Major Project Report

Maintenance Prediction using Machine Learning Techniques

Submitted in partial fulfilment of
The requirement for the award of the degree of

Master of Technology
In
Software Engineering

Submitted by
SHUBHAM SHUKLA
(2K16/SWE/15)

Under the supervision of
Dr. Ruchika Malhotra
Associate Professor
Department of CSE

Department of Computer Science and Engineering
DELHI TECHNOLOGICAL UNIVERSITY
Bawana Road, Delhi – 110042

July 2018

CERTIFICATE

I, hereby certify that the Project titled “Maintenance Prediction Using Machine Learning Techniques” submitted By SHUBHAM SHUKLA, Roll number: 2K16/SWE/15, Department of Computer Science and Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of project work carried out by the student under my supervision. To the best of my knowledge, this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Date: 16-07-2018

Dr. RUCHIKA MALHOTRA

Associate Professor

DECLARATION

I, SHUBHAM SHUKLA, 2K16/SWE/15 a student of M.TECH (Software Engineering) declare that the project Dissertation titled “Maintenance Prediction Using Machine Learning Techniques” which is submitted by me to Department of Computer Science and Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma, Fellowship or other similar title or recognition.

Place: DTU, Delhi SHUBHAM SHUKLA

Date: 16-07-2018 (2K16/SWE/15)

ACKNOWLEDGEMENT

I am very thankful to Dr. Ruchika Malhotra (Associate Professor,Computer Science Eng. Dept.) and all the faculty members of the Computer Science Engineering Dept. of DTU. They all provided immense support and guidance for the completion of the project undertaken by me. It is with their supervision that this work came into existence.

I would also like to express my gratitude to the university for providing the laboratories, infrastructure, test facilities and environment which allowed me to work without any obstructions.

I would also like to appreciate the support provided by our lab assistants, seniors and peer group who aided me with all the knowledge they had regarding various topics.

SHUBHAM SHUKLA

MTECH SWE

2K16/SWE/15

ABBREVIATIONS

PCA	**P**rinciple **C**omponent **A**nalysis
ANN	**A**rtificial **N**eural **N**etwork
SVM	**S**oftware **V**ector **M**achine
PRED	**P**rediciton
MMRE	**M**ean **M**agnitude of **R**elative **E**rror
UIMS	**U**ser **I**nterface **S**ystem
QUES	**Q**uality **E**valution **S**ystem
NNEP	**N**eural **N**etwork **E**volutionary **P**rogramming
GRNN	**G**eneral **R**egression **N**eural **N**etwork
LOOCV	**L**eave **O**ne **O**ut **C**ross **V**alidation
SDLC	**S**oftware **D**evelopment **L**ife **C**ycle
LCOM	**L**ack of **C**ohesion of **M**ethods
DIT	**D**epth **O**f **I**nheritance
NOC	**N**umber **O**f **C**hildren
MPC	**M**essage **P**assing per **C**lass
NOM	**N**umber **O**f **M**ethods
DAC	**D**ata **A**bstraction **C**oupling
Pred	**P**rediction
MSE	**M**ean **S**quare **E**rror

ABSTRACT

Maintenance of software is a repetitive stage in the software development life cycle. It starts just after the software product is deployed to the customer and this phase ends when the product is no longer in use or its been outdated. There are different exercises carried out in software maintenance phase, for example, the addition of advance feature, deletion of unwanted feature, error correction, adaption to new condition and so forth. Maintainability of software is the quality property of the product software which decides the path with which these adjustments or modifications can be performed to give better outcome.

The principle target of this report is to analyse machine learning technique, its use along with genetic algorithm particularly with Ward neural network or NNEP GRNN(Neural Network Evolutionary programming) for maintainability prediction of software and tried to improve the performance and efficiency measures obtained from previous studies using different machine learning techniques but with improvised approach using object oriented metrics only, Here In this work the datasets which are used is same as other study because it is more accurate and useful for software maintenance and hence we follow the same datasets and obtain the result with better efficiency. This procedure is connected to appraise practicality on two diverse case studies, that is Quality Evaluation System(QUES) and User Interface System (UIMS). In this approach, for updating the weight amid learning phase, genetic algorithm is utilized. Later, We plot graph and table and depict its performance with various other machine leaning techniques such as General Regression Neural Network, Decision table/tree etc. Neural network along with evolutionary/genetic programming is analysed as the best model for prediction than comparing with other ML techniques like ward neural network (i.e neural network with different slabs in the hidden layer). Neural network along with genetic algorithm i.e the hybrid approach that we used with PCA is capable in reducing MSE (Mean squared error) and MMRE (maximum magnitude of relative error) into a less error value so that this approach can be used further for new data or software industry as well.

CONTENTS

Certificate i

Declaration ii

Acknowledgement iii

Abbreviation iv

Abstract v

1 Introduction 1

2 Related work / Litreture Survey 5

3 Research Methodology 10

3.1 Overview. 10

3.2 Machine Learning Techniques. 10

3.2.1 Linear Regression. 10

3.2.2 Software Vector Machine. 12

3.2.3 Decision Tree. 14

3.2.4 Neural Network. 15

3.2.5 Genetic Algorithm. 17

3.2.6 GA-NN Hybrid Approach. 19

4 Research Background 20

4.1 Overview. 20

4.2 Independent and Dependent Variables. 20

4.3 Empirical Data Collection. 22

4.4 Prediction Accuracy Measures. 23

4.5 Cross Validation. 25

4.5.1 Leave One Out Cross Validation. 25
4.6 Tools & Libraries Used for Result Calculation. 26
4.6.1 Libraries Used. 26
4.7 Classification of UIMS Dataset. 27
4.7.1 Linear Regression. 27
4.7.2 Decision Tree. 28
4.7.3 General Regression NN. 29
4.7.4 Neural Network (Ward NN) 31
4.7.5 GA-NN Approach. 33
5 Results and Discussion 35
5.1 Machine Learning Techniques. 35
5.2 Graph for MMRE Error. 37
5.3 Squared Error Graph. 38
5.4 Pred(25) and Pred(30) Values. 39
5.4.1 Pred(25) Values. 39
5.4.2 Pred(30) Values. 40
5.5 Threats to Validity. 40
6 Conclusion and Future Work 42
References 43

LIST OF FIGURES

3.1	Linear Regression	11
3.2	Software Vector Machine	13
3.3	Decision Tree	14
3.4	Artificial NN	16
3.5	Ward NN	17
3.6	Genetic Algorithm	18
3.7	GA-NN Hybrid Approach	19
4.1	Result of Decision Tree	28
4.2	Training and Testing of GRNN	29
4.3	Result of GRNN	30
4.4	Training and Testing of Ward NN	31
4.5	Result of Ward NN	32
4.6	Training and Testing of GA-NN	33
4.7	Result of GA-NN	34
5.1	MMRE Error for ML Technique	37
5.2	Squared Error for ML Technique	38
5.3	Pred(25) Values for ML Technique	39
5.4	Pred(30) Values for ML Technique	40

LIST OF TABLES

4.1	Descriptive Statistics of UIMS Dataset	22
4.2	Descriptive Statistics of QUES Dataset	23
5.1	Result of all ML Techniques, Genetic Algorithm, Hybrid Approach	36

CHAPTER 1

INTRODUCTION

Maintenance of Software plays a more critical part in arranging and executing new programming items. Hence any new information which can better depict or propose an answer for speedier and most practical determination of issues in software maintenance is exceptionally helpful. As indicated by Software Life Cycle Procedures Standard, the software Maintenance is characterized as an essential process in Software life cycle. Along these lines assessment of programming support undertakings, which is perceived as the most critical and most exorbitant part of the software life cycle, can altogether add to the effectiveness of software management maintenance.

Software maintenance is a methodology of adjusting a product framework or module after dispatch to rectify mistakes, grow execution or different properties, or acclimate to a changed conditions. We can state that Software maintenance can devour as much as 92% of the general effort consumed on a framework in software life-cycle. In order to predict software maintenance, one of the overall best instrument is prediction by utilizing Maintainability Index, keeping in a view that software applications will be more viable. We can likewise go for risk analysis phase to compute cost factor analysis. In this manner the design team might be call for planning a more reliable outline while anticipating the Maintainability Index in early period of development of the software. Those support costs are regularly viewed as the cost of the product engineers in upkeep in light of the fact that those developers must be not just experienced in software development yet in addition to have profound information about the area the specific software product. Prediction precision measures are then

utilized for the assessment of those systems which depend on the machine learning models.

Regarding software Maintenance prediction, recent studies of Maintenance provide a detailed review of several studies. The prediction techniques are classified into 2 general categories

- Expert analysis: This sort of strategy is not been used generally. This method assesses the software maintenance based on expert involvement for similar tasks. The precision of such estimates exceptionally relies upon the ability of the expert and to what degree expert involvement associated with which new project agrees. expert judgment is not effective for maintenance prediction as requirement of each and every software varies unexpectedly.

- Machine Learning: The approach of machine learning is widely used in maintenance prediction, all previous studies concluded with the use of machine learning approaches and optimized them. Some stochastic search algorithm like genetic algorithm are often used with machine learning algorithm in order to provide better result.

As per various survey papers from last decades, Optimization of various machine learning approaches targeted the most. Expert analysis could perform better in the field estimation of effort but it does not worth for maintenance prediction and hence researchers and practitioners are working in maintenance prediction estimation keeping their mind on machine learning approaches and stochastic algorithm and their hybrid approaches so as to provide best among all approaches. In this thesis we also concentrate on machine learning approaches including General regression, Linear regression, Software vector machine, Decision Tree, Neural network and hence for

optimization of result we also discussed hybrid approach which is the combind performance of neural network along with evolutionary algorithm/genetic algorithm.

There are numerous machine learning techniques are proposed in the studies like linear regression, Support vector machine, TreeNet, GRNN, ANN etc. but in order to best predict the maintenance it is very difficult to find out the best approach of machine learning out of all the proposed techniques. Hence deep study is required to find out the best approach and hence to provide best result and hence this leads to following research question:

- RQ1: Which is the best machine learning approach used for maintenance prediction? In this research question, we determine the efficiency of most of the machine learning techniques and the techniques which suits best for prediction accuracy measures are termed to be the best machine learning approach.

- RQ2: How to utilize machine learning techniques with stochastic search algorithm? In this research question, We used genetic algorithm with ANN and provide improvised result to the individual machine learning approach.

- RQ3: What is the time constraint of machine learning approach and hybrid approach with stochastic algorithm? In this research question, we compare the time complexity of each and every techniques we used.

The objective of this report is to compare all the machine learning approaches including linear regression, Decision tree, Software vector machine and then compare this techniques with the hybrid approach i.e. neuro-genetic algorithm of NNEP(Neural network evolutionary programming approach). For this, various machine learning are used and all are implemented on same dataset. For testing and training of neural network we used two famous datasets that is Quality Evaluation System(QUES) and

User Interface System (UIMS) consist of 71 and 39 classes respectively and hence consider only object oriented metrics that we detailed discussed later in the chapter including its method as well in a class. We later applied LOOCV i.e leave out out cross validation where only one data attribute is used for testing and rest are used for training and hence in each iteration testing and training data set varies with each iteration.

Rest part of thesis consist of various chapters: In second chapter, we explained the work related to this study of maintenance prediction in short. In third chapter different Machine learning methods/approaches for software maintenance prediction and the research methodology have been discussed along with the explanation of genetic algorithm/evolutionary procedure. In fourth chapter, entire research background on maintenance prediction have been talked about. We explained the famous datasets that we used and metrics associated with them, After explanation of datasets and metrics cross validation methods and feature selection methods are explained along with prediction accuracy measures and later we explained the machine learning API used for training the neural network. In fifth chapter the result is explained after computing various machine learning algorithm. Threats to validity and future related to approaches towards maintenance prediction work have been explained in chapter sixth and then all the references related to the given study have been mentioned.

