DEFECT DETECTION MODEL FOR UNCERTAIN DATA BY A NOVEL ENSEMBLE LEARNING

E. Sreedevi[1],V. PremaLatha[2] ,Y.Prasanth[3],V.UdayKumar[4]
[1, 2,4]Asst. Professor, Department of C.S.E, K.L.E.F,Vaddeswaram,Guntur, A.P
[3]Professor,Department of C.S.E, K.L.E.F, Vaddeswaram, Guntur, A.P

Abstract- To detect defect in uncertain dataset which becomes a challenging issue,as the software product increases in terms of size and its complexity. To determine the success of a software productivity we need to evaluate two quality keys features as the software metrics and defect prediction.metric relationships and probabilistic estimators are used to focus on defect prediction for Traditional classification models. Issues related like McCabes versus Halstead lines of code counts for predicting software defects have been explored in traditional models.This paper we explain about a applied novel multi-learner model which is ensembled to predict software metrics using multilearners.The defect on the PROMISE defect dataset are effectively predicted and classified using these models.When compared to the traditional learning models it shows that ensemble model effectively optimizes the true positive rate.

Keywords- Defect detection, ensemble model, classification, PROMISE

I. INTRODUCTION

The defect is a foible in the software program which is the source of failing it to perform its functions. to find the vulnerabilities in the SDLC phases which occurs due to manual or automatic errors are found with the help of an optimized way provided by the Defect prediction.

In present era software quality is becoming more and more essential, As the dependency of software programs are increasing. A customer may turn out to dissatisfaction for software defects such as failures and faults which may affect the quality of software. To assess or evaluate a Software Project or Process in order to gain up with continuous or nominal features they include the tools or processes of Software Metrics and Computations [1]. To find the internal behaviour of the metrics, performance, quality and reliability, most of the software metrics usually use correlations or similarity measures. It is too burdensome to produce a quality end product due to the increasing of software constraints and complexity metrics.

II. RELATED WORK

To find the stochastic process in terms of defect variables and find the interval between the variable rate the defect prediction models are formulated [1][2].To formulate the number of defects that occur during the defect dependency test they formulated by using non-homogeneous poison process. Find the poison process P(t) for each defect, the probability of finding k defects by the time t and it is expressed in terms of the Poisson distribution with mean m (t)as

$$\text{Prob}\ (\text{P}\ (\text{t}) = \text{k}) = \text{m}\ (\text{t})\ \text{n}.\ e^{-mt} / n!$$

To find the defect distribution in the testing phase an exponential model is used in the SDLC mostly in the phases of regression testing and integrated testing.Defect can occur at any stage during the testing phase or failure mode, this is the basic assumption and it is the best indication of reliability of its software.

$$F(t) = (k+1)\ (\lambda . e^{-\lambda t})$$

To predict the presence of defects based on the training samples Naïve Bayes is a very effective classification technique. In a Naïve Bayes model, a bug prediction is examined as a binary classifier which analyses the historical metric data by training and predicting the predictor. If the attribute types in the metric data are mixed type, then it will be challenging to predict the defects due to missing values or uncertain data.

2.1 Neural NetworkFramework

To predict attacks from defect dataset,A feed forward neural network model is proposed. This model is embedded in three levels, which is an input level, a non-linear hidden level and the linear output level. Let the set of training defect data instances are represented as X = {ins1, ins2, ins3, ….insm} and the defect prediction is Y = {atk1, atk2, atk3, ….atkn} as shown in Fig. 1. The Gaussian function is used as basis functionas

$$Gau\ (ins_i, c_i) = exp\ (-|\ ins_j - c_i\ |/2\sigma_i^2)$$

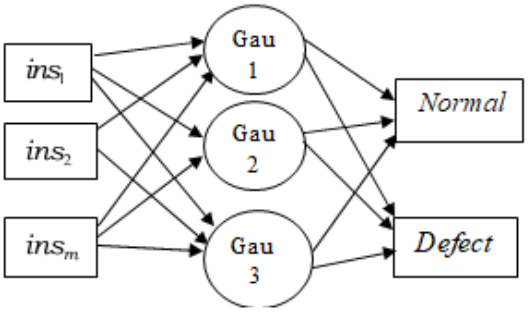


Figure 1.Nonlinear Neural Network Structure

2.2 Metric SelectionMeasures

A well-known measure in information theory, i.e. entropy is used to find the impurity of the random samples in order to haft the uncertain data effectively using the information gain correctly[7]. Entropy assesses the impurity level in a group of samples. The evaluation of information gain and gain ratio can be derived as follows:*n*

$$Entropy = \sum_{i=0}^{u} - p_i \log (p_i)$$

The Trained instance is partitioned accordingly to measure metrics in order to determine the effectiveness of split metric in decision tree construction that uses to gain information[7][5]. The gain measure of an attribute A, relative to a collection of instances I, is definedas,

$$Gain\ (I, A) = Entropy\ (I) - \sum_{i \in Values(A)} \frac{|I_v|}{|I|} Entropy\ (I)_v$$

