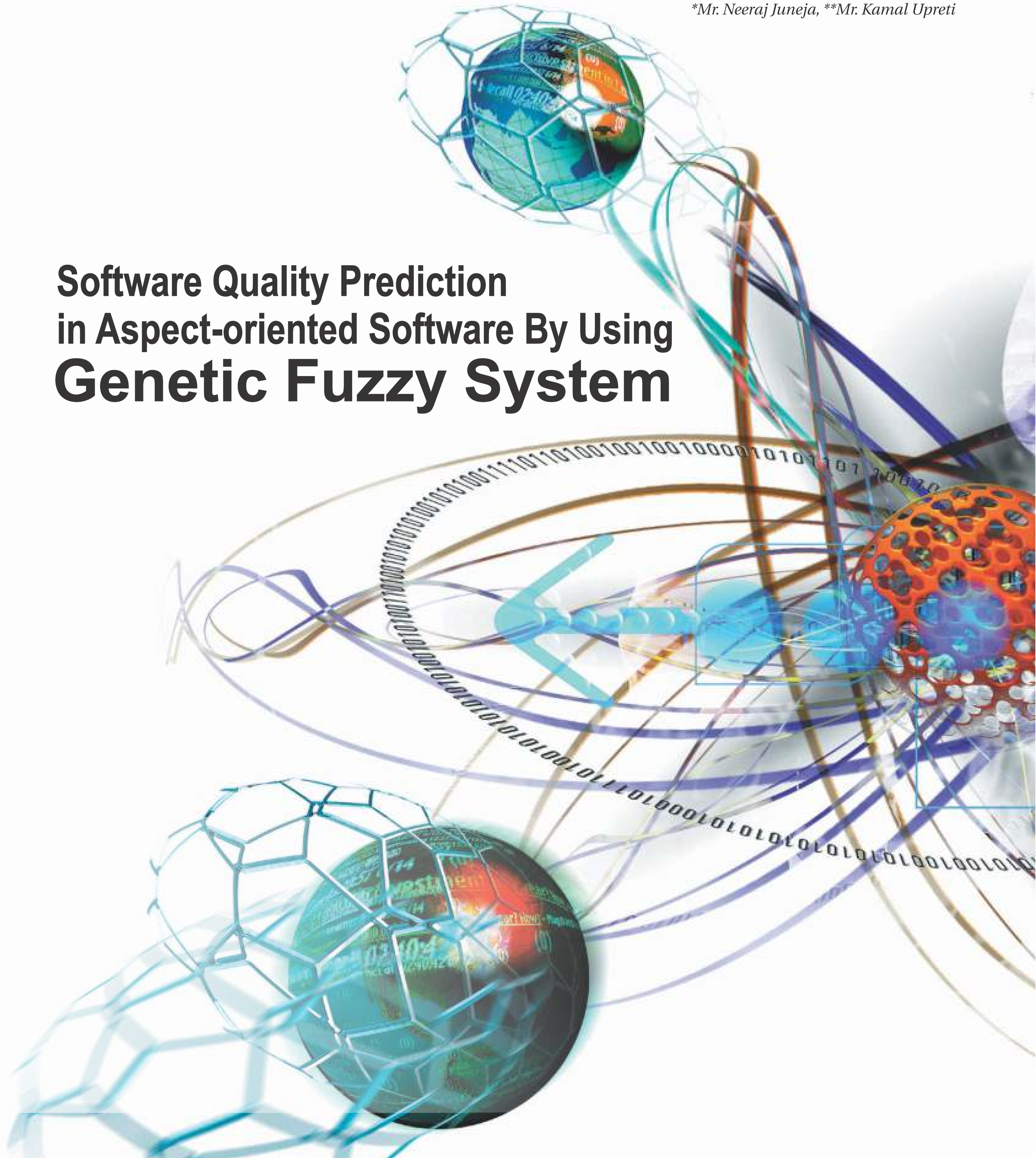
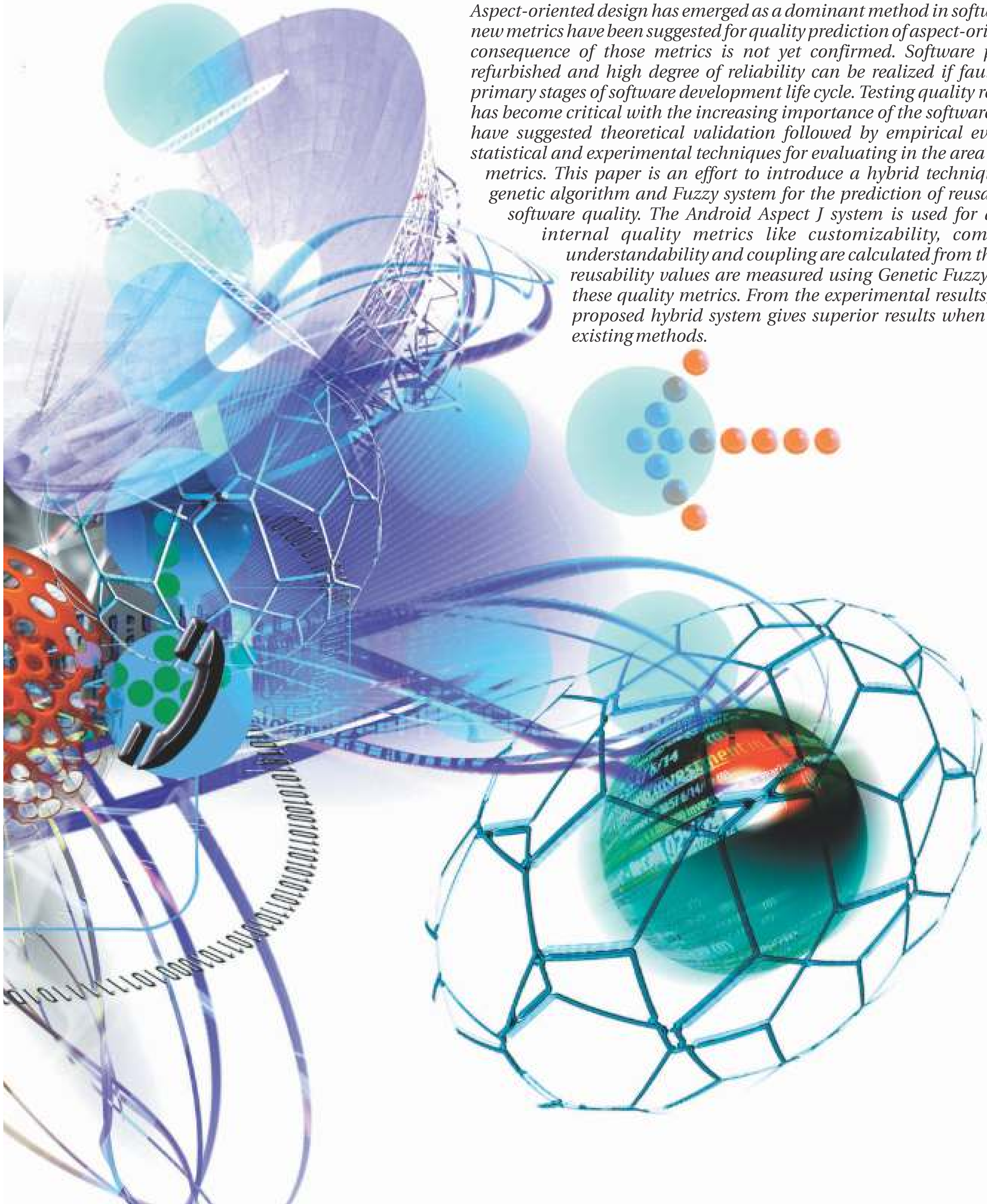
*Mr. Neeraj Juneja, **Mr. Kamal Upreti

Software Quality Prediction in Aspect-oriented Software By Using Genetic Fuzzy System



ABSTRACT

Aspect-oriented design has emerged as a dominant method in software industry and many new metrics have been suggested for quality prediction of aspect-oriented programs, but the consequence of those metrics is not yet confirmed. Software process control can be refurbished and high degree of reliability can be realized if faults are predicted in the primary stages of software development life cycle. Testing quality related issues of software has become critical with the increasing importance of the software quality. Many authors have suggested theoretical validation followed by empirical evaluation using proven statistical and experimental techniques for evaluating in the area of relevance of any new metrics. This paper is an effort to introduce a hybrid technique, the combination of genetic algorithm and Fuzzy system for the prediction of reusability parameter of the software quality. The Android Aspect J system is used for quality prediction. The internal quality metrics like customizability, commonality, portability, understandability and coupling are calculated from the Aspect J software. The reusability values are measured using Genetic Fuzzy System on the basis of these quality metrics. From the experimental results, it is observed that our proposed hybrid system gives superior results when compared to the other existing methods.



Keywords: *Green Advertising, Attitude toward Advertising, Brand Image, Intention to Purchase, Moderation.*

* Assistant Professor, Delhi Institute of Advanced Studies

** Assistant Professor, Delhi Institute of Advanced Studies

INTRODUCTION

With the prompt development of computers and their applications, software quality prediction at a primary stage in software industry is the need of the hour. Software quality prediction model was constituted for making critical decisions such as the distribution of resources in module verification and validation. Nevertheless, due to the complicated conditions of software development process in the early stage, the applicability and accuracy of these models is jeopardized [3].Although multiple predictive models are mentioned in the literature, still there is a scarcity of reliable tools which are applicable for real industrial systems. The main hindrance in building a reliable predictive tool for real industrial systems was the absence of representative samples. Since universal models do not exist, choosing an appropriate quality model was a difficulty [2].

