Software Fault Prediction using Bio-Inspired Algorithms to Select the Features to be employed: An Empirical Study

Asad Ali
University of Salerno
Fisciano, Italy *abasyn.asad@gmail.com*

Carmine Gravino
University of Salerno
Fisciano, Italy *gravino@unisa.it*

Abstract

In recent past, the use of bio-inspired algorithms got a significant attention in software fault predictions, where they can be used to select the most relevant features for a dataset aiming to increase the prediction accuracy of estimation techniques. The most-earlier and widely investigated algorithms are Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). More recently, researchers have analyzed other algorithms inspired from nature. In this paper, we consider GA and PSO as baseline/benchmark algorithms and evaluate their performances against seven recently-employed bio-inspired algorithms and meta-heuristics, namely Ant Colony Optimization, Bat Search, Bee Search, Cuckoo Search, Harmony Search, Multi-Objective Evolutionary Algorithm, and Tabu Search, for feature selection in software fault prediction. We present experiments with seven open source datasets and three estimation techniques: Random Forest, Support Vector Regression, and Linear Regression. We found that it is not always true that the recently introduced algorithms outperform the earlier introduced algorithms.

Keywords: Fault prediction, Feature selection algorithms, Bio-inspired algorithm.

1. Introduction

In the last decades, researchers have introduced and investigated tools and techniques to increase the accuracy of fault predictions. Among them machine learning (ML) techniques are widely used, able to learn from the past completed data and provide predictions based on that data [14]. However, not all (features of) data contribute to the accuracy of fault predictions. In fact, it can happen that some features in a dataset reduce the prediction accuracy. Hence, selecting the most relevant features of a dataset is an important and daunting task. However, feature selection (FS) algorithms are undermined in the past as shown by one of the comprehensive reviews performed by [13] about 60% studies (majority of those are studies published in 90s and early 2000) have ignored FS.

FS can significantly increase the performance of the estimation techniques, especially in datasets for Software Fault Prediction (SFP) which are considered as high-dimensional data. For instance, [6] illustrated that the accuracy can be improved or remains unchanged when about 85% of features are excluded. Similarly, [9] shows that best accuracy can be achieved from 68% to 90% of reduction of number of features.

The bio-inspired FS algorithms have been successfully investigated in variety of optimization problems, because it does not stick in the local optima due to its random behavior. The widely used bio-inspired algorithms such as GA, PSO, Ant Colony Optimization (ACO), Bat Search (BS), and Cuckoo Search (CS) are employed in the domain of software fault predictions. However, these and some other bio-inspired algorithms do not always fit in all situations and their performance (in terms of better accuracy) varies with the type and size of datasets employed, i.e., a particular bio-inspired algorithm which provides best results with one type of dataset in the context of software fault predictions may provide worst result with another type of dataset. In a recent Systematic Literature Reviews (SLR) about bio-inspired features selection in SFP [1], GA

and PSO resulted to be the earlier introduced bio-inspired (FS) algorithms that started to be investigated in late 2000. From this SLR and the studies employing bio-inspired algorithms for similar problems (e.g., software effort estimation), we can observe two findings: (a) the FS algorithms always performed better when compared with the baseline estimation techniques; (b) in majority of studies, limited number of FS algorithms are investigated with fewer number of datasets and estimation techniques. Hence, in this work we decided to evaluate the performance of variety of bio-inspired algorithms as feature selection and determine their accuracy.

1.1. Related work

While in most of the SFP studies, the bio-inspired algorithms are investigated as hyperparameter optimization (e.g., [4]) in a few studies they are employed to extract the relevant features of a dataset. For example, in [15] GA is used to select features from the dataset Nasa with the Bagging technique and the study results show that the accuracy of the fault predictions is improved using GA. In [16] the combination of GA and PSO has provided better results than baseline classifiers and the results illustrated that there is no significant difference between them. Similarly, the study in [7] describes the use of three bio-inspired algorithms (GA, BS, and ACO) with the Random Forest (RF). All these three bio-inspired algorithms have provided different results in different experiments in terms of accuracy, without the possibility of highlighting a trend. Four datasets of tera-Promise are employed. An optimized form of ACO (Fuzzy Mutual Information ACO) is used to extract the relevant features from the Nasa dataset and significantly better predictions are obtained with respect to the other employed approaches [10]. Menzies et al. [11] compare the performances of various FS algorithms such as Information Gain, Correlation-based FS, Consistency-based FS, and Relief in the SFP and conclude that there is no algorithm which came out as the best performed in all cases. In [8] the authors employ GA to select a subset of features and illustrate that GA together with the Logistic Regression performs better than extreme learning machine and different kernels of SVM (Support Vector Machine).

