Cross Project Software Defect Prediction Using Ensemble Learning, a Comprehensive Review

Muhammad Salman Saeed
1Department of Computer Science, Virtual University of Pakistan, Lahore, Pakistan
*Corresponding Author: Salman Saeed. Email: salman.saeed403@gmail.com

Abstract: Software defect prediction (SDP) is a key task in software engineering that aims to identify potential defects in software projects to improve quality and reduce maintenance costs. Traditional defect prediction models are often limited by the availability of labeled data within a single project. Cross-Project Defect Prediction (CPDP) addresses this limitation by leveraging data from multiple projects. This review paper provides a comprehensive survey of the state-of-the-art in CPDP, with a special focus on the application of ensemble learning methods. Ensemble learning, which combines multiple models to improve prediction accuracy, shows great promise in enhancing CPDP performance. This paper systematically reviews various ensemble learning methods, including bagging, boosting, and stacking, and their applications in CPDP. Key challenges such as data heterogeneity, model transferability, and evaluation metrics are discussed. Furthermore, this review also highlights recent advances, comparative analysis, and future research directions in CPDP using ensemble learning. This article aims to provide a thorough understanding of the current situation and guide future research in developing more robust and accurate CPDP models.

Keywords: Software defect prediction, cross project defect prediction, ensemble learning

1 Introduction

Software defect prediction (SDP) plays a vital role in improving software quality and reliability. Ensuring high software quality is critical to reducing maintenance costs, increasing user satisfaction, and maintaining software system reliability [1], [2]. Traditionally, SDP leverages historical defect data within a project, known as within-project defect prediction (WPDP), to build machine learning models that predict future defects. WPDP has proven effective in many situations, but it suffers when dealing with limited or missing defect data, especially in new or ongoing projects where historical defect data may be unavailable or insufficient. Cross-Project Defect Prediction (CPDP) emerges as a compelling alternative to WPDP. CPDP utilizes defect data from source projects to predict defects in target projects. By leveraging data from multiple projects, the CPDP addresses the data scarcity issue inherent to the WPDP. This approach is particularly beneficial for new or early-stage projects where there is little or no defect history. Additionally, CPDP benefits from rich data diversity across projects, capturing a broader range of software defect characteristics and providing more general insights into defect prediction [3].

However, the CPDP also faces its own set of challenges [4]. Selecting relevant source data is critical to building an effective predictive model. The heterogeneity of data characteristics between different projects complicates the integration and analysis of data. Selecting an appropriate machine learning model to handle such heterogeneous data and transfer knowledge across different domains and contexts is another significant challenge. Furthermore, effective evaluation of CPDP models requires robust evaluation metrics that can account for differences between source and target projects. The convergence of software engineering and machine learning offers promising solutions to these challenges. By leveraging historical project data, machine learning techniques have the potential to transform software engineering into a proactive field. These techniques analyze patterns and relationships in defect data, enabling the identification of areas of code that are more likely to hide defects in future projects. Researchers have proposed various machine learning methods to enhance the software development process, ranging from traditional algorithms to advanced techniques such as deep learning and ensemble learning. This shift to predictive defect management brings significant benefits, including improved software quality and optimized resource allocation through targeted testing and code reviews of defect-prone areas.

The concept of ensemble learning is rooted in the idea that a collection of models when appropriately aggregated, can compensate for the weaknesses of individual models and exploit the strengths of individual models. This aggregation usually improves generalization performance and reduces overfitting. There are number of ensemble learning techniques that are already proposed in different studies [5-8]. Techniques such as bagging, boosting, and stacking are commonly used to build ensembles, each with unique mechanisms for enhancing model diversity and performance. For example, bagging reduces variance by averaging the predictions of independently trained models, while boosting focuses on correcting the errors of previous models to reduce bias. Stacking, on the other hand, combines multiple models via a meta-learner to optimize the overall prediction by learning how best to integrate the output of the base learner. Ensemble learning, which combines multiple machine learning models to improve prediction accuracy and robustness, shows great promise in the context of CPDP. By aggregating predictions from multiple models, ensemble learning can reduce the variance and bias associated with individual models, resulting in more reliable and accurate defect predictions [9-12]. Various ensemble learning techniques, such as bagging, boosting, and stacking, have been applied to CPDP, demonstrating improvements in prediction performance.

This paper provides a comprehensive review of cross-project software defect prediction (CPDP), with a special focus on ensemble learning techniques. We delve into the core concepts of CPDP, explore the challenges it poses, and take a comprehensive look at the application of ensemble learning in this field. This review covers state-of-the-art methods, highlighting the advantages and limitations of different approaches. Additionally, we discuss key challenges such as data heterogeneity, model transferability, and evaluation metrics. This paper also provides an in-depth introduction to recent advances, comparative analysis, and future research directions in CPDP using ensemble learning. By synthesizing the existing body of knowledge, this review aims to provide a thorough understanding of the current situation and guide future research in developing more robust and accurate CPDP models.

Methodology:

This section outlines a systematic approach to a comprehensive review of cross-project software defect prediction (CPSD) using ensemble learning. This review aims to synthesize existing research, identify trends, and highlight key findings and gaps in the literature.

A paper [13] addresses the challenge of interpreting complex neural networks, often considered black-box models. These networks perform exceptionally well on real-world tasks but lack transparency, hindering human understanding and trust. The authors propose a method to extract probabilistic automata to explain recurrent neural networks (RNN). Their approach improves existing methods by using probability to compensate for the loss of expressive power and adaptively identify the correct level of abstraction. The extracted model is useful for detecting adversarial texts. The paper exploits cross-project defect prediction, where a model is trained on data from one project and applied to predict defects in another project. This approach helps overcome data scarcity within individual projects and improves predictive performance. The authors use ensemble learning techniques to combine multiple models. Ensemble methods improve overall accuracy and robustness by aggregating predictions from different models. The authors evaluate their approach on real-world datasets using state-of-the-art architectures like Gated Recurrent Units (GRUs) and Long Short-Term Memory (LSTM) networks. Their method significantly outperforms existing approaches in terms of accuracy and scalability.

Another paper [14] addresses the challenge of cross-project defect prediction, where the availability of defect-related data from different projects poses an open issue. While much research has focused on within-project defect prediction, the authors explore the less explored area of multi-item classification for cross-project defect prediction. Their goal is to predict defects in different projects using an ensemble-based approach. The authors employ ensemble-based statistical models, specifically gradient boosting and random forests. These models combine predictions from multiple base models to improve overall accuracy and robustness. The main results of this study show that multiple-item/multi-category classification applies to cross-project data and produces results comparable to within-project defect data. By leveraging ensemble learning and multinomial classification, this paper helps improve defect prediction for different software projects.

According to the authors of an article [15] Cross-project defect prediction (CPDP) is a challenging field that aims to predict defects in software projects with limited data availability, thereby creating generalizable prediction models. However, heterogeneity across projects creates additional difficulties. While several approaches have been proposed to address these challenges, the class-imbalanced nature of cross-project defect data has not been thoroughly investigated. In this paper, the authors propose a novel cost-aware integration approach for CPDP. The authors balanced the class distribution by resampling the data. This paper mitigates the impact of class imbalance on model performance. The base classifier is trained on the resampled data. These classifiers serve as the building blocks of the ensemble. This method automatically adjusts the cost value during model training. This adaptability enhances the practical effectiveness of the method. The ensemble combines predictions from multiple base classifiers. The final combination of classifiers is used for defect prediction.

Another paper [16] addresses the challenge of Cross-Project Defect Prediction (CPDP), whose goal is to predict defects in software projects using data from external projects. CPDP is particularly useful when a project lacks historical tag defect data. The authors propose a novel approach that combines feature transfer and ensemble learning to enhance defect prediction across different projects. The authors utilize feature transfer, which involves transferring knowledge from labeled data in an external project to a target project. By extracting relevant features and patterns, models can learn from similar contexts, thereby improving prediction accuracy. Ensemble learning combines the predictions of multiple base models. The authors use an ensemble of classifiers to improve overall performance. This approach mitigates the impact of individual model biases and improves robustness.

