Volume 5, Issue 5, MAY 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Defect Prediction and Ranking by Taking the Advantages of Fuzzy Mathematics

Priyanka C A
MTech [SE], CSE Department & VTU
Karnataka, India

Dr. G. G Sivasankari
HOD and Professor CSE Department &VTU
Karnataka, India

Abstract-Intending to overcome obstacles confronted amid programming advancement movement through abusing programming quality forecast and positioning by taking favorable circumstances of fluffy math. Throughout the most recent couple of decades numerous product quality displaying fluffy strategies have been produced and utilized as a part of genuine programming quality forecast. This paper introduces an alternate methodology of ahead of schedule programming quality expectation and positioning. Quality forecast is finished by characterizing programming modules as FP or NFP. Later, modules are positioned utilizing.

Keywords: Software quality, metric, fluffy ordering algorithm, fault-prone module.

I. INTRODUCTION

The climbing convolution in the PC programming frameworks and the regularly rising client desires has added to the essential to deal with the product quality (greatness), concoct a building control called Software Quality Engineering. Programming issue forecast is one of the quality affirmation exercises in the Software Quality Engineering control. Large portions of the product frameworks, for example, those executed in telecom and restorative territories oblige an abnormal state of value confirmation. The administration of these frameworks requires an assessing procedure of the nature of programming modules, which could be possible by actualizing programming issue expectation procedures. A task chief or a quality certification gathering can enhance the item quality by distributing important spending plan and HR to manage the shortcoming inclined modules distinguished by deficiency expectation models and positioning. Likewise, the exertion of testing stage would be minimized. Programming deficiency forecast models oblige programming measurements, gathered with mechanized instruments and issue information having a place with past programming variant or comparative programming venture. In programming shortcoming expectation issues, we have X = {x1, x2 … xn} where x speaks to programming module that is described by programming measurements and Y = {fp, nfp}, where fp and nfp be the deficiency inclined and non-blame inclined individually arrangement is done through fluffy surmising framework and ID3 (Iterative Dichotomiser 3) algorithm. Further, modules are ranked on the basis of degree of fault proneness using fuzzy ordering algorithm T. J. Ross (2010).

Remaining part of the paper is organized as follows: next section discusses about the related research on this topic. Section 3 describes the brief background work data mining technique and fuzzy set theory Zadeh, L. A (1965). Section 4 provides information about proposed model. Section 5, 6 explains prediction and ranking procedure along with results and at last conclusion of the topic is given in section 7.

II. RELATED WORK

Decision tree method is used for classifying the modules as fault prone and non-fault prone Khoshgoftaar, T. M. and N. Seliya et al (2002), Neural Network approach Wang,Jie Zhu and Bo Yu et al(2005), Bayesian Methods Pai, G. J. and J. B. Dugan et al (2007) and Fuzzy Integration technique A potential technique to automatically evaluate qualitative attributes is to use software metrics as quantitative predictors and Fuzzy Logic provides a number of functions for aggregating two or more fuzzy sets or fuzzy relations .While doing literature survey its observed that various learning approach exist such as supervised, semi-supervised, and unsupervised have used for predicting fault model Seliya, N. and T. M. Khoshgoftaar et al (2007). Supervised learning approach is the best method found to identify FP or NFP module prediction.

By and large, these methodologies use programming measurements and shortcoming information of prior programming discharges. Menzies et al. (2007) demonstrated that abscond indicators can be found out from static code qualities since they are helpful, simple to utilize, and broadly utilized. Taking signs from. Menzies et al. (2007) , Pandey and Goyal et al (2009) displayed an early blame expectation model utilizing methodology development and programming measurements. Different arrangement models have been created for grouping a product module as FP and NFP. Khoshgoftaar et al. Pizzi, N. J. et al (2008) connected relapse trees with order guideline to group issue inclined programming module utilizing a vast telecom framework as a contextual investigation.

From audit it has been watched that the choice tree impelling calculations, for example, CART, ID3 and C4.5 are productive strategy for module arrangement. These calculations utilization fresh estimations of programming

measurements and group modules as deficiency inclined or not blame inclined. Anyway, it has been discovered that early stage programming quality measurements have fluffiness in nature and fresh esteem task is by all accounts unrealistic. Likewise a product module can't be penny precent deficiency inclined or not blame inclined as every one has a related level of flaw inclination. The vast majority of the studies reported in writing have concentrated just on characterization of modules as issue inclined or not-blame inclined. Then again, it will be more valuable if the modules can likewise be positioned on the premise of their shortcoming inclination and some measure of flaw inclination is inferred.

III. BACKGROUND

A. Data Mining Technique

Information mining involves the general procedure of separating learning from a lot of information. Diverse sorts of information mining methods are talked about in the writing, for example, relapse, order and affiliations Sayyad, S. J., and T. J. Menzies(2005). The emphasis here is on characterization system, which is the undertaking of characterizing the information into predefined classes considering different prescient attributes.

The consequence of a grouping method is a model which makes it conceivable to group future information focuses in view of an arrangement of particular attributes in a mechanized way. Numerous grouping methods are accessible in writing, for example, ID3, C4.5, logistic relapse, k-closest neighbor, Artificial Neural Networks (ANN) and Support Vector Machines (SVM). These systems have been effectively connected in diverse areas like managing an account and money, restorative, showcasing in the retail division, and credit scoring in the different segment. This paper concentrates on the utilization of information mining system alongside fluffy demonstrating to order programming modules as FP or NFP.

B. Fuzzy Approach

Fresh or Classical set can be characterized as an accumulation of very much characterized unmistakable item. At the end of the day, fresh sets contain objects that fulfil exact properties of participation. For fresh set, a component x in the universe X is either an individual from some fresh set (P) or not. This parallel issue of enrolment can be can be spoken to by a trademark work as:

$$\chi_P(x) = \begin{cases} 1, if\ x \in P \\ 0, if\ x \notin P \end{cases} \quad (1)$$

Where χP (x) gives the unambiguous enrolment of the component, set P contains x.

Fuzzy Sets

A fluffy set is a situated containing component that has fluctuating level of participation in the set. Not at all like fresh set, components in a fluffy set need not be finished and can likewise be individual from other fluffy sets on the same universe.

Let $\tilde{P}$ is a fluffy situated of P , if a component in the universe, say x, is an individual from fluffy set $\tilde{P}$, then mapping is given by a participation capacity $\mu\tilde{P}(x)$. Gives the level of enrolment for every components in the fluffy set $\tilde{P}$ and is characterized in the reach [0, 1] where, 1 speaks to components that are totally in $\tilde{P}$, 0 speaks to components that are totally not in $\tilde{P}$, and values somewhere around 0 and 1 speak to fractional participation in $\tilde{P}$. Formally, a fluffy set $\tilde{P}$ can be spoken to by Zadhe's documentation [2] as: $\tilde{P} = \{\frac{\mu_1}{x_1} + \frac{\mu_2}{x_2} + \cdots + \frac{\mu_n}{x_n}\}$ where, $\mu_1, \mu_2, \ldots, \mu_n$ are the membership values of the elements $x_1, x_2 \ldots . x_n$ respectively, in the fuzzy set $\tilde{P}$.

Fuzzy Ordering

Propensity to settle on choices is relied on upon positioning specific model when the issue or activity connected with out of succession or ambiguity.it is difficult to think of the choice which issue is the best, which is second best, etc. Fluffy requesting handles out of grouping or vagueness. Case portrays more insight about fluffy requesting, let two fluffy sets are $\tilde{Q}_1$ and$\tilde{Q}_2$. Fluffy set $\tilde{Q}_1$ is more prominent then $\tilde{Q}_2$ if the accompanying condition satisfies,

$$T(\tilde{Q}_1 \geq \tilde{Q}_2) = max_{(x1 \geq x2)} \{min(\mu_1(x_1), \mu_2(x_2))\} \quad (2)$$

Where $T(\tilde{Q}_1 \geq \tilde{Q}_2)$the fact of the matter is esteem on the interim [0, 1] and 〖 μ〗_1 (x_1), μ_2 (x_2) speaks to the level of participation of the first component in the fluffy set Q1 and Q2 separately. The summed up type of eq. (2) for k fluffy sets Q1, Q2… Qk. At that point, reality is given as,

$$T(Q \geq \tilde{Q}_1, \tilde{Q}_2, \ldots \tilde{Q}_k) = [(T(\tilde{Q} \geq \tilde{Q}_1) \wedge T(\tilde{Q} \geq \tilde{Q}_2) \wedge \quad \ldots \wedge T(\tilde{Q} \geq \tilde{Q}_k)] \quad (3)$$

$$T(Q \geq \tilde{Q}_1, \tilde{Q}_2, \ldots \tilde{Q}_k) = min[(T(\tilde{Q} \geq \tilde{Q}_1) \wedge T(\tilde{Q} \geq \tilde{Q}_2) \wedge \quad \ldots \wedge T(\tilde{Q} \geq \tilde{Q}_k)] \quad (4)$$

IV. MODEL ARCHITECTURE AND IMPLEMENTATION

A. Model Architecture

Figure 1 Shows assumption of complete system architecture. Information about software faults of software is assumed to be stored in Software metrics. This information helps in quality prediction of software at development stage. Data of similar domain software projects gives better training to the model. Model architecture also assumes that Iterative Dichotomise 3 classification algorithm is the best way for purpose of fault prediction.

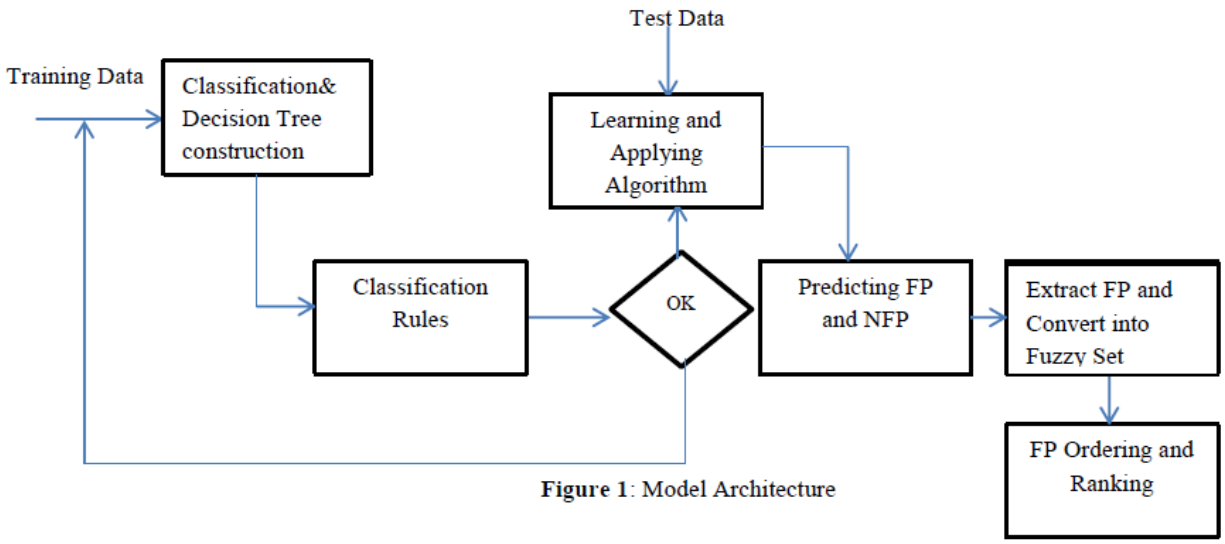

Fig. 1 Model Architecture

B. Implementation

MATLAB is the product used to actualize this model by utilizing three noteworthy ventures as tails: (i) Pre-handling information, (ii) Learning, (iii) Prediction and Ranking.

Information Pre-preparing: In administered learning calculations information preparing is extremely crucial part. A large portion of the information introduce in genuine venture are boisterous, due to huge size code and intricacy of the undertaking may contain commotion. To get superb preparing information must be pre-processed. Fragmented, loud, and repetitive information are basic spot properties of a few true venture information. There are numerous conceivable explanations behind veering off from ordinary expected result. KC2 Public NASA datasets incorporate 21 system level measurements proposed by Maurice Halstead and Thomas McCabe. In any case, a few scientists generally utilize just 13 measurements from these datasets. The KC2 venture is the science information preparing unit of a stockpiling administration framework utilized for accepting and transforming ground information for missions. This venture information set contains 522 system modules, of which 107 modules have one or more blames while staying 415 modules are without deficiency i.e., no flaws.

TABLE I Software Metrics of KC2 Dataset

Metrics	**Descriptions**
LOC	McCabe's line count of code
EL	Executable LOC
CL	Comment LOC
BL	Blank LOC
CCL	Code and comment LOC
N1	Total no. of operators
N2	Total no. of operands
CC	McCabe's cyclomatic complexity
EC	McCabe's essential complexity
DC	McCabe's design complexity
BC	Branch count of flow graph
n1	No. of unique operators
n2	No. of unique operands

Every system module in the KC2 was portrayed by 21 product measurements (5 separate lines of code measurements, 3 McCabe measurements, 4 base Halstead measurements, 8 determined Halstead measurements, 1 branch-tally) and 1 target metric, which says whether a product module is flaw inclined or not. Out of these 21 product measurements, just 13 measurements (5 separate lines of code measurements, 3 McCabe measurements, 4 base Halstead measurements, and 1 branch- number) are considered on the grounds that 8 determined Halstead measurements don't contain any additional data for programming shortcoming expectation. Propositions measurements are recorded in Table 1.

Realizing: Many characterization models have been proposed, for example, neural systems, bolster vector machines, Decision trees (DT) and others. Choice trees are more appealing than others. The most famous is ID3 (Interactive Dichotomiser 3) acquainted by Quinlan are utilized with create choice tree for order from typical information. The information spoke to in choice tree can be spoken to as grouping "IF-THEN" principles.

Prediction and Ranking: When choice tree is developed, order principles are extricated from the tree by following a way from the root to a leaf hub. These arrangement guidelines are connected on the one allotment KC2 dataset to prepare the classifier and diverse parts of these dataset are utilized for module forecast. Classifier can order programming modules as FP or NFP however it can't appoint the rank to a module on the premise of level of flaw inclination. Subsequently, a fluffy requesting calculation is connected on these FP and NFP module to get the level of issue inclination and positioning of the modules.

V. RESULTS

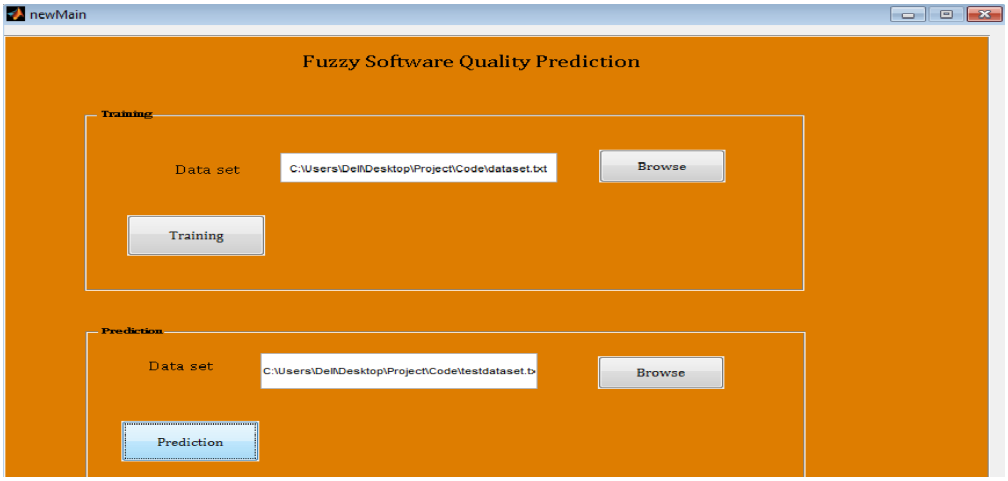

Fig 2: User interphase to predict faults.

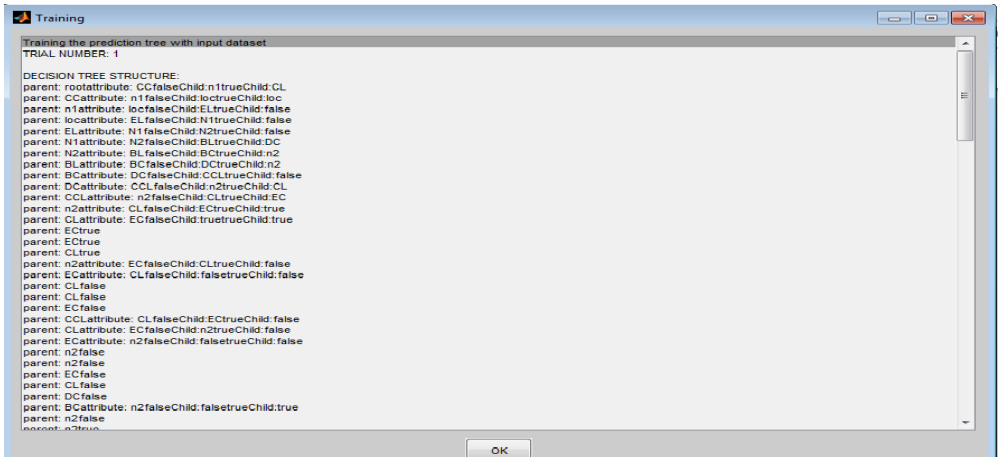

Fig 3: Decision tree construction.

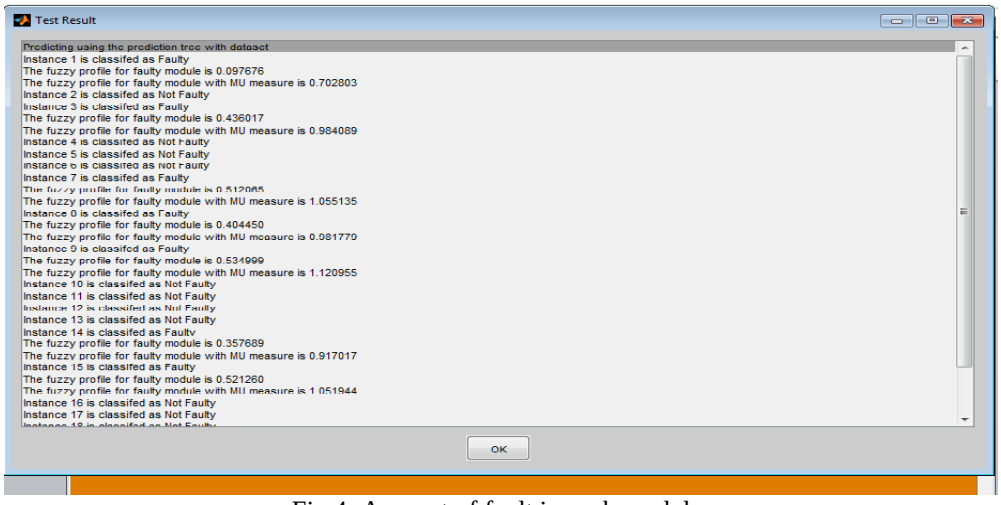

Fig 4: Amount of fault in each module.

Figure 2. Provides the user interface to browsing the data and predicting the faulty module. Figure 3. Shows the constructed decision tree using ID3 algorithm. Figure 4. Gives the information about amount of fault present in each faulty module

VI. CONCLUSIONS

This study has proposed another model for forecast and positioning of deficiency inclined module for a vast programming framework. ID3 calculation is utilized to characterize programming modules as deficiency inclined then again not blame inclined. In the meantime, fluffy requesting calculation is connected to rank fault-prone modules on the premise of their level of issue inclination. Positioning of issue inclined Module alongside grouping discovered to be another way to help in organizing and allotting test assets to the individual programming modules. The proposed model has been connected and contrasted and existing model for the KC2 information of NASA. The outcomes watched are guaranteeing and display great precision and consistency, when contrasted and a portion of the prior models.

ACKNOWLEDGMENT

I would like to thank Dr.G.G.Sivasankari Head of the department, CSE, AMC Engineering College Bangalore, For her constant motivation and support for above stated work, and I would like to extend my sincere thanks to all faculty of computer science and engineering, AMC Engineering College, Bangalore, for their cooperation.

REFERENCES

[1] T. J. Ross. Fuzzy Logic with Engineering Applications. Willy-India Publication, 2010.
[2] Zadeh, L. A. Fuzzy Sets. Information and Control, 1965; 8(3): 338-353.
[3] Khoshgoftaar, T. M. and N. Seliya. Software Quality Classification Modeling Using the SPRINT Decision Tree Algorithm. 4th IEEE International Conference on Tools with Artificial Intelligence, Florida, 2002; 365-374.
[4] Wang,Jie Zhu and Bo Yu. Combining Classifiers in Software Quality Prediction: A Neural Network Approach.ISNN 2005, LNCS 3498, pp.921-926.
[5] Pai, G. J. and J. B. Dugan. Empirical Analysis of Software Fault Content and Fault Proneness Using Bayesian Methods. IEEE Trans on Software Eng., 2007; 33(10): 675-686.
[6] Pizzi, N. J. Software Quality Prediction Using Fuzzy Integration: A Case Study. SoftComputing-A Fusion of Foundations, Methodologies & Application., 2008; 12(1): 67-76.
[7] Seliya, N. and T. M. Khoshgoftaar. Software Quality Estimation with Limited Fault Data: A Semi-Supervised Learning Perspective. S/W Quality Journal, 2007; 15 (3): 327-344.
[8] Menzies, T., J. Greenwald and A. Frank. Data Mining Static Code Attributes to Learn Defect Predictors. IEEE Trans on Software Eng., 2007; 33 (1): 2-13.
[9] Pandey, A. K. and N. K. Goyal. A Fuzzy Model for Early Software Fault Prediction Using Process Maturity and Software Metrics. International Journal of Electronics Engineering,2009; 1(2): 239-245.
[10] Sayyad, S. J., and T. J. Menzies. The PROMISE Repository of Software Eng. Databases.University of Ottawa, Canada, http://promise.site.uottawa.ca/SERepository, 2005.
[11] Han, J., M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann Publication, USA, 2001.

