Genetic Evolutionary Learning Fitness Function (GELFF) for Feature Diagnosis to Software Fault Prediction

P. Patchaiammal, R. Thirumalaiselvi

Abstract: ***Nowadays, proper feature selection f+orFault prediction is very perplexing task. Improper feature selection may lead to bad result. To avoid this, there is a need to find the aridity of software fault. This is achieved by finding the fitness of the evolutionaryAlgorithmic function. In this paper, we finalize the Genetic evolutionarynature of our Feature set with the help of Fitness Function. Feature Selection is the objective of the prediction model tocreate the underlying process of generalized data. The wide range of data like fault dataset, need the better objective function is obtained by feature selection, ranking, elimination and construction. In this paper, we focus on finding the fitness of the machine learning function which is used in the diagnostics of fault in the software for the better classification.***

Keywords: Fitness Function, Parents, Chromosomes, Crossover, Mutations, Fault Diagnostics, Genetic evolutionary Programming (GEP), Machine Learning (ML), Feature Selection, Feature extraction,Genetic evolutionary Learning Fitness Function (GELFF).

I. INTRODUCTION

For using machine learning algorithm, we might deduct the feature of historical data. By using previous experience the type of features are formed as feature set to represent the samples. From the set, one has to select the features which affect the result of training set. This process is known as feature selection. The random data usage to train Machine Learning may give flaw in some cases. These features reduce the accuracy of classification and also increase the fault. Those features are to be removed. This is known as feature reduction. Genetic evolutionary programming consider the dataset as population. The population has number of chromosomes with gene sequence is considered as population in Genetic evolutionary programming. Fitness function is used to find best chromosomes as a parent for new population creation. The solution is obtained by using crossover and mutation operation.

Genetic Evolutionary Programming is used to solve the problem by studying the data from its origin which includes all sciences. Using this we make predictions and update our ideas by observing the data. So if we have historical dataset then, by using the data we solve the problem with a code which train a machine learning model on that data to predict the result. Even though, there are lots of model available and the data are increasing every year so there is a need for new machine learning model to predict the faults which can reduce rework. Machine learning democratizes the scientific discovery of prediction model to develop Green Engineering in software development. Python is a general purpose readable and compact application to build all type of classifications. This paper helps how one can check the fitness of the feature set by using the available dataset. In our previous paper, we formed Hypothesis Space by using Chi-square Hypothesis Testing of Eight NASA PROMISE Data Repository. In this paper, we find the fitness function, which is relevant to the diagnostics of Fault in software Development.

Genetic evolutionary programming (GEP), which is a subset of evolutionary Machine learning programing.GEP used to solve problem by using biological evolution. Genetic evolutionary programming is the best way to find out a functional relationship of data in order to categorize the feature. GEP is applied to the software development for the diagnostics and deduction of fault. Machine learning is used to select best features based on the experience gained from the dataset.

This paper is used to form a fitness function for the feature set by using genetic evolutionary programming. The best of fitness function is evaluated with the help of genetic evolutionary programming. We use supervised machine learning algorithm known as Random Forest for classification for finding the fitness of the training data.

Revised Manuscript Received on September 22, 2019.

P. Patchaiammal, Research Scholar, Bharath University, Chennai, E-mail: sarandsk1@gmail.com.

R. Thirumalaiselvi, Research Supervisor, Bharath University, Chennai, Assistant Professor, Computer Science Department, Govt. Arts College (Men), Nandanam, Chennai.

II. MACHINE LEARNING

Machine learning is a complete tool for prediction analysis. ML provide a perfect base for tackling numerous big data challenges. ML provide traditional linear algebraic performance, with that of latest hardware and software performance. This paper is used to enabling the potential usage of ML to the widest range of data set. ML techniques are less sensitive to problem area and its characteristics. Therefore if we use the availability of the objective function as a mathematics experience then the design of feature type is easy.

Fortunately, various ML techniques were available which can handle efficiently a large number of parameters to form population. By using evolutionary programming like genetic evolutionary with machine learning function, one can easily find the fitness of the feature set. This fitness will also help to form a sustainable diagnostics model for prediction. We use this combined genetic evolutionary programming with random forest to form deep fitness function which reduces the variance in fault classification. Genetic evolutionary programming may contain a population of different size. Genetic evolutionary programming usually start by random value of population which may be fit or unfit for a problem. To form new generation, Genetic evolutionary programming use operations like selection, crossover and mutation for measuring the fitness of the problem.

This paper gives the extreme view of machine learning solution for finding the fitness of the feature set in the available population. Genetic evolutionary programing is applied to check the fitness of the machine learning function in the population. Various generations are formed with parents, crossover and mutation.

There are three major types of machine learning algorithms are available.

A. Supervised Learning

These are mostly used machine learning algorithms. In this type the training set is formed by known label and reset. A model is prepared through a training process, which make predictions. The training continues until the model continues the level of accuracy on the dataset. This algorithm makes the learning method as explicit one. This makes the prediction based on the data given in the training set. This algorithm ends only if it achieve the acceptable degree of performance level. Fault prediction problem needs a classification of fault so classification algorithm known as random forest is used to form the fitness function of feature set. The best of fitness is found by using genetic evolutionary programming. Finally the level of performance is verified by linear classifier.

B. Unsupervised Learning

The training dataset has no known label and result. The model is prepared by reducing the present input dataset by reducing the present input data structure through mathematical process.