In majority of the previous studies employing bio-inspired algorithms for FS in SFP, we can observe: (a) no (proper) statistical tests performed; (b) classification-based estimation techniques/datasets used; (c) no comprehensive comparisons performed, i.e., a maximum of two or three bio-inspired FS algorithms are investigated.

1.2. Contribution of the paper

We want to consider the two earlier employed algorithms (GA and PSO) as benchmark (baseline) and compare their results with recently investigated bio-inspired algorithms ACO, BS, CS, Bee Search, Tabu Search (TS), Multi-Objective Evolutionary Algorithm (MOEA), and Harmony Search (HS)). We employ 7 regression-based Java open source datasets from Promise repository [2]. To build fault prediction models, we use three estimation techniques, namely Support Vector Regression (SVR), Linear Regression (LR), and RF. All the experiments are performed with Weka [7] and all the bio-inspired algorithms are embedded in a package called Metaphor Search [5].

Previously, GA has outperformed Evolutionary algorithm and the hybrid of GA outperformed GA [9]. Similarly, [10] mention that ACO provided better results than GA. Moreover, [8] show that BS provided better accuracy than GA and ACO and then, with different datasets in the same study, GA and ACO have outperformed BS. Wahono et al. [16] compare GA and PSO and claim that there is no significance difference between them. In a nutshell, from the previous researches it is not clear whether the earlier-employed bio-inspired algorithms (GA and PSO) or the recently-employed algorithms (e.g., ACO, BS) extract the most relevant features from a dataset which leads to a more accurate prediction of software faults. To this end, we have defined the following research question:

***RQ**: Which recently-employed bio-inspired FS algorithms perform better than the baseline algorithms (GA and PSO) in terms of accurately predicting software faults?*

Structure of the paper. Section 2 describes the study design, while results and discussion are presented in Section 3. Findings and suggestions for researchers are provided in Section 4. Future work section conclude the paper.

2. Study design

2.1. Datasets

We employed seven open source software fault prediction datasets which can be used for regression problems: Ant 1.7, Ivy 2.0, Jedit 4.3, Lucene 2.4, Poi 3.0, Synapse 1.2, and Velocity 1.5 from Promise repository [2]. Note that all the datasets have same number of features (21) and different number of observations (instances), ranging from 214 to 745. Velocity is considered is a smaller dataset in terms of number of observations (214), while Ant is a bigger dataset having 745 observations. Jedit is characterized by a smaller fault percentage (2.23%), while the dataset with the larger percentage of faults is Velocity (66.35%). In our empirical study, we focused on these regression-based datasets because the classification-based datasets have been widely investigated in SFP studies (the study [1]) has shown that classification-based datasets are used with bio-inspired FS algorithms in about 73% of studies). Thus, we decided to elaborate them further in our study.

2.2. Employed estimation techniques and Bio-inspired feature selection algorithms

Regarding the estimation techniques, RF is one of the leading ensemble classifiers from the flavor of decision tree, having a lower percentage of overfitting and high accuracy. SVR is from the SVM class and can be used for both linear and non-linear problems (having various kernel functions, e.g., linear, polynomial). LR which is applied to draw a linear relationship between the independent and dependent variables.

We investigated and evaluated the performance of 9 different bio-inspired algorithms: two earlier employed algorithms (GA and PSO) are considered as the baseline, while the others (ACO, BS, Bee Search, CS, MOEA) are considered as recently-employed algorithms in [1]. We also selected HS and TS because these two algorithms have satisfactory performance in a recent empirical study about SDEE [12] and hence we also wanted to evaluate its contribution for software fault prediction. In particular, to set the parameters of the estimation techniques, we have employed 'MultiSearch' algorithm in Weka to automatically search and tune the parameters of RF, SVR, and LR. The Search algorithm we choose in MultiSearch is 'Default', which acts as Grid Search.

