International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-5, January 2020

Boosted Relief Feature Subset Selection and Heterogeneous Cross Project Defect Prediction using Firefly Particle Swarm Optimization

N. Kalaivani, R. Beena

Abstract: ***The exponential growth in the field of information technology, need for quality-based software development is highly demanded. The important factor to be focused during the software development is software defect detection in earlier stages. Failure to detect hidden faults will affect the effectiveness and quality of the software usage and its maintenance. In traditional software defect prediction models, projects with same metrics are involved in prediction process. In recent years, active topic is dealing with Cross Project Defect Prediction (CPDP) to predict defects on software project from other software projects dataset. Still, traditional cross project defect prediction approaches also require common metrics among the dataset of two projects for constructing the defect prediction techniques. Suppose if cross project dataset with different metrics has to be used for defect prediction then these methods become infeasible. To overcome the issues in software defect prediction using Heterogeneous cross projects dataset, this paper introduced a Boosted Relief Feature Subset Selection (BRFSS) to handle the two different projects with Heterogeneous feature sets. BRFSS employs the mapping approach to embed the data from two different domains into a comparable feature space with a lower dimension. Based on the similarity measure the difference among the mapped domains of dataset are used for prediction process. This work used five different software groups with six different datasets to perform heterogeneous cross project defect prediction using firefly particle swarm optimization. To produce optimal defect prediction in the Heterogeneous environment, the knowledge of particle swarm optimization by inducing firefly algorithm. The simulation result is compared with other standard models, the outcome of the result proved the efficiency of the prediction process while using firefly enabled particle swarm optimization.***

Keywords: ***Software defect, Cross project, Heterogeneous cross project, Boosted relief feature selection, Firefly, Particle swarm optimization.***

I. INTRODUCTION

In recent years, approaches for software defect prediction in earlier stages of software development is highly focused and various effective models of defect prediction approaches are developed and a lot of attention is given by both commercial and academic communities.

Many of the existing studies commonly intensive on Within Project Defect Prediction (WPDP), which allows the prediction model to be trained with the historical data to detect software defects on other software modules of same project [1].In simple, while performing WPDP the intra project dataset is used for both training and testing. But, in real time investigation, the researches confront on gathering such voluminous historical data within the same project, this problem can be tackled by getting data from other projects to assist the learning of target software projects. Due to this reason, CPDP is introduced, which is the art of using dataset of inter project to perform software defect prediction in the target project along with small ratio of local data [2].

However, while deploying CPDP it works under the assumption of both the source and the target project data must comprised of same feature's or metrics. When the data distribution ratio changes or the feature space of source and target project are dissimilar then it is not possible to achieve the expected result by using these techniques.

Hence, the need for Heterogeneous Cross Project Defect Prediction Models (HCPDP) are demanded for the above-mentioned scenario where the source project and the target project dataset comprised of different features or metrics and their space or size is not same.

In many cases, the datasets of software defect detection are imbalance, that is, number of modules which are defective is commonly much smaller in ratio compared to modules with defect free [3]. This nature of imbalance in data can leads to worst prediction performance, where the possibility of predicting defect can be low while the entire performance is high.

The ultimate objective of this paper is to handle such worst cases of low volume defect prediction data by developing nature inspired Heterogeneous cross project software defect prediction model.

II. PROBLEM STATEMENT

In general, software defect prediction models are constructed using intra-project dataset. But in practice, lack of training data with defect modules, at earlier stages of software testing restricts the efficiency of prediction process. Even in cross project defect prediction, it is accomplished only when those projects dataset must have same set of metrics with same size. In addition, it also faces the class imbalance issue, which increases the difficulty for the researchers to predict the defects in software modules.

Manuscript published on January 30, 2020.
* Correspondence Author
Mrs.N.Kalavani*, Research Scholar, Department of Computer Science, Kongunadu Arts and Science College, Coimbatore.
Dr.R.Beena, Associate Professor and Head, Department of Computer Science, Kongunadu Arts and Science College, Coimbatore.

To overcome this issue, this proposed work proposed a two stage Heterogeneous cross project defect prediction which uses data from other projects even in presence of different metrics with varied feature space.

In this work, boosted relief feature subset selection is used to make the source project to be trained with the target project space by mapping in to the equal feature space.

To predict the defect, firefly enable particle swarm optimization is used for classifying the defect and non-defect modules very effectively.

III. RELATED WORK

In last years, research related to Heterogeneous cross project defect prediction, but still it faces severe challenges, because of using inter projects with different metrics and size as primary factor for discovering defect prediction in software project. This section discusses about some of the research works related to both cross project defect prediction and Heterogeneous project defect prediction.

Nam et al. [4] in their work to handle the Heterogeneous project defect prediction they utilized the concept of selection of metrics and selection of similar metrics using metric matching to construct the model of software defect prediction. But this model rejected dissimilar metrics, which may cover beneficial evidence for training.

Jing et al. [5] developed a Heterogeneous cross project defect prediction using canonical correlation analysis, they created a common correlation space to subordinate cross project data. Then source and target project dataset are place into the solution space for project defect prediction.

In existing models of CPDP, the main issue in software defect prediction is class imbalance which is taken as an important factor by Ryu et al. [6]. The authors developed a value cognitive boosting method of support vector machine which exposed sampling methods to overcome the imbalance of class issues such as defect and non-defect ration is greatly varied in cross project defect prediction. But this approach is not appreciated for using the class imbalance issue in case of heterogeneous environment. While using sampling strategy, it may discard some of the interesting samples that might influence the prediction process. So that, these approaches are not best suited for solving the class imbalance problem in HCPDP.

Zimmermann et al. [7] designed a large-scale experiment for CPDP using 12 real world projects. This research work stated that the conventional model of software defect prediction doesn't suit best for handling dissimilar metrics with varying size.

He et al. [8] in their work performed CPDP type of software defect prediction by concentrating on selection of dataset for training process. They reported that the performance of the prediction model depends on the dataset's distributional attributes.

Turhan et al. [9] designed a nearest neighbor filter approach to choose identical data from the source project. They involve only nearest neighbors of each test data to build training dataset, which consist of similar metrics of local data.

Ma et al. [10] introduced a transfer naïve bayes which doesn't concentrate only on matching training dataset, they used information of all appropriate metrics or features in training dataset. This model converts the information of cross project data to their corresponding weight dataset. Depending on the weight data, the defect prediction approach was built.

Nam et al [11] used transfer component analysis to CPDP and designed a novel enhanced TCA model by choosing an appropriate normalization done automatically for preprocessing the dataset.

Hall et al. [12] proposed a data imbalance related to particular classification methods may generate worst performance. Avoiding this problem, a learning model that minimizes the error rate of prediction would often generate hopeless predictive approach which supposed to predict all the modules as non-defect.

Wang et al. [13] revealed the importance of class imbalance problem and offer guidance and appreciated information for constructing good software defect predictor. Grbac et al. [14] investigated about various machine learning models to handle the class imbalance for predicting software defect data. By altering the training data, they exploit both feature selection and sample of data.

Jaechang et al [15] in their work developed a heterogeneous defect prediction which involves in using two different project datasets with varying metrics. Using empirical model with mathematical approach of software defect prediction, they proved that using limited instances can also produce better result.

Li et al. [16] in their work compared different data filter s and proposed a hierarchical selection-based filter to improve CPDP performance. From the result it is proved that using data filter concept can improve the performance of CPDP.

IV. PROPOSED METHODOLOGY

A. Boosted Relief Feature Subset Selection and Heterogeneous Cross Project Defect Prediction Using Firefly Particle Swarm Optimization

This proposed work enhances the process of heterogeneous defect prediction by developing two different stages. The prediction of software defects in heterogeneous environment, is very challenging because using different projects data for predicting the defected modules, with different metrics as features with varied size. Fig 1 depicts the overall workflow of the proposed model of boosted relief function with firefly enabled particle swarm optimization for heterogeneous defect prediction.

In this proposed Boosted Relief Feature Selection with Firefly Enabled Particle Swarm Optimization, six different public open repositories of software defect datasets involved in heterogeneous cross project software defect prediction is used. In real life datasets, metrics are in different measures, to make each metric to be treated at equal importance, the values of dataset are normalized using min-max normalization.

B. Min-Max Normalization

It is one among the common approaches to normalize data, here each feature, the minimum value of that feature becomes 0, the maximum value becomes 1 and each other value gets converted into decimal value ranges between 0 and 1.

$$Min - Max\ Norm = \frac{value - min}{max - min}$$

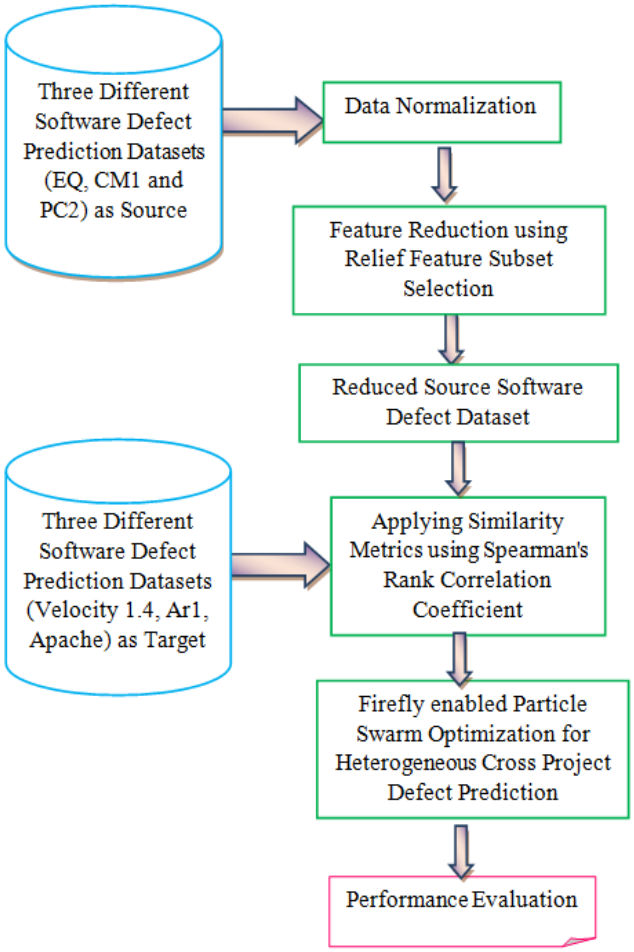


Fig.1. Boosted Relief Feature Selection with Heterogeneous Cross Project Defect Prediction using Firefly Particle Swarm Optimization.

Once the datasets are normalized, the next step is to select the source project and the target project for heterogeneous cross project defect prediction. In this work EQ, CM1 and PC2 are used as source project datasets and Velocity 1.4, Ar1 and Apache are used as target datasets. Both source and target project datasets belong to different groups of software, they have dissimilar metrics or features and size of the source project is large than the target project. In this proposed work, before performing prediction process,the size of the source project is reduced to the equal size of target project by performing boosted relief feature subset selection which weights and ranks the features of source project and it chooses the first n number of ranking features whose size is equal to the target dataset under consideration.

C. Boosted Relief Feature Subset Selection for Reducing the Size of Source Project Dataset

Standard Relief algorithm uses filter approach for selecting sensitive features, which contribute more during classification. This model computes a feature score of every feature and the ranking is applied from the obtained score and selecting top scoring features forfeature selection is its major task. Depending on the nearest neighbor's instance pair with a same class label, the feature value is computed by finding difference among the neighboring instances alone.

Given a dataset with m number of instances and d features belonging to two classes namely defect and defect free classes. Within the dataset by using min-max normalization each feature is scaled to the 0 to 1 interval.

The relief algorithm will be repeated for n times. Beginning with d long weight vector of zeros, during each iteration, select the feature Y belonging to a random instance and the closest instances to Y by calculating distance from each class. The instance with same class and closest is referred as near-hit and the instance with different class which is closest is called near-miss. Weight vector Wt is updated for each feature as follows:

$$Wt_i = Wt_i - (y_i - nH_i)^2 + (y_i - nM_i)^2$$

Where nH is the nearest hit and nM is the nearest miss, thus the weight of any selected feature decreases if it varies from that feature in nearby instance of the same class more than nearby instances of the other classes and the increases in case of vice versa. After n iterations, each element of the weight vector is divided by n. This becomes the relevance features; the selected features relevance will be greater than a predefined threshold τ.

Because of the voluminous growth in software dataset, removing useless, erroneous or noisy features are an important task. To improve the quality of the heterogeneous Defect prediction most relevant features has to be selected based on the number of features involved in the target project dataset.

For heterogeneous defect prediction using traditional feature subset selection becomes time consuming and they are inconsistency during classification process. This work introduced an ensemble model for boosting the weak learner, relief algorithm to achieve better result in selection of most relevant metrics to be used with target dataset.

The Boosted Relief Feature Subset Selection

Input:
TD: Training instance set of Source project dataset,
F: set of metrics or features F= $\{f_1, f_2, ..f_m\}$
N: Number of iterations
Begin
Assign all weights Wt[F] := 0;
For i = 1 to N
arbitrarily choose an instance IRi;
find nearest-hit nH and nearest-miss nM;
for j =1 to m
Assign $Wt[F_j]$ = $-diff_{MK}(F_j,IRi,nH)/n + diff_{MK}(F_j,IRi,nM)/n$
Sort WR[F]

remove N/m of remaining features with lowest weights

end for

return last Boosted Relief weight estimates for remaining features.

