JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 410-420
https://publishoa.com
ISSN: 1309-3452

Enriched Hybrid Recursive Feature Elimination Algorithm with Learning Vector Quantization Classifier for Heterogenous Cross Project Defect Prediction

[1] J. Deepa lakshmi, [2] Dr. M. Chandran

[1]Ph.D Research Scholar, [2]Associate Professor
[1]deepalakshmi.j03@gmail.com, [2]onchandran@gmail.com
[1, 2,] Dept. of Computer Science,
[1,2,] Sri Ramakrishna Mission Vidyalaya College of Arts and Science, Coimbatore, India

ABSTRACT

Objectives: The ultimate objective of this study is to enhance software defect prediction in the presence of insufficient training instances for defect cases in the majority of real-world scenarios. This paper focuses on overcoming the issue of class imbalance by constructing an enriched model in Heterogenous Cross Project Defect Prediction (HCPDP). In this proposed work the source project and target project with different feature metric and size are used for HCPDP.

Methods: By using a hybrid recursive feature elimination approach, the feature size of the source project is condensed to match the feature size of the target project. This is accomplished by integrating the fuzzy linear support classifier which represents the instances in terms of membership. The features which are more informative in discrimination among clean and buggy module is preserved and the least scored features are eliminated from the feature list. The weighted Jaccard Index is used for finding the dissimilarity among the source and target projects. Those computed instance of values are used for predicting the software defect by inducing learning vector quantization.

Findings: As software usage grows tremendously, Heterogenous Cross Defect Prediction has emerged as an essential study area in software engineering. Despite the fact that there are numerous literatures accessible, class imbalance and over fitting are the most significant issues that affect the accuracy rate of prediction models. The newly developed Hybrid Recursive Feature Elimination with Learning Vector Quantization uses two different software projects with different feature size. By adopting hybrid recursive feature elimination, the large dataset of the source project is reduced to the size of the target project, and similar instances between datasets are used to improve the accuracy of defect-prone modules using learning vector quantization.

Novelty: On six different heterogeneous projects for software defect prediction, the proposed Hybrid Recursive Feature Elimination with Learning Vector Quantization (HRFE-LVQ) for HCPDP outperforms standard classification models.

Keywords: Software Defect Prediction, heterogenous cross project defect prediction, hybrid recursive feature elimination algorithm, weighted Jaccard Index, learning vector quantization.

Introduction

Software products are growing in size and complexity in tandem with the rapid growth of their features and needs. [1]. typically, software is an assemblage of a big corpus with thousands of code lines. Maintaining excellent software quality is a significant challenge in the real process of software development. The software defect prediction is a popular approach developed to deploy restricted testing resources wisely and reduce the chances of post-release faults. The basic strategy of software defect prediction models is to use machine learning algorithms to construct a classification model from previous datasets and then forecast if new software modules include problems. By focusing on those projected defect-prone modules, accurate prediction findings can help allocate suitable testing resources [2].

Existing software defect prediction methods works well when there is adequate historical data are available. But in many situations, building a WPDP model with insufficient historical data is extremely difficult for a new project [3]. Hence, Heterogenous Cross-project defect prediction (HCPDP) is an effective method for resolving the issue that has been suggested and used by the research community. But it is not always possible for defect data to originate from independent projects with same features. It is very challenging to achieve HCPDP is since there is no equivalent relationship between the two heterogeneous feature sets that have no common instances [4].Due to the presence of multiple unnecessary Cross Project (CP) modules within the data, the majority of existing HCPDP models produce poor results. The class imbalance occurs when the number of samples with defect is very less. The learning rate of the existing prediction models highly depends on the dataset involved in Heterogenous defect prediction.

This research work aims to construct a novel heterogenous cross project defect prediction by inducing the hybrid Recursive feature elimination which is the integration of Learning vector Quantization classifier and recursive model with a neural network. This proposed work correctly maps the data pattern form features space to the class space. The devised Learning vector Quantization has the strong and adaptive learning ability for defect prediction among heterogenous cross projects and it is explained in detail in the following sections.

Related Work

Yu Zhao et al., [5] Manifold Feature Transformation is used to achieve cross-project defect prediction in their work.The feature space is reduced to match the distribution of source and target using manifold space. The naïve bayes classifier is used for predicting the software defect detection.

Chen et al., [6] they offered a unique SDP model for fault prediction that incorporates class overlap elimination with ensemble imbalance training. Initially, the overlapping non-defective samples are removed using the neighbor cleaning method. The entire dataset is then arbitrarily under sampledrepeatedly to provide fair subsets for training different classifiers. Lastly, the AdaBoost technique is used to combine these individual classifiers to create a complete prediction system.

Bowes et al., [7] examined the specific defects predicted by classification techniques and examine the degree of prediction uncertainty caused by such classifiers. They assess the performance of four different classification models for defect prediction in NASA data sets using a sensitivity analysis.

Xu et al [8] it uses the domain coping and adaptive method to integrate the information from two realms into a corresponding subspace with a lower size, then uses the dictionary supervised learning to quantify the contrast between the different mapped regions of information.

