INTEGRATED APPROACH FOR IMPROVING CROSS-PROJECT SOFTWARE DEFECT PREDICTION PERFORMANCE

By

YAHAYA ZAKARIYAU BALA

Thesis Submitted to the School of Graduate Studies, Universiti Putra Malaysia, in Fulfilment of the Requirements for the Degree of Doctor of Philosophy

April 2024

FSKTM 2024 15

Abstract of thesis presented to the Senate of Universiti Putra Malaysia in fulfilment of the requirement for the degree of degree of Doctor of Philosophy

INTEGRATED APPROACH FOR IMPROVING CROSS-PROJECT SOFTWARE DEFECT PREDICTION PERFORMANCE

By

YAHAYA ZAKARIYAU BALA

April 2024

Chairman : Pathiah binti Abdul Samat, PhD
Faculty : Computer Science and Information Technology

This research addresses three critical challenges in cross-project defect prediction (CPDP): distribution differences, redundant features, and model overfitting. These issues often degrade prediction accuracy and robustness in various domains. To tackle these challenges, this study proposes a holistic approach named Transformation, Feature Selection, and Multi-learning (TFSM). This research is divided into three objectives: firstly, to proposed transformation, feature selection and multi-learning techniques that can mitigate distribution differences between datasets, identify and eliminate redundant features and combat model overfitting, respectively. Secondly, to integrate these techniques into a TFSM and implement. Thirdly, to evaluate each technique and the integrated approach. The research methodology involves the formulation, implementation, and evaluation of each technique individually and their integrated approach, TFSM. Experimental evaluations are conducted using open-source software projects sourced from the open source repository, with F1_score serving as the primary evaluation metric.

Results from the experiments demonstrate significant improvements in predictive performance. The transformation techniques effectively reduce distribution differences, enhancing the model's ability to generalize across diverse datasets. Feature selection methods successfully mitigate the negative impact of redundant features, streamlining the learning process and improving model interpretability. Additionally, the multi-learning approach proves effective in reducing model overfitting by aggregating diverse model outputs. When integrated into the TFSM approach, these techniques collectively demonstrated a marked improvement in CPDP performance. The TFSM approach leverages the strengths of each individual technique, resulting in a synergistic effect that enhances the model's predictive accuracy. This approach addresses the multifaceted challenges inherent in CPDP, providing a more reliable and effective solution for defect prediction in software projects. This work contributes to the ongoing efforts in the software engineering community to develop more accurate and reliable defect prediction models, ultimately aiding in the development of higher-quality software. Future work will focus on further refining these techniques and exploring their applicability to a broader range of software projects and repositories.

Keywords: Cross-Project, Defect, Machine Learning, Prediction, Software.
SDG: GOAL 9: Industry, Innovation and Infrastructure

Abstrak tesis yang dikemukakan kepada Senat Universiti Putra Malaysia sebagai memenuhi keperluan untuk ijazah Doktor Falsafah

PENDEKATAN BERSEPADU UNTUK MENINGKATKAN PRESTASI RAMALAN KECACATAN PERISIAN MERENTAS PROJEK

Oleh

YAHAYA ZAKARIYAU BALA

April 2024

Pengerusi : Pathiah binti Abdul Samat, PhD
Fakulti : Sains Komputer dan Teknologi Maklumat

Penyelidikan ini menangani tiga cabaran kritikal dalam ramalan kecacatan merentas projek (CPDP): perbezaan pengagihan, ciri-ciri berlebihan dan model lebih muat. Isu ini sering menurunkan ketepatan ramalan dan keteguhan dalam pelbagai domain. Untuk menangani cabaran ini, kajian ini mencadangkan pendekatan holistik yang dinamakan Transformasi, Pemilihan Ciri dan Pembelajaran Berbilang (TFSM). Penyelidikan ini terbahagi kepada tiga objektif: pertama, untuk cadangan transformasi, pemilihan ciri dan teknik pembelajaran berbilang yang boleh mengurangkan perbezaan pengagihan antara set data, mengenal pasti dan menghapuskan ciri-ciri berlebihan dan melawan model lebih muat, masing-masing. Kedua, untuk menyatupadukan teknik-teknik ini ke dalam Transformasi, Pemilihan Ciri dan Pembelajaran Berbilang (TFSM) dan dilaksanakan. Ketiga, untuk menilai setiap teknik dan pendekatan bersepadu. Metodologi penyelidikan melibatkan perumusan, pelaksanaan, dan penilaian setiap teknik secara individu dan pendekatan bersepadu mereka, TFSM. Penilaian eksperimen dijalankan menggunakan projek perisian sumber terbuka yang diperoleh daripada repositori sumber terbuka, dengan F1_score berfungsi sebagai metrik penilaian utama. Hasil eksperimen menunjukkan peningkatan yang ketara dalam prestasi ramalan. Teknik transformasi secara berkesan mengurangkan perbezaan pengagihan, meningkatkan keupayaan model untuk membuat generalisasi merentas set data yang pelbagai. Kaedah pemilihan ciri berjaya mengurangkan kesan negatif ciri-ciri berlebihan, memperkemas proses pembelajaran dan meningkatkan kebolehtafsiran model. Selain itu, pendekatan multi-pembelajaran terbukti berkesan dalam mengurangkan model lebih muat dengan mengagregatkan output model yang pelbagai. Apabila disepadukan ke dalam pendekatan TFSM, teknik ini secara kolektif menunjukkan peningkatan yang ketara dalam prestasi CPDP. Pendekatan TFSM memanfaatkan kekuatan setiap teknik individu, menghasilkan kesan sinergistik yang meningkatkan ketepatan ramalan model. Pendekatan ini menangani pelbagai cabaran yang wujud dalam CPDP, menyediakan penyelesaian yang lebih dipercayai dan berkesan untuk ramalan kecacatan dalam projek perisian. Kerja ini menyumbang kepada usaha berterusan dalam komuniti kejuruteraan perisian untuk membangunkan model ramalan kecacatan yang lebih tepat dan boleh dipercayai, akhirnya membantu dalam pembangunan perisian berkualiti tinggi. Kerja akan datang, akan menumpukan pada memperhalusi lagi teknik ini dan meneroka kebolehgunaannya pada rangkaian projek perisian dan repositori yang lebih luas.

Kata kunci: Kecacatan, Perisian, Projek Silang, Pembelajaran Mesin, Ramalan.

SDG: MATLAMAT 9: Industri, Inovasi dan Infrastruktur

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my supervisor, Dr. Pathiah Abdul Samat, for her invaluable guidance, encouragement, and continuous support throughout this journey. Her expertise and insights have been instrumental in shaping this work.

I am also immensely grateful to my entire supervisory committee, Dr. Khaironi Yatim Sharif and Dr. Noridayu Manshor for their constructive feedback and unwavering support. Their collective wisdom has significantly contributed to the quality and depth of this research.

To my family, especially my parents, my wife, brothers, and sisters, your unwavering love and encouragement have been my source of strength and motivation. Thank you for believing in me and supporting me in every possible way. I would like to extend my heartfelt appreciation to my sponsor, the Tertiary Education Trust Fund (TETFund), and Federal University of Kashere for providing the financial support necessary to pursue this research. Your investment in my education and professional development is deeply appreciated.

Finally, I would like to thank my institution, the Universiti Putra Malaysia, for providing the resources and environment conducive to academic and personal growth. Your commitment to excellence in education has been a cornerstone of my academic journey. Thank you all for your invaluable contributions to this work.

This thesis was submitted to the Senate of Universiti Putra Malaysia and has been accepted as fulfilment of the requirement for the degree of Doctor of Philosophy. The members of the Supervisory Committee were as follows:

Pathiah binti Abdul Samat, PhD
Senior Lecturer
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Chairman)

Khaironi Yatim bin Sharif, PhD
Senior Lecturer, Ts.
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

Noridayu binti Manshor, PhD
Senior Lecturer
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

ZALILAH MOHD SHARIFF, PhD
Professor and Dean
School of Graduate Studies
Universiti Putra Malaysia

Date: 7 November 2024

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AUC	Area Under the Curve
CKSDL	Cost-sensitive Semi-supervised Dictionary Learning
CPDP	Cross Project Defect Prediction
CTDP	Collective Transfer Learning Defect Prediction
DBSCAN	Density-Based Spatial Clustering of Applications
DCPDP	Direct Cross-Project Defect Prediction
DPC	Density Peaks Clustering
EM	Expectation-Maximization
EMD	Earth Mover Distance
FCR	Feature Class Relevance
Fr	Feature relevance
FRV	Feature Relevance Vector
FRV	Feature Relevance Vector
FS	Feature Selection
FS	Feature Selection
FV	Feature Vector
GA	Genetic Algorithm
GIS	Genetic Instance Selection
HYDRA	Hybrid Model Reconstruction Approach
KNN	K-Nearest Neighbor
LDF	Local Density of Features
LR	Logistic Regression
MFTCPDP	Manifold Feature Transformation Cross-Project Defect Prediction
NASA	National Aeronautics and Space Administration

NB	Naïve Bayes
NN	Neural Network
RF	Random Forest
S	Source
SDP	Software Defect Prediction
SF	Selected Features
SFD	Similarities of Features Density
SMET	Source Mean Transformation
SMOT	Source Mode Transformation
Sr	Reduced Source
St	Transformed Source
SVM	Support Vector Machine
T	Target
TCA	Transfer Component Analysis plus
TDS	Training Data Selection
TFSM	Transformed Feature Selection and Multi-learning
Tr	Reduced Target
UMR	Unified Metric Representation
WPDP	Within Project Defect Prediction

CHAPTER 1

INTRODUCTION

1.1 Background

Due to the expansion of software size and complexity, modern software programs are increasingly vulnerable to defects during the release or update of their versions. To ensure their corrective and developing maintenance, developers should frequently update these software applications by adding new functions or fixing bugs (Zhao et al., 2021). However, these updates can result in the introduction of new defects. Defects arising from particular software have a significant impact on business credibility and may result in fatal consequences such as loss of time and additional costs, and in the case of critical software, even health issues. According to the CISQ Consortium for Information & Software Quality 2022 report, the cost of poor software quality in the US has grown to at least $2.41 trillion (Herb Krasner, 2022). They, therefore, suggested that enhancing the testing activities can save more than a third of such an amount.

All these reported statistics showed the importance of software maintenance and software testing. Inspection of the entire code source is always difficult due to either limited resources or tight release schedules. Software testing is quite expensive (Rahmann and Ansari, 2021). Predicting defects in software development is a crucial task that allows organizations to proactively allocate resources and reduces the cost of software maintenance (Bejjanki et al., 2020).

A software defect prediction (SDP) model is used to track the most defective part of a software product before testing activities. Traditionally, software defect prediction models are built using historical data from a single project. These models employ various machine learning techniques to identify patterns and relationships within the code and project-specific data to predict where defects are likely to occur. However, these models tend to perform optimally within the context of the project from which the data was collected, known as within-project defect prediction (WPDP). Applying these models to a new project with different characteristics and codebases often results in reduced prediction accuracy (Li et al 2022).

Using historical records in the WPDP methodology demands an organization-wide effort to gather them for a long time. Because it requires the defect data up front, the first category of WPDP (as described above) is not helpful in practice. The second category typically yields only positive results in software projects that are equivalent in terms of software development, team experience, programming language, and application domain (Shao et al., 2021); Chen and Dai, 2021). In general, the WPDP methodology cannot be used for newly released software products because of a lack of historical records (Zhao et al., 2021; Tang et al., 2021).

Considering the challenges of obtaining historical records in WPDP methodology, a cross-project defect prediction (CPDP) technique was introduced (Amasaki, 2020; Luo et al, 2021). In the CPDP technique, the defect datasets of one software project (source) accessible in the repository are utilized to predict defects in another software project in progress (target) (Chen et al., 2020). CPDP is of great significance in the software development industry. It can help organizations make informed decisions about resource allocation, code review prioritization, and quality assurance strategies for projects that lack sufficient historical defect data. This is especially valuable for new or small-scale projects where defect prediction based on historical data is limited.

However, the prediction performance of CPDP is poor (Zhao et al, 2021). This is because projects differ in terms of programming languages, development methodologies, and team dynamics. Therefore, the data from one project may not be directly applicable to another. In addition, identifying the most relevant features for defect prediction becomes more complex when applying models across projects. In addition, in CPDP, the identification and handling of feature redundancy is crucial for improving model efficiency and effectiveness. Feature redundancy occurs when two or more features in a predictive model convey similar or redundant information. This redundancy can lead to suboptimal model performance, increased computational costs, and potential overfitting.

Furthermore, model overfitting is a significant concern in cross-project defect prediction, as it can lead to reduced generalization performance when applying predictive models to new and unseen projects. Overfitting occurs when a model learns the training data too well, capturing noise or random fluctuations that do not represent true patterns. In the context of cross-project defect prediction, overfitting can hinder the model's ability to adapt to the unique characteristics of different projects.

These challenges ignite a series of questions on how to handle the challenges faced by CPDP and improve its prediction performance. Recommended solutions included a data transformation technique in which any available source projects can be transformed to minimize the data distribution difference between the source and target projects, as well as selecting relevant features and improving the classifier used for building the prediction model. However, the existing CPDP models do not provide optimum performance.

1.2 Problem Statement

Cross-project defect prediction has gained a lot of attention and has been considered particularly important in the field of software engineering. However, poor prediction performance is a challenge for CPDP. In general, the prediction performance of all the existing techniques is still low (Abdu et al., 2023). Improving the prediction performance of cross-project defect prediction, particularly aiming for a minimum F1_score of 0.75, is essential for several reasons, including that achieving a high F1_score ensures that the model strikes a balance between precision and recall. This is crucial in real-world scenarios, where reliable defect prediction is necessary for effective decision-making in software development. The F1_score accounts for both false positives and false negatives. By targeting a threshold of 0.75, the model aims to minimize these errors, enhancing its ability to accurately identify defects and reduce the risk of overlooking or misclassifying issues, and this model is more likely to generalize well across diverse software development environments.

One of the major factors affecting the prediction performance of CPDP is the data distribution difference (Zhao et al., 2022). Example of a particular real-world scenario where the difficulties associated with distribution differences in CPDP have become visible in software development companies that work on numerous projects concurrently. Within this company, any project could have distinctive characteristics of its own, including technologies, development teams, programming languages, and customer needs. It is challenging to transfer defect prediction models that have been learned on one project to another because of this variability. Assuming this software development company used past data from one of its projects to build a defect prediction model. They now wish to use the same model for a new project.

On the other hand, the new project might use a different coding standard, a different programming language, and team members with different backgrounds. The prediction performance of the defect prediction model may be greatly impacted by these variations since the model might not be able to accurately represent the unique characteristics and peculiarities of the new project.

To address these challenges, researchers have proposed a number of techniques, including training data selection (Yuan et al., 2020) and data transformation (Zhao et al., 2021). However, existing techniques ignored the most important statistical distribution characteristic (mean and mode). The mean and mode, which stand for the average values and most frequent, respectively, are indicators of central tendency in a distribution (Condon et al., 2023). If these measures are ignored, significant information regarding the usual behavior of defects in different projects may be lost. Mean and mode are less affected by extreme values compared to other measures like median or percentile, making them suitable choices for datasets with varying distribution characteristics across projects. Mean and mode transformations help mitigate the impact of distribution differences on the predictive model's performance. In addition, mean and mode transformations retain essential information about the data distribution, ensuring that valuable insights are not lost during the transformation process.

Another challenge that affects the prediction performance of CPDP is feature redundancy (Saeed, 2023). The majority of software projects exhibit redundant features, and the set of features used to train the model has a significant impact on the prediction performance of CPDP (Li et al., 2021). Required feature sets for CPDP may vary throughout projects. It might not be the best idea to use the same set of features for all projects because some features might be more useful in one project but less so in another. It is crucial to automatically choose and extract the best features for every project. To address these challenges, various feature selection techniques have been proposed. However, existing feature selection techniques for CPDP do not account for outliers. Excluding outlier consideration may result in potential model poor prediction performance across diverse projects.

Moreover, another challenge with CPDP is overfitting that arose from utilizing individual classifiers to construct CPDP models (Javed et al., 2024). When a single classifier is used, overfitting may occur as a result of the model becoming too complicated and specialized for the source project, which would lead to poor results on the target project. As such, it is critical to address the overfitting problem. None of the existing techniques explored stacking ensemble to mitigate overfitting in CPDP.

1.3 Research Questions

To effectively evaluate the experimental results of the proposed techniques based on the heighted problems, four research questions (RQ1 to RQ4) were established as follows.

RQ1: Does the proposed transformation techniques reduce the impact of distribution difference and redundant features on the prediction performance of CPDP?

RQ2: Does feature selection reduce the impact of redundant features on the prediction performance of CPDP?

RQ3: Does the proposed Multi-learning technique reduce the negative impact of model overfitting on the prediction performance of CPDP?

RQ4: Does the integration of transformation, feature selection and multi-learning techniques improved the prediction performance of CPDP?

1.4 Research Objective

The aim of this work is to propose an integrated approach to solving the three problems (distribution difference, high dimensional features, and overfitting) and improving the prediction performance of cross-project defect prediction (CPDP). To achieve this objective, the following three sub-objectives were outlined:

i. To propose Transformation, Feature selection and multi-learning for reducing distribution difference, redundant features and model overfitting.

ii. To integrate the proposed transformation, feature selection and multi-learning and implement.

iii. To evaluate the proposed transformation, Feature selection and multi-learning and integrated approach in order to assess their effectiveness in improving performance of CPDP.

1.5 Significant of the Study

This research is significant in the context of improving cross-project defect prediction models by addressing critical limitations in existing techniques. The study successfully proposed and implement transformation technique based on mean and mode distribution. This is significant for normalizing and adapting the data to mitigate the impact of variations in data distributions across projects, promoting a more standardized representation and generalization of models in cross-project defect prediction

The study also successfully proposed and implement feature selection. This is significant for identifying and prioritizing the most relevant features that influence the occurrence of defects. It helps in reducing the dimensionality of the data, focusing the model on the critical factors, thereby enhancing the efficiency of the defect prediction process.

In addition, the study successfully proposed and implement multi-Learning technique. This is significant for leveraging knowledge gained from multiple models to enhance the performance of defect prediction of CPDP. Furthermore, the study successfully implements the integration of transformation, feature selection and multi-learning techniques can lead to improved prediction performance, ultimately aiding in more effective software testing and maintenance efforts.

1.6 Scope of the Study

This study aims to enhance the performance of cross-project defect prediction by integrating transformation, feature selection, and multi-learning. The focus is on addressing the challenges posed by distribution differences, redundant features, and model overfitting in predicting software defects across diverse projects. The aim of transformation is to reduce distribution differences using mean and mode transformations. Feature selection is to identify and select the most relevant features that significantly contribute to defect prediction accuracy. Multi-learning is to enhance model accuracy using the stacking ensemble method. Defect datasets from various open-source software projects were collected. Model performance was measured using the F1-Score to balance precision and recall. Statistical analysis was conducted using the Wilcoxon signed-rank test and Cliff delta to evaluate the significance of performance differences and the effect size and quantify the practical significance of performance improvements.