C. Semi - Supervised Learning

Semi - supervised learning is a mixture of labeled and unlabeled example. It has desired prediction problem and no known problem of learning structure to organize the data for prediction.This algorithm is used to define the boundaries of the unlabeled data.

III. GENETIC EVOLUTIONARY PROGRAMMING (GEP)

GEP is best for finding optimum solution for prediction problem. GEP is designed like simulation program and the design parameters are entered to the simulation program to find systematic different possible solution. GEP helps to find the most appropriate ML algorithm for optimizing the solution in given Computational time. GP uses the selection of population size which is very much needed to solve the problem. The Population size significantly control the parameter selection for GEP performance.GEP finds a solution for random initialized parameter with upper and lower limit by forming iteration.The solution consist of the variable assigned within the limit. Each solution is considered as chromosomes in genetic evolutionary programming. The new solution generation commences after assigning fitness values for each chromosome.

The genetic evolutionary feature classification set is developed accordingly to the developers, users, organization and process in use. This paper finds the fitness of the feature set in order to suggest the developers and users about the fault which occurs over and above organizational work and the strategies of project to reduce fault through the learning techniques. The development organization process used to deliver the software according to the tasks assumed by the users. This learning method is used to reduce the fault so that the security and project performance is ameliorated. This also helps to reduce the cost, time and rework so as to form green engineering in software development.

The success of fault prediction model achieved by proper analysis and classification of fault. Wrong analysis leads to misclassification of software faults. So there is a need to find the origin of fault occurrence.

By using the genetic evolutionary programming to form fault feature set, one can form best fault classification.

In this paper, by analyzing historical categories of fault from past data the feature set of fault prediction model for future is formed. If the collected data is in best quality then one can reach occurring in future fault prediction. Therefore origin analysis of fault is necessary. To achieve origin cause of a fault feature set the selection and extraction technique is used.

In this paper the features of fault dataset are validated by using the machine learning fitness function. In genetic evolutionary programming the existence of fitness of gene is calculated by using the fitness function. The fitness function is applied to all sample to form generation. Each generation the highest fitness value generation is chosen for best fitness of the sample taken from the population. The best individuals are selected as parents to form next new generations. This process goes on iteration until to get the best parents and fitness solution. The process concludes with the linear classifier graph to show the fitness of the function as which is fitted with the feature set selected.

Key terms for genetic evolutionaryfitness function

Population: Random set of sample. It is a point which is used for predicting the task.

Sample: It is a data point set 'X' for prediction.

Target Function: It is a modelling process which is used to approximate the function. In general, the target function for data point 'X' is represented as f (x).

Hypothesis: It is a function which is similar to true function of target model function.

Hypothesis Cases:A good hypothesis is testable and it can beeither true case or false case. In general, a hypothesis must be false. The outcome of test data set should make the hypothesis so that is not true.

Result: It is used for make predictions which creates evidence to best fits.

Hypothesis Set: In ML there exist a pattern y= f(x) between the input 'x' and the output 'y' 'f(x)' is a target function. The function 'f' has to be chosen from the hypothesis space. The function f (x) is an unknown function in which ML try to guess a hypothesis function h(x) to approximate the target function f(x). H(x) is the set of all possible hypothesis. Goal of hypothesis set H(x) is that the ML process has to find the best approximates to the unknown target function. $H= \{h_1, h_2, ..., h_n\}$ where $h \in H$. This function explain the relation between 'x' and 'y' for all possible inputs.

Classifier: It is unsupervised term which is used to learn the model form the training data. It includes the sample of training instances and training example. It is a data point 'x' in available. Training set used for make a determined efforts to deal with a particular modeling task which is to be used for prediction.

Data: Data is gathered to test the hypothesis. The hypothesis used to show the data is irrelevant/relevant. 'H' should be chosen before looking at the data.

Featureset: The unbounded features of an object can be chosen by using pattern learning. The finite features of an object is classified by using idealized function.

IV. GENETIC EVOLUTIONARY LINEAR CLASSIFIER METHOD

It is done by using a linear function of inputs. The inputs are mentioned as solar values (i.e.,) features.For feature selection a typical domainset is formed. This domain contains various categories of fault.A classification task is used to separate the original fault by eliminating the redundant fault and also wrong fault. This classification follows "gene" expression to separate the original fault from wrong to avoid misclassification. This domain set form the training input set of learning process. Some initial Filterization method used to bring the number of feature. The gene expression is used to standardize the feature magnitude.

For machine learning model selection Feature selection plays a vital role. Feature selection avoid over fitting by reducing the bias in it and form the proper training set with different dataset. Feature selection is used before the model is trained to get accurate estimation method. In feature selection, the first step is data preparation, the second one is model selection and the final step is apply training on selected features. The decision formed to select the feature on the training input is passed to the model to enhance the better result.

A. Linear Learning Function

All ML Algorithms are used a target mapping function ($\underline{f}$) which maps the training input dataset ($\underline{x_1, x_2, ..}$ x_n) by making function prediction to reach a task ($\underline{y}$).In some case there might be fault which is not having enough attributes to sufficiently havalivize the best mapping form X to Y. In order to perform predictive modelling most accurate prediction is necessary. This is approved by common type of ML mapping:

(1)

This function is used to estimate the target function (f) to predict the output variable (y) for given input variables (x).

The Linear learning function for finding fitness function using GEP is:

(2)

where 'Y' is the dependent variable. 'X' is an independent variable. 'is the intercept constant. 'is the coefficient of the relationship between 'X' and 'Y'.