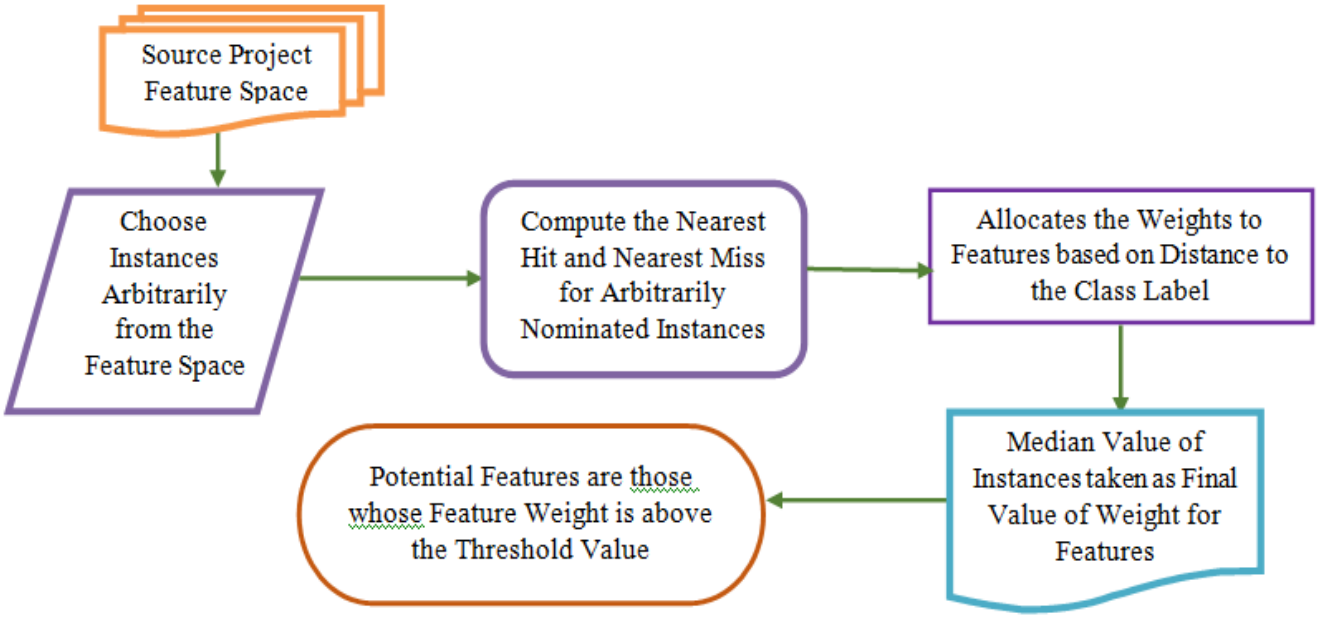


Fig. 2. Boosted Relief Feature Subset Selection

where $diff_{MK}$ refers to minkowski distance measure used for finding distance among two different instances

$$diff_{MK(R_i,R_j)} = \left(\sum_{i=1}^{n} |R_i - R_j|^p\right)^{1/p}$$

If p value is assigned to 1 then the distance is Manhattan distance or if it is 2 then it is euclidean distance.

Thus, by using boosted Relief feature selection the feature space of the source project dataset is reduced to the equivalent size of target project dataset, by selecting the top features which weight scores are high.

D. Firefly enabled Particle Swarm Optimization based Heterogeneous software defect prediction

Kennedy and Eberhart [20] as a behavioral inspiration of flocks and schooling they developed a particle swarm optimization approach to produce optimized result for a given search space. In this proposed work, PSO is used to classify and predict the defect and defect free modules of heterogeneous cross projects. It is initialized with a random population of particles by assigning their position and velocity on the software defect dataset. During each iteration the velocity and the position of the particles are updated and based on it the fitness value of each particle is determine with the influence of two parameters namely global best(glbst) and personal best position(plbst). So far visited best position by a particle is referred as plbstand the best position so far analyzed among all particles visited since so far is denoted as glbst. Each Particles velocity and position are restructured a follows:

$$Vlc(t+1) = wt.vlc(t) + pcn_1 rnd_1(plbst(t) - pst(t)) + pcn_2 rnd_2 (glbst(t) - pst(t)); t = 2,3,\dots p \qquad (1)$$

$$pst(t+1) = pst(t0 + Vlc(t+1)) \qquad (2)$$

Where, pst and vlc are position and velocity of particle, correspondingly. wt is inertia weight, pcn1 and pcn2 are positive constants, termed acceleration coefficients which regulate the impact of plbst and glbst on the search process, p is the number of iterative, rnd1 and rnd2 are arbitrary values in the range [0, 1].

PSO – Global Best:With entire swarm, the position of a particle which is greatly considered as best is denoted as global best (glbst). With the start topology the entire details of each particle in the swarm is gathered. Each particle has its own position in the search space, velocity and its personal best position. Using objective function, the smallest value of position is known as plbst. Among entire particles, the smallest position obtained particle is treated as global best position.

PSO local Best: The local best permits each particle to be inclined towards the best fit particle chosen from its neighbors which replicates the ring topology. The information about the position and velocity are exchanged within the nearby particles to move towards the best optimal solution space.

Local Best PSO : The lbest PSO or local best PSO approach allows each particle to be influenced by the best fit particle selected from its neighbor and it replicates a social topology of ring. To represent the local knowledge of the atmosphere the information are exchanged within the nearby particles.

E. Firefly Algorithm

Xin-She Yang in 2007[24] developed the firefly algorithm with the motivation of mimicking the behavior of flashing by fireflies for the reason of searching food [23]. Based on the brightness emitted by each firefly, other fireflies are attracted. Depending on the distance, the brightness of the attraction may increase or decrease.

For all the fireflies, the brightness referred as light intensity is compared with other fireflies. Fireflies which emits low light intensity move towards the one with high intensity to decrease the distance and update its own light intensity brightness. The high brightness firefly is considered to have least distance and thus it is declared as best solution of an objective function.

Procedure for Firefly Algorithm

Input: max_iter, population of fireflies
ffl(x_i)(i=1,2,…..,n); t = 0;
Objective functions: objffl(x), where x=(x_1,…..,x_d);
Light intensity LI_i for each firefly at ffl(x_i) is discovered using objective functionobjffl(x_i);
While(t<Max_iter)
{
For i=1 to n
{
For j=1 to n
{
If (LIj>LIi) then move firefly ffl(x_i) towards ffl(x_j)
Attractiveness of light intensity varies with the distance
Estimate new instances and update light intensity;
}
}
Fireflies are ranked accordingly and discovers current best
} {end while}

F. Firefly enabled Particle Swarm Optimization

The standard particle swarm optimization repeatedly discovers the optimum solution to the heterogeneous defect prediction using a random population of particles. The random population often fails to involve influencing instances which may give more information among independent and dependent class variable. To overcome this problem this proposed work uses firefly algorithm where the representative particles which are chosen as initial population in a random manner, is selected by firefly intelligence so the particle with high influence towards prediction process are having high chance of selection which is failed in standard particle swarm optimization. Firefly selects the global best position constitute the initial population of particles.

The initial population is partitioned into different sets and the firefly algorithm is applied to select the most prominent instances as the initial population of particles to produce optimized software defect prediction.

G. Particle Swarm Multivariate Linear Regression Classifier (PSMLRC)

To classify the dataset for heterogeneous cross project dataset the Particle Swarm Multivariate Linear Regression Classifier (PSMLRC) is modelled which performs the regression task and the parameters are fine tuned using particle swarm intelligence. This PSMLRC models a target prediction value (i.e) class label defect or defect free, based on the independent variables. It is used to find the relationship among metrics (features) and prediction.

The PSMLRC does the process to predict the dependent variable value either as defect or non-defect related to a given independent variables or metrics (m). so, this model discovers the linear relationship among input (m) and the output(C). Here the metrics involved in defect prediction are considered as input (m) and output (defect or defect-free) is denoted as C is the class label of an instance. The linear regression line is the best fit line for this model, As the input uses several metrics to predict the outcome of a different variable, this model uses multivariate linear regression [22, 23]to determine relationship among multiple independent variables and the class label known as dependent variable.

The prediction of defect or defect free software is formulated as follows:

Pred(X) = D + (E1 * V1) + (E2 * V2)+(E3 * V2)+(E4 * V4)…..+(En* Vn)

Where n is the number of metrics used as independent variables, Pred(x) is the predicted output of this PSMLRC model, D is the Y-intercept, V_1-V_n are the independent variables or metrics known as predictors, $E_{1,..}E_n$ are the regression coefficients. Here the particle swarm optimization fine tunes the parameter values of Y-intercept and the regression Coefficients to produce optimized result.

Algorithm: Firefly enabled Particle Swarm Multivariate Linear Regression Prediction Model

Step 1: Parameters Initialization for PSO

- Assign the population size for particle to be involved (psz)
- Assign maximum iteration (mxi)

Step 2: Parameter Initialization for Firefly

- **max_iter, population of fireflies**, t = 0;
- objffl(x), where x=(x_1,…..,x_d);
- Light intensity LI_i for each firefly at ffl(x_i) is discovered using objective function objffl(x_i);
- For each set 1 to psz
- While (t< mxi)
- For q = 1 to d
- For r = 1 to d
- If (LIj>LIi) then move firefly ffl(x_i) towards ffl(x_j)
- Attractiveness of light intensity varies with the distance
- Estimate new instances and update light intensity
- End for r
- End for q
- Fireflies are ranked and find the global best particles and discover its position
- End while
- End for each

Step 3: Apply the particle swarm optimization for fine tuning the parameters of multvariate linear regression using the global best particles

- Set generation = 1
- Randomly select the D and E(1..n) in the given valuing ranges.
- Calculate the new position pos_iandvelocity Vlc_i of i-th particle and then input instances of heterogeneous defect dataset independent variables V_i into the MLR model for prediction
- The parameters D and E(1..n) are fine-tuned using particle swarm optimization

Step 4: Prediction using multivariate linear regression classifier

- The forecasting of the class label CL is done as follows:

$$Pred(CL) = D + (E1 * V1) + (E2 * V2) + (E3 * V2) + (E4 * V4) \ldots\ldots + (En * Vn)$$

Step 5: Generation of Off Spring

- Global best value is generated according to Equations 1 and 2 after updating the position value.
- Global best value is inputted into the MLR model and fitness function value is calculated again
- Set generation = gen + 1

Step 6: Iteration Stops

- If generation = max- iter then stop the process and parameters of the MLR model are finally obtained.
- Else go back to Step 2.

Output: Heterogeneous dataset predicted as defect or defect-free.

From the proposed algorithm as mentioned above the particle swarm multivariate linear regression model is used for predicting the heterogeneous defect project dataset as either defect or defect-free. The parameters of the multivariate linear regression prediction model is fine-tuned using the particle swarm optimization and the population of particles involved in fine tuning is selected by the fireflies so that the best population among the swarm is utilized to achieve most promising result in defect prediction on heterogeneous cross project dataset.

V. RESULTS AND DISCUSSIONS

The proposed model uses two state approach for optimized heterogeneous defect prediction among software projects by introducing the boosted relief feature subset selection and firefly enable particle swarm multivariate linear regression model for classification and prediction of defects in heterogeneous cross project dataset. The proposed model is deployed using matlab software and their simulation results are analyzed in this section. This work used five different public open repositories of software defect datasets [17,18,19] involved in heterogeneous cross project software defect prediction. The detailed characteristics of each dataset are given in the table - I.

