IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-1, Jan 2019 E-ISSN: 2347-2693

Stochastic Bio-Inspired Gene Optimization Based Trapezoidal Fuzzy Logic for Software Failure Prediction Based On Event Log Files

P. Saravanan[1*], V. Sangeetha[2]

[1]Dept. of Computer Science, Government Arts College, Dharmapuri, Tamilnadu, India
[2]Dept. of Computer Science, Government Arts College, Pappireddipatti, Dharmapuri, Tamilnadu, India

[*]*Corresponding Author : sivajisaravanans@gmail.com, Tel.: +91-9095223244*

Available online at: www.ijcseonline.org

Accepted: 19/Jan/2019, Published: 31/Jan/2019

Abstract—Software fault detection plays a significant role in the management of software systems quality to locate the fault and to identify the cause. Few research works has been developed for detecting the cause of failure occurrence from event log files. Performance of conventional software failure prediction technique was not effective. In order to overcome such limitation, a Stochastic Bio-inspired Genetic-based Trapezoidal Fuzzy Logic (SBG-TFL) Model is proposed using event log files. The SBG-TFL Model is designed to identify the failure cause with the portfolio formation of good parameters. The SBG-TFL Model at first constructs the projects portfolio with help of optimal parameters selected from event log files with application of Stochastic Bio-inspired Gene Optimization (SBGO) Algorithm. The formation of projects portfolio assists for SBG-TFL Model to reduce the amount of time taken for analysing the failure behaviour of a systems application. SBG-TFL Model applies Trapezoidal Fuzzy Logic Model to formulated projects portfolio in order to effectively predict the failure causes of software application. SBG-TFL Model increases the accuracy and true positive rate of software failure prediction. The SBG-TFL Model conducts the experimental process on metrics such as recall precision and software failure identification time with respect to different software code size. The experimental result shows that SBG-TFL Model is able to improve the precision of software failure detection and also reduces software failure identification time when compared to state-of-the-art-works.

Keywords— Event Logs, Fuzzy Rule, Paremeter, Projects Portfolio Software Failure, Trapezoidal Membership Function

I. INTRODUCTION

Software fault prediction is essential processes in software quality management. The complexity involved in software development processes software reliability models need to possess the capability of dealing with different parameters. In last decades event log used to analysis the failure behaviour of a systems application. Failure prediction performance of conventional technique was not efficient. This research work focus on software fault detection from event log files to identify the cause of software failure occurrence through projects portfolio formation. Projects portfolio formation is a task of gathering all information regarding a software application program and the process in which they are carried out and managed. A lot of research works have been designed for detecting software failure occurrence and cause prediction. A Rule Based Logging (RBL) approach was presented in [1] for analysing the software failures and improving the recall and precision rate of software failure identification. Precision of software failure detection was poor. UiLog system was intended in [2] to detect the root cause of software faults. Time taken for identification of software failure was more. An analytical model was designed in [3] for recovery from failures caused through Mandel bugs. The failure causes remained unaddressed. A fuzzy rule based algorithm was applied in [4] for efficient prediction of faults in software during requirement analysis phase. True positive rate of software failure identification was lower. Random Indexing and Support Vector Machines was used in [5] to discover software failure with aid of log files. Performance of software failure detection was not sufficient. A statistical model was developed in [6] using event logs to achieve the improved failure detection performance. Accuracy and recall of failure identification was not at required level.

Analysing and predicting effort related with predicting and fixing software faults was presented in [7]. Computational complexity of fault prediction was very higher. Systematic review of diverse techniques developed for software fault identification using machine learning techniques was examined in [8].

Quad Tree-based K-Means algorithm was intended in [9] for predicting faults in program modules. Performance of this algorithm was not efficient for attaining higher fault prediction accuracy. A taxonomical classification of diverse techniques for software fault prediction techniques was presented in [10] to improve the performance. In order to solve the above mentioned existing drawbacks Stochastic Bio-inspired Genetic-based Trapezoidal Fuzzy Logic (SBG-TFL) Model is introduced. The contributions of SBG-TFL Model is formulated.

- To enhance the performance of software failure prediction based on event log files with higher accuracy and minimum time Stochastic Bio-inspired Genetic-based Trapezoidal Fuzzy Logic (SBG-TFL) Model is developed. SBG-TFL Model is designed with help of Stochastic Bio-inspired Gene Optimization (SBGO) Algorithm and Trapezoidal Fuzzy Logic model.

- To create the projects portfolio with good parameters for multiple software projects in a given dataset Stochastic Bio-inspired Gene Optimization (SBGO) Algorithm is applied in SBG-TFL Model. SBGO Algorithm performs rank selection two point crossover and Gaussian mutation in order to select the optimal parameters for the projects portfolio formation.
- To identify the root cause of software failures with higher recall. Trapezoidal Fuzzy Logic model is employed in SBGO Algorithm. Trapezoidal Fuzzy Logic model employed trapezoidal membership function and fuzzy rule in order to find the cause of software failure occurrences.

The rest of structure is ordered as follows In Section 2, Stochastic Bio-inspired Genetic-based Trapezoidal Fuzzy Logic (SBG-TFL) Model is explained with assists of architecture diagram. In Section 3, Simulation settings are explained and the result discussion is presented in Section 4. Section 5 describes the related works. Section 6 concludes the paper.