Hosseini et al [9] under took a consolidated research study to determine the trend of measurements, data techniques, linked outcomes and systems. They focuse don fundamental research to comprehend the contextual and the matic of existent CPDP activities.

Xinglong et al [10] constructed a transfer learning model which acquires knowledge from a domain and it uses to detect defects on the other domain. It generates a projective matrix among source and target projects which are heterogeneous to create the similar distribution.

Methodology: Hybrid Recursive Feature Elimination and Learning Vector Quantization for HCPDP

In this proposed hybrid recursive feature elimination is developed for selecting the potential features which match with the dimension of the target project to perform heterogenous defect prediction. In real-world datasets, metrics are measured differently, hence the values of the dataset are standardized using min-max normalization to ensure that each metric is treated equally. Learning vector quantization is designed to predict software defect, and six distinct public open archives of software defect datasets are used in HCPDP. Because the source and target projects have different feature sizes, they cannot be used together to conduct prediction. As a result, a novel hybrid recursive feature elimination technique is developed to minimize the size of the source project feature to that of the target project. The obtained reduced feature subset of source project and target project difference is determined by applying Jaccard similarity measure. The discovered difference values are used for prediction process. The linear vector quantization is used for predicting the heterogenous cross project defect prediction as shown in the figure 1.

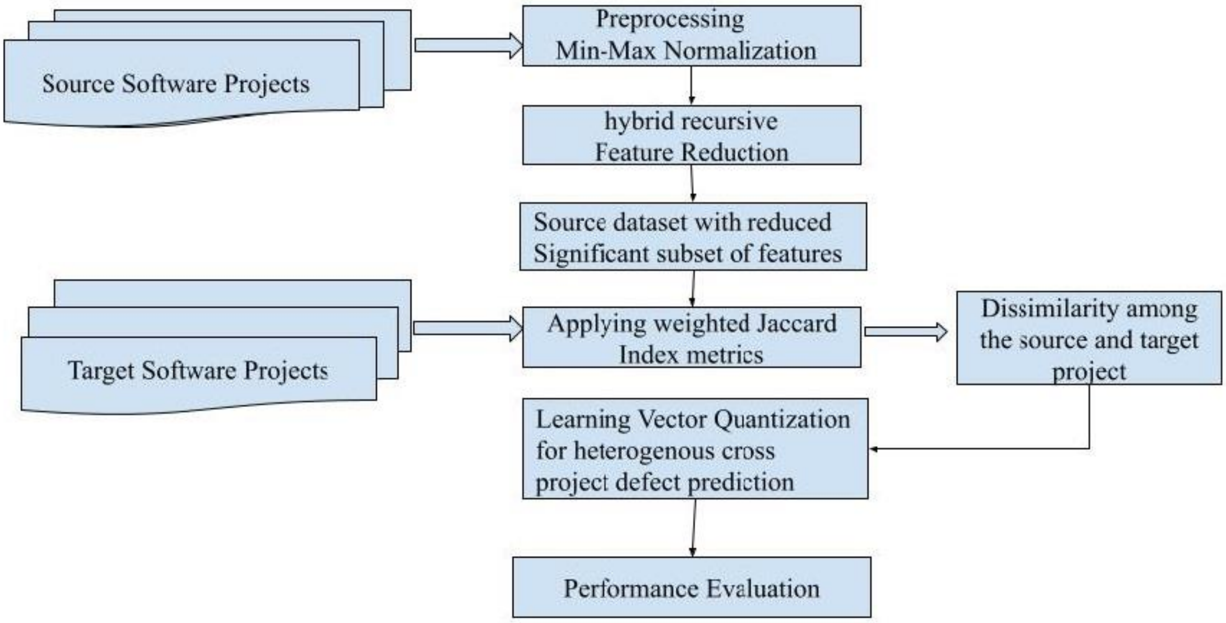

Figure 1 Overall Architecture of Hybrid Recursive Feature Elimination and Linear Vector Quantization for HCPDP

Figure 1 Overall Architecture of Hybrid Recursive Feature Elimination and Linear Vector Quantization for HCPDP

The detailed description of each process is explained in the following sections.

Dataset Description

As indicated in table 1, five different sets with six unique projects have been used in this study for heterogeneous cross project defect prediction.

Table 1 Dataset Description

Dataset	Projects	Instances	Features	Granularity
AEEM	EQ	324	61	class
	JDT	997	61	class
MORPH	Tomcat	858	10	class
NASA	MW1	403	37	function
RELINK	SAFE	56	26	file
SOFTLAB	ar4	107	30	function

D'Ambros et al. [11], created AEEM dataset and its two projects EQ comprised of 324 instances and JDT comprised of 997 instances with 61 features. It features are metrics of entropy change, previous defect, churn of source code and source code with its label granularity as clean or buggy. The tomcat project which belongs to MORPH dataset is collected from PROMISE repository, with 10 features of CM based metrics and 858 instances. Wu et al., [12] constructed RELINK dataset and its SAFE project with 56 instances and 26 features of code complexity metric is used for heterogenous cross project defect prediction. NASA dataset is one of the most popular benchmark datasets, its MW1 project is collected from NASA experts [13] with 403 instances and 37 features.