Table-I illustrates five groups with six different dataset, as this paper focuses on heterogeneous based cross project defect prediction, this proposed work doesn't perform software defect prediction across same metric set of dataset. The prediction variable of each dataset explains the characteristic as buggy or clean. By using the relief feature subset selection, median value is taken as matching metrics so that heterogeneous defect prediction has been achieved. In AEEM group [18], EQ dataset is used which consist of 61 metrics which are related to previous defect metrics, object-oriented metrics, entropy metrics and source code chum are taken into the account. The Velocity-1.4 dataset belongs to Morph group [17], which consist of 20 metrics which are related to CK metrics, OO metrics and Mccabe's cyclomatic metrics.

From SOFTLAB group, Ar1 dataset is used for heterogeneous cross project defect prediction, which consists of 121 instances with 29 metrics. Both SOFTLAB [17] and NASA [19] contains proprietary datasets from Turkish and NASA Company. These both groups dataset used Hallstead and McCabe's metrics but NASA additionally uses complexity measure like percentage of comment, parameter count, etc. From NASA, CM1 and PC2 are used with number of instances 344 and 1585 respectively. The CM1 consist of 37 metrics and PC2 consist of 36 metrics. Relink group is used to improve the defect prediction by increasing the quality of the software defect data with 26 code complexity measure, in this work Apache dataset with 194 instances and 26 metrics are used.

Evaluation Metric

To determine the performance of the proposed heterogeneous based cross project defect prediction Boosted Relief+FFLY-PSMLRC, three different evaluation metrics are used. They are precision, recall and accuracy.

Precision (prcs)

It is the ratio of total number of instances correctly predicted as buggy instances to the total number of instances predicted as buggy in the heterogeneous cross project defect dataset.

$$prcs = \frac{\text{Total number of instances correctly predicted as buggy}}{\text{Total number of instances predicted as buggy}}$$

Table -I: Five different group of Software with six different Defect Datasets

Group	Dataset	No of instances	No of metrics	Prediction Variable
AEEEM	EQ	324	61	class
MORPH	Velocity-1.4	196	20	class

SOFTLAB	Ar1	121	29	Function
NASA	CM1	344	37	function
	PC2	1585	36	function
ReLink	Apache	194	26	file

Recall (rcl)

It is the ratio of total number of instances correctly predicted as buggy to the total number of instances that are actually buggy in the heterogeneous cross project defect dataset.

$$rc1 = \frac{\text{Total number of insances correctly predicited as buggy}}{\text{Total number of instances actually buggy}}$$

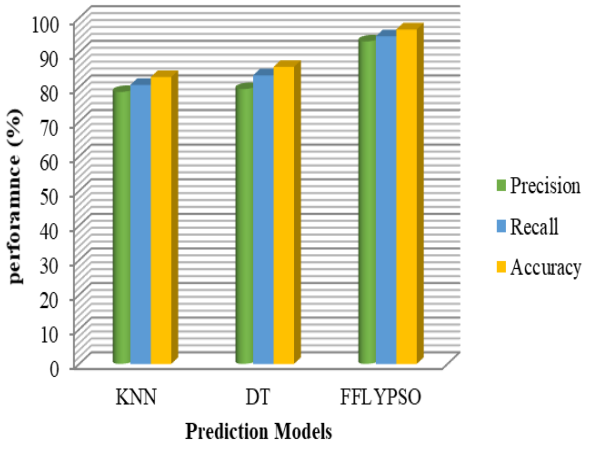
Accuracy (Acc)

It is the ratio of sum of instances that are predicted as truly buggy and clean to the total number of instances in the heterogeneous cross project defect dataset

$$Acc = \frac{\text{Sum of number of instances correctly predicted as buggy and clean}}{\text{Total number of instances in dataset}}$$

Table - II: Performance Comparison of NASA Group dataset CM1 and SOFTLAB group dataset Ar1

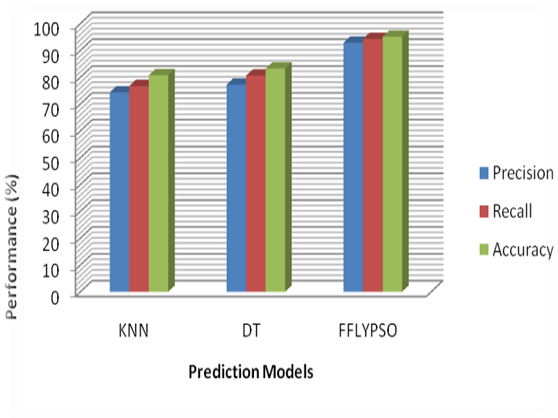
Prediction Models	Precision	Recall	Accuracy
KNN	78.6	80.64	82.92
DT	79.43	83.4	85.86
FFLY-PSMLRC	93.2	94.7	96.68



From the Table - II it is observed that two different software defect datasets namely CM1 and AR1 is used of performing the heterogeneous project defect prediction, handling dissimilar metrics to determine the defect in software project is very challenging task. By using the similarity measure among the source and the destination dataset the most similar instances are considered for prediction process. With the presence of vagueness in such heterogeneous dataset using conventional model like k-nearest neighbor and decision tree produces worst result because of the complexity in predicting the buggy or clean module with heterogeneous environment. So, this work introduced firefly-based particle swarm optimization which overcomes the vagueness by applying best fitness function to determine the appropriate samples of dataset which can increase the accuracy of prediction in software defect discover more efficiently than the other two models.