CHAPTER 2
RELATED WORK/LITRETURE SURVEY

The most costly phase is software maintenance in software development life cycle (SDLC). It is the last period of the lifecycle and starts simply after the release of the software or the product. During the whole duration of the product/software, it gets altered consistently in view of advancement in the innovation and competition among industries. According to the study by Pressman and Friedman [5], most part of the total efforts are being used in maintenance phase of the current framework. As indicated by his study, some part of the work is to redress faults and some is utilized for the portability and the significant part is utilized to implement the modification made in the need.

Hence as considering the importance of maintaining software in software development life cycle, Many scholars and researchers in this field have worked towards its importance so that cost of this maintenance phase could minimize. Maintenance prediction is the basic requirement in order to deal with the alteration in the software because of competition among industries and hence previous studies basically consider maintenance prediction as the tool to deal with the high cost of this phase in software development life cycle. Software maintainability is the important attribute of the product or software which decides the simplicity with which these alterations can be performed. If we can predict the maintainability precisely, cost and time related with the activity of maintenance can be exceptionally diminished. Hence this chapter discussed the some un-improvised work similar to this thesis and all the previous work related to maintenance prediction .

Research paper by Tong-Seng Quah, Mie Mie Thwin [3] also used machine learning i.e neural network for prediction if quality using object oriented paradigm i.e. metrics

and concentrate mainly on reliability and neural network using QUES as the dataset having metrics are DIT,MPC,RFC etc., but at last they also concluded the use of

GRNN i.e. general regression neural network perform better then simply using neural network for quality prediction[4], This study is used for estimating both the maintenance and reliability. They also used the Ward neural network and general regression neural network and conclude general regression neural network performed better than ward neural network.

As discussed before machine learning and evolutionary algorithm gained a lot of attention towards prediction in software engineering wither it would be effort prediction or maintenance prediction. Study from 2010 by A. kaur and R. Malhotra in their study [9] also use the soft computing approaches for prediction of software maintenance. This work performed better then all the preceded works as this is the hybrid approach which combine ANN and FIS i.e. artificial neural network and Fuzzy inference systems, they also used the famous dataset proposed by li and henry that is QUES and UIMS and maintenance is measured using change in number of lines per class, it could be deletion or addition of a line. UIMS and QUES consist of 39 and 71 classes respectively which we used for our technique, They normalized the data using z-score normalization. They formulate the architecture as one hidden layer in which five neurons are setup using hyperbolic tangent as activation function in hidden layer. Later they used MARE, MRE in order to estimate the models variability towards overestimation and underestimation and then p-value I order to find out the no correlation hypothesis and they concluded the working of ANFIS was better than GRNN and hence evolution of soft computing for maintenance prediction.

Paper by A. chug and R. Malhotra in their study [11] showed comparison of various techniques and evolve GMDH as best among all i.e. Group method of data handling, This method used for unstructured system in which high order input output is required.

GMDH is an algorithm which is not based on regression analysis but heuristic. It is called neural network in polynomial way and output is also not that linear function but polynomial which is the requirement of unstructured data it is not fully dependent on

the working of feed forward neural network, It can be used for reliability in software analysis or software prediction including maintenance and effort. Additionally, it was assumed that better cohesion and advantage to industry would be picked up, if a similar model can adequately predict both maintainability quality and practicality of recently created software since they work to accomplish the in general objective of software quality. The target of this study was to use GMDH model alongside two different models GA & PNN. At last they conclude over their research that Additional research is arranged trying to join GMDH model with other information mining methods to create prediction models which can evaluate the software maintainability more precisely with minimum error in precision.

Other than prediction method sometimes other factors played a vital role in software maintainability prediction, It could be the usage of datasets or metrics related to the maintainability. Similarly in the study of H. sharma and A. chug in their study [1 , In this study the software metrics are compared i.e. dynamic metrics and static metrics, software metrics assist us with making significant estimates for products of software and guide us in taking specialized choices like budget estimation, cost estimation, quality confirmation testing, debugging of software, software performance enhancement, and ideal undertaking assignments. Numerous design metrics have proposed in study to quantify different builds of Object Oriented (OO) paradigm, for example, coupling, class, cohesion, inheritance polymorphism and data hiding and utilize them further in deciding the different aspects of software quality. Basically, the usage of metrics like static have found to be deficient for current OO software because it has run time polymorphism present in it, template class and methods dynamic binding unexecuted code that are left out due to particular input conditions. This gap

gave a sign to center around the utilization of dynamic metrics rather than conventional static metrics to catch the software attributes and further use them for maintenance prediction [15]. As the dynamic metrics are more exact in catching the execution

conduct of the software framework, in the current empirical investigation with the utilization of open source code, they validate and confirm the predominance of dynamic metrics over static metrics.

Along with machine learning, Genetic algorithm also gained a lot of popularity from last one decade especially in the field of prediction and estimation, and hence many researchers and scholars are continuous working to improve genetic algorithm or evolutionary algorithm. Lot of studies have been done in previous years and one of the study namely statistical comparison of modelling methods by A. kaur [12] also explains the improvised result using genetic algorithm, This study also worked on different datasets than the datasets previous studies used, In this study four open source software are used and its different version are analysed three years and hence all the open source software has different number of classes. These open source softwares are analysed for maintenance prediction using genetic algorithm, decision table, Radial basis function neural network(RBFN), Bayes net and sequence minimal optimization(SMO, after the proposed work it was concluded that genetic algorithm performs good number than proposed machine learning algorithm, genetic algorithm scored very less error in terms and MAE and RMSE and friedman test ranked genetic algorithm first. It also compared the time complexities of several algorithms and in terms of time complexity genetic algorithm did not perform well and almost all the machine learning algorithm proposed in that study outperformed evolutionary algorithm in terms of time complexity using Friedman test.

Considering time complexities as the factor of concern, A study based on SVM and other kernel method was proposed in their study under machine learning [18], In this

study web service was taken as to depict the usage of SVM for maintainability prediction, According to them SOC is the basic foundation of present software development. The nature of SOC can be best evaluated by the utilization of software metrics . In this study, numerous object oriented software metrics are considered so

as to plan a model for maintenance prediction of SOC factor. Additionally support vector machine with various kind of kernels are considered for maintenance prediction of SOC paradigm. This paper additionally centers around the importance of feature selection algorithm, for example, univariate, cross, logistic, rough-set and principal component analysis. The results demonstrate that, SOC maintainability can be anticipated by use of different object oriented metrics. The outcome additionally showed that, it is conceivable to find a little subset of object oriented metrics out of all available object-oriented metrics, that empowers maintainability prediction with more accuracy.

Overall study concentrate on artificial intelligence and genetic algorithm and hence hybrid approach for this is mandatory. Various study also worked over the combined approach of genetic algorithm and neural network. Research paper by Lov Kumar and Santanu ku.[19] worked on the hybrid approach of neural network and genetic algorithm. The majority of the maintainability prediction models in studies depend on method for example, basic neural network and regression analysis. In this paper, three AI techniques for example, FLANN, GA, PSO and CSA are discussed. These three AI methods are applied for maintenance prediction and use two famous datasets that is Quality Evaluation System (QUES) and User Interface System (UIMS). This study likewise focussed on feature selection feature as well, for example RSA and PCA.

CHAPTER 3
RESEARCH METHODOLOGY

3.1 OVERVIEW

In this study we proposed the approach which is based on machine learning and evolutionary algorithm and later hybrid approach also proposed which is the hybrid of machine learning as well stochastic algorithm i.e. evolutionary algorithm. Ward Neural network is considered as a neural network having feed forward mechanism and genetic algorithm is used with this neural network and it provided improvised result than simply using machine learning approach for maintenance prediction. Here for machine learning algorithm we used linear regression, linear regression using outlier function, Decision tree, General regression neural network(GRNN), Ward neural network and later evolutionary algorithm.

3.2 MACHINE LEARNING TECHNIQUES:

3.2.1 Linear regression

Linear Regression is a process which examine the connection between independent and dependent variables which is indicated as Y and X respectively. Linear regression is a technique of the factual method which is utilized by dependent varaibles for data regression. Independent variables may contain categorial or continuous values while dependent variables only have continuous variables. We can state in a different way, LR is a method, utilized for prediction of the Y variable based on the values of X variables also, It is likewise utilized as a prediction of continuous quantity.

Fundamentals

To start with linear regression, It is mandatory to have some basic measurements ideas. Such that:

- Correlation: Correlation describe the relation between two variables and its value lies between [-1,+1], It is denoted by 'r'.
- Variance: It measures how the dataset is spread, It is denoted by " "
- Standard deviation: It also measure how the dataset is spread but having different form i.e form is " "
- Normal distribution
- Residual: It is nothing but Error which is simply calculated by subtracting predicted with actual value {Actual value-predicted value}.

Linear Regression Line

If we have to apply linear regression, our principle objective is to fit a line which is nearest to most of the points by appropriation in the sample space. In this manner we continuously try to diminish the separation (i.e. Error) between data points and fitted line.

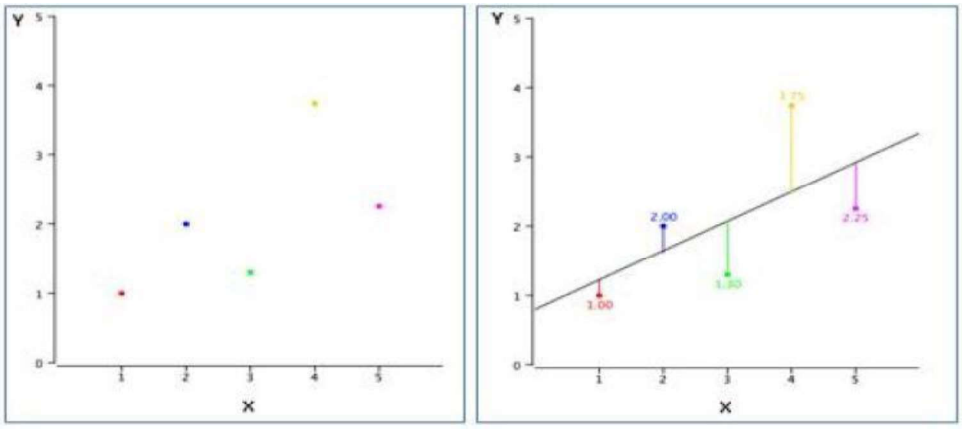


Figure 3.1: Linear Regression

In the right figure, we have shown a line which is called an approximate line which is based on the points drawn in the left figure. The line drawn in the right figure explain the relation between y and x axis. In order to find or draw the approximate line, we always follow linear regression. For the single value of the independent variable (X) and dependent variable(Y). The relation shown above can be equated in linear equation as:

$$\mathbf{Y = B0 + B1X}$$

Where,
0= intercept y-axis and it's a constant term.
1= relationship coefficient between Y and X.

- B0 and B1 parameters = these are two parameters which specify a line and it is estimated using datasets.
- Y1,Y2,….X1,X2,…this are the known values and for them least square criterion is used.

As an intercept at y-axis passing through approximate line, B0 is used. It is denoted as bias in machine learning which added to the offset for all the prediction which we made. The slope of the line here is showed using B1, it also explain how the value of X translated into the value of Y before the addition of bias into them.

