International Journal of Computations, Information and Manufacturing (IJCIM) 3(2) -2023

Contents available at the publisher website: G A F T I M . C O M

International Journal of Computations, Information and Manufacturing (IJCIM)

Journal homepage: https://journals.gaftim.com/index.php/ijcim/index

ROLE OF FEATURE SELECTION IN CROSS PROJECT SOFTWARE DEFECT PREDICTION- A REVIEW

Muhammad Salman Saeed

Department of Computer Science, Virtual University of Pakistan, Lahore, Pakistan.

A R T I C L E I N F O

Keywords:
Software defect prediction, cross project defect prediction, feature selection, machine learning

Received: Sep, 11, 2023
Accepted: Oct, 09, 2023
Published: Dec, 22, 2023

A B S T R A C T

Software Defect Prediction (SDP) is crucial for enhancing software quality and minimizing issues after release. The advent of machine learning, particularly in Cross-Project Defect Prediction (CPDP), has garnered significant attention for its potential to enhance defect predictions in one project by leveraging information from another. A critical factor influencing CPDP effectiveness is feature selection, the process of identifying the most relevant features from an available set. This review article thoroughly examines the role of feature selection in CPDP. Existing feature selection methods are systematically analyzed and classified within the CPDP context, encompassing both traditional and state-of-the-art approaches. The review delves into the challenges and opportunities presented by diverse project characteristics, data heterogeneity, and the curse of dimensionality. Additionally, the article underscores how feature selection impacts model performance, generalization, and adaptability across various software projects. Through synthesizing findings from multiple studies, trends, best practices, and potential research directions in the field are identified. In conclusion, this review article provides valuable insights into the significance of feature selection for enhancing the reliability and efficiency of CPDP models.

1. INTRODUCTION

In today's digital era, software development is integral to nearly every aspect of our lives, from smartphone apps to critical infrastructure systems. The quality and reliability of software are paramount, as defects or errors can result in system failures, security vulnerabilities, and financial losses. Identifying and preventing such issues is a top priority for software professionals. There are many studies that elaborate the concept of software defect prediction (Iqbal, A., Aftab, S, et al., 2.019). Software defect prediction (SDP) takes a proactive approach to mitigate the impact of defects by using data-driven techniques to identify potential issues early in the development process (Omondiagbe, et al., 2022). This enables organizations to allocate resources efficiently, focus testing efforts, and enhance overall software quality. Cross-project software defect prediction is an emerging area that extends the application of SDP beyond a single project. It leverages data and insights from completed projects to predict defects in new software projects. However, scaling across projects presents challenges due to differences in domain, scale, and development processes among projects. Selecting relevant features or variables that accurately differentiate between defective and non-defective software components is crucial for successful defect prediction. There are also few studies about the feature selection methods and techniques (Jindal, R., Ahmad, A., & Aditya, 2022). Feature selection, a data preprocessing technique, plays a key role in shaping the effectiveness and

efficiency of prediction models. It reduces dimensionality, eliminates noise, and improves model generalization (Zhu, Y., Zhao, Y., Yu, Q., & Chen, X, 2022). The feature selection process becomes complex in cross-project forecasting due to data heterogeneity. We have also reviewed some machine learning concepts from the studies (Abbas, S., 2023) . There are many studies that provide the review of the literature about the software defect prediction and different methods and techniques of software defect prediction and feature selection (Matloob, F., 2019). This review paper provides an in-depth exploration of feature selection techniques in predicting software defects across projects. We examine challenges in this domain, including data heterogeneity, class imbalance, and knowledge transferability between projects. Additionally, we conduct a comprehensive survey of existing feature selection methods and their adaptability to cross-project scenarios. Our aim is to offer researchers and practitioners a deeper understanding of the complexities in this field, facilitating the development of more robust and adaptable defect prediction models. We draw insights from systematic literature reviews (Daoud, M. S., Fatima, A., Khan, 2021) and follow similar methodologies in this study.