1.7 Thesis Organization

Chapter 2 disused literature review. Chapter 3 outlines the research methodology used in the study. Chapter 4 discussed proposed transformation, feature selection, multi-learning, and integration of all three techniques. Chapter 5: Result. This chapter presents a thorough evaluation of the proposed integrated technique, providing detailed results and analysis. Finally, chapter 6 presented a conclusion and future work.

REFERENCES

Abdu, A., Zhai, Z., Abdo, H. A., Algabri, R., & Lee, S. (2023). Graph-Based Feature Learning for Cross-Project Software Defect Prediction. Computers, Materials and Continua, 77(1).

Akour, M., Alsmadi, I., & Alazzam, I. (2017). Software Fault Proneness Prediction: A Comparative Study between Bagging, Boosting, and Stacking Ensemble and Base Learner Methods. International journal of Data analysis techniques and strategies, 9(1):1-16.

Alazba, A., & Aljamaan, H. (2022). Software Defect Prediction Using Stacking Generalization of Optimized Tree-Based Ensembles. Applied Sciences, 12(9): 45-77.

Ali, S. S., Zafar, M. S., & Saeed, M. T. (2020, January). Effort Estimation Problems in Software Maintenance–a survey. In 2020 3rd international conference on computing, mathematics and engineering technologies (iCoMET) (pp. 1-9). IEEE.

Ali, U., Aftab, S., Iqbal, A., Nawaz, Z., Bashir, M. S., & Saeed, M. A. (2020). Software Defect Prediction Using Variant based Ensemble Learning and Feature Selection Techniques. International Journal of Modern Education & Computer Science, 12(5).

Alsawalqah, H., Hijazi, N., Eshtay, M., Faris, H., Radaideh, A. A., Aljarah, I. & Alshamaileh, Y. (2020). Software defect prediction using Heterogeneous Ensemble Classification Based on Segmented Patterns. Applied Sciences, 10(5): 17-45.

Alshammari, F. H. (2022). Software Defect Prediction and Analysis Using Enhanced Random Forest (extrf) Technique: A Business Process Management and Improvement Concept in IOT-based Application Processing Environment. Mobile Information Systems, 2022:11. doi.org/10.1155/2022/2522202.

Amasaki, S. (2020). Cross-version defect prediction: use historical data, Cross-project data, or both? Empirical Software Engineering, 25: 1573-1595.

Balogun, A. O., Basri, S., Mahamad, S., Abdulkadir, S. J., Capretz, L. F., Imam, A. A., & Kumar, G. (2021). Empirical analysis of rank aggregation-based multi-filter feature selection methods in software defect prediction. Electronics, 10(2): 179.

Bejani, M. M., & Ghatee, M. (2021). A systematic review on overfitting control in shallow and deep neural networks. Artificial Intelligence Review, 54(8): 6391-6438.

Bejjanki, K. K., Gyani, J., & Gugulothu, N. (2020). Class Imbalance Reduction (CIR): a novel approach to software defect prediction in the presence of class imbalance. Symmetry, 12(3): 407.

Bian, Y., Wang, Y., Yao, Y., & Chen, H. (2019). Ensemble Pruning Based on Objection Maximization with a General Distributed Framework. IEEE Transactions on Neural Networks and Learning Systems, 31(9): 3766-3774.

Chen, D., Chen, X., Li, H., Xie, J., & Mu, Y. (2019). Deepcpdp: Deep learning based cross-project defect prediction. IEEE Access, 7: 184832-184848.

Chen, S., Ye, J. M., & Liu, T. (2020). Domain adaptation approach for cross project software defect prediction. J. Softw. 31: 266–281.

Chen, Y., & Dai, H. (2021, September). Improving Cross-Project Defect Prediction with Weighted Software Modules via Transfer Learning. In Journal of Physics: Conference Series, IOP Publishing, 2025 (1): 012100.

Chidamber, S. R., & Kemerer, C. F. (1994). A metrics suite for object-oriented design. IEEE Transactions on Software Engineering, 20(6): 476–493. https://doi.org/10.1109/32.295895.Computers, Materials and Continua, 77(1).

Condon, D., Drougas, A., & Abrokwah, M. (2023). The Mean May Not Mean What You Think It Means: The Use and Misuse of Measures of Central Tendency. The Journal of Applied Business and Economics, 25(4), 74-88.

Cruz, A. E. C, Ochimizu, K. (2009). Towards logistic regression models for predicting fault-prone code across software projects. In: 2009 3rd international symposium on empirical software engineering and measurement, ESEM 2009. 460–463.

D'Ambros M, Lanza M, Robbes R (2010) An extensive comparison of bug prediction approaches Proceedings of the 7th IEEE working conference on mining software repositories, MSR'10, 31– 41.

Gong, L., Rajbahadur, G. K., Hassan, A. E., & Jiang, S. (2021). Revisiting the impact of dependency network metrics on software defect prediction. IEEE Transactions on Software Engineering, 48(12): 5030-5049.

Goyal, S. (2022). Effective software defect prediction using support vector machines (SVMs). International Journal of System Assurance Engineering and Management, 13(2): 681-696.

He, Z., Shu, F., Yang, Y., Li, M., & Wang, Q. (2012). An investigation on the feasibility of cross-project defect prediction. Automated Software Engineering, 19: 167-199.

Herb Krasner (2022) The Cost of Poor-Quality Software in the US: A 2022 Report", Consortium for Information & Software Quality, https://www.it-cisq.org/the-cost-of-poor-quality-software-in-the-us-a-2022-report/. Accessed 8th May 2024.

Hu, Z., & Zhu, Y. (2023). Cross-project defect prediction method based on genetic algorithm feature selection. Engineering Reports, 5(12), e12670.

Huang, Y., Li, X., & Wang, J. (2023). Enhancing Defect Prediction Models by Removing Outliers Using Statistical Methods. Empirical Software Engineering Journal, 28(3): 567-590.

Javed, K., Shengbing, R., Asim, M., & Wani, M. A. (2024). Cross-Project Defect Prediction Based on Domain Adaptation and LSTM Optimization. Algorithms, 17(5): 175.

Jin, C. (2021). Cross-project software defect prediction based on domain adaptation learning and optimization. Expert Systems with Applications, 171, 114637.

Jureczko, M., & Madeyski, L. (2010, September). Towards identifying software project clusters with regard to defect prediction. In Proceedings of the 6th international conference on predictive models in software engineering. 1-10.

Kang, H., & Do, S. (2024). ML-Based Software Defect Prediction in Embedded Software for Telecommunication Systems (Focusing on the Case of SAMSUNG ELECTRONICS). Electronics, 13(9): 1690.

Karim, O. E. & Mahmoud, O. E. (2008). Predicting defect-prone software modules using support vector machines. Journal of Systems and Software, 81(5): 649– 660.

Khatri, Y., & Singh, S. K. (2022). Cross project defect prediction: a comprehensive survey with its SWOT analysis. Innovations in Systems and Software Engineering, 1-19.

Laradji, I. H. Alshayeb, M., & Ghouti, L. (2015). Software defect prediction using ensemble learning on selected features. Information and Software Technology, 58: 388-402.

Larracy, R., Phinyomark, A., & Scheme, E. (2021, November). Machine learning model validation for early stage studies with small sample sizes. In 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC) (pp. 2314-2319). IEEE.

Li, F., & Zhao, X. (2024). Hybrid Mean and Mode-Based Scaling for Reducing Distribution Differences in CPDP. Journal of Software: Evolution and Process, 36(1): 54-70.

Li, Y., & Chen, W. (2020). A comparative performance assessment of ensemble learning for credit scoring. Mathematics, 8(10): 1756.

Li, Z., Niu, J., Jing, X. Y., Yu, W., & Qi, C. (2021). Cross-project defect prediction via landmark selection-based kernelized discriminant subspace alignment. IEEE Transactions on Reliability, 70(3): 996-1013.

Li, Z., Zhang, H., Jing, X. Y., Xie, J., Guo, M., & Ren, J. (2022). Dssdpp: data selection and sampling based domain programming predictor for cross-project defect prediction. IEEE Transactions on Software Engineering.

Liang, M., Chang, T., A, B., Duan, X., Du, L., Wang, X., & Gao, H. (2021). A stacking ensemble learning framework for genomic prediction. Frontiers in genetics, 12: 1-9.

Luo, H., Dai, H., Peng, W., Hu, W., & Li, F. (2021). An Empirical Study of Training Data Selection Methods for Ranking-Oriented Cross-Project Defect Prediction. Sensors, 21(22): 7535.

Mabayoje, M. A., Balogun, A. O., Jibril, H. J. Atoyebi, J. O. Mojeed, H. A., & Adeyemo, V. E. (2019). "Parameter tuning in KNN for software defect prediction: an empirical analysis". Jurnal Teknologi dan Sistem Komputer, 7(4): 121-126.

Maddipati, S. S., & Srinivas, M. (2021) A Hybrid Approach for Cost Effective Prediction of Software Defects International Journal of Advanced Computer Science and Application 12(2): 145 – 152.

Matloob, F., Ghazal, T. M., Taleb, N., Aftab, S., Ahmad, M., Khan, M. A., & Soomro, T. R. (2021). Software defect prediction using ensemble learning: A systematic literature review. IEEE Access, (9): 98754-98771.

Mehta, S., & Patnaik, K. S. (2021). Improved prediction of software defects using ensemble machine learning techniques. Neural Computing and Applications, 33: 10551-10562.

Mienye, I. D., & Sun, Y. (2022). A survey of ensemble learning: Concepts, algorithms, applications, and prospects. IEEE Access, 10: 99129-99149.

Mustaqeem, M., Mustajab, S. and Alam, M. (2024), A hybrid approach for optimizing software defect prediction using a gray wolf optimization and multilayer perceptron. International Journal of Intelligent Computing and Cybernetics, 17(2): 436-464, doi10.1108/IJICC-11-2023-0385.

Nguyen, T., & Tran, H. (2024). Improving Cross-Project Defect Prediction through Feature Selection and Outlier Removal. IEEE Transactions on Software Engineering, 50(1): 112-130.

Ni, C., Liu, W. S., Chen, X., Gu, Q., Chen, D. X., & Huang, Q. G. (2017). A cluster-based feature selection method for cross-project software defect prediction. Journal of Computer Science and Technology, 32(6): 1090-1107.

Pal, S., & Sillitti, A. (2022). Cross-project defect prediction: a literature review. IEEE Access, 10: 118697-118717.

Pandey, S. K., & Tripathi, A. K. (2021). An empirical study toward dealing with noise and class imbalance issues in software defect prediction. Soft Computing, 25(21): 13465-13492.

Panichella, A., Oliveto, R., & De Lucia, A. (2014, February). Cross-project defect prediction models: L'union fait la force. In 2014 Software Evolution Week-IEEE Conference on Software Maintenance, Reengineering, and Reverse Engineering (CSMR-WCRE), IEEE. 164-173.

Qiu, S, Lu, L, Jiang, S. (2018). Multiple- weights model for cross-project software defect prediction. IET Softw 12: 345–355.

Qiu, S., Xu, H., Deng, J., Jiang, S., & Lu, L. (2019). Transfer convolutional neural network for cross-project defect prediction. Applied Sciences, 9(13): 2660.

Rahmann, W., & Ansari, G. A. (2021). Ensemble techniques-based software fault prediction in an open-source project. In Research Anthology on Usage and Development of Open-Source Software. IGI Global. 693-709.

Ravichandran, T., Gavahi, K., Ponnambalam, K., Burtea, V., & Mousavi, S. J. (2021). Ensemble-based machine learning approach for improved leak detection in water mains. Journal of Hydroinformatics, 23(2): 307-323.

Rudd, J. M. (2020). An empirical study of downstream analysis effects of model pre-processing choices. Open journal of statistics, 10(5): 735-809.

Sabzevari, M., Martínez-Muñoz, G., & Suárez, A. (2022). Building heterogeneous ensembles by pooling homogeneous ensembles. International Journal of Machine Learning and Cybernetics, 1-8.

Saeed, M. S. (2023). Role of Feature Selection in Cross Project Software Defect Prediction-A Review. International Journal of Computations, Information and Manufacturing (IJCIM), 3(2): 37-56.

Saeed, M. S., & Saleem, M. (2023). Cross Project Software Defect Prediction Using Machine Learning: A Review. International Journal of Computational and Innovative Sciences, 2(3): 35-52.

Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4): 1249.

Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. SN computer science, 2(3): 160.

Shao, Y., Zhao, J., Wang, X., Wu, W., & Fang, J. (2021). Research on Cross-Company Defect Prediction Method to Improve Software Security. Security and Communication Networks.

Sharma, U., & Sadam, R. (2023). How far does the predictive decision impact the software project? The cost, service time, and failure analysis from a cross-project defect prediction model. Journal of Systems and Software, 195: 111522.

Sheng, L., Lu, L., & Lin, J. (2020). An adversarial discriminative convolutional neural network for cross-project defect prediction. IEEE Access, 8: 55241-55253.

Singh, P., & Nevendra, M. (2020). Prediction priority of defective modules for testing resource allocation. Automated Software Testing: Foundations, Applications and Challenges 15: 95-109.

Soleymani, R., Granger, E., & Fumera, G. (2020). F-measure curves: A tool to visualize classifier performance under imbalance. Pattern Recognition, 100: 107146.

Taneja, S., Gupta, C., Goyal, K., & Gureja, D. (2014, February). An enhanced k-nearest neighbor algorithm using information gain and clustering. In 2014 Fourth International Conference on Advanced Computing & Communication Technologies, IEEE, 325-329.

Tang, S., Huang, S., Zheng, C., Liu, E., Zong, C., & Ding, Y. (2021). A novel cross-project software defect prediction algorithm based on transfer learning. Tsinghua Science and Technology, 27(1): 41-57.

Thota, M. K., Shajin, F. H., & Rajesh, P. (2020). Survey on software defect prediction techniques. International Journal of Applied Science and Engineering, 17(4): 331-344.

