March-April 2020
ISSN: 0193-4120 Page No. 6035 - 6041

Hybrid Firefly based Software Defect Prediction on Imbalanced Data

[1] Dr. S. A. Sahaaya Arul Mary, *Professor and Head,Computer Science and Engineering Saranathan College of Engineering, Trichy, Tamilnadu, India.*

.

[2] C. Shyamala, *Assistant Professor, Dept. of Computer Science and Engineering, Saranathan College of Engineering, Trichy, Tamilnadu, India.*
Email: samjessi@gmail.com.

Article Info
Volume 83
Page Number: 6035 - 6041
Publication Issue:
March - April 2020

Article History
Article Received*: 24 July 2019*
Revised*: 12 September 2019*
Accepted*: 15 February 2020*
Publication*: 01 April 2020*

Abstract:

Increasing complexity of software systems has led to the increased necessity for analyzing and testing these software for their efficient functioning. This work proposes a meta-heuristic based software defect prediction model for faster and better performance. The proposed model is composed of two major stages; the feature selection stage and the defect prediction stage. The feature selection stage uses Cuckoo Search, a metaheuristic classifier model as the search method to identify the features that are mandatory. The defect prediction stage uses the Hybridized Firefly model to identify defects in the software. Experiments with state-of-the-art models from literature indicates enhanced performances, exhibiting the efficiency of the proposed model.

Keywords: Software Defect Prediction; Cuckoo Search; Firefly Algorithm; Simulated Annealing; Feature Selection

I. INTRODUCTION

Software defect prediction is the process of identifying defects in software modules to enable faster and more efficient testing process. Prior identification of prospective defective modules leads to better analysis of the identified modules. Testing can be concentrated on these defective modules [1], hence providing better focus on the defective modules. Such indications can also be highly useful for guiding the code review process, hence providing better quality assurance for the software being analyzed.

Defect prediction in software is a complex task, mainly due to the large number of inconsistencies involved in the process and the fact that software systems are large-scale and complex. Further, another major issue in this field is that sufficient training data pertaining to the required module might not be available. Hence it is always a practice to perform cross-project prediction [2]. Cross-project prediction is the process of using log records from past data to perform training and to create a trained model. This trained model is used tao predict defects for the current software system.

Software defect prediction is a supervised learning problem, which has several major issues that has to addressed for effective prediction. The first and the major issue is the presence of data imbalance. Log records pertaining to defect prediction is composed of several records depicting the non-defective cases and a very few instances depicting the defective cases. This leads to an imbalance in the data, hence affecting the prediction process. Another vital issue is the complex relationships existing between

Published by: The Mattingley Publishing Co., Inc.

attributes and faults [3,4]. Since log records are used as the base data, it could be observed that the records contain huge number of attributes, hence leading to huge data sizes. However, on analysis it could be revealed that not all attributes contribute to the prediction process. Elimination or appropriate handling of such data is mandatory. The final issue is the absence of standard measures for analysis [5]. This work presents a hybridized Firefly based prediction model and Cuckoosearch based Feature Selection model that aims to address most of the issues to provide enhanced predictions.

The remainder of this paper is structured as follows; section II presents a review of the literature, section 3 presents a detailed view of the proposed architecture, section 4 presents the results and discussions and section 5 concludes the work.

II.RELATED WORKS

Defect prediction is an emerging domain due to the increased use of software products. This section discusses some of the recent and prominent contributions in this domain.

An Artificial Neural Network based approach for software defect prediction, HyGRAR was proposed by Miholca et al. in [6]. This model proposes a hybrid combination of gradual relational association rule mining and artificial neural networks to predict defective software modules from non-defective modules. The HyGRAR model is an extension of the GRAR model proposed by Czibula et al. [7]. This model extends the association rules using fuzzy relations for effective defect predictions. A cross-project model-based software defect prediction model was proposed by Yu et al. [8]. The model operates based on Binary Logistic Regression and is considered to be a self-assessment model. A similar model, using multiple linear regression and Genetic Algorithm was proposed by Afzal et al. [9]. A three way decision framework for software defect prediction was proposed by Li et al. [15]. This is a cost-sensitive decision making framework, based on the widely used two-stage classification model.