Data Preprocessing

The six different projects are normalized to treat all the features with equal importance. Min-Max normalization is used to convert different range of feature value to common range [0,1]. It is formulated using the equation 1

$$\mathrm{P(I)} = \frac{\mathrm{I-Min(I_{1\ldots n})}}{\mathrm{Max(I_{1\ldots n})-Min(I_{1\ldots n})}} \text{ eq (1)}$$

Where I refer to an instance in the dataset, Min and Max are minimum range of value rand maximum range of value of the instances in the concern feature.

Hybrid Recursive Feature Elimination

After preprocessing the dataset, the source project undergoes feature reduction process using hybrid recursive feature elimination algorithm. The HRFE identifies the weakest attributes and eliminates it from the feature set until it reaches enumerated number of significant subset of features. This algorithm initially searches for a feature subset with all features involved in the training dataset and effectively removes least contributed features until they reach desired size of features [14]. This is accomplished by adapting a machine learning model as the important portion of the model, features are ranked by their importance, least important features are discarded from the feature set and finally refitting the model. This will be iterated until it reaches a particular size of features equaling to the target project. While using HRFE it needs the information of predefined number of attributes to keep, here it should be equal to the feature size of target project.

The problem of feature selection method can be represented in two different aspect such as

- With the p<n dataset, discover p features that offer smallest anticipated generality error λ
- Given a maximum acceptable generality error, discover smallest p.

In these two problems the generality error which is unknown has to be computed. The proposed Linear Support Vector Classifier Recursive Feature Elimination algorithm discovers criteria for ranking by integrating Fuzzy linear support vector classifier for understanding weight vector.

The linear SVC encodes the HCPDP dataset (i.e) source projects which are considered as training data with binary classification problem is denoted as clean +1, buggy -1. It is represented mathematically as $\{\{a_i, y_i\}_{i-1}{}^{T}, a_i \in D^p, y_i \in \{-1, +1\}\}$,

Where a represents the source project dataset, yi denotes the class label D is the dataset with p size feature subset

The hyperplane discriminated the clean and buggy class is denoted:

$$Wt\ 1\ a + s = 0$$

Where wt signifies the weight vector, input dataset is denoted by a and the s refers to the bias parameter in the hyperplane. The parameter s is used for confirming that hyperplane is place in the correct position after movement done horizontally. Hence after training wt., the bias value is resolute. When the support vector classier is implied, the hyperplane is considered as decision function and it is formulated as

$$f\ (a) = \mathrm{sign}(wt \cdot a + s)$$

The linear support vector classifier obtains the maximized marginal distance as hyperplane to enhance the discrimination among the two classes of the dataset. The quadratic problem is considered of optimizing the hyperplane functionality and finally the function for classification is denoted as

$$f(a) = \mathrm{sig}(\textstyle\sum_{i=1}^{n} y_{i\lambda_i^*}(a * a_i) + s^*)$$

If the value of f(a)> 0, then it means the instance belongs same category as samples marked with buggy modules, otherwise it belongs to clean modules.

Algorithm: Feature selection of HRFE- LVQ

Input: Training dataset $\{a_i, y_i\}_{i=1\ldots N}$

Output: Ranked feature list RL

```
• S= {1,2…D}
• RL ≠θ
• While S is not empty
    o Restrict the features of $a_i$ to the remaining s
    o Train SVC to get weight vectors
```

- Compute the ranking condition $C_k = wt_k^2$
- Look for features with smallest value of $C_{k,}$ called feature m
- Add feature m into RL
- Remove feature m from S

End

The SVC is trained to discover the weight vectors of the features. Compute ranking criteria for all features. With the obtained weight vectors, it eliminates the lowest weight squares features from the feature list. Finally, it chooses the feature sub set attributes with the highest scores as prospective subsets of features by excluding other attributes that are not members of this subset. The workflow of the enriched hybrid recursive feature elimination process is depicted in the figure 2.