V. GENETIC EVOLUTIONARY PREDICTION

EvolutionaryGenetic evolutionary algorithm is used to find optimize prediction. These algorithms are dynamic in nature. The various steps involved in evolutionary algorithms are population generation, which forms new solution from current values. The second one is fitness function which finds better solution from several solutions. The final one is variation generation, which is used to find new suitable solution in chosen population according to the calculated fitness function.

Genetic evolutionary programming is a random classification evolutionary algorithm. Genetic evolutionary programming has a set of parameters to define a proposed solution of a problem known as chromosome. A chromosome has features to define individual in population. The number of solutions in genetic evolutionary programming defines a size of a population. Each individual has solution. It means each individual has fitness value. In order to find best in set of individuals fitness function is used. The selection of best individual has done by generation forming mating. Always there is a need for high quality in mating offspring selection, which avoid bad individuals and keep good individual. This is known as selecting high quality individual. This formation is stopped by finding optimal individual.In some situation there is recombination in the information of the parents to generate new individual creation to replace the old population. This has done by crossover in genetic evolutionary programming.

The variation diversity is achieved by mutation in which one or more gene values of chromosome from initial state are altered and the solution changed entirely from the previous one. The general structure of genetic evolutionary learning is described in diagram 1. This diagram shows the steps for genetic learning from gene representation till evaluation. The evaluation of genetic algorithm fitness function is done by using Linear Discriminant analysis that is linear classifier. This learning classifier is used to find the best fitness values in order to find the fault in the software project to reduce rework.

A. Chromosome representation

Chromosome buildswith gene in genetic evolutionary programming. The feature collection helps to form genes. The best set of features helps to design accurate prediction machine learning model. The total number of labels available in a dataset forms feature elements which is considered as genes. The gene total defines the chromosome length. The first step in genetic evolutionary programming is gene selection and representation. The various ways of gene representations are done by binary, decimal, string, etc. The main goal of gene selection is selecting necessary features as gene element. Generally, binary representation of gene is used. If the gene value is 1 the gene is selected, otherwise the gene is not selected, means the gene value is 0. Feature selection for prediction has one-to-one mapping in between gene and feature element.

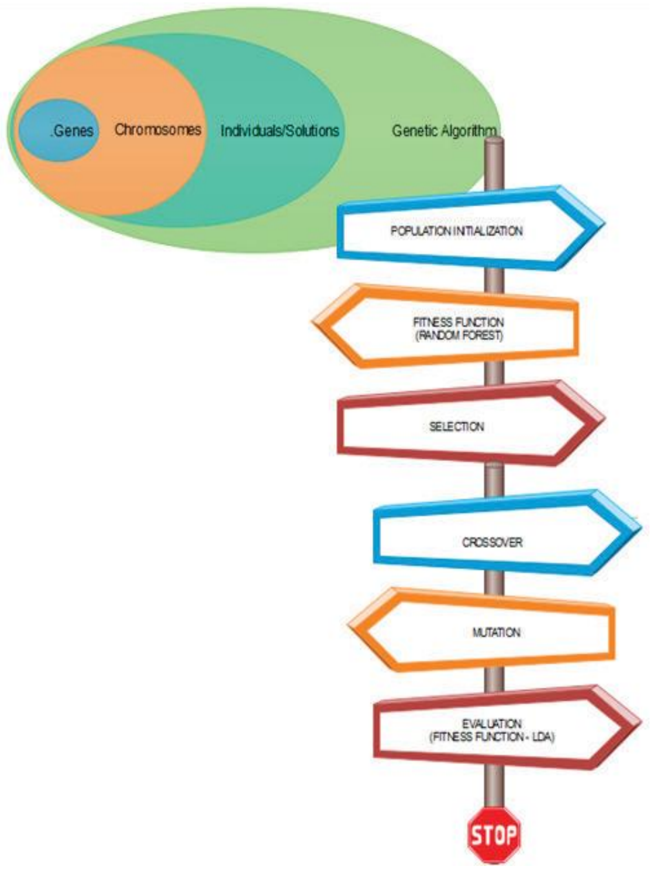

Genes
Chromosomes
Individuals/Solutions
Genetic Algorithm
POPULATION INITIALIZATION
FITNESS FUNCTION (RANDOM FOREST)
SELECTION
CROSSOVER
MUTATION
EVALUATION (FITNESS FUNCTION - LDA)
STOP

Diagram 1: Basic structure of Genetic Evolutionary Learning

B. Fitness Function

Fitness function is also known as evaluation function which is used to identify the closeness of a solution to optimize the machine learning problem. Fitness function is also used to regulate how fit a solution to a problem. By using genetic evolutionary algorithm, each solution is represented as a chromosome which is a general representation of string of binary numbers.

Chromosomes are formed as solution set and the best solution is obtained from this set.

C. Formation of fitness function

Fitness set is formed by Fitness value corresponding to each chromosomes. From the set, the Fitness value which is highest is chosen as Fitness function. That Fitness function is used to select best parents in the current population set. This formation of Fitness function used for finding accurate prediction classifier of a given dataset. This classification is done with the help of supervised machine learning classifier algorithm.

D. Frame work for Genetic evolutionary Learning Fitness Function(GELFF)

In this paper, we use linear classifier as the Evolutionary Genetic evolutionary learning function to find the best fitness result. The linear classifier used here is the mathematical function. This function is one of the best machine learning function for the problems, which uses natural selection. The natural selection method is used in the genetic evolutionary algorithm problems to find only the fittest individual result over various generations.

