International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Development of A Novel Classifier Model Implementing Bio-Inspired Algorithms Tt Optimize Existing Models for Software Defect Detection

Surya Tejas V[1], Sumukha U[2], Jash Singh[3], Karthik G[4], Sahana P. Shankar[5]

[1,2,3,4]Student, Ramaiah University of Applied Sciences, 470-P, Peenya 4th Phase, Peenya, Bengaluru, Karnataka 560058, India
[5]Assistant Professor, Department of Computer Science and Engineering Ramaiah University of Applied Sciences Bengaluru, India

Abstract
Bio-inspired algorithms (BIA) have emerged in recent times as a means of optimizing traditional computing methods. They have been derived based on events happening in nature. This study of bionics connects the functions, biological structures and organizational patterns and principles found in nature with the our current day technologies, mathematical and metaheuristic algorithms have been derived [1]. Algorithms such as particle swarm optimization, differential evolution, cuckoo search and firefly algorithms all have both global search and intensive local search capabilities, which may be partly why they are so efficient[2] . Detecting and preventing software defects is a very significant and important need in the software industry. There are many standard models using different techniques for implementation for software defect detection. Software defect detection dataset uses attributes of a software which indicates the presence of a defect. Models such as classifiers learn patterns in these attributes that are used to represent the presence or absence of a defect[4]. There have already been works related to using bio-inspired algorithms for feature selection to improve the efficiency of standard classifiers. The main goal of this paper is to develop a model which implements Harris' hawks optimization method to further optimize the features selected by bat algorithm and create hyperparameters for XGB classifier. The paper also includes to develop and compare the above model with three other models; one model without any optimization, feature selection or feature extraction methods, a model using a standard feature selection method, a model incorporating bio- inspired algorithm for feature selection. Then the metrics for the model is calculated and presented. The models are trained over a software defect detection dataset. The models are then compared to draw conclusions.

Keywords: Bio-inspired algorithms, software defects, bat algorithm, classifiers, feature selection, hyperparameters, metrics

1. Introduction:

Optimization is an important and integral problem of engineering disciplines. The generic optimization algorithms used earlier showed to require large computational efforts leading to failure as the problem size scales up. On more research to have better optimization techniques, bio-inspired algorithms started to emerge. These algorithms are developed, driven and motivated by the mechanisms and principles found in nature[2][3]. Nature always finds the most optimal method to solve problems with balance in resources. From observation we can see that nature finds the most optimal strategy for solving a problem while at the same time making the most efficient use of its resources. Bio inspired algorithms are derived based on nature for this very feature of nature's mechanisms. BIAs are heuristic approaches that mimic these optimal strategies and mechanisms of natures for optimization[3]. This area of bio-inspired algorithms hold a massive potential in regards to optimizing models and the way we approach algorithms currently. Comparing and analyzing models which use standard methods to that of those using bio-inspired algorithms gives us clarity over how bio-inspired algorithms help and the degree to which they can help in developing machine learning models. This analysis also provides an insight into the methods of implementing bio inspired algorithms and if they can contribute into solving real life problems and needs. Software defect detection, as mentioned earlier is a very important need for the software industry. The requirement of software defect detection arises in order to release a defect free satisfactory product. Defect prediction ensures the right development action is taken to prevent software degradation preventing future problems[4]. Hence, developing a model in this field using bio- inspired algorithm will help us discover the versatility of bio-inspired algorithm to be used in different types of problems and how much they can help enhancing currently used algorithms. Although there have been different models developed for software defect detection using algorithms such as Naïve Bayes, neural networks, random forest, K-nearest neighbors and other such algorithms implementing such a dataset for software defect detection also proposes an idea of how bio inspired algorithms can be used to enhance these models and help in solving the latest problems[5]. There have also been related works that use bio-inspired algorithms for feature selection followed by using a standard classifier. However, bio-inspired algorithms can be further used to optimize the models. This paper aims to develop a model which implements Harris' hawks optimization method to optimize the features selected by bat algorithm and create hyperparameters for XGB classifier and draw conclusions after comparing it with other models we implemented.