3.2.2 Software Vector Machine

SVM is a method which is supervised in nature that analyse the data and sort it out into one of the two categories. SVM is one of the best classifier among those which

are sort of linear mathematical behind SVM [3] [4]. Software vector machine has a clever way to prevent overfit and hence it worked with a relative larger number of features without requiring or doing too much computation. It is a good alternative to bayesian learning because of active computational process. SVM fundamentally a two-class classifier. In order to choose the best possible prediction we define the distance between support vector and the hyperplane should be as far as possible, where support vectors are the points in extreme position at the dataset and maximum distance to the support vector reside in the hyperplane.

Advantages of SVM over another classifier

- SVM is a high dimensional input space
- It can sparse document vectors and helpful for regularization of parameters. SVM has a clever way to prevent overfit It is a good alternative to bayesian learning because of active computational process.

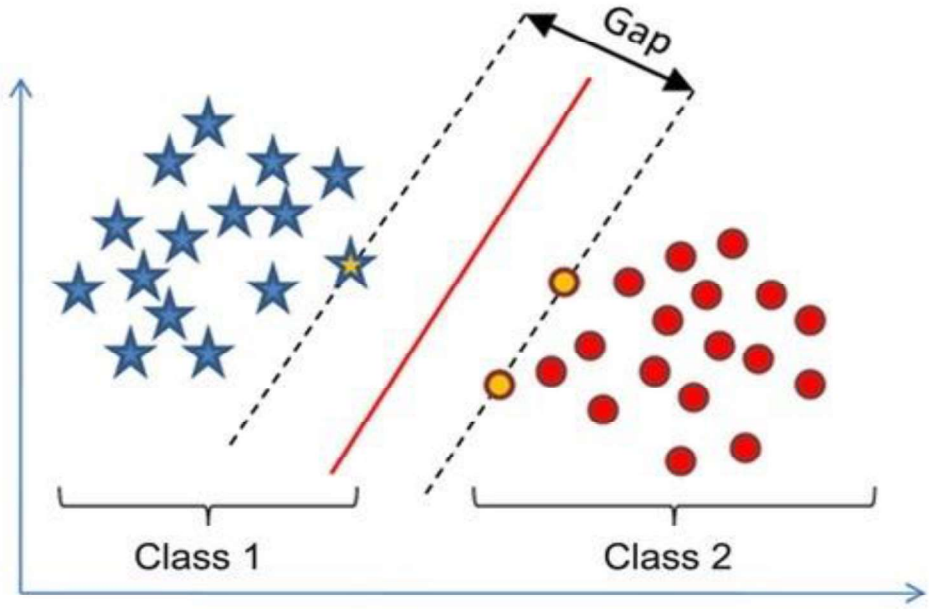


Figure 3.2: Software Vector Machine

3.2.3 Decision Tree

Decision tree is said to be a machine learning techniques which is based on classification problem, In order to classify we can use decision tree. Decision tree has a shaped which is similar to tree like structure diagram which is utilized to find a course of action. Each branch of the decision tree used to represent reaction, possible occurrence and decision. Decision tree can solve two types of problem or decision tree is basically useful in making two types of decision: One is classification and another is Regression. Example of classification problem is to discriminating between three types of flowers based on certain features or we can called this decision tree as classification tree. Example of regression is used when target variable is continuous or numerical in nature, Each independent variable is used in order to fit a regression based model to the desired variable [8] and we called this type of tree as regression decision tree.

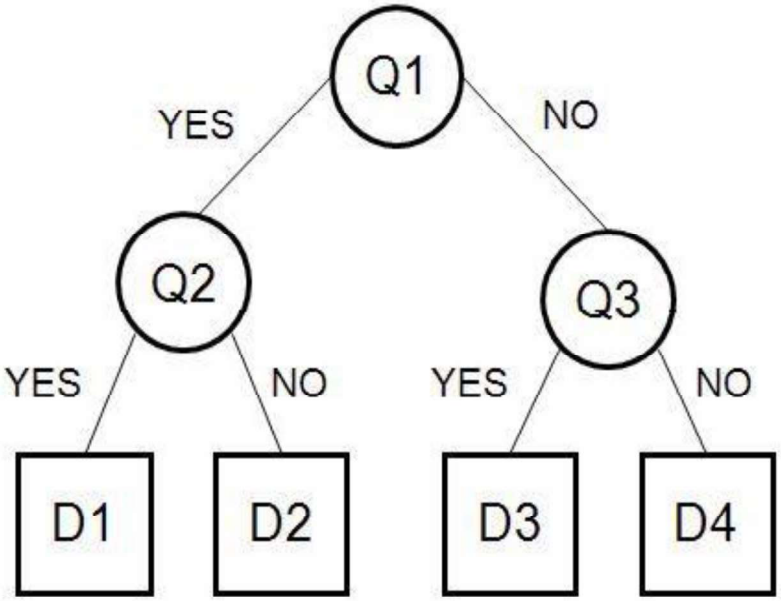


Figure 3.3: Decision Tree

Advantages of decision tree

- Simple to visualize, understand and interpret.
- Less effort is required for data preparation.

3.2.4 Neural Network

Artificial neural network(ANN) are processing systems propelled by the biological neural systems that constitute living brains. Such systems learn or it continuously enhance its execution to do tasks. by thinking about examples, generally without task-specific systems. Neural Network explains artificial neurons called perceptron. It is based on training and testing/using mode.

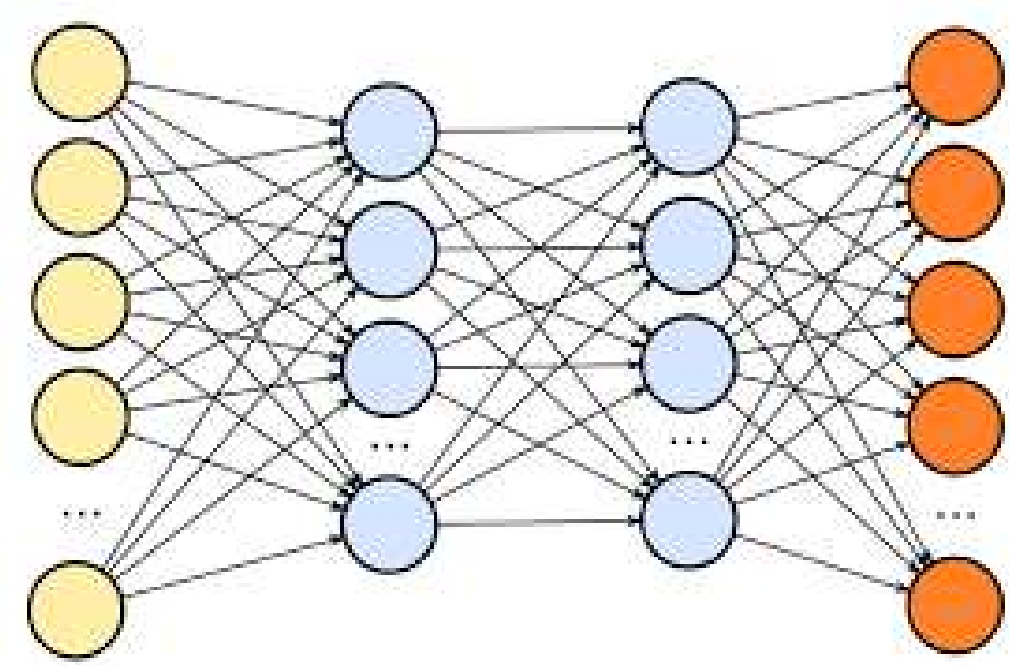
In the training mode, the neuron can be trained using particular input patterns or with the training datasets and In the testing the output is finalized on the basis of detected output matched with the available datasets. It consist of input layer, output layer and the hidden layer. In the hidden layer activation function is used which could be step function, sigmoid function, sign function [4]. Weights and threshold weightage also given accordingly.

If output is below threshold then neural network need to be improved using different activation function and layer arrangement in input, output and hidden layer. This arrangement is known as multilayer perceptron arrangement.

In our project we used ward neural network as an artificial neural network. This neural network is on back propogation network [5] that has different slab arranged in the hidden layer. This slab arranged system called feature analyser. Slab is nothing but the

arrangement of neurons in the hidden layer, having different activation function is used which helps in different visualization of data.

We used three slabs with different activation function like hyperbolic, Gaussian function and Gaussian complement respectively and later we compared our neural network with General regression neural network. We used here is the ward neural network which is based on backpropagation algorithm having three slabs in the hidden layer. Each slab is having different activation function or can say each neuron in the hidden layer having different activation function.



Input layer Hidden layer output layer

Figure 3.4: Artificial Neural Network

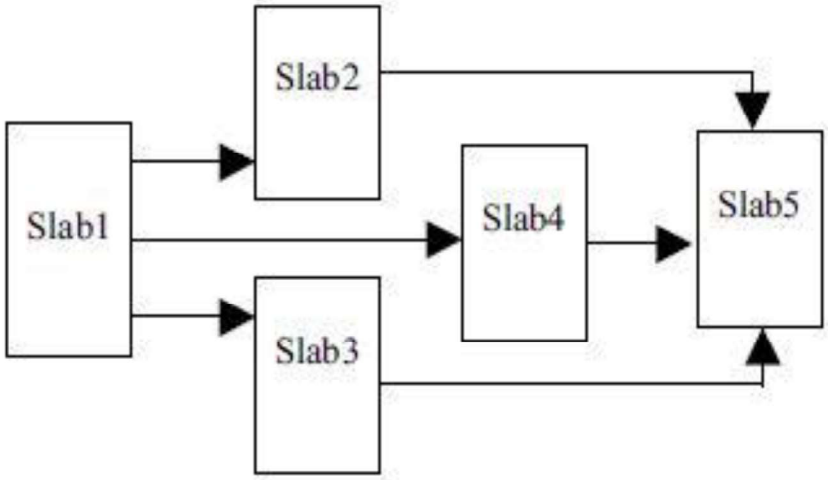


Figure 3.5: Ward Neural Network

3.2.5 Genetic algorithm

Evolutionary algorithm is an inquiry based calculation and based with respect to mechanics of natural determination and common genetics qualities. It depends on Darwinian hypothesis [17] and it invigorates the procedure of advancement. evolutionary algorithm is used as a part of search, improvement and machine learning. The objective of genetic algorithm is to continue enhancing execution to achieve some ideal focuses. genetic algorithm utilizes stochastic progress principles and it doesn't use deterministic standards. genetic algorithm does not work with only parameters, but it worked with the parameter coding. It works over the number of population in focuses however not with a solitary point. It utilizes objective work data and stochastic function information and not deterministic tenets. Fundamental genetic algorithm has populace of 2n to 4n trail solution is utilized, where n is the quantity of factors. Every possible solution generally denoted by a string of double variables which relates to the

chromosomes in evolutionary algorithm. The string length utilized as a part of it can be made large enough to accomplish any coveted wellness of estimate and consequently any desired exactness can be acquired or accomplished. After trial arrangements are chosen, another generation (a new arrangement of strings) is delivered by choosing, utilizing stochastic mechanism, The "fittest" parent to create "children" from among the trail arrangement.