II. STOCHASTIC BIO-INSPIRED GENETIC-BASED TRAPEZOIDAL FUZZY LOGIC MODEL

A Stochastic Bio-inspired Genetic-based Trapezoidal Fuzzy Logic (SBG-TFL) Model is designed with aim of improving the performance of software failure discovery to detect the root cause of software faults from event log files. In existing rule-based logging approach was developed to find software failures using event log files. Precision rate of software failure prediction was poor. Further UiLog system was developed to identify fault types and discover the root cause of the software faults. Optimizing the procedure of log processing was remained unsolved which increases the time complexity for predicting root cause of software faults.In order to overcome the above mentioned existing issues in software failure and cause prediction. A Stochastic Bio-inspired Genetic-based Trapezoidal Fuzzy Logic (SBG-TFL) Model is introduced. The SBG-TFL Model is developed with application of Stochastic Bio-inspired Gene Optimization (SBGO) Algorithm and Trapezoidal Fuzzy Logic model in order to achieve higher precision and to minimize the software failure identification time. The overall architecture diagram of Stochastic Bio-inspired Genetic-based Trapezoidal Fuzzy Logic (SBG-TFL) Model is demonstrated in Figure 1.

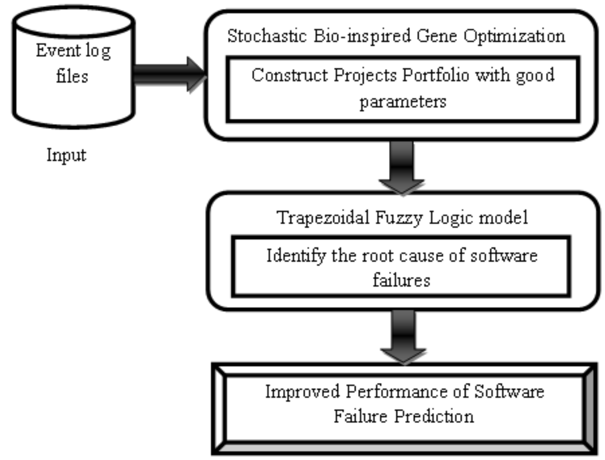

Figure.1 Architecture Diagram of Genetic-Based Fuzzy Logic Frame

Figure 1 demonstrates the process involved in SBG-TFL Model to enhance the performance of event log based software failure cause detection. As depicted in Figure 1, SBG-TFL Model initially takes the event log files as input. Then, SBG-TFL Model employed Stochastic Bio-inspired Gene Optimization to build the projects portfolio with good parameters from event log files with aiming at minimizing the time taken for software failure identification. The projects portfolio formation in SBG-TFL Model includes of all the information about activities of multiple software users' for efficient software failure prediction. After that, SBG-TFL Model applied Trapezoidal Fuzzy Logic model on constructed projects portfolio with objective of determining the root cause of software failures with higher accuracy. As a result SBG-TFL Model increased the precision and recall of software failure cause identification and also reduces the software failure identification time in an effective manner. The detailed process of proposed SBG-TFL Model is explained in forthcoming subsections.

A. Stochastic Bio-inspired Gene Optimization Algorithm

Stochastic Bio-inspired Gene Optimization (SBGO) Algorithm is used in SBG-TFL Model in order to create the projects portfolio with good parameters for identifying the cause of failures in software project management. On the contrary to existing works SBG-TFL Model employed SBGO Algorithm in order to form projects portfolio with good parameters for analysing the cause of software failure occurrence. In SBGO Algorithm rank selection two point crossover and Gaussian mutation is performed to select the optimal parameters from event log files for projects portfolio formation which is main contribution of proposed SBG-TFL Model.

The SBGO Algorithm is a random probabilistic search method inspired by the natural process of biological evolution. The SBGO Algorithm operates on a population of potential solutions. The SBGO Algorithm is utilized to make high-quality solutions to optimization based on bio-inspired operators such as rank selection; two point crossover and Gaussian mutation. The evolution of SBGO Algorithm initiates from a population of randomly generated individuals. The population in every iteration termed a generation. During each generation, the fitness of individual in the population is determined. The fitness refers to value of the objective function in the optimization problem being solved. The more fit individuals are stochastically chosen from the current population to form projects portfolio and each individual's chromosome is changed to form a new generation. The process of SBGO Algorithm terminates when an optimal solution is attained.

Besides, Projects portfolio formation in SBG-TFL Model is a task of collecting the all information about a software application program and the function in which they are performed and managed. Let consider the software application program includes of many small projects like $Pr_1, Pr_2, Pr_{3,...} Pr_n$. The SBGO Algorithm employed in SBG-TFL Model formulates the projects portfolio with good parameters based on event log files. The event logs plays a key role in software failure analysis as they include of significant information about regular and anomalous system activities under real workload conditions. An event log is used by industry and academia to analysis the failure behaviour of a variety of software systems. An event logs are system-generated files which maintains sequences of events occurred during operations in the form of text-entries. The SBG-TFL Model used event log files collected from Blue Gene/P Intrepid system in order to find out the cause of software failure occurrence. Each log entry consists of 15 fields namely RECID, MSG_ID, COMPONENT, SUBCOMPONENT, ERRCODE, SEVERITY, EVENT_TIME, FLAGS, PROCESSOR, NODE, BLOCK, LOCATION, SERIALNUMBER, ECID, and MESSAGE. The description about parameters in event logs file is shown in below Table 1.

Table.1 Parameters Information about Event Logs

Attribute Name	Description
RECID	Series number for an event record in log
MSG-ID	Source of the message
COMPONENT	Software component detecting and reporting the event
SUBCOMPONENT	Area that generated the message
ERRCODE	Fine-grained event type information
SEVERITY	INFO, WARNING, ERROR, FATAL)
EVENT_TIME	The start time of event
LOCATION	The location where the event occurs
MESSAGE	overview of event condition

Table 1 demonstrates the parameters information regarding event log files to construct the projects portfolio for identifying failures cause of software application. In order to select the good parameters for projects portfolio construction Stochastic Bio-inspired Gene Optimization (SBGO) Algorithm is designed in proposed SBG-TFL Model. In SBGO Algorithm, a stochastic means a random process which refers to a collection of random parameter in event log files. The process involved in Stochastic Bio-inspired Gene Optimization (SBGO) Algorithm for projects portfolio construction with optimal parameters is shown in below Figure 2.