This domain was also explored in an unsupervised learning context. A self-organizing maps based model proposed by Abaej et al. [10] performs grouping by creating weighted vectors and threshold based labeling. Other studies using clustering includes K-means clustering based model proposed by Bishnu et al. [11], which uses quad trees to identify the cluster centers and a similar K-means clustering based model by Varade et al. [12], which uses hyper-quad trees to initialize cluster centers. An X-means clustering model for defect prediction was proposed by Catal et al. [13]. This model identifies clusters and uses threshold values for software metrics to determine if the clusters are defective or non-defective. Other similar model includes feature-selection based defect prediction models by Park et al. [14].

III.HYBRID FIREFLY BASED SOFTWARE DEFECT PREDICTION

Defect prediction in software is one of the major activities that can aid in reduced testing costs and in-turn overall software cost. This paper presents a metaheuristic model-based software defect prediction model that aids in faster and more accurate prediction of defects. The proposed hybrid Firefly architecture is composed of two major phases; Cuckoo Search based Feature Selection module and the defect prediction module using the Hybrid Firefly classifier. Workflow of the proposed architecture is shown in figure 1.

3.1 **Cuckoo Search based Feature Selection**

Defect prediction in software is usually performed on the log files pertaining to programming. These files contain all the information related to the software being constructed, hence is bound to be large with huge number of attributes. Although most of these attributes are representative of the software's state and do not contribute in terms of predicting defects, it is mandatory to record them for logging purposes. Hence analyzing the attributes and identifying the best set of attributes can result in providing better predictions and can also enable

March-April 2020
ISSN: 0193-4120 Page No. 6035 - 6041

huge reduction in cost.