2.3. Validation method and Evaluation criteria

To validate the accuracy of obtained fault prediction models, we applied k-fold cross-validation which divides a dataset into equally-sized k subsets [3], where one subset is used for testing and the other k-1 subsets are used for building and training the estimation techniques. In particular, we employed a 10 fold-cross-validation, repeated 10 times.

To evaluate and compare predictions we consider mean of Absolute Errors (AE, i.e., difference between predicted and actual faults) and verify if there is a statistical significant difference between absolute errors/residuals obtained with different estimation models by using the Wilcoxon Signed Rank test (with α=0.05). Apart from the Wilcoxon test, it is advisable to determine effect size and we consider the Vargha and Delaney's A12 test [12] because, unlike the pooled Cohen'd, it does not require that data to be normally distributed. The effect size can be regarded as small if A12 is from 0.57 to 0.64, medium if A12 is from 0.65 to 0.71, and large if it is over 0.71. A value less than 0.57 is classified as negligible.

3. Results and discussion

Tables 1 shows the settings (combinations of datasets and estimation technique) for which the recently-employed bio-inspired feature selection algorithms performed better than the baselines GA and PSO, for RF estimation technique. In particular, the first column (Better) shows the newly introduced bio-inspired algorithm, the middle column (Worse) indicates the baseline (GA and PSO), while the third column (Setting) reports the employed dataset and estimation technique. The second column also shows the result (i.e., p-value) of the performed statistical test between brackets together with the indication of effect size (i.e., L means large, M medium, S small, and N negligible).

First, we discuss the experiments in which a particular algorithm has 'completely outperformed the baselines', i.e., provided better results than both GA and PSO.

Table 1: Cases when recently-introduced bio-inspired algorithms performed better than baselines (using RF)

Better	Worse	Setting	Better	Worse	Setting
ACO HS	GA (0.02 M) GA (0.007 M)	RF/Ant	ACO BS	PSO (<0.001 L) PSO (<0.001 L)	RF/Synapse
TS BS ACO HS CS	GA(0.06 S), PSO(<0.001 L) GA (0.241 N), PSO (0.046 N) PSO (<0.01 M) PSO (<0.01 M) PSO (0.1 S)	RF/Ivy	MOEA HS BS CS	GA (<0.001 L), PSO (<0.001 L) GA (<0.001 M), PSO (0.006 M) GA (<0.001 L), PSO (<0.001 L) PSO (<0.001 L)	RF/Lucene
ACO TS BS HS CS Bee	GA (<0.001 M), PSO (<0.001M) GA (<0.001 M), PSO (0.003 M) GA (0.006 S), PSO (0.009 S) GA (0.067 M), PSO (0.4 N) GA (<0.001 L) GA (0.06 S)	RF/Jedit	HS BS CS	GA (<0.001 L), PSO (<0.001 L) GA (0.006 L), PSO (0.001 L) GA (0.2 L), PSO (<0.001 L)	RF/Poi
	None outperformed baselines	RF/Velocity			

From Table 1, we can see that TS and BS have 'completely outperformed the baselines' when employed with the dataset Ivy. However, when BS is compared with GA, it provided better predictions but, without a notable effect size, i.e., effect size of small, medium or large. Similarly, for RF/Jedit, ACO, TS, and BS 'completely outperformed the baselines' with a notable effect size, and HS with a negligible effect size. For RF/Lucene, MOEA, HS, and BS 'completely outperformed the baselines' algorithms with a notable effect sizes. Similarly, for RF/Poi, the bio-inspired algorithms which 'completely outperformed the baselines' are HS, BS, and CS (with notable effect size). When RF is investigated with Synapse, ACO and BS provided better results than only PSO (with notable effect size). Interestingly, with the combination of RF/Velocity, none of the bio-inspired algorithms has provided better results than the baselines, i.e., GA and PSO have outperformed all recently-introduced bio-inspired algorithms.