This framework is used to select the subset of the feature attributes related to the optimization problems. In the previous paper we selected totally 29 feature attributes for fault prediction classification taxonomy creation. This paper used to check whether those features are really necessary for classification or not. This evolutionary method is applied to improve the best result of the features selected. This is also used to select some unnecessary features for elimination.

The idea used in this paper is we extract the best genes that is attributes from each individuals. The solution set is called individuals or chromosomes of population. This set of individuals are the possible solutions of the problem we considered.

VI. IMPLEMENTATION OF GELFF

A. Methodology

In the dataset, some duplicates and irrelevant values are removed and modified dataset formed which is passed to hybrid machine learners in Python classifiers. Various preprocessing methods are used in order to increase the quality of the data sets to improve the performance of the predictor.

B. Data Collection

The largest dataset is collected from Bugzilla to form initial population. Then random page of the initial population is formed. By evaluating the members in the population the fitness function of the solving problem is generated and if it is lesser than the population is accepted with more fitted chromosomes.

Otherwise, again new population generated using genetic evolutionaryoperator’s reproduction, crossover and mutation. This new population again undergoes new negativity test of fitness study. This process continues under to reach proper feature set.

C. Experiment Results

Collected features from dataset are loaded into the python program to find the best fitness result of the function linear classifier. Python makes the best fitness result as the final result. Python classifier used to makes the feature selection as the best. Linear classifier is one of machine learning technique which is used in this paper as the genetic learner method.

The programmatic observed results are shown in the following diagrams. The total features are used in the python program names are

[‘LOC_BLANK’, ‘BRANCH_COUNT’, ‘CALL_PAIRS’, ‘LOC_CODE_AND_COMMENT’, ‘LOC_COMMENTS’, ‘CONDITION_COUNT’, ‘CYCLOMATIC_COMPLEXITY’, ‘CYCLOMATIC_DENSITY’, ‘DECISION_COUNT’, ‘DECISION_DENSITY’, ‘DESIGN_COMPLEXITY’, ‘DESIGN_DENSITY’, ‘EDGE_COUNT’, ‘ESSENTIAL_COMPLEXITY’, ‘ESSENTIAL_DENSITY’, ‘LOC_EXECUTABLE’, ‘PARAMETER_COUNT’, ‘HALSTEAD_CONTENT’, ‘HALSTEAD_DIFFICULTY’, ‘HALSTEAD_EFFORT’, ‘HALSTEAD_ERROR_EST’, ‘HALSTEAD_LENGTH’, ‘HALSTEAD_LEVEL’, ‘HALSTEAD_PROG_TIME’, ‘HALSTEAD_VOLUME’, ‘MAINTENANCE_SEVERITY’, ‘MODIFIED_CONDITION_COUNT’, ‘MULTIPLE_CONDITION_COUNT’, ‘NODE_COUNT’, ‘NORMALIZED_CYLOMATIC_COMPLEXITY’, ‘NUM_OPERANDS’, ‘NUM_OPERATORS’, ‘NUM_UNIQUE_OPERANDS’,NUM_UNIQUE_OPERATORS’,’NUMBER_OF_LINES’,’PERCENT_COMMENTS’ ,’LOC_TOTAL’,’Defective’].

The features are plotted against the populations used in Diagram 2. This diagram has two axis with values as features and population collected.

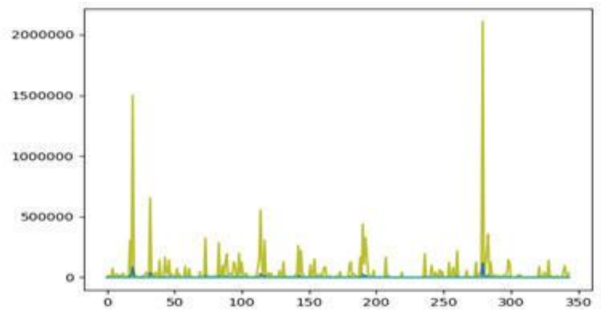


Diagram 2 A performance plot of features in population

A performance plot for showing the linear classifier as the best of fault diagnostic classifier is shown in Diagram 3. This diagram also form the straight line for the features used. This shows that linear classifier is the best for the performance measure of fault prediction classification.

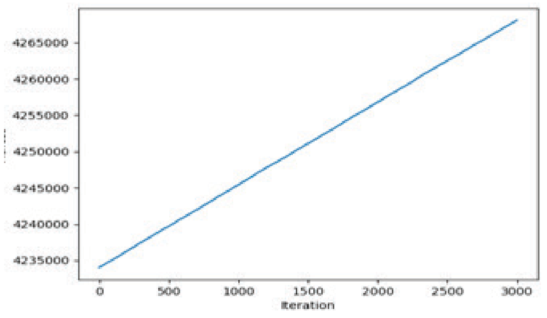


Diagram 3 A plot to show linear classifier is the best fitness function

In this section, the results of the dataset are analyzed and only 29 features selected from our previous paper is considered for fitness result. In the first experiment, parents are selected against the 29 fault features in the means of 3500 generations. Applying more generation will conclude the best result. The difference between the beginning of the generation and end of the generations are separately shown in the charts. It is described in the Table 1 and the corresponding performance chart of 1-25 generations are shown in Diagram 4 and 3476-3500 generations are shown in Diagram 5.