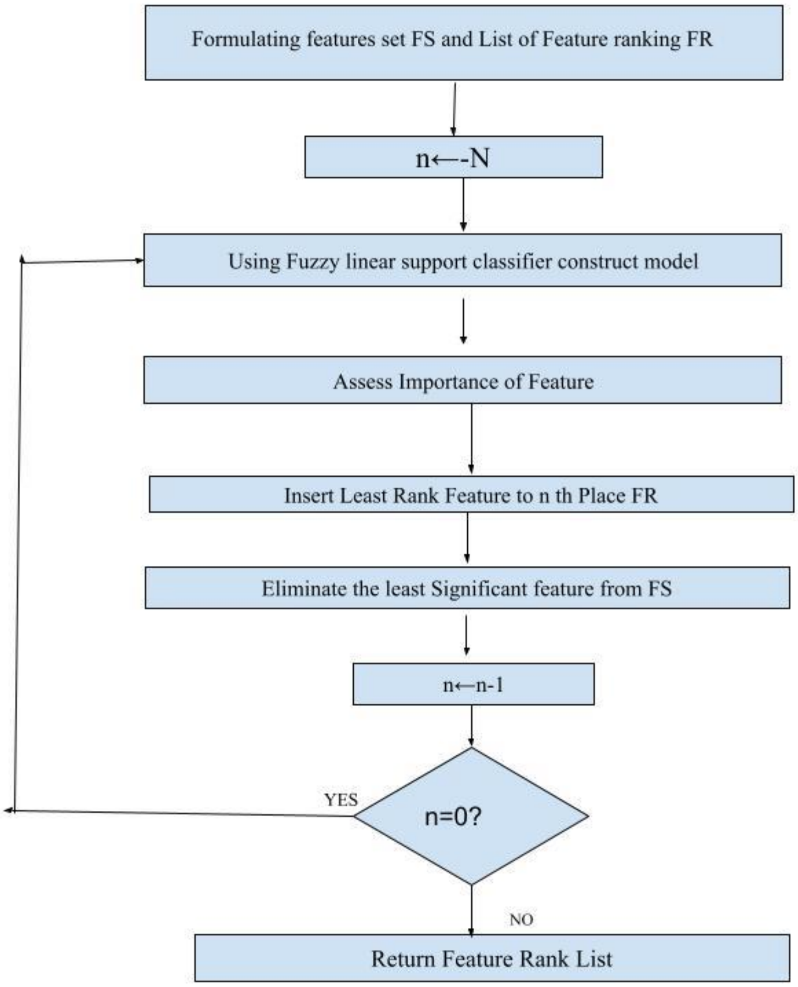
Formulating features set FS and List of Feature ranking FR
n←-N
Using Fuzzy linear support classifier construct model
Assess Importance of Feature
Insert Least Rank Feature to n th Place FR
Eliminate the least Significant feature from FS
n←n-1
YES
n=0?
NO
Return Feature Rank List

Figure 2 Hybrid Recursive Feature Elimination

Weighted Jaccard Index based Dissimilarity measure among Source project and Target Project

Once the source project feature size is reduced to match the target project using Fuzzy linear support vector classifier. The dissimilarity among the source and target project is determined using Jaccard Index [17], which is a statistic-based measure used for assessing the diversity of instance sets of two different projects to accomplish heterogeneous cross project defect prediction.

Let assume that $sp = \{sp_1, sp_2, \ldots, sp_n\}$ $tp = \{tp_1, tp_2, \ldots, tp_n\}$are two different source and target projects their dissimilarity is measure as

$$J_w(sp,tp) = 1 - \frac{\sum_i \min(sp_i, tp_i)}{\sum_i \max(sp_i, tp_i)}$$

Linear Vector Quantization (LVQ) based Heterogenous Cross project Defect Prediction

Linear Vector Quantization belongs to one of the artificial neural network models [18]. It is developed based on the biological inspiration of neural systems functionalities. Its prototype is a supervised learning model and a competitive learning algorithm is involved in training the network. In general, LVQ has two layers they are input layer and output layer as displayed in the figures 3 and 4.

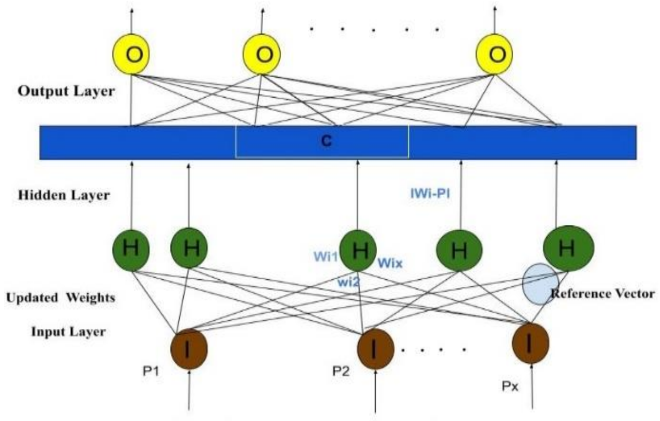

Figure :3 Network Structure of Linear Vector Quabtization

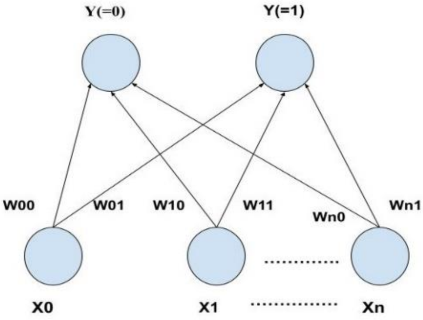

Figure 4 : Simplified View of LVQ

To perform Heterogenous Cross Project Defect Prediction, in this work LVQ model is adapted. The LVQ network architecture for classification of patterns is accomplished by designing three layers as described below.

- **Input Layer:** It receives the instances of HCPDP dataset as input vector with n instances and m features.
- **Competitive Layer:** It learns to classify the input's based on their patterns
- **Output or Linear Layer:** It converts the classes of the competitive layer into the user-specified target categories in this work is defective (true/ false) or Class (buggy/clean).

The LVQ network belongs to forward neural network which accomplishes advantage of attaining global optimal without normalization by unswervingly manipulating the variance among input vector and the competitive layer. This work's input layer consisted of a variable number of neurons depending on the project's feature size. Learning is used by the neurons in the competition layer to classify the input vector [19]. The output layer which is linear in nature is made up of two neurons that correspond to the buggy and clean classes.LVQ picks the endearing neuron based on the least distance between the input vector and the reference vector during the training and testing phase, so that the neuron's output is 2 and the other's output is 0. The reference vector parameter is updated mathematically as expressed in the equation

$$\Delta \mathrm{wt}_{i,j} = \begin{cases} +\vartheta\left(a_i - \mathrm{wt}_{i,j}\right), \text{forcorrectclassifications} \\ -\vartheta\left(a_i - \mathrm{wt}_{i,j}\right), \text{forincorrectclassifications} \end{cases}$$

Where $wt_{i,j}$ refers to the weight value of the referctor vector, a_i refers to the input value.

Assume that size of input data is (l, n) where n refers to the training instances and l belongs to features of each instances and the class label with the size of (n,1) [20]. To begin, it initialises the weights of size (l,d) from the first 'd' training instances with labels, hence it should be removed from all training dataset. The number of classes is denoted by d. Then repeat over through the remaining input data and updating the winning vector using the euclidean distance measure for each training instance. The weight updation rule is formulated as shown in the equaiton

$$wt_{ij} = wt_{ij}(old) - \alpha(t)\ 1\ (a_i^k - wt_{ij}(old))$$

where learning rate at time t is denoted by α, the winning vector is j and i refers to i^{th} feature of the kth training instance. Once training phase is completed, the trained weights are used for classifying unknown instances during testing phase [21]. The observed output which is generated by the proposed LQV is compared with the actual output to discover the accuracy of the model.

Algorithm: Hybrid Recursive Feature Elimination with Linear Vector Quantization for Heterogenous Cross Project Defect Prediction

Input: Source Project sp(m,n), Target Project tp(p,v) where m ≠p, n and v are number of instances.

Output: Defect Prediction {Clean, Buggy}

Procedure:

```
Begin
// Covert the raw dataset of sp and tp into fuzzy value dataset by determining membership value of each instances attribute values.
    $\mu_D(a) = \begin{cases} 0; & a \leq e \\ \left(\frac{a-e}{f-e}\right) & e < a \leq f \\ \left(\frac{g-e}{g-f}\right) & f \leq a < g \\ 0; & a \geq g \end{cases}$
  • Sp= {1,2…n}
  • RL ≠θ
  • While sp is not empty
      o Restrict the features of $a_i$ to the remaining sp
      o Train SVC to get weight vectors
      o Compute the ranking condition $C_k = wt_k^2$
      o Look for features with smallest value of $C_{k,}$ called feature m
      o Add feature m into RL
      o Remove feature m from sp
// Determine the dissimilarity among the instances of sp and tp using weighted Jaccard Index
  • With reduced feature subset ‘q’ of sp , discover the distance among the sp and tp by applying weighted Jaccard index
        $J_w(sp,tp) = 1 - \frac{\sum_i \min(sp_i,tp_i)}{\sum_i \max(sp_i,tp_i)}$
// Linear Vector Quantization based HCPDP
  • Initialize the weight vectors (wt)
  • While I <noe
      o Choose a training instance of dataset
      o Pick the nearest prototype to the input vector X
      o Calculate winning vector
            $\Delta wt_{i,j} = \begin{cases} +\vartheta(a_i - wt_{i,j}), forcorrectclassifications \\ -\vartheta(a_i - wt_{i,j}), forincorrectclassifications \end{cases}$
      o Updated winning vector
            $vwt_{ij} = wt_{ij}(old) - \alpha(t)\ 1\ (a_i^k - wt_{ij}(old))$
    End
  • Choose the testing dataset and classify the clean and buggy modules in the cross-project dataset
End
```

The algorithm describes the complete functionality of the HCPDP by handling the class imbalance problem [22]. For the prediction process, two distinct projects with heterogeneous feature sizes are employed, and the vast volume of the primary project is condensed using a hybrid recursive feature elimination approach based on fuzzy support vector classifiers. The difference among the two source and target project is computed using Weighted Jaccard Index. The Linear Vector Quantization is used for classifying the modules in the software project as clean or buggy.

Simulation Results and Discussion

In this section, simulation result of proposed hybrid recursive feature elimination and linear vector quantization is used for heterogenous defect prediction. The dataset is collected from PROMISE database with a six different software projects [23]. Python software is used for deploying the proposed model Hybrid Recursive Feature Elimination with Learning Vector Quantization (HRFE-LVQ). The performance of proposed model is compared with three different classification models Logistic Regression (LR), Random Forest (RF) and Support Vector Machine (SVM).

