Received September 11, 2019, accepted October 1, 2019, date of publication October 10, 2019, date of current version October 22, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2946862*

# A New Model for Predicting Component-Based Software Reliability Using Soft Computing

**CHANDER DIWAKER[1], PRADEEP TOMAR[2], ARUN SOLANKI[2], ANAND NAYYAR[3], NZ JHANJHI[4], AZWEEN ABDULLAH[4], AND MAHADEVAN SUPRAMANIAM[5]**

[1]Department of Computer Science and Engineering, U.I.E.T., Kurukshetra University, Kurukshetra 136119, India
[2]Department of Computer Science and Engineering, Gautam Buddha University, Greater Noida 201308, India
[3]Graduate School, Duy Tan University, Da Nang 550000, Vietnam
[4]School of Computing and IT (SoCIT), Taylor's University, Subang Jaya 47500, Malaysia
[5]Research and Innovation Management Centre, SEGi University and Colleges, Petaling Jaya 47810, Malaysia

Corresponding author: NZ Jhanjhi (noorzaman.jhanjhi@taylors.edu.my)

**ABSTRACT** Software engineering is the process of developing software by utilizing applications of computer engineering. In the present day, predicting the reliability of the software system become a recent issue and an attractive issue for the research area in the field of software engineering. Different techniques have been applied to estimate and predict the reliability of a system. To make new software from the beginning is a difficult task. Component-Based Software Engineering (CBSE) helps in minimizing these efforts in making new software because it utilizes factors like reusability, component dependency, and component interaction that results in decreasing complexity of the system. Soft computing may be applied to estimate reliability. A new model is proposed to estimate the reliability of Component-based Software (CBS) using series and parallel reliability models and later on, the proposed component-based software reliability model is evaluated using two soft computing techniques- Fuzzy Logic and PSO. The experimental results conclude that the proposed reliability model has a lower error rate in predicting CBSE reliability as compared to reliability prediction utilizing fuzzy logic and PSO.

**INDEX TERMS** CBSE, CBS, CBSR, factors of CBSR metrics, reusability.

## I. INTRODUCTION

Software engineering consists of building, designing, testing, and validation of various software products. Repeating all the steps from beginning in making a new product is a very hard job that should be completed within the prescribed time period. As the technologies vary according to time, the concepts like component reusability, component interaction, and failure rate must be used to make a new product within time. Component-Based Software Engineering (CBSE) is a branch of software engineering that mainly depends on component dependency, component interaction and component reusability. In CBSE, the reliability relies on the capability of the reusable component with minimum change to produce new output with minimum faults which can satisfy customer needs [1]. Interaction of components and dependability are important in evaluating reusability in CBSE.

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Messina.

Component-Based Software (CBS) is a recent approach in the field of software engineering that focuses on aggregating components into complex software systems with the rapid development of component technology. This approach provides several advantages such as productivity, quality, reusability, reduces maintenance overheads and time-to-market. The reliability can be predicted by calculating the reliability of components individually and the interconnection methodology between components [2]. Reliability forecasting of CBS involves failure forecasting techniques that evaluate system reliability quantitatively.

There are various methods of reliability prediction such as architecture based models, Gokhale model [3], Laprie model [4], Shooman model [5], Yacoub model [1], Everett model [6], etc.

These models are based on state, path, and behavioral addiction. The common parameters used in these models were availability, errors in arithmetic algorithms, mean repair times, component reliabilities, transition probabilities, components dependencies, operation profile, transition probabilities, failure behavior of components and interfaces, constant failure rate, number of faults, execution of a set of components, series and parallel combination of components factors used, average execution time of component etc.

The reliability predicting models are related to the factors like effort, Kilo Delivered Lines of Code (KDLOC), fault density, reusability, availability, performance, serviceability, capability, maintainability, interface complexity, adaptability, fitness value and computational time, average execution time, reliability, probability, failure rate, fitness function, ants, etc.

## II. SOFT COMPUTING TECHNIQUES

Soft Computing techniques have become popular in the optimization of solutions for large problems. In soft computing, arbitrary numbers are produced to utilize either as beginning appraisals or during the learning and search process. Soft Computing techniques have many applications. There are several commonly used soft computing techniques like Genetic Algorithm (GA) [7], Neural Network (NN) [8], Fuzzy logic [9], Support Vector Machine (SVM) [10] and Swarm Optimization methods like Artificial Bee Colony (ABC) [11], Ant Colony Optimization (ACO) [12] and Particle Swarm Optimization (PSO) [13], etc. Soft computing techniques can be used in predicting software reliability. Soft computing includes factors like fitness value, actors, fitness function, target, etc. These techniques were compared with respect to the factors and parameters used for predicting software reliability. It was observed that PSO, ACO, ABC, and Fuzzy logic can be utilized to analyze the concepts of CBSE. Diwaker and Tomar [14] compared the performance of PSO, ABC, and ACO to check the integrity of the components and component interface. It was found that PSO and Fuzzy logic is suitable for a small problem and provided efficient results within time. PSO is selected for the assessment of the component-based software model because it provides the solution faster as compared to other techniques.

### A. PARTICLE SWARM OPTIMIZATION (PSO)

PSO include random movement of components in open area to achieve the target in less time with high speed. Every particle refreshes its data according to the overall traffic rate of particles [3].

The speed of different parts changes as per their past experience, looking through expertise, and data close-by. The fitness function assumes a significant job in PSO. The fitness function is picked according to the prerequisites. It is a troublesome methodology because of haphazardness. Figure 1 indicates the optimal solution in search of a target in various execution cycles

PSO includes multidimensional space which involves the position of each particle in that space using the following equations:

$$x_{id}^{t+1} = x_{id}^{t} + \upsilon_{id}^{t+1} \ldots \quad (1)$$

$$\upsilon_{id}^{t+1} = \omega\upsilon_{id}^{t} + c_1.\Psi_1.(p_{id}^{t} - x_{id}^{t}) + c_2..\Psi_2.(p_{gd}^{t} - x_{id}^{t}) \quad (2)$$

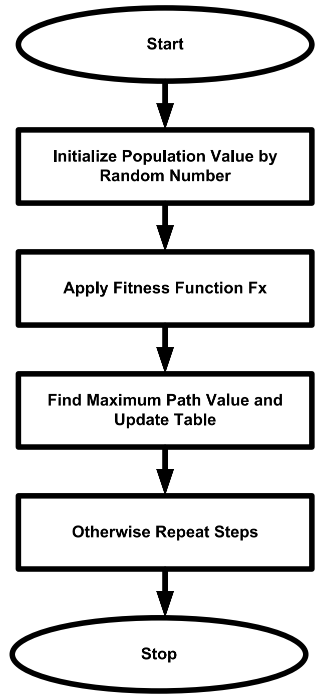



**FIGURE 1.** Working of PSO.

**TABLE 1.** Parameters used in PSO.

| Parameter used in PSO | Description |
|---|---|
| $\upsilon^{t}_{id}$ | A component in dimension d of the i[th] particle velocity in iteration t. |
| $x^{t}_{id}$ | A component in dimension d of i[th] the particle position in iteration t. |
| $c_1, c_2$ | Constant weight factors |
| $p_i$ | Best position achieved so long by particle i |
| $p_g$ | Best position found by the neighbors of particle i |
| $\Psi_1, \Psi_2$ | Random factors in the [0,1] interval |
| $\Omega$ | Inertia weight. |

The operation of the method depends on the way of the neighborhood's selection. In the essential calculation, either a worldwide ($g_{best}$) or nearby ($l_{best}$) neighborhood is utilized. In the worldwide neighborhood, every particle is viewed as when figuring $p_g$. On account of the nearby neighborhood, the area is just made out of a specific number of particles among the entire populace. The nearby neighborhood of a given molecule does not change during the emphasis of the calculation.

An imperative ($\upsilon_{max}$) is forced on $\upsilon^{t}_{id}$ to guarantee combination. The estimation of $\upsilon$max is generally kept inside the interim $[-x_{id}^{max}, x_{id}^{max}]$. $x_{id}^{max}$ is the most extreme incentive for a molecule position [6]. An enormous inactivity weight ($\omega$) favors worldwide pursuit, while a little idleness weight favors neighborhood search. At whatever point dormancy is used, now and then it diminishes straight at the time of the cycle of the calculation, beginning at an underlying value near 1 [6], [7]. An elective detailing of Eq. 1 as shown in

Eq. 2 adds a narrowing coefficient that replaces the speed requirement ($\upsilon_{max}$) [3]. The PSO calculation requires tuning of certain parameters: the individual and sociality loads (c1, c2) and the idleness factor ($\omega$). Both hypothetical and exact examinations are accessible to help in the choice of genuine qualities [1], [3]–[7].