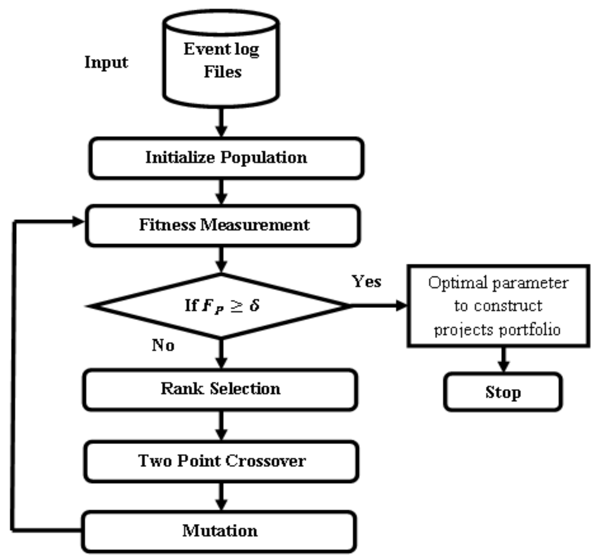

Figure.2 Process of Stochastic Bio-Inspired Gene Optimization Algorithm for Optimal Parameter Selection to Form Projects Portfolio

Figure 2 depicts the block diagram of SBGO Algorithm to formulate the projects portfolio with good parameters for detecting cause of software failures. As illustrated in Figure 2, SBGO Algorithm includes of initialization, fitness measurement, rank selection, two point crossover and Gaussian mutation process in order to choose the optimal parameters from event log files for projects portfolio construction. The input to SBGO Algorithm is an event log files. Initially, SBGO Algorithm initializes population with assists of parameters in an event log files. Subsequently, fitness value is evaluated for each parameter in event log files based on dependence of parameters to software failure identification. If fitness value of parameter is greater than a threshold δ, then the parameter is selected as optimal to form projects portfolio. Otherwise, rank selection, two point crossover and Gaussian mutation process is carried out to create the new offspring. The process of SBGO Algorithm is repeated until the optimal parameters are chosen for projects portfolio construction. After selecting the optimal parameters, projects portfolio is created for multiple software projects in given software application in order to effectively identify the root cause of software failures. The detailed process of SBGO Algorithm is shown in below

B. Initialization
SBGOA randomly initializes the gene population with support a set of parameters an event log files.

C. Fitness Measurement
Fitness function is an objective function which is utilized to find an optimal solution for a given problem the problem is to select the good parameters for projects portfolio construction. Fitness function is calculated in SBGO Algorithm for each parameter in an event log files in order to select the optimal parameter for creating projects portfolio. In SBGO Algorithm, fitness function measurement based on correlation or dependence of parameters to predict the software failures. The correlation between the parameters is determined using Pearson correlation in SBGO Algorithm. fitness value of parameters in an event log files computed using below mathematical formula,

$$F_P = \frac{\sum P_1P_2 - \frac{(\sum P_1)(\sum P_2)}{n}}{\sqrt{\left(\sum {P_1}^2 - \frac{(\sum P_1)^2}{n}\right)(\sum {P_2}^2 - \frac{(\sum P_2)^2}{n})}} \quad (1)$$

From equation (1), n represents the number of parameters in an event log files, P_1 and P_2 are ithparameters. $\sum P_1P_2$ indicates sum of cross product of $P_1$and P_2, $\sum P_1$ indicates the dependence of parameter P_1 with respect to P_2, $\sum P_2$ point outs the dependence of parameter P_2 with respect toP_1. By using equation (1), fitness value is estimated for each parameter in an event log files. SBGO Algorithm selects the parameter with higher fitness value for constructing the projects portfolio to predict the software failures.

D. *Rank Selection Operation*
In SBGO Algorithm, Selection operation is used to selects the population from initial population. During the selection process parameter with higher fitness values is elected in order to form projects portfolio. Proposed SBGO Algorithm used rank selection for performing selection operation. Rank Selection sorts the population based on fitness value and ranks them. Parameters in an event log files are selected as per their selection probability. Rank selection is an explorative technique of selection employed in SBGO Algorithm. In Rank Selection, sum of ranks is evaluated and then selection probability of each individual is determined. Thus the sum of ranks of parameters $R_{Sum_i}$is evaluated as,

$$R_{Sum_i} = \sum_{i=1}^{n} P_{R_{i,j}} \quad (2)$$

From equation (2),$i$and $j$varies from one to$n$ generation where $P_{R_{i,j}}$ denotes the rank of parameter. Followed by, selection probability of each parameter is mathematically obtained as follows,

$$RankSelection = \frac{P_{R_{i,j}}}{R_{Sum_i}} \quad (3)$$

E. Two Point Crossover(TPC)
A crossover operation is performed in SBGO Algorithm in order to change chromosomes of parameters from one generation to the next generation to efficiently perform projects portfolio formation with good parameters. The SBGO Algorithm designed in SBG-TFL Model used two Point Crossover. The Two point Crossover changes chromosomes from one generation to next generation. In addition Two point Crossover process takes more than one parent chromosomes and generates an offspring from them.