S. No	Generation	
1	0	[[9.554e-03 5.268e-04 4.707e-01 4.704e-01 1.022e+00 1.888e+00 1.528e+00
2	1	[[1.298e+00 1.177e+00 1.549e+00 2.449e-01 1.621e+00 1.989e+00
3	2	[[1.330e+00 4.591e-01 8.817e-01 5.491e-01 1.442e+00 5.813e-01
4	3	[[1.330e+00 4.591e-01 8.817e-01 5.491e-01 1.442e+00 5.813e-01
5	4	[[1.330e+00 4.591e-01 8.817e-01 5.491e-01 1.442e+00 5.813e-01
6	5	[[1.33 0.459 1.56 0.549 1.442 0.581 0.472 0.233 0.304 1.794
7	6	[[1.33 0.459 1.56 0.549 1.442 0.581 0.472 0.233 0.304 1.794
8	7	[[1.33 0.459 1.56 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
9	8	[[1.33 0.459 1.56 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
10	9	[[1.33 0.459 2.16 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
11	10	[[1.33 0.459 2.16 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
12	11	[[1.33 0.459 2.658 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
13	12	[[1.33 0.459 2.852 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
14	13	[[1.33 0.459 3.631 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
15	14	[[1.33 0.459 4.327 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
16	15	[[1.33 0.459 4.573 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
17	16	[[1.33 0.459 5.46 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
18	17	[[1.33 0.459 5.899 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
19	18	[[1.33 0.459 5.899 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
20	19	[[1.33 0.459 6.312 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
21	20	[[1.33 0.459 6.745 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
22	21	[[1.33 0.459 6.751 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
23	22	[[1.33 0.459 7.437 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
24	23	[[1.33 0.459 8.214 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
25	24	[[1.33 0.459 8.904 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
...	...	...

Table 1 Performance report of parents with respect to generations 1 to 3499

3476	3475	[[1.330e+00 4.591e-01 1.513e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3477	3476	[[1.330e+00 4.591e-01 1.514e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3478	3477	[[1.330e+00 4.591e-01 1.514e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3479	3478	[[1.330e+00 4.591e-01 1.514e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3480	3479	[[1.330e+00 4.591e-01 1.515e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3481	3480	[[1.330e+00 4.591e-01 1.515e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3482	3481	[[1.330e+00 4.591e-01 1.516e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3483	3482	[[1.330e+00 4.591e-01 1.516e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3484	3483	[[1.330e+00 4.591e-01 1.516e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3485	3484	[[1.330e+00 4.591e-01 1.517e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3486	3485	[[1.330e+00 4.591e-01 1.517e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01

3487	3486	[[1.330e+00 4.591e-01 1.518e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3488	3487	[[1.330e+00 4.591e-01 1.518e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3489	3488	[[1.330e+00 4.591e-01 1.518e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3490	3489	[[1.330e+00 4.591e-01 1.519e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3491	3490	[[1.330e+00 4.591e-01 1.519e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3492	3491	[[1.330e+00 4.591e-01 1.520e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3493	3492	[[1.330e+00 4.591e-01 1.520e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3494	3493	[[1.330e+00 4.591e-01 1.520e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3495	3494	[[1.330e+00 4.591e-01 1.521e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3496	3495	[[1.330e+00 4.591e-01 1.522e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3497	3496	[[1.330e+00 4.591e-01 1.522e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3498	3497	[[1.330e+00 4.591e-01 1.523e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3499	3498	[[1.330e+00 4.591e-01 1.523e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3500	3499	[[1.330e+00 4.591e-01 1.523e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01

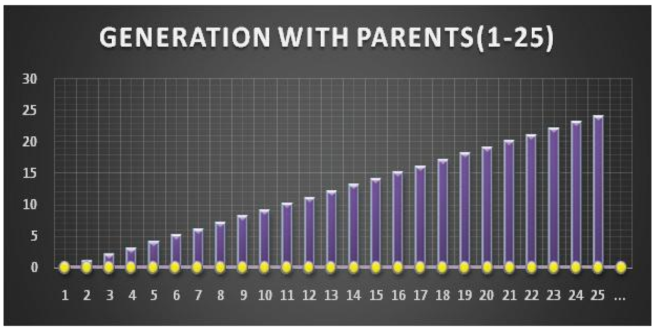


Diagram 4A plot of Parents formation up to generation 0-25

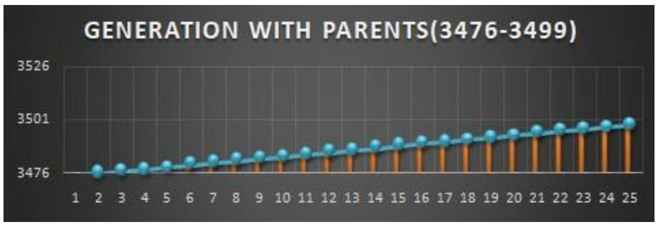


Diagram 5 A plot of Parents formation up to generation 3476-3499

The performance summary of crossover with respect to generations 0-3499 has been demonstrated in the Table 2. This table contains the generation values with respect to crossover values to be used in the genetic evolutionary algorithmic results. The performance strategy of the crossover chart is plotted in the Diagram 6 and Diagram 7.