2. RELATED WORK

This paper investigates various studies that explore the role and significance of different feature selection techniques in CPDP. In (Ali, U., Aftab, S., 2020), researchers introduced a novel method for predicting bugs in software projects that differs from traditional model training. The method adjusts classifier selection based on the similarity between source and target projects, demonstrating improved accuracy and reduced workload for wrong predictions across projects. The proposed method incorporates a correlation-based feature selection (CFS) technique. CFS assesses the correlation of a feature subset by considering individual predictive power and redundancy among features. Using a heuristic search algorithm, CFS identifies the optimal subset of features that maximizes correlation with the class variable while minimizing mutual correlation between features. In (Aftab, S., 2021) investigates factors affecting the performance of cross-project ageing-related bug prediction (CPARBP), which identifies ageing-related bugs in software systems using data from other projects. The paper employs a feature selection technique called TLAP (Transfer Learning with Adaptive Projection). TLAP combines domain adaptation and feature selection to reduce domain differences and feature redundancy between source and target projects for cross-project ageing-related error prediction. TLAP includes three steps: project clustering, adaptive projection, and feature ranking. It clusters source projects, uses an adaptive projection matrix to project features onto a common subspace, and clusters features based on correlation and redundancy scores to select informative and discriminative features for CPARBP. In (Iqbal, A, 2019) authors address the software defect prediction problem, proposing a novel class imbalance learning method. This method combines cluster-based oversampling, filtering, and transfer learning techniques to handle challenges such as class imbalance, noise, and heterogeneity in flawed data. The feature selection technique utilized is a hybrid approach that combines cluster-based oversampling, filtering, and transfer learning, addressing common challenges in software engineering.

The proposed technique is compared with existing methods, reporting better performance on various metrics. The study by (Rahman, A. U., Abbas, S, 2021) explores the feasibility and performance of a hybrid search-based cross-project defect prediction algorithm that utilizes data from similar projects to identify faulty modules. Four search-based hybrid algorithms are proposed, combining search-based algorithms with machine learning techniques. The study evaluates these algorithms on 12 datasets from the PROMISE repository using various metrics. Results are compared with baseline machine learning techniques and other existing CPDP methods. Iqbal, A., & Aftab, S. (2019), focused on the challenge of predicting software defects across projects, known as CPDP. The paper contends that current CPDP evaluation methods lack consideration for the practical costs and benefits of applying defect prediction models. To address this, the paper suggests using cost metrics such as expected cost reduction (ECR) and return on s (ROI) to gauge CPDP model performance. The feature selection technique employed is termed cost-sensitive feature selection, aiming to choose features minimizing the expected cost of a defect

prediction model rather than focusing solely on the number or accuracy of defects. The paper introduces a general framework for cost-sensitive feature selection applicable to any existing feature selection algorithm. Additionally, four different cost-sensitive feature selection algorithms—cost-sensitive filter, cost-sensitive wrapper, cost-sensitive embedding, and cost-sensitive hybrid algorithm—are compared.

In (Catolino, G., Di Nucci, D., & Ferrucci, F, et al.,2019,), the challenge of CPDP is addressed, aiming to predict defects in target projects using data from other source projects. The proposed method seeks to enhance CPDP by simplifying training data, considering both similarity and the number of defects for each training instance. The paper introduces a new metric, Defect Similarity (DS), to measure instance similarity based on defect labels and ratios. A training data selection algorithm called TDSelector is proposed, which selects high-quality instances for CPDP from multiple source projects based on DS values and defect labels. The paper reports that TDSelector outperforms baseline methods and enhances AUC values.

A study (Ahmed, U, et al., 2022) presents a novel feature selection method, cross-project feature selection (CPFS), utilizing a two-stage process of feature ranking and selection for CPDP. The evaluation of CPFS on datasets from open-source projects using metrics like AUC, F-measure, G-mean, and Balance shows significant improvements in CPDP performance. The paper also identifies challenges and future directions in feature selection for CPDP, such as handling heterogeneous data, integrating domain knowledge, and combining multiple feature selection techniques. A novel recommender system is introduced to propose optimal training data selection methods for CPDP. CPDP utilizes data from other projects when the historical data of the target project is insufficient. The paper compares four existing training data selection methods and introduces two new methods, TCA+ with Burak filter (TCA+B) and cluster-based TCA+ (TCA+C). The evaluation, conducted on 10 datasets from the PROMISE repository using metrics like F-measure and G-mean, claims that the proposed recommender system achieves better results than existing methods in most cases. In a research paper (Aziz, N., & Aftab, S. 2021), a novel framework for CPDP is proposed, utilizing Abstract Syntax Tree (AST) and Multi-Kernel Transfer Convolutional Neural Network (MKT-CNN). The main challenge addressed is the imbalance and distribution differences between data from source and target projects in CPDP. The paper employs AST to represent source code and MKT-CNN to extract semantic features and reduce divergence. The proposed node granularity and MK-TCNN framework aim to extract transferable features for CPDP by minimizing data distribution differences. Results demonstrate the method's superiority in terms of F-measure, G-mean, and AUC. The paper also analyzes the influence of MKT-CNN model components and parameters, providing insights into the feature learning and transfer process. A novel CPDP method utilizing a dissimilarity space (DS) to represent features of software modules from multiple source and target projects. A weighted voting scheme is employed to combine different source-to-target correlations from various sources. Feature selection is performed using the feature clustering and ranking (FECAR) method. The evaluation, conducted on 10 open-source Java projects, compares the proposed method with existing CPDP methods, reporting improved accuracy, precision, recall, and F-measure. An empirical analysis investigates the impact of factors such as the number and size of source projects, similarity measures, and weighting schemes on the proposed approach. In a paper (Ali, U., Aftab, S., 2020), a novel feature selection method for CPDP is presented, based on the concept of information flow.