F. Gaussian Mutation Operation
The mutation operation performed SBGOA order to randomly change the value of each bit of parameter chromosome along with the probability. The SBGOA used Gaussian Mutation (GM) to randomly change the points chromosome for formulating projects portfolio. GM employed in SBGOA maintains genetic diversity from one generation to the next generation. GM utilized in proposed SBGOA and mutation chooses a random point from each population an individual new offspring chromosome. Accordingly Gaussian density function population 'X' is mathematically determined.

$$f_{Ga(0,\sigma^2)}(x) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{x^2}{2\sigma^2}} \quad (4)$$

From equation (4) σ^2 denotes the variance population and y indicates a selected gene population consider $x \in [m,n]$ a real variable. The Gaussian mutation '$Mutation_{Ga}$' change the population 'X' into the next generation using below formulation.

$$\mathrm{Mutation_{Ga}(x) := min(max(N(x,\sigma), m), n)} \quad (5)$$

From equation(5) SBGOA carried outs the GM operation for effectively construct project portfolio with good parameter and predicting the SF cause with higher accuracy and the Algorithmic process of SBGO. Stochastic Bio-Inspired Gene Optimization based Projects Portfolio Construction

Input: Collections of parameters in event log files, collection of projects in software application
Output: Construct Projects portfolio with good parameters
Step 1: Begin
Step 2: Initialize population with aid of parameters in event log files
Step 3: Determine fitness function using (1)
Step 4: If (fitness value of parameter $F_P > \delta_T$) then
Step5: Select parameter as optimal to construct projects portfolio
Step 6: else
Step 7: Rank selection operation is carried out using (2)
Step 8: Two point crossover is performed

Step 9: Gaussian Mutation is performed using (4) and (5)
Step 10: Go to Step 3
Step 11: End if
Step12: For each small projects in a given software application
Step13: construct projects portfolio with selected optimal parameters using event logs
Step 14: End for
Step 15: End
Algorithm 1 Stochastic Bio-Inspired Gene Optimization

Algorithm 1 demonstrates the step by step process of Stochastic Bio-Inspired Gene Optimization algorithm for projects portfolio creation. By using the above algorithmic process SBGO Algorithm efficiently form portfolio for multiple projects in a given software application program with optimal parameters. The created projects portfolio helps for SBG-TFL Model to analysis the cause of software failures with minimum time. The SBG-TFL Model reduces the amount of time required for software failures identification in a significant manner.

G. Trapezoidal Fuzzy Logic model

After formulating projects portfolio, Trapezoidal Fuzzy Logic model is applied to identify the software failure causes. On the contrary to conventional techniques SBG-TFL Model used Trapezoidal Fuzzy Logic model for software failure identification which is another contribution of proposed technique. Trapezoidal Fuzzy Logic model initiates with the concept of a fuzzy set. In Trapezoidal Fuzzy Logic model, membership function is a function which indicates the degree to which a given input belongs to a set. The SBG-TFL Model employed Trapezoidal membership function for efficient software failure cause prediction. The output of a Trapezoidal membership function is ranges between 0 and 1. In Trapezoidal Fuzzy Logic system, the input space is represented as the universe of discourse. The proposed SBG-TFL Model considers the severity level in projects portfolio formation in order to identify the root cause of software failure occurrences. The severity level in projects portfolio construction includes of four levels namely INFO, WARNING, ERROR, and FATAL. INFO events present information about the progress of system software. WARNING events are recoverable “soft” errors for example ECC correctable, single-symbol error. ERROR denotes harmful events that allow the application to continue running for example the failure of a redundant component. FATAL represents an application or system crash. Therefore, SBG-TFL Model focuses on events with “FATAL” severity to discover the cause of software failures.

The Trapezoidal membership function which denotes a fuzzy set X is represented as μ_X. For each project Pr_i in software application program with constructed portfolios in X, the Trapezoidal membership value $\mu_{X(Pr_i)}$ is termed as membership degree of Pr_i in the fuzzy set X. The trapezoidal membership degree $\mu_{X(Pr_i)}$ signifies the grade of membership of set of software projects Pr_i to the fuzzy set X. Here, value 0 indicates that software project Pr_i is not a member of the fuzzy set whereas value 1 represents that software project Pr_i is fully a member of the fuzzy set. A Trapezoidal Fuzzy Membership function is a curve which illustrates how the root cause of software project failure is predicted based on formulated projects portfolio. The event consider a fuzzy set collection of software projects $X = \{Pr_1, Pr_2, Pr_3\}$ in given software application with constructed projects portfolio. The trapezoidal membership function of project Pr_i is obtained using fuzzy rule. The Trapezoidal Fuzzy Logic model constructs the following fuzzy rule to find the Trapezoidal membership function for each project in a given data set to identify the root cause of software failures. The fuzzy rule designed for predicting the cause of software failures is mathematically formulated as follows,

$$= \begin{cases} if\ severity\ level\ of\ project\ portfolio\ is\ FATAL, then \mu_{X(Pr_i)} = 1 \\ else, \qquad \mu_{X(Pr_i)} = 0 \end{cases}$$

From equation (6) Trapezoidal membership value determined for each project in a fuzzy set with formulated projects portfolio given software program. The output of Trapezoidal fuzzy member function is shown in below.

