quickly and solve it out, so they can be used for prediction in stock prices.
- **Traveling Salesman Problem** - For some level of approximation, neural networks can be able to solve the traveling salesman problem.
- **Functional Approximation Problem** - A functional approximation problem is that type of problem in which function is being selected from a well-defined class

and is evaluated to match closely or approximates some well defined target function. [17]

Here we will use one of the applications of neural network i.e. functional approximation for the purpose of Effort Estimation

1.3 Functional Approximation:

In functional approximation, a function is being developed for computing the values generated as software effort in which various features are being used as its variables. The task is to select any function from a well defined class which will be nearly equal to its target function in a specified way. In functional approximation three steps are being followed:

- Data preparation from each considered database
- Consider the attributes and their associated values as function variables.
- The problem of estimating software effort is then be transformed into the problem of functional approximation to achieve some sort of accuracy in it. [15]

1.4 Non Linear Regression

Non linear regression is one of the types in regression analysis. Nonlinear regression is a statistical technique that will help for describing some sort of nonlinear relationships in experimental data. In this the data observed will be modeled by any function that is dependent upon some independent variables. Nonlinear regression models are generally assumed to be parametric, where the model is described as a nonlinear equation. Parametric nonlinear regression assumes the dependent variable as a function that combines nonlinear parameters with some independent variables. The model can be built as univariate that consists of single response variable or multivariate which will contain multiple response variables in output. [13]

1.4.1 Algorithms used for Non Linear Regression includes:

- **Gauss- Newton Algorithm**: This algorithm provides a method for solving non-linear least squares problems which are being arisen for some instances in non linear regression. The Gauss–Newton algorithm can only be used for minimizing a

sum of squared function values, but it has the advantage that second derivatives are not required in it, which is difficult for calculations.

- **Gradient Descent Algorithm:** Gradient descent is an optimization algorithm used to find a local minimum of a function, which will take steps proportional to the *negative* of the gradient of a particular function at the current point. If steps are taken as proportional to the positive of the gradient, one approaches a local maximum of that function; the procedure is then known as gradient ascent. Gradient descent has another name as steepest descent, or the method of steepest descent.
- **Levenberg- Marquardt Algorithm:** This algorithm is used for solving non linear least square problems. These problems are mostly arising in least square curve fitting. The Levenberg- Marquardt algorithm comes between the Gauss–Newton algorithm (GNA) and Gradient descent. The LMA is more reliable than the GNA. The LMA tends to be a bit slower than the GNA for well-behaved functions and reasonable starting parameters, LMA can also be treated the additional use of trust region approach in Gauss–Newton algorithm. [20]

1.5 **Parameters of interest:** Once the effort is being estimated, various parameters are being calculated like-

- Error
- Relative Error i.e. RE
- Magnitude of Relative Error i.e. MRE
- Mean Magnitude of Relative Error i.e. MMRE, and
- Mean Square Error i.e. MSE

These computations are being done to check the accuracy fluctuation which can thus be reduced to achieve the accurate value.

CHAPTER 2

LITERATURE REVIEW

Chetan Nagar and Anurag Dixit (2011) [8] said Competition is increasing regularly in the software industries; for such scenarios the accuracies in effort estimation becomes an important task. Efforts estimation provides basis for other software activities like scheduling, planning. Many methods are existing for efforts and cost estimation, but people do not know about their usage. Industries only concentrate on KLOC for efforts estimation, but there are many other parameters like efforts of reuse must be included in effort estimation. An IT industry makes two types of projects:

- First, which has a specified customer; all requirements will be given by that particular customer.
- Second, in which requirements are gathered by using some survey.

In case of second type of projects, effort estimation become much difficult because there is no specific customer and efforts are needed to be estimated accurately. One more requirement is to complete the project within time limit, because there are number of competitors. So a systematic approach is required for efforts estimation

These steps must be considered for accurate effort estimation.

- Estimation team must contain experience person, because an experienced person can estimate more accurately.
- Never use any single method for estimation. If we are using one method for estimation and some mistake will occur then there is no other alternate way to check it.
- Maintain a database for reference to next projects.
- Make a work break down structure and follow it strictly.
- Estimate efforts for all phases of software development

Saleem Basha, Dhavachelvan (2011) [4] discussed various estimation models in which Empirical Software Effort estimation models are considered to be more accurate. Accuracy depends upon understanding the quality characteristic of the software data, its reliability and many more factors. Many organizations and research people have applied max of their effort for developing the efficient models so that effort being estimated comes nearby to the accurate values. In these types of estimating modeling techniques, the parameters being estimated can be easily deployed from the empirical data which is usually gathered from different sources of several old software projects.

Various empirical Estimation models are:

- COCOMO Estimation model
  - COCOMO 81
  - COCOMO II
- SEER SEM Estimation model
- SLIM Estimation model
- REVIC Estimation model

COCOMO Estimation model

COCOMO 81 i.e. Constructive Cost Model is categorized as one of the model used for estimating effort, its cost, and respective schedule for software projects in 1981. In COCOMO 81, effort is being stated in the units of Person Months i.e. abbreviated as PM and is calculated as

$$PM = a * Size^{b} * \prod_{i=1}^{15} EM_i$$

Here in this formula, “a” and “b” are two of the domain constants present in the model. This model usually consist fifteen effort multipliers.

COCOMO II proposed an enhanced scheme for estimating the effort. In COCOMO II, the effort requirement can be calculated as

15

$$PM = a * Size^{E} * \prod_{i=1} EM_i$$

$$\text{Where } E = B + 0.01 \sum_{j=1}^{5} SF_j$$

Here SF are the 5 scale factors i.e. Very Low(vl), Low(l), Nominal(n), High(h), Very High(vh).

Many disadvantages are being found in COCOMO II estimation model but it is still influential to use due to their better accuracy compared to other estimation schemes.[12]

SEER- SEM model

SEER i.e. expanded as System Evaluation and Estimation of Resources is project managerial software application which is designed specifically for estimating, planning and monitoring the effort and resources which are being needed for the development of software and/or its respective maintenance.

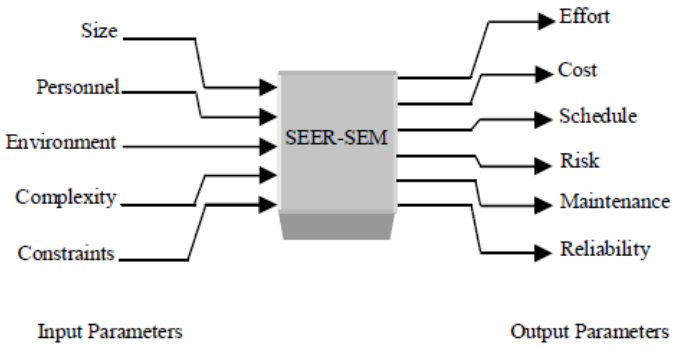


Figure 2.1:SEER-SEM I/O Parameters[3]

SLIM Estimation model

This estimation model was being established by the developer Larry Putnam. SLIM employs one of the principles known as Rayleigh distribution which is developed among the personnel level and time taken. SLIM is acceptable in all the vast projects which exceed upto 70,000 loc.

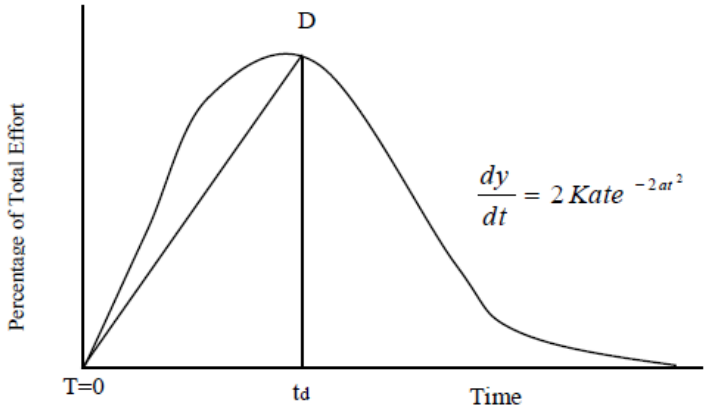


Figure 2.2: The Rayleigh curve[3]

Rayleigh curve is being used for predicting the effort. The above drawn curve shows the manpower calculated in terms of person per time being stated as a function of time.

REVIC Estimation Model

REVIC is considered as one of the direct successor of COCOMO model. More cost drivers are being used in it than that of COCOMO. REVIC uses statistical techniques like PERT to determine the input value which is present in form of lines of code. Various effective estimates for every component of program are used for calculating the effective loc and thus standard deviation (SD). These effective loc and standard deviation (SD) are used in equations for estimation purpose preferably as the sum of the estimates in loc.

After discussing all these models it is stated that all the models are highly credible but not accurate to use because some of them stay in the term of prediction, and prediction doesn't considered to be truly proved in this estimation models. Finally it is concluded that no particular model is good for all types of situations and environment.

Practical Software Project Estimation in 3rd edition (McGraw-Hill) [18] explain three techniques which are being used for Effort Estimation

- Estimation by use of equations
- Estimation by use of comparison
- Estimation by use of analogy

Estimating by using equations

One of the techniques for estimating software project has involved the use of equations. These equations allows the developing teams for calculating estimate of a specific metric so that some effort or duration just by insertion of the measured size of our software project in the exact equations.

Estimating by using comparison

In this the experts are allowed for achieving some more described estimates except those which can be achieved through equations. This type of estimation involves the considerations in terms of the attributes of the project to be estimated.

Estimation by using analogy

The estimation which is based upon any analogy involves the selection of some complete projects that nearly matches to the features of the already developed project. [17]

Pichai Jodpimai, Peraphon Sophatsathit (2012) [6] tries for improving the estimation in effort by including some of the principles of mathematics and a technique i.e. artificial neural network.

Three major steps are being considered:

- Data preparation from each considered database
- Consider only the relevant features so that its number is being reduced.
- The problem of estimating software effort is then be transformed into the problem of classification and then functional approximation in FLANN.

For constructing the COCOMO 81 model, 16 independent features and one dependent feature is being carried out.

$$Effort = a*(KSLOC^{b})*(\prod_{i=1}^{15} EM_{i}).$$

where Effort consist of an estimated value which is being measured in units of person-months; Above mentioned *a* and *b* are those parameters which are used for representing three software development modes, which are named as Organic, Semi-detached, and

Embedded; KSLOC in above formula shows 1000s of loc in program without including any blanked and comments in lines. The values of KSLOC are being estimated or can be easily derived through fpa i.e. function point analysis; EM written here is the effort multipliers which is being calculated through various factors of environment and can also be called as effort drivers.

From these above features, some relevant features are being assumed to be selected that shows some significant effect on the effort estimation of a particular software. The way of selecting those features is same as that of development of a function for computing the amount of effort by using these above features as its variables. In this, a neural network is being used for performing functional approximation due to simplified nature &effective results.

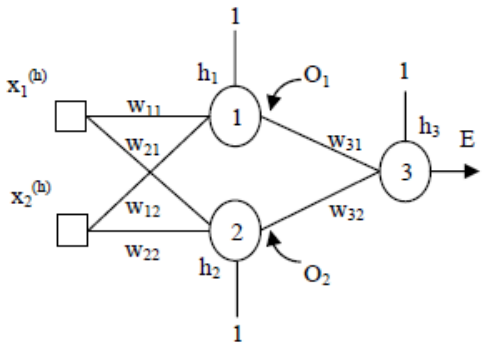


Figure 2.3: A neural network model[6]

The algorithm being used here is Feature selecting algorithm. With the help of this algorithm, the number of given features are being reduced and only the relevant ones will be considered.

Next step is the transformation of the problem of estimating software effort into the problem of classification and functional approximation. Two major steps in creating the software effort function are:

- Classification of given features into two ranges by using Feed Forward Neural Network technique. A single neural network is used in classification purpose.
- Functional approximation is then performed for each group. Two neural networks are being used in this step for creating software effort functions.

After doing all this, the output accuracy will be assessed through various metrics i.e. MRE, Mean MRE and PRED

$$MRE_i = \frac{|\hat{y}_i - y_i|}{y_i}$$

Here y_i is the given values of actual effort, & other is estimated effort which is being calculated, these are used in software project *i*. N present in below equations is the total no. of projects.

$$MMRE = \frac{1}{N}\sum_{i=1}^{N} MRE_i$$

$$PRED(x) = \frac{1}{N}\sum_{i=1}^{N} \begin{cases} 1 & if \quad MRE_i \le x \\ 0 & Otherwise \end{cases}$$

Small value of Mean Magnitude of Relative Error and large value of Percentage Relative Error Deviation is mostly preferred.

P. Subitsha and J. Kowski Rajan (2014) [14] has discussed that Software Effort Estimation models adopt a neural network technique for providing a solution to improve the accuracy. ANN is a technique which is used for modeling the complex relationships stated between various individualistic and relying features.

.Different types of techniques used in it are:

- Multilayered Percepton Network
- Radial Basis Functional Neural Network
- Support Vectors Machines
- Particle Swarp Optimization

If a proper estimation technique is being selected then it also provides a significant impact on performance.

Effort Estimation process is done in those phases:

- Data Preprocessing
- Technique Setup
- Input Selection

- Evaluation criteria
- Testing

Evaluation is being carried out on the basis of MMRE and PRED accuracy measures.

$$MRE_i = |\text{actual effort}_i - \text{estimated effort}_i| / \text{actual effort}_i$$

MMRE is the mean of above calculated MRE

These models indicate that by using different data preprocessing steps, possible data quality issues are being addressed which can play an important role in software effort estimation. In this more impact on the budget estimations can be possible with further extensions which will be beneficial for staff.

Muhammad Waseem Khan and Imran Qureshi (2014) [7] lists many parameters, i.e., time, effort, and personnel used for the software development that can be estimated using Effort estimation. Software estimation consists of three stages, i.e.,

1st stage is size estimation,

2nd stage is effort estimation, and time estimation, while

3rd stage consists of cost estimation and staffing estimation

At last this paper concluded that the proposed neural network approach yeilds high effort estimation accuracy then the other models using only a few features of it.

CHAPTER 3

PRESENT WORK

In this chapter, we are going to present the problem of our research work, its objectives, the methodology that we used for our proposed approach and the introduction of the tool which will be used to complete the task. In the 3.1 section we explain how we formulated our problem and what the approach we are going to use. In the 3.2 and 3.3 section the objectives and methodology of the work done. In the methodology the flow of our work with the help of flow chart is explained.

3.1 Problem Formulation

Effort estimation is considered to be main problem for all type of stakeholders. Effort is being estimated in planning stage of software development so that overall project gives good results. In short, a good estimation becomes the necessity for project managers because it acts as a basis for the projects planning and schedule.

By doing literature survey the most effective way is to estimate the effort by using a function. A function is being developed for computing the values of effort through various features as in the form of their variables. Then by approximating those functions effort was estimated in Feed forward neural network. In functional approximation these three steps are being followed:

- Data preparation from each considered database
- Consider the attributes and their associated values as function variables.
- The problem of estimating software effort is then be transformed into the problem of functional approximation to achieve some sort of accuracy in it.

But there exists a problem of data adequacy so the proposal is to estimate the efforts using Functional approximation in which each data will be considered without any partiality because the data being missed can be effective to achieve nearby accurate values. The coding for it will be done in MATLAB.

3.2 Objective

Objective of the project is

- To use all features/ attributes as a whole so that there will be no problem of data adequacy.
- To estimate the effort such that proposed model will have least error values as compare to various predefined models like Doty, Bailey Basili, Halstead, etc.
- To make it helpful for project managers and experts for spending less time on the task of estimating the effort and more time to some more important issues.

3.3 Methodology

Neural network is a technique that is being used for estimating the effort. Neural networks have their own adaptive nature so it is good to use them here. It will help for approximating those functions which depends upon large number of inputs and are mostly unknown. MATLAB is a tool that is being used for implementing neural network technique so that effort will be estimated in a good way.

3.3.1 **Process Followed In The Proposed Technique**

As we are doing our task through functions so it becomes easy for desired input and target values to be treated as functions and then they will be able to fit through neural networks. A tool named as nftool is being used for fitting data into neural network. Steps to be followed are:

- Definition of the problem –For defining a problem a set of input is being arranged as columns in matrix and also a set of target values is being arranged as column in some other matrix.
- A tool named as nftool will be used for training a network for fitting dataset into it.
- The tool will be opened by a simple command i.e. nftool in a command window

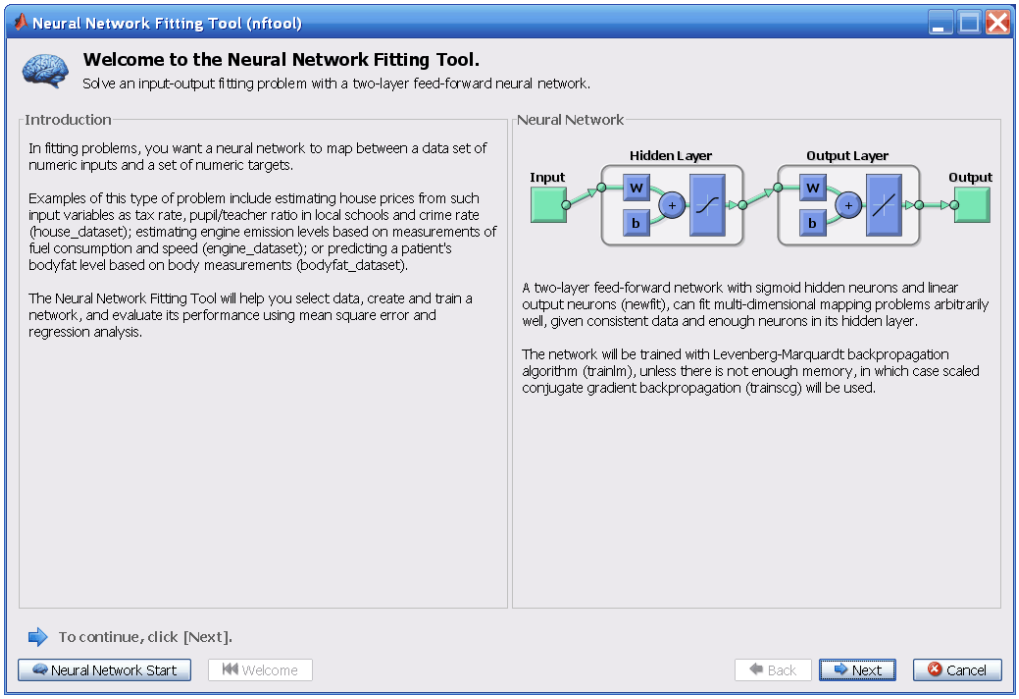


Fig 3.1: Window showing start to the neural network

- As the home page of nftool is opened now the task is to select the input and target matrixes which are already arranged in problem definition step

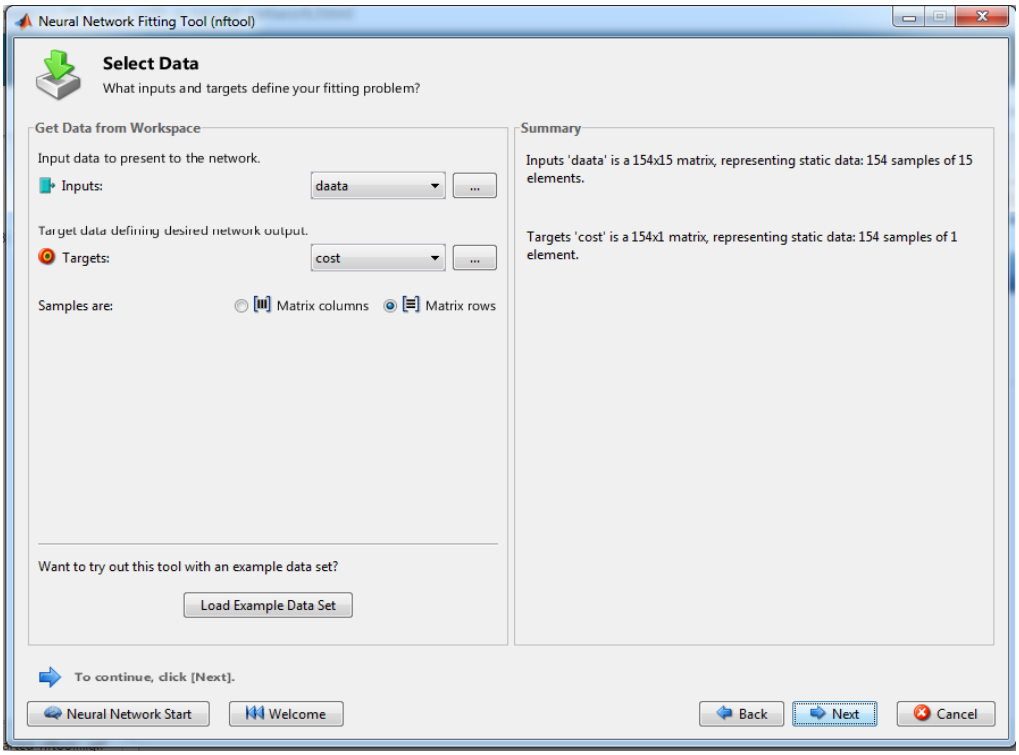


Fig 3.2: Window showing data selection

- Window containing ratio of validation and test data will now be opened. Data present in entire dataset will be partitioned among three parts i.e. Training data, Testing data and Validation data. During training of the data it will be presented to the network so that it will be modified in accordance with its error. Validation data is being used for measuring the generalization in network & at last testing data will provide some independent measures in performance in and after training.