Performance Comparison of Source Project AEEM (EQ) and Target Project NASA (MW1)

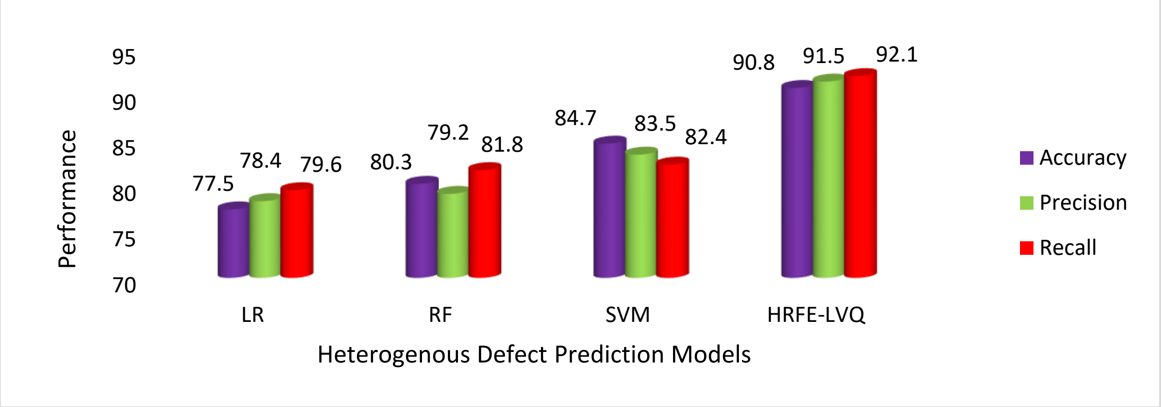

Figure 5 Comparison of Source Project AEEM (EQ) and Target Project NASA (MW1)

Figure 5depict the performance of the EQ and MW1 for heterogenous cross project defect prediction. The source project EQ feature size is reduced to match the target project MW1 by proposed hybrid recursive feature elimination model [24]. The irrelevant and least informative features of EQ project is eliminated by ranking the features. The similarity of both the source and target project is evaluated using Jaccard distance metrics. With the obtained measure the presence of defect is predicted by developing Learning Vector Quantization for Heterogeneous cross project Defect Prediction. While comparing with logistic regression, random forest and support vector machine the proposed model accomplishes highest rate of accuracy, precision, recall and F-measure for heterogenous cross project defect prediction [25].

Performance Comparison of Source Project AEEM (JDT) and Target Project SOFTLAB (ar4)

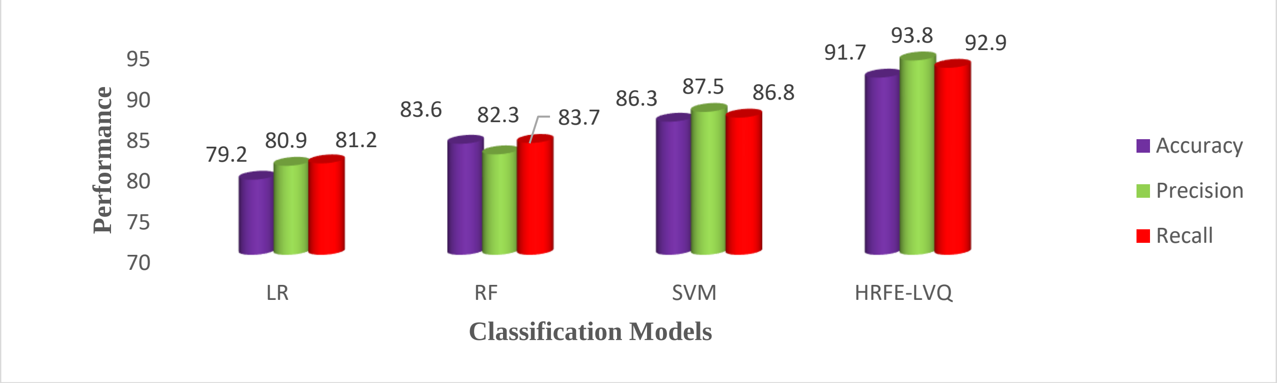

Figure 6: Comparison of Source Project AEEM (JDT) and Target Project SOFTLAB (ar4)

Figure 6 displays the heterogenous cross project defect prediction of JDT as source project and ar4target project. The accuracy, precision, recall and F-measure of newly constructed algorithm produced better result compared to SVM, logistic regression and random forest. The hybrid feature selection model tackles the problem of impreciseness in selection of significant features and eliminating irrelevant features improves the efficiency of HCPDP [26]. The classification algorithm Learning Vector Quantization improves the prediction rate more promisingly compared to linear regression, support vector machine and random forest. The existing models suffers from class imbalance and produce least performance compared to proposed model.