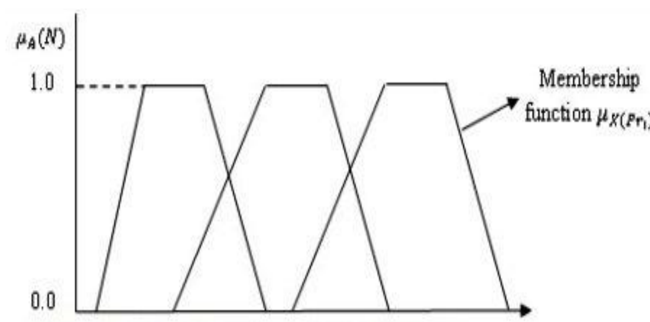

Figure.3 Output of Trapezoidal Fuzzy Membership function

Figure 3 depicts the output of trapezoidal fuzzy member function to detect the root cause of software failures from constructed projects portfolio information. The Trapezoidal membership values 0 and 1 point outs the fuzzy members that belong to fuzzy set partially. The Trapezoidal membership value 1 indicates that the prediction of software failure causes. On the contrary membership value 0 indicates that the there is no failure occurrence. The algorithmic process of Trapezoidal Fuzzy Logic model for identification of software failures cause is shown in below.

Trapezoidal Fuzzy Logic based Software Failure Prediction Algorithm

Input: Collections of Software Projects $Pr_i = Pr_1, Pr_2, Pr_3, .. Pr_n$, Constructed Projects Portfolio
Output: Improved Precision and reduced software failure identification time
Step 1: Begin
Step 2: For each projects in software program with portfolio formation
Step 3: Construct the Trapezoidal fuzzy member function $\mu_{X(Pr_i)}$ with aid of fuzzy rule
using (6)
Step 4: If ($\mu_{X(Pr_i)} = 1$, then
Step 5: The root cause of software failure occurrence is detected
Step 6: else
Step 7: No software failure occurrence
Step 8: End if
Step 9: End for
Step 10: End

Algorithm.2 Trapezoidal Fuzzy Logic based Software Failure Prediction

Algorithm 2 shows step by step algorithmic process involved Trapezoidal Fuzzy Logic model for software failure identification. The above algorithmic process Trapezoidal Fuzzy Logic model effectively detect the root cause of software failures occurrence which resulting in improved precision and recall for software failure prediction.

III. EXPERIMENTAL SETTINGS

In order to validate the performance of proposed, Stochastic Bio-inspired Genetic-based Trapezoidal Fuzzy Logic (SBG-TFL) Model is implemented in java language using event log files collected from Blue Gene/P Intrepid system [21]. This event log files are gathered from a period of 6 months on the Blue Gene/P Intrepid system. The attributes in the log includes of following 15 fields namely RECID, MSG_ID, COMPONENT, SUBCOMPONENT, ERRCODE, SEVERITY, EVENT_TIME, FLAGS, PROCESSOR, NODE, BLOCK, LOCATION, SERIALNUMBER, ECID, and MESSAGE. The elaborate descriptions about these attributes are already depicted in Table 1. The SBG-TFL Model selects the optimal parameters from event log files in order to form projects portfolio for effective identification of software failure causes. The performance of SBG-TFL Model is evaluated in terms of precision recall and software failure identification time. The experimental evaluation of SBG-TFL Model is carried out for many instances with respect to different number of projects and software code sizes and averagely ten results is demonstrated in graph and tables.

IV. RESULTS AND DISCUSSIONS

The result analysis of proposed SBG-TFL Model is discussed in this section. The performance of SBG-TFL Model is compared against with existing Rule Based Logging (RBL) approach [1] and UiLog system [2] respectively. The effectiveness of SBG-TFL Model is estimated along with the following parameters with the aid of tables and graphs.

A. Measure of Precision

In SBG-TFL Model, Precision (P) is defined as the ratio number of failures causes detected based on projects portfolio formation the number of source code in software program taken as input the precision is measured in terms of percentages (%) and formulated.

$$P = \frac{\text{Number of failures cause detected}}{\text{Software code size}} * 100 \quad (7)$$

From equation (7) precision for predicting the SF is determined with respect different size of software code size. When precision of SFI higher method said to be more effectual.

Table.2 Tabulation for Precision

Software Code Size (KB)	Precision (%)		
	RBL approach	UiLog system	SBG-TFL Model
10	65	71	88
20	66	74	89
30	69	75	90
40	70	76	91
50	71	78	92
60	73	79	94
70	75	81	95
80	76	82	96
90	78	83	97
100	79	85	98

Table 2 shows the tabulation result analysis of precision for software failures causeidentification based on diverse size of software code taken in the range of 10 KB-100 KB. The SBG-TFL Model considers the framework with different number of software code size for predicting the root cause of software fault occurrence using java language.When considering the 60 KB software program code size for conducting the experimental process, proposed SBG-TFL Model attains 94 % precision whereas RBL approach [1] and UiLog system [2] acquires 73 % and 79 % respectively. From that, it is descriptive that the precision obtained forsoftware failures causes identification using proposed SBG-TFL Model is higher than existing works [1], [2]. Based on the above table values, the graph is drawn is below figure.

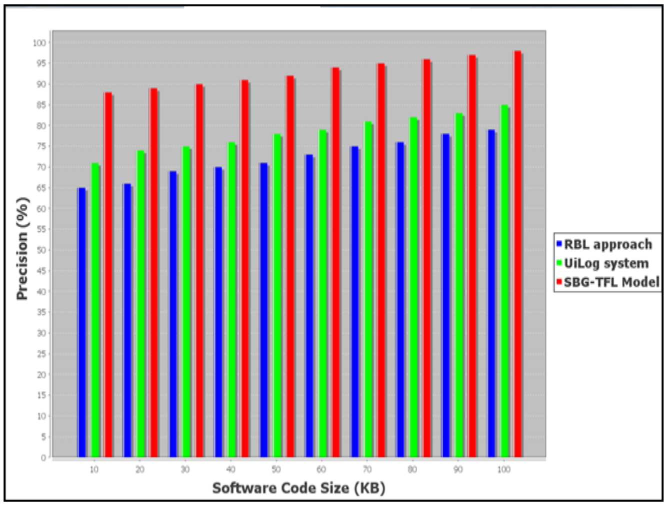

Figure.4 Measurement of Precision for Software Failure Identification Time

proposed SBG-TFL Model attains 94 % precision whereas RBL approach [1] and UiLog system [2] acquires 73 % and 79 % respectively. From that, it is descriptive that the precision obtained for software failures causes identification using proposed SBG-TFL Model is higher than existing works [1], [2].

Figure 4 describes the impact of precision result is obtained for software failures cause identification versus varied number of software code size in the range of 10 KB – 100 KB using three methods namely RBL approach [1] and UiLog system [2] and SBG-TFL Model. As illustrated in figure, the proposed SBG-TFL Model provides better precision to identify the root cause of software failures as compared to RBL approach [1] and UiLog system [2]. Further while increasing the size of software program code the precision of software failures detection is also gets increased for all three methods. But, comparatively the precision using proposed SBG-TFL Model is higher. This is because of the application of Stochastic Bio-Inspired Gene Optimization algorithm and Trapezoidal Fuzzy Logic model in proposed SBG-TFL Model.

The Stochastic Bio-Inspired Gene Optimization algorithm developed in SBG-TFL Model formulates the portfolio for an each projects in given software application based on event log files. Projects portfolio collects the all activities of multiple software users' about software program to predict the failure causes. After that, the Trapezoidal Fuzzy Logic model designed in SBG-TFL Model applied to create projects portfolio in order to find the root cause of software fault occurrences. Trapezoidal Fuzzy Logic model computes the trapezoidal membership function for each projects with portfolio formation in a given software application using fuzzy rule. The determined trapezoidal membership function assists for SBG-TFL Model to efficiently identify the root cause of failure occurrence with higher accuracy. This process resulting in enhanced precision for software failures causes detection. The SBG-TFL Model increases the precision of software failures cause prediction by 29 % and 19 % when compared to RBL approach [1] and UiLog system [2] respectively.

B.Measure of Recall

SBG-TFLM Recall (R) is measured the ratio number of failures causes that correctly detected based on projects portfolio formation the number of source code in software program. The recall true positive rate estimated in terms of percentages (%) mathematically represented.

$$P = \frac{\text{Number of failures causes that are correctly detected}}{\text{Software code size}} * 100$$

From equation (8) recall for detecting the software failure computed with respect diverse size of software code size and recall SF prediction is higher the method is said to be more effective.

Table.3 Tabulation for Recall

Software Code Size (KB)	**Recall (%)**		
	RBL approach	**UiLog system**	**SBG-TFL Model**
10	61	68	82
20	62	71	84
30	64	72	86
40	67	75	87
50	68	76	88
60	69	78	90
70	71	79	91
80	73	80	92
90	75	81	93
100	76	83	94

The tabulation 3 shows the result analysis of recall for software failures cause prediction based on varied size of software code size taken in the range of 10 KB-100 KB is demonstrated in Table 3. When considering the 70 KB software program code size for performing the experimental evaluation, proposed SBG-TFL Model achieves 91 % recall whereas RBL approach [1] and UiLog system [2] gets 71 % and 79 % respectively. From that, it is expressive that the recall result attained for software failures causes detection using proposed SBG-TFL Model is higher than existing works [1], [2]. By using on the above table values, the graph is plotted is below figure.

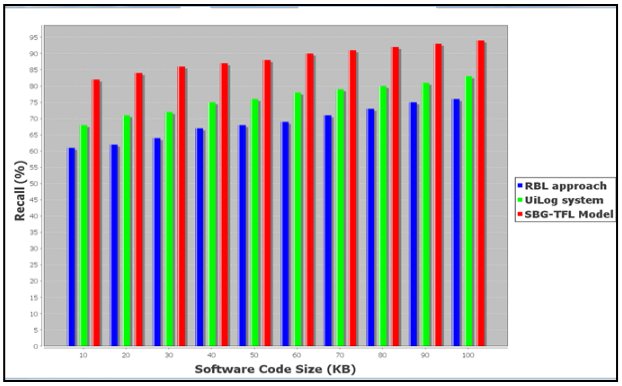

Figure.5 Measurement of Recall for Software Failure Identification Time

Figure 5 explains the impact of recall result is acquired for software failures cause prediction versus different number of software code size in the range of 10 KB – 100 KB using three methods namely RBL approach [1] and UiLog system [2] and SBG-TFL Model. As depicted in figure the proposed SBG-TFL Model provides better recall to detect the root cause of software failures occurrence as compared to RBL approach [1] and UiLog system [2]. Besides, while increasing the size of software program code the recall of software failures identification is also gets increased for all three methods. The comparatively the recall using proposed SBG-TFL Model is higher. This is owing to the application of Trapezoidal Fuzzy Logic model in proposed SBG-TFL Model.

With application of Stochastic Bio-Inspired Gene Optimization algorithm SBG-TFL Model creates the projects portfolio for a given software application using event log files where it gathers multiple users activities of software application for effective fault cause analysis. After formation of projects portfolio Trapezoidal Fuzzy Logic model utilized in SBG-TFL Model applied to discover the root cause of software fault occurrences. The SBG-TFL Model evaluates the trapezoidal membership function for every constructed projects portfolio in a given software application with aid of fuzzy rule. The estimated trapezoidal membership function supports for SBG-TFL Model to correctly predict the cause of failure occurrence of software application. This process results in improved true positive rate to detect the root cause of software failures occurrences. The SBG-TFL Model enhances the recall of software failures cause detection by 30 % and 16 % as compared to RBL approach [1] and UiLog system [2] respectively.

C.Measure the Software Failure Identification Time

The SBG-TFLM Software failure identification time(SFIT) measures the amount of time taken to identify the SF and identification time is measured in terms of milliseconds(ms) and expressed.

$$SFIT = N * t \quad (9)$$

From equation(9) time taken for predicting SF from ELF evaluated with respect to diverse size of software code size(N). The SFI time is lower the method is said to be more efficient.

Table.4. Tabulation for Software Failure Identification Time

Software Code Size (KB)	Software Failure Identification Time (ms)		
	RBL approach	UiLog system	SBG-TFL Model
10	23.2	20.9	14.7
20	28.5	26.1	18.1
30	35.9	30.2	23.5
40	40.1	34.4	29.7
50	48.4	41.7	35.4
60	53.2	46.3	39.3
70	59.6	51.2	44.5
80	64.2	55.8	49.2
90	68.1	61.4	54.8
100	72.8	65.2	58.6

The tabulation result analysis of amount of time taken for software failures causes detection based on different size of software code size in the range of 10 KB-100 KB is portrayed in Table 4. When taking the 80 KB software program code size for carried outing the experimental work proposed SBG-TFL Model consumes 49.2ms software failure identification time whereas RBL approach [1] and UiLog system [2] gets 64.2ms and 55.8ms respectively. From it is descriptive that the software failure identification time using proposed SBG-TFL Model is lower than existing works [1], [2]. With help of above table values the graph is drawn is below figure.

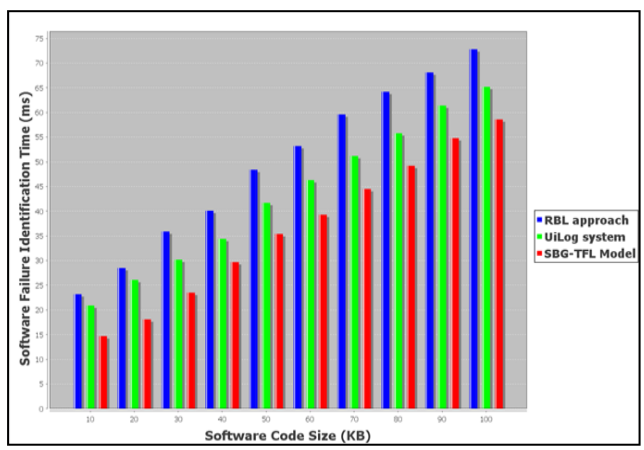

Figure.6 Measurement of Software Failure Identification Time

Figure 6 describes the impact of software failure identification time result versus various size of software code in the range of 10 KB – 100 KB using three methods namely RBL approach [1] and UiLog system [2] and SBG-TFL Model. As exposed in figure the proposed SBG-TFL Model provides better software failure identification time when compared to RBL approach [1] and UiLog system [2]. In addition while increasing the size of software program code, the software failure identification time is also gets increased for all three methods. Comparatively the software failure identification time using proposed SBG-TFL Model is lower. This is due to the application of Stochastic Bio-Inspired Gene Optimization algorithm in proposed SBG-TFL Model. With aid of Stochastic Bio-Inspired Gene Optimization algorithmic process SBG-TFL Model formulates the projects portfolio with good parameters using event log files. Stochastic Bio-Inspired Gene Optimization selects optimal parameters from event log files for projects portfolio formation through initialization fitness measurement rank selection two point crossover and Gaussian mutation process. The construction of projects portfolio for a given software application helps for SBG-TFL Model to lessens the complexity of software failure analysis. SBG-TFL Model significantly predicts the cause of failure occurrence of software application with minimum amount of time. SBG-TFL Model minimizes the software failure identification time by 27 % and 17 % as compared to RBL approach [1] and UiLog system [2] respectively.

V. RELATED WORKS

Bayesian Network Classifiers was employed in [11] to discover the faulty software codes. Predicting the cause of failure occurrence was remained unsolved. A multivariate fuzzy logic and fuzzy time series based prediction algorithm was introduced in [12] to detect the web software faults. However, processing time was higher.

A Mamdani type fuzzy inference system (FIS) was designed in [13] with aiming at solving the issues in software fault detection problem. However, recall of software fault detection was remained unaddressed. A Combined-Learning Based Framework was presented in [14] in order to improve software fault prediction accuracy. Computational complexity of failure detection was not considered.

A Neuro-Based Software Fault Prediction was developed in [15] with Box-Cox Power Transformation to improve the software reliability. Fault Prediction performance was poor. A cluster based fault prediction classifiers were introduced in [16] that improves the probability of fault detection with higher accuracy cause of failure occurrence and time complexity was remained unsolved.

Analytics-driven testing (ADT) was designed in [17] with application of a forecasting regression model to identify where and what types of software system failures is occurred. Information Intelligence Process Models was presented in [18] to predict defects in the software technology project. Fault detection accuracy was lower. The Asymmetric Kernel Principal Component Analysis Classifier (AKPCAC) was introduced in [19] in order to perform software defect prediction. A kernel based asymmetric learning method was developed in [20] with objective of improving software defect prediction performance time needed for failure identification was higher.

VI. CONCLUSION

An effective Stochastic Bio-inspired Genetic-based Trapezoidal Fuzzy Logic (SBG-TFL) Model is developed with objective of detecting the failure cause of software application from event log files with higher precision and minimum time. The objective of SBG-TFL Model is achieved with application of Stochastic Bio-inspired Gene Optimization (SBGO) Algorithm and Trapezoidal Fuzzy Logic Model. With assist of SBGO Algorithm the SBG-TFL Model initially creates the projects portfolio with support of optimal parameters chosen from event log files. This process resulting in reduced time for software failure identification. The SBG-TFL Model used Trapezoidal Fuzzy Logic Model on formulated projects portfolio to find the failure causes of software application which resulting in enhanced precision for software failure detection. The performance of SBG-TFL Model is estimated in terms of recall, precision, software failure identification time and compared with two existing methods. The experimental result illustrates that SBG-TFL Model provides better performance with an improvement of recall of software failure prediction and reduction of software failure identification time when compared to state-of-the-art works.

REFERENCES

[1] Marcello Cinque, Domenico Cotroneo, and Antonio Pecchia, "Event Logs for the Analysis of Software Failures: a Rule-Based Approach", IEEE Transactions on Software Engineering, Volume 39, Issue 6, Pages 806 – 821, 2013
[2] De-Qing Zou, Hao Qin, Hai Jin, "UiLog: Improving Log-Based Fault Diagnosis by Log Analysis", Journal of Computer Science and Technology, Springer, Volume 31, Issue 5, Pages 1038–1052, September 2016
[3] Michael Grottke, Dong Seong Kim, Rajesh Mansharamani, Manoj Nambiar, Roberto Natella, and Kishor S. Trivedi, "Recovery From Software Failures Caused by Mandelbugs", IEEE Transactions On Reliability, Volume 65 , Issue 1, Pages 70 – 87, July 2015
[4] Subhashis Chatterjee, Bappa Maji, "A New Fuzzy Rule Based Algorithm for Estimating Software Faults in Early Phase of Development", Soft Computing, Springer, Volume 20, Issue 10, Pages 4023–4035, June 2015
[5] Ilenia Fronza, Alberto Sillitti, Giancarlo Succi, Mikko Terho, Jelena Vlasenko, "Failure prediction based on log files using Random Indexing and Support Vector Machines", Journal of Systems and Software, Volume 86, Issue 1, Pages 2-11, January 2013
[6] Yuan Yuan, Shiyu Zhou, Crispian Sievenpiper, Kamal Mannar, Yibin Zheng, "Event log modeling and analysis for system failure

prediction", IIE Transactions Journal, Volume 43, Issue 9, Pages 647-660, 2011
[7] Maggie Hamill, Katerina Goseva-Popstojanova, "Analyzing and predicting effort associated with finding and fixing software faults", Information and Software Technology, Elsevier, Volume 87, Pages 1-18, July 2017
[8] Ruchika Malhotra, "A systematic review of machine learning techniques for software fault prediction", Applied Soft Computing, Elsevier, Volume 27, Pages 504-518, February 2015
[9] Partha S. Bishnu, Vandana Bhattacherjee, "Software Fault Prediction Using Quad Tree-Based K-Means Clustering Algorithm", IEEE Transactions on Knowledge and Data Engineering, Volume 24, Issue 6, Pages 1146 – 1150, June 2012
[10] Santosh S. Rathore, Sandeep Kumar, "An empirical study of some software fault prediction techniques for the number of faults prediction", Soft Computing, Springer, Volume 21, Issue 24, Pages 7417–7434, 2016
[11] Karel Dejaeger, Thomas Verbraken, Bart Baesens, "Toward Comprehensible Software Fault Prediction Models Using Bayesian Network Classifiers", IEEE Transactions on Software Engineering, Volume 39, Pages 237-257, 2013
[12] Subhashis Chatterjee, Arunava Roy, "Novel Algorithms for Web Software Fault Prediction", Quality and Reliability Engineering International, Wiley Publications, Volume 31, Pages 1517–1535, 2015
[13] Ezgi Erturk, Ebru Akcapinar Sezer, "Software fault prediction using Mamdani type fuzzy inference system", International Journal of Data Analysis Techniques and Strategies, Volume 8, Issue 1, Pages 14 – 28, 2016
[14] Chubato Wondaferaw Yohannese, Tianrui Li, "A Combined-Learning Based Framework for Improved Software Fault Prediction", International Journal of Computational Intelligence Systems, Volume 10, Issue 1, Pages 647 – 662, 2017
[15] Momotaz Begum, Tadashi Dohi, "A Neuro-Based Software Fault Prediction with Box-Cox Power Transformation", Journal of Software Engineering and Applications, Volume 10, Pages 288-309, 2017
[16] Pradeep Singh and Shrish Verma, "An Efficient Software Fault Prediction Model using Cluster based Classification", International Journal of Applied Information Systems, Volume 7, Issue 3, Pages 35-41, May 2014
[17] Feras A. Batarseh, Avelino J. Gonzalez, "Predicting failures in agile software development through data analytics", Software Quality Journal, Springer, Pages 1–18, 2015
[18] Manjula Gandhi Selvaraj, Devi Shree Jayabal, Thenmozhi Srinivasan, and Palanisamy Balasubramanie, "Predicting Defects Using Information Intelligence Process Models in the Software Technology Project", The Scientific World Journal, Hindawi Publishing Corporation, Volume 2015 (2015), Article ID 598645, Pages 1-6, 2015
[19] Jinsheng Ren, Ke Qin, Ying Ma, and Guangchun Luo, "On Software Defect Prediction Using Machine Learning", Journal of Applied Mathematics, Hindawi Publishing Corporation, Volume 2014 (2014), Article ID 785435, Pages 1-8, 2014
[20] Ying Ma, Guangchun Luo, "Kernel Based Asymmetric Learning for Software Defect Prediction", IEICE Transaction on Information & System, Issue 1, Pages 1-4, January 2010

Authors Profile

P. Saravanan.MCA.,SET., Assistant Professor in Department of computer science, Government Arts College, Dharmapuri, Tamilnadu, India.

Dr.V.Sangeetha. M.Sc., Mphil., P.hD, Assistant Professor in Department of computer science, Government Arts College, Pappireddipatti, Tamilnadu, India.