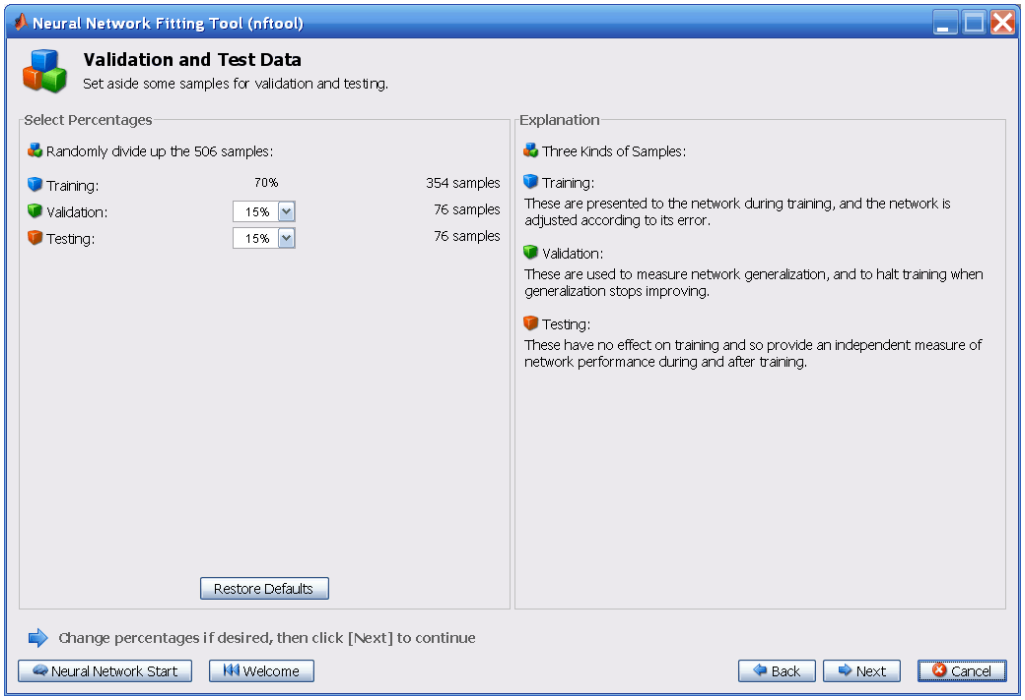


Fig 3.3: Window showing data partition

- After selecting the hidden neurons for training next task is to Train the Network. In MATLAB 2010 training is done with Levenberg- Marquardt algorithm. This algorithm can handle vast size of data and operates it in very less time. Some other algorithms are also present but training through Levenberg- Marquardt is preferred the most.
- Training will be performed untill stoppage in validation of network. Training is required iteratively untill the performance becomes good and target values are not achieved. We can change the number of hidden neurons also if training performance is not improving.

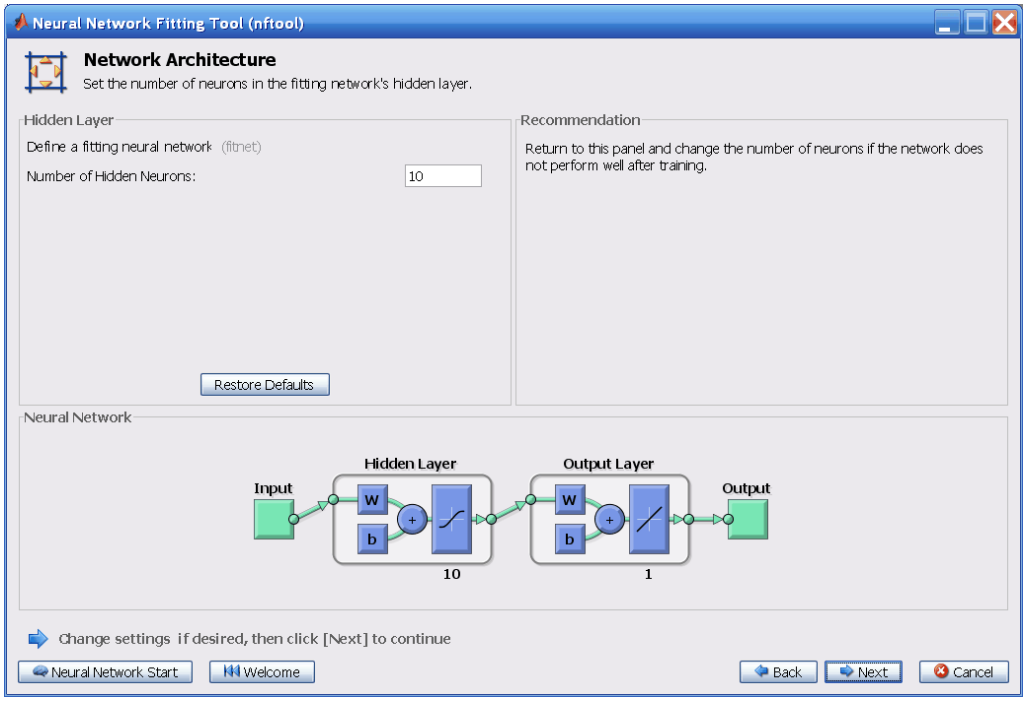


Fig 3.4: Window showing number of hidden neurons

- The network is trained after adjusting the neurons till satisfied outputs are not achieved. Here according to this diagram when input is entered till the time output doesn't match the target values the network will be trained again and again.

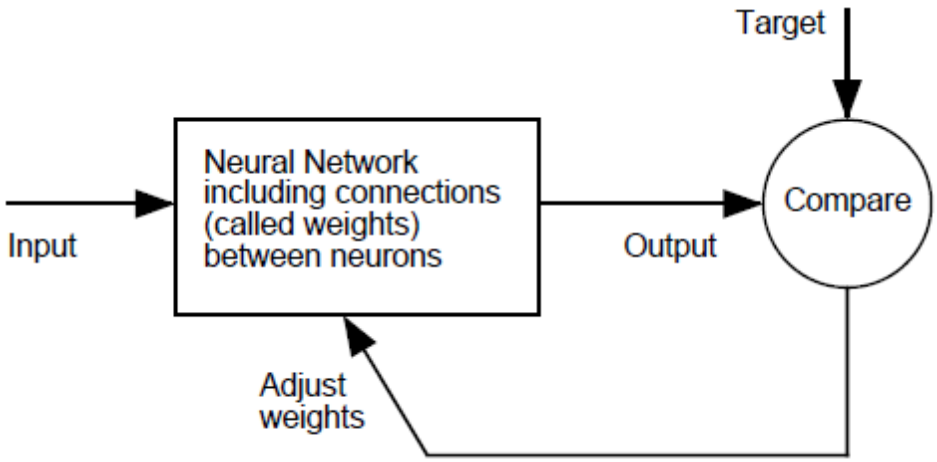


Fig 3.5: A process used for adjusting neurons [2]

- Once the neurons are adjusted and network is trained using Levenberg Marquardt algorithm a code is generated and results are being saved.

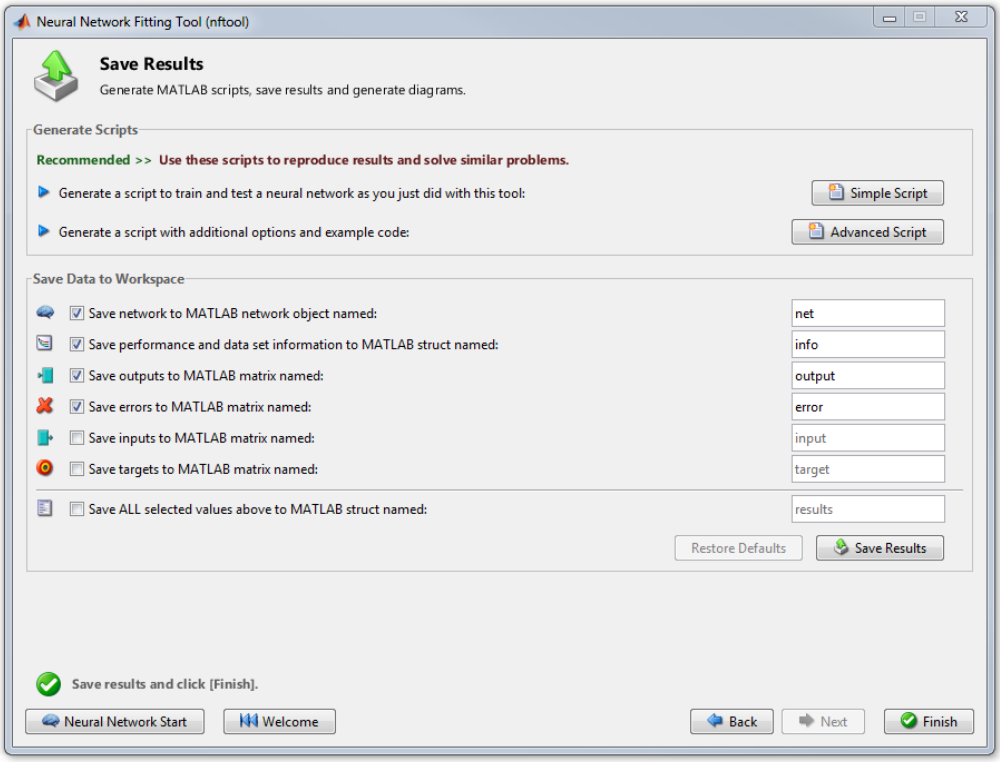
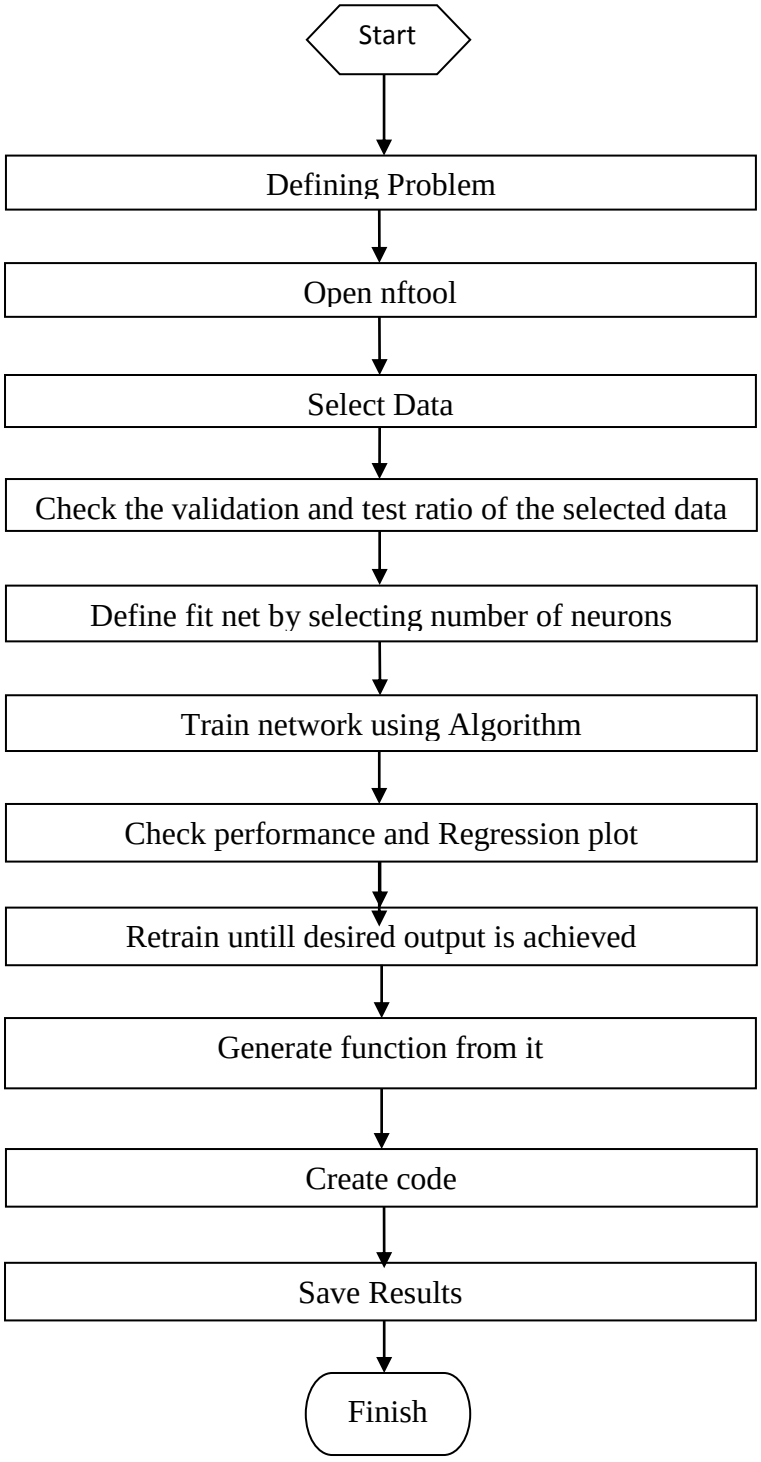


Fig 3.6: Window showing code generation and saved results

Flowchart of the Proposed Technique

In this part we are presenting the flow chart of the proposed approach. Actually flow chart will show the steps of work being done to achieve the desired goal.



3.4 MATLAB 2013

MATLAB stands for "Matrix Laboratory". MATLAB is a high level language. It is a program for doing numerical computation, analyzing images and data. It was originally designed for solving linear algebra type problems using matrices. Its name is derived from Matrix Laboratory. It helps us in solving the problems faster than other languages and used in various applications such as signal processing, image processing, communications, computational biology and control design. MATLAB system has following these parts:

- Desktop tools and development environment
- Mathematical function library
- The language
- Graphics
- External interfaces

Features of Matlab:

- Environment for managing the code, files and data.
- 2-D and 3-D graphics functions for analyzing the data.
- Provide interactive tools for solving problems.
- Provide functions for integrating the MATLAB based algorithms with external applications.
- High level language
- It helps in solving the problems faster than other languages.

3.4.1 **Standard Windows In Matlab:**

- Command Window: - The windows where we type commands and execute them.
- Workspace Window: - This window will show variables and provides allowance for editing those variables through array editor, allows the loading of variables from different files and to clear those variables.
- Current Directory window: - This window shows current directory and some MATLAB files present in existing folder, provides an easy way for changing those folders and for loading various files.
- History window: - This window shows commands which are already executed. These commands can be re-executed by clicking on them.

3.4.2 **MATLAB help**

- Help option is present on the top of the window in the right side.
- MATLAB help is a powerful way for learning the MATLAB.
- It not only contains the theoretical background, but also shows demos for implementation.
- We can search any command by typing in the search box.
- It explains the commands searched by you with examples.

CHAPTER 4

RESULTS AND DISCUSSIONS

In this chapter we have presented various results of the estimations done with our proposed approach and compared with the existing work done. First of all when the historical NASA projects are being analyzed it is being noticed that there are many attributes present in it for each project with their associated values. The various attributes are:

Table 4.1: Attributes present in NASA 60 and NASA 93 projects [17]

No.	Attribute	Description
1	RELY	Required Reliability
2	DATA	Database Size
3	CPLX	Product Complexity
4	TIME	Execution Time Constraint
5	STOR	Main Storage Constraint
6	VIRT	Virtual Machine Volatility
7	TURN	Computer Turnaround time
8	ACAP	Analyst Capability
9	AEXP	Application Experience
10	PCAP	Programmer Capability
11	VEXP	Virtual Machine Experience
12	LEXP	Programming Language Experience
13	MODP	Modern Programming Practices
14	TOOL	Use of Software Tools
15	SCED	Required Development Schedule
16	KSLOC	Kilo Source Lines of Code
17	EFFORT	Actual Effort is measured in Person months

These attributes and their values for each project are deployed in excel sheet so that effort is being calculated by using some predefined models i.e. Doty, Bailey Basili and Halstead models.

The equations for calculating effort through these three models are:

Effort in Doty Model = $5.288(KLOC)^{1.047}$ Person- months

Effort in Bailey Basili Model = $5.5+0.73\ (KLOC)^{1.16}$ Person- months

Effort in Halstead Model = $5.2(KLOC)^{1.50}$Person- months

Here, KLOC is the actual size of project expanded as kilo lines of code

Once these efforts are being calculated by using these equations, the results of them will then be compared with actual effort of each project which is already known through the datasets. The differences in the results will be calculated through some parameters like Error, Relative Error, Mean square error (MSE) and Mean Magnitude of Relative Error (MMRE)

Error = Actual – Estimated

Relative Error i.e. RE = (Actual-Estimated) / Actual

Magnitude of Relative Error = |Actual – Estimated| / Actual

Mean Square Error = $\sum$(Actual - Estimated)^2 / Total number of projects

Mean Magnitude of Relative Error = $1/N \sum MRE_i$ where i is Total number of projects

Through these computations it becomes easy to know how far the results are and how much is required for achieving the actual values. When MSE and MMRE of NASA projects are calculated through above discussed pre defined models, the results are:

Table 4.2 Parameters results through predefined models

Models	**MSE**	**MMRE**
Doty	813765.33	2.948949
Bailey Basili	579878.08	2.810909
Halstead	479796374.1	39.2535

These resultant values of errors i.e. MSE and MMRE after doing their calculations will show the fluctuations among actual and estimated effort. Here shows the graph where comparison among actual and estimated effort of pre defined models is being drawn.

First graph is showing the comparison among actual and estimated effort of Doty model. A dataset of various NASA projects is being taken for evaluating the comparison and fluctuating results.

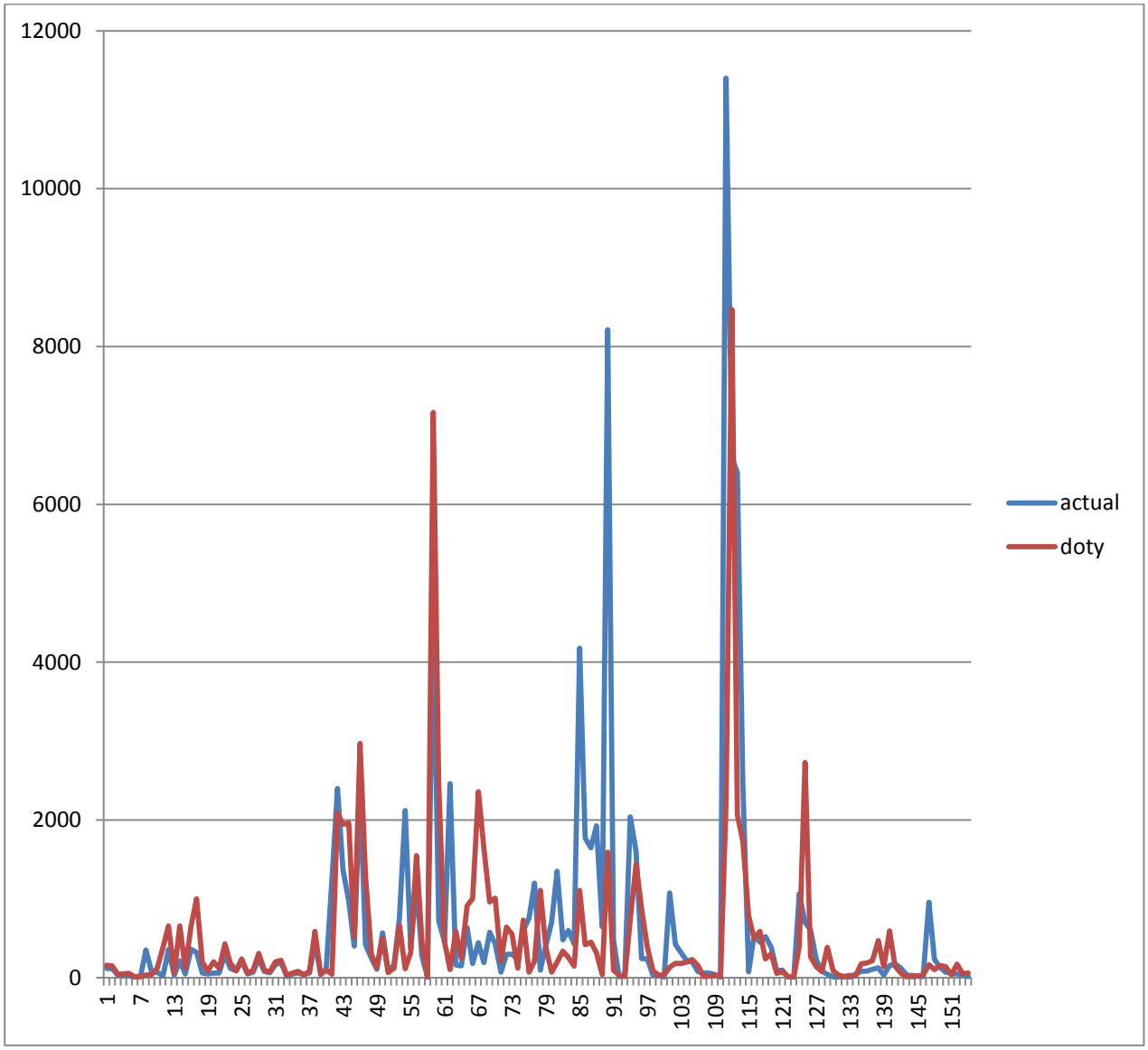


Fig 4.1: Comparison of actual effort and effort calculated by Doty Model

Second graph is showing the comparison among actual effort and estimated effort being calculated through Bailey Basili model. The same dataset of NASA projects is being used for calculating the effort and then comparisons are being made so that it will be known that how much is the difference between actual and estimated values. A graph representation is a best option to show the comparisons.

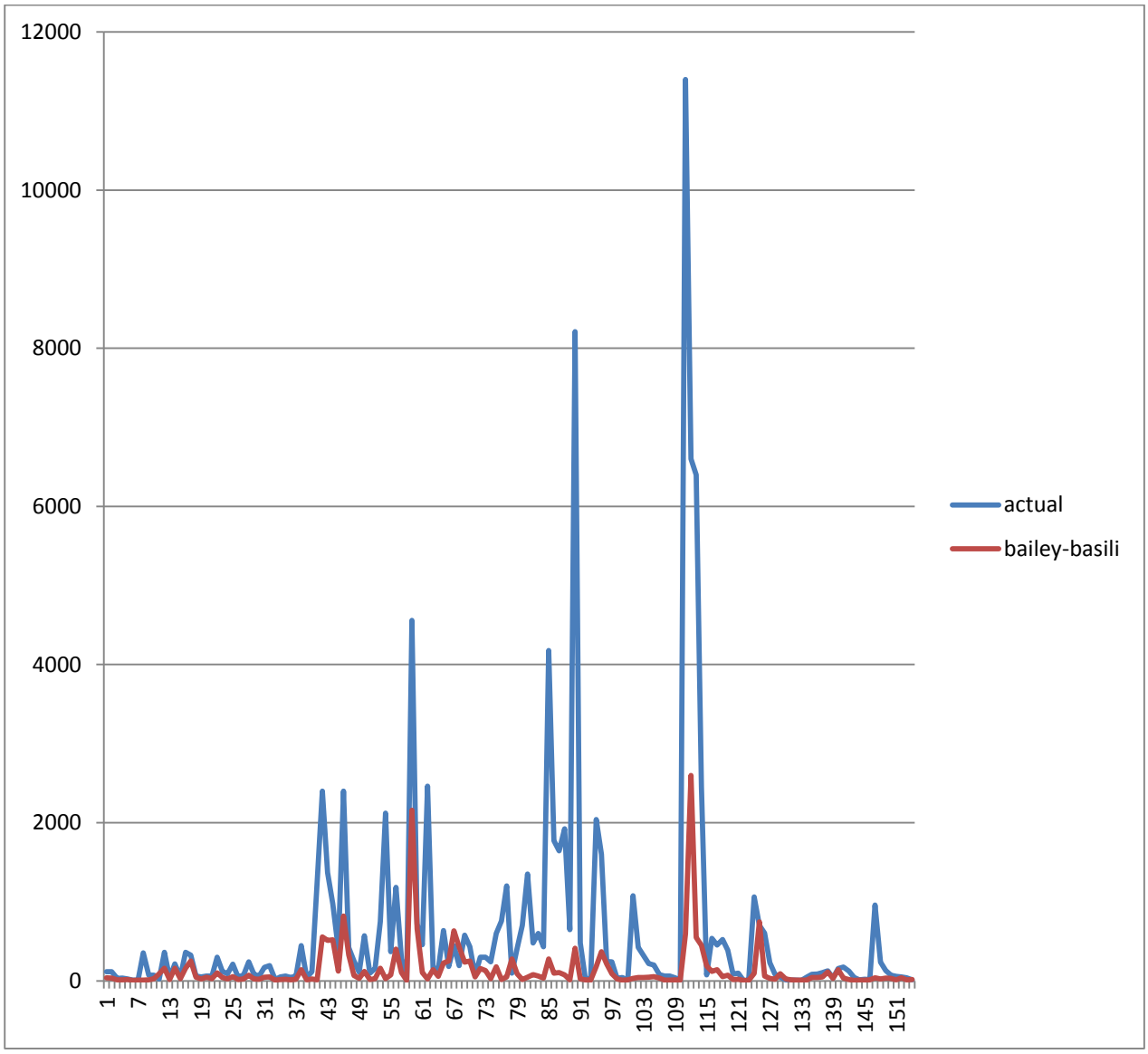


Fig 4.2: Comparison of actual effort and effort calculated by Bailey Basili model

The last pre defined model being used for comparison of results and for showing fluctuations is Halstead model.

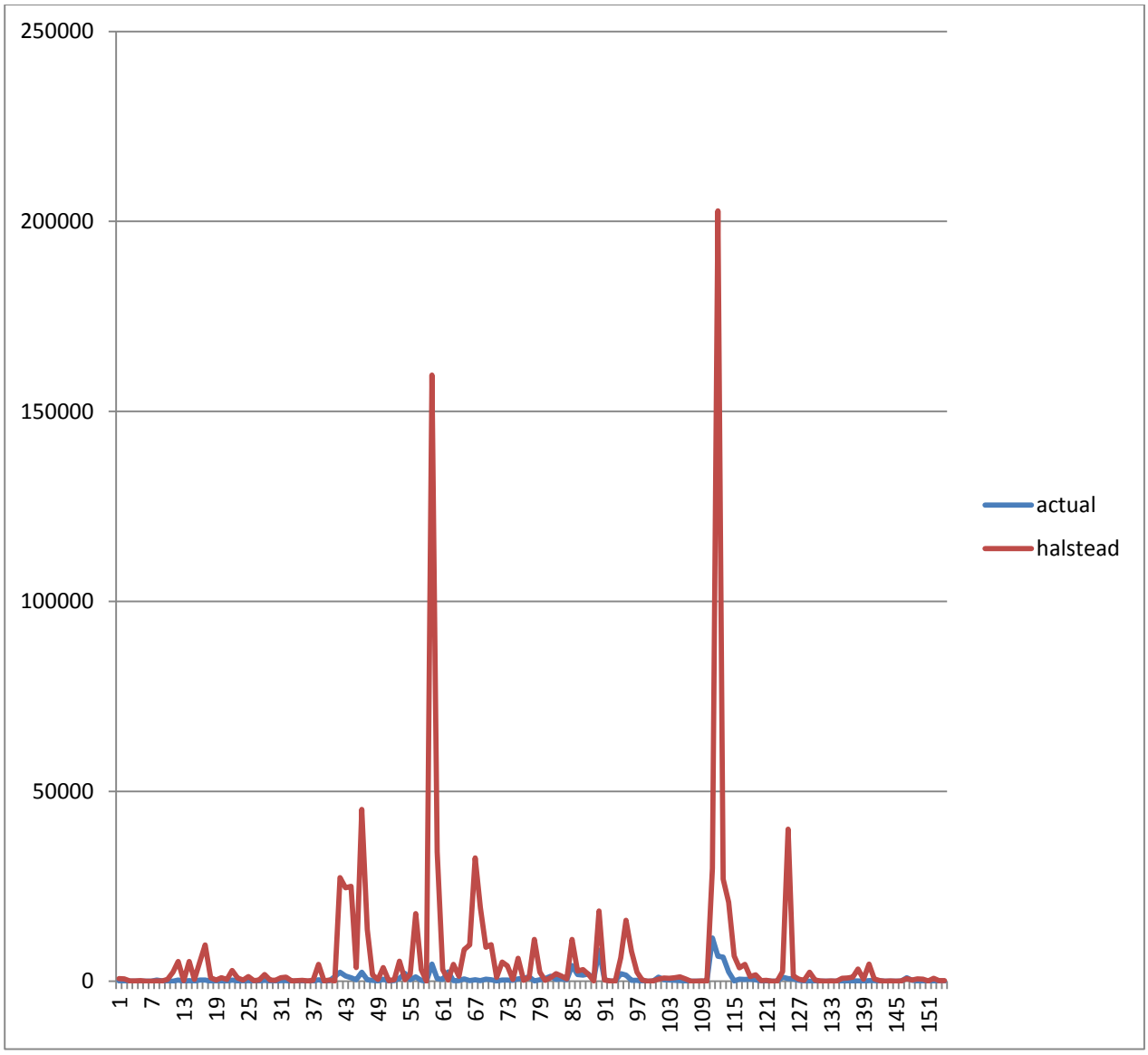
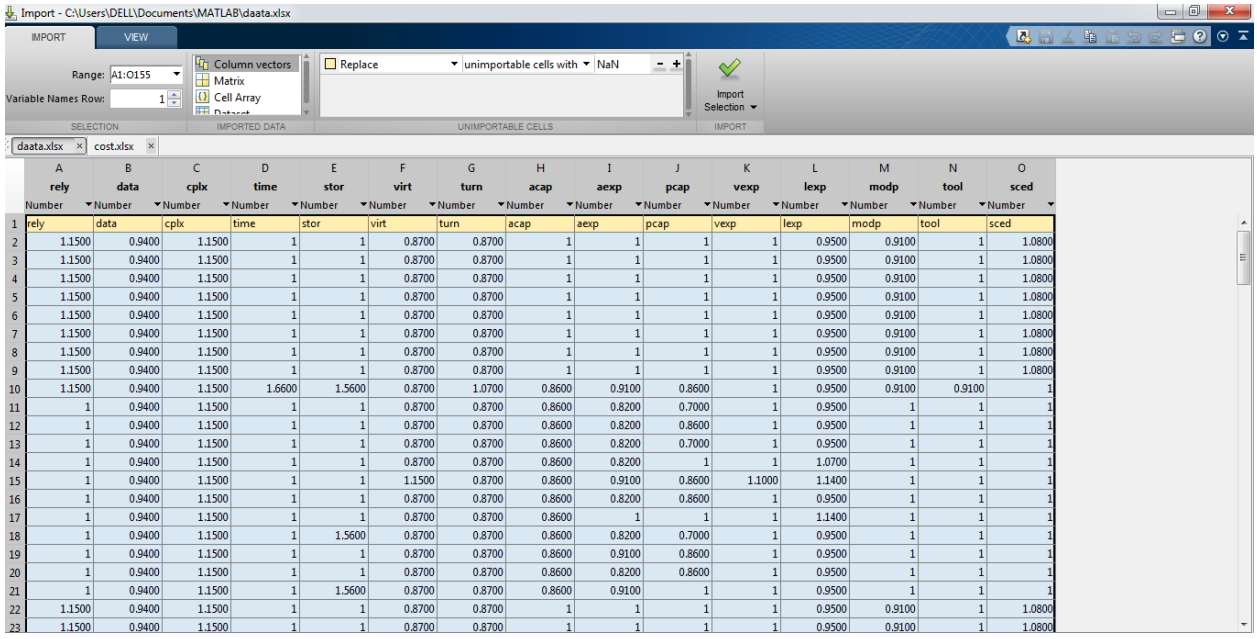


Fig 4.3 Comparison of actual effort and effort calculated by Halstead model

As these models results in very large values of MSE and MMRE and their graphs are highly fluctuating so now the proposed approach is to reduce this gap i.e. the results of MSE and MMRE should be less. For this, the new approach is to use neural networks technique. As neural networks are adaptive in nature so it is good to use for estimating the effort. This all task will be performed through MATLAB. As the proposed approach is to use functional approximation and non linear regression so a function is being generated as

software effort in which the attributes are being used in the form of its variables. Through non linear regression data which is being observed will be modeled by the generated function that depends upon some independent variables.