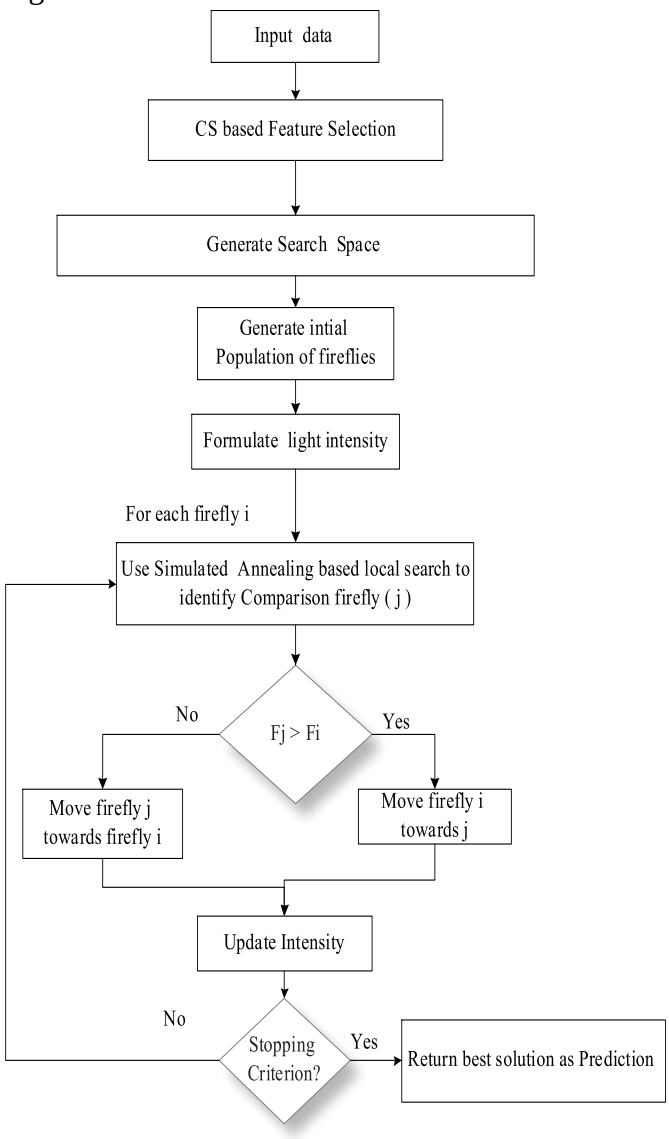

Figure 1: Hybrid Firefly Architecture for Defect Prediction

Feature selection techniques in-general can be categorized as filters and wrappers. Wrapper based feature selection models perform feature reduction using learning algorithms, while filter based feature selection models uses correlation between features as a major criterion for selecting or eliminating features. The proposed model uses wrapper-based feature selection method, with CFS based subset evaluator as the attribute evaluation model and Cuckoo Search as the search method.

CFS Subset evaluator [16] operates by selecting a subset of the best features and hence subsequently identifying the best subset for features. A subset with high class correlation and low inter-correlation is used as the selection criterion.

A feature is said to be relevant iff there exists some v_i and c for which $p(Vi = vi) > 0$ such that

$$p(C = c|V_i = v_i) \neq p(C = c)$$

If the correlation between the components are known, and the inter-correlation between is provided, then the correlation can be predicted by

$$r_{zc} = \frac{k\overline{r_{zi}}}{\sqrt{k + k(k-1)\overline{r_{ii}}}},$$

Where r_{zc} is the correlation between the summed components and the outside variable, k is the number of components, ($\overline{r_{zi}}$) is the average of the correlations between the components and the outside variable, and ($\overline{r_{ii}}$) is the average inter-correlation between components [17, 18, 19].

Cuckoo search, being a prediction model has been incorporated as the search mechanism, while the subset evaluation is being performed by the CFS subset evaluator. This results in a sub-set of the log data that can effectively provide better predictions and reduced computation complexity.

3.2 Hybrid Firefly for Defect Prediction

Firefly algorithm [20] is a swarm based model, operating based on the behaviour of fireflies. Movement of Fireflies is determined by the brightness or light intensity of the fireflies in the swarm. Light intensity of the Fireflies is considered as the fitness with which the swarm movement is performed.

The major assumptions of Firefly algorithm are as follows:

- A firefly has the probability to get attracted to any other Firefly, and all the Fireflies are considered unisexual
- Level of attraction exhibited by a Firefly is directly proportional to its light intensity or brightness
- All Fireflies are attracted towards the brightest Firefly

Published by: The Mattingley Publishing Co., Inc.

March-April 2020
ISSN: 0193-4120 Page No. 6035 - 6041

- Brightness of a Firefly is determined by its distance
- If no firefly is brighter than a given firefly, then it moves randomly

The process of Hybrid Firefly based defect prediction is performed in three major phases; the search space creation phase, Firefly distribution phase and the Firefly movement phase using hybridized local search model.

1.1.1. Search Space Creation using Log Data

The software defect log records obtained after feature selection is used as the base for creating the search space. Every node is considered as a solution and forms the search space. The number of features contained in the log records determines the dimension of the search space. Along with the training data, a single instance from test data is also added to the search space. After every convergence, the test data is changed to the next instance and the same process is repeated. Hence every iteration converges with predicting a final class for the test instance.

1.1.2. Firefly Distribution

Search space creation is followed by distributing fireflies on the nodes in search space. Since the problem under analysis is a classification problem, all the fireflies are distributed on the test instance. Firefly movement begins from the test data and the final location of Fireflies is used to determine the prediction for the test data.

1.1.3. Firefly Movement using Hybridized Local Search

Local search is the process in which the Fireflies are moved towards the best available solution. Movement of fireflies is determined by the fitness function. Light intensity/ brightness of a firefly is considered as the fitness function. Fitness is determined by the distance criterion and is given by,

$$Intensity_i = 1 \Big/ \sqrt{\sum_{j=1}^{attr} (X_{test,j} - X_{i,j})^2}$$

Where $X_{test,j}$ refers to the j^{th} attribute of the test data and $X_{i,j}$refers to the j^{th}attribute of the firefly i.