Table 2: Performance report of crossover with respect to generations 1 to 3499

S. No	Generation	Crossover
1	0	[[9.554e-03 5.268e-04 4.707e-01 4.704e-01 1.022e+00 1.888e+00 1.528e+00
2	1	[[1.298e+00 1.177e+00 1.549e+00 2.449e-01 1.621e+00 1.989e+00
3	2	[[1.330e+00 4.591e-01 8.817e-01 5.491e-01 1.442e+00 5.813e-01
4	3	[[1.330e+00 4.591e-01 8.817e-01 5.491e-01 1.442e+00 5.813e-01
5	4	[[1.330e+00 4.591e-01 8.817e-01 5.491e-01 1.442e+00 5.813e-01
6	5	[[1.33 0.459 1.56 0.549 1.442 0.581 0.472 0.233 0.304 1.794
7	6	[[1.33 0.459 1.56 0.549 1.442 0.581 0.472 0.233 0.304 1.794
8	7	[[1.33 0.459 1.56 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
9	8	[[1.33 0.459 1.56 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
10	9	[[1.33 0.459 2.16 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
11	10	[[1.33 0.459 2.16 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
12	11	[[1.33 0.459 2.658 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
13	12	[[1.33 0.459 2.852 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
14	13	[[1.33 0.459 3.631 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
15	14	[[1.33 0.459 4.327 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
16	15	[[1.33 0.459 4.573 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
17	16	[[1.33 0.459 5.46 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
18	17	[[1.33 0.459 5.899 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
19	18	[[1.33 0.459 5.899 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
20	19	[[1.33 0.459 6.312 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
21	20	[[1.33 0.459 6.745 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
22	21	[[1.33 0.459 6.751 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
23	22	[[1.33 0.459 7.437 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
24	23	[[1.33 0.459 8.214 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735

25	24	[[1.33 0.459 8.904 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
...	...	...
3476	3475	[[1.330e+00 4.591e-01 1.513e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3477	3476	[[1.330e+00 4.591e-01 1.514e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3478	3477	[[1.330e+00 4.591e-01 1.514e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3479	3478	[[1.330e+00 4.591e-01 1.514e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3480	3479	[[1.330e+00 4.591e-01 1.515e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3481	3480	[[1.330e+00 4.591e-01 1.515e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3482	3481	[[1.330e+00 4.591e-01 1.516e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3483	3482	[[1.330e+00 4.591e-01 1.516e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3484	3483	[[1.330e+00 4.591e-01 1.516e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3485	3484	[[1.330e+00 4.591e-01 1.517e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3486	3485	[[1.330e+00 4.591e-01 1.517e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3487	3486	[[1.330e+00 4.591e-01 1.518e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3488	3487	[[1.330e+00 4.591e-01 1.518e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3489	3488	[[1.330e+00 4.591e-01 1.518e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3490	3489	[[1.330e+00 4.591e-01 1.519e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3491	3490	[[1.330e+00 4.591e-01 1.519e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3492	3491	[[1.330e+00 4.591e-01 1.520e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3493	3492	[[1.330e+00 4.591e-01 1.520e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3494	3493	[[1.330e+00 4.591e-01 1.520e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3495	3494	[[1.330e+00 4.591e-01 1.521e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3496	3495	[[1.330e+00 4.591e-01 1.522e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3497	3496	[[1.330e+00 4.591e-01 1.522e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3498	3497	[[1.330e+00 4.591e-01 1.523e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3499	3498	[[1.330e+00 4.591e-01 1.523e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3500	3499	[[1.330e+00 4.591e-01 1.523e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3500	3499	[[1.330e+00 4.591e-01 1.523e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01

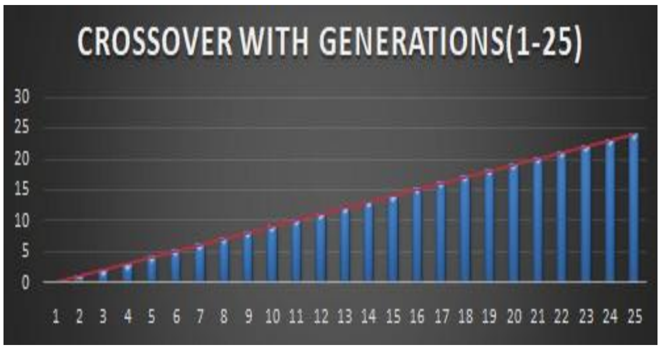


Diagram 6 A plot of Crossover with respect to generation 0 to 25.

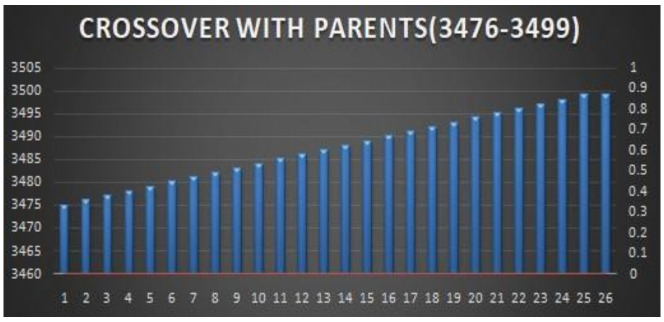


Diagram 7: A plot of Crossover with respect to generation 3476 to 3499

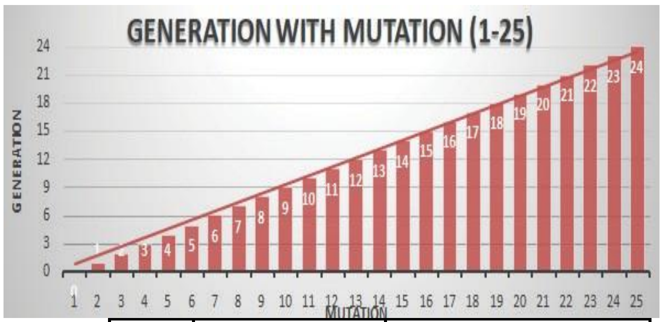
The mutations used in the fitness function calculations are listed in the Table 3. This table includes 3500 values of mutations corresponding to the generations from 0 to 3499. The corresponding chart plotting is shown in the Diagram 8 and Diagram 9.