2. Background:

CM1 dataset has been used for this project from promise repository. It includes 22 attributes or columns. CM1 is a NASA spacecraft instrument written in "C". Data comes from McCabe and Halstead features extractors of source code. These features were defined in the 70s in an attempt to objectively characterize code features that are associated with software quality. The nature of association is under dispute. The McCabe metrics are a collection of four software metrics: essential complexity, cyclomatic complexity, design complexity and LOC, Lines of Code. The Halstead falls into three groups: the base measures, the derived measures, and lines of code measures. The attributed in the dataset include:

1. loc: (numeric) McCabe's line count of code
2. v(g):(numeric) McCabe "cyclomatic complexity"
3. ev(g): (numeric) McCabe "essential complexity"
4. iv(g): (numeric) McCabe "design complexity"
5. n: (numeric) Halstead total operators + operands

6. v: (numeric) Halstead "volume"
7. l: (numeric) Halstead "program length"
8. d: (numeric) Halstead "difficulty"
9. i : (numeric) Halstead "intelligence"
10. e: (numeric) Halstead "effort"
11. b: (numeric) Halstead parameter
12. t: (numeric) Halstead's time estimator
13. lOCode: (numeric) Halstead's line count
14. lOComment : (numeric) Halstead's count of lines of comments
15. lOBlank : (numeric) Halstead's count of blank lines
16. lOCodeAndComment: numeric
17. uniq_Op: (numeric) unique operators
18. uniq_Opnd: (numeric) unique operands
19. total_Op : (numeric) total operators
20. total_Opnd: (numeric) total operands
21. branchCount: (numeric) of the flow graph
22. defects : (numeric) module has/has not one or more reported defects

The details regarding derivation of these attributes and their relation to software defect prediction is out of the scope of this paper. Firstly, the data is imported from the csv file into a dataframe. Then its split into data for training and testing into the ratio 8:2 respectively.

We have used XG Boost (XGB) classifier on this dataset to create a model. XG Boost is an algorithm that uses multithreading and parallel processing and is similar to a gradient boosting algorithm. It implements the usage of a loss function and choosing a weak learner to make predictions. Usually a decision tree is used as a weak learner. An additive model is then added to add up the predictions of the weak learners. After many such iterations the process stops after an optimized value for the loss function is reached. In gradient boost this addition of trees or weak learners happen one at a time while XGB classifier uses a multithreaded approach to use the CPU efficiently. This model is said to be very apt for structured data and for deriving classification and regression and predictive solutions[6].

Principle component analysis can be used as a dimensionality reduction method to contain only the important features that impact the model and its predictions. After implementing PCA, the number of attributes into consideration is reduced to 10.

The bat algorithm is a swarm intelligence optimization algorithm with quick convergence speed and strong searching ability[7]. It is based on how bats use echolocation to identify or differentiate between food and background barriers. Bats fly with a particular velocity at a position with frequency ranging between two values, varying wavelength and a particular loudness. Depending on the proximity to the target they change the frequency or wavelength of emitted pulses. New solutions are generated by adjusting the various parameters of loudness, frequency, amplitude and pulse rate. Once the prey is found the rate increases while amplitude decreases. Based of this mechanism the algorithm is created[8].

The inspiration for HHO lies in the way Harris hawks in nature exhibit the cooperative behaviour and attack called surprise pounce. In this strategy several Harris' hawks surprise pounce on a single prey from many directions cooperatively to surprise it . The initial phase called the exploitation phase is when the group of hawks are exploring the environment. Position functions are assigned to the prey and the hawks. A rule generates solution based on random locations and other hawks. Another rule generates difference

of location of the hawk with position best so far and the average position of the group of hawks. Coefficients to randomize the positions in the feature space are used. Second phase is the transition from exploration to exploitation. In this phase energy of the prey is considered, which decreases during its escaping behaviour. When escaping energy is high the hawks search different locations to locate the prey in exploration phase and when its low the algorithm tries to exploit the neighbourhood of solutions during the exploitation phase. The third phase is the exploitation phase where the Harris' hawks perform their surprise pounce attacks by attacking the detected prey. The hawks change strategies as the prey tries to escape based on its escaping patter. Four such strategies are proposed in HHO model to mimic this attack. The hawks encircle the prey softly or hard based on the energy retained by the prey. Over time the energy reduces and the hawks get closer to increase their cooperative killing chances[9].