Taking into account the employed newly introduced bio-inspired algorithms (7), the baseline algorithms (2), and the datasets (7), in total we have 98 different experiments for RF. Thus, we can summarize the results for RF as follow:

- *we have only about 34 (35%) cases where the baselines are outperformed by recently-employed algorithms (with a notable effect size in 31 cases).*
- *the (recently-employed) bio-inspired algorithms which mostly 'completely outperformed the baselines' are: BS (4 times), HS (3 times), and TS (2 times), while the worst performed (recently-employed) bio-inspired algorithm is Bee Search, providing better results than GA just in a single experiment.*

Due to space constraints we do not report similar tables for SVR and LR (as Table 3), but in the following we summarize the results, which confirm that the recent bio-inspired algorithms outperformed the baseline in less than 50% experiments, and even with a lesser percentage when it comes to the results with a notable effect size.

In the case of SVR/Jedit, HS, TS, CS, and MOEA all have 'completely outperformed the baselines', with a statistically significant difference. With SVR/Lucene, MOEA, TS, ACO, Bee, and HS 'completely outperformed the baselines' (but no statistically significant difference is found, apart for HS vs PSO). In the case of SVR/Poi, the algorithms that 'completely outperformed the baselines' are HS, ACO, and TS. However, only in 2 out 6 cases a statistically significant difference was found (HS vs GA and ACO vs GA). Similarly, for SVR/Synapse, Bee Search, CS, and TS outperformed baselines, but only in 2 out 6 cases significant better predictions were obtained (i.e., Bee vs GA and CS vs GA). As for SVR/Velocity, all the recently-employed algorithms did not outperform the baselines.

In a nutshell, when using SVR, among 98 different experiments comparing recently introduced bio-inspired algorithms and baselines, we can summarize the results as follow:

- *Only in 45 (46%) cases the baselines are outperformed and, among them, in 19 experiments the differences are statistically significant with a small, medium, or large effect size. The recently-employed algorithms which 'completely outperformed the baselines' in more experiments are TS (4 times) and CS (3 times).*

Regarding the use of LR, we report that in the case of Ant dataset, HS 'completely

outperformed the baselines'. However, the effect size is negligible, so we cannot claim that HS provided significantly better predictions than the baselines. Similarly, with the LR/Ivy combination, only HS 'completely outperformed the baselines', but with notable effect size only in case of GA. Similarly, MOEA, Bee, and ACO outperformed only GA and only in the case of MOEA statistically significant difference was found. With LR/Jedit, HS 'completely outperformed the baselines' (with a notable effect size) while CS, ACO, and MOEA outperformed only GA (with a notable effect size). In the case of LR/Lucene combination, MOEA, HS, and BS 'completely outperformed the baselines' (but without any notable effect size). Similarly, when LR is employed with Poi, both HS and BS 'completely outperformed the baselines' (with a notable effect size). For LR/Synapse combination, ACO, MOEA, HS, and BS outperformed only PSO but none provided significantly better fault predictions. As for LR/Velocity, all the recently-employed algorithms did not outperform the baselines. With Velocity, there is no recently-employed algorithm which have outperformed any of the two baselines (i.e., GA and PSO). With RF, HS and BS performed better, with SVR, HS and TS have better performance while with LR, only HS managed to provide better prediction accuracy.

Thus, among the 98 experiments with LR, the results can be summarized as follow:

- *Only in 27 experiments (28%) the baselines are outperformed, and, among them, in 12 cases we found difference in the predictions with small, medium, or large effect size. Furthermore, the algorithms which 'completely outperformed the baselines' in more experiments are HS (5 times), BS (2 times), and MOEA (1 time).*

To complete the discussion about our research question, we also analysed the cases where the baseline algorithms (i.e., PSO and GA) performed better than the recently investigated bio-inspired algorithms. We have observed that GA outperformed the recently-employed bio-inspired algorithms in 96 different experiments. For each of the baseline algorithm (GA and PSO), we have a total of 147 experiments (i.e., 7 bio-inspired algorithms * 7 datasets * 3 estimation techniques), so GA has obtained better performance in 65% of the cases. However, the experiments where GA outperformed the other algorithms with a notable effect size are 61 (41%). Similarly, PSO provided better results than the recent bio-inspired algorithms in 90 (61%) experiments and in 53 (36%) cases the results were significantly better, with a small, medium, or large effect size. For one dataset (i.e., Velocity), both GA and PSO completely outperformed all the recent bio-inspired algorithms, regardless of the estimation techniques employed.