### B. FUZZY LOGIC

Fuzzy logic is a condition-based approach that depends on the degree of truth rather than classified any problem in two cases such as true or false.

Fuzzy logic provides a mapping of unknown input statistics information to scalar statistics data [4]. It includes four parts: fuzzifier, Fuzzy Inference System (FIS), rules, and de-fuzzifier. The general architecture of a fuzzy system is shown in figure 2. Fuzzifier accepts crisp input and arranges that crisp set in a sequential manner, then a set of rules and computation intelligence is applied to evaluate results. De-fuzzifier optimizes the evaluated results and checks for the best solution on the basis of computational methods and rules applied. Fuzzy logic helps in solving the problem with dynamic nature.

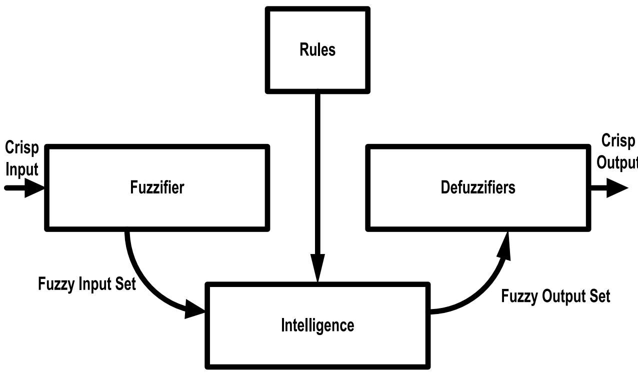



**FIGURE 2.** Working on fuzzy logic.

## III. RELATED WORK

Diwaker and Tomar [14] developed a simulation-relied framework that permits an inclusive fault injection study on hyper-visor with a broad range of arrangements. It was reported that many hardware errors can broadcast through different paths for an extended time before being observed. The issues in building error tolerance procedure s for the hyper-visor were also discussed.

Jaiswal and Giri [15] estimated CBS reliability using FIS and ANFIS with 2 dissimilar membership functions. It was observed that the Fuzzy Inference System (FIS) and Adaptive Neuro-Fuzzy Inference System (ANFIS) provide better results for 5 membership functions as compared to 3 membership functions. Four factors component dependency, operational profile, reusability, and application complexity were considered parameters. This work may be extended by considering fault density, maintainability, serviceability, software quality, performance, availability, usability, functionality, ability, capability and future research.

Tyagi and Sharma [16] proposed an ANFIS model for estimating Component-Based Software Reliability (CBSR) with different statistics sets. This hybrid approach required less calculative time. The output was calculated in the form of Root Mean Square Error (RMSE). ANFIS performed better than FIS. The model performed complex execution for big data sets.

Singh and Toora [17] developed a Neuro-Fuzzy-hybrid Algorithm (NFA) proposed for the component classification. The parameters used were volume, coupling, regularity, reuse frequency, and complexity. The performance of NFA was better than Fuzzy due to its adaptability and learning capability. *MATLAB was* used to implement for NFA. The results presented less percentage average error in NFA.

Lal and Kumar [18] spotlighted on the appraisal of frequently used soft computing techniques which assist in estimating and prediction of the reliability of various software system used in the medical system, mechanical engineering, computer engineering, and software engineering, etc. including both software and hardware. Different parameters have been considered to analyze soft computing to highlight future aspects to predict software reliability. It was observed PSO and fuzzy logic may be utilized where quick response and output with fewer percentage errors are required. A new model can be developed with factors such as component interaction, component dependency, complexity, failure rate and re-usability with utilization f concept of soft computing.

Lal and Kumar [18] utilized fuzzy logic to forecast CBS reliability. A range of rules was inputted to FIS for structuring and analysis of component-based software reliability. The simulation was done using MATLAB. The various steps in work were the recognition of components; analysis and designing of the reliability model for CBSS, and evaluate the reliability of the projected model with the current model. The outcome presents better results as compared to the conservative approach of guesstimate software reliability.

Tyagi and Sharma [19] introduced heuristic component dependency graphs (HCDGs) to guesstimate CBSS reliability including component reliability and CBSR. Estimation of reliability utilizing the ACO (ACOREL) algorithm was utilized to recognize the most utilized path. This path assists in guesstimate path reliability. The parameters considered were heuristic information, component-time, component path, probability, number of components, reliability of average execution, pheromone amount to guesstimate CBSR.

Diwaker and Tomar [21] proposed an approach to evaluate dynamic software performance including the effects of soft errors. A model was utilized that merged abstract calculating on a high level with calculating instructions on a low level. The outcomes of fault injection testing authenticate the dynamic program reliability model. The analysis of various dynamic software performances including the effects of soft errors was also presented.

Singhal *et al.* [22] present a model to survey the reusability utilizing fluffy rationale. The parameters considered were Modularity, Maintainability, Flexibility, Interface Complexity, and Adaptability. Different participation capacities, for example, Triangular, Trapezoidal and Gaussian enrolment were used. 243 fluffy sets were created and enrolment capacities were delegated Least, Less, Modder, More, Most.

Tyagi and Sharma [20] proposed a model that focused on 4 factors that highly affect CBSS reliability. The approach used fuzzy-logic estimating CBSS reliability. These factors were reusability, operation profile, complexity and component dependency. The value of these parameters was set as low, medium and high. $3^4$ (81) set of the combination were formed and the reliability was calculated using FIS. Other factors may also add to future work.

Diwaker and Tomar [21] presented a survey of architecture based reliability models with different parameters consideration in building a reliability model. Many factors for reliability prediction were identified and discussed such as reusability, component dependency, complexity, component interaction, failure rate, faults, and testing of failure which helped in computing the performance of CBRM and affects the reliability of the system. A new software reliability model can be built to predict reliability by considering significant factors.

Singhal *et al.* [22] discussed and compared the working principle and applications of PSO, GA, ACO, and BCO. The applicability of these optimization techniques for various problems was also discussed. These techniques may be integrated to make hybrid techniques that can be utilized for assessing the applicability of two or more than two optimization techniques for solving a given problem.

Toader [23] presented a scheduling mechanism named job shop scheduling using ACO and PSO that helped in solving confliction of resources clash, reduce make-span and total computation time. The job shop scheduling was evaluated using ACO, PSO and First Come First Serve (FCFS)and compared using two parameters i.e. fitness function and running time for different data sets. PSO presents a better outcome with respect to pheromone trail and pheromone evaporation rate parameters. In the future, Simulated Annealing (SA) or GA as hybrid techniques may be applied to analyze the performance of job scheduling tasks.

## IV. FACTORS AFFECTING CBS

The major factors involved in the CBSE metric are reusability, Component Dependency, Component Interaction, and complexity. The ranking and priority of these factors to be used with any program/system may be varied for a particular type of problem.

i) *Reusability:* Reusability consists of logic in a program, a loop, percentage of the line of code, a function of a number of classes that are used in making new software. The cost is calculated by equation [24]:

$$C_s = C_{nr} - C_r$$

where $C_s$ is saving cost; $C_{nr}$ is developing software with no reuse; $C_r$ is the cost of developing software with reuse. The saving cost $C_s$ can be calculated using a line of code, the function used repeatedly and the function those are not used repeatedly.

The relation between component reliability and reusability can be expressed as [15]:

$$\text{Component Reliability} \propto \text{Reusability}$$

The sub-parameters of Reusability are understandability, portability, variability, flexibility, and maintainability

ii) *Complexity:* Complexity depends on the number of statements in program code whether using RISC & CISC instructions, the time taken in executing an instruction, memory storage, and usage, type of platform used. More complexity results in low reliability.

Software expansion cost is inversely proportional to the complexity and volume of the software system. The relationship between reliability and complexity can be expressed as [19]:

$$\text{Reliability} \propto (1/\text{Complexity})$$

iii) *Component Interaction:* The interaction shows the interfaces connecting components. This helps in making components more reusable. Hence the overall reliability will be increased.

The following metrics are based on the interaction of components in a system: component average interaction density, component incoming interaction density, component packing density, component outgoing interaction density, component interaction density, etc. The relationship between reliability and component interaction can be expressed as [20]:

$$\text{Reliability} \propto \text{Component Interaction}$$

iv) *Component Dependency:* Component Dependency represents the dependability of components on other components. More dependability shows a more combinational view of components. More components dependability results in low reliability. The relationship between reliability and component dependency can be expressed as [15]:

$$\text{Reliability} \propto 1/\text{Component Dependency}$$

v) *Failure:* Failure rate is the rate at which a system or component fails. It includes MTTR, MTBF, MTTF, availability. More failure in the system results in low reliability. Software Reliability is given as [25]:

$$r(t) = e^{-\lambda t}$$

where $r(t)$ = continuous-time system reliability, and $\lambda$ is its failure rate.