3. Related work:

Paper on Bio-inspired algorithms in software fault prediction: A systematic literature review[10] has conducted a literature review on the topic of software fault prediction and how there have been works of the usage of bio-inspired algorithms in the said field. It is a literature review inclusive of 34 studies regarding feature selection and parameter optimization displaying how bio inspired algorithms perform effectively in those tasks. Feature selection using firefly algorithm in software defect prediction[11] proposes the implementation of firefly algorithm for feature selection with classifiers such as Naïve Bayes, Support Vector Machine and K- Nearest Neighbors. It shows the way in which firefly algorithm can be implemented with classifiers for software defect detection. The paper on Metaheuristic optimization based feature selection for software defect prediction[12] provides an insight on the usage of metaheuristic algorithms being very effective in providing high quality solutions within a reasonable period of time for feature selection by finding solutions by searching the full search space. The results for the paper show how the proposed methods make a very significant improvement in prediction performance for classifiers. Software Defect Prediction Using Optimized Cuckoo Search Based Nature-Inspired Technique[13], paper shows the implementation of cuckoo search for finding software defects and bugs. A model for software defect prediction using support vector machine based on CBA (classification based on association rules)[14] shows us the implementation of bat algorithm with centroid strategy for software defect prediction. It makes use of non-linear computing ability of support vector machines and the optimization ability of bat algorithm

4. Proposed work:

Harris' Hawks optimization (HHO) algorithm to extract hyperparameters from selected features. The features are first selected using bat algorithm and then HHO is performed on it. Then we train the XGB classifier over the hyperparameters generated by HHO. Using the above methods we create an ensemble model. Then we find the best metrics for the model.

Now we create three models which do not include the hyperparameterisation of Harris' hawks optimization algorithm.

5. XGB classifier:

For the first model we have considered using only the XGB classifier over the CM1 dataset.

Below is the pseudocode for XGBoost Classifier:

Initialization:

- Initialize model parameters:
- Learning rate (`eta`), number of trees (`num_trees`), maximum depth of trees (`max_depth`), regularization parameters, etc.
- Initialize an empty ensemble of trees.

Training:

- For each boosting round:
- Compute the gradient and Hessian of the loss function for each instance.
- Fit a decision tree to the negative gradient (residuals) using the Hessian as weights.
- Update the ensemble with the new tree, scaled by the learning rate.
- Apply regularization techniques like shrinkage and feature subsampling.

Prediction:

- To make predictions:
- For each input sample:
- Aggregate predictions from all trees in the ensemble.
- Apply any necessary transformations (e.g., sigmoid function for binary classification).

The performance and metrics were obtained for the above model.

6. XGB classifier with PCA for feature selection:

Then a model of XGB classifier that is trained on reduced number of features is created. The reduced features were given using Principle Component Analysis.

PCA feature selection algorithm where the number of components is set at k:

- Center the data by subtracting the mean from each feature.
- Scale the data by dividing each feature by its standard deviation.
- Calculate the covariance matrix of the standardized data.
- Use eigenvalue decomposition on the covariance matrix to compute eigenvectors and eigenvalues.
- Sort the eigenvectors by their corresponding eigenvalues in descending order.
- Select the top k eigenvectors corresponding to the largest eigenvalues.
- Project the standardized data onto the selected eigenvectors to obtain the transformed data matrix.
- Select the features corresponding to the columns of the transformed data matrix.
- Return the indices of the selected features.

This algorithm utilizes PCA to reduce the dimensionality of the data while selecting the most informative features for downstream analysis. XGB classifier model was created based on these selected reduced attributes.

7. XGB classifier with Bat algorithm for feature selection:

For the third model we will be using bat algorithm for feature selection over the XGB classifier. The pseudocode is as follows:

- Objective function $f(x)$, $x = (x_1 \ldots x_d)^T$
- Initialize the bat population $x_i\ (i = 1, 2, \ldots, n)$ and Vi
- Define pulse frequency f_i at X_i
- Initialize pulse rates r_i and the loudness A_i

- while (t< Max number of iterations)
- Generate new solutions by adjusting frequency,
- and updating velocities and locations/solutions
- if (rand > r_i)
- Select a solution among the best solutions
- Generate a local solution around the selected best solution
- end if
- Generate a new solution by flying randomly
- if (rand < A_i & $f(x_i) < f(x^*)$)
- Accept the new solutions
- Increase r_i and reduce A_i
- end if
- Rank the bats and find the current best X*
- end while
- Postprocess results and visualization

After best features are selected, we train XGB classifier over these selected features.