To sum up, GA provided better results in 41% and PSO in 36% experiments than the recently employed algorithms, with a notable effect size.

4. Findings and suggestions

To sum up, if we have to use (regression based) SFP data, it is better to use HS and BS if the employed estimation technique is either RF or LR, while we should prefer to use TS with SVR. Moreover, caution is required about the use of more recently employed algorithms because the baseline algorithms (GA and PSO) outperformed all the recent algorithms with Velocity dataset for all the three estimation techniques. GA should be preferred regardless of the (regression based) SFP datasets.

However, if we have to suggest a bio-inspired algorithms for feature selection in the context of SFP we can for sure indicate to start with HS (it is the only one which performed better in the majority of experiments) and then proceed with others (e.g., BS). Bee Search, MOEA, ACO, and CS should not be used in the future SFP studies according to our results.

5. Future work

We will evaluate the performance of the same set of algorithms but for the software fault classification problem, employing different type of datasets such as NASA MDP and other open source projects from Github. Apart from evaluating their prediction accuracy, we intend to evaluate the consistency of input metrics selected by various feature selection algorithms using same training samples, different training samples, different validation methods as well as by shuffling the order of various features in a dataset.

References

1. Ali, A., Gravino, C.: “Review of Bio-inspired Algorithms in Software Fault Prediction: A Systematic Literature Review submitted to a conference”. In International Conference on Open Source Systems and Technologies (ICOSST), 2020.
2. Boetticher, G., Menzies, T., Ostrand, T.: “PROMISE Repository of empirical software engineering data”. http://promisedata.org/ repository
3. Briand L., Emam K., Surmann D., Wieczorek I.: Maxwell K.: “An assessment and comparison of common software cost estimation modeling techniques”. ICSE, 313-322, 1999.
4. Di Martino S., Ferrucci F., Gravino C., Sarro F.: "A genetic algorithm to configure support vector machines for predicting fault-prone components". PROFES. pp. 247-261, 2011.
5. Fong, S., Biuk-Aghai, R.P., Millham, R.C.: "Swarm Search Methods in Weka for Data Mining". ICMLC, (2018) pp.: 122-127.
6. Gao, K., Khoshgoftaar, T.M., Wang, H., Seliya, N.: “Choosing software metrics for defect prediction: an investigation on feature selection techniques”. Softw. Pract.Exper, 41(5) (2011): 579-606.
7. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: "The WEKA data mining software: an update". ACM SIGKDD explorations newsletter 11(1) (2009): 10-18.
8. Ibrahim, D. R., Ghnemat, R., Hudaib, A.: "Software Defect Prediction using Feature Selection and Random Forest Algorithm". New Trends in Comp Sciences, pp. 252-257. IEEE, 2017.
9. Kumar, L., Rath, S. K.: “Application of genetic algorithm as feature selection technique in development of effective fault prediction model”. ICECEE, pp. 432-437, IEEE, 2016.
10. Manivasagam G., Gunasundari R.: “An optimized feature selection using fuzzy mutual information based ant colony optimization for software defect prediction”. IJET; v(7)1.1, 2018.
11. Menzies, T., Greenwald, J., Frank, A. “Data mining static code attributes to learn defect predictors”. IEEE TSE 2007. 33(1), 2–13
12. Mohamed, H., Idri, A., Abran, A.: "Investigating heterogeneous ensembles with filter feature selection for software effort estimation". In IWSM, pp. 207-220. ACM, 2017.
13. Malhotra, R.: “A systematic review of machine learning techniques for software fault prediction,” Appl Soft Computing, 27 (2015): 504–518.
14. Nam, J., Fu, W., Kim, S., Tim Menzies, T., Tan, L.: “Heterogeneous Defect Prediction”. IEEE Trans. Software Eng. 44(9): 874-896 (2018)
15. Wahono, R. S., Suryana, N.: "Genetic feature selection for software defect prediction". Adv Science Letters 20(1):239-244, 2014
16. Wahono, R.S., Suryana, N., Ahmad, S.: "Metaheuristic optimization based feature selection for software defect prediction". Journal of Software 9(5) (2014): 1324-1333.