A neural network toolbox i.e. nftool is used through MATLAB to get the work done. In nftool problem is being defined through selection of input matrixes and one target matrix. The matrices arranged in excel are imported into MATLAB environment



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	rely	data	cplx	time	stor	virt	turn	acap	aexp	pcap	vexp	lexp	modp	tool	sced
	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number
1	rely	data	cplx	time	stor	virt	turn	acap	aexp	pcap	vexp	lexp	modp	tool	sced
2	1.1500	0.9400	1.1500	1	1	0.8700	0.8700	1	1	1	1	0.9500	0.9100	1	1.0800
3	1.1500	0.9400	1.1500	1	1	0.8700	0.8700	1	1	1	1	0.9500	0.9100	1	1.0800
4	1.1500	0.9400	1.1500	1	1	0.8700	0.8700	1	1	1	1	0.9500	0.9100	1	1.0800
5	1.1500	0.9400	1.1500	1	1	0.8700	0.8700	1	1	1	1	0.9500	0.9100	1	1.0800
6	1.1500	0.9400	1.1500	1	1	0.8700	0.8700	1	1	1	1	0.9500	0.9100	1	1.0800
7	1.1500	0.9400	1.1500	1	1	0.8700	0.8700	1	1	1	1	0.9500	0.9100	1	1.0800
8	1.1500	0.9400	1.1500	1	1	0.8700	0.8700	1	1	1	1	0.9500	0.9100	1	1.0800
9	1.1500	0.9400	1.1500	1	1	0.8700	0.8700	1	1	1	1	0.9500	0.9100	1	1.0800
10	1.1500	0.9400	1.1500	1.6600	1.5600	0.8700	1.0700	0.8600	0.9100	0.8600	1	0.9500	0.9100	0.9100	1
11	1	0.9400	1.1500	1	1	0.8700	0.8700	0.8600	0.8200	0.7000	1	0.9500	1	1	1
12	1	0.9400	1.1500	1	1	0.8700	0.8700	0.8600	0.8200	0.8600	1	0.9500	1	1	1
13	1	0.9400	1.1500	1	1	0.8700	0.8700	0.8600	0.8200	0.7000	1	0.9500	1	1	1
14	1	0.9400	1.1500	1	1	0.8700	0.8700	0.8600	0.8200	1	1	1.0700	1	1	1
15	1	0.9400	1.1500	1	1	1.1500	0.8700	0.8600	0.9100	0.8600	1.1000	1.1400	1	1	1
16	1	0.9400	1.1500	1	1	0.8700	0.8700	0.8600	0.8200	0.8600	1	0.9500	1	1	1
17	1	0.9400	1.1500	1	1	0.8700	0.8700	0.8600	1	1	1	1.1400	1	1	1
18	1	0.9400	1.1500	1	1.5600	0.8700	0.8700	0.8600	0.8200	0.7000	1	0.9500	1	1	1
19	1	0.9400	1.1500	1	1	0.8700	0.8700	0.8600	0.9100	0.8600	1	0.9500	1	1	1
20	1	0.9400	1.1500	1	1	0.8700	0.8700	0.8600	0.8200	0.8600	1	0.9500	1	1	1
21	1	0.9400	1.1500	1	1.5600	0.8700	0.8700	0.8600	0.9100	1	1	0.9500	1	1	1
22	1.1500	0.9400	1.1500	1	1	0.8700	0.8700	1	1	1	1	0.9500	0.9100	1	1.0800
23	1.1500	0.9400	1.1500	1	1	0.8700	0.8700	1	1	1	1	0.9500	0.9100	1	1.0800

Fig4.4 Sample window showing imported data in MATLAB

Once these matrices are imported i.e. the problem is defined according to nftool data will be partitioned among three divisions i.e. Training, Testing and Validation. During training, the data will be presented to the network so that it will be modified in accordance to the error. Validated data is used for measuring the generalization in the network & at last testing data will provide some independent measures of performance encountered in and after its training. As we know neural network is consisted of input, output and hidden neurons. Here the input neurons will be the set of input matrices values, we can adjust number of hidden neurons according to the performance of network. The default network which is used here to fit the function is feed forward neural network. It contains sigmoid transfer function in its hidden layer and linear transfer function at output layer. After adjusting the neurons the time is to train the network. The training is being done with the help of Levenberg – Marquardt algorithm. As our motive

is to reduce the error and the purpose of this algorithm is to solve non linear least square problems so it is good to use it here.

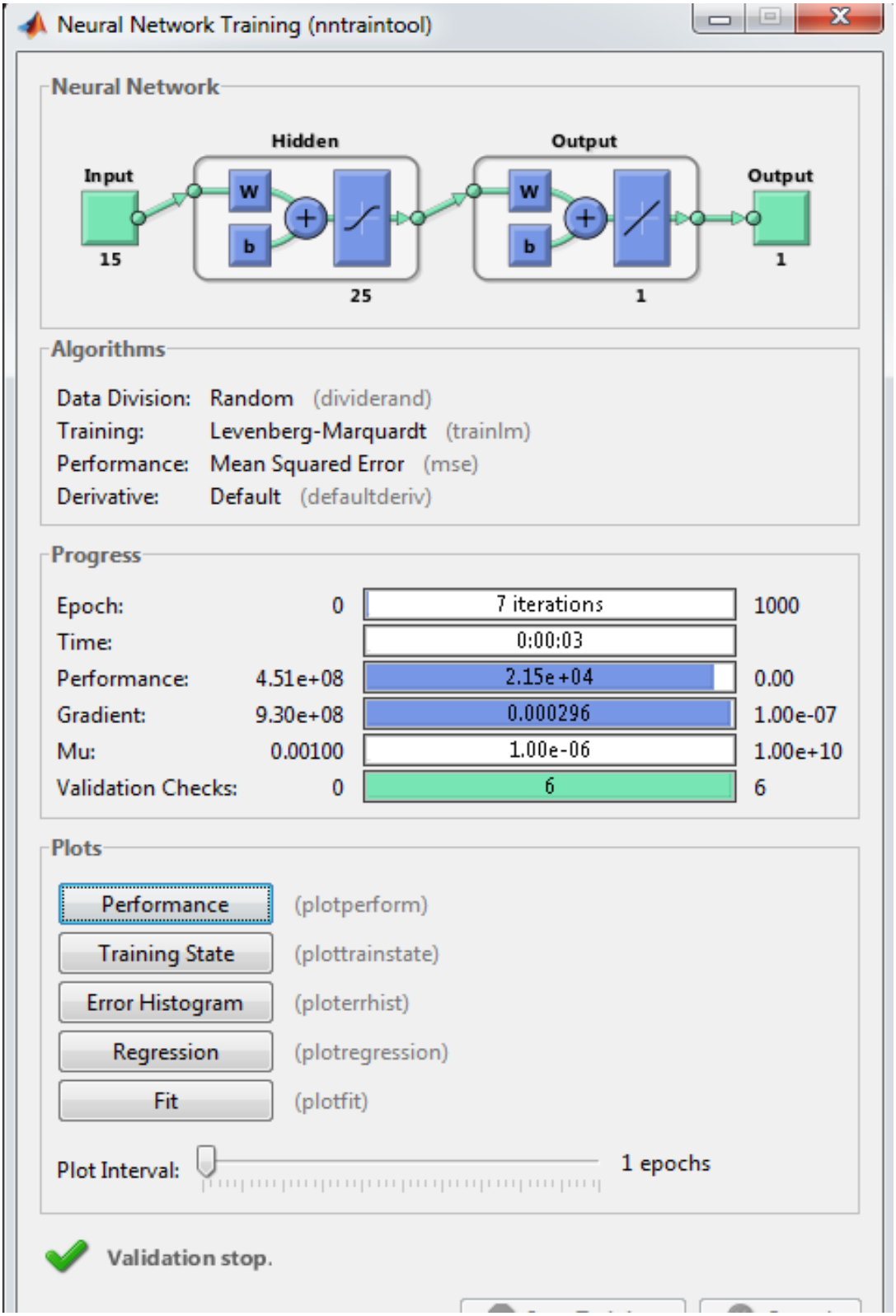


Fig. 4.5 Window showing progress when all validation stops

This window will open after the training of network. This window is divided into four parts: In first part a view to neural network is provided. Tilted line consists of weights and biases in hidden layer shows sigmoid transfer function. Straight line in output layer shows linear transfer function. Second part shows the algorithms and performance criteria's on which results are based. Mean square error is required for checking the performance so that it can show that how close we reach to the actual values. In progress window Third part shows the progress window: Epochs are number of iterations that are required to train the network. Time shows the time elapsed to train the network in seconds. Mu is control parameter in Levenberg - Marquardt algorithm which will be used to train the network. Maximum value of mu is 1.00e+10 and it is best to be achieved.

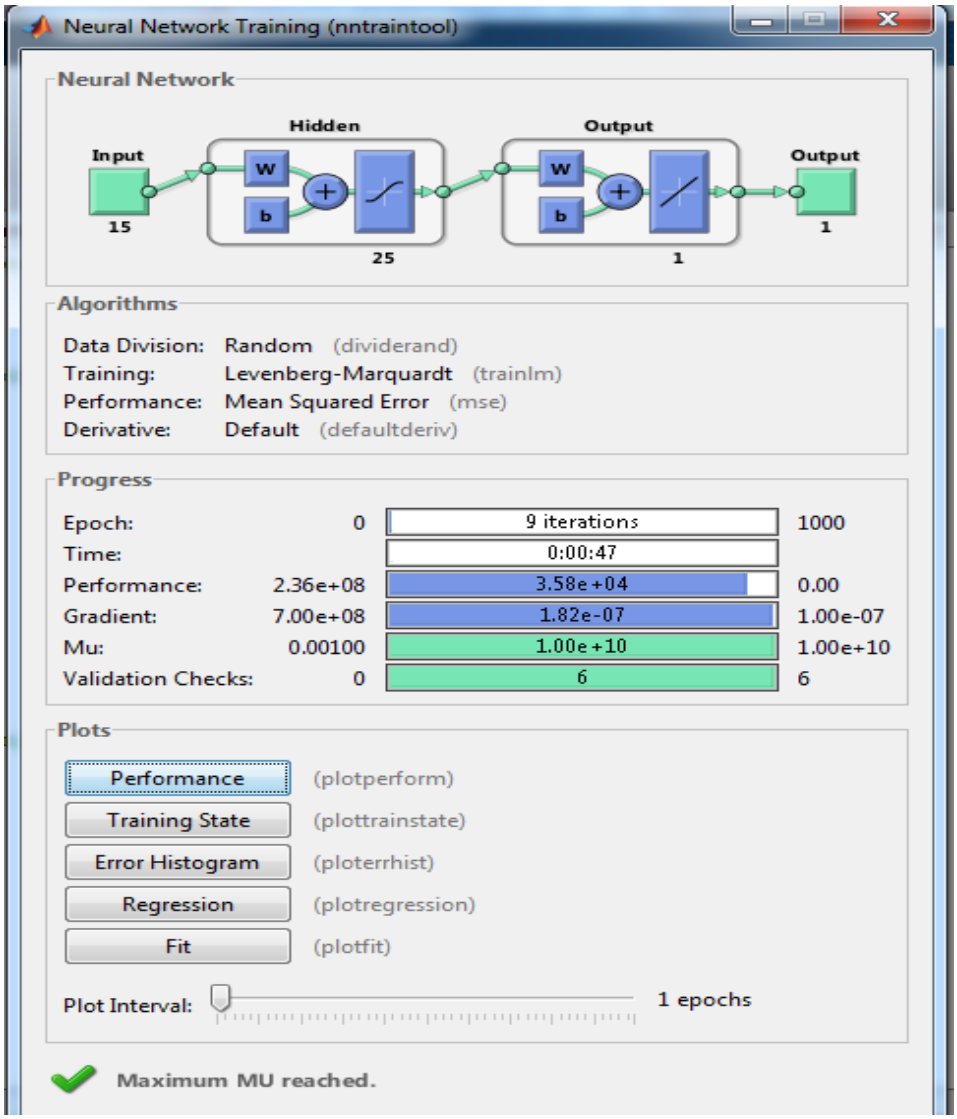


Fig 4.6: Window showing progress when maximum mu is achieved.