8. XGB classifier with PCA for feature selection and HHO for optimization:

Finally, the ensemble model which implements Harris' Hawks Optimization was implemented over the dataset to generate hyperparameters or the most optimal parameters for XGB classifier model. These parameters are generated on the reduced dataset after using bat algorithm for reduction. Then the hyperparameters were used to create the XGB classifier.

Below is the pseudocode for HHO algorithm.

- Initialize the population of hawks (solutions).
- Randomly initialize the position of each hawk in the search space.
- Set algorithm parameters such as the population size, maximum number of iterations, and control parameters.
- Evaluate the fitness of each hawk using an objective function.
- Assign fitness values to each hawk based on its position in the search space.
- Repeat until a termination condition is met (e.g., maximum number of iterations):
- Sort the hawks based on their fitness values.
- Update the leader hawk:
- The best-performing hawk becomes the leader (global best solution).
- Update the position of each hawk:
- Perform exploration: Randomly perturb the position of each hawk to explore new solutions.
- Perform exploitation: Move each hawk towards the leader's position to exploit promising regions.
- Update the position of each hawk based on the exploration and exploitation steps.
- Perform boundary checking:
- Ensure that the updated positions of hawks are within the boundaries of the search space.
- Update the fitness values of hawks based on their new positions.
- Return the best solution found (position of the leader hawk) along with its fitness value.

This algorithm leverages the hunting behavior of Harris' hawks, where hawks cooperate and compete to efficiently search for prey, to perform optimization in a multidimensional search space. Then the metrics for the above created model was obtained. The different metrics obtained for the above four models were compared to determine the best model for the dataset.

9. Results and Analysis:

We compare the metrics of the following models.

XGB classifier model without any feature selection and optimization:

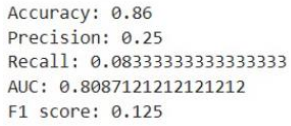

Figure 1 Performance metrics of XGB classifier without feature selection

XGB classifier with PCA:

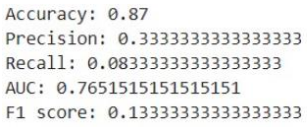

Figure 2 Performance metrics of XGB classifier with PCA

XGB classifier with bat:

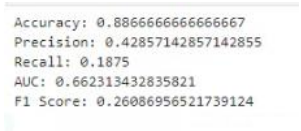

Figure 3 Performance metrics of XGB classifier with BAT

XGB classifier with bat and HHO:

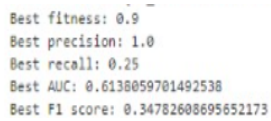

Figure 4 Performance metrics of ensemble model

The above results are tabulated for easy comparison.

Metric/Model	XGB classifier	XGB classifier with PCA for feature selection	XGB classifier with Bat algorithm for feature selection	XGB classifier with bat for feature selection and HHO for optimization
Accuracy	0.86	0.87	0.886	0.9
Precision	0.25	0.333	0.428	1.0
Recall	0.083	0.083	0.188	0.25
AUC	0.808	0.765	0.662	0.613
F1 score	0.125	0.133	0.261	0.347

From the table we can observe that the model XGB classifier with bat for feature selection and HHO for optimization performs the best with the highest accuracy of 0.9, precision 1.0, recall 0.25, AUC 0.613 and F1 score of 0.347. The standard model of XGB classifier arguably has the worst metrics of accuracy of 0.86, precision 0.25, recall 0.083, AUC 0.808 and F1 score of 0.125 compared to other models we have implemented. This shows that implementing feature selection methods and optimization techniques

successfully improved the results.

10. Conclusion:

Summary of key findings:

In summary, this project shows us the implementation of bio-inspired algorithms to optimize currently existing algorithms. Harris' hawk's optimization optimizes standard models as well as models optimized by bio- inspired algorithms for feature selection to a higher degree. We can also observe that the bat algorithm enhanced the standard XGB model more compared to PCA. It also shows how meta-heuristic approaches are more suitable in optimizing a predictive model compared to traditional methods. We can also see the versatility of using bio-inspired algorithm even in latest crucial problems as that of software defect detection. Furthermore, we can see the drastic enhancement on the standard model when HHO has been used for generating hyperparameters. Based on the results, it is safe to say that bio-inspired algorithms have a great potential in optimizing solutions for modern day problems and how algorithms inspired by nature can pave a way to very efficient and optimal enhancement of current technologies. Harris' Hawk's optimization has successfully optimized existing standard models and even models with feature selection to a higher degree.