Hybrid Firefly Algorithm

1. Input Log Data (D) for Defect Prediction
2. Apply Cuckoo Search based Feature selection to obtain pre-processed data (D')
3. Generate Search Space using D'
4. Generate initial population of Fireflies
5. Distribute Fireflies on the test instance to be predicted
6. Initiate random Firefly movement
7. Formulate light intensity of Fireflies using Eq.1
8. For each Firefly i
a. Use Simulated Annealing based local search to identify the best Firefly j
b. Compare the intensities of Firefly i to j
c. Initiate movement depending on the firefly with best intensity
d. Update intensity of Firefly
9. If stopping criterion not reached goto step 8
10. Identify the node n with maximum firefly concentration
11. Provide Prediction based on the selected node n

Firefly movement is initiated with each Firefly comparing its intensity with the intensity of the all the other Fireflies. However, since all the Fireflies are distributed on the test data, an initial random movement is initiated. The movement is initiated by identifying the most attractive firefly within the vicinity of the current Firefly. The actual Firefly algorithm compares each firefly with every other firefly prior to movement. However, it is computationally intensive. Hence the proposedmodel hybridizes this local search approach using Simulated Annealing.

Simulated Annealing (SA) is a metaheuristic-based model that operates by identifying the global

Published by: The Mattingley Publishing Co., Inc.

March-April 2020
ISSN: 0193-4120 Page No. 6035 - 6041

optimum of a function based on heating and controlled cooling of materials. The major advantage of Simulated Annealing is that it has the ability to operate on very large search spaces to provide optimal results.

The proposed model incorporates Simulated Annealing as a local search process to reduce the computational requirements of the Firefly local search mechanism. Intensity levels of all the Fireflies are passed to the Simulated Annealing module. The intensity levels are analyzed and the Firefly with best intensity is returned by the Simulated Annealing module. This results in all the other Fireflies moving towards the selected Firefly and the selected Firefly moves in a random direction.

This process is repeated until the swarm reaches defined number of iterations or until the swarm experiences stagnation. Stagnation is the process in which all the Fireflies converge into a single best solution and hence they exhibit no movement. The converged point in identified as the prediction for the test data.

IV.RESULTS AND DISCUSSION

Experiments were conducted on the Promise data (cm1), a huge corpus of data taken from spacecraft instrument management from NASA. It contains 22 attributes (21 features+1 class) and 498 instances. The data exhibits an imbalance ratio of 9.3. Comparisons were performed with the three-way decision-based software defect prediction model (3WD) [15] and Cuckoo search and BFO based model (CS-FS & CS-BFO) [21].

ROC curve of the proposed model and the CS-FS & CS-BFO model is shown in figure. It could be observed that the ROC curve of the proposed model exhibits higher surface area compared to the ROC curve of the compared model. Higher surface area in ROC curve refers to better prediction levels, hence superiority of a classifier. This shows the enhanced predictability of the proposed model.

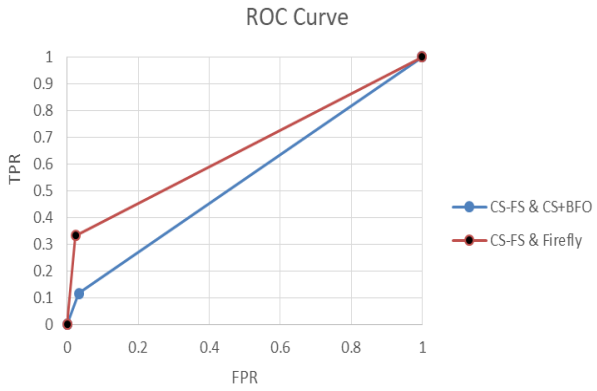

Figure 2: ROC Curve

A comparison of the accuracy levels of the proposed model with the 3WD model and CS-FS&CS-BFO model is shown in figure. It could be observed that the proposed model exhibits higher accuracy levels compared to the models proposed in literature. The proposed model was observed to exhibit 2%-8% improved accuracy levels compared to the other models.

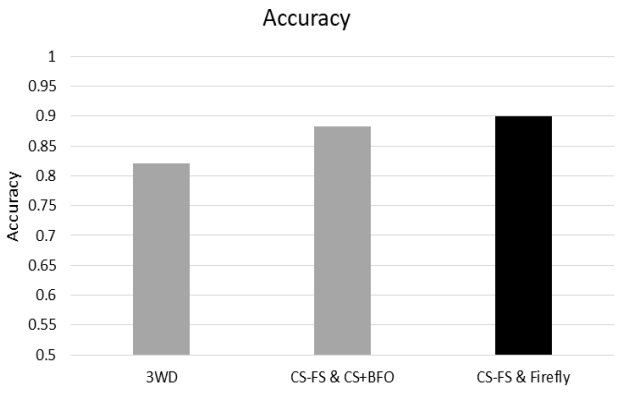

Figure 3: Accuracy

A comparison of the F-Measure levels of the proposed model with the CS-FS & CS-BFO model is shown in figure. It could be observed that the proposed model exhibits 28% improved F-Measure levels compared to the existing model in literature.

Published by: The Mattingley Publishing Co., Inc.

This exhibits the high prediction efficiency of the proposed model.

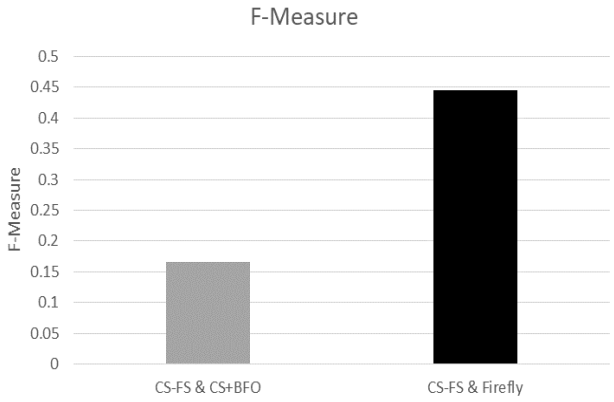

Figure 4: F - Measure

A tabulated view of the performance metrics is shown in table 1. It could be observed that the proposed model exhibits better performances in terms of all the metrics, hence exhibiting the efficiency of the proposed model.

Table 1: Performance Metrics

	CS-FS & CS+BFO	CS-FS & Firefly
FPR	0.032467532	0.022727273
TPR	0.117647059	0.333333333
Recall	0.117647059	0.333333333
Precision	0.285714286	0.666666667
Accuracy	0.883040936	0.9
F-Measure	0.166666667	0.444444444
TNR	0.967532468	0.977272727
FNR	0.882352941	0.666666667

4 CONCLUSION

Defect prediction in software systems is of vital significance due to the increased usage of software systems in all areas. Effectively identifying defects can lead to a huge reduction in the cost involved in developing the software, which in turn can be beneficial to the customers. This work proposes a hybrid Firefly based defect prediction model, composed of two major stages; the feature selection stage and the defect prediction stage. The feature selection stage uses Cuckoo Search as the search mechanism and CFS as the subset evaluator. The defect prediction module uses Hybridized Firefly algorithm with enhance local search, hybridized using Simulated Annealing. Experiments were performed on the NASA dataset and results indicate improve accuracy levels at 2%-8% and improved F-Measure levels of upto 28% when compared to existing models in literature. Future works will concentrate on improving the predictions and reducing the computational complexity of the model.

REFERENCES

[1] Chang, R. H., X. D. Mu, and Li Zhang.2011"Software defect prediction using non-negative matrix factorization." Journal of Software Volume 6 No 11 2114-2120.

