*International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868*
*Foundation of Computer Science FCS, New York, USA*
*Volume 11 – No. 2, July 2016 – www.ijais.org*

# Defect Prediction Model for AOP-based Software Development using Hybrid Fuzzy C-Means with Genetic Algorithm and K-Nearest Neighbors Classifier

Pankaj Kumar
Department of CSE
MU, Chittorgarh, Rajasthan

S. K. Singh
School of Computing Science and Engineering
Galgotias University, Uttar Pradesh

## ABSTRACT
The number of defects has often been considered a vital indicator of quality of software. It is well known that we cannot go back and add quality. Software Quality and reliability are considered to be one of the most important concerns of software product. In this paper, we give a brief overview of an Aspect-Oriented Programming (AOP) and a model is proposed to predict defects. The model is empirically validated on the PROMISE Software Engineering Repository dataset with three different types of methods. One is Fuzzy C-Means Clustering (FCM) approach and another is K-Nearest Neighbors (KNN) classifier technique, have been performed in real data. Third is a hybrid approach (i.e. combination of fuzzy c-means and genetic algorithms) have been performed. The performance of data is evaluated in terms of Reliability, Accuracy, Mean Absolute Error (MAE), and Root Mean Square Error (RMSE).

## General Terms
AOP, AOSD

## Keywords
AOP, AOSD, Software Quality, Defect

## 1. INTRODUCTION
Software product has one of the most comprehensive components of our daily lives. Our goal is to produce high-quality software product that provides value to end users and meets their prospects. To achieve this goal, our software, like a novel with printed pages, must be as defect free as possible. End users expect it to be so, and if we do not provide software products that are defect free, our end users may not be our end users for long time.

Software Quality is presumed to be one of the most crucial concerns of software product. Despite the availability of various systematic approaches and rules in exercises that provide a framework within which to manage, the inherent complexity of large scale software product makes such products not only difficult to manage but also result in delivered software [1, 2, 3], not free from defect. Large scale software products are highly vulnerable to various types of defect and delivering perfect software product to the satisfaction of customer is simply a nightmare for developer.

AOP is an emerging technique that attempts to exhort this issue by focusing primarily on modularity and composition mechanism of localized properties, methods and contents that have impact across the application development. It is based on the concept of concerns [3-5] that cut across the software product. Poor modularization of concerns across the program is normally considered to be responsible for the defects remaining in the product even after delivery. Much of the research in the field of software development is, how to make software defect free. We can minimize the number of software defects by changing the way of designing and development process or by using automated tools. But the key question is how defect is correlated with crosscutting module through scattering and tangling and degrade the quality as it poses a significant challenge to develop reliable and robust software.

Mistakes that happen during creation of software product are errors. When these affect the next process, they are termed defects. When the defects reach the customer and cause operational problems, they are known as failure. The life cycle of a defect begins as error and ends as a failure, going through a series of value changes. Software engineers distinguish software defect from software failures.

Defect prediction is to discover defects as efficiently as possible even after they are introduced into AOP based software product. An insufficient amount of valuable research work in this field has been carried out previously. Regardless of this it is hard to know a reliable approach to identifying defects in components. Using complexity measures, the techniques construct models, which classify components as likely to contain defect or not. During architecting, designing, and implementing software that is relatively easy to test increase the efficiency of the testing process and make defects more discoverable.

Over the last five decades, there are number of natural and social sources are provided in large scale and complicated databases for fast access of ICT. The clustered data hold various valuable parameters to make it compatible in many areas. For example, chromosomes are stored in the sequence of DNA and RNA in biological system but web document is formed using XML, HTML and CSS etc. So, it is more or less impossible to examine the information by manually.

The main objective of this paper is to design a defect Prediction model using Fuzzy C-Means clustering approach and Genetic Algorithms (i.e. Fuzzy C-Means + Genetic Algorithm)). It is called as hybrid system. After that we compare the results of hybrid system with Fuzzy C-Means and K-Nearest Neighbors Classifier results. The results after classification of software defects defined in terms of certain reliable and efficient parameters like Accuracy, Reliability, Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) in order to compare all the approaches.

The paper is composed as follows. The second section provides an overview of the defect prediction model. The third section provides the background of research design. The fourth section provides the brief introduction about Fuzzy C-Means clustering approach. The fifth section provides an

overview about the K-Nearest Neighbors (KNN) classifier technique. The sixth section provides an overview about genetic algorithms. The sixth section discusses about evolution parameters like Accuracy, Reliability, Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). The eighth section describes results and analysis. The ninth section concludes summarizing the contribution of the paper and future work directions.

## 2. DEFECT PREDICTION MODEL

Predicting defects is the proactive process of determining many types of defects found in software's content, design and codes in producing high quality product. A software defect is a fault or deficiency or mistake or error or bug in a product that causes it to perform unexpectedly. From a user's perspective, a defect is anything that causes the product not to meet their expectations. Figure 1 shows the general model for defect prediction.

Quality if software is one of the main properties. We propose a defect prediction model which is based on actual software to maintain the quality of the software. The proposed defect prediction model is based on hybrid Fuzzy C-Means clustering technique with genetic algorithm. To define and validate the prediction model, we use the PC1 (pc1.arff) dataset [6] for software prediction model from NASA Metrics Data Program (MDP). It is a public datasets.

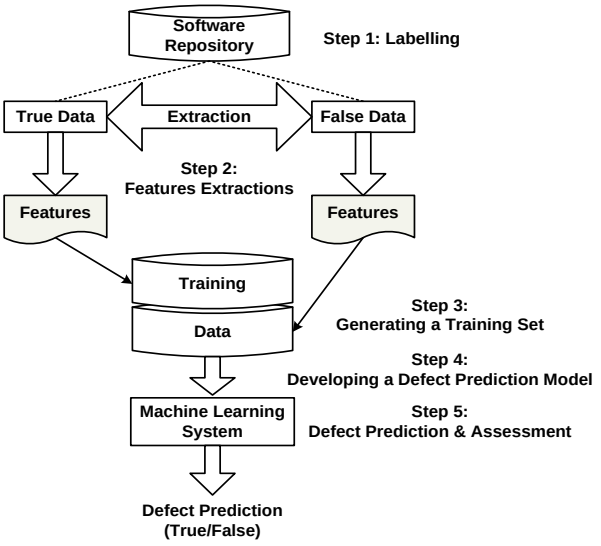


**Figure 1: Defect Prediction Model**

Fuzzy c-means clustering is used to predict any defects in the dataset using the membership grade technique. In the same manner K-Nearest Neighbors classifier is used to predict the defect in the database using the distance function in place of membership grade. Genetic algorithm and fuzzy c-mean clustering are used in hybrid method. First, clusters are divided by the FCM and then the clusters are optimized by the genetic algorithms in supervised manner.

## 3. RESEARCH DESIGN

The research design consists following three steps:

### 3.1 Find out the Required Qualitative and Quantitative Attributes of Software Systems

First of all, we would find out required qualitative and quantitative attributes from the PC1 (pc1.arff) dataset [6] for software prediction model which is taken from NASA Metrics Data Program. The database has 1109 modules and 320 requirements.

### 3.2 Select the Suitable Attributes Values as Representation of Statement

Secondly, the suitable attributes value is used as with defect and without defect. We set here as 0 (zero) for with defect and 1 (one) for without defect to create dataset in MATLAB R2009a. Each dataset contains 22 (5 different lines of code measure, 3 McCabe metrics, 4 base Halstead measures, 8 derived Halstead measures, a branch-count, and 1 goal field) attributes which describes the complexity, difficulty, volume, and some other characteristics. Following are the attributes:

1. loc numeric
2. v(g) numeric
3. ev(g) numeric
4. iv(G) numeric
5. N numeric
6. V numeric
7. L numeric
8. D numeric
9. I numeric
10. E numeric
11. B numeric
12. T numeric
13. lOCode numeric
14. lOComment numeric
15. locCodeAndComment numeric
16. lOBlank numeric
17. uniq_Op numeric
18. uniq_Opnd numeric
19. total_Op numeric
20. total_Opnd numeric
21. branchCount numeric
22. defects {false,true}

### 3.3 Analyze, Refine and Normalize the Attributes Values and Explore According to Need

After getting the attributes value, we go for analyze, refine and normalize the value of attributes using above discussed technologies.

## 4. FUZZY C-MEANS (FCM) CLUSTERING ALGORITHMS

Fuzzy C-Means (FCM) algorithm [7], one of the important and most popular fuzzy clustering techniques, was originally proposed by Dunn in 1973 and had been improved by Bezdek in 1981 is frequently used in pattern recognition. It is based on

minimization of the following objective function:

$$J_m(U,V) = \sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ij})^m ||x_k - v_i||^2 \quad 1 \le m < \infty \qquad (1)$$

Where $J_m(U,V)$ is the sum of squared error for the set of fuzzy clusters described by the membership grade of U, and the concerned set of cluster centers' V. $||*||$ is some inner product induced norm. In the formula, $||x_k - c_i||^2$ describes the distance between the data $x_k$ and the cluster centre $v_i$. The squared error is used as a performance level that measures the weighted sum of distances between cluster centres and elements in the corresponding fuzzy clusters. The number m governs the impact of membership grades in the performance. The partition becomes fuzzier with increasing m and it has been shown that the Fuzzy C-Means Clustering algorithm converges for any $m \in (1,\infty)$.

The FCM algorithm consists of the following steps:

1. Suppose that m-dimensional n data points represented by $X = (x_1, x_2, x_3, \dots\dots\dots\dots x_n)$ are to be clustered.
2. Assume the number of centers to be made, that is, $V = (v_1, v_2, v_3, \dots\dots\dots\dots v_c)$ where $2 \le c \le n$
3. Randomly select a cluster center c.
4. Compute the fuzzy membership grade $\mu_{ij}$ using equation 2.

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{d_{ij}}{d_{ik}}\right)^{\left(\frac{2}{m}-1\right)}} \qquad (2)$$

5. Compute the fuzzy center $v_j$ using equation 3.

$$v_j = \frac{\sum_{i=1}^{n}(\mu_{ij})^m x_i}{\sum_{i=1}^{n}(\mu_{ij})^m}, \qquad \forall\, j = 1,2,\dots\dots c \qquad (3)$$

6. Iterate step 4 and 5 until the minimum value of j is not achieved or $\left\|U^{(k+1)} - U^k\right\| < \beta$

Where k is the loop, $\beta$ is the termination criterion of membership grade between $[0,1]$, U is the fuzzy matrix which is defined as $(\mu_{ij})_{n*c}$ and j is the objective function.

## 5. K-NEAREST NEIGHBORS (KNN) CLASSIFIER ALGORITHMS

K-Nearest Neighbors (KNN) [8] is a simple and non-parametric lazy learning algorithm i.e. supervised machine learning system. It has been used in pattern recognition and statistical estimation in the beginning of 1970's.

The main objective of the KNN algorithm is to use a database in which the data sets are divided into different distinct classes to predict the classification of a new sample data set. It is often used when we do not know the conditional distribution of the response given by the predictors.

A data set is classified by a majority vote of its neighbors, with the data set being allocated to the class most its K-nearest neighbors calculated by a distance function. If K = 1, then the data point is simply allocated to the class of its nearest neighbor. Some distance functions are Euclidean, Manhattan, Minkowski and hamming for continuous variables.

The KNN algorithm consists of the following steps:

1. A positive integer k is specified, along with a new sample
2. We select the k entries in our database which are closest to the new sample
3. We find the most common classification of these entries
4. This is the classification we give to the new sample

## 6. GENETIC ALGORITHM

Genetic Algorithms (GA) [9] are adaptive heuristic search and optimization algorithms that resemble the principle of natural selection and genetics. It is very different from traditional search and optimization technique used in different manufacturing and optimization problems. Because of their clarity, ease of operation, least requirement, and global possibility, GAs has been proudly used in a wide variety of problem domains. GAs was developed by John Holland of the University of Michigan in 1965 which is inspired by Darwin's theory about evolution.

GAs is begun with a set of solutions (represented by chromosomes) called population. Solutions from one population are taken and used to form a new population. This is induced by an expectation, that the new population will be better solution than the old one. Solutions which are elected to form new solutions (offspring) are selected according to their fitness value - the more fitted they are the more chances they have to reproduce. This is repeated until some condition is satisfied. The flowchart of GAs is shown in the figure 2.

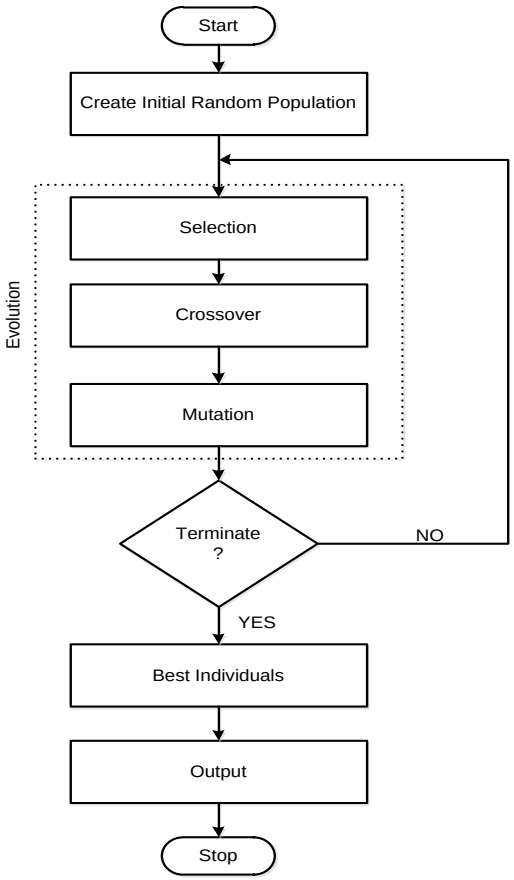


**Figure 2: Flowchart of GAs**

The Genetic algorithm consists of the following steps:

1. [Start] Create initial random population of n chromosomes for optimal solutions for the problem
2. [Fitness] Evaluate the fitness value using fitness function f(x) of each chromosome x in the population
3. [New population] Create a new population by iterating following steps until the new population is complete
4. [Selection] Select two parent chromosomes from a population according to their fitness value
5. [Crossover] With a cross over the parents to form a new offspring (children). Otherwise, offspring is an exact copy of parents.
6. [Mutation] With a mutation new offspring/child at each locus (position in chromosome).
7. [Accepting] Add new offspring/child in a new population
8. [Replace] Select new generated population for a further step of algorithm
9. [Test] If the end condition is satisfied, stop, and return the best solution in current population
10. [Loop] Go to step 2

# 7. PARAMETERS FOR EVALUATION

Following are the parameters, which we take for evaluation:

## 7.1 Accuracy

Accuracy is the degree of veracity and closeness of a quality of any measurement system. It is also called reproducibility or repeatability. In other way we can say that if system is more accurate than reliability is more and vice-versa.

## 7.2 Reliability

Reliability has both quantitative and qualitative aspects. It is associated with unpredicted defects/failures of system or services and understanding why these defects occur is key to make better reliability. The main reasons why defects occur include:

- The system is not fit for purpose or more especially the design is inherently unable.
- The system may be overstressed in some way.
- Defects can be caused by wear-out
- Defects might be caused by variation.
- Misuse of the item may cause defects.
- Wrong specifications may cause defects.

## 7.3 Mean Absolute Error (MAE)

The mean absolute error (MAE) is the average magnitude of the errors which is used to measure how intimate to predictions are to the possible outcomes without considering their direction. It is used for accuracy in continuous variables. The mean absolute error is defined as follows

$$MAE = \frac{|a_1 - c_1| + |a_2 - c_2| + \cdots + |a_n - c_n|}{n}$$

Where $a_i$ is the predicted value, $c_i$ is the true value or calculated value and n is the number of sample pair.

## 7.4 Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) is a frequently used measure of the differences between values predicted by a model or an estimator and the values actually observed.

It is just the square root of the mean square error as shown in equation given below:

$$RMSE = \sqrt{\frac{(a_1 - c_1)^2 + (a_2 - c_2)^2 + \cdots + (a_n - c_n)^2}{n}}$$

The MAE and the RMSE can be used together to diagnose the variation in the errors in a set of forecasts. The RMSE will always be larger or equal to the MAE; the greater difference between them, the greater the variance in the individual errors in the sample. If the RMSE=MAE, then all the errors are of the same magnitude. Both the MAE and RMSE can range from 0 to ∞.

# 8. RESULTS AND ANALYSIS

It is clear that what we really need to check, in order to assess the accuracy of a sequence of predictions, is the association between the predictions and the truth. We would say, informally, that a model was giving good results if what we observed tended to be in close agreement with what we had earlier predicted. The simulation of the prediction model is done by using MATLAB R2009a.

In this simulation, training and testing methods are being used, wherein a project is chosen for training the system. The PC1 (pc1.arff) dataset [6] is used for software prediction model from NASA MDP. It is a public datasets. One by one approach is applied. At last Fuzzy C-means Clustering and Genetic Algorithm based hybrid approach (GA-FCM) is applied on the same project and the final calculated values to classify the modules of project as defect prone or defect free.

The PC1 (pc1.arff) dataset having 1109 modules as we discussed earlier section, but in order to reduce complexity in calculations we take only 100 modules for training purpose. The original dataset contains 22 columns, shows the data entry values and the last column show the attribute value of dataset, i.e., the data is with defect or without defect. If we plot the data set with 100 modules and all considered columns (22 columns).

The software defect prediction model is implemented using Fuzzy C-means clustering, k-Nearest Neighbors Classifier and hybrid (Fuzzy c-means + Genetic Algorithm) approaches. Table 1 shows the performance comparison for all the algorithms.

**Table 1: Performance Comparison for all the algorithms**

| Parameters | FCM | KNN | GA-FCM |
|---|---|---|---|
| Accuracy | 80.24 | 89.31 | 99.24 |
| Reliability | 61.07 | 82.38 | 48.20 |
| Mean Absolute Error | 00.35 | 00.21 | 00.23 |
| Root Mean Squared Error | 00.18 | 00.22 | 00.01 |

Figure 3 represents the entire data (defected and not defected) using graph.

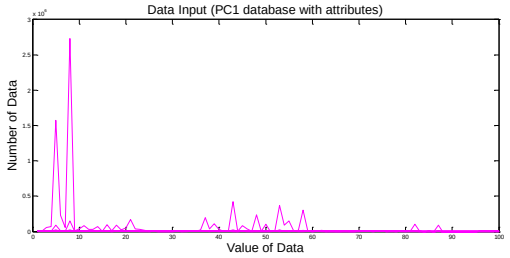


**Figure 3: Input PC1 dataset with all attributes**

Class Distribution of training dataset for defects:

Data with positive attribute (%): 22%

Data with negative attribute (%): 78%

Figure 4 show only the defected data. Where X-axis contains the value of data and Y-axis contains the number of data.

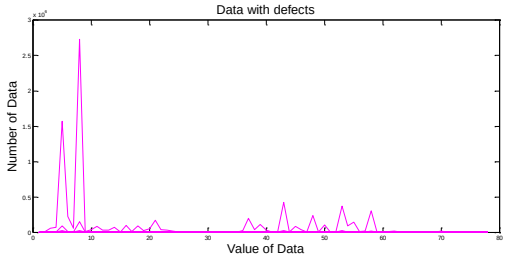


**Figure 4: Input PC1 dataset with defect attributes**

Figure 5 show only the data without defect. Where X-axis contains the value of data and Y-axis contains the number of data.

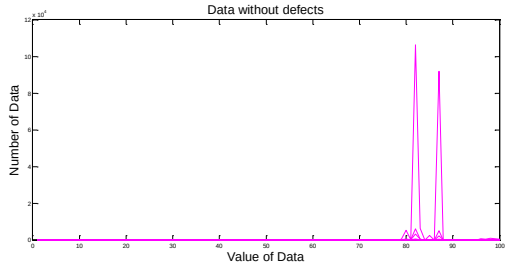


**Figure 5: Input PC1 Dataset with without Defect Attributes**

## 9. CONCLUSION

In this paper, the software defect prediction model is implemented using fuzzy c-means clustering, K-nearest neighbors' classifier and hybrid (FCM + GA) approaches. Table 1 shows the performance comparison for all the algorithms. It was found that the hybrid approaches gives more accuracy and less defects as compared to Fuzzy C-Means Clustering approach and K-Nearest Neighbors method on the basis of evaluation parameters: Reliability, Accuracy, Mean Absolute Error (MAE), and Root Mean Square Error (RMSE).

In the Future we want to improve the proposed defect prediction model by a better approach to get better accuracy & reliability. So that, the performance of the system should be improve.

## 10. REFERENCES

[1] N. E. Fenton and S.L. Pfleeger, "Software Metrics-A Rigorous and Practical Approach," PWS Publishing company, 2nd edition, 1997.

[2] S. H. Kan, "Metrics and Models in Software Quality Engineering," Addison Wesley, 2nd edition, 2002.

[3] P. Kumar, "Aspect-Oriented Software Quality Model: The AOSQ Model," in Advanced Computing: An International Journal (ACIJ), Volume: 3, Number: 2, Page No.:105-118, March 2012.

[4] M. Ceccato and P. Tonella. "Measuring the effects of software aspectization" 1st Workshop on Aspect Reverse Engineering (WARE), Volume: 12, 2004.

[5] P. Kumar and S. K. Singh, "A Systematic Assessment of Aspect-Oriented Software Development (AOSD) using JHotDraw Application," IEEE International Conference on Computing, Communication and Automation (ICCCA2016), Greater Noida, India, April 2016.

[6] NASA Metrics Data Program Web Site: http://promise.site.uottawa.ca/SERepository/datasets/pc1.arff.

[7] Fuzzy C-means (FCM) algorithm, online available at: http://hayoungkim.tistory.com/20.

[8] *K-Nearest Neighbors*, online available at: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm.

[9] R.K. Gupta, "Genetic Algorithms-an Overview", IMPULSE, ITM University, Vol. 1, 2006.