Performance Comparison of Source Project Relink (Safe) and Target Project MORPH (Tomcat)

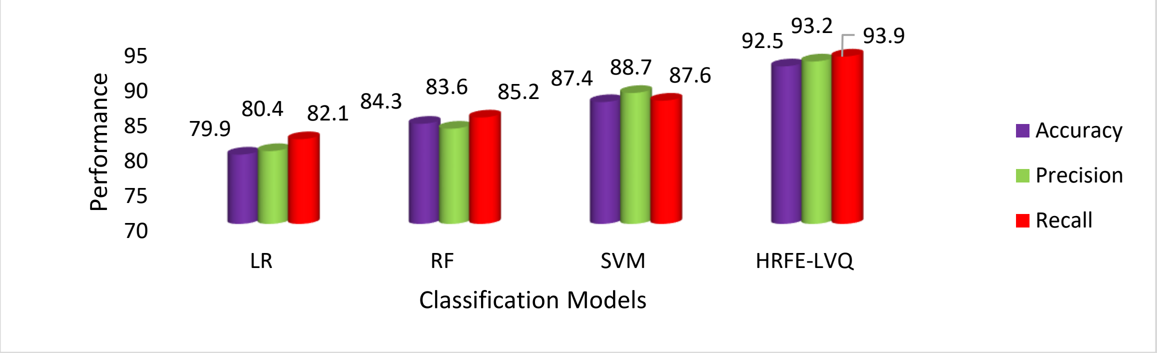

Figure 7 Comparison of Source Project Relink (Safe) and Target Project MORPH (Tomcat)

Figure 7exhibits HCPDP of Safe as source project and Tomcat as target project using four different classification models. [27]. the nature of two distinct features of source and target project is very challenging with different feature size. The source project has high volume of features it is reduced by applying hybrid recursive elimination algorithm. The irrelevant and redundant features are eliminated and top n attributes are used for prediction. The Jaccard similarity-based distance is computed among safe and tomcat projects. The performance of the proposed model produced better result in heterogenous cross project defect prediction while comparing with SVM, RF and LR [28]. The existing models suffers from over fitting problem because of using unusual heterogenous project defect prediction.

Conclusion

The primary objective of tackling class imbalance to empower the accuracy rate of heterogenous cross project defect prediction is accomplished by the newly constructed Hybrid Recursive Feature Elimination with Linear Vector Quantization. Two different projects with variant features are effectively utilized by diminishing the source project dimensionality to match with target project by devising a novel recursive feature elimination algorithm. Unlike, the traditional feature selection algorithm, this proposed recursive model determine the relevant attributes with highest score rate to be involved n prediction process. The linear vector quantization algorithm is adapted in this HCPDP for handling the issue of class imbalance among testing and training labels of defect prone modules to be predicted. Thus, the proposedHRFE-LVQ produced highest rate of accuracy with three different set of HCPDP method. The other three existing models suffers from over fitting and with less samples of testing data their learning rate is highly affected in heterogenous cross project defect prediction to enhance the software testing process.

References

1. Huda, S., Liu, K., Abdelrazek, M., Ibrahim, A., Alyahya, S., Al-Dossari, H., Ahmad, S. (2018). An ensemble oversampling model for class imbalance problemin software defect prediction. 6:24184-24195, DOI: 10.1109/ACCESS.2018.2817572.
2. Zakari, A.; Lee, S.P.; Abreu, R.; Ahmed, B.H.; Rasheed, R.A. Multiple fault localization of software programs: A systematic literature review. Inf. Softw. Technol. 2020, 124, 106312,https://doi.org/10.1016/j.infsof.2020.106312.
3. Ochodek, M.; Staron, M.; Meding, W. SimSAX: A measure of project similarity based on symbolic approximation method and software defect inflow. Inf. Softw. Technol. 2019, 115, 131–147,https://doi.org/10.1016/j.infsof.2019.06.003
4. Xu Z., Liu J., Luo X., Yang Z., Zhang Y., Yuan P., Tang Y., Zhang T. Software defect prediction based on kernel PCA and weighted extreme learning machine. Inf. Softw. Technol. 2019;106:182–200, https://doi.org/10.1016/j.infsof.2018.10.004.
5. Zhao, Y.; Zhu, Y.; Yu, Q.; Chen, X. Cross-Project Defect Prediction Method Based on Manifold Feature Transformation. Future Internet 2021, 13, 216. https://doi.org/10.3390/fi13080216
6. Chen, L., Fang, B., Shang, Z., and Tang, Y. (2018). Tackling class overlap and imbalance problems in software defect prediction. 26(1):97-125,https://doi.org/10.1007/s11219-016-9342-6