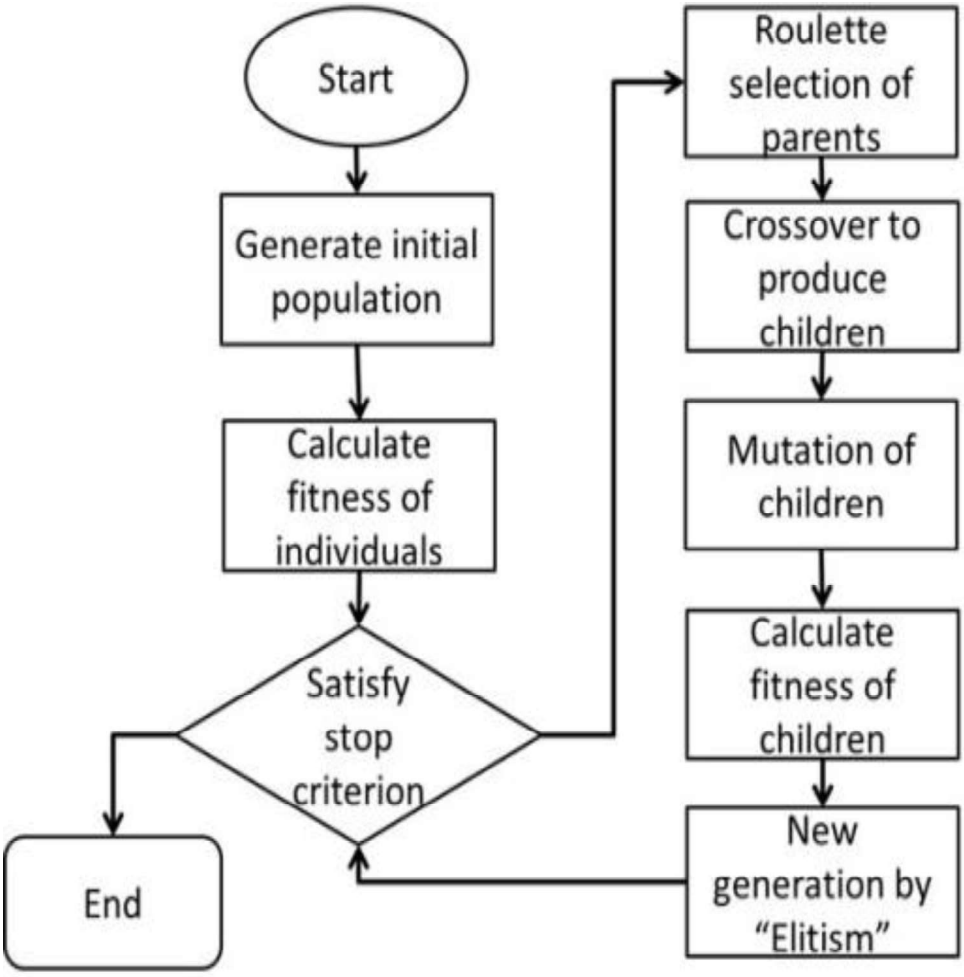


Figure 3.6: Genetic Algorithm

3.2.6 GA-NN Hybrid Approach

In this approach, for updating the threshold and weight during initial or learning phase, genetic algorithm is used and neural network for estimation [19] [21]. In our project We used ward neural network along with evolutionary algorithm in order to obtain better result. This approach basically works with the following points:

- Random population of 'n' chromosomes is generated out of which each weight is extracted.
- For training the network, weight is used as input.
- If stopping criteria does not met then Min fitness value chromosome is replaced with Max fitness value chromosome and again crossover is performed, and again weight set is extracted which again the previous task until stopping criteria met.

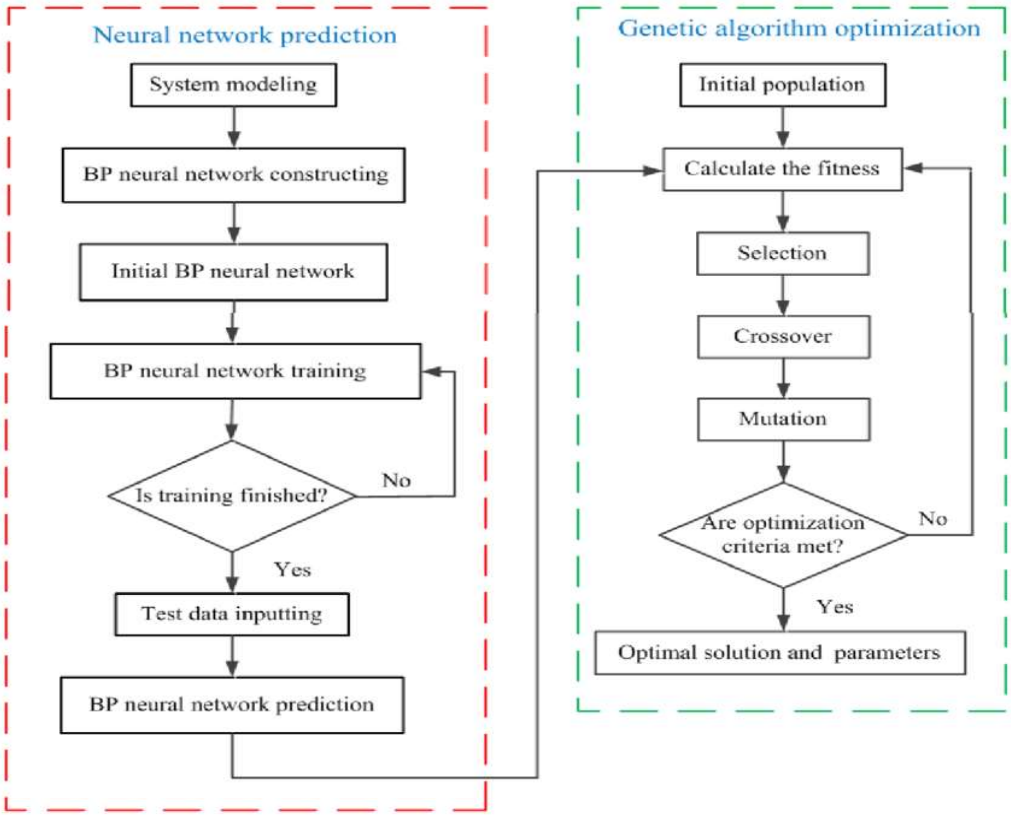


Figure 3.7: GA-NN Hybrid Approach

CHAPTER 4
RESEARCH BACKGROUND

4.1 OVERVIEW

This chapter concern over the selection of independent and dependent variables and the techniques that are useful for validation of machine learning techniques. For maintenance prediction we paid attention towards object oriented parameters such as cohesion, coupling, complexity, abstraction and inheritance which are all the part of used datasets. We are using C&K metrics which is generally used by practitioners. This metric suite contains various dependent variables and various independent variables and later we calculated the "change" variable in terms of addition and any modification in LOC.

4.2 INDEPENDENT AND DEPENDENT VARIABLES

Many object oriented metrics are selected from the used metrics as independent variables like

Cohesion, coupling, abstraction, inheritance and complexity like WMC, DIT, NOC, LCOM, LOC etc mostly of which are proposed by Chidamber et al. [22] and also by Li and Henry [23], And using the same metrics suite we have categorized some dependent variables as well and change or modification in lines of code like addition, deletion or any modification.

Some of the metrics are defined as:

- WMC (weighted method per class) = It is the summation of all the complexities obtained from the different classes either local or global classes.

$$\mathbf{WMC} = \sum_{i=1}^{n} Ci$$

Where Ci is used to refer the complexities of each classes

- DIT (Depth of Inheritance Tree) = Here root class is considered as zero and it is the depth from the root class

- RFC (Response For a class) = It is the collection of local and non-local functions in a class or can say local or non-local method in a class.

- NOC (Number of Children) = It calculates the dependency of subclasses with the main class i.e this metrics is used to count the number of dependent classes i.e the child classes of the main class.

- LCOM (Lack of cohesion of methods) = It decides the count of disjoint subsection of local functions in a class. The function said to be disjoint set function.

- MPC (Message passing coupling) = It counts the calling function from the class or messages require that sent out from the class.

- DAC (Data abstraction coupling) = Object or member of another class used within a class.

- NOM (Number of Methods) = Number of member or function in a class.

- SIZE1 (Number of properties) = It counts the number of local function or method in a class

- SIZE2 (Number of properties) = It counts number of local function and the data attributes in a class.

- CHANGE (Changed line in a class) = It counts the deletion and addition of lines in a class, modification done in a class and all these modification has a count of two.

4.3 EMPIRICAL DATA COLLECTION

In our study we are using one of the most popular datasets used for object oriented maintainability prediction proposed by Li and Henry [23] namely QUES and UIMS datasets. These datasets are generally used by most of the researches and practitioners in their previous studies, we also used the same datasets so that it would be helpful in order to compare the results with those studies in which the same datasets has been used before. UIMS(User interface management system) contain 39 classes and QUES(Quality evaluation system) has 71 classes. Ada were used for implementation of both the systems. These datasets is a collection of independent and dependent variables. Dependent variable is CHANGE and rest are independent variables which are already been explained.

Table 4.1: Descriptive statistics of UIMS Dataset

Metric	Minimum	Maximum	Mean	Standard Deviation
WMC	0	69	11.38	15.90
DIT	0	4	2.15	0.90
NOC	0	8	0.95	2.01
RFC	2	101	23.21	20.19
LCOM	1	31	7.49	6.11
MPC	1	12	4.33	3.41
DAC	0	21	2.41	4.00
NOM	1	40	11.38	10.21
SIZE1	4	439	106.44	114.65
SIZE2	1	61	13.47	13.47
CHANGE	2	253	42.46	61.18

Table 4.2: Descriptive statistics of QUES Dataset

Metric	Minimum	Maximum	Mean	Standard Deviation
WMC	1	83	14.96	17.06
DIT	0	4	1.92	0.53
NOC	0	0	0.00	0.00
RFC	17	156	54.44	32.62
LCOM	3	33	9.18	7.31
MPC	2	42	17.75	8.33
DAC	0	25	3.44	3.91
NOM	4	57	13.41	12.00
SIZE1	115	1009	275.58	171.60
SIZE2	4	82	18.03	15.21
CHANGE	6	217	64.23	43.13

4.4 PREDICTION ACCURACY MEASURES

It is always mandatory to measure the accuracy of models we maid, and hence prediction accuracy measures are used to determine it. As we already discussed that neural network or all other machine learning classifier are used to compare find the expected values. These expected values are compared with actual values using prediction accuracy measure and hence accuracy can be find out. In this study we have used MMRE, MSE, Pred(25) and Pred(30). This prediction accuracy can be calculated using following manner:

- MMRE: This is most common prediction accuracy measurement used by most of the prediction models either for effort model, cost most or our model that is maintenance prediction model. The Mean Magnitude of Relative error is calculated

$$\mathbf{MMRE} = \sum_{i=1}^{k} \boldsymbol{MRE(i)}$$

Where MRE is defined as the ratio between the predicted value or estimated value to the actual value or defined value.

$$\mathbf{MRE} = \frac{\boldsymbol{Actual\ value - predicted\ value}}{\boldsymbol{Actual\ value}}$$

- MSE: Mean squared deviation or Mean squared deviation used for the measurement of prediction accuracy in which average square if actual and predicted values is taken. It is always positive as it is squared prediction accuracy measures. Values closer to zero are considered to be the good result.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

For n predictions that generated from a sample having n data points.
Here Y is the vector of the predicted variable.
Here n is denoted by the total number of data points in a sample.

- PRED(x): It checks the amount of estimated values variation from MRE value or can say average fraction of MREs.

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

If,

$$\mathbf{PRED(x)} = \frac{M}{N}$$

Where x is the specified value.

M is the number of such cases where MRE error is less than or equal to x, total number of datasets represented by N.

