International Journal of Sustainable Intelligent Technologies
ISSN: 3139-1249
IJSIT-11

ENHANCING MODEL GENERALIZATION IN CROSS PROJECT SOFTWARE DEFECT PREDICTION UNDER IMBALANCED DATA

[#1]**HUSSAIN ABRAR,** *Dept of CSE,*
[#2]**Dr.D.SRINIVAS REDDY,** *Professor, Dept of CSE,*
Vaageswari College of Engineering(Autonomous), Karimnagar, TG.

ABSTRACT: Strong machine learning frameworks that can detect problematic modules in a target project using data from many source projects, irrespective of data distribution or class balance, are necessary for cross-project software defect prediction (CPDP) using imbalanced data in order to improve model generalization. Over time, CPDP forecasts lose value due to numerous flawed cases. The majority is favored by several models. Data resampling (SMOTE and undersampling), cost-sensitive learning, transfer learning, domain adaptation, advanced techniques, and ensemble methods address this. It enhances model stability and minority class recognition. Distribution and sync alignment promote cross-field knowledge and minimize project discrepancies. This uses imbalance control and generalization to improve memory, F1-score, and AUC. Fault-prediction systems become more reliable and scalable in many software development scenarios.

Keywords: *Cross-Project Fault Prediction, Class Imbalance, Generalization, Software Quality and Machine Learning.*

1. INTRODUCTION

Cross-project software defect prediction (CP-SDP) models are improved by researchers. The same coding standards, development processes, and defect patterns are used in the instruction and testing of several traditional defect prediction techniques. More defect data may be required by younger businesses. Cross-project systems anticipate problems by using data from other projects. It is necessary to apply models from many source projects to a target project with distinct team dynamics, working conditions, and distributions.

The main problem is that CP-SDP changes the project data flow. Code complexity, coupling, coherence, and churn are influenced by architecture, programming languages, and organization. Modifications reduce model accuracy and complicate the understanding of source data. Program-setting stability and distribution alignment help to foster generality. Transfer learning, feature normalization, representation learning, and domain adaptation are helpful.

Predicting project problems is hampered by poor class matching. Problematic modules are uncommon in software source codes. Issues are exacerbated by uneven numbers. Non-defective classes are favored by unequal dataset models. As a result, most people are correct but have trouble recognizing errors. Unbalance is caused by working on numerous projects with different source and target datasets. Cost-sensitive learning, resampling (minority faults or majority events), deceptive data, and customized loss functions may all favor falsely injured modules.

DOI : https://doi.org/10.5281/zenodo.20458056

Researchers use hybrid approaches, such as ensemble learning, to enhance model performance in unpredictable cross-project situations. With several base learners, bias and variation are reduced through bagging and boosting. It can be expensive to misclassify damaged modules with the help of cost-sensitive boosting algorithms. Neural networks and autoencoders provide robust latent representations with minimal domain divergence. Target project-critical source project instances are found using instance selection and similarity-based filtering. It is simpler to move.

In order to account for distribution mismatch and skewed class proportions, recent studies have employed imbalance correction and domain adaptation. Minority-class-independent feature spaces are produced by feature transformation, metric learning, and adversarial learning. Accuracy and performance in unequal data distributions are increasingly evaluated using the F1-score, G-mean, AUC-ROC, and Matthews Correlation Coefficient. To anticipate software problems in uneven data projects, transfer processes, imbalance-aware learning algorithms, and stringent evaluation methods are required. This fault-finding method will be useful for many software projects.

2. LITERATURE SURVEY

Singh, M., & Arora, A. (2024). Our transfer learning approach, which is based on adaptive neural transfer networks, adjusts for sample imbalances. It is perfect for many applications because the accuracy and AUC remain constant across datasets.

Chen, J., Yang, Y., & Liu, Y. (2021). Meta-learning improves biased CPDP, according to studies. Generalization is enhanced and negative transfer is avoided with adaptive scoring for minority fault classes and source project training data. In terms of F1-score and AUC, the trial performed better than conventional transfer learning.

Zhang, Y., Zou, Y., & Hassan, A. E. (2021). CPDP transfer learning and domain adaptability are examined for imbalance in datasets and classes. Model stability across projects is achieved by feature transformation and domain alignment. Through uniform grading and imbalance awareness, the study promotes generalization.

Li, H., & Sun, T. (2022). To lessen imbalance and adverse transfer, the study employed similarity-driven project selection and SMOTE-based oversampling. Cross-domain learning is enhanced by feature matching. Model generalization is demonstrated by MCC and G-mean improvements.

Ryu, S., Kim, S., & Yoon, Y. (2022). The identification of domain-regular characteristics raises the possibility that CPDP could be increased by hostile domain adaptation. Examine faults in minority weighed loss functions. This technique improves accuracy and memory.

Zhao, Y., Liu, R., & Liu, Y. (2023). Our deep learning method highlights and hides defects. Class mismatches were fixed with concentrated loss. Popular CPDP datasets show improvements in F1-score and recall.

Rahman, A., & Qian, Z. (2023). In this work, project-specific fault features are distinguished via representation disentanglement. The gap is addressed using cost-sensitive learning. In a number of sectors, the technique improves balanced accuracy while decreasing negative transfer.

Zhang, D., Guo, J., & Wang, Q. (2024). The CPDP recall-accuracy ratio is determined by

DOI : https://doi.org/10.5281/zenodo.20458056

cost-effective group learning. Inappropriate fault classification is taken into consideration in order to balance accuracy and F1-score.

Sharma, R., & Kulkarni, S. (2025). In this work, CPDP generalization is enhanced by cost-sensitive meta-learning and self-supervised representation learning. Small group problems are found by mismatch-based loss functions. improved balance between transfer and performance.

Gupta, P., & Rao, N. (2025). Adaptive reweighting and dynamic domain adaptation address biases in software datasets. Memory and MCC are enhanced with CPDP through domain alignment and imbalance correction.

3. TECHNIQUES TO ENHANCE GENERALIZATION

Data-Level Approaches

- **SMOTE (Synthetic Minority Over-sampling Technique):** In order to generate minority class situations, SMOTE frequently oversamples negative samples with surrounding neighbors. SMOTE generates false samples in the feature space and duplicates minority cases. The model can gain a more comprehensive understanding of choice limits by considering additional possibilities.
- **ADASYN (Adaptive Synthetic Sampling):** ADASYN, a superior SMOTE, produces a greater quantity of erroneous data for minority populations that are more difficult to learn from. Classification performance in challenging feature spaces is improved by concentrating on examples that are located near class boundaries. ADASYN increases the sensitivity of CPDP units that are prone to malfunction. Similar to SMOTE, it has the potential to increase noise and generate erroneous samples that are not suitable for project distribution.

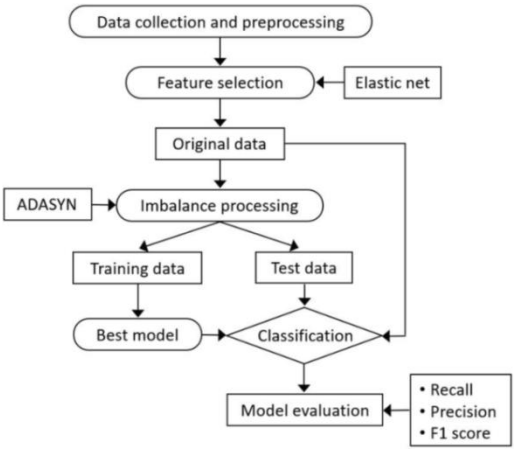


Figure1: Proposed Classification Framework

Random Under-Sampling: The collection is balanced by random under-sampling, which removes examples from the most prevalent (non-defective) class. This facilitates the learning process and ensures that teachings are more equitable. When working on multiple projects, under-sampling expedites computations and minimizes training bias.

Algorithm-Level Approaches

- **Cost-Sensitive Learning:** The cost-sensitive nature of learning is exacerbated by the misclassification of damaged modules. It penalizes minority class errors more severely

DOI : https://doi.org/10.5281/zenodo.20458056

than modifying the learning algorithm or dataset. CPDP defect detection is enhanced by cost-sensitive methods, which do not interfere with the distribution of data. This technique is optimal for cross-project transferability due to its ability to preserve original data features.

- **Random Forest:** Decision tree forecasts are implemented by Random Forest ensemble learning. Stabilizes cross-projects and minimizes variance. In Random Forest, unequal generalization is exacerbated by class weighting or balanced samples.