Here we achieve the maximum value of mu. The graph for it is

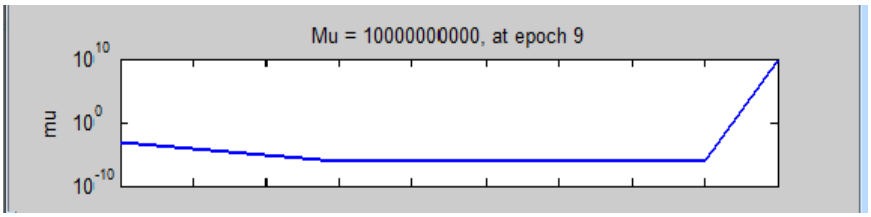


Fig 4.7: Graph showing maximum mu

Opposite to it, the least the value of gradient the best it is. By adjusting the number of hidden neurons this value is also achieved.

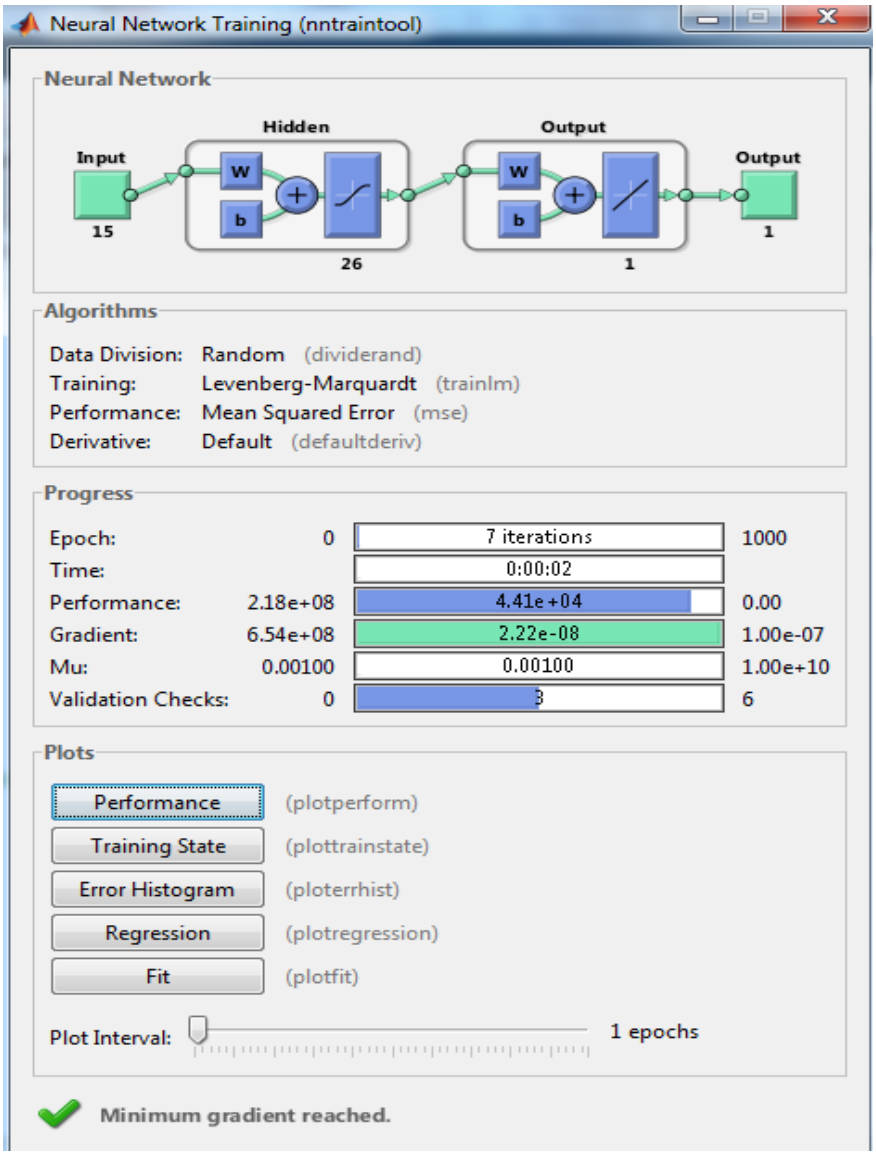


Fig 4.8: Window showing progress when minimum gradient is achieved

Here we achieve the minimum value of gradient. The graph showing it is

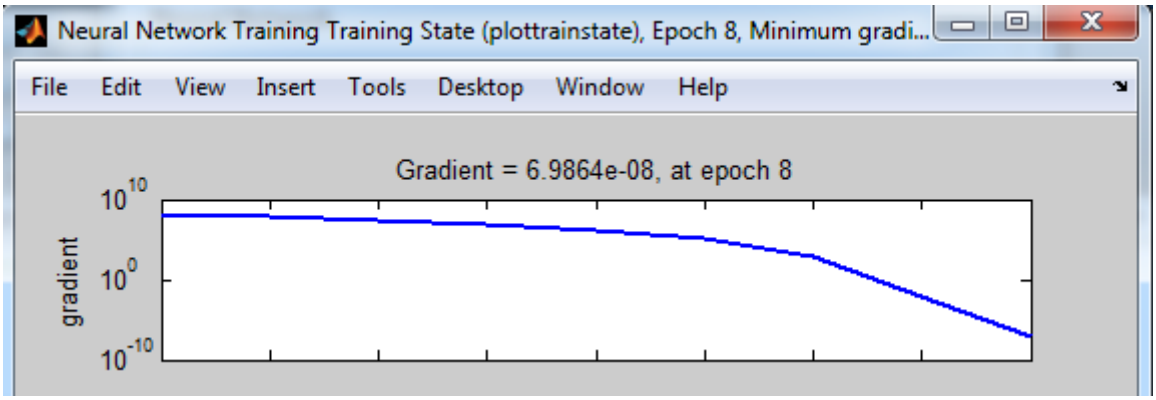


Fig 4.9: Graph showing minimum gradient

The regression plot showing best results is

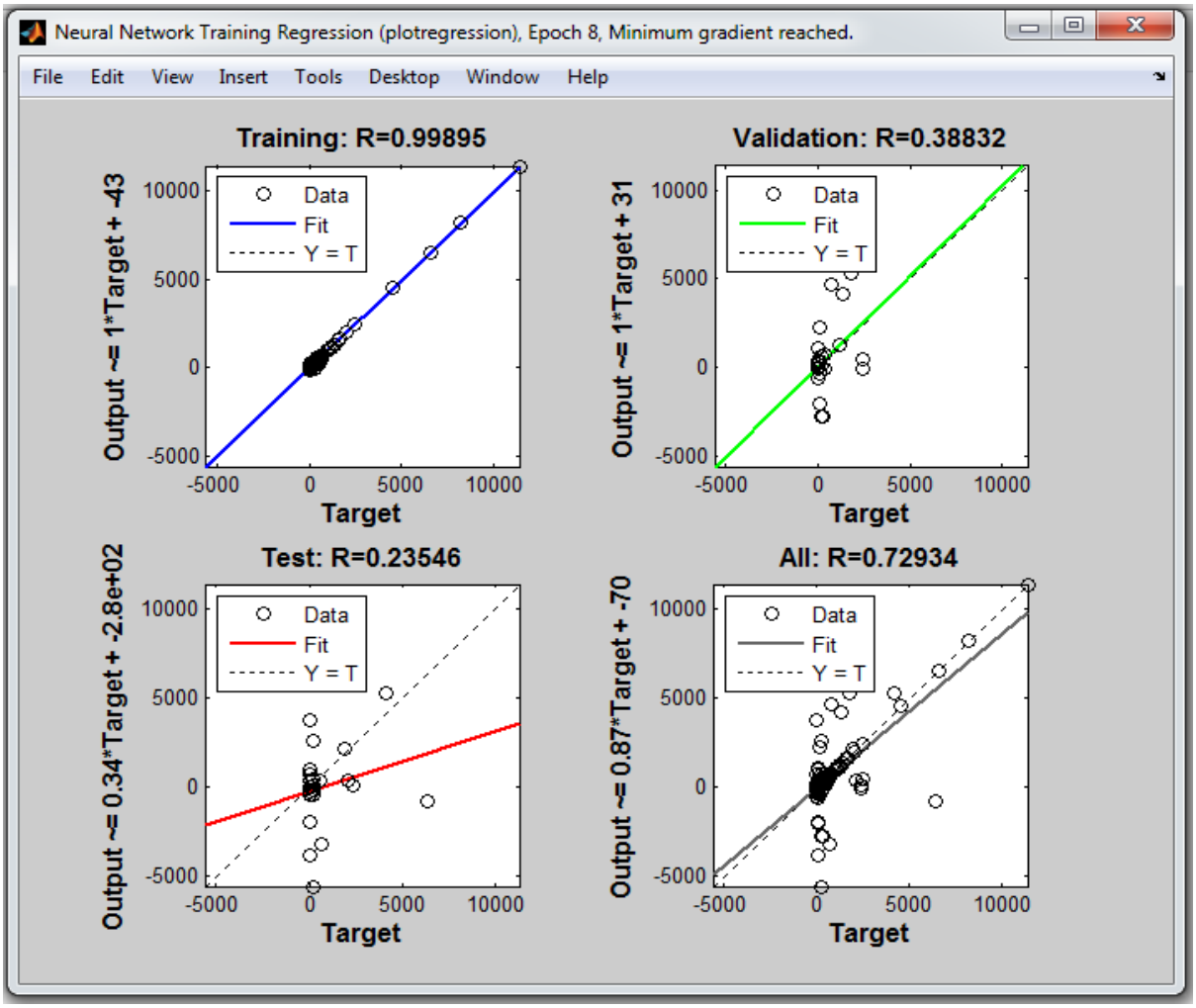


Fig 4.10: Regression plot when minimum gradient is achieved

With this proposed approach a graph is being represented for showing comparisons and fluctuating results among actual and estimated effort. As stated above, these two graphs are showing very less fluctuations than those being concluded through comparison with various pre defined models. MSE is one of the parameter which is desired to have least values for showing the best results. This graph shows the comparison of efforts when least MSE value is achieved.

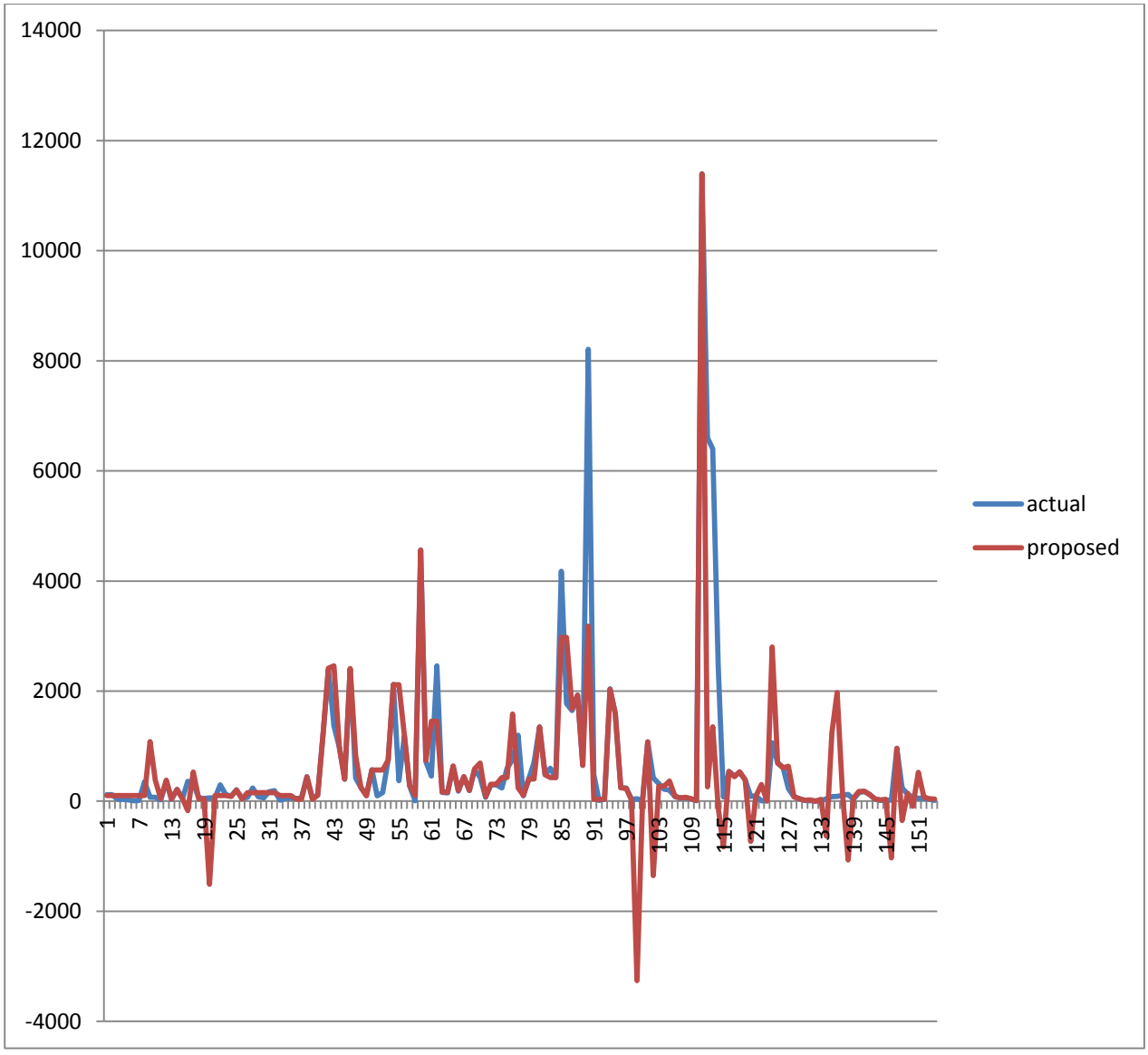


Fig 4.11:. Comparative results of actual and estimated effort which gives least MSE

MMRE which is being expanded as Mean magnitude of Relative error is also one of the parameter which needs to get the lowest values to fulfill the proposed approach. Here in this graph comparison of the values is being represented when least value of MMRE is achieved.

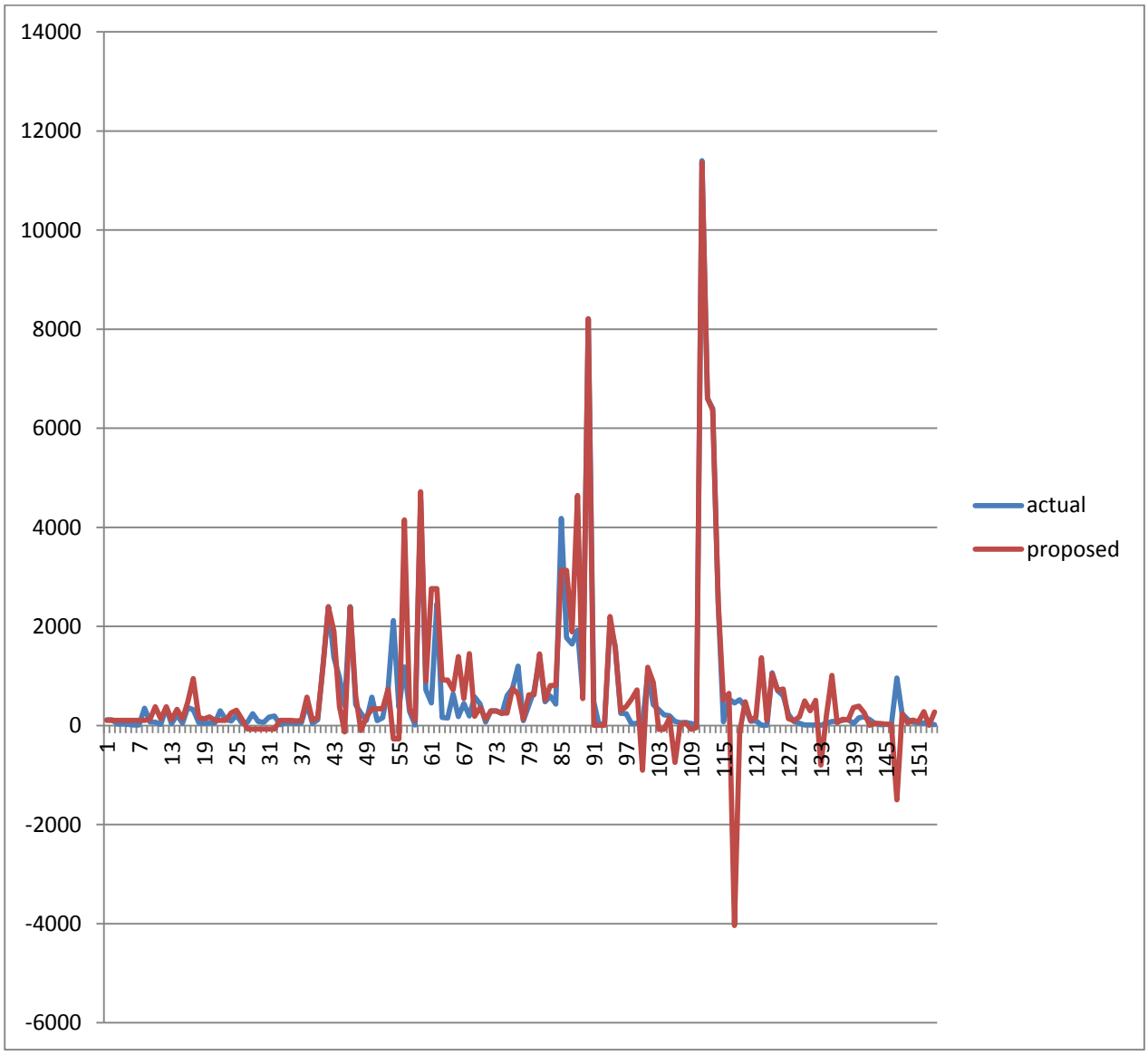


Fig 4.12: Comparative results of actual and estimated effort which gives least MMRE

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

A perfect reliability and accuracy in estimation of development effort has always been a challenge for every type of stakeholder. A number of effort estimates are being forecasted through various models. Their accuracy is being checked through parameters like MRE, MMRE, etc. By comparing their results the proposal model has achieved the nearby accurate values. It is stated that Functional link neural network gives better results than other Neural networks so it is best to use for estimating the effort.

Approximately nearby values are achieved but this Estimation can be improved more in future by implementing some other technique which is not discussed here.

CHAPTER 6

LIST OF REFERENCES

Books

[1] Chatterjee S., Haddi A., "Regression Analysis by Example" Wiley Publications (5th Edition), Aug 2012

[2] Demuth H., Mark Beale, Martin Hagan "Neural Network Toolbox™ 6" User's Guide

[3] Trendowicz A. and Jeffery R., "Software Project Effort estimation – Foundation and Best Practices Guidelines for Success", Springer Cham Haidel berg, New York, 2014

Articles

[4]Basha S. and Dhavachelvan,"Analysis of empirical Software effort estimation models" International Journal of Computer Science and Information Security, Vol. 7, No. 3(2010)

[5] Dhiman A. , Diwakar C. , " Optimization of COCOMO II Effort Estimation using Genetic Algorithm", American International Journal of Research in Science, Technology, Engineering and Mathematics, ISSN: 23283580, 2012

[6] Jodpimai P.,"Estimating Software Effort with minimum features using neural functional approximation", Advanced Virtual and Intelligent Computing (AVIC) Center Bangkok, Thailand (2012)

[7] Khan Muhammad Waseem and Qureshi Imran, "Neural Network based Software Effort Estimation", International Journal of Advanced Networking and Applications Volume: 05, Issue: 04, (2014) ISSN: 0975-0290

[8] Nagar C. and Dixit A., "Software Efforts and Cost Estimation with a Systematic Approach" Journal of Emerging Trends in Computing and Information Sciences, Volume 2 No.7 (JULY 2011) ISSN 2079-8407

[9] Ramakrishnan N., "Analysis of effort prediction as an influencing parameter of success in Software development projects", International journal of Advanced Research in Computer Science and Software Engineering Volume: 05, Issue: 02, (2015) ISSN: 2277-128X

[10] Sandhu P., Prashar M., et.al, " A Model for Estimation of Efforts in development of Software Systems", World Academy of Science, Engineering and Technology, 2009

[11] Sharma V., Rai S., Dev A., "A Comprehensive study of Artificial Neural Networks" International journal of Advanced Research in Computer Science and Software

[12] Shepperd M. and C. Schofield, "Estimating Software Project Effort Using Analogies," Proc. IEEE Transactions on Software Engineering, IEEE Press, Nov. 1997

[13]Srivastva S., Tripathi K., "Artificial Neural network and Non Linear Regression: A comparative Study", International Journal of Scientific and Research publications, Volume: 02. Issue 12, Dec 2012, ISSN 22503153

[14]Subitsha P., Kowski J., "Artificial Neural Network Models For Software Effort Estimation". International Journal of Technology Enhancements and emerging Engineering Research, VOL 2, ISSUE 4 76 ISSN 2347-4289

[15] Zainnudin Z. and Pauline ONG, "Functional Approximation using Artificial Neural Networks", International Journal of Systems Applications, Engineering and Development Volume: 01, Issue: 4 (2007)

WebPages

[16] http://cs.stanford.edu/ people/ eroberts/ courses/ soco/ projects/ 2000 – 01/ neural_networks/ Applications/ index.html

[17] NASA60 data set,
http://promise.site.uottawa.ca/SERepository/datasets/cocomonasa.arff

[18] NASA93 data set,
http://promise.site.uottawa.ca/SERepository/datasets/cocomonasa_2.arff

[19] Practical Software Project Estimation 3rd edition (McGraw Hill), Effort Estimation Techniques

[20] http://sfb649.wiwi.hu – berlin.de / fedc_homepage/ xplore/ ebooks/ html/ csa/ node 172.html

[21] http://tex.stackexchange.com/ questions/ 132444/ diagram – of – an – artificial – neural - network.

CHAPTER 7
APPENDIX

A. **Abbreviations**

PM	Person Months
FPA	Function Point Analysis
IT	Information Technology
COCOMO	Constructive Cost Model
KLOC	Kilo Lines of Code
SD	Standard Deviation
RE	Relative Error
MRE	Magnitude of Relative Error
MMRE	Mean Magnitude of Relative Error
MSE	Mean Square Error
PRED	Percentage of Relative Error Deviation
LMA	Levenberg Marquardt Algorithm