The mutations used in the fitness function calculations are listed in the Table 3. This table includes 3500 values of mutations corresponding to the generations from 0 to 3499. The corresponding chart plotting is shown in the Diagram 8 and Diagram 9.

Table 3 Performance report of crossover with respect to generations 1 to 3499

S. No	Generation	Mutation
1	0	[[9.554e-03 5.268e-04 3.144e-01 4.704e-01 1.022e+00 1.888e+00
2	1	[[1.298e+00 1.177e+00 6.571e-01 2.449e-01 1.621e+00 1.989e+00
3	2	[[1.330e+00 4.591e-01 5.301e-01 5.491e-01 1.442e+00 5.813e-01
4	3	[[1.330e+00 4.591e-01 5.214e-01 5.491e-01 1.442e+00 5.813e-01
5	4	[[1.330e+00 4.591e-01 1.560e+00 5.491e-01 1.442e+00 5.813e-01
6	5	[[1.330e+00 4.591e-01 6.532e-01 5.491e-01 1.442e+00 5.813e-01
7	6	[[1.33 0.459 0.653 0.549 1.442 0.581 0.472 0.233 0.304 1.794
8	7	[[1.33 0.459 1.297 0.549 1.442 0.581 0.472 0.233 0.304 1.794
9	8	[[1.33 0.459 1.404 0.549 1.442 0.581 0.472 0.233 0.304 1.794
10	9	[[1.33 0.459 1.988 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
11	10	[[1.33 0.459 2.658 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
12	11	[[1.33 0.459 2.178 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
13	12	[[1.33 0.459 3.631 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
14	13	[[1.33 0.459 3.532 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
15	14	[[1.33 0.459 3.862 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
16	15	[[1.33 0.459 5.46 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
17	16	[[1.33 0.459 4.697 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
18	17	[[1.33 0.459 5.327 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
19	18	[[1.33 0.459 5.842 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
20	19	[[1.33 0.459 5.383 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
21	20	[[1.33 0.459 6.144 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
22	21	[[1.33 0.459 6.978 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
23	22	[[1.33 0.459 6.746 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
24	23	[[1.33 0.459 8.904 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
25	24	[[1.33 0.459 8.672 0.549 1.442 0.581 0.472 0.233 0.304 1.794 0.813 0.735
...	...	...
3476	3475	[[1.330e+00 4.591e-01 1.514e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3477	3476	[[1.330e+00 4.591e-01 1.514e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3478	3477	[[1.330e+00 4.591e-01 1.514e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3479	3478	[[1.330e+00 4.591e-01 1.515e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3480	3479	[[1.330e+00 4.591e-01 1.514e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3481	3480	[[1.330e+00 4.591e-01 1.516e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3482	3481	[[1.330e+00 4.591e-01 1.516e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3483	3482	[[1.330e+00 4.591e-01 1.516e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3484	3483	[[1.330e+00 4.591e-01 1.517e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3485	3484	[[1.330e+00 4.591e-01 1.516e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3486	3485	[[1.330e+00 4.591e-01 1.518e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3487	3486	[[1.330e+00 4.591e-01 1.518e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3488	3487	[[1.330e+00 4.591e-01 1.517e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3489	3488	[[1.330e+00 4.591e-01 1.519e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3490	3489	[[1.330e+00 4.591e-01 1.519e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3491	3490	[[1.330e+00 4.591e-01 1.519e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3492	3491	[[1.330e+00 4.591e-01 1.519e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3493	3492	[[1.330e+00 4.591e-01 1.520e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01
3494	3493	[[1.330e+00 4.591e-01 1.520e+03 5.491e-01 1.442e+00 5.813e-01 4.722e-01

Diagram 8 A plot of Crossover with respect to generation 0 to 25



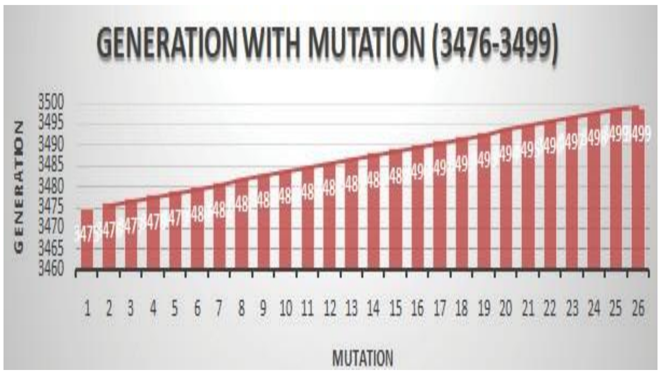


Diagram 9 A plot of Crossover with respect to generation 3476 to 3499

Finally the best fitness results regarding each generations are shown in the Table 4. This table contains totally 3500 best fitness results corresponding to the generations from 0 to 3500. The performance plots are shown in the Diagram 10 and Diagram 11 respectively.