Table - III: Performance Comparison NASA group dataset PC2 and RELINK group dataset Apache

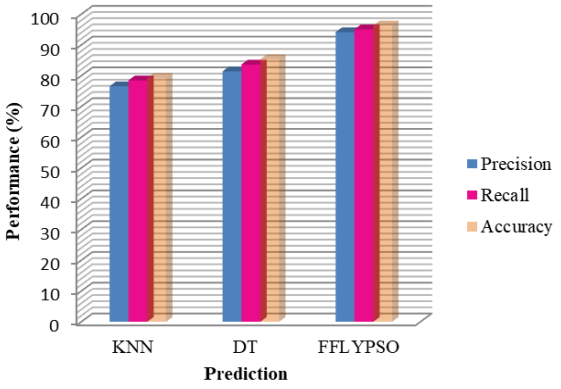
Prediction Models	Precision	Recall	Accuracy
KNN	74.31	76.68	80.67
DT	77.24	80.58	83.19
FFLY-PSMLRC	92.83	94.29	95.07



From the Table - III it is experimental that NASA group dataset CM1 and SOFTLAB group dataset Ar1 were involved in heterogeneous cross project defect prediction, handling disparate metrics to fix the defect in software project is very challenging task. By using the similarity measure among the source and the destination dataset the most similar instances are considered for prediction process. With the presence of vagueness in such heterogeneous dataset using conventional model like k nearest neighbor and decision tree produces worst result because of the complexity in predicting the buggy or clean module with heterogeneous environment. So, this work presented firefly based particle swarm optimization which overcomes the vagueness by applying best fitness function to determine the appropriate samples of dataset which can increase the accuracy of prediction in software defect discover more efficiently than the other two models.

Table - IV: Performance Comparison AEEM group dataset EQ and MORPH group dataset Velocity-1.4

Prediction Models	Precision	Recall	Accuracy
KNN	76.58	78.49	79.27
DT	81.43	83.64	85.36
FFLYPSMVLR	94.16	95.08	96.35



The resultant table and the figure expose the possibility of heterogeneous defect prediction among different metrics of dataset namely **AEEM group dataset EQ and MORPH group dataset Velocity-1.4** is well handled by the proposed model of Firefly-Based Particle Swarm Multivariate Linear Regression (FFLYPSMVLR) optimization. This is because, determining the most similar instances of two different datasets are done by the process of global best population selected by the firefly algorithm and the searching of the instances are done using particle swarm optimization with in the global best population. As the metrics used by two different datasets are entirely different, direct comparison cannot be done so the source dataset features space is converted to the size of the target dataset feature space. Source dataset is used as training dataset and target dataset is used as testing dataset. The FFLYPSO algorithm performs defect prediction more optimally then the other two standard models KNN and Decision Tree.

VI. CONCLUSION

This work insists the importance of heterogeneous based cross project defect prediction to overcome the issue of handling low volume of defect patterns in software project dataset. This presented model works under two different phases such as Dimensionality reduction and prediction of software defects using heterogeneous database. For Dimensionality reduction the boosted relief feature selection model is used and the basic qualities of the relief algorithm is improved using the boosting methodology. The prediction process is carried out by using Firefly Enabled Particle Swarm Optimization (FFLYPSO), this model uses firefly's knowledge to select the population of instances to produce optimized result instead of random selection. The multivariate linear regression is used for classification and their parameters are fine-tuned by the global and local best search mechanism of particle swarm optimization. The simulation results proved the performance of the developed model FFLYPSO achieves precise accuracy while comparing the existing approaches. In future, different datasets can be used for analysis, with varying metaheuristic algorithms.