The method involves three steps: removing features with very low label information flow, reducing the number of features by selecting those with the highest information flow, and finding the best combination of features using the F-measure. This study aims to enhance the performance of defect prediction models through effective feature selection. Another study (Aziz, N., & Aftab, S. 2021). proposes an innovative approach to software defect prediction, focusing on attribute selection to improve accuracy. The method involves generating and ranking pairwise attribute combinations, selecting candidate attributes based on frequency, and employing a forward search algorithm to select the final attribute subset. The technique is evaluated using datasets from the NASA MDP and other sources, demonstrating its effectiveness in

improving prediction accuracy. This study contributes to the field by providing a systematic approach to selecting relevant properties for defect prediction models. An empirical study of the CPDP method is presented. The study demonstrates motivation through a case study involving two projects and two CPDP approaches, using various datasets and performance metrics. The study aims to contribute to the field of software defect prediction by exploring prediction diversity across projects and methodologies. The transfer learning to address the CPDP problem. The proposed CFIW-TNB model combines instance and feature weighting in a Naive Bayesian framework, aiming to enhance CPDP performance by incorporating instance and feature correlations. Experimental evaluation of real-world datasets demonstrates the superiority of CFIW-TNB over existing CPDP methods.

The challenge of detecting security vulnerabilities in software with limited training data is addressed. The proposed solution utilizes serialized Abstract Syntax Trees (AST) and bidirectional LSTM networks to create transferable representations, demonstrating effectiveness in identifying vulnerable functions in different projects.

A paper (Sharma, U., & Sadam, R. 2023) introduces a new approach to CPDP, leveraging data from another project to find software bugs in one project. The method, called HDA (heterogeneous domain adaptation), can handle situations where two projects have different capabilities. HDA outperforms existing methods on multiple pairs of projects in different datasets. In a paper (Jahanshahi, H., Cevik, M., & Başar, A, 2021), the focus is on the problem of heterogeneous cross-project defect prediction (HCPDP), aiming to predict defects in a target project using data from different source projects. The challenge lies in potential differences in data distributions, feature spaces, and defect rates between source and target projects. The paper introduces optimal transport (OT) theory, a mathematical framework for measuring distance and mapping between different probability distributions. Based on OT theory, the paper proposes two prediction algorithms: OT-based HCPDP (OT-HCPDP) and OT-based HCPDP with feature selection (OT-HCPDP-FS). Evaluation on 20 open-source projects compares these methods with existing HCPDP approaches, claiming better performance in terms of accuracy, recall, precision, F-measure, and AUC-ROC. A study (Ali, U., Aftab, S., 2020) introduces Genetic Instance Selection (GIS) as a new approach for CPDP. GIS combines genetic algorithms, nearest neighbor filtering, and feature selection to select the most relevant data and features for CPDP. The paper evaluates GIS using different versions of multi-version projects and compares it to other CPDP and within-project defect prediction (WPDP) methods. Results indicate that GIS significantly outperforms the baseline CPDP method and is comparable to the WPDP method in terms of F1 and G metrics commonly used for evaluating defect prediction models. The goal is to develop a model trained on current source projects to predict defects in target projects in software engineering. The article emphasizes the significant impact of feature selection and hyper-parameter tuning on defect prediction accuracy. Controlled experimental methods and quantitative approaches are used, applying various techniques to preprocess the dataset. A hybrid feature selection method, combining filter and wