INTERNATIONAL JOURNAL OF
RECENT TRENDS IN MULTIDISCIPLINARY RESEARCH
November-December 2024, Vol 4(06), 07-10 ISSN No: 2583-0368

Optimized Fuzzy Classifier Approach for Predicting Defects

Kaushalya Thopate[1], Diya Shaikh[2], Muaz Shaikh[3], Pushkraj Shahane[4]
[1]Asst Prof. Department of Computer Science Engineering Vishwakarma Institute of Technology, Pune, Maharashtra, India.
[2,3,4] Students, Department of Computer Science Engineering, Vishwakarma Institute of Technology, Pune, Maharashtra, India.

OPEN ACCESS
Article Citation:
Kaushalya Thopate[1], Diya Shaikh[2], Muaz Shaikh[3], Pushkraj Shahane[4], "Optimized Fuzzy Classifier Approach for Predicting Defects", International Journal of Recent Trends in Multidisciplinary Research, November-December 2024, Vol 4(06), 07-10.

Publishedby5thDimension Research Publication

Abstract: The product and process quality in developing software is characterized by software attributes. Certain attributes of quality of software like the density of defect and rate of failure form the measures of the software product and its development process. Care is taken to utilize software metrics and code level measurement and defect data in the building of defect predictors or software quality models. The assumption is the software metrics can gather the end product quality. In general, the building of software metric models takes place through data that defect which is collected from a system release that is developed previously or software projects that are similar. By validating these models, fault proneness is readily predicted of program modules that are development currently. The available quality improvement resources can be justified through the application of a low-quality Fault-Prone (FP) prediction to those programs. High software reliability and quality are achieved by using the available resources effectively.

Key Words: Predicting Defects, Fault-Prone (FP), Software Defect Prediction (SDP)

1. Introduction

Based on the stage of the software development process, ideal features are selected by metrics validation for defect prediction. Defects can be prevented beforehand through defect prevention techniques and help in the Quality Improvement Program (QIP). Risks related to specific defect types can be identified through Root Cause Analysis (RCA). Defects in software products are identified and classified through Orthogonal Defects Classification (ODC) Another tool which is used in creating, revising and archiving defect prevention is "Performance and Continuous Re-Commissioning Analysis Tool (PACRAT)." An ideal solution for defect prediction is impossible because of the software products' diversified nature. In software development, resource allocation is the most expensive component.

Class imbalance and noisy attributes of a dataset are the qualities that can affect the performance of classification. There is an imbalanced nature to the dataset which is not possible practically in Software Defect Prediction (SDP) as a prediction of most defects are considered defect prone. It is not easy to learn from imbalanced datasets. A clear understanding is not possible through the lack of information associated with the minority class and there is a decrease in SDP performance as there are noisy attributes to the datasets. Such simple methods are not sufficient to assume that the noisy points in the datasets are erroneous. In Machine Learning feature selection is made use generally when the learning task has to go through big dimensional and noisy attribute datasets. Local search is used throughout the whole process in many of the feature selection protocols, which as a consequence is near optimal to optimal solutions are not easy to attain.

The issue of SDP in medical software is addressed in this work. The membership function should be chosen efficiently for using Fuzzy classifier and the selection of rule efficiently for either classification or regression problem on data that is separable linearly. One such optimization technique of recent origin is Cuckoo Search (CS) that is kindled from the behavior of Cuckoos along with Levy flight behavior of birds and fruit-flies.

2.Optimization Is Np-Hard

As all real-life problems are described as a certain type of optimization issue, it is a main applicable area in mathematics and computer science. The modes of mathematical relations within the objective function, constraints potentially and decision parameters describe the hardness of a specific issue. The hard optimization issues could be either continuous or discrete where

continuous problems can be either constrained or unconstrained. The optimal solution of a constrained problem has to be located inside the feasible space $F \square\square S$ which means that the solution satisfies all the constraints. Since much of the optimization protocols begin randomly, unfeasible solutions in the initialization phase, it is expected that followed by few iterations these solutions can attain the feasible area. The handling of equality constraints is a tedious issue for optimization methods as their existence could make the possible space very little when compared with the total space for search.

3.Optimization Methods On SDP

Optimization is explained as the process to search a vector in a function which gives an optimal solution. Existing solutions are the feasible value and optimal solution is the extreme value. Generally, these optimization solutions are solved through optimization algorithms. Optimization algorithms are classified into two depending on a plain classification method to optimize based on the algorithms' character and they are deterministic and stochastic. Illustration for deterministic includes hill climbing where the same sets of solutions are produced from the initial starting point to the end. whereas, stochastic algorithms produce a different solution even with the same initial value. Generally, two types of stochastic algorithms – heuristic and metaheuristic. Recent studies have proved that nature-inspired metaheuristic techniques show powerful performance and efficiency when obtaining a solution to recent non-linear numerical issues of global optimizing. To a certain extent, all metaheuristic algorithms work hard to make a compromise between randomization and local search. Genetic Algorithms (GA) can identify the near-optimal global solutions from search space not computing the gradient data. Binary string vector representation such as chromosome structure of biology is used in original GA's learning. But binary GA has issues as its objective of learning is to enhance the correctness of the network and speed convergence is not given importance and local error.

At every step, there is an iteration of GA and a population shapes another population. The following steps are salient in a GA cycle:

 Selection: For reproduction, an initial step is chosen by individuals/chromosomes.

 Fitness value: This is very significant during the selecting process and is entirely random. Species with good fitness values are selected regularly to reproduce.

 Reproduction: Through preferred individuals, offspring are allowed. New chromosomes are formed by recombination and mutation methods.

 Evaluation: Assessment of fitness of new chromosome takes place.

 Replacement: In this step, the existing population is interchanged by new ones.

A bird in searching space is the potential solution for the optimized problem which is termed as "particle" and the particle's position is the potential solution. A group of particles is initialized randomly at first in PSO and every particle traverse the solution space. The movement direction and particle displaced of a vector is decided by the moving direction and fitness corresponding is computed using a function to judge if the target is attained or not. Through iterations, the optimal solution is attained. By tracing two extreme values in every iteration, the particles update themselves.

Ant Colony Optimization (ACO) the algorithm is a scheme that is used widely depending on the foraging behaviour of ants that is the behavior of ants while searching for food. In real life, while ants leave their nest in search of food, leave a trail of pheromone which is a chemical substance which evaporates but leaves a trail. The best source of food is the one which has more pheromone trail and following this the other ants take the route. In artificial ant systems, they collect information regarding the characters of problems by building solutions. This message is encoded in pheromone. The ant is guided by the pheromone and the problem representation is modified using the pheromone and this change can be seen by the other ants. Pheromone evaporates when the process of solution building happens. This assists in the avoidance of convergence of all ants to a single.

4. Methodology

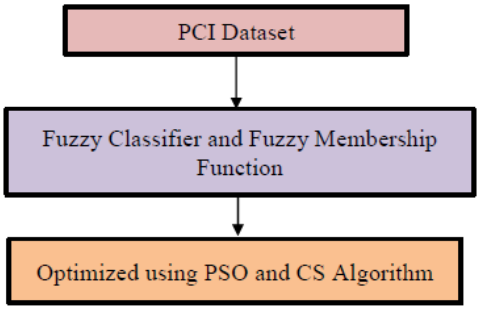

Figure 1 Flowchart of Proposed Methodology

PCI dataset is used in this section with fuzzy rule selection, membership function, PSO and CS. PSO is a metaheuristic inspired in social behaviors, which is very useful in optimization problems. As CS uses the global search of L´evy flights or process for random walks it is more advantageous as it can use infinite means and variance, the search space can be explored effectively than algorithms by standard Gaussian process. This together with both local and search capabilities and guaranteed global search, it can make CS more effective. Indeed, many investigation and applications have shown that CS is much effective. Figure 1 shows the flowchart of the proposed methodology.

Defect detectors are computed as:
a = Classifier predicts no defects and module has no error.
b= Classifier predicts no defects and module has the error.
c = Classifier predicts some defects and module has no error.
d = Classifier predicts some defects and module has the error.

Accuracy, detection probability (pd) or recall, precision (prec), probability of false alarm (pf), and effort are computed as Equation (4.5-4.9):

$$Accuracy = \frac{a+d}{a+b+c+d}$$

$$recall = \frac{d}{b+d}$$

$$pf = \frac{c}{a+c}$$

$$prec = \frac{d}{c+d}$$

$$effort = \frac{c.LOC + d.LOC}{TotalLOC}$$

Fuzzy Classifier

A natural way is provided by this concept to deal with issues where the vital knowledge of impreciseness in the lack of precisely explained criterion. Here, linguistic uncertainties control the phenomena taken into consideration. There is a fuzzifier, fuzzy engine, and a defuzzifier in a typical fuzzy system. Because of its associated simplicity, Mamdani technique is more prevalently used because of its fuzzy interference engine, even though many different approaches exist. A sequence of fuzzy interface rules describes the internal structure of the fuzzy engines. A typical fuzzy system has 4 steps

- An input value is translated into linguistic terms using membership functions. The quantum of the particular input value that is considered, suit the linguistic constraints, that is finalised using the membership functions

 Rules of Fuzzy-Logic are wrapped, holding into study the different permitted input and output membership function combinations. These rules are then explained by the observation of experts in that field analyzed.

 The rules which are formulated are now made use in the membership functions and the outputs of every rule is aggregated. A fuzzy interference engine performs this function that maps the input membership functions along with the output membership functions making use of the derived Fuzzy Logic.

- Defuzzification is the process of conversion of the net output of fuzzy to a solid number.

The basis of Fuzzy-Logic model is the reasoning concept while it works on a value that is by nature approximate. It is an upgrade of the traditional Boolean Logic which can contain only true or false and in Fuzzy-Logic, the truth of each statement is not a whole number but a degree. As this is based on the intuition of humans and behavior, this model added to a yes or no answer.

Inputs are used in this model and are put in a particular range. The rules' set is described after this which shows and predicts the way in which inputs are utilized in achieving the output and determining the defined value of the fuzzy set. There are a metrics set for the model or Reliability Relevant Metric List (RRML) that is made available from the software metrics. These metrics are appropriate to their corresponding phases during the software development life cycle.

This was processed by Lotif A. Zadeh at the University of California in Berkeley. This is logic with multiple values in-between conventional examinations such as true/false, yes/no and low/high, etc. Control field is the most important application area of fuzzy logic. Fuzzy control is applied successfully to different issues like fans control, complex aircraft engines and control surfaces, wheel slip control, helicopter control, automatic transmission, industrial and missile guidance. Fuzzy rules are suitable for the representation of the classification knowledge. It is mainly the concept from numbers to linguistic attributes, making them easy to be read and interpreted.

5. Conclusion

The proposed method can improve the quality of results and increase the diversity of solutions to a problem. The fuzzy systems were designed for adjusting the parameters for PSO, which was obtained in two systems statistical evidence of an improvement in the quality of the results of the method of PSO when applied in the minimization of benchmark mathematical

functions. The CS algorithm mimics the breeding behavior ofcuckoos, where each individual searches the most suitable nest to lay an egg (compromise solution) in order to maximize the egg's survival rate and achieve the best habitat society. Fuzzy set theory is used to create the fuzzy membership search domain where it consists of all possible compromise solutions. CS algorithm searches the best compromise solution within the fuzzy search domain simultaneously tuning the fuzzy design boundary variables. Experimental results show that the CS-fuzzy classifier has higher classification accuracy by 1.4% & 0.28% for PC1 dataset and by 0.94% & 0.43 for PC2 dataset when compared with fuzzy classifier and PSO-fuzzy classifier. In the next chapter, discuss hybrid CS method.

Reference

1. *Arshad, A., et al.: The empirical study of semi-supervised deep fuzzy C-mean clustering for software fault prediction. IEEE Access* ***6****, 47047–54706 (2018).*
2. *Bal, P.R., Kumar, S.: WR-elm: Weighted regularization extreme learning machine for imbalance learning in software fault prediction. IEEE Trans. Reliab.* ***69****(4), 1355–1375*
3. *Borandaq, E.: Software fault prediction using an RNN-based deep learning approach and ensemble machine learning techniques. Appl. Sci.* ***13****(3), 1639 (2023)*
4. *Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Art. Intell. Res.* ***16****, 321–357 (2002)*
5. *Desuky, A.S., Hussain, S.: An improved hybrid approach for handling class imbalance problem. Arab. J. Sci. Eng.* ***46****, 3853–3864 (2021)*
6. *Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res.* ***3****, 1157–1182 (2003)*