4.5 CROSS VALIDATION

4.5.1 Leave One Out Cross validation

Cross validation is useful in order to predict model is overfit or not and this is the helpful to find the best fit model. It is a process to find out the error rate in the model.

It is useful Once the model has been implemented in order to find the error rate, it gives the approximate value of the error rate. Cross validation guarantees the data is useful for new model as well. Our cross validation is Leave one out cross validation.

It is a special case of K-fold validation where K is equals to N, N is nothing but number of data points in the sample. When total number of splits that is going to have is exactly equal to number of data points in a sample called leave one out cross validation, Here we do not have a subsample as a validation sample, here we have one data point and we take one data point as a validation sample and build a model with rest data points and test with holdout data point and get error e1, and again change a data points and find another error e2,and continue to do the same thing and at last we will get as many error as number of data points and average error would be summation of error to the number of data points. It somehow take more computation time because it requires the same iteration as the number of data points.

4.6 TOOLS AND LIBRARIES USED FOR RESULT CALCULATION

Here we simply used the python 3.5 and all the libraries and all the API has to be installed in it, As python language is a good scope for machine learning as it supports almost all the machine learning API and it also useful in the following features:

- Classification
- Selection of attributes
- Visualization
- Technical computational problem
- Widely used in machine learning
- Collection of various API and libraries used in machine learning including Numpy, Scipy, Keras, Tensorflow etc. These all libraries are written in python ans hence using python in machine learning project is nowadays mandatory.

4.6.1 Libraries Used

- ***Tensorflow***: Tensorflow is a open source software library which is used for various machine learning approaches to deal with various ranges of tasks. It was released on 9 november 2015 having apache 2.0 open source license. It can work over linux, macOS, windows, Android, website and it is written in python, C++, CUDA. It is widely require open source software library for various machine learning algorithm including neural network and genetic programming. It provides API for python officially.

- ***Scipy***: It is also an open source python library released in 2001 under community library project. It has widely used in software computing including scientific and technical. It is used for various computational process like optimization of classifier , special functions which are highly computational, FFT signal, interpolation of data

and integration of data. It is presently under BSD licence. There are various sub package present under this library like constants, fftpack, io, linalg, ndimage, signal,

spatial, stats, special, weave etc. It is basically multidimensional data structure under Numpy and hence used for various transformation as well.

- ***Numpy***: Numpy is a library for python language and hence used for huge computational process which is required during classification problem and hence used by various classifier and in machine learning technique like neural network and genetic programming. It is useful for multidimensional array and large computation algorithm to make them easier. It is developed under community project in 2006. It is written in programming language python. The basic function of Numpy is in multidimensional array or can say it is useful for n-dimensional array.

- ***Keras***: It is also open source library written in python. Keras is useful for most for the neural network classifiers as it is developed over tensorflow and hence useful in machine learning projects including neural network. Useful in high computational neural network and evolutionary programming. It was developed in 2015 written in python language under various developers. It is also useful in nowadays in IOS and Android. It is useful for GPU and JVM.

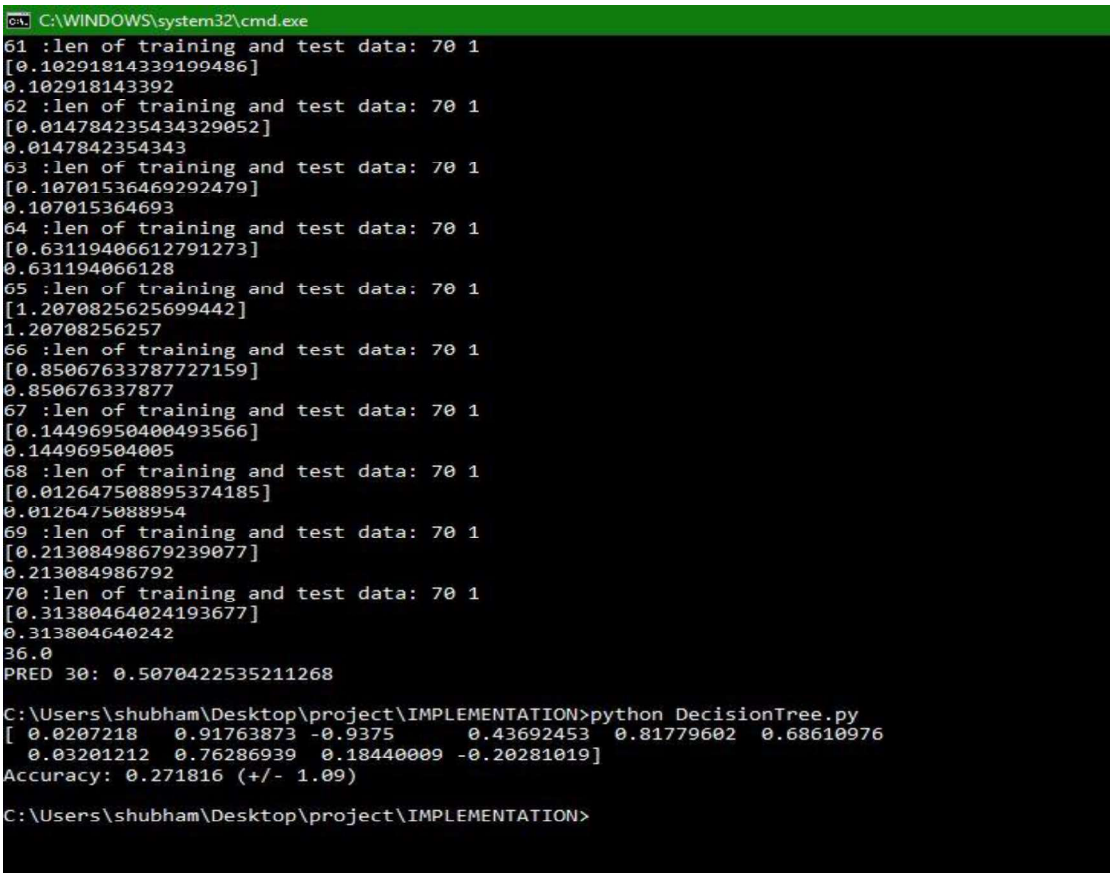
4.7 CLASSIFICATION OF UIMS DATASET

4.7.1 Linear regression:

Here we used UIMS dataset and applied one of the single simple machine learning technique called linear regression. After prediction the value is calculated based on the obtained value. We applied leave one out cross validation here and hence number of

splits here is equal to the number of data points or number of datasets. After classification we find the squared mean error rate is 897.30 and MMRE error is 33%

and hence this is not fully satisfactory as linear regression is not that accurate for huge data and hence it concluded the result with average satisfaction.



4.7.2 Decision Tree

Figure 4.1: Result of Decision Tree

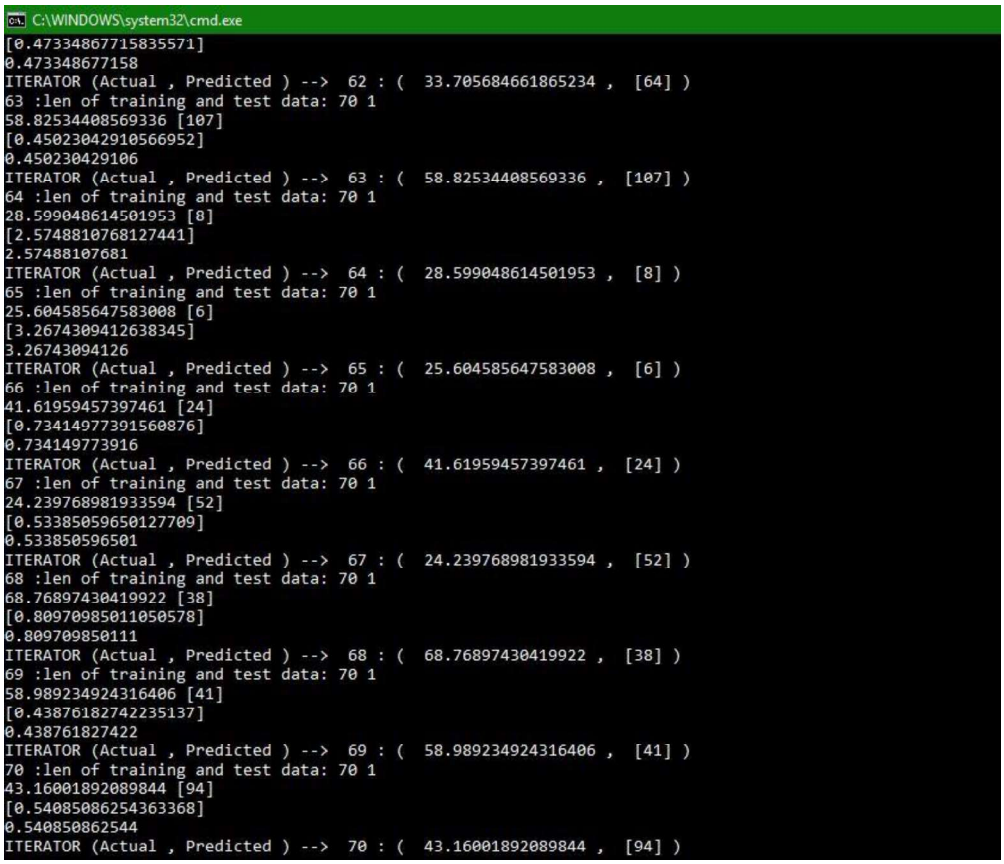
Decision tree for classification is one for the best technique to find out the prediction accuracy. It is used to determine the course of action. Its prediction is better than linear

regression as It is one of the best technique for classification problem. The accuracy obtained by this technique has been shown in the figure given above.

4.7.3 General Regression neural network

General regression neural network requires tensorflow backend, Tensorflow is a open source software library which is used for various machine learning approaches to deal with various ranges of tasks. It was released on 9 november 2015 having apache 2.0 open source license. It can work over linux, macOS, windows, Android, website and it is written in python, C++, CUDA. It is widely require open source software library for various machine learning algorithm including neural networm and genetic programming.

The testing and training of neural network using LOOCV and GRNN having 70



datasets as training and one as testing shown below

Figure 4.2: Training and Testing of GRNN

The output obtained from general regression neural network is good as comparison to linear regression and decision tree and it achieve comparatively less error rate than previous used machine learning techniques.