Table 4 Performance report of crossover with respect to generations 1 to 3499

S. No	Generation	Fitness Best Results
1	0	4234045.847
2	1	4234045.847
3	2	4234045.847
4	3	4234061.291
5	4	4234085.214
6	5	4234085.214
7	6	4234099.289
8	7	4234107.588
9	8	4234116.54
10	9	4234128.26
11	10	4234147.759
12	11	4234157.28
13	12	4234172.052
14	13	4234194.75
15	14	4234194.75
16	15	4234206.693
17	16	4234211.297
18	17	4234232.55
19	18	4234238.379
20	19	4234258.945
21	20	4234266.376
22	21	4234271.671
23	22	4234281.167
24	23	4234291.301
25	24	4234306.788
...	...	...
3476	3475	4371459.477
3477	3476	4371469.385
3478	3477	4371477.433
3479	3478	4371486.902
3480	3479	4371500.597
3481	3480	4371520.492
3482	3481	4371524.752
3483	3482	4371527.37
3484	3483	4371548.796
3485	3484	4371553.493
3486	3485	4371568.07
3487	3486	4371572.284
3488	3487	4371580.822
3489	3488	4371600.723
3490	3489	4371600.723
3491	3490	4371603.29
3492	3491	4371618.07
3493	3492	4371642.919
3494	3493	4371652.185
3495	3494	4371677.36
3496	3495	4371680.295
3497	3496	4371701.739
3498	3497	4371701.739

3499	3498	4371703.338
3500	3499	4371720.187

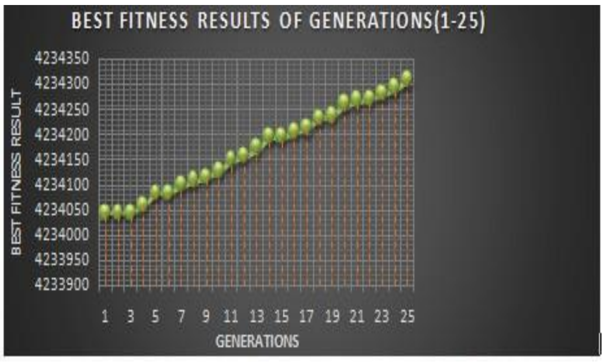


Diagram 10 A plot of Crossover with respect to generation 0 to 25

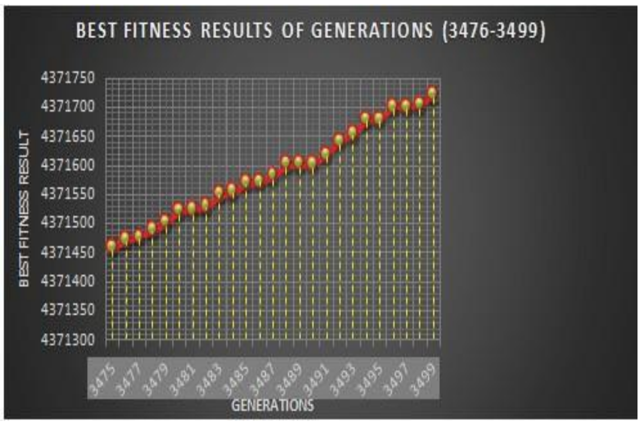


Diagram 11 A plot of Crossover with respect to generation 3476 to 3499

VII. RESULT AND CONCLUSION

Fitness function used to check the measure of how best the solution is best to the given problem. It is used to show the right solution and also used to improve the solution. If the best solution is low then, the solution is found using the fitness function is bad. If the best solution is high then, the solution is found using the Fitness function is good. This classification helps to provide the reliable details about the fault in requirement design and implementation phase.We focus to provide better classification for fault report analyst. This classification provide the fault list with their cause and occurrence phase.

In this paper fitness function used is linear classifier. This learning technique is used to find the best fitness function result of the population used. The best result is used to check how reliable the features are with the problem selected. This is used to find the optimality of the features selected. The final results of the best fitness results Best solution of fitness is **4371720.187** corresponding to the generation **3499** are shown in Table 4 which is used to provide the scattered knowledge about fault in order to reduce the rework of software projects.

REFERENCES

1. Adrian Schr¨oter, Thomas Zimmermann and Andreas Zeller, “Predicting Component Failures at Design Time”, ISESE’06, September 21–22, 2006, Rio de Janeiro, Brazil.
2. Norman E. Fenton and Martin Neil ,” A Critique of Software Defect Prediction Models,” IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 25, NO. 3, MAY/JUNE 1999.
3. Kim Herzig, Sascha just and Andreas Zeller, “It’s Not a Bug, It’s a Feature: How Misclassification Impacts Bug Prediction”.
4. AmrutaGadekar ,PranjaliTaralkar, Nikita Waghmare and Rahul Dapke ,” Finding Bug by using Data Reduced Techniques,” (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (5) , 2015, 4263-4265.
5. Ahmed Mateen , MarriamNazir and Salman Afsar Awan , “Optimization of Test Case Generation using Genetic Algorithm (GA),” International Journal of Computer Applications (0975 – 8887) Volume 151 – No.7, October 2016.
6. D. E. GOLDBERG and J. H. HOLLAND,” GUEST EDITORIAL Genetic Algorithms and Machine Learning, ,Machine Learning 3: 95-99, 1988.
7. Ali Alajmi and Jonathan Wright, ”Selecting the most efficient Genetic Algorithm sets in solving unconstrained building optimization