Tong H., Liu B., & Wang S. H. (2019). Transfer-learning oriented class imbalance learning for cross-project defect prediction, https://arxiv.org/abs/1901.08429.

Tong, H., Zhang, D., Liu, J., Xing, W., Lu, L., Lu, W., & Wu, Y. (2024). MASTER: Multi-Source Transfer Weighted Ensemble Learning for Multiple Sources Cross-Project Defect Prediction. IEEE Transactions on Software Engineering.

Vashisht, R. & Rizvi, S. A. M. (2020). Feature Extraction to Heterogeneous Cross Project Defect Prediction. In 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), IEEE, 1221-1225.

Vujovic, Z. (2021). Classification model evaluation metrics. International Journal of Advanced Computer Science and Applications, 12(6): 599-606.

Wang, Q., & Liu, S. (2024). "Improving Cross-Project Defect Prediction with Mean and Mode-Based Normalization in Stacking Ensembles." Empirical Software Engineering, 29(1): 78-96.

Watanabe, S., Kaiya, H., & Kaijiri, K. (2008). Adapting a fault prediction model to allow inter language reuse. Proceedings of the 4th international workshop on Predictor models in software engineering. New York, NY, USA: ACM, 19-24.

Wen, W., Zhang, B., Gu, X., & Ju, X. (2019). An empirical study on combining source selection and transfer learning for cross-project defect prediction. In: IBF 2019–2019 IEEE 1st international workshop on intelligent bug fixing. Institute of Electrical and Electronics Engineers Inc., 29–38.

Wu, J., Chen, L., & Zhang, Y. (2023). Statistical Normalization Methods for Cross-Project Defect Prediction. Software Quality Journal, 31(2): 123-145.

Wu, R., Zhang, H., Kim, S., & Cheung, S. (2011). Relink: recovering links between bugs and changes. Paper presented at the Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering.

Xia, X., Lo, D., Pan, S. J., Nagappan, N., & Wang, X. (2016). Hydra: Massively compositional model for cross-project defect prediction. IEEE Trans. Software. Eng., 42: 977–998.

Xing, Y., Lin, W., Lin, X., Yang, B., & Tan, Z. (2022). Cross-project defect prediction based on two-phase feature importance amplification. Computational Intelligence and Neuroscience, 2022(1): 2320447.

Xu, Z., Yuan, P., Zhang, T., Tang, Y., Li, S., & Xia, Z. (2018). HDA: Cross-project defect prediction via heterogeneous domain adaptation with dictionary learning. IEEE access, 6: 57597-57613.

Yacouby, R., & Axman, D. (2020, November). Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In Proceedings of the first workshop on evaluation and comparison of NLP systems 79-91.

Yang, Y., Xia, X., Lo, D., Bi, T., Grundy, J., & Yang, X. (2022). Predictive models in software engineering: Challenges and opportunities. ACM Transactions on Software Engineering and Methodology (TOSEM), 31(3): 1-72.

Yuan, Z., Chen, X., Cui, Z., & Mu, Y. (2020). ALTRA: cross-project software defect prediction via active learning and tradaboost. IEEE Access, 8: 30037-30049.

Zhang, Q., & Ren, J. (2021). Software-defect prediction within and across projects based on improved self-organizing data mining. The Journal of Supercomputing, 78(5):6147-6173.

Zhang, Y., Lin, H., & Chen, X. (2023). Stacking Ensemble Learning for Cross-Project Defect Prediction: Mitigating Overfitting with Multi-Model Integration. Journal of Software: Evolution and Process, 35(2): 143-161.

Zhang, Y., Lo, D., Xia, X., & Sun, J. (2018). Combined classifier for cross-project defect prediction: an extended empirical study. Frontiers of Computer Science, 12(2): 280-296.

Zhao, Y., Zhu, Y., Yu, Q., & Chen, X. (2021). Cross-Project Defect Prediction Method Based on Manifold Feature Transformation. Future Internet, 13(8): 216.

Zhao, Y., Zhu, Y., Yu, Q., & Chen, X. (2022). Cross-Project Defect Prediction Considering Multiple Data Distribution Simultaneously. Symmetry 14(2): 401.

Zimmermann, T., Nagappan, N. Gall, H., Giger, E., & Murphy, B. (2009). Cross-project defect prediction: A large-scale experiment on data vs. domain vs. process. In Proceedings of the 7th Joint Meeting of the European Software Engineering Conference and the ACM Signoff Symposium on the Foundations of Software Engineering, New York, NY, USA, 91–100.

Zou, Q., Lu, L., Yang, Z., & Xu, H. (2021, October). Multi-source cross project defect prediction with joint Wasserstein distance and ensemble learning. In 2021 IEEE 32nd International Symposium on Software Reliability Engineering (ISSRE) IEEE, 57-68.