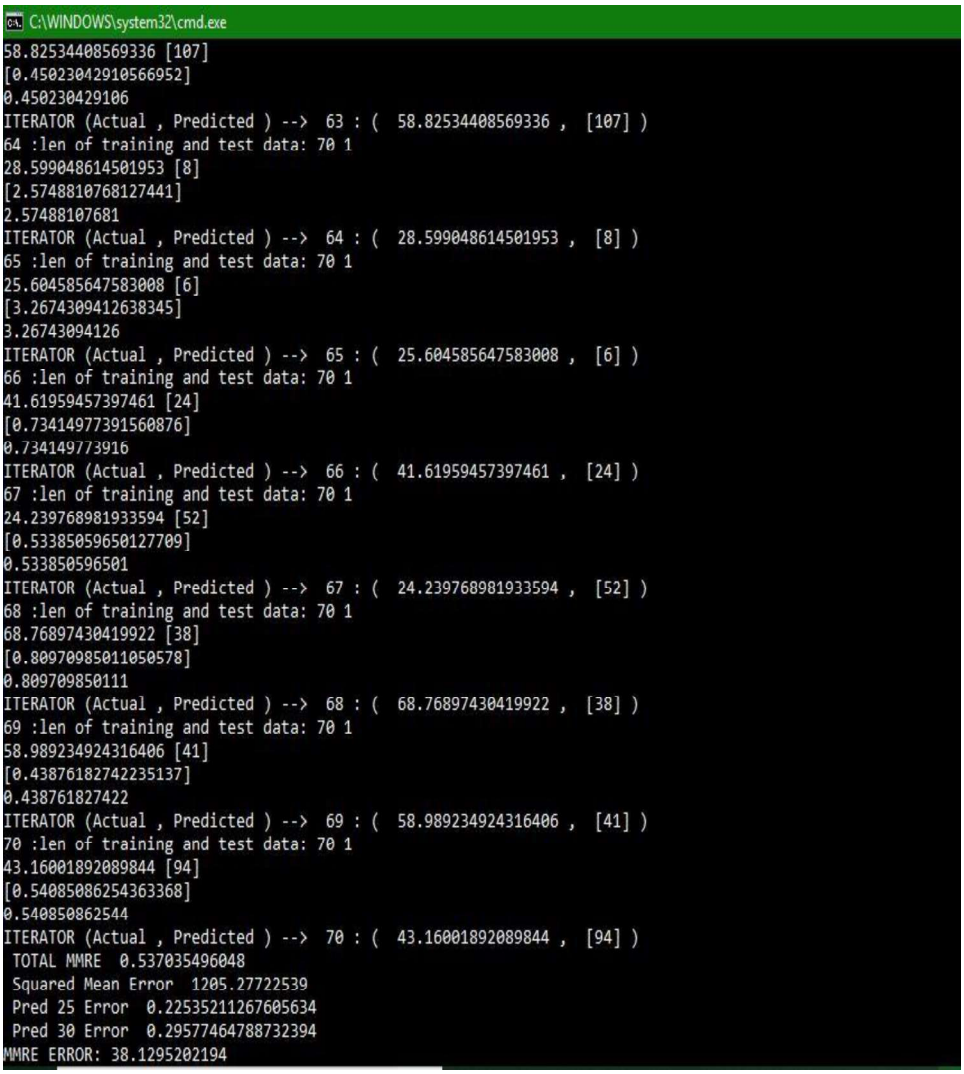
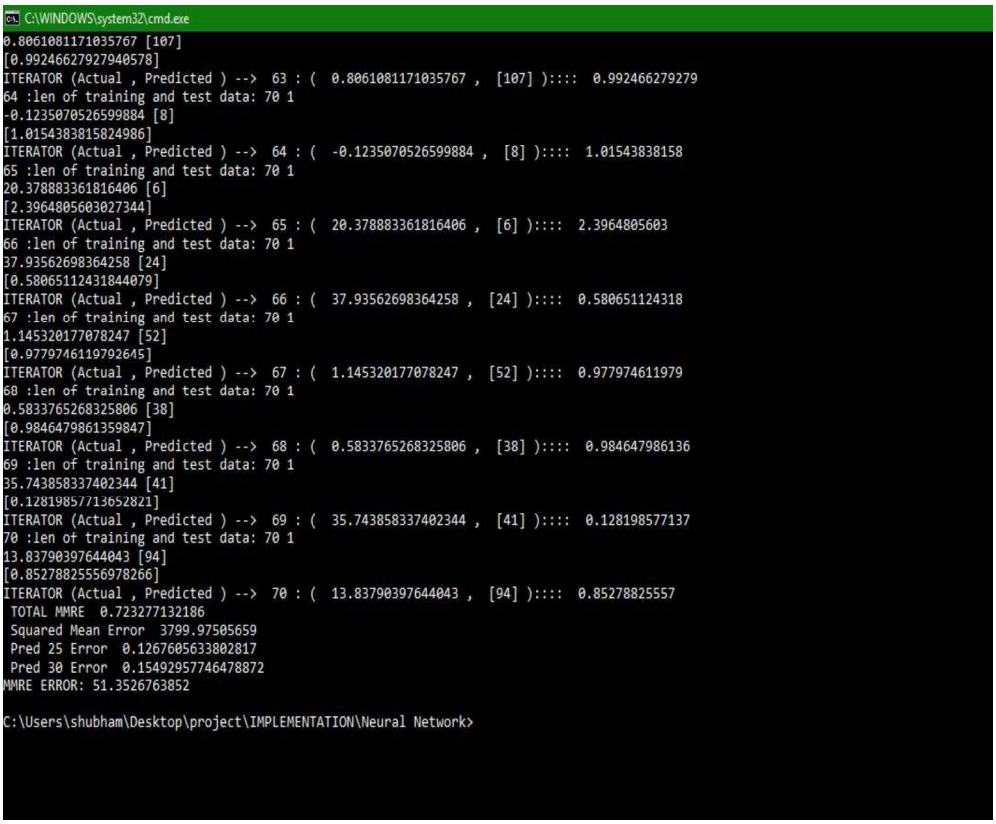


Figure 4.3: Result of GRNN



4.7.4 Neural Network (Ward Neural Network)

Figure 4.4: Training and Testing of Ward Neural Network

This result shows the importance of neural network for prediction. The result obtained using ward neural network is better than all the previous approaches. Total MMRE error rate and squared mean error rate is comparatively low in all aspects hence neural network provides better solution to the problem related to maintenance prediction, Also the prediction 25 and prediction 30 error providing satisfactory result than compare to previous machine learning techniques. But in order to get more

satisfactory result we have combined neural network with evolutionary programming.

the threshold and weight during initial or learning phase, and neural network is used for estimation the predicted value or desired result, Here we used ward neural network as a neural network having 3 slabs with different activation function in the hidden layer.

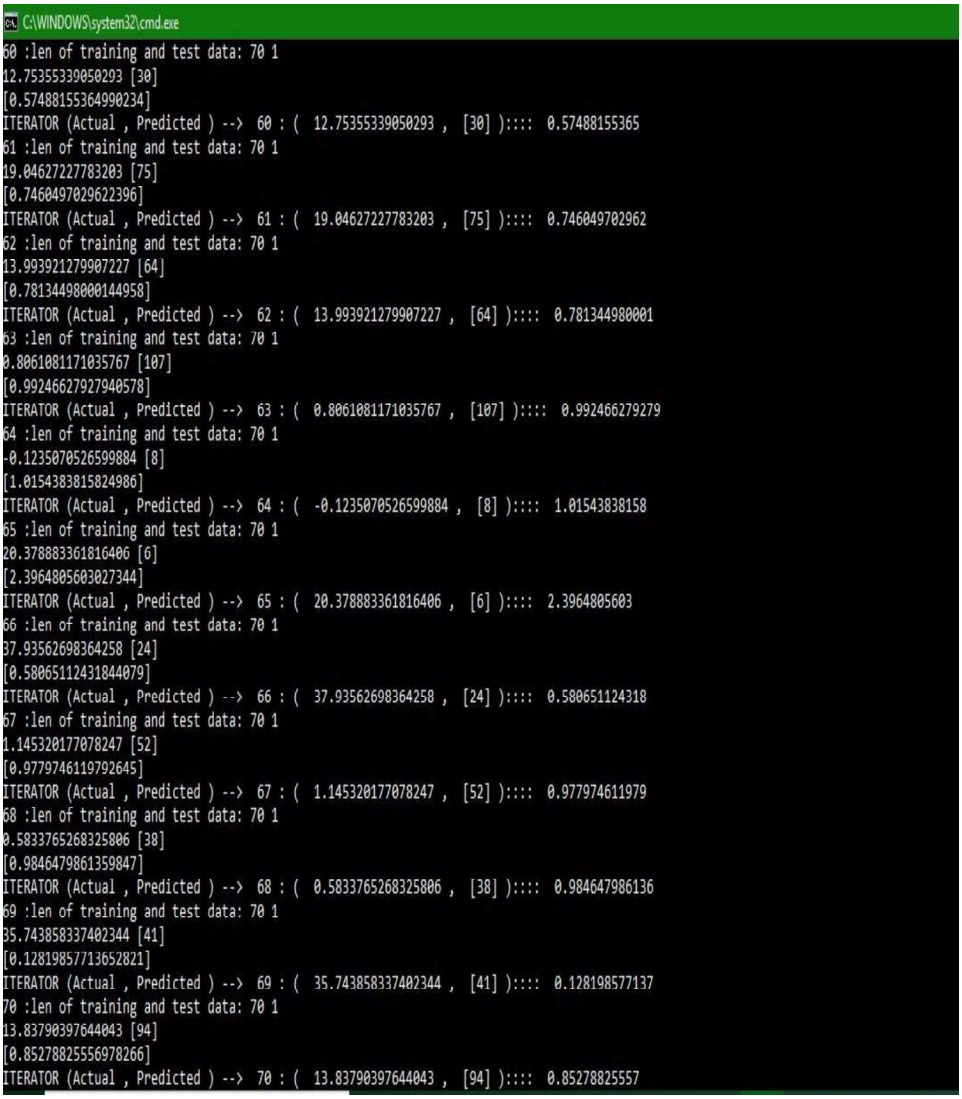
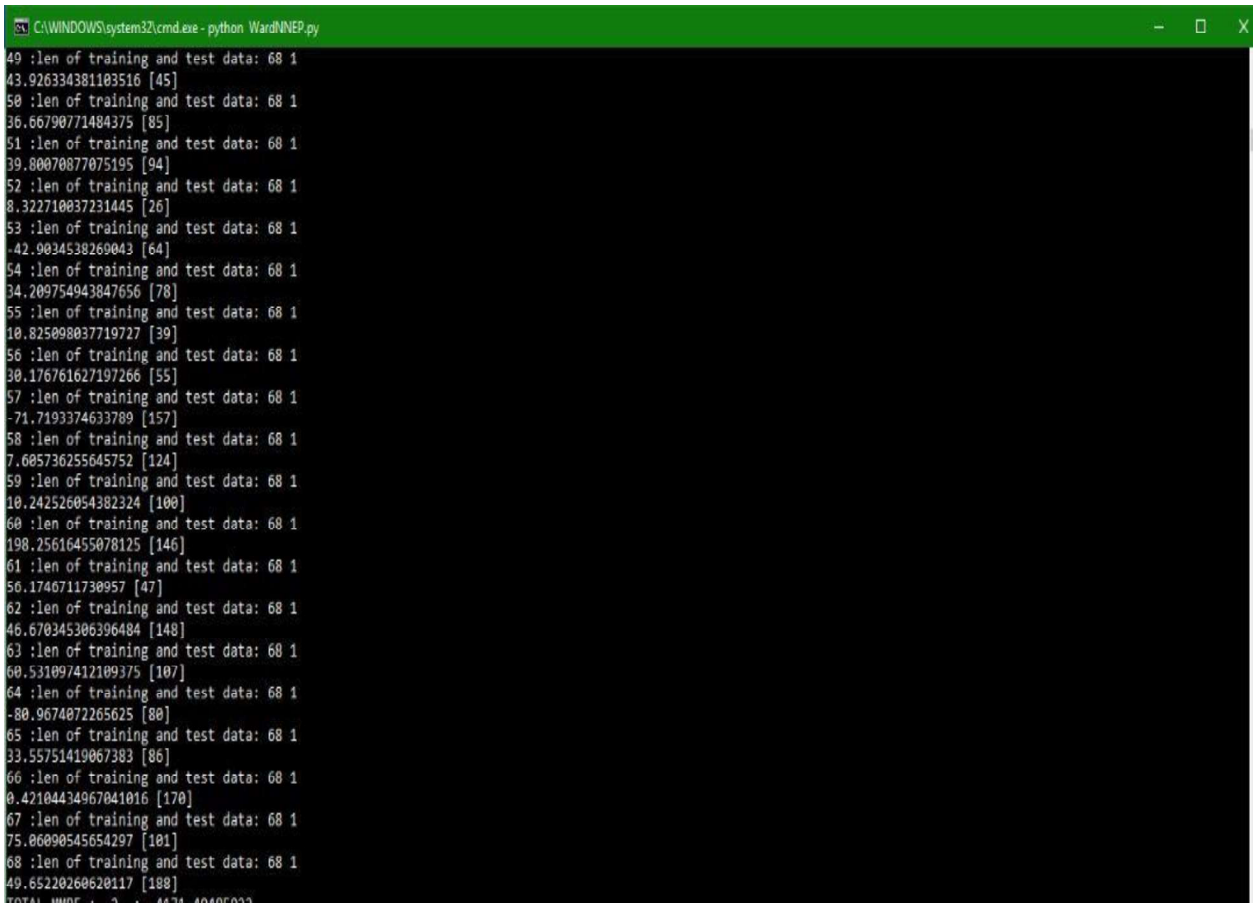