[2] Panichella, R. Oliveto, A. D. Lucia, 2014Cross-project defect prediction models: L'union fait la force, in: IEEE Conference on Software Maintenance, Reengineering and Reverse Engineering (CSMR-WCRE), 2014 Software Evolution Week, 164–173.,

[3] Arora, Ishani, Vivek Tetarwal, and Anju Saha.2015 "Open issues in software defect prediction." Procedia Computer Science Volume 46: 906-912.

[4] Sandhu PS, Brar AS, Goel R, Kaur J, Anand S. A 2010model for early prediction of faults in software systems. In: 2nd International Conference on Computer and Automation Engineering. Singapore; 281-285..

[5] Gray D, Bowes D, Davey N, Sun Y, Christianson B. 2011Further thoughts on precision. In: 15th Annual Conference on Evaluation & Assessment in Software Engineering. Durham;. . 129-133.

[6] Miholca, Diana-Lucia, Gabriela Czibula, and Istvan Gergely Czibula. 2018"A novel approach for software defect prediction through

March-April 2020
ISSN: 0193-4120 Page No. 6035 - 6041

hybridizing gradual relational association rules with artificial neural networks." Information Sciences Volume 441 152-170.

[7] G. Czibula, I. G. Czibula, D.-L. Miholca, 2017Enhancing relational association rules with gradualness, International Journal of Innovative Computing, Communication and Control, Volume 13 No 1 289–305.

[8] L. Yu, A. Mishra, 2012Experience in predicting fault-prone software modules using complexity metrics, Quality Technology & Quantitative Management Volume 9 No 4, 421–433.

[9] W. Afzal, R. Torkar, R. Feldt,2012 Resampling methods in software quality classification, International Journal of Software Engineering and Knowledge Engineering Volume 22 No 2 203–223.

[10] G. Abaei, Z. Rezaei, A. Selamat, 2013Fault prediction by utilizing self-organizing map and threshold, in: 2013 IEEE International Conference on Control System, Computing and Engineering (ICCSCE), 465–470.

[11] P. Bishnu, V. Bhattacherjee, 2012Software fault prediction using quad tree-based k-means clustering algorithm, IEEE Transactions on Knowledge and Data Engineering Volume 24 No 6 1146–1150.

[12] S. Varade, M. Ingle, 2013Hyper-quad-tree based k-means clustering algorithm for fault prediction, International Journal of Computer Applications Volume 76 No 5, 6–10.

[13] C. Catal, U. Sevim, B. Diri, 2009Software fault prediction of unlabeled program modules, in: Proceedings of the World Congress on Engineering (WCE), 212–217.

[14] M. Park, E. Hong, 2014Software fault prediction model using clustering algorithms determining the number of clusters automatically, International Journal of Software Engineering and Its Applications Volume 8 No 7 199–205.

[15] Li, W., Huang, Z., and Li, Q., 2016 'Three-way decisions based software defect prediction.', Knowledge-Based Systems, Volume 91, 263-274.

[16] Miholca, Diana-Lucia, Gabriela Czibula, and Istvan Gergely Czibula. 2018"A novel approach for software defect prediction through hybridizing gradual relational association rules with artificial neural networks." Information Sciences Volume 441 152-170.

[17] Loo, Robert.2002 "A caveat on using single-item versus multiple-item scales." Journal of managerial psychology Volume 17 No 1 68-75.

[18] Hogarth, Robert M. 1977"Methods for aggregating opinions." Decision making and change in human affairs. Springer, Dordrecht,. 231-255.

[19] Zajonic R.B. 1962. "A note on group judgements and group size." Human Relations, Volume 15:177–180,

[20] Johari, N. F., Zain, A. M., Noorfa, M. H., & Udin, A.2013 Firefly algorithm for optimization problem. In Applied Mechanics and Materials. Trans Tech Publications. Vol. 421, 512-517

[21] Keerthi, S. S., & Lin, C. J. 2003Asymptotic behaviors of support vector machines with Gaussian kernel. Neural computation, Volume 15 No 7, 1667-1689

Published by: The Mattingley Publishing Co., Inc.