International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 3 | ISSN: 2456-3307

Application of Adaptive Neural Fuzzy Inference System for the Prediction of Software Defects

G. Rajendra[1], Dr. M. Babu Reddy[2]
[1]Research Scholar, Department of Computer Science, Rayalaseema University, Kurnool, Andhra Pradesh, India
[2]HOD, Department of Computer Science, Krishna University, Machilipatnam, Andhra Pradesh, India

ABSTRACT

One of the major challenges in the process of software development is the prediction of software defects for the reduction of cost for the implementation of any software. Great cut off cost is done in software field because of the implementation of the predicting defective modules. Many scholars have used many techniques from various aspects such as data mining for the prediction datasets of the software defects that can be downloaded from NASA repositories. The datasets downloaded are pretty much imbalances on their own. In the current study we compute the Adaptive inference for neural fuzzy systems. Subtractive Clustering Method (SCM) has given the base structure for the initial fuzzy inference system and innovative rule for learning is used for their training. The balance in the datasets is lost but the performance can be understood with the help of the values **(AuC)** for the classifier. We did the computation and compared neural networks that are cost sensitive with the FIS. The section which consists of the results contains the curves of operating characteristics of the receiver which are computed and presented. The cost sensitive methods compared with the FIS, and the drawn results are found to be satisfactory.

Keywords: SCM, FIS, SCM, Neural fuzzy systems.

I. INTRODUCTION

A major contribution to the software development is done by the software that predicts the defects. The prediction of the defects can minimize the cost of the development of software and it will fortify the product quality in the software development. The detection of defects in software development has many approaches

1. Model of simple metrics and estimation of defects: In metrics of s/w, the no of defects is implied on the code in lines (CIN). The equation that is derived is Defect number (d) =0.018*CIN+4.86. However total complexity of software can't be computed based on CIN alone.
2. Models of fitting and complex metrics: For computation of the complexness of any software the cyclomatic metrics (CMM) was introduced. CMM of any program is **C (p) =N-L+2**, where N is no of edges and L is no of vertices. The introduction of measures of Halsted complexities by Halsted takes various language algorithm implementations. It is observed that effort determines the defect number and it is also determined by the level of difficulty and volume. **Defects (f) = ¯ /3000**. However this is a model for the statistical data given no future computational result can apply and prove the implementation based on this model.
3. Model of regression: Actually new modules can be validated on the model of linear regression. It found that magnitude mean of relative error (MMRE) which is in middle of predicted and true defect number as 0.48. Using CMM for complexity along with regressions of logistics, when an applied discriminated

analyzation is done to acquire a total of 92% accuracy.

4. Model for prediction just in time: This model gives a round off 68%-accuracy along with 64%-recall within commercial and open source projects on computing a large statistical data. This model has a limitation on the validating difficult on practically.
5. Models of practicality: For the prediction of software defects an object oriented set of rules called CK metrics were introduced. It has many metrics introduced in them.
6. Models for prediction of history metrics: Particular program characteristics were not extracted by history metrics like social network of developer, anti-pattern and component network. This model can't be put to implementation if historical data is missing and also for new projects.
7. Prediction of defects by cross project: Projects lacking in historical information and also new projects are predicted with these models.

II. RELATED WORK

Faruk Arar et.al implemented Artificial Neural Community (ANN) for building a model for software program illness Prediction. He applied Artificial Bee Colony (ABC) for optimizing connection weights in ANN. This version changed into implemented to five publicly to be had datasets from the NASA repository. Jun Zheng et al studied 3 fee-sensitive algorithms for software illness prediction. The performances of the 3 algorithms are evaluated by means of the usage of four datasets from NASA projects. A comparison of soft computing algorithms for software program illness prediction is performed through Ertruk. E Erturk applied Adaptive Neuro-fuzzy inference system for software disorder prediction. Rodriguez used datasets from PROMISE repository and implemented feature selection and genetic algorithms for predicting defective modules. Guo counseled the use of the Random wooded area for predicting software program modules. Selection tree Freshmen are used for expecting the illness densities in SDP datasets.

Methodology

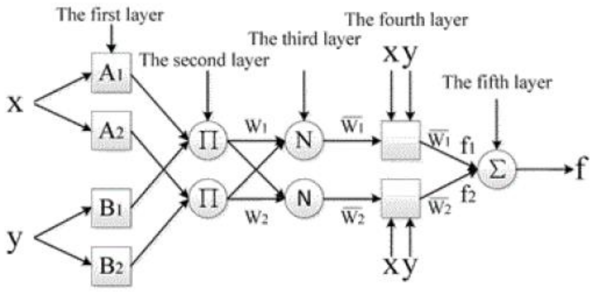


Fig.1 Adaptive Neuro Fuzzy Inference System with two inputs

The trouble with fuzzy inference gadget is identification of regulations. Policies are generated both by means of the usage of grid partitioning or subtractive clustering methods.

Grid Partitioning

Grid partitioning divides every input variables into sub intervals and forms the rule for each feasible interval of enter variables. For the variables with continuous values, this method generates a huge collection of guidelines which over fits the facts. As a result this approach isn't always suitable for the datasets such as continuous variables.

Subtractive Clustering

Subtractive clustering generates the nice tuned clusters for every input variable. A rule is generated for each cluster of enter variables. Subtractive clustering is a set of rules considers every facts point as a candidate for cluster Centre. First, density measure for all information factors is calculated. For xi it is defined as

$$D_i = \sum_{j=1}^{n} e\left[\frac{-(x_i - x_j)}{(r_a/2)}\right]$$

A record point with many neighboring factors has excessive density measure. r_a is the range of statistics point that the minimal wide variety of neighboring statistics points lies. Records factor with highest density cost is chosen as first cluster centre. r_b is then used as radius which defines the neighborhood in which density discount is finished.

$$D_i = D_i - D_{c1}\, e\left[\frac{-(x_i - x_{c1})}{(r_b/2)}\right]$$

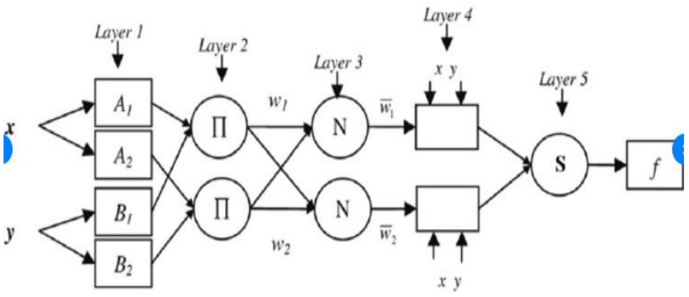
After reduction is completed statistics point with highest density degree is selected as cluster xc2. Next we use cluster xc2 for density discount. This manner is repeated till Dck < AD1.

Education FIS

There are various methods which might be proposed to assemble the version for software disorder prediction. In this paper we're offering Adaptive Neuro Fuzzy Inference device for constructing the classifier.

Adaptive Fuzzy Inference device (ANFIS)

ANFIS is a five layered architecture which derives Sugeno kind Fuzzy Inference system the usage of hybrid studying rule. Figure 1 suggests the structure of ANFIS.



Effects & dialogue

Table 1: Outcomes of neurons

Dataset /Alg	CS NN l:2,n:15	CS NN l:2,n:30	CS NN l:2,n:60	CS NN l:3,n=90	ANFIS
kc1	0.5	0.5	0.5	0.5	0.8482
kc2	0.64958	0.66451	0.660324	0.534307	0.8998
cm1	0.5	0.5	0.5	0.5	0.8904
pc1	0.58976	0.548134	0.637584	0.561159	0.8416

Layer 1 is an adaptive layer that computes the club values of enter variables. In this paper we're using Gaussian club function.

Layer 2 is a set layer that computes the firing energy of enter variables using product operator.

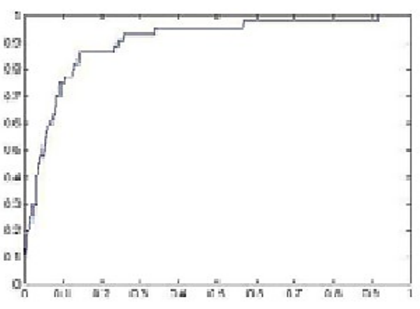
Layer 3 is a fixed layer that computes the normalized weight of the firing rule

Layer 4 is an adaptive layer that uses neural networks for first-rate becoming of consequent parameters (p, q, r) of node characteristic f(x, y) = p*x + q*y + r.

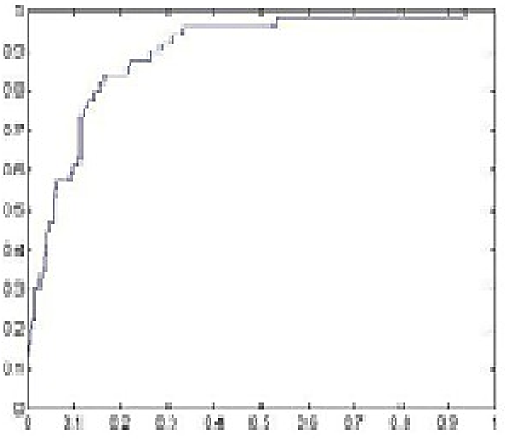
Layer 5 is a hard and fast node that computes the overall sum of input indicators.

Expertise Extraction based on Evolutionary gaining knowledge of (keel) is an open source tool to behavior experiments on distinct datasets to assess the performance of numerous studying algorithms. We carried out experiments on software program disorder prediction using four datasets downloaded from NASA information repository to become aware of faulty susceptible modules. Neural networks are one of the methodologies for excellent fitting nonlinear relationships between attributes. Fee sensitivity is carried out to neural networks to improve the performance of the classifier.

In Keel experimentation phase, fee sensitive Neural Networks are applied the usage of JAVA. For experimentation, we imported 4 SDP datasets in facts management segment. Inside the experimentation segment, we blanketed dataset and fee touchy neural community algorithm is imported and the consequences are showed the use of imbalance take a look at methods. These sets of rules are repeated with one of a kind parameter values. Inside the first generation, varieties of layers are fixed to two with a total of 15 neurons. In the next iterations the wide variety of neurons are extended to 30 accompanied by using 60. Ultimately the set of rules is examined by way of 3 layers with ninety neurons; these outcomes are presented in table 1.



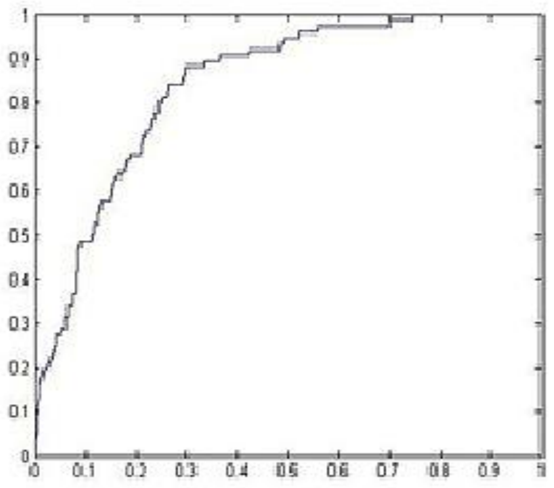
Kc2 dataset



Cm1 dataset

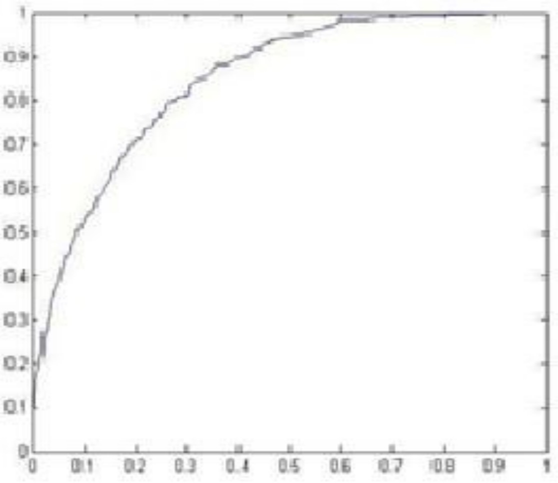
We implemented Adaptive Neuro Fuzzy Inference device for software illness prediction. In MAT lab, there may be a toolbox Neuro Fuzzy device field which implements ANFIS. In the information section, education information turned into loaded. In FIS section, preliminary Sugeno Fuzzy Inference system was derived with the aid of using subtractive clustering method. Subtractive clustering approach takes 4 parameters range of impact (0.3), Squash thing (1.25) accept ratio (0.5) & Reject Ratio (0.15). In training segment, FIS became educated the usage of ANFIS set of rules. In trying out segment, FIS became tested with unseen records. The Receiver Operating Traits (ROC) was plotted in opposition to true fine charge with false wonderful price. AuC values are decided for every dataset of software illness prediction and consequences are in comparison with price sensitive neural networks. The performance of the classifier is measured in terms of location beneath ROC Curve (AuC) values

for imbalanced datasets. In table 1 we compared the effects of fee sensitive neural networks (considering different parameter values) with ANFIS.



Pc1 dataset

For all datasets, the overall performance of ANFIS found excellent. ROC curve are generated via plotting fake high quality charge against real positive price. Figures 2-5 illustrates these ROC curves on diverse SDP datasets.



Kc1dataset

III. CONCLUSION

There are numerous techniques for predicting faulty prone modules the use of facts mining like selection timber, Random forests, guide Vector Machines and Neural Networks. In this paper, we proposed Adaptive Neuro Fuzzy Inference System (ANFIS) for figuring out defective inclined modules. ANFIS technique generates sugeno fuzzy model. Initially FIS changed into generated by way of subtractive clustering technique and it turned into skilled by means of ANFIS. We carried out cost touchy neural Networks on SDP datasets with exclusive parameters and the consequences are compared with ANFIS. The performance is measured in terms of AuC values and located first-rate results with ANFIS.

IV. REFERENCES

1. Omer Faruk Arar, Kursat Ayan,” Software defect prediction using cost-sensitive neural network" Applied Soft computing, April 2015.
2. Jun Zheng," Cost-sensitive boosting neural networks for software defect prediction", Expert Systems with Applications.2010 June; 37(6), 4537- 4543.
3. Ezgi Erturk, Ebru Akcapinar Sezer,"A comparison of some soft computing methods or software fault prediction", Expert systems with applications.2014
4. J.Nam, S.J.Pan, and S.Kim,"Transfer defect learning", In Proceedings of the 2013 International Conference on Software Engineering.ICSE '13, 382-391.
5. Song Q, Jia Z, Shepperd M, Ying S, Lin J, "A general software defect proneness prediction framework", IEEE Transations on Software Engineering.2011;37,356-370.
6. S.Shivaji, E.Whitehead, R.Akella, and S.Kim."Reducing features to improve code change-based bug prediction", Software Engineering.IEEE Transactions on Systems.2013; 39(4), 552-569.
7. K O.Elish, M O.Elish, "Predicting defect prone software modules using support vector machines", Journal of systems & Softwares.2008; 649-660.