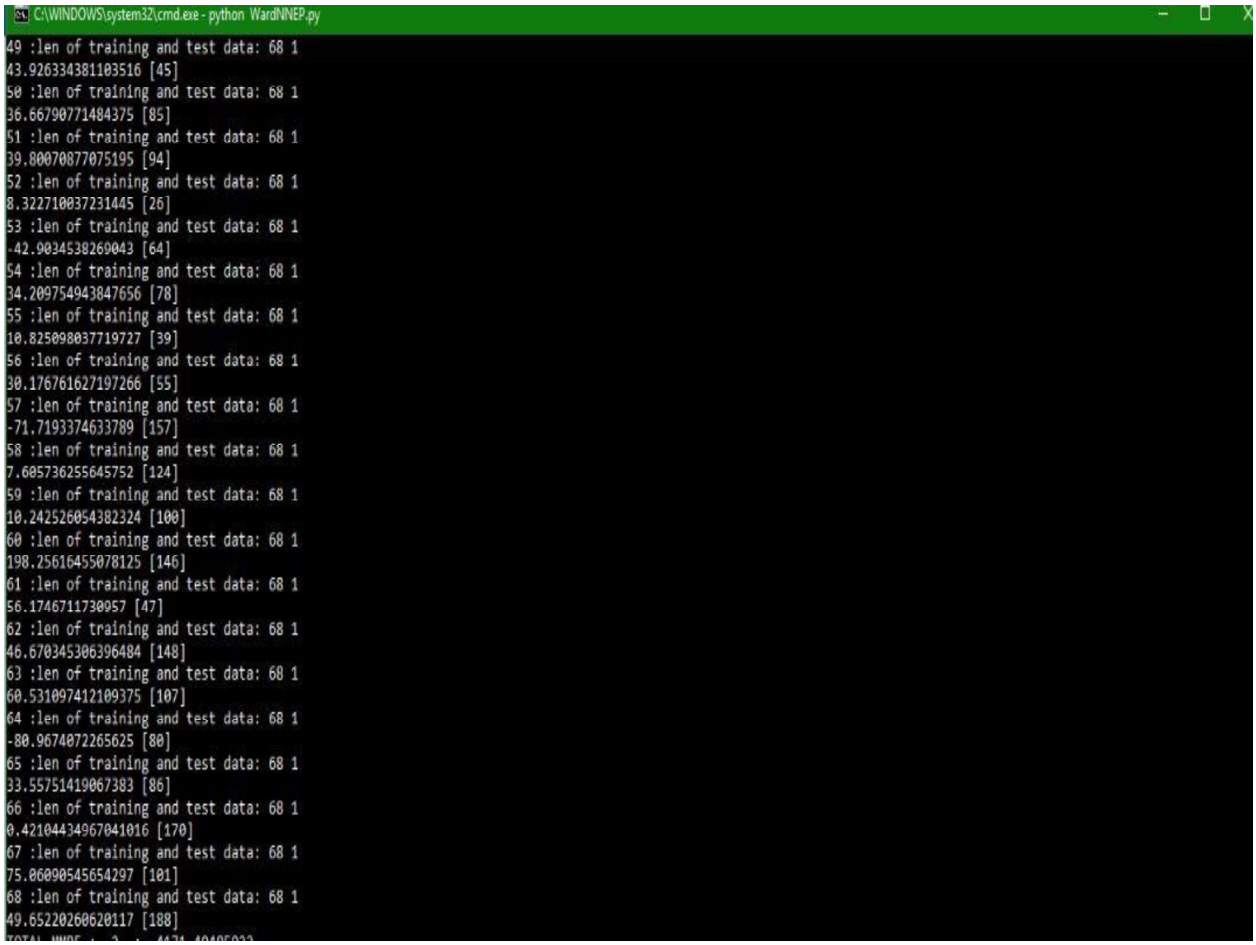
Figure 4.5: Result of Ward Neural Network

4.7.5 GA-NN Approach

Training of GA-NN



Testing of GA-NN



CHAPTER 5
RESULTS AND DISCUSSION

We have implemented two famous datasets UIMS and QUES for maintenance prediction using lots of machine learning techniques, Genetic algorithm and also hybrid approach of genetic algorithm which is combind with neural network to make it a hybrid approach. All the result obtained from hybrid approach along with machine learning techniques has already been discussed in the previous chapters. Here In this chapter we are discussing and comparing the results obtained from all the techniques using tables and graph.

5.1 MACHINE LEARNING TECHNIQUES

Here we have showed the graph for MMRE comparison between all the ML techniques, than comparison between Pred(25) and Pred(30) and lastly compared with Squared error among all the ML techniques.

- All the obtained values of machine learning techniques including neural network and genetic algorithm using prediction accuracy measures .

Table 5.1: Result of all Machine Learning Techniques, Genetic Algorithm, Hybrid Approach

MODELS	MMRE	SQUARED ERROR	PRED 25	PRED 30
Linear Regression	0.387576	770.5298	0.45070422	0 .4929577
Linear Regression(With FS)	0.5613561	1355.39941	0.38028169	0.4788732
Linear Regression(With PCA)	0.4745902	897.301634	0.33802816	0.4788792
Decision Tree	0.35702764	934.830985	0.59154929	0.4084507
Decision Tree(With FS)	0.39029907	1244.9295	0.47887323	0.5211267
Decision Tree(With PCA)	0.45385655	1500.14788	0.47887323	0.5070422

Ward Neural Network	0.65268218	1505.92718	0.30985915	0.4084507
WardNeural Network(With FS)	0.59916230	1151.08552	0.39436619	0.4929577
WardNeural Network(With PCA)	0.48689520	1145.89700	0.35211267	0.4647887
GRNN	0.41522456	823.457714	0.39436619	0.4647887
GRNN(With FS)	0.37092986	526.238768	0.52112676	0.6056338
GRNN(With PCA)	0.41522456	823.457714	0.39436619	0.4647887
NNEP Ward	0.51557413	401.547666	0.46065573	0.6245901
NNEP Ward(With FS)	0.47654471	605.988765	0.67965281	0.7233451
NNEP Ward(With PCA)	0.28578047	403.760901	0.57704918	0.6426229

5.2 GRAPH FOR MMRE ERROR

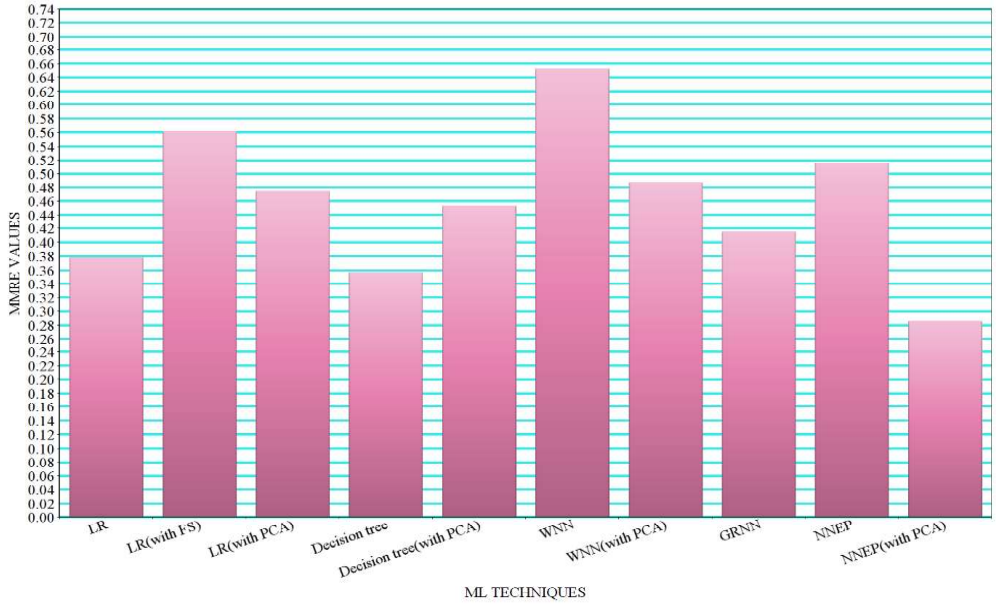


Figure 5.1: MMRE Error for ML Technique

In the above graph It can be easily seen that the MMRE obtained from Neural network with evolutionary programing using principle component analysis achieved good result of having only 0.27 MMRE error and hence it depicts that using neural network with evolutionary programming is a great way for prediction.