REFERENCES

1. Z. He, F. Shu, Y. Yang, M. Li, and Q. Wang, "An investigation on the feasibility of cross-project defect prediction," Automated Software Engineering, vol. 19, no. 2, pp. 167–199, 2012.
2. D. Gunarathna, B. Turhan, and S. Hosseini, "A systematic literature review on cross-project defect prediction," Master's thesis, University of Oulu - Information Processing Science, Oct. 2016.
3. S. Herbold, A. Trautsch, and J. Grabowski, "Global vs. local models for cross-project defect prediction," Emp. Soft. Engin., pp. 1–37, 2016.
4. J. Nam, W. Fu, S. Kim, T. Menzies and L. Tan, "Heterogeneous Defect Prediction" in IEEE Transactions on Software Engineering, vol. 44, no. 09, pp. 874-896, 2018.
5. Ming Cheng, Guoqing Wu, Min Jiang, Hongyan Wan, Guoan You, Mengting Yuan, Heterogeneous Defect Prediction via Exploiting Correlation Subspace, International Journal of Software Engineering and Knowledge Engineering, Vol. 26, No. 09n10, pp. 1571-1580 (2016).
6. D. Ryu, J.-I. Jang, and J. Baik, "A transfer cost-sensitive boosting approach for cross-project defect prediction," Software Quality Journal, vol. 25, no. 1, pp. 1–38, 2015.
7. T. Zimmermann, N. Nagappan, H. Gall, E. Giger, and B. Murphy, "Cross-project defect prediction: A large scale experiment on data vs. domain vs. process," in Proceedings of the Joint 12th European Software Engineering Conference and 17th ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC-FSE'09, pp. 91–100, August 2009.
8. P. He, B. Li, X. Liu, J. Chen, and Y. Ma, "An empirical study on software defect prediction with a simplified metric set," Information and Software Technology, vol. 59, pp. 170–190, 2015.
9. B. Turhan, T. Menzies, A. B. Bener, and J. Di Stefano, "On the relative value of cross-company and within-company data for defect prediction," Empirical Software Engineering, vol. 14, no. 5, pp. 540–578, 2009.
10. Y. Ma, G. Luo, X. Zeng, and A. Chen, "Transfer learning for cross-company software defect prediction," Information and Software Technology, vol. 54, no. 3, pp. 248–256, 2012.
11. J. Nam, S. J. Pan and S. Kim, "Transfer defect learning," Proceedings of the 35th International Conference on Software Engineering. San Francisco, 2013, pp. 382-391.
12. T. Hall, S. Beecham, D. Bowes, D. Gray, and S. Counsell, "A systematic literature review on fault prediction performance in software engineering," Software Engineering, IEEE Transactions on, vol. 38, no. 6, pp. 1276–1304, 2012.
13. S. Wang and X. Yao, "Using class imbalance learning for software defect prediction," Reliability, IEEE Transactions on, vol. 62, no. 2, pp. 434–443, 2013.
14. T. G. Grbac, G. Mausa, and B. D. Basic, "Stability of software defect prediction in relation to levels of data imbalance," In SQAMIA, 2013, pp. 1–10.
15. Jaechang Nam, Wei Fu,, Sunghun Kim, Tim Menzies,, Lin Tan, Heterogeneous Defect Prediction, IEEE Transactions on Software Engineering, vol. 44, pp. 874-896, Sept. 2018.
16. Y. Li, Z. Huang, Y. Wang, and B. Fang, "Evaluating Data Filter on Cross-Project Defect Prediction: Comparison and Improvements", IEEE Access, vol. 5, pp. 25646–25656, 2017
17. X. Y. Jing, F. Wu, X. Dong, F. Qi, and B. Xu, "Heterogeneous cross-company defect prediction by unified metric representation and cca-based transfer learning," In Proceedings of the 10th Joint Meeting on Foundations of Software Engineering, 2015, pp. 496–507.
18. M. D'Ambros, M. Lanza, and R. Robbes, ``An extensive comparison of bug prediction approaches," in Proc. 7th IEEE Work. Conf. Mining Software Repositories (MSR), May 2010, pp. 31_41.
19. http://promise.site.uottawa.ca/SERepository/datasets-page.html.
20. Kennedy, J.; Eberhart, R. Particle swarm optimization. IEEE Proc. Int. Conf. Neural Netw. 1995, 4, 1942–1948;
21. David .A. Freedman Statistical Models: Theory and Practice, Cambridge University Press. p. 26, (2009).
22. Rencher, Alvin C.; Christensen, William F. (2012), "Chapter 10, Multivariate regression – Section 10.1, Introduction", Methods of Multivariate Analysis, Wiley Series in Probability Statistics, 709 (3rd ed.), John Wiley & Sons.

23. Amarita Ritthipakdee,Arit Thammano, Nol Premasathian, Duangjai Ji t kongchuen, Firefly Mating Algorithm for Continuous Optimization Problems, Computational Intelligence and Neuroscience, 10 pages, Volume 2017,.
24. Yang,X.S.(2008). Nature-Inspired Metaheuristic Algorithms, Luniver Press. ISBN 978-1-905986-10-1.

AUTHOR'S PROFILE

Mrs. N. Kalavani, is a Research Scholar, Department of Computer Science, Kongunadu Arts and Science College, Coimbatore. Her research interest includes software engineering and datamining. She is doing her research work under the guidance of Dr.R.Beena, Associate Professor and Head, Department of Computer Science, Kongunadu Arts and Science College, Coimbatore. She has 15 years of Teaching Experience and published articles in refereed journals. Email : kalaivhani@gmail.com

Dr. R. Beena, is an Associate Professor and Head in the Department of Computer Science, Kongunadu Arts and Science College, She has received her PhD degree from Bharathiar University, Coimbatore. Her research interest includes software engineering, datamining and networking concepts. She has 20 years of Teaching Experience and published many articles in international journals and conferences. Email : beenamridula@yahoo.co.in