Set of all distinct values for an metric are indicated by value (A) , and Iv is the subset of I for which the metric a has instance value v. The gain ratio of a metric A is given as:

$$GainRatio = \frac{Gain(I, A)}{SplitInfo(I, A)}$$

Pruning method is used to increase the metric relationships efficiency in order to overcome the uncertain problem, especially in noisy or uncertain data.Few of the stopping conditions used in the literature are:

- Horizon the class homogeneity
- Horizon the minimum number of instances for a non- terminal node. Ex: Range [2,230] instances
- Horizon a maximum tree depth. Ex: Range: [2,9] levels
- Horizon the predictive accuracy threshold within a node. Range: {75%, 89%}.

Decision tree models that are ensembled namely CART,C4.5, Bayesian tree and random forest for defect analysis are developed classification models using Naïve Bayes [8].Defect prediction with large number of attributes set are concluded that to handle uncertainty there exists no single traditional model.

For discovering the defected features of the software product, a machine learning or statistical approach can be used to perform for the software defect prediction. DecisionTree, RandomForest, Naïve Bayesian , SVM with logistic regression etc of software defect detection are using a large number of statistical models and machine learning which we can use the in many of our recent studies where SVM that includes Chi-square ,RBF and log polynomial functions using non-linear kernel[9][10].

III. Selection Based Ensemble Model Feature

To address the problem of class imbalance and software defect prediction is proposed classifiers which is an ensembled in this paper. Feature selection and probability estimation functions are the basic features of this model. a group of trained classifiers are generated which is the process of learning models in Ensemble learning.

High true positive rate and eliminates data imbalance issue are used to predict that are used in learning modelmultiple learners are trained which are Ensemble learning to solve the unbalanced problem. Homogeneous and heterogeneous approaches are the two types of Ensemble learners that are classified algorithm differently developed by each base learner model. The multi-learning ensemble model basic framework is shown in Fig 2.

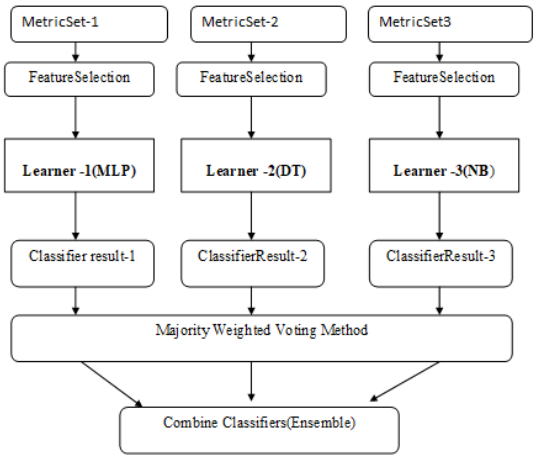


Figure 2.Ensemble Model

Model learning and Model applying are the two phases of basic framework. Ensemble E including base classifiers is generated in the learning part. Entire training data is divided into subsets and then to generate one base classifier to each of them in the process of ensemble. The class defect predicators of the base classifiers is integrated using H in the applying phase to form the final classifier of the ensemble model An Ensemble is turns out to be more efficient than anyofthe base classifiers for defect prediction.

H*=F(h1,h2..hs)

Or

$$H(x_i) = sign(\sum_{i=1}^{N} w_i h_i(x_{ii})$$

Where w_i is the weight of the selected classifier h_i

3.1 Selection Feature Function

With respect to the defect class distribution, Chi-square statistic is computed that can evaluate the defect for Chi-square based defect's selection. To find the difference between the observed defect distribution to the actual defect distribution a non- parametric statistical approach used. To improve the software defect detection Many feature selection algorithms have been ushered such as Decision induction tree, correlation-based classifier, Bayesian probabilistic method and ensemble decisiontree.

Ensemble Based Defect Detection Model Algorithm:

Input: Defect datasets as D List;
Output: Classified defect results;
Procedure:
// Ensemble Selection
For each metric set Data[i] in Dlist
Do
For each attribute Aj in Data[i]
Do
For each instance I(Aj) in Aj
do
If(Typeof(I(Aj))==Numerical&&Aj==Null) then
I(Aj)= (Max(D1(Aj),D2(Aj))/(m - S.D(D1, D 2));
End if
If(Typeof(I(A_j)==Nominal &&A_j=Null) then
Find conditional posterior probability estimation as
P1 (vi) = log(Prob(vi / Cm)) + log(Prob(Cm))
P2 (vj) = log(Prob(vj / Cm)) + log(Prob(Cm))

$P_1(V_j)$ is associated with classifier-1 and

$P_2(V_j)$ is associated with classifier-2.
A j = maxprob({P1 (Vi)},{P2 (V j)});

$$Value = (Chi - Square)\chi^2$$

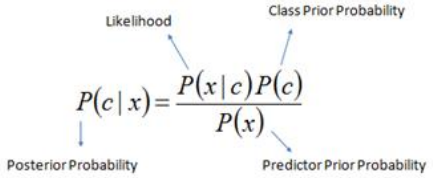
$$= \sum_{k=1}^{n} (Obser_k - Expect_k)^2 / Expect_k$$

if(maxprob ϵ $P_1(v_i)$ && Value > threshold) then
$I(A_j)=V_i$
else
$I(A_j)=V_j$
End if
Done
Done

//Ensemble Applying
Input: Ranked Features Data as FData;
Output: Model Learning and Defect Detection
Procedure: Read defect data feature set as FData
For each Feature FData[i] in FData
Do
For each instance $I(A_i)$ in A_i do
Do
Divide the data instances of $FA(D_i)$ into 'k' independent sets. Select classifier $C_{i/i=1\ldots m}$
While i<k
Do
if($C_i$is MLP) Then

$$I_j = \sum_i w_{ij} O_i + \theta_j$$

else if(C_i is NB) then



$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \cdots \times P(x_n|c) \times P(c)$$

else if(C_i is Decision tree)
end if
end while
Calculate misclassified error rate and statistical true positive rate
Done
Done

Defect datasets are taken as input in the above algorithm to fill the missing values in the continuous and nominal attributes. Extract phase based decision patterns for decision making an ensemble classifier is used to Filtered phase datasets are that are processed .Training samples are based on existence of defects to predict the classification technique which is very effective in Naïve bayes. Abinary classifier is considered as bug prediction in naïve Bayesmodel by analyzing historical metric data

That is trained and predicted in this model. It is challenging to predict the defects due to missing values or uncertain data if there are attributes of metric data that are mixed type. Decision tree pruning: Pruning is needed to eliminate over fitting tree in the decision tree. pre-pruning and post-pruning are the two main strategies. when constructed the decision tree, Pre- pruning tree is adding a stop criterion α. When the impurity level caused by

the separation by one node decreases less than α, we can see this node as a leaf node, and stop the construction of the tree. It is very important on how to choose the value of α, attributes when the precise will be not so high, if α is too large, this will cause the stop of splitting and the constructed decision tree has no big difference from the originalone if α is too small.

IV. Experimental Results AndDiscussions

4.1 Datasets

The software metrics for this case study were gathered from the NASA (PROMISE) metrics data program repository. PROSIME repository consists of eight defect datasets which includes at least 45 software metrics, including 25 product metrics ,15 process metrics and 5 execution metrics. We investigated four different ensembles of feature selection techniques with different ranking functions such as IG,MI,and Chi-square in our experiment

4.2 Execution Result

Randomized Rules: 339
Time taken to run Proposed Model: 0.31 seconds
=== Accuracy By Class ===

TP Rate	FPRate	Class
0.971	0.086	Y
0.98	0.585	N

==========

Accuracy 0.9753 0.3355

Table 1. Accuracy Comparison of Ensemble Model with other Classification Models

Algorithm	KC1	CM1	JM1	MC2	PC4	MW1	PC1	PC3
Ensemble Model	**0.9700**	**0.9400**	**0.9600**	**0.9720**	**0.9680**	**0.9650**	**0.9710**	**0.9680**
SVM	**0.9200**	**0.9190**	**0.9450**	**0.9670**	**0.9250**	**0.9310**	**0.9600**	**0.9500**
DT	**0.8700**	**0.8167**	**0.8960**	**0.9100**	**0.8780**	**0.9320**	**0.8560**	**0.8870**
Naïve Bayes	**0.9110**	**0.8970**	**0.8850**	**0.8545**	**0.8320**	**0.9420**	**0.9109**	**0.9100**

Table 1, describes the traditional classification models in terms of true positive rate in the accuracy comparison n of the ensemble model. From the table it is noticed that the ensemble model has high accuracy rate compared to traditional models. Figure 3, describes the accuracy comparison of the ensemble model with the traditional classification models in terms of true positive rate. From the table it shows that the ensemble model has high accuracy rate compared to traditionalmodels.

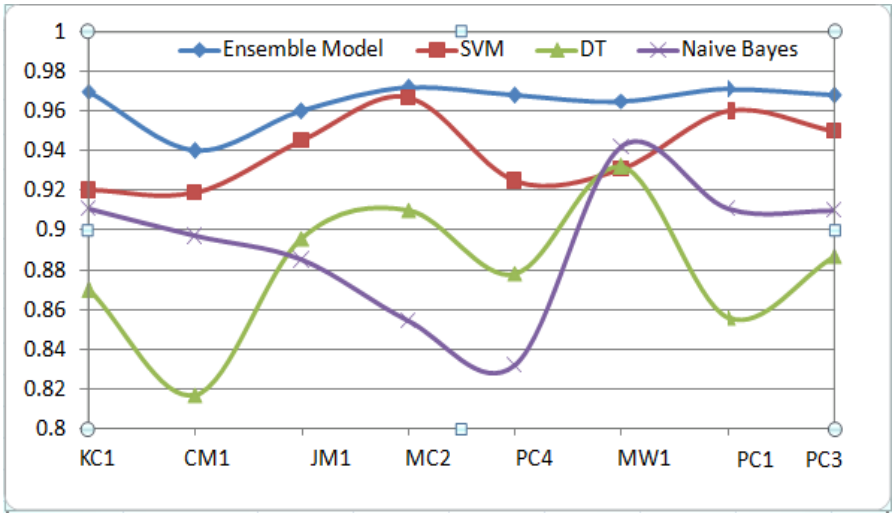


Figure 3. Accuracy Comparison Of Ensemble Model With Other Classification Models

V. CONCLUSION

An efficient way has been applied an ensemble model along with multiple learners to detect the defects in this paper. To predict software metrics using multi-learners it uses novel multi-learner ensemble model. This model effectively predicts and classify the defects on the PROMISE defect dataset. Experimental results pageant that ensemble model effectively optimizes the true positive rate compared to traditional learning models. In future this work can be extended to optimize the defect detection process using the dynamic optimization ensemblemodel.

REFERENCES

[1] T. Zimmermann and N. Nagappan, "Predicting defects with program dependencies," *Empirical Software Engineering and Measurement (ESEM 2009) 3rd International Symposium*, pp. 435- 438, 2009.

[2] E. Ceylan, F. Kutlubay, and A. Bener, "Software defect identification using machine learning techniques", *Software Engineering and Advanced Applications, (SEAA '06) 32nd EUROMICRO Conference*, pp.240-247, 2006.

[3] Y. Liu, T. M. Khoshgoftaar, and N. Seliya, "Evolutionary optimization of software quality modeling with multiple repositories", *IEEE Tranactions of. Software Engging*, Vol. 36, No. 6, pp. 852-864, 2010.

[4] D. J. Drown, T. M. Khoshgoftaar, and N. Seliya, "Evolutionary sampling and software quality modeling of high-assurance systems", *IEEE Transactions on Systems Man and Cybernetics Part A: Systems and Humans*, Vol.39, No.5, pp.1097-1107, 2009.

[5] A. Shanthin, and R. M. Chandrasekaran, "Applying Machine Learning for Fault Prediction Using Software Metrics," *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 2, 2012.

[6] M. Jureczko, and L. Madeyski, "Towards identifying software project clusters with regard to defect prediction," *Proceedings of the 6th International Conference on Predictive Models in Software Engineering*, pp. 9, 2010.

[7] D. Gray, D. Bowes, N. Davey, Y. Sun, and B. Christianson, "Software defect prediction using static code metrics underestimates defect proneness," *Neural Networks (IJCNN), The 2010 International Joint Conference*, pp. 1-7, 2010.

[8] T. Menzies, "The PROMISE software engineering Repository, University of Ottawa, Faculty of Engineering," *Available: promise.site.uottawa.ca/SERepository/datasetspage. html*, 2012.

[9] J. Nam, S. J. Pan, and S. Kim, "Transfer defect learning", *35th International Conference on Software Engineering*, pp. 382-391, 2013.

[10] M. D'Ambros, M. Lanza, and R. Robbes, "An extensive comparison of bug prediction approaches", *7th IEEE Work. Conf. Min. Softw. Repos. (MSR 2010)*, pp. 31-41, 2010.