5.3 SQUARED ERROR GRAPH

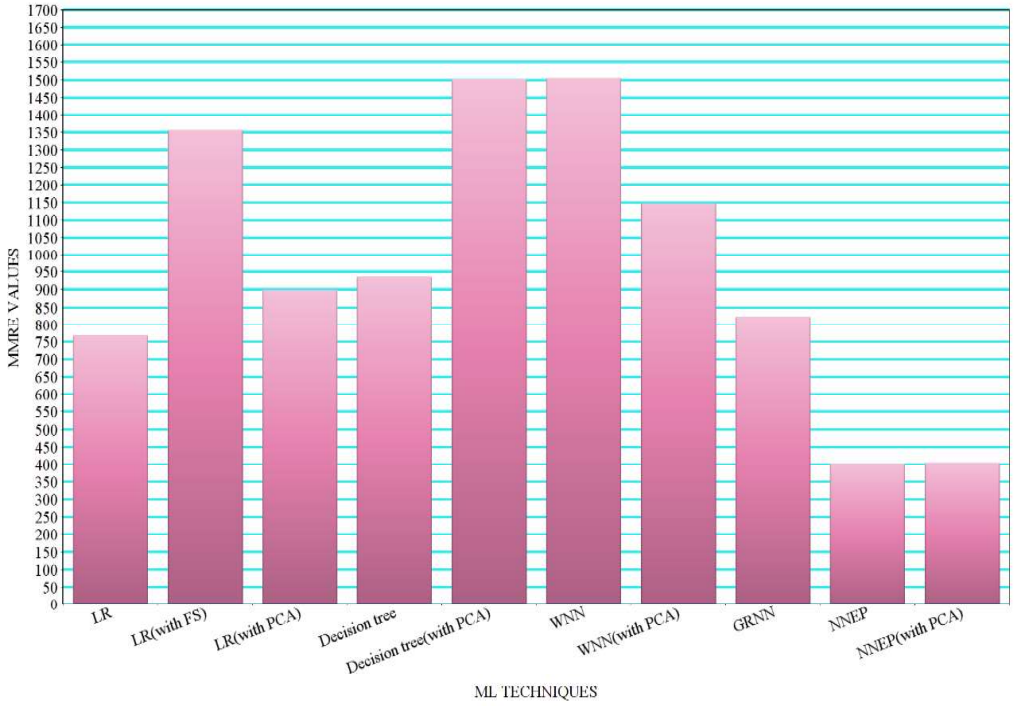


Figure 5.2: Squared Error for ML Technique

5.4 PRED(25) AND PRED(30) VALUES

5.4.1 Pred(25) values

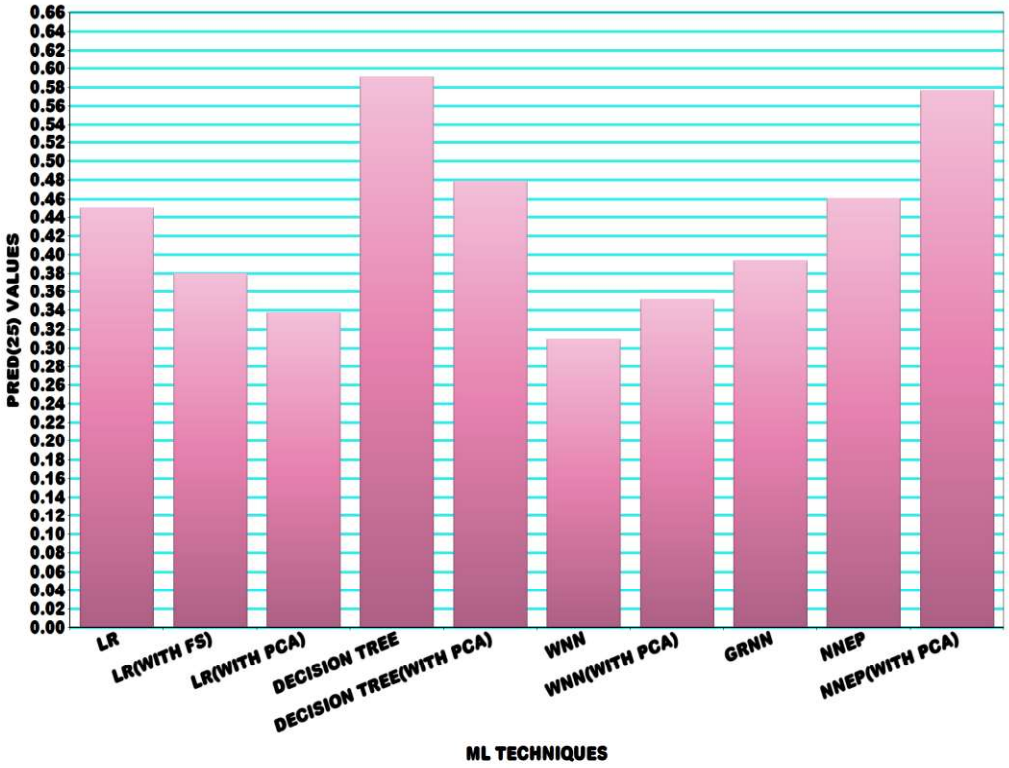


Figure 5.3: Pred(25) values for ML Technique

5.4.2 Pred(30) values

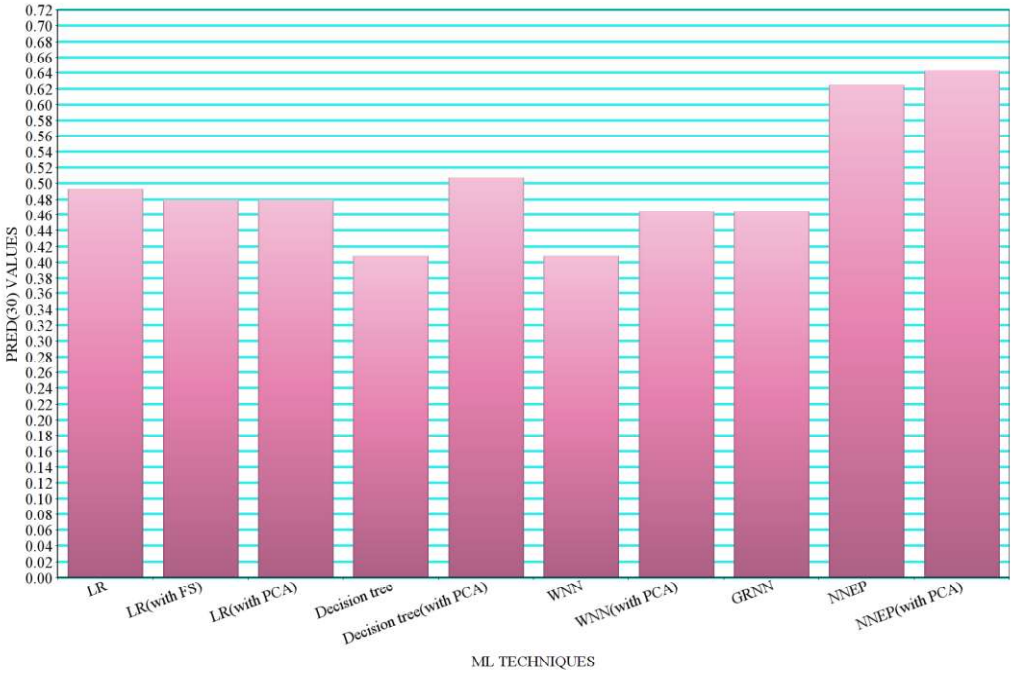


Figure 5.4: Pred(30) values for ML Technique

5.5 THREATS TO VALIDITY

This section plays a critical role for so that it could help of possible bias source to the output or result obtained to the study. It is mandatory to examine the threats to validity of the result that we obtained.

1. Here in this project we tried our machine learning techniques using UIMS and QUES datatsets which are designed in ADA language. It likely should be valid in other object

oriented language as well i.e C++ and java. Further research required for other development of language other than ADA and should likely for other object oriented language.

2. We analyse there are only ten or eleven metrics present which are beneficial for designing a model for maintanibility prediction and hence other metrics should also be considered for prediction of object oriented software for maintainability prediction.

3. Here we analyse the metrics and datasets by using various classifier including neural network and genetic algorithm . Further some new feature selection techniques other than PCA can we used to predict better result like RST, Statistical test etc which could improve Hybrid approach to a new way.

CHAPTER 6
CONCLUSION AND FUTURE WORK

After considering the result with different metrics and datasets and using different ML technique including genetic algorithm and hybrid approach and analyzing the graph and table of comparison, It can be concluded that hybrid approach of genetic algorithm and neural network provided best result than all the old models and old ML techniques.

Hence combining two or more than two techniques is a better way for object oriented maintanabilty prediction. It is also noted that different tecnhiques provide output is not consistent for all the prediction accuracy measures i.e different outcomes obtained from altered techniques. None of the techniques is consistent throughout the datasets and as per the result obtained from prediction accuracy measures Like for some extent MMRE result is best for NNEP technique but not satisfactory result obtained from same techniques using PRED(25) or PRED(30). Hence a consistency among the technique is required so that it can be used in a generalized way.

The result that we obtained using hybrid approach is satisfactory and exploratory. However there might me various others possibilities and factors are can limit the results. Hence further study could improve the economically feasibility and other factors as well.

REFERENCES

[1] W. Li and S. Henry, "Object-Oriented Metrics that Predict Maintainability," Journal of Systems and Software, vol. 23, no2, pp. 111-122, 1993.

[2] Y. Zhou and B. Xu, Predicting the maintainability of open source using design metrics, Wuhan University Journal of Natural Sciences, vol.13, no.1, pp.14- 20, 2008.

[3] V. R. Basili, L. C. Briand and W. L. Melo, A validation of object-oriented design metrics as quality indicators, IEEE Trans. Software Engineering, vol.22, no.10, pp.751-761, 1996.

[4] S. R. Chidamber and C. F. Kemerer, Towards a metrics suite for object oriented design, Proc. Of the 6th ACM Conference on Object-Oriented Programming, Systems Languages, and Applications, Phoenix, AZ, 1991.

[5] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, The Annals of Mathematical Statistics, vol.11, no.1, pp.86-92, 1940.

[6] I. Samoladas, I. Stamelos, L. Angelis and A. Oikonomou, Open source software development should strive for even greater code maintainability, Communications of the ACM, vol.47, no.10, pp.83-87,2014.

[7] M. O. Elish and K. O. Elish, Application of tree net in predicting object-oriented software maintainability: A comparative study, Proc. of European Conference on Software Maintenance and Reengineering, Kaiserslautern, Germany, pp.69-78, 2009.

[8] A. Kaur, K. Kaur and R. Malhotra, Soft computing approaches for prediction of software maintenance effort, International Journal of Computer Applications, vol.1, no.16, pp.339-515, 2010.

[9] M. M. T. Thwin and T. S. Quah, Application of neural networks for software quality prediction using object-oriented metrics, Journal of Systems and Software, vol.76, no.2, pp.147-156, 2005.

[10] R. K. Bandi and V. K. Vaishnavi, Turk DE: Predicting maintenance performance using object oriented design complexity metrics, IEEE Trans. Software Engineering, vol.29, no.1, pp.77-87, 2003.

[11] R. Malhotra and A. Chug, Software maintainability prediction using machine learning algorithm, Software Engineering: An International Journal, vol.2, no.2, pp.9-36, 2012.

[12] A. Kaur, K. Kaur, "Statistical comparison of modelling methods for software maintainability prediction", Int. J. of Software Eng. and Knowledge Eng., vol. 23, no. 06, pp. 743- 774, 2013.

[13] M. Reformat and V. Wu, "Analysis of software maintenance data using multi-technique
approach," Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence, 2003, pp. 53-59,2013

[14] Elish M, Al-Khiaty M (2013) A suite of metrics for quantifying historical changes to predict future change-prone classes in object-oriented software. J Softw Evol Process 25(5):407–437

[15] H. Sharma and A. Chug, "Dynamic metrics are superior than static metrics in maintainability prediction: An empirical case study," 2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), Noida, 2015, pp. 1-6.

[16] Lov Kumar, Debendra Kumar Naik, Santanu Kumar Rath,Validating the Effectiveness of Object-Oriented Metrics for Predicting Maintainability, Procedia Computer Science, Volume 57, Pages 798-806, 2015.

[17] Shafiabady A, Mahrin MN, Samadi M (2016) An empirical investigation of evolutionary algorithm for software maintainability prediction. In: 2016 IEEE students' conference on electrical, electronics and computer science (SCEECS), pp 1–6.

[18] Kumar, L., Kumar, M. & Rath, S.K. Int J Syst Assur Eng Manag (2017) 8: 205.

[19]Lov Kumar, Santanu Ku. Rath, "Hybrid functional link artificial neural network approach for predicting maintainability of object-oriented software", Journal of Systems and Software, January 2016.

[20] Chug, Anuradha & Malhotra, R. (2016). Benchmarking framework for maintainability prediction of open source software using object oriented metrics. 12. 615-634.

[21] R. Jain, D. Sharma and S. K. Khatri, "Hybrid artificial intelligence model based on neural network simulation models for software maintainability prediction," 2017

International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS), Dubai, 2017, pp. 705-708.

[22] W. Li and S. Henry, "Object-Oriented Metrics that Predict Maintainability," *Journal of Systems and Software*, vol. 23, no 2, pp. 111-122, 1993

[23] S. Chidamber, R. Shyam and C. Kamerer, “Towards a metrics Suite for Object-Oriented Design Proceedings”, *Proceeding of Conference on object – oriented programming systems, languages and applications, OOPSLA'91*, pp.197-211, November, 1991.