7. *Bowes D, Hall T, and Petric J, (2018). Software defect prediction: do differentclassifiers and the same defects? 26(2):525-552, https://doi.org/10.1007/s11219-016-9353-3.
8. Z. Xu, P. Yuan, T. Zhang, Y. Tang, S. Li and Z. Xia, "HDA: Cross-Project Defect Prediction via Heterogeneous Domain Adaptation with Dictionary Learning," in IEEE Access, vol. 6, pp. 57597-57613, 2018,DOI: 10.1109/ACCESS.2018.2873755.
9. Hosseini, Seyedrebvar, Turhan, et al. (2019). A systematic literature review and meta-analysis on cross project defect prediction. IEEE Transactions on Software Engineering, 45, pp. 21–147,DOI: 10.1109/TSE.2017.2770124.
10. Xinglong Yin, Lei Liu, Huaxiao Liu, Qi Wu, Heterogeneous cross-project defect prediction with multiple source projects based on transfer learning[J]. Mathematical Biosciences and Engineering, 2020, 17(2): 1020-1040, doi: 10.3934/mbe.2020054
11. Zhou Xu, Peipei Yuan, Tao Zhang, Yutian Tang, Shuai Li, Zhen Xia, Hda: Cross-Project Defect Prediction via Heterogeneous Domain Adaptation With Dictionary Learning, Volume6, 2018.
12. Abdullateef O. Balogun , ShuibBasri , SaipunidzamMahamad , Said J. Abdulkadir , Malek A. Almomani , Victor E. Adeyemo , Qasem Al-Tashi , Hammed A. Mojeed , Abdullahi A. Imam, Amos O. Bajeh, Impact of Feature Selection Methods on the PredictivePerformance of Software Defect Prediction Models: An Extensive Empirical Study,Symmetry **2020**, 12(7), 1147 , **https://doi.org/10.3390/sym12071147**
13. Peng He ,Yao He, LvjunYu,Bing Li, An Improved Method for Cross-Project Defect Prediction bySimplifying Training Data, Mathematical Problems in EngineeringVolume 6, 2018, pp 1-18,https://doi.org/10.1155/2018/2650415.
14. Hoang Luong, Huong Phan Le Trong, Nghia Duong, Tin Dang, Thuan Nguyen, Tong Nguyen, Hai. (2021). Dimensionality Reduction on Metagenomic Data with Recursive Feature Elimination, DOI:10.1007/978-3-030-79725-6_7
15. Moulton R, Jiang Y (2018). "Maximally Consistent Sampling and the Jaccard Index of Probability Distributions", International Conference on Data Mining, Workshop on High Dimensional Data Mining: 347–356, DOI
10.1109/ICDM.2018.00050.
16. Emre Akarslan, Learning Vector Quantization based predictor model selection for hourly load demand forecasting, Applied Soft Computing, Volume 117, March 2022, 108421,https://doi.org/10.1016/j.asoc.2022.108421
17. Xinglong Yin, Lei Liu, Huaxiao Liu, Qi Wu. Heterogeneous cross-project defect prediction with multiple source projects based on transfer learning[J]. Mathematical Biosciences and Engineering, 2020, 17(2): 1020-1040.
18. L. Chen, B. Fang, Z. Shang, et al., Negative samples reduction in cross-company software defects prediction, Inf. Software Technol., 62 (2015), 67–77
19. Zhou Xu, Peipei Yuan, Tao Zhang, Yutian Tang, Shuai Li, Zhen Xia, HDA: Cross-Project Defect Prediction via Heterogeneous Domain Adaptation with Dictionary Learning, IEEE. Translations and content mining, VOLUME 6, pp 57597- 57613, 2018
20. Pravas Ranjan Bal, Sandeep Kumar, Cross Project Software Defect Prediction using Extreme Learning Machine: An Ensemble based Study, Proceedings of the 13th International Conference on Software Technologies (ICSOFT 2018), pages 320-327
21. Nam, J., Fu, W., Kim, S., Menzies, T., and Tan, L. (2017). Heterogeneous defect prediction. IEEE Transactions on Software Engineering.
22. Jie Wu, Yingbo Wu, Nan Niu, Min Zhou, MHCPDP: multi source heterogeneous cross project defect prediction via multi source transfer learning and autoencoder,Software Quality Journal (2021) 29:405–430
23. Hosseini, Seyedrebvar, Turhan, et al. (2019). A systematic literature review and meta-analysis on cross project defect prediction. IEEE Transactions on Software Engineering, 45, pp. 111–147
24. HadiJahanshahi, MucahitCevik, AyseBasar, Moving from cross-project defect prediction to heterogeneous defect prediction: a partial replication study, ASCON '20: Proceedings of the 30th Annual International Conference on Computer Science and Software Engineering, November 2020, pages 133–142
25. Ahmad Hasanpour, PouryaFarzi, Ali Tehrani, Reza Akbari, Software Defect Prediction Based on Deep Learning Models: Performance Study, pp 1-10, 2020

26. Z. Tian, J. Xiang, S. Zhenxiao, Z. Yi and Y. Yunqiang, "Software Defect Prediction based on Machine Learning Algorithms," 2019 IEEE 5th International Conference on Computer and Communications (ICCC), 2019, pp. 520-525
27. Marjuni, Aris ,Adji, Teguh, Ferdiana, Ridi. (2019). Unsupervised software defect prediction using signed Laplacian-based spectral classifier. Soft Computing. 23(6), 2019.
28. R. Vashisht, S. A. M. Rizvi, "Feature Extraction to Heterogeneous Cross Project Defect Prediction," 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2020, pp. 1221-1225