Another article [17] addresses the challenge of Cross-Project Software Defect Prediction (CPDP), whose goal is to predict defects in software projects using data from external projects. Traditional defect prediction often faces a lack of data within a single project, so cross-project prediction is crucial. The authors propose a novel hybrid multi-model migration approach (HMMTA) specifically designed for CPDP. Their results highlight situations where HMMTA may not be feasible due to sensitivity to parameter selection. This analysis provides insights into cross-domain transfer learning and provides guidance for researchers and practitioners in the field of defect prediction. In summary, Zhang et al.'s work contributes to effective cross-project defect prediction by combining ensemble learning with transfer methods. HMMTA solves the distribution difference and class imbalance problems and improves the prediction performance of different software projects.

The paper [18] by Jindal, Ahmad, and Aditya introduces a hybrid model for predicting software defects across different projects. The authors propose a two-stage model that combines ensemble learning and genetic algorithms. In software testing, defect prediction can be done within a project (WPDP) or across different projects (CPDP). This article focuses on CPDP, where data from certain projects is used to train a machine learning model that is then used to predict defects in other projects. This method is considered faster for software defect prediction. The ensemble learning stage of this model is part of the two-stage model proposed by the authors. Ensemble learning is a machine learning concept where multiple models are trained to solve the same problem and combined to get better results.

By using ensemble learning, the authors were able to improve the performance of the CPDP model. In summary, this paper proposes a new method for cross-project defect prediction using a hybrid model that combines ensemble learning and genetic algorithms. The authors demonstrate the effectiveness of their model using datasets from the PROMISE repository, achieving promising results. This research contributes to the field of software defect prediction by providing a faster and more efficient method to predict software defects for different projects.

A paper [19] introduces an advanced heterogeneous defect prediction model that uses an ensemble learning technique. The model incorporates eleven classifiers and uses both supervised and unsupervised machine learning algorithms to predict the defect proneness of software modules. In terms of cross-project defect prediction (CPDP), the paper discusses three major categories of software defect prediction (SDP): Within-project defect prediction (WPDP), CPDP, and heterogeneous defect prediction (HDP). In WPDP, the model is trained with the labeled instances of a project and predicts new instances of the same project. However, as new types of projects are being developed rapidly, the labeled instances of the same projects are often not available for the training of a model. Therefore, developers seek expertise from their own or others' past experiences and try to predict the defect proneness of modules of new projects. This concept is generally referred to as transfer learning (TL). Based on TL, researchers have proposed CPDP, in which the model predicts defect proneness for new projects lacking in historical limitation data by taking advantage of expertise available from other projects. The only of CPDP is that it only works with the same metrics set between projects. Regarding ensemble learning, the paper uses an ensemble learning technique that incorporates precisely eleven classifiers. In the heterogeneous ensemble method, different base learners are generated using different machine learning techniques. These base learners are combined, and the final prediction is performed by integrating the results of the base learners either statistically or by voting. This approach helps in predicting the defect proneness of the software modules. In summary, the paper presents a novel approach to heterogeneous defect prediction using an ensemble learning technique. The authors demonstrate the effectiveness of their model using datasets from the PROMISE repository, achieving promising results. This research contributes to the field of software defect prediction by providing a more efficient method for predicting software defects across different projects.

Another article [20] presents a framework for cross-project defect prediction. The authors propose a two-stage framework to address the challenges of complex structure and imbalanced data in cross-project defect prediction. In the first stage of the framework, Principal Component Analysis (PCA) is applied for dimensionality reduction of the dataset into two components. PCA is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. This transformation is defined in such a way that the first principal component has the largest possible variance, and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. In the second phase, the Synthetic Minority Over-sampling Technique (SMOTE) of data sampling is applied to handle the class imbalance problem. SMOTE is an oversampling technique where the synthetic samples are generated for the minority class. This technique helps to overcome the overfitting problem which occurs when exact replication of the minority class instances is done. After these two stages, the ensemble classifiers random forest and XGBoost are applied for an effective defect-prediction model. Ensemble learning is a machine learning concept where multiple models are trained to solve the same problem and combined to get better results. By using ensemble learning, the authors were able to improve the performance of their cross-project defect prediction model. The authors conducted experiments on eight open-source software projects and compared the results with a few baseline techniques. The results indicate that the proposed framework gave comparable performance of cross-project defect prediction to some baseline methods. This research contributes to the field of software defect prediction by providing a more efficient method for predicting software defects across different projects.

A paper [21] introduces a novel method for multi-source cross-project defect prediction (MSCPDP). The authors propose a method called MASTER, which stands for Multi-Source Transfer Weighted Ensemble Learning. In terms of cross-project defect prediction, the paper discusses the concept of MSCPDP, which attempts to transfer defect knowledge learned from multiple source projects to the target project. This approach has drawn increasing attention due to its advantages over single-source cross-project defect prediction (SSCPDP). However, two main problems restrict the performance of existing MSCPDP models: how to effectively extract the transferable knowledge from each source dataset and how to measure the amount of knowledge transferred from each source dataset to the target dataset. To address these issues, the authors propose the MASTER method. MASTER measures the weight of each source dataset based on feature importance and distribution difference. It then extracts the transferable knowledge based on the proposed feature-weighted transfer learning algorithm. Regarding ensemble learning, the paper uses an ensemble learning technique as part of the MASTER method. Ensemble learning is a machine learning concept where multiple models are trained to solve the same problem and combined to get better results. By using ensemble learning, the authors were able to improve the performance of their MSCPDP model. In summary, the paper presents a novel approach to multi-source cross-project defect prediction using a called method MASTER that combines feature-weighted transfer learning and ensemble learning. The authors demonstrate the effectiveness of their model using datasets from multiple software projects, achieving promising results. This research contributes to the field of software defect prediction by providing a more efficient method for predicting software defects across different projects.

The paper [22] provides empirical insights into the effectiveness of three resampling ensemble methods (bagging, boosting, and dagging) with Deep learning neural network (DLNN) for cross-project software defect prediction compared to a single DLNN model. An empirical study is conducted using five data sets. In terms of cross-project defect prediction, this article discusses the process of leveraging historical data from other systems to predict defects in new software systems. This approach is more practical and industrially feasible than project forecasting, which lacks data in the early stages of software development. However, due to the differences between source and target software projects, the accuracy of cross-project prediction models is typically lower than the accuracy of within-project models. In summary, this paper provides an empirical comparison of DLNN resampling ensemble methods for cross-project software defect prediction. The authors demonstrate the effectiveness of their model using five datasets, achieving promising results. This research contributes to the field of software defect prediction by providing a more efficient method to predict software defects for different projects.

In terms of cross-project defect prediction, a paper [23] discusses the process of knowledge transfer from source software projects to target software projects. The authors propose a novel CPDP method named MJWDEL to learn transfer weights to evaluate the importance of each source item to the target task. They applied TCA techniques and logistic regression (LR) to train sub-models for each source and target items. They also designed the joint Wasserstein distance to understand the source-target relationship, and then calculated the transfer weights of different sub-models based on this. The transferred weights can then be used to reweight these sub-models to determine their importance in knowledge transfer to the target task. Regarding ensemble learning, this paper uses ensemble learning techniques as part of the MJWDEL approach. Ensemble learning is a machine learning concept where multiple models are trained to solve the same problem and combined to get better results. The paper proposes a new approach to multi-source cross-project defect prediction using a method called MJWDEL that combines joint Wasserstein distance and ensemble learning. The authors demonstrate the effectiveness of their model using datasets from multiple software projects, achieving promising results. This research contributes to the field of software defect prediction by providing a more efficient method to predict software defects for different projects.

The paper [24] explores the application of ensemble learning methods in cross-project defect prediction usage of. Background on Cross-Project Defect Prediction (CPDP). This paper addresses the challenge of distribution differences between different project data, a common problem in CPDP. The authors propose a manifold embedding distribution adaptation (MDA) method to reduce the distribution gap in the manifold feature subspace. The method consists of two processes: manifold feature learning and joint distribution adaptation. Manifold feature learning solves the problem of feature distortion caused by the nonlinear distribution of high-dimensional data. Joint distribution adaptation takes into account the importance of marginal and conditional distributions. This paper also discusses the class imbalance problem, which often leads to misclassification of defective instances. To solve this problem, the authors introduced subclass discriminant analysis (SDA), an effective feature learning method.

Another article [25] proposes a new method for software defect prediction using ensemble learning. The author introduces a new algorithm, the 2-step Sparrow Optimized Extreme Learning Machine (2SSSA), to enhance the predictive performance of the Extreme Learning Machine (ELM) algorithm in software defect prediction. The ELM algorithm is a single hidden layer feedforward neural network that faces the challenges of random parameter selection and limited generalization ability. To overcome these challenges, the authors propose the use of swarm intelligence optimization algorithms. However, these optimization methods may encounter challenges related to getting stuck in local optimal solutions. To enhance the ability of the original Sparrow algorithm to escape local extrema, the authors adopted pinhole imaging inverse learning and somersault foraging strategies. The performance of 2SSSA in terms of optimization and convergence speed is evaluated using various benchmark functions. Furthermore, the authors propose an ensemble algorithm for software defect prediction, denoted as 2SSEBA, which adopts the two-step optimization Sparrow algorithm (2SSSA) to optimize the extreme learning machine. Ensemble learning is known for its significant enhancement of predictive performance and model generalization capabilities.

A study [26] proposes a new method for early cross-project defect prediction (eCPDP) using transfer learning method. The authors highlight the limitations of existing TL-based CPDP techniques, which are not suitable for the unit testing phase as they require the entire historical target project data. This limitation results in missed opportunities to improve product reliability by applying predictive results at an early stage. To solve this problem, the authors propose a TL-based CPDP technique called eCPDP that can be applied during the unit testing phase. This technique utilizes singular value decomposition (SVD) and requires only the source project data of the TL. In summary, this paper provides a comprehensive study of early cross-project defect prediction methods. It introduces a new technology, eCPDP that can be applied during the unit testing phase, helping practitioners find and fix defects earlier than other TL-based CPDP techniques12. This paper provides new insights and methods for solving common problems in CPDP, making a significant contribution to the field of software defect prediction.

Another study [27] discusses the use of extreme learning machines (ELM) to predict software defects in different projects. The authors note that many researchers have developed defect prediction models using traditional machine learning techniques for defect prediction within projects. However, these models often perform poorly on different datasets. To address this issue, the authors studied the use of ELM for cross-project defect prediction. They also explored the use of ELM for defect prediction in nonlinear heterogeneous ensembles. This ensemble model is called nonlinear heterogeneous extreme learning machine ensemble (NH ELM) and aims to improve the above problems. In summary, this paper presents a comprehensive study of cross-project software defect prediction using ELM and ensemble learning. It introduces a new ensemble model, NH ELM that improves the accuracy and completeness of predictions. This paper makes a significant contribution to the field of software defect prediction by providing new insights and methods.

Another paper [28] discusses a new method for Predicting software defects in different projects.

The authors note that existing methods for such predictions are not always effective because the data can be difficult to separate and may be unbalanced. To solve these problems, the authors propose a new method called two-stage ensemble learning (TSEL). The method is divided into two parts: the Ensemble Multi-kernel Domain Adaptation (EMDA) stage and the Ensemble Data Sampling (EDS) stage. During the EMDA phase, they developed a predictor that combined the advantages of multi-core learning and domain adaptation techniques. During the EDS stage, they used a technique called RESample with replacement (RES) to learn multiple different predictors and combine them.

By following this systematic and rigorous approach, this review aims to provide a comprehensive and insightful overview of cross-project software defect prediction using ensemble learning, contributing to the advancement of knowledge and practice in this critical area of software engineering.

Conclusions:

In this comprehensive review, we have systematically explored the field of cross-project software defect prediction (CPDP) using ensemble learning, synthesizing existing research, identifying trends, and highlighting key findings and gaps in the literature. Ensemble learning techniques, including methods such as bagging, boosting, and stacking, have been shown to significantly enhance the predictive performance and robustness of CPDP models. By combining multiple base models, ensemble methods can mitigate the biases and limitations of individual models, leading to improved accuracy and generalization across different projects. Cross-project defect prediction addresses the challenge of data scarcity within individual projects by leveraging data from external projects. Techniques such as resampling, feature transfer, and cost-aware integration are crucial in handling class imbalance and heterogeneity across projects. These methods ensure that the models are trained on balanced datasets and can effectively transfer knowledge from source to target projects.

Various innovative approaches have been proposed to improve CPDP, including probabilistic automata for interpreting complex neural networks, hybrid models combining ensemble learning with genetic algorithms, and advanced feature transfer methods. These approaches contribute to the advancement of CPDP by addressing specific challenges such as the lack of transparency in neural networks, parameter sensitivity, and distribution differences between projects. The scalability and adaptability of CPDP models are enhanced through techniques like adaptive cost value adjustment, manifold embedding distribution adaptation, and feature-weighted transfer learning. These methods ensure that CPDP models can be effectively applied to diverse and large-scale datasets, maintaining high predictive performance across different software projects. Numerous studies have empirically validated the effectiveness of ensemble learning methods in CPDP using real-world datasets. Results indicate that ensemble-based approaches consistently outperform single-model methods in terms of accuracy, robustness, and scalability, making them a preferred choice for software defect prediction. Despite the advancements in CPDP using ensemble learning, there remain several open issues and research gaps. Future research should focus on developing more transparent and interpretable models, exploring the impact of different feature selection methods, and addressing the challenges of dynamic and evolving software environments. Additionally, more empirical studies are needed to validate the proposed methods across a wider range of projects and domains.

In conclusion, this review has provided a comprehensive overview of the current state of CPDP using ensemble learning. The findings underscore the potential of ensemble methods to significantly improve defect prediction across software projects, contributing to the development of more reliable and efficient software systems. This review serves as a valuable resource for researchers and practitioners in the field, guiding future efforts to advance the state of CPDP and address the remaining challenges.

References:

[1] Khleel, N. A. A., & Nehéz, K. (2023). A novel approach for software defect prediction using CNN and GRU based on SMOTE Tomek method. *Journal of Intelligent Information Systems*, 1-35.

[2] Esteves, G., Figueiredo, E., Veloso, A., Viggiato, M., & Ziviani, N. (2020). Understanding machine learning software defect predictions. *Automated Software Engineering*, *27*(3-4), 369-392.

[3] Agrawal, A., & Malhotra, R. (2022). Cross project defect prediction for open source software. *International Journal of Information Technology*, *14*(1), 587-601.

[4] Jahanshahi, H., Cevik, M., & Başar, A. (2021). Moving from cross-project defect prediction to heterogeneous defect prediction: a partial replication study. *arXiv preprint arXiv:2103.03490*.

[5] 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.

[6] Ganaie, M. A., Hu, M., Malik, A. K., Tanveer, M., & Suganthan, P. N. (2022). Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, *115*, 105151.

[7] Zhang, Y., Liu, J., & Shen, W. (2022). A review of ensemble learning algorithms used in remote sensing applications. *Applied Sciences*, *12*(17), 8654.

[8] Fang, Z., Wang, Y., Peng, L., & Hong, H. (2021). A comparative study of heterogeneous ensemble-learning techniques for landslide susceptibility mapping. *International Journal of Geographical Information Science*, *35*(2), 321-347.

[9] Matloob, F., Aftab, S., & Iqbal, A. (2019). A Framework for Software Defect Prediction Using Feature Selection and Ensemble Learning Techniques. *International Journal of Modern Education & Computer Science*, *11*(12).

[10] 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 and Computer Science*, *13*(5), 29.

[11] Abbas, S., Aftab, S., Khan, M. A., Ghazal, T. M., Hamadi, H. A., & Yeun, C. Y. (2023). Data and Ensemble Machine Learning Fusion Based Intelligent Software Defect Prediction System. *Computers, Materials & Continua*, *75*(3).

[12] Iqbal, A., & Aftab, S. (2020). Prediction of defect prone software modules using MLP based ensemble techniques. *International Journal of Information Technology and Computer Science*, *12*(3), 26-31.

[13] Zhang, T., Du, Q., Xu, J., Li, J., & Li, X. (2020, December). Software defect prediction and localization with attention-based models and ensemble learning. In *2020 27th Asia-Pacific Software Engineering Conference (APSEC)* (pp. 81-90). IEEE.

[14] Goel, L., Sharma, M., Khatri, S. K., & Damodaran, D. (2020). Prediction of cross project defects using ensemble based multinomial classifier. *EAI Endorsed Transactions on Scalable Information Systems*, *7*(25), e5-e5.

[15] Li, Y., Wen, M., Liu, Z., & Zhang, H. (2022). Using Cost-cognitive Bagging Ensemble to Improve Cross-project Defects Prediction. *Journal of Internet Technology*, *23*(4), 779-789.

[16] Zeng, F., Lin, W., Xing, Y., Sun, L., & Yang, B. (2022). A Cross-project Defect Prediction Model Using Feature Transfer and Ensemble Learning. *Tehnički vjesnik*, *29*(4), 1089-1099.

[17] Zhang, S., Jiang, S., & Yan, Y. (2023). A hybrid multiple models transfer approach for cross-project software defect prediction. *International Journal of Software Engineering and Knowledge Engineering*, *33*(03), 343-374.

[18] Jindal, R., Ahmad, A., & Aditya, A. (2022). Ensemble Based-Cross Project Defect Prediction. In *Ubiquitous Intelligent Systems: Proceedings of ICUIS 2021* (pp. 611-620). Springer Singapore.

[19] Ansari, A. A., Iqbal, A., & Sahoo, B. (2020). Heterogeneous defect prediction using ensemble learning technique. In *Artificial intelligence and evolutionary computations in engineering systems* (pp. 283-293). Springer Singapore.

[20] Goel, L., Sharma, M., Khatri, S. K., & Damodaran, D. (2020). Defect prediction of cross projects using PCA and ensemble learning approach. In *Micro-Electronics and Telecommunication Engineering: Proceedings of 3rd ICMETE 2019* (pp. 307-315). Springer Singapore.

[21] 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*.

[22] Elish, M. O., & Elish, K. (2021). An Empirical Comparison of Resampling Ensemble Methods of Deep Learning Neural Networks for Cross-Project Software Defect Prediction. *International Journal of Intelligent Engineering & Systems*, *14*(3).

[23] 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)* (pp. 57-68). IEEE.

[24] Qiu, S., Lu, L., Jiang, S., & Guo, Y. (2019). An investigation of imbalanced ensemble learning methods for cross-project defect prediction. *International Journal of Pattern Recognition and Artificial Intelligence*, *33*(12), 1959037.

[25] Dong, X., Liang, Y., Miyamoto, S., & Yamaguchi, S. (2023). Ensemble learning based software defect prediction. *Journal of Engineering Research*, *11*(4), 377-391.

[26] Kwon, S., Ryu, D., & Baik, J. (2021, December). eCPDP: Early cross-project defect prediction. In *2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS)* (pp. 470-481). IEEE.

[27] Bal, P. R., & Kumar, S. (2018). Cross Project Software Defect Prediction using Extreme Learning Machine: An Ensemble based Study. In *ICSOFT* (pp. 354-361).

[28] Li, Z., Jing, X. Y., Zhu, X., Zhang, H., Xu, B., & Ying, S. (2019). Heterogeneous defect prediction with two-stage ensemble learning. *Automated Software Engineering*, *26*, 599-651.