Contributions to the field:

XGB classifier is a fairly new model introduced in 2014. This model was not experimented enough with for the application of training over software defect detection datasets. We can observe that standard XGB classifier model has a fairly good accuracy of 0.86. We have also explored the areas of using feature selection to optimize the model and the suitability of implementing it with the software defect detection dataset. We can see how compatible feature selection is for the given model to improve its performance. Most importantly, we have explored the latest trending ideologies of using bio-inspired algorithm to obtain important results as to how they improve existing models and how better they perform in optimization of currently used standard models. The ensemble model created displays creative ways of implementing the newly developed bio- inspired algorithms to produce the best results. This gives us an inspiration to explore and find algorithms inspired by nature which arguably is having the most efficient methods to balance and produce the best results for any given situation.

Future scope and ideas:

Drawing inspiration from nature in generating algorithms has proven to be very effective compared to the methods currently in use. The above research shows an example of how implementation of bio inspired algorithms can better the currently used algorithms. The future scope in this field includes exploring different methods of implementing currently developed bio-inspired algorithms in different domains as they have proven to be very versatile in terms use cases. Furthermore, many other bio-inspired algorithms can be created and developed based on new inspirations derived form nature. Overall, the entire field of implementing and developing bio inspired is new, and exploring this uncharted territory could lead to the discovery of a possible gold mine in the field of algorithm development and optimization.

References

1. Fan, X., Sayers, W., Zhang, S., Han, Z., Ren, L. and Chizari, H., 2020. Review and classification of bio-inspired algorithms and their applications. *Journal of Bionic Engineering*, *17*, pp.611-631.
2. Fister Jr, I., Yang, X.S., Fister, I., Brest, J. and Fister, D., 2013. A brief review of nature-inspired algorithms for optimization. *arXiv preprint arXiv:1307.4186*.

3. Binitha, S. and Sathya, S.S., 2012. A survey of bio inspired optimization algorithms. *International journal of soft computing and engineering*, *2*(2), pp.137-151.
4. Perreault, L., Berardinelli, S., Izurieta, C. and Sheppard, J., 2017, October. Using classifiers for software defect detection. In *26th International conference on software engineering and data engineering* (pp. 2-4).
5. Shankar, S.P., Pushp, D., Makadia, N., Dubey, R., Nayak, A. and Chaudhari, S.S., 2023, September. Software Defect Predictor and Classifier Tool Using Machine Learning Techniques. In 2023 International Conference on Network, Multimedia and Information Technology (NMITCON) (pp. 1-6). IEEE
6. Ramraj S, Uzir N, Sunil R, Banerjee S. Experimenting XGBoost algorithm for prediction and classification of different datasets. International Journal of Control Theory and Applications. 2016;9(40):651-62.
7. Yin, Z., Pan, L. and Fang, X., 2019. Bio-inspired computing: theories and applications. In *Proceedings of the 14th International Conference, BIC-TA 2019*.
8. Chawla, M. and Duhan, M., 2015. Bat algorithm: a survey of the state-of-the-art. *Applied Artificial Intelligence*, *29*(6), pp.617-634.
9. Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M. and Chen, H., 2019. Harris hawks optimization: Algorithm and applications. *Future generation computer systems*, *97*, pp.849-872.
10. Ali, A. and Gravino, C., 2020, December. Bio-inspired algorithms in software fault prediction: A systematic literature review. In *2020 14th International Conference on Open Source Systems and Technologies (ICOSST)* (pp. 1-8). IEEE.
11. Anbu, M. and Anandha Mala, G.S., 2019. Feature selection using firefly algorithm in software defect prediction. *Cluster Computing*, *22*, pp.10925-10934.
12. Wahono, R.S., Suryana, N. and Ahmad, S., 2014. Metaheuristic optimization based feature selection for software defect prediction. *J. Softw.*, *9*(5), pp.1324-1333.
13. Srinivasa Kumar, C., Sirisati, R.S. and Thonukunuri, S., 2021. Software Defect Prediction Using Optimized Cuckoo Search Based Nature-Inspired Technique. In *Smart Computing Techniques and Applications: Proceedings of the Fourth International Conference on Smart Computing and Informatics, Volume 2* (pp. 183-192). Springer Singapore.
14. Rong, X., Li, F. and Cui, Z., 2016. A model for software defect prediction using support vector machine based on CBA. *International Journal of Intelligent Systems Technologies and Applications*, *15*(1), pp.19-34.