Software Reliability = [1-probability of failure]

The relationship between reliability and failure can be expressed as [16], [20]:

$$\text{Reliability} \propto 1/\text{Failure}$$

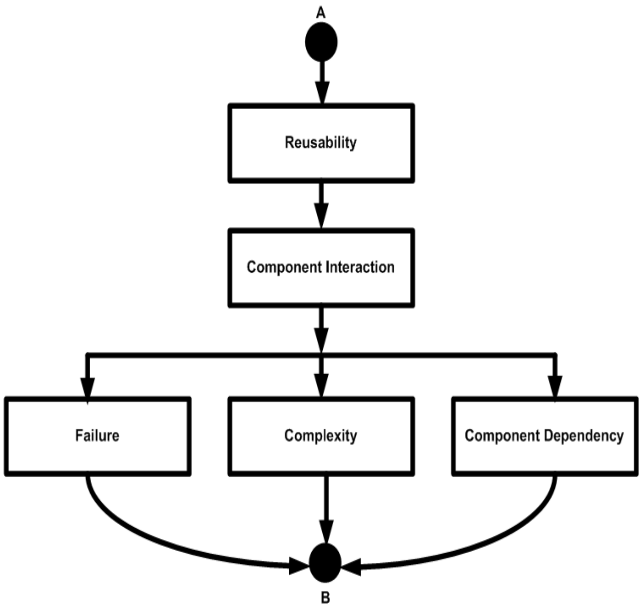



**FIGURE 3.** Component-based software reliability model.

## V. THE PROPOSED MODEL: COMPONENT-BASED SOFTWARE RELIABILITY MODEL

In the previous section, five different factors of CBSR have been discussed that helps in assessing the reliability of components. A new model is proposed with the integration of these factors for predicting reliability. The reliability can be estimated directly by the proposed model. The proposed approach uses fuzzy logic and PSO for the assessment of the proposed model. The model uses five factors of CBS i.e. component interaction, component dependency, complexity, reusability and failure rates. Fuzzy logic shows better outcomes than PSO. The mathematical equation is built by assessing the relationship between reliability and other factors [16],[20], [15], [26].

$$\text{Reliability} = \frac{(\text{Reusability} \times \text{Component Interaction})}{(\text{Component Dependency} \times \text{Complexity} \times \text{failure})} \quad (3)$$

Using series and parallel method of calculating the reliability of a system, the mathematical model can be expressed as shown in figure3.

The reliability is shown in Eq. 3 can be expressed as:

$$\text{Reliability} = \text{Reu}^{*}\text{Ci}^{*}(1-(1-\text{Com})(1-\text{Cd})(1-\text{f})) \quad (4)$$

In equation 4, Reu is the probability of occurrence of reusability, Ci is the probability of occurrence of Component Interaction, Com is the probability of occurrence of complexity, Cd is the probability of occurrence of Component Dependency, f is the probability of occurrence of failure rate. Table 2 shows a few combinations formed by applying Eq. 4 on different parameters. Equation 4 is formulated so that the two parameters reusability and component interaction effect directly and remaining factors affect less on overall reliability.

**TABLE 2.** Combination of various parameters for prediction of reliability using equation (4).

| Parameters /Factors | Reusability (reu) | Complexity (com) | Component Interaction (ci) | Component Dependency (cd) | Failure (f) | Reliability (R) |
|---|---|---|---|---|---|---|
| 1 | Hreu | Hcom | Hci | Lcd | Hf | L |
| 2 | Hreu | Hcom | Hci | Lcd | Mf | L |
| 3 | Hreu | Hcom | Hci | Lcd | Lf | M |
| 4 | Hreu | Hcom | Mci | Hcd | Hf | L |

In table 2, H, M, and L present high value, medium value, and low value. Hreu presents the high value of reusability, Hcom presents the high value of complexity, Hci presents the high value of component interaction, Lcd presents the low value of component dependency and Hf presents the high value of failure rate and vice versa. In table 1, few combinations are shown; similarity 243 cases will be formed to predict reliability.

*Component-Based Software Reliability Model:* Figure 4 shows the process of predicting reliability by using fuzzy and PSO techniques. A mathematical model has been proposed for predicting reliability. Two soft computing techniques have been used to predict reliability i.e. Fuzzy logic and PSO. Then the results of both techniques are compared.

### A. PREDICTING RELIABILITY USING FUZZY LOGIC

All 243 rules are created in FIS. Five factors are used as input using Mamdani-style inferences. The value of five factors has been set as low, medium and high. Therefore, the total rules framed are 243.

Steps involved in the Fuzzy algorithm used for creating rule base consist of the following steps:

- Assessment of soft computing techniques to estimate and predict CBSR.
- Identify the factors that affect CBS reliability and methods for estimating these factors.
- Create 243 rules for implementing fuzzy logic.
- Design FIS for rule base, based on identified factors.
- Fuzzify the inputs
- De-fuzzify the outputs
- Estimate the error percentage.

#### 1) CREATING RULE FOR PROPOSED MODEL

All possible pairs of inputs variables are considered, yielding a total of $3^5$ sets. Reliability for all 243 combinations was classified based on expert opinion as High, Medium and Low. These classifications are used to create 243 rules using FIS. Example of a few pairs is as follows:

If Reusability is high, component Interaction is high, Complexity is high, the Component dependency is high, failure is high, and then the reliability will be high

If Reusability is high, component Interaction is Low, Complexity is medium, the Component dependency is low, failure is high, and then the reliability will be medium

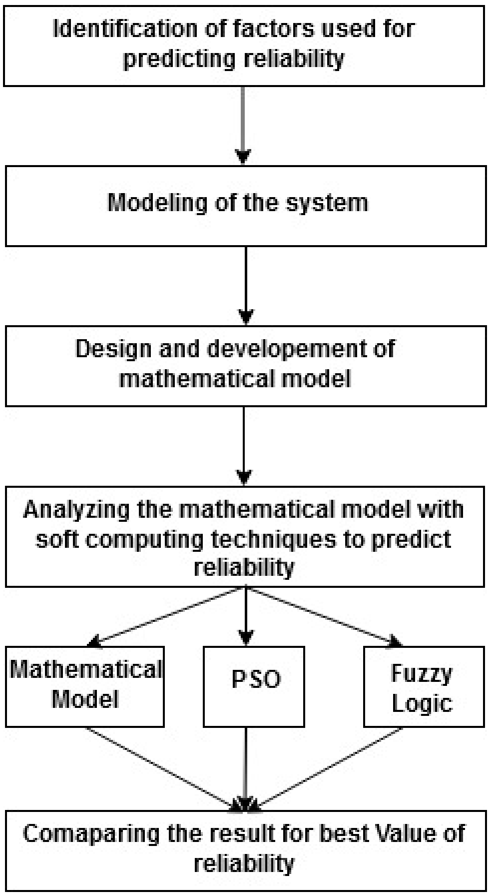



**FIGURE 4.** Flow chart for proposed model.

If Reusability is medium, component Interaction is high, Complexity is medium, the Component dependency is low, failure is low, and then the reliability will be low

If Reusability is low, component Interaction is medium, Complexity is high, the Component dependency is low, failure is low, and then the reliability will be low

All 243 rules were entered to create a rule base. Rules are depending on the particular set of inputs, using Mamdani-style FIS.

#### 2) MEMBERSHIP FUNCTIONS FOR INPUT PARAMETERS

Membership functions were defined for fuzzifying the Reusability, Component Interaction, Complexity, Component Dependency and failure rate as inputs. All these input parameters are divided into three stages: Low, Medium and High, as shown in table 2. The complete inference engine is given in Table 3.

#### 3) FUZZY INFERENCE SYSTEM (FIS)

After obtaining the fuzzified outputs shown in figure 5, using de-fuzzification, the results can be obtained in the form of a crisp value.

Table 3 shows various parameters included in FIS for calculating reliability through CBSRM.

**TABLE 3.** Parameters used in fuzzy inference system.

| | |
|---|---|
| System | Name = 'Reliability'<br>Type = mamdani<br>NumInputs =5<br>InLabels = Reusability, Component Interaction, Component Dependency, Complexity and Failure Rate<br>NumOutputs = 1<br>NumRules = 243<br>OutLabels = Reliability<br>AndMethod = min<br>OrMethod = max<br>ImpMethod = min<br>AggMethod = max<br>DefuzzMethod = centroid |
| Input1 | Name = 'Reusability'<br>Range = [01], NumMFs = 3<br>MF1 = 'Low':'trimf',[0.01 0.20 0.34]<br>MF2 = 'medium':'trimf',[0.35 0.53 0.68]<br>MF3 = 'high':'trimf',[0.69 0.82 0.99] |
| Input2 | Name = 'Component Interation'<br>Range = [01], NumMFs = 3<br>MF1 = 'low':'trimf',[0.01 0.20 0.34]<br>MF2 = 'medium':'trimf',[ 0.35 0.53 0.68]<br>MF3 = 'high':'trimf',[ 0.69 0.82 0.99] |
| Input3 | Name = 'Component Dependency'<br>Range = [01], NumMFs = 3<br>MF1 = 'low':'trimf', [0.01 0.20 0.34]<br>MF2 = 'medium':'trimf',[ 0.35 0.53 0.68]<br>MF3 = 'high':'trimf', [0.69 0.82 0.99] |
| Input4 | Name = 'Complexity'<br>Range = [01], NumMFs = 3<br>MF1 = 'low':'trimf', [0.01 0.20 0.34]<br>MF2 = 'medium':'trimf', [0.35 0.53 0.68]<br>MF3 = 'high':'trimf', [0.69 0.82 0.99] |
| Input5 | Name = 'Failaure Rate'<br>Range = [01], NumMFs = 3<br>MF1 = 'low':'trimf', [0.01 0.20 0.34]<br>MF2 = 'medium':'trimf', [0.35 0.53 0.68]<br>MF3 = 'high':'trimf', [0.69 0.82 0.99] |
| Output1 | Name = 'Reliability'<br>Range = [01], NumMFs = 5<br>MF1 = 'Low':'trimf', [0.01 0.20 0.34]<br>MF2 = 'Medium':'trimf',[0.35 0.53 0.68]<br>MF3 = 'High':'trimf', [0.69 0.82 0.99] |

**TABLE 4.** Value assigned to parameters considered for PSO simulation in matlab.

| Name of Parameters | Values |
|---|---|
| Maxitr | 4000 |
| p (Swarm Size) | 20 |
| W | 0.9 |
| c1 | 1.414 |
| c2 | 1.414 |
| itr= 100 | 100 |
| D (Design Variables) | 5 |
| Lb (Lower Bound) | 0 |
| Ub (Upper Bound) | 1 |
| Tolerance | 0.00000001 |

### *B. PREDICTING RELIABILITY USING PSO*

Table 4 shows the parameters considered in PSO for measuring reliability. These parameters are Maxitr (Maximum Iteration), p (Swarm Size), W (weight), c1(constant), c2

**TABLE 5.** Reliability measurement using equation (4) with fuzzy logic and PSO.

| S. NO. | Reusability | Component Interaction | Component Dependency | Complexity | Failure | Reliability Using Fuzzy Logic | Reliability Using PSO | Reliability Calculated using Formula | Error Rate using Fuzzy Logic | Error Rate using PSO | % Error Rate using Fuzzy Logic | % Error Rate using PSO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.8147 | 0.6557 | 0.4387 | 0.7513 | 0.3517 | 0.438 | 0.3913 | 0.485853992 | 0.047853992 | 0.094554 | 0.047853992 | 0.094553992 |
| 2 | 0.189 | 0.509 | 0.1155 | 0.4791 | 0.8819 | 0.16 | 0.1978 | 0.090966422 | -0.069033578 | -0.10683 | 0.069033578 | 0.106833578 |
| 3 | 0.6081 | 0.5141 | 0.5088 | 0.4448 | 0.6713 | 0.443 | 0.2872 | 0.28460021 | -0.15839979 | -0.0026 | 0.15839979 | 0.00259979 |
| 4 | 0.5058 | 0.8634 | 0.1371 | 0.7478 | 0.0704 | 0.158 | 0.0053 | 0.348360572 | 0.190360572 | 0.343061 | 0.190360572 | 0.343060572 |
| 5 | 0.7284 | 0.642 | 0.7584 | 0.1684 | 0.7327 | 0.796 | 0.1719 | 0.442518832 | -0.353481168 | 0.270619 | 0.353481168 | 0.270618832 |
| 6 | 0.6905 | 0.9525 | 0.1552 | 0.6118 | 0.1052 | 0.153 | 0.4634 | 0.464698241 | 0.311698241 | 0.001298 | 0.311698241 | 0.001298241 |
| 7 | 0.4677 | 0.8105 | 0.501 | 0.6931 | 0.9734 | 0.442 | 0.5414 | 0.377526665 | -0.064473335 | -0.16387 | 0.064473335 | 0.163873335 |
| 8 | 0.6237 | 0.0257 | 0.0207 | 0.5744 | 0.3994 | 0.153 | 0.3535 | 0.012016622 | -0.140983378 | -0.34148 | 0.140983378 | 0.341483378 |
| 9 | 0.4195 | 0.6464 | 0.0911 | 0.2161 | 0.5607 | 0.155 | 0.0081 | 0.186291462 | 0.031291462 | 0.178191 | 0.031291462 | 0.178191462 |
| 10 | 0.6936 | 0.1511 | 0.9969 | 0.8395 | 0.1492 | 0.795 | 0.428 | 0.104758595 | -0.690241405 | -0.32324 | 0.690241405 | 0.323241405 |
| 11 | 0.3239 | 0.4396 | 0.9453 | 0.7069 | 0.2121 | 0.439 | 0.0189 | 0.140587806 | -0.298412194 | 0.121688 | 0.298412194 | 0.121687806 |
| 12 | 0.3528 | 0.6885 | 0.4216 | 0.777 | 0.2239 | 0.436 | 0.2025 | 0.218587292 | -0.217412708 | 0.016087 | 0.217412708 | 0.016087292 |
| 13 | 0.9915 | 0.64 | 0.7555 | 0.9126 | 0.7708 | 0.795 | 0.9606 | 0.631452024 | -0.163547976 | -0.32915 | 0.163547976 | 0.329147976 |
| 14 | 0.2124 | 0.7348 | 0.0001 | 0.678 | 0.5295 | 0.15 | 0.0395 | 0.132428893 | -0.017571107 | 0.092929 | 0.017571107 | 0.092928893 |
| 15 | 0.2284 | 0.2356 | 0.5857 | 0.4664 | 0.6827 | 0.151 | 0.0305 | 0.050036429 | -0.100963571 | 0.019536 | 0.100963571 | 0.019536429 |
| 16 | 0.9464 | 0.7616 | 0.4357 | 0.3999 | 0.6089 | 0.443 | 0.7451 | 0.62531786 | 0.18231786 | -0.11978 | 0.18231786 | 0.11978214 |
| 17 | 0.4699 | 0.1837 | 0.2462 | 0.5463 | 0.3692 | 0.156 | 0.1845 | 0.067698421 | -0.088301579 | -0.1168 | 0.088301579 | 0.116801579 |
| 18 | 0.5995 | 0.6941 | 0.6739 | 0.7973 | 0.9306 | 0.781 | 0.0518 | 0.414204085 | -0.366795915 | 0.362404 | 0.366795915 | 0.362404085 |
| 19 | 0.9834 | 0.7493 | 0.2526 | 0.391 | 0.2142 | 0.152 | 0.9495 | 0.473308388 | 0.321308388 | -0.47619 | 0.321308388 | 0.476191612 |
| 20 | 0.8821 | 0.2123 | 0.4829 | 0.3154 | 0.2641 | 0.44 | 0.785 | 0.138483511 | -0.301516489 | -0.64652 | 0.301516489 | 0.646516489 |
| 21 | 0.7974 | 0.9072 | 0.1148 | 0.9989 | 0.4634 | 0.15 | 0.5651 | 0.723023304 | 0.573023304 | 0.157923 | 0.573023304 | 0.157923304 |
| 22 | 0.5394 | 0.6486 | 0.3594 | 0.976 | 0.5072 | 0.443 | 0.4954 | 0.347204163 | -0.095795837 | -0.1482 | 0.095795837 | 0.148195837 |
| 23 | 0.0745 | 0.1609 | 0.1486 | 0.9368 | 0.9091 | 0.157 | 0.0019 | 0.011928419 | -0.145071581 | 0.010028 | 0.145071581 | 0.010028419 |
| 24 | 0.4442 | 0.7012 | 0.7968 | 0.0631 | 0.8369 | 0.438 | 0.1551 | 0.301801595 | -0.136198405 | 0.146702 | 0.136198405 | 0.146701595 |
| 25 | 0.8466 | 0.1165 | 0.001 | 0.9603 | 0.907 | 0.15 | 0.7075 | 0.0982682 | - | - | 0.0517317 | 0.6092317 |

**TABLE 5.** ***(Continued.)*** **Reliability measurement using equation (4) with fuzzy logic and PSO.**

| S. NO. | Reusability | Component Interaction | Component Dependency | Complexity | Failure | Reliability Using Fuzzy Logic | Reliability Using PSO | Reliability Calculated using Formula | Error Rate using Fuzzy Logic | Error Rate using PSO | % Error Rate using Fuzzy Logic | % Error Rate using PSO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 8 | | | 46 | 0.051731754 | 0.60923 | 54 | 54 |
| 26 | 0.0975 | 0.8527 | 0.1324 | 0.3519 | 0.7239 | 0.161 | 0.0017 | 0.070231145 | -0.090768855 | 0.068531 | 0.090768855 | 0.068531145 |
| 27 | 0.0306 | 0.939 | 0.7205 | 0.331 | 0.9741 | 0.153 | 0.0049 | 0.028594246 | -0.124405754 | 0.023694 | 0.124405754 | 0.023694246 |
| 28 | 0.1075 | 0.442 | 0.5373 | 0.3191 | 0.5548 | 0.155 | 0.0066 | 0.040850482 | -0.114149518 | 0.03425 | 0.114149518 | 0.034250482 |
| 29 | 0.9606 | 0.512 | 0.6475 | 0.0674 | 0.8791 | 0.796 | 0.7365 | 0.472279603 | -0.323720397 | -0.26422 | 0.323720397 | 0.264220397 |
| 30 | 0.9463 | 0.7616 | 0.155 | 0.9089 | 0.1803 | 0.156 | 0.6839 | 0.675225709 | 0.519225709 | -0.00867 | 0.519225709 | 0.008674291 |
| 31 | 0.1949 | 0.5132 | 0.6588 | 0.2105 | 0.7359 | 0.158 | 0.0561 | 0.092906809 | -0.065093191 | 0.036807 | 0.065093191 | 0.036806809 |
| 32 | 0.1178 | 0.0077 | 0.4815 | 0.3358 | 0.8656 | 0.15 | 0.0042 | 0.000865076 | -0.149134924 | -0.00333 | 0.149134924 | 0.003334924 |
| 33 | 0.3109 | 0.7797 | 0.2933 | 0.2048 | 0.0767 | 0.151 | 0.0532 | 0.116631347 | -0.034368653 | 0.063431 | 0.034368653 | 0.063431347 |
| 34 | 0.9808 | 0.5503 | 0.0031 | 0.0635 | 0.5721 | 0.15 | 0.9167 | 0.324117918 | 0.174117918 | -0.59258 | 0.174117918 | 0.592582082 |
| 35 | 0.5539 | 0.19 | 0.7395 | 0.7173 | 0.3528 | 0.44 | 0.1879 | 0.100225006 | -0.339774994 | -0.08767 | 0.339774994 | 0.087674994 |
| 36 | 0.5604 | 0.2156 | 0.6414 | 0.2039 | 0.3223 | 0.441 | 0.2729 | 0.097446666 | -0.343553334 | -0.17545 | 0.343553334 | 0.175453334 |
| 37 | 0.2359 | 0.0618 | 0.3583 | 0.5866 | 0.3977 | 0.151 | 0.0532 | 0.012249286 | -0.138750714 | -0.04095 | 0.138750714 | 0.040950714 |
| 38 | 0.1307 | 0.6207 | 0.6642 | 0.3569 | 0.8847 | 0.153 | 0.1932 | 0.079105516 | -0.073894484 | -0.11409 | 0.073894484 | 0.114094484 |
| 39 | 0.7417 | 0.9589 | 0.5829 | 0.0436 | 0.1065 | 0.443 | 0.4156 | 0.457717328 | 0.014717328 | 0.042117 | 0.014717328 | 0.042117328 |
| 40 | 0.1052 | 0.747 | 0.6172 | 0.5426 | 0.4125 | 0.153 | 0.0424 | 0.070500661 | -0.082499339 | 0.028101 | 0.082499339 | 0.028100661 |
| 41 | 0.2607 | 0.1927 | 0.9911 | 0.5081 | 0.3912 | 0.15 | 0.1248 | 0.050102995 | -0.099897005 | -0.0747 | 0.099897005 | 0.074697005 |
| 42 | 0.882 | 0.2659 | 0.5928 | 0.2388 | 0.1513 | 0.445 | 0.7834 | 0.172829126 | -0.272170874 | -0.61057 | 0.272170874 | 0.610570874 |
| 43 | 0.8428 | 0.7447 | 0.2251 | 0.7696 | 0.9289 | 0.158 | 0.6956 | 0.619665999 | 0.461665999 | -0.07593 | 0.461665999 | 0.075934001 |
| 44 | 0.2128 | 0.0569 | 0.5891 | 0.0727 | 0.9679 | 0.15 | 0.0594 | 0.011960223 | -0.138039777 | -0.04744 | 0.138039777 | 0.047439777 |
| 45 | 0.723 | 0.4867 | 0.8943 | 0.8646 | 0.4737 | 0.796 | 0.4326 | 0.349233607 | -0.446766393 | -0.08337 | 0.446766393 | 0.083366393 |
| 46 | 0.7109 | 0.6716 | 0.1092 | 0.8855 | 0.9517 | 0.156 | 0.5038 | 0.47508836 | 0.31908836 | -0.02871 | 0.31908836 | 0.02871164 |
| 47 | 0.4516 | 0.9458 | 0.4151 | 0.0528 | 0.6443 | 0.44 | 0.3511 | 0.342952681 | -0.097047319 | -0.00815 | 0.097047319 | 0.008147319 |
| 48 | 0.0974 | 0.6658 | 0.0795 | 0.0147 | 0.7868 | 0.152 | 0.0079 | 0.052309362 | -0.099690638 | 0.044409 | 0.099690638 | 0.044409362 |
| 49 | 0.3538 | 0.604 | 0.3337 | 0.0482 | 0.5407 | 0.151 | 0.0849 | 0.151449877 | 0.000449877 | 0.06655 | 0.000449877 | 0.066549877 |
| 50 | 0.9887 | 0.3171 | 0.5773 | 0.2432 | 0.767 | 0.444 | 0.7374 | 0.2902285 | - | - | 0.1537714 | 0.4471714 |

**TABLE 5.** *(Continued.)* **Reliability measurement using equation (4) with fuzzy logic and PSO.**

| S. NO. | Reusability | Component Interaction | Component Dependency | Complexity | Failure | Reliability Using Fuzzy Logic | Reliability Using PSO | Reliability Calculated using Formula | Error Rate using Fuzzy Logic | Error Rate using PSO | % Error Rate using Fuzzy Logic | % Error Rate using PSO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 8 | | | 46 | 0.153771454 | 0.44717 | 54 | 54 |
| 51 | 0.0959 | 0.335 | 0.651 | 0.7522 | 0.1 | 0.157 | 0.0015 | 0.029625967 | -0.127374033 | 0.028126 | 0.127374033 | 0.028125967 |
| 52 | 0.7977 | 0.5856 | 0.4265 | 0.8726 | 0.9852 | 0.444 | 0.0608 | 0.466627988 | 0.022627988 | 0.405828 | 0.022627988 | 0.405827988 |
| 53 | 0.1515 | 0.6203 | 0.3747 | 0.0676 | 0.9778 | 0.153 | 0.0394 | 0.092759101 | -0.060240899 | 0.053359 | 0.060240899 | 0.053359101 |
| 54 | 0.4701 | 0.0078 | 0.2735 | 0.3835 | 0.3124 | 0.15 | 0.1389 | 0.002537532 | -0.147462468 | -0.13636 | 0.147462468 | 0.136362468 |
| 55 | 0.4852 | 0.1577 | 0.1055 | 0.7722 | 0.5262 | 0.158 | 0.0514 | 0.06912881 | -0.08887119 | 0.017729 | 0.08887119 | 0.01772881 |
| 56 | 0.1832 | 0.2626 | 0.0466 | 0.4608 | 0.7548 | 0.155 | 0.0039 | 0.042044229 | -0.112955771 | 0.038144 | 0.112955771 | 0.038144229 |
| 57 | 0.3291 | 0.2351 | 0.518 | 0.9622 | 0.3231 | 0.155 | 0.0579 | 0.0764172 | -0.0785828 | 0.018517 | 0.0785828 | 0.0185172 |
| 58 | 0.7429 | 0.0032 | 0.7707 | 0.6562 | 0.6402 | 0.795 | 0.3706 | 0.00230985 | -0.79269015 | -0.36829 | 0.79269015 | 0.36829015 |
| 59 | 0.7541 | 0.8895 | 0.1257 | 0.0366 | 0.0864 | 0.154 | 0.464 | 0.154595597 | 0.000595597 | -0.3094 | 0.000595597 | 0.309404403 |
| 60 | 0.0849 | 0.0152 | 0.85 | 0.3416 | 0.8686 | 0.152 | 0.0111 | 0.001273733 | -0.150726267 | -0.00983 | 0.150726267 | 0.009826267 |
| 61 | 0.8779 | 0.8867 | 0.2372 | 0.6339 | 0.4057 | 0.155 | 0.828 | 0.649241252 | 0.494241252 | -0.17876 | 0.494241252 | 0.178758748 |
| 62 | 0.0633 | 0.5593 | 0.1969 | 0.5969 | 0.9211 | 0.15 | 0.0226 | 0.0344994 | -0.1155006 | 0.011899 | 0.1155006 | 0.0118994 |
| 63 | 0.9009 | 0.1066 | 0.762 | 0.1085 | 0.0839 | 0.436 | 0.4053 | 0.077368921 | -0.358631079 | -0.32793 | 0.358631079 | 0.327931079 |
| 64 | 0.4891 | 0.368 | 0.0092 | 0.1123 | 0.6446 | 0.15 | 0.0994 | 0.123726806 | -0.026273194 | 0.024327 | 0.026273194 | 0.024326806 |
| 65 | 0.1919 | 0.032 | 0.9576 | 0.908 | 0.1276 | 0.154 | 0.1257 | 0.006119903 | -0.147880097 | -0.11958 | 0.147880097 | 0.119580097 |
| 66 | 0.3223 | 0.36 | 0.9222 | 0.3927 | 0.9466 | 0.439 | 0.2474 | 0.115735257 | -0.323264743 | -0.13166 | 0.323264743 | 0.131664743 |
| 67 | 0.3502 | 0.2277 | 0.9651 | 0.8163 | 0.1976 | 0.441 | 0.1097 | 0.079330331 | -0.361669669 | -0.03037 | 0.361669669 | 0.030369669 |
| 68 | 0.135 | 0.3788 | 0.2458 | 0.6863 | 0.9823 | 0.152 | 0.1097 | 0.05092385 | -0.10107615 | -0.05878 | 0.10107615 | 0.05877615 |
| 69 | 0.7629 | 0.9522 | 0.191 | 0.0779 | 0.7692 | 0.156 | 0.7096 | 0.601361943 | 0.445361943 | -0.10824 | 0.445361943 | 0.108238057 |
| 70 | 0.4314 | 0.2158 | 0.4634 | 0.3942 | 0.5823 | 0.152 | 0.2627 | 0.080455278 | -0.071544722 | -0.18224 | 0.071544722 | 0.182244722 |
| 71 | 0.6818 | 0.9818 | 0.8208 | 0.1527 | 0.7393 | 0.795 | 0.2364 | 0.642894267 | -0.152105733 | 0.406494 | 0.152105733 | 0.406494267 |
| 72 | 0.8675 | 0.0525 | 0.2342 | 0.6152 | 0.8041 | 0.153 | 0.6148 | 0.04291461 | -0.11008539 | -0.57189 | 0.11008539 | 0.57188539 |
| 73 | 0.7048 | 0.3881 | 0.3972 | 0.6182 | 0.9309 | 0.442 | 0.5598 | 0.269182805 | -0.172817195 | -0.29062 | 0.172817195 | 0.290617195 |
| 74 | 0.2318 | 0.7803 | 0.9705 | 0.9212 | 0.5675 | 0.153 | 0.1041 | 0.180691692 | 0.027691692 | 0.076592 | 0.027691692 | 0.076591692 |
| 75 | 0.1246 | 0.7492 | 0.485 | 0.9771 | 0.877 | 0.153 | 0.0101 | 0.0932152 | - | 0.0831 | 0.0597847 | 0.0831152 |

**TABLE 5.** *(Continued.)* **Reliability measurement using equation (4) with fuzzy logic and PSO.**

| S. NO. | Reusability | Component Interaction | Component Dependency | Complexity | Failure | Reliability Using Fuzzy Logic | Reliability Using PSO | Reliability Calculated using Formula | Error Rate using Fuzzy Logic | Error Rate using PSO | % Error Rate using Fuzzy Logic | % Error Rate using PSO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 3 | | | 36 | 0.059784764 | 15 | 64 | 36 |
| 76 | 0.7717 | 0.6402 | 0.5865 | 0.6365 | 0.5893 | 0.445 | 0.5789 | 0.46354452 | 0.01854452 | -0.11536 | 0.01854452 | 0.11535548 |
| 77 | 0.6845 | 0.6133 | 0.4895 | 0.0961 | 0.8711 | 0.443 | 0.2913 | 0.394834027 | -0.048165973 | 0.103534 | 0.048165973 | 0.103534027 |
| 78 | 0.2644 | 0.5731 | 0.7149 | 0.1174 | 0.1697 | 0.153 | 0.1073 | 0.119869307 | -0.033130693 | 0.012569 | 0.033130693 | 0.012569307 |
| 79 | 0.3229 | 0.1278 | 0.3173 | 0.8216 | 0.8641 | 0.44 | 0.2127 | 0.040583585 | -0.399416415 | -0.17212 | 0.399416415 | 0.172116415 |
| 80 | 0.4662 | 0.906 | 0.352 | 0.9463 | 0.2767 | 0.44 | 0.3935 | 0.411746344 | -0.028253656 | 0.018246 | 0.028253656 | 0.018246344 |
| 81 | 0.2584 | 0.6878 | 0.7014 | 0.0936 | 0.86 | 0.159 | 0.0522 | 0.170993221 | 0.011993221 | 0.118793 | 0.011993221 | 0.118793221 |
| 82 | 0.8022 | 0.0296 | 0.6127 | 0.0435 | 0.2556 | 0.443 | 0.7659 | 0.017197052 | -0.425802948 | -0.7487 | 0.425802948 | 0.748702948 |
| 83 | 0.8965 | 0.2402 | 0.2876 | 0.4741 | 0.5914 | 0.15 | 0.8243 | 0.182374629 | 0.032374629 | -0.64193 | 0.032374629 | 0.641925371 |
| 84 | 0.6199 | 0.912 | 0.3584 | 0.7367 | 0.4767 | 0.442 | 0.0936 | 0.515370391 | 0.073370391 | 0.42177 | 0.073370391 | 0.421770391 |
| 85 | 0.4252 | 0.6657 | 0.8748 | 0.996 | 0.9096 | 0.445 | 0.1233 | 0.283042825 | -0.161957175 | 0.159743 | 0.161957175 | 0.159742825 |
| 86 | 0.8012 | 0.7849 | 0.4393 | 0.683 | 0.7797 | 0.436 | 0.6106 | 0.604237824 | 0.168237824 | -0.00636 | 0.168237824 | 0.006362176 |
| 87 | 0.8296 | 0.5981 | 0.7258 | 0.4447 | 0.2208 | 0.795 | 0.0136 | 0.437314766 | -0.357685234 | 0.423715 | 0.357685234 | 0.423714766 |
| 88 | 0.0555 | 0.3625 | 0.8173 | 0.7177 | 0.6735 | 0.157 | 0.0431 | 0.019779958 | -0.137220042 | -0.02332 | 0.137220042 | 0.023320042 |
| 89 | 0.7569 | 0.485 | 0.2821 | 0.0378 | 0.1868 | 0.154 | 0.4861 | 0.160887831 | 0.006887831 | -0.32521 | 0.006887831 | 0.325212169 |
| 90 | 0.3746 | 0.1397 | 0.9072 | 0.1737 | 0.1798 | 0.436 | 0.2021 | 0.049040303 | -0.386959697 | -0.15306 | 0.386959697 | 0.153059697 |
| 91 | 0.9961 | 0.0688 | 0.8113 | 0.7567 | 0.6038 | 0.795 | 0.9657 | 0.067285101 | -0.727714899 | -0.89841 | 0.727714899 | 0.898414899 |
| 92 | 0.4671 | 0.8673 | 0.2048 | 0.5254 | 0.4835 | 0.159 | 0.2849 | 0.326147374 | 0.167147374 | 0.041247 | 0.167147374 | 0.041247374 |
| 93 | 0.2525 | 0.0981 | 0.4009 | 0.4585 | 0.0553 | 0.156 | 0.037 | 0.017178846 | -0.138821154 | -0.01982 | 0.138821154 | 0.019821154 |
| 94 | 0.7609 | 0.3628 | 0.3639 | 0.7552 | 0.7731 | 0.433 | 0.0686 | 0.266300892 | -0.166699108 | 0.197701 | 0.166699108 | 0.197700892 |
| 95 | 0.9569 | 0.6837 | 0.435 | 0.5225 | 0.973 | 0.442 | 0.0654 | 0.649466929 | 0.207466929 | 0.584067 | 0.207466929 | 0.584066929 |
| 96 | 0.341 | 0.1587 | 0.6916 | 0.3784 | 0.3969 | 0.436 | 0.6563 | 0.04785999 | -0.38814001 | -0.60844 | 0.38814001 | 0.60844001 |
| 97 | 0.2386 | 0.7311 | 0.2673 | 0.7397 | 0.0102 | 0.15 | 0.0563 | 0.14151021 | -0.00848979 | 0.08521 | 0.00848979 | 0.08521021 |
| 98 | 0.2716 | 0.7822 | 0.2465 | 0.6907 | 0.9941 | 0.15 | 0.0661 | 0.212153399 | 0.062153399 | 0.146053 | 0.062153399 | 0.146053399 |
| 99 | 0.8133 | 0.8061 | 0.5199 | 0.2541 | 0.8904 | 0.437 | 0.2012 | 0.629869781 | 0.192869781 | 0.42867 | 0.192869781 | 0.428669781 |
| 100 | 0.2192 | 0.3599 | 0.5834 | 0.61 | 0.8071 | 0.152 | 0.0315 | 0.076417567 | -0.075582433 | 0.044918 | 0.075582433 | 0.044917567 |

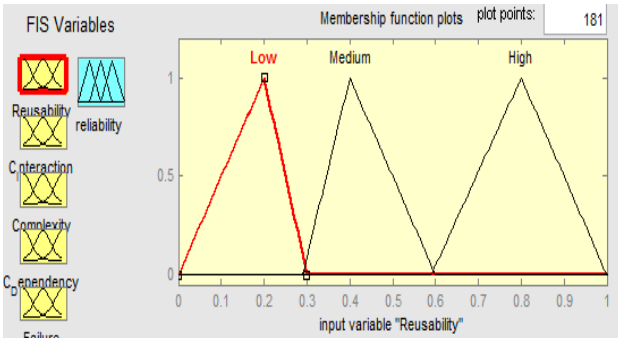



**FIGURE 5.** FIS for CBSRM.

(Constant), itr(iteration), D (Design variables), Lb (Lower Bound), Ub (Upper Bound), Tolerance.

## VI. RESULT AND ANALYSIS

The value of the parameters is selected randomly. Table 5 shows reliability measurement using Equation 4, Fuzzy and PSO. The error rate is calculated as

$$\text{Error rate (Fuzzy)} = R_{pm} - R_{fuzzy}$$
$$\text{Error rate (PSO)} = R_{pm} - R_{PSO}$$

$R_{pm}$ is the reliability calculated from the proposed model, $R_{fuzzy}$ is the reliability calculated form FIS, $R_{PSO}$ is the reliability calculated from PSO. The average value of percentage error is low in case of fuzzy as compared to PSO. The values are obtained from running the programs for 100 iterations.

Table 5 shows the estimation of reliability as calculated from three different methods i.e. directly from equation 1, fuzzy inference engine and using PSO. The value of all parameters will lie between 0 and 1 including all inputs and output. The simulation of PSO was done using MATLAB. Then, the output of the three methods is evaluated and compared. The output obtained from fuzzy logic is nearer to the output calculated from the direct equation.

The numerical value of all factors/parameters like Reusability, Component Interaction, Component Dependency, Complexity, and failure lies between 0 and 1. The low value lies between 0<Low<=0.34. Medium value lies between 0.35<=Medium<=0.68 and High value lies between 0.69<High<=1 for all factors. The resultant value of Reliability lies between $0<R\leq 1$.

*Evaluation of CBSRM:* MATLAB is used for PSO implementation using 5 factors of reliability i.e. Reusability, Complexity, Component Interaction, Component Dependency, and failure. Figure 6 and figure 7, shows the graph between Reliability calculated from the proposed mathematical model, Fuzzy logic, and PSO. X-axis presents a number of iterations for which their techniques for predicting reliability run and the Y-axis presents the best cost of three techniques at a particular iteration. The three techniques are reliability prediction using CBSRM, PSO and Fuzzy Logic. It is found that the Fuzzy inference engine provides better results as compared to PSO. The QWS data sets are used that are available on internet.

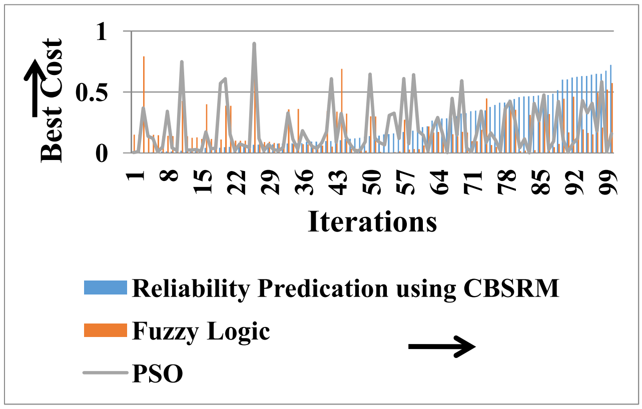



**FIGURE 6.** Comparison of reliability prediction using CBSRM, fuzzy logic and PSO techniques.

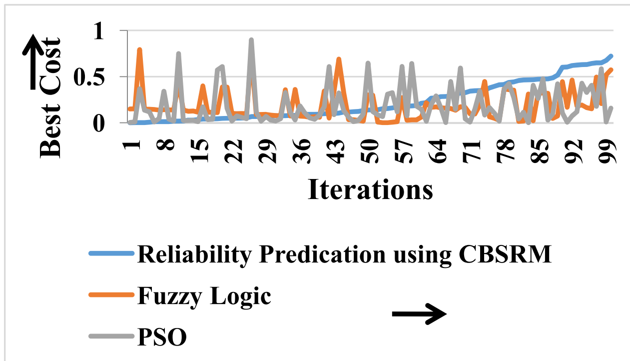



**FIGURE 7.** Comparison of reliability prediction using CBSRM, fuzzy Logic and PSO techniques.

In Figure 6 and Figure 7, the red line shows Fuzzy Logic variations and the green line shows PSO variations, and the blue line shows reliability prediction through CBSRM with respect to the best cost and iterations. PSO and FIS are used for 100 iterations only. The iteration may be increased. The parameters consideration may also be increased. An increase in parameters for reliability modeling results in an increase in complexity. The results show that FIS presents a better result for proposed CBSRM and show less error as compared to PSO.

## VII. CONCLUSION

It is difficult to make new product/software from a new stage. Identification and parameter consideration is also a hard task. Many reliability models have been made using different considerations of factors. In the present day, soft computing becomes popular in the field of estimating and predicting software reliability. In this paper, a new mathematical model is proposed to guesstimate the reliability of Component-based Software (CBS) using series and parallel reliability models approach. The output of the proposed model is compared with the outputs of soft computing techniques PSO and Fuzzy logic to compare the best value of reliability. The result shows that Fuzzy logic is more compatible for predicting reliability as compared to PSO. It is observed that the proposed reliability model has a lower error rate in predicting CBSE reliability as compared to reliability prediction utilizing fuzzy logic and PSO. Other factors may be added to enhance the proposed model for future work. Adding more factors in the proposed model results in an increase in complexity due to the formation of a large combination of parameters.

## REFERENCES


[1] S. Yacoub, B. Cukic, and H. H. Ammar, "A scenario-based reliability analysis approach for component-based software," *IEEE Trans. Rel.*, vol. 53, no. 4, pp. 465–480, Dec. 2004.

[2] K. Goseva-Popstojanova, A. P. Mathur, and K. S. Trivedi, "Comparison of architecture-based software reliability models," in *Proc. 12th IEEE Int. Symp. Softw. Rel. (ISSRE)*, Nov. 2001, pp. 22–31.

[3] S. S. Gokhale and K. S. Trivedi, "Analytical models for architecture-based software reliability prediction: A unification framework," *IEEE Trans. Rel.*, vol. 55, no. 4, pp. 578–590, Nov. 2006.

[4] A. Costes, C. Landrault, and J.-C. Laprie, "Reliability and availability models for maintained systems featuring hardware failures and design faults," *IEEE Trans. Comput.*, vol. C-27, no. 6, pp. 548–560, Jun. 1978.

[5] M. L. Shooman, "Structural models for software reliability prediction," in *Proc. 2nd IEEE Int. Conf. Softw. Eng.*, Oct. 1976, pp. 268–280.

[6] W. W. Everett, "Software component reliability analysis," in *Proc. IEEE Symp Appl. Syst. Softw. Eng. Technol (ASSET)*, Mar. 1999, pp. 204–211.

[7] R. Tavakkoli-Moghaddam, J. Safari, and F. Sassani, "Reliability optimization of series-parallel systems with a choice of redundancy strategies using a genetic algorithm," *Rel. Eng. Syst. Saf.*, vol. 93, pp. 550–556, Apr. 2008.

[8] S. Moon, K. Shin, and D. Jeon, "Enhancing reliability of analog neural network processors," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 6, pp. 1455–1459, Feb. 2019.

[9] M. Ravichandra and A. V. Ramani, "Measuring software reliability using fuzzy logic," *Int. J. Soft Comput.*, vol. 9, no. 5, pp. 314–317, 2014.

[10] Y. Ma, Q. Zhang, D. Li, and Y. Tian, "LINEX support vector machine for large-scale classification," *IEEE Access*, vol. 7, pp. 70319–70331, 2019.

[11] S. Yang, M. Lu, and L. Ge, "Bayesian network based software reliability prediction by dynamic simulation," in *Proc. IEEE 7th Int. Conf Softw. Secur. Rel. (SERE)*, Jun. 2013, pp. 13–20.

[12] S. Katiyar, N. Ibraheem, and A. Q. Ansari, "Ant colony optimization: A tutorial review," *MR Int. J. Eng. Technol.*, vol. 7, pp. 35–41, 2015.

[13] C. Leboucher, H.-S. Shin, R. Chelouah, M. S. Le, P. Siarry, and M. Formoso, "An enhanced particle swarm optimization method integrated with evolutionary game theory," *IEEE Trans. Games*, vol. 10, no. 2, pp. 221–230, Jan. 2018.

[14] C. Diwaker and P. Tomar, "Evaluation of swarm optimization techniques using CBSE reusability metrics," *Int. J. Control Theory Appl.*, vol. 2, pp. 189–197, Jan. 2016.

[15] G. P. Jaiswal and R. N. Giri, "Software reliability estimation of component based software system using fuzzy logic," *Int. J. Comput. Appl.*, vol. 127, no. 7, pp. 16–20, 2015.

[16] K. Tyagi and A. Sharma, "An adaptive neuro fuzzy model for estimating the reliability of component-based software systems," *Appl. Comput. Inform.*, vol. 10, nos. 1–2, pp. 38–51, 2014.

[17] H. Singh and V. K. Toora, "Neuro fuzzy logic model for component based software engineering," *Int. J. Eng. Sci.*, vol. 1, pp. 303–314, Jul. 2011.

[18] R. Lal and N. Kumar, "Design and analysis of reliability for component-based software system by using soft computing approaches," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 4, no. 6, pp. 929–932, 2014.

[19] K. Tyagi and A. Sharma, "A heuristic model for estimating component-based software system reliability using ant colony optimization," *World Appl. Sci. J.*, vol. 31, no. 11, pp. 1983–1991, 2014.

[20] K. Tyagi and A. Sharma, "A rule-based approach for estimating the reliability of component-based systems," *Adv. Eng. Softw.*, vol. 54, pp. 24–29, Dec. 2012.

[21] C. Diwaker and P. Tomar, "Identification of factors and techniques to design and develop component based reliability model," *Int. J. Sci. Res. Comput. Sci. Eng.*, vol. 5, no. 3, pp. 107–114, 2017.

[22] S. Singhal, S. Goyal, S. Goyal, and D. Bhatt, "A comparative study of a class of nature inspired algorithms," in *Proc. 5th Nat. Conf. INDIACom*, 2011, pp. 1–8.

[23] F. A. Toader, "Production scheduling by using ACO and PSO techniques," in *Proc. IEEE Int. Conf. Develop. Appl. Syst. (DAS)*, May 2014, pp. 170–175.

[24] K. S. Jasmine and R. Vasantha, "Cost estimation model for reuse based software products," in *Proc. Int. Multi Conf. Eng. Comput. Sci.*, vol. 1, 2008, pp. 1–4.

[25] J. D. Musa, *Software Reliability Engineering: More Reliable Software, Faster and Cheaper*. New York, NY, USA: McGraw-Hill, 2004.

[26] G. P. Jaiswal and R. N. Giri, "A fuzzy inference model for reliability estimation of component based software system," *Int. J. Comput. Sci. Technol.*, vol. 3, no. 3, 2015, pp. 177–182.

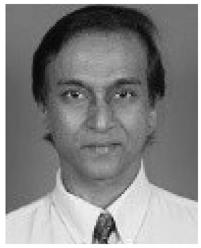

**CHANDER DIWAKER** received the master's and doctorate degrees in computer science and engineering from YMCAIE Faridabad and Gautam Buddha University, Greater Noida, India, in 2004 and 2018, respectively. He is currently an Assistant Professor with U.I.E.T., Kurukshetra University, Kurukshetra, India. He has been published many research articles in reputed journals and conferences. His main research interests include in the area of software reliability, computational intelligence, and computer networks.

**PRADEEP TOMAR** received the Ph.D. degree from MDU, Rohtak, India. Before joining Gautam Buddha University, he was a Software Engineer with multinational Company, Noida, and a Lecturer with M. D. University, Rohtak, Haryana, and Kurukshetra University, Kurukshetra, Haryana. He has been an Assistant Professor with the School of Information and Communication Technology, Gautam Buddha University, Greater Noida, India, since 2009. He has good teaching, research, and software development experience and vast administrative experience at university level on various posts, such as research coordinator, examination coordinator, admission coordinator, programme coordinator, timetable coordinators, proctor, and hostel warden. Dr. Tomar is also a member of Computer Society of India (CSI), Indian Society for Technical Education (ISTE), Indian Science Congress Association (ISCA), International Association of Computer Science and Information Technology (IACSIT), and International Association of Engineers (IAENG). He has qualified the National Eligibility Test (NET) for Lectureship in Computer Applications, in 2003, Microsoft Certified Professional (MCP), in 2008, SUN Certified JAVA Programmer (SCJP) for the JAVA platform, standard edition 5.0, in 2008, and qualified the IBM Certified Database Associate—DB2 nine Fundamentals, in 2010.

**ARUN SOLANKI** received the Master of Technology degree from YMCA University, Faridabad, Haryana, India, and the Ph.D. degree in computer science and engineering from Gautam Buddha University, in 2014. He has been with the Department of Computer Science, Gautam Buddha University, Greater Noida, India, since 2009. He has published many research articles in SCI/Scopus-indexed international journals. His research interests include expert systems, machine learning, and search engines.

**ANAND NAYYAR** received the Ph.D. degree in computer science from Desh Bhagat University, in 2017, with specialization in wireless sensor networks. He is currently with the Graduate School, Duy Tan University, Da Nang, Vietnam. He has published more than 250 research articles various national and international conferences and international journals (Scopus/SCI/SCIE/SSCI indexed). He has authored/coauthored/edited 25 books on computer science, and is associated with more than 400 international conferences as a Program Committee/Advisory Board/Review Board Member. His research interests include wireless sensor networks, MANETS, swarm intelligence, cloud computing, the Internet of Things, blockchain, machine learning, deep learning, cybersecurity, network simulation, and wireless communications. He is affiliated with more than 50 associations as a Senior Member and a Lifetime Member and also as a ACM Distinguished Speaker.

**NZ JHANJHI** received the Ph.D. degree in IT from University Technology Petronas, Malaysia. He has great international exposure in academia, research, administration, and academic quality accreditation. He was with King Faisal University for a decade, and he is currently with Taylor's University, Malaysia. He has 19 years of teaching and administrative experience internationally. He has an intensive background of academic quality accreditation in higher education besides scientific research activities, he had worked for academic accreditation for more than a decade and earned ABET accreditation twice for three programs at College of computer sciences and IT, King Faisal University, Saudi Arabia. He has edited/authored more than 11 research books with international reputed publishers, earned several research grants, and a great number of indexed research articles with reputed research journals and conferences. He has supervised several research students of postgraduate including master's and Ph.D. at KFU Saudi Arabia and Taylor's University, Malaysia. Dr. Jhanjhi is an Associate Editor of IEEE Access, and a member of the editorial board of several research journals around the globe.



**AZWEEN ABDULLAH** is currently an Associate Professor with Taylor's University. He is also a Professional Development Alumni of Stanford University and MIT and his work experience includes 30 years as an Academic in institutions of higher learning and as the Director of research and academic affairs at two institutions of higher learning, the Vice President for educational consultancy services, 15 years in commercial companies as a Software Engineer, a Systems Analyst and as a Computer Software Developer and IT/MIS consultancy and training.

**MAHADEVAN SUPRAMANIAM** received the DBA degree from the Twintech International University College of Technology, Malaysia, and the Master of Software Engineering degree from the University of Malaya. He serves as the Director of the Research and Innovation Management Centre and Institute of Graduate Studies of SEGi University and Colleges. His expertise lies in Research and Development and policies, Enterprise Resource Planning, Computer Science and Security System, Business Process Management, and Integrated Technologies for industries. He has shared most of his experiences on his expertise area through public talks and has written many articles and books, which has been published all over the world. He has authored/coauthored/edited 25 books on computer science and is associated with more than 400 international conferences as a Program Committee/Advisory Board/Review Board Member. His research interests include wireless sensor networks, MANETS, swarm intelligence, cloud computing, the Internet of Things, blockchain, machine learning, deep learning, cybersecurity, network simulation, and wireless communications.