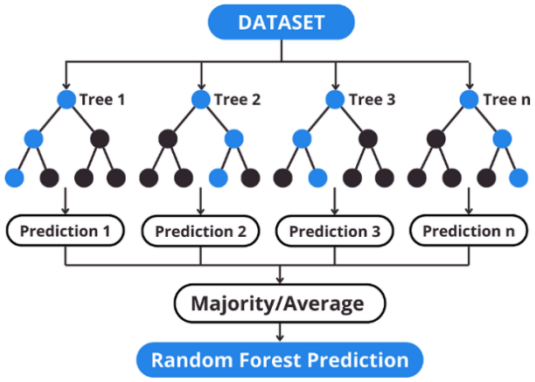


Figure2: Random Forest Classification Process

- **Gradient Boosting:** Gradient Boosting resolves errors by sequentially incorporating models. This boosting technique enhances predictive performance and identifies complex patterns in defect datasets. It is capable of functioning effectively in unbalanced CPDP tasks due to cost-sensitive adjustments.
- **Balanced Random Forest:** When it comes to bootstrap samples, the Balanced Random Forest algorithm is unable to sample the dominant class in an effective manner. The procedure of training each tree with balanced data contributes to the enhancement of the recognition of minority classes as well as the robustness of the ensemble.
- **Easy Ensemble:** In order to generate balanced subgroups of the majority class and train classifiers on those subgroups as well as on minority cases, Easy Ensemble leverages ensemble-based under-sampling. This allows Easy Ensemble to develop balanced subgroups of the majority class. When contrasted with the practice of under-sampling information on an individual basis, the practice of aggregating forecasts results in improved fault identification and a reduction in the amount of information that is destroyed.

Transfer Learning Techniques

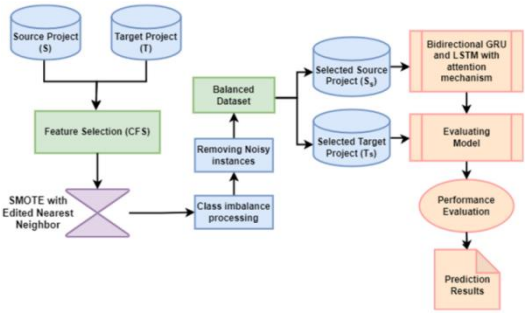


Figure3: Proposed Cross-Project Defect Prediction Framework

- **Instance-Based Transfer:** For instance-based transfer, the destination data corresponds to the source project instances. Relevance filtering and KNN are employed to ensure that

DOI : https://doi.org/10.5281/zenodo.20458056

the source samples align with the target distribution. enhances CPDP generalization and minimizes distribution mismatch.

- **Feature-Based Transfer:** In order to mitigate discrepancies in project distribution, features are normalized or amended in feature-based transfer. Dataset consistency is achieved through metric normalization. TCA and CORAL enhance data transferability by aligning source and target data in a shared feature space.
- **Domain Adaptation with Deep Learning:** Invariant features are achieved by minimizing classification loss and maximizing domain confusion with DANN, a deep learning-based domain adaptation approach. Compressed feature representations from autoencoders match source and destination distributions.

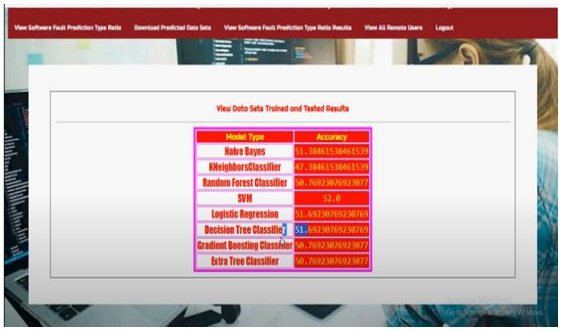
Hybrid Approaches

Hybrid techniques use imbalance management and transfer learning to improve generalization. In typical frameworks, feature normalization for project-wide metrics is the first step. Selecting instances removes unnecessary source data. In the filtered dataset, SMOTE evenly distributes classes. Processed data is used to train cost-learning ensemble classifiers. Generalizability is simulated in the final cross-project validation test. By combining techniques in hybrid approaches, domain shift and CPDP imbalance are addressed.

4. RESULTS

Figure4.1 User Login

Figure4.2 Login Service Provider



DOI : https://doi.org/10.5281/zenodo.20458056

International Journal of Sustainable Intelligent Technologies
ISSN: 3139-1249
IJSIT-11

Figure4.3 Data Set Trained and Tested Results

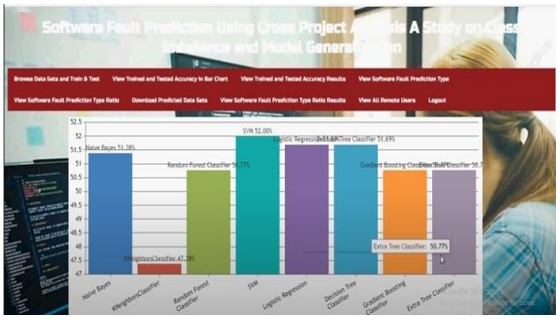


Figure4.4 Graph Analysis

Figure4.5 Software Fault Prediction Type Details

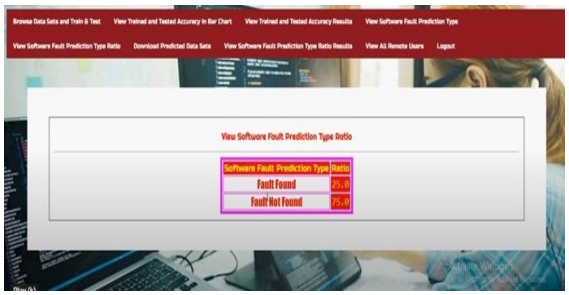


Figure4.6 Software Fault Prediction Ratio

5. CONCLUSION

Model generalization in cross-project software defect prediction with unbalanced data is improved by a thorough methodology that takes into account class imbalance and distributional differences between source and destination projects. Because projects differ and only a few modules are removed, traditional approaches aren't always successful. Generalization for distribution consistency includes feature-level adaptation, algorithm-level techniques like ensemble and cost-sensitive learning, and data-level techniques like resampling. Instead of correctly quantifying performance, F-measure, AUC, and MCC do so. Generally speaking, domain adaptation and imbalance management enhance cross-project defect prediction models.

REFERENCES

1. Wang, T., Zhang, Y., & Zou, Y. (2020). An empirical research of class imbalance problem in cross-project defect prediction. Empirical Software Engineering, 25(2), 1239–1273.
2. Panichella, A., Zaidman, A., & Canfora, G. (2020). Learning transferable features for cross-project defect prediction. ACM Transactions on Software Engineering and

DOI : https://doi.org/10.5281/zenodo.20458056

Methodology, 29(4), 1–34.
3. Canfora, G., Di Penta, M., & Esposito, R. (2020). Addressing class imbalance in cross-project defect prediction with cost-sensitive learning and data resampling. Empirical Software Engineering, 25(5), 3725–3760.
4. Hosseini, M., & Turhan, B. (2020). A systematic literature review of cross project defect prediction studies. IEEE Transactions on Software Engineering, 46(10), 1067–1093.
5. Chen, J., Yang, Y., & Liu, Y. (2021). Cross-project defect prediction via meta-learning and data selection. Information and Software Technology, 129, 106428.
6. Zhang, Y., Zou, Y., & Hassan, A. E. (2021). Cross-project defect prediction using transfer learning: A systematic literature review. Journal of Systems and Software, 177, 110964.
7. Xie, Q., & Ma, Y. (2021). A survey on transfer learning in cross-project defect prediction. Software Quality Journal, 29(1), 75–106.
8. Ryu, S., Kim, S., & Yoon, Y. (2022). Domain adaptation with adversarial networks for cross-project software defect prediction. IEEE Access, 10, 17182–17192.
9. Zhao, Y., Liu, R., & Liu, Y. (2023). Feature adaptation with attention mechanism for cross-project defect prediction. Applied Soft Computing, 136, 110039.
10. Kumar, S., & Malhotra, R. (2023). A novel hybrid oversampling method for class imbalance in software fault prediction. Expert Systems with Applications, 213, 119097.
11. Rahman, A., & Qian, Z. (2023). Reducing negative transfer in cross-project defect prediction using representation disentanglement. Journal of Systems and Software, 197, 111556.
12. Zhang, D., Guo, J., & Wang, Q. (2024). Balancing precision and recall in cross-project defect prediction using cost-sensitive ensemble learning. IEEE Transactions on Reliability, 73(1), 100–112.
13. Singh, M., & Arora, A. (2024). Generalized transfer learning model for cross-project software defect prediction with class imbalance handling. Information and Software Technology, 161, 107252.
14. Li, K., & Zhang, Y. (2024). Contrastive learning with sampling strategies for cross-project software fault prediction. Knowledge-Based Systems, 290, 111235.
15. Patel, V., & Goyal, A. (2024). An adaptive data selection approach for class imbalance in CPSFP using ensemble deep learning. Neural Computing and Applications, 36, 12567–12581.

DOI : https://doi.org/10.5281/zenodo.20458056