Conventionally, a software quality metric is taken as a function that inputs software data and outputs a single value interpretable as the amount to which software possesses an attribute that affect quality [6].The following approaches are applicable for the prediction of software reliability (i) back propagation neural network (BPNN), (ii) threshold-accepting-based neural network (TANN) (Ravi and Zimmermann), (iii) Pi–Sigma network (PSN), (iv) multivariate adaptive regression splines (MARS), (v) generalized regression neural network (GRNN), (vi) multiple linear regression (MLR), (vii) dynamic evolving neuro–fuzzy inference system (DENFWAS) and (viii) Tree Net [4]. Aspect-oriented programming (AOP) is an embryonic paradigm that proposes a tool to encapsulate and implement crosscutting concerns with other modules. There is a notion that this approach improves the modularization of code, in turn reducing the complexity of object-oriented programs (OOP). Majority of the recommended frameworks for complexity measurement in the paradigm of AOP are applicable for Aspect J programming language [5].

The paper is segmented into five sections. Section II focuses on the related work in the similar domain of software quality prediction. Section III talks about the proposed methodology of Genetic Fuzzy System and how it is applied to find out the reusability metric value for a sample Aspect Oriented System. In section IV, results and comparative analysis of the experiment have been presented in terms of predictive accuracy of the approaches. Section V concludes the paper by summarizing the work carried out in finding the reusability values.

RELATEDWORKS

Software quality prediction was performed by various soft computing techniques. The techniques studied in the literature review were based on a single quality characteristic. Therefore they lack in their pursuit to predict the quality from a broader perspective keeping into consideration multiple quality characteristics. The following research articles were studied in the literature of software quality prediction and their contribution towards the same is presented as follows:

P. M. Shanthi and K. Duraiswamy [7] have presented an experimental validation of software quality metric suites on open source software for the prediction of fault-proneness in Object Oriented Systems. The three metrics used are Chidamber and Kemerer (CK) Metrics, Robert C. Martin Metric Suite and McCabe's Metric Suite. Using these metrics suites, the defects present in different versions of Rhino software were analyzed for the prediction of software quality.

Jagat Sesh Challa et al. [8] have presented a tool for accurately quantifying the software quality parametersas specified in WASO/IEC 9126 model. Given the volatile nature of the software quality attributes, the fuzzy multi-criteria approach was chosen for the evaluation of software quality.

Jagmohan Mago and Parwinder Kaur [9] have prominently focused on the design aspects, especially during the early phases of software development. By analyzing the metric data they have forecasted the quality of the object oriented system.

N. Rajasekhar Reddy and R.J. Ramasree [10] have recommended many SQA models to achieve required goals but most of them are software development design specific. Thus, they have concluded that there is a wide scope in research to develop generalized Software quality evaluation and assurance models those can generate Software quality evaluation test cases, which are specific to software development model selected.

Yajnaseni Dash and Sanjay Kumar Dubey [11] have elaborated the application of principal component analysis (PCA) to neural network modeling in the quest to increase the predictability of neural network. The purpose of principal component analysis is to supplement the performance of discriminant software quality models. On one hand it reduces the multidimensional problem of the huge data by converting it to a small orthogonal datasets, on the other side, it finds out the factorial relationship and most of the variance of the original dataset. The resultant orthogonal domain metrics provides a base for the easier prediction by neural networks. Thus principal component analysis has incredible importance in the evaluation of original data and predicting the software quality using neural network models.

Jagmohan Mago et al. [12] have proposed a fuzzy model to produce high quality Object Oriented applications. They have suggested a system centeredon fuzzy logic to judge the quality of OO design. It uses the popular CK metric suite and Mamdani Inference Engine. Here, a decision making system, based on fuzzy inference mechanism was presented by them. They designed a multi-faceted Fuzzy logic framework with six input metrics and each metric further being defined by three membership functions LOW MEDIUM HIGH and rule base consists of all the possible combinations i.e. 734 rules to evaluate the Quality of design of the software. The results proposed by this approach were compared with number of industrial software tools such as Analyst4j and ViZZ Analyzer and were found to be better than results produced by the software tools.

Anand Handa and Ganesh Wayal [13] have suggested a new approach for estimating of software projects development time. The highlighting feature of their work is the usage of a two-sided Gaussian membership function in fuzzy technique for software development time estimation and eventually it is authenticated with gathered data. They have obtained the advantages of fuzzy logic and good generalization. The results produced by the approach indicated that Two-sided Gaussian membership function is much better than other mentioned models. In order to achieve more accurate estimation, voting the estimated values of several techniques and combine their results maybe be useful.

Yogesh Singh and Anju Saha [14] have presented the usage of object oriented design metrics by applying neural networks for software testability prediction. The testability was measured in connection with the effort that is required for testing. The role of independent variables was played by the design metrics in the domain of object oriented systems and dependent ones are taken as two J Unit based test metrics. The software metrics that were used comprise of measures concerned to inheritance, encapsulation, cohesion, size, coupling, and polymorphism. A comparison was made to analyze the prediction performance of neural networks against the two types of statistical analysis methods: least squares regression and robust regression. The results produced by this study specify that the prediction model using neural networks is better than that of the regression models based on statistical techniques.

Gagan Tiwari and Arun Sharma [15] have conducted a detailed survey on these techniques and identified several factors or characteristics determining the maintainability of software. There is a significant variation in these factors for different software development approaches because of the architectural difference. Therefore there is a strong need to analyze the maintainability in depth to identify the characteristics and sub-characteristics which are putting an impact on maintainability aspect of software.

In the techniques discussed above, the software quality prediction was majorly carried out for object oriented software through Artificial Neural Networks (ANN) and Fuzzy Logic. Very few researchers have taken Aspect Oriented Systems into consideration while predicting Software Quality. Moreover, it was observed that the quality prediction frameworks proposed in the past were centered on a single quality characteristic and they lack in their capacity to predict the quality from a broader perspective keeping into consideration multiple quality characteristics. The real potential of applying soft computing techniques to predict the quality of the software has not been discovered to its fullest.

This paper attempts to find out the fuzzy rules for reusability based on a Fuzzy Inference System (FIS). The number of rules is then optimized using Genetic Algorithms (GA). Finally, Defuzzification is carried out to predict the reusability based on internal metric values. The results thus obtained are compared with Fuzzy and Neural Network techniques and our proposed model is found to be more accurate in terms of predictive accuracy of reusability.

PROPOSED MODEL USING GENETIC FUZZY SYSTEM

Many effectual techniques have been used for predicting the software quality. Also, there is an increase in demand for software quality prediction. Achieving high accuracy in prediction of software quality in object oriented programming using traditional methods involves some challenges like imprecise nature of the software quality metrics. Here we combined Genetic algorithm (GA) with Fuzzy Inference System (FIS) to resolve that problem. The quality of the Aspect oriented system can be predicted or calculated by using one of the quality metrics reusability. For that, the internal quality metrics are to be calculated in Aspect Oriented system such as customizability, portability, coupling, commonality and understandability. These internal metrics which are considered are calculated separately from the software projects written in Aspect J as per the given formulas below:

Internal Metrics

1. Customizability (CU):

Customizability refers to the ability to modify the component through its limited available information, like interfaces and parameters. For example in Java Beans, components can be customized through its customizable parameters

$$C = \frac{\text{No.of set methods}}{\text{Total Number of properties}}$$

Customizability values lies within a range of 0 and 1.

2. Understandability (U):

Understandability is the capability of the component to allow the user to understand the suitability of the component, and how it can be used for particular tasks and conditions of use.

$$U = \frac{\text{No.of observable properties}}{\text{Total Number of properties}}$$

Understandability values lies within a range of 0 and 1.

3. Portability (P):

Portability is defined as the capability of a component to be moved from one environment to another with little modification, if required.

$$P = \frac{\text{No.of customizable properties}}{\text{Total Number of properties}}$$

Portability values lies within a range of 0 and 1.

4. Interface Complexity (IC):

Interface Complexity play a major role while measuring the overall complexity of the component. To obtain better reusability the interface complexity is desired to be as low as possible.

$$IC(G) = e - n + 2p$$

e = Number of edges

n = Number of nodes

p = Number of connected components

Interface Complexity value should be less in order to improve reusability.

5. Coupling (C):

Coupling is the amount to which each program module depends on each one of the other modules.

$$C = 1 - \frac{1}{d_i + 2 * c_i + d_o + 2 * c_o + g_d + 2 * g_c + w + r}$$

d_i = Number of input data parameters

c_i = Number of input control parameters

d_o = Number of output data parameters

c_o = Number of output control parameters

g_d = Number of global variables used as data

g_c = Number of global variables used as control

w = Number of modules called (fan-out)

r = Number of modules calling the module under consideration (fan-in)

3.2 Fuzzy Inference System (FIS)

Based on the internal metrics calculated from Aspect J projects, the reusability value is calculated by using Genetic Fuzzy system. The five metric values are given as the input for Fuzzy system. Here the Genetic algorithm is used for rule optimization purpose. The inference system covers three major operations: Fuzzification, Rules Evaluation and Defuzzification. Fuzzy inference provides a mapping from a given input to an output by means of a fuzzy logic. Then, the mapping provides a basis from which decisions can be made, or patterns discerned. The process of fuzzy inference involves Membership Functions, Logical Operations, and If-Then Rules. The schematic diagram of the fuzzy inference system (FIS) is shown in Fig. 1.

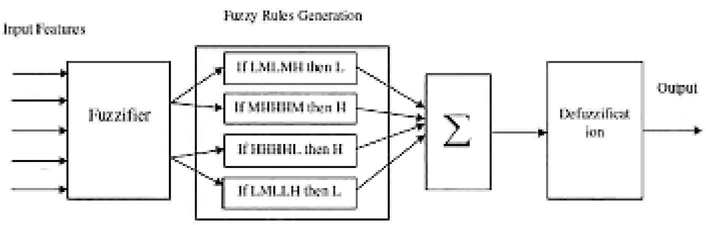


Figure 1: Fuzzy Inference System Structure

FUZZIFICATION

In fuzzification process, the crusty quantities are changed into fuzzy. In the method proposed in this paper, the fuzzification process is carried out by using the features that are extracted from Aspect J projects. The extracted features are *CU, U, P, IC,* and *C* for each feature the fuzzification process is performed. In the fuzzification process, all the *CU, U, P, IC* and *C* features values of training metrics are collected and lowest (min) and highest (max) values of each feature is computed by making use of the following equations:

$$ML^{(M)} = \min + \left(\frac{\max - \min}{3}\right) \quad (1)$$

$$XL^{(M)} = ML + \left(\frac{\max - \min}{3}\right) \quad (2)$$

In above equations (1) & (2), $ML^{(M)}$ and $XL^{(M)}$ are the minimum and maximum limit values of the feature *M*. Using the above equations, three conditions are formed for creating the fuzzy values.

- All the mean values of training metrics are compared with (1). If any value is found to be less than $ML^{(M)}$ value, then it is set as L.
- All the mean values of training metrics are compared with (2). If any value is found to be less than $XL^{(M)}$ value, then it is set as H.
- If any value is greater than $ML^{(M)}$ and less than $XL^{(M)}$, then this value is set as *M*.

FUZZY RULES GENERATION

The reusability of a system is a measure of five metric calculated from Aspect J. These combined features are useful in calculating the reusability components. The proposed GFS model considered all five metric values as input and calculates the crisp value of reusability using Rule Base. All inputs can be classified as Very Low, Low, medium, high and very high. The rule base is designed after considering different combinations of inputs. Each rule can be mapped to one of the five inputs based on the opinions given by domain experts. Following step is to generate fuzzy rules based on the fuzzy values of each feature. The generated fuzzy rules will train the FIS with the output values between L and H.

If CU is Low, IC is High, U is low, P is Low and Cohesion is Low then R is very Low

Rule Optimization using Genetic Algorithm:

The rule base is designed by taking into consideration different combinations of inputs, thereby, obtaining huge set of rules.

The set of rules were then optimized for calculation using GA. This algorithm encodes a possible solution to a specific problem on a simple chromosome-like data structure and applies recombination operators to these structures in a way to preserve critical information. The proposed method is designed to use genetic algorithm steps such as selection, mutation and crossover. Selection involves the selection of chromosomes based on some factors, in mutation the chromosomes are mutated and the crossover is done for further processing.

Chromosome generation:

The initial stage of GA is the generation of chromosome for the rule optimization problem. Here, 'N' numbers of random chromosomes are generated from the rules. The generated chromosomes are considered as the initial chromosome, which is represented as,

$$X_i = [x_0^{(i)} x_1^{(i)} \ldots\ldots\ldots x_{N_L-1}^{(i)}];0 \le i \le N_P - 1, 0 \le j \le N_L - 1 \quad (1)$$

where, $x_j^{(i)}$ is the j^{th} gene of the chromosome,

N_p is the population pool,

N_L is the length of the chromosome.

The length of the chromosomes

$$N_L = \frac{\left(\frac{x_{i(max)} - x_{i(min)}}{\sigma} + 1\right)}{2}$$

where, $x_{i(max)}$ is the maximum value of gene, $x_{i(min)}$ is the minimum value of gene, and σ is the number of rules in fuzzy system.

Fitness Function:

The following formula is used to calculate the fitness function.

$$F_i^1 = \sum_{i=0}^{N_L} ((C_i + W_i)/2)$$

$$F_j^2 = \sum_{i=0}^{N_L} ((F_i + T_i)/2)$$

$$F_i = F_i^1 + F_i^2$$

Here the fitness value of each chromosome is calculated dependent on factors calculated for each rules. After calculating the fitness values, select the solution and go to selection step otherwise, moves towards the next step 3.

Crossover in Genetic Algorithm:

The crossover operation is carried out between two parent chromosomes to get a new chromosome called offspring. Based on the crossover rate C_r, the genes are shortlisted and a new child chromosome is generated. After generating a new chromosome, the fitness function is applied to the newly generated child chromosome. The formula for computing the crossover rate is given

$$\text{Crossover Rate}(C_r) = \frac{\text{Number of Gene Crossovered}}{\text{Chromosome Length}}$$

The traditional method of single point cross over is employed in our proposed method. The performed crossover operation ensures that the architectures stay legal, as the supergenes stay intact during the crossover operation which means no responsibility can be dropped out of the system or can be duplicated. Here the crossover operation combines two subsets of responsibilities with their respective architectural structures by administering a one-point crossover.

Mutation in Genetic Algorithm:

The proposed technique is based on GA with Mutation so that the convergence of the solution is fastened. The mutation operation is carried out based on the mutation rate (M_r). Here, genes are mutated arbitrarily based on the given mutation rate. The formula for computing the mutation rate is given as,

$$\text{Mutationrate}(M_r) = \frac{M_P}{N_L}$$

where, M_p is the mutation point.

N_L is the chromosome length.

Flow Chart Representation of Genetic Algorithm:

The flow chart represents the process that is done in the genetic algorithm. The first step is the initialization of genes. The assignment of genes to the chromosomes is entirely based on our requirement. Then the fitness values are calculated for selecting the genes. Afterwards, crossover is done. There are types of crossover. After that the mutation is done. Then the offspring is produced and it is inserted in to the population. If the criteria is met then the process is stopped, otherwise the process is repeated.

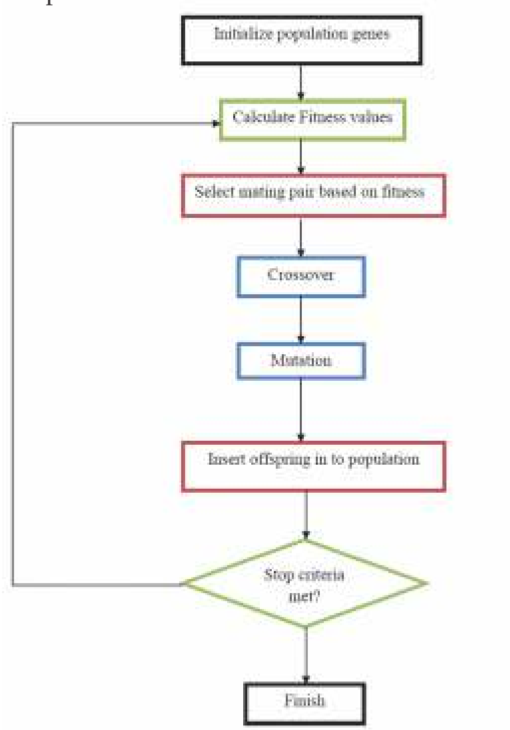


Fig. 2: Block diagram for GA

Defuzzification

In the Defuzzification process, the fuzzy set is used as an input and the output obtained is a single number which is either of the values LOW (L) or HIGH (H). The output value indicates quality of the Aspect oriented system. The FIS is trained with the fuzzy rules and the testing process is performed by the number of metric values.

RESULT AND COMPARATIVE ANALYSIS:

The quality of the aspect oriented systems can be calculated in terms of reusability by using Genetic Fuzzy system (GFS). The internal metrics such as customizability, portability, understandability, commonality and coupling are calculated from the Android Aspect J system. Based on the metric values, the reusability can be calculated using GFS and quality can be predicted. The gradual results gained from the proposed method are described below. The implementation is done in MATLAB 7.12.

These values give rise to rules generated in the FIS. In our proposed system, 5 linguistic variables are used such as very low, low, medium, high and very high. So, more than 243 rules were obtained. In the pursuit to find optimal rules, GA is used. The reusability value comparisons have been explained below:

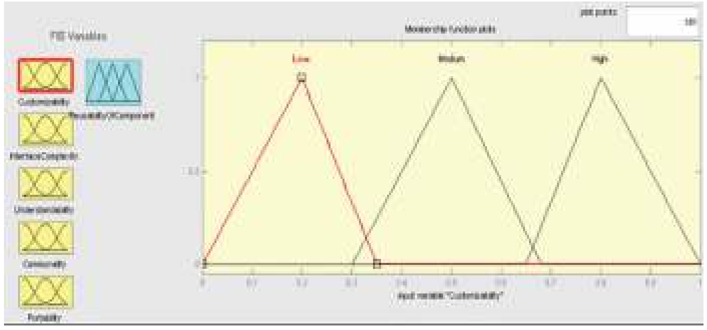


Fig. 4: Fuzzy membership functions

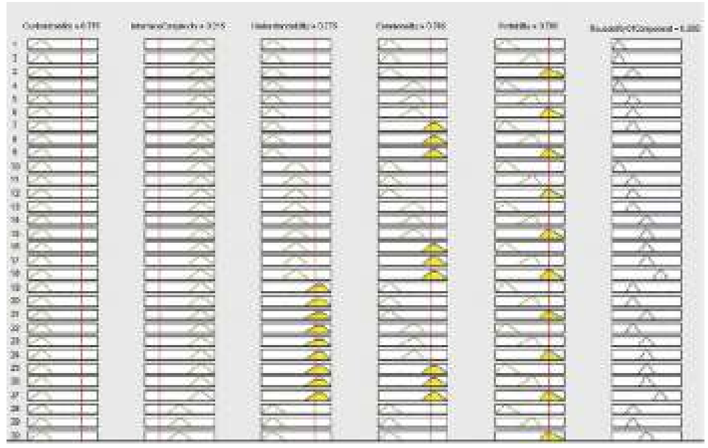


Fig. 5: Fuzzy Rules

Table 1: Metric values of Aspect J system

Aspect J Projects	Metrices Customizability	Portability	Understandability	Commonality	Coupling
Project 1	0.65	0.7	0.85	0.62	0.75
Project 2	0.61	0.6	0.72	0.45	0.8
Project 3	0.7	0.81	0.75	0.65	0.7

Fig. 3 and Fig.4 represents the internal metric vaules and Fuzzy system's membership functions.Based on these rules the reusability value is calculated. If the set of input [0.75, 0.412, 0.731, 0.543, 0.765] is given for GFS the output calculated as

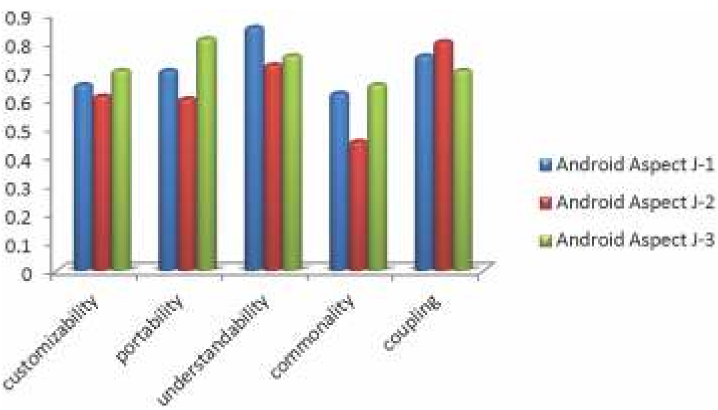


Fig. 3: Internal metric values

reusability metric is 0.874. The proposed method is compared with other soft computing techniques. The reusability metric value comparisons have been explained below:

COMPARISON

The model being proposed is an amalgamated model based on Genetic algorithm and Fuzzy system. The Genetic algorithm is used for rule optimization. The derived results are compared with quality prediction system with neural networks and Fuzzy system alone represented in following table and Graph.

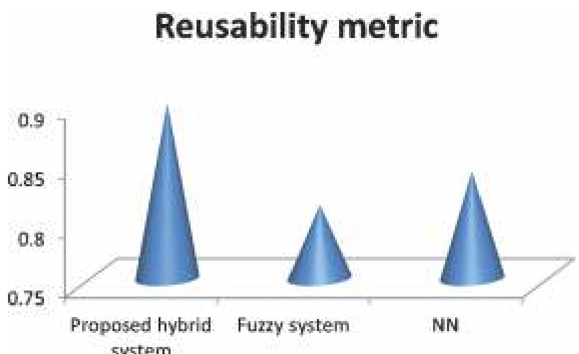


Fig. 6: Reusability Comparison

Table II: Metric Comparison

Methods	Reusability Metric Value
Proposed hybrid system	0.8971
Fuzzy system	0.812
NN	0.84

When comparing with other methods, the proposed hybrid system gives better results in the prediction of reusability metric value as indicated in the above table. The input metrics of 3 Android Aspect J projects are taken for consideration and 5 linguistic membership functions are decided. All the 3^5= 243 possible rules were supplied to Genetic Algorithm (GA) module to reduce the set of rules from 243 to 30 only. The reusability value is computed from these optimal set of rules using Fuzzy Inference System. When using fuzzy system alone, predicting the reusability is very difficult because of large number of rules. When using neural network, large combination of input values affect the performance. Our proposed method has succeeded in overcoming these difficulties by merging more than one soft computing technique to predict the quality of the software.

CONCLUSION

This paper aims to propose an amalgamated framework of genetic algorithm and Fuzzy system to predict the reusability of the software. The Android Aspect J system is used for quality prediction. The internal quality metrics such as customizability, commonality, portability, understandability and coupling are calculated from the Aspect J software in different threshold values. Providing the quality metrics as input, the reusability values are measured using GFS. The set of optimized rules are obtained from GA for Fuzzy inference system. From the experimental results, it is observed that our suggested system is better positioned in comparison to the other conventional approaches like Fuzzy and Neural Network in terms of its predictive accuracy.

FUTURE SCOPE AND DISCUSSION

Quality is a multifaceted concept and depends on several quality parameters. This paper is predicting the quality only on the basis of reusability and other factors like maintainability and reliability have been overlooked. The future scope of the presented attempt will be to find out a single value of quality from the perspective of all the different quality parameters. In future, it is also desirable to take some industrial projects both in the domain of Aspect Oriented systems and object oriented systems so as to put forth the quality improvement that Aspect Orientation brings in the software.

REFERENCES

[1] *David Grosser and Houari A. Sahraoui and Petko Valtchev, "An analogy-based approach for predicting design stability of Java classes" in proceedings of IEEE metrics, pp. 252-262, 2003.*

[2] *Salah Bouktif, Danielle Azar, Doina Precup, Houari Sahraoui and Bal´azs K´egl, "Improving Rule Set Based Software Quality Prediction: A Genetic Algorithm-based Approach, Journal of Object Technology, Vol. 3, No. 4, pp. 227–241, April 2004.*

[3] *YAO Lan and YANG Bo, "An Approach to Early Prediction of Software Quality", Journal of Electronic Science and Technology of China, Vol.5, No.1, pp.23-28, March 2007.*

[4] *N. Raj Kiran and V. Ravi, "Software reliability prediction by soft computing techniques", The Journal of Systems and Software, April 2007.*

[5] *Rajesh Kumar, P.S. Grover, Avadhesh Kumar, "A Fuzzy Logic Approach to Measure Complexity of Generic Aspect Oriented Systems", Journal of Object Technology, vol. 9, no. 3, pp. 43-57,May-June 2010.*

[6] *Mokhtar Beldjehem, "A Unified Granular Fuzzy-Neuro Framework for Predicting and Understanding Software Quality", International Journal of Software Engineering and Its Applications Vol. 4, No. 4, pp. 17-36 , October 2010.*

[7] *P. M. Shanthi and K. Duraiswamy, "An Empirical Validation of Software Quality Metric Suites on Open Source Software for Fault-Proneness Prediction in Object Oriented Systems", European Journal of Scientific Research, Vol. 51, No. 2, pp.168-181, 2011.*

[8] *Jagat Sesh Challa, Arindam Paul, Yogesh Dada, Venkatesh Nerella, Praveen Ranjan Srivastava and Ajit Pratap Singh, "Integrated Software Quality Evaluation: A Fuzzy Multi-Criteria Approach", Journal of Information Processing Systems, Vol.7, No.3, pp.473-518, September 2011.*

[9] *Jagmohan Mago and Parwinder Kaur, "Analysis of Quality of the Design of the Object Oriented Software using Fuzzy Logic", International Journal of Computer Applications, pp. 21-25, 2012.*

[10] *N. Rajasekhar Reddy and R.J.Ramasree, "Software Quality Modeling and Current State of the Art", International Journal of Soft Computing and Engineering (IJSCE), Vol.-2, pp. 502-511, March 2012.*

[11] *Yajnaseni Dash and Sanjay Kumar Dubey, "Application of Principal Component Analysis in Software Quality Improvement", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, pp. 202-205, April 2012.*

[12] *Jagmohan Mago, Sarabjit Kaur and Kumar Saurabh, "Fuzzy Model to Analyze and Interpret Object Oriented Software Design", International Journal of Electrical, Electronics and Computer Engineering, pp 41-46, May 2012.*

[13] *Anand Handa And Ganesh Wayal, "Software Quality Enhancement Using Fuzzy Logic with Object Oriented Metrics in Design", International Journal of Computer Engineering and Technology, Vol. 3, pp. 169-179, January- June (2012).*

[14] *Yogesh Singh and Anju Saha, "Application of Artificial Neural Networks for Assessing the testability of Object Oriented Software", International Journal of Scientific & Engineering Research, Vol. 3, pp. 1-8, July-2012.*

[15] *Gagan Tiwari and Arun Sharma, "Maintainability Techniques for Software Development Approaches – A Systematic Survey", International Journal of Computer Applications, pp.28-31, November 2012.*

[16] *Ahmed, M.A. and H.A. Al-Jamimi, Machine Learning Approaches for Predicting Software Maintainability: A Fuzzy-based Transparent Model. IET Software, 2013.*

[17] *P. K. Singh, O. P. Sangwan, A. Pratap, A. P. Singh, "A Quantitative Evaluation of Reusability for Aspect Oriented Software using Multi-criteria Decision Making Approach", World Applied Sciences Journal, Volume 30, Issue 12, Pages 1966-76, 2014.*

[18] *Hossein Momeni, Shiva Zahedian, "Aspect-Oriented Software Maintainability Assessment Using Adaptive Neuro Fuzzy Inference System (ANFIS)", Journal of Mathematics and Computer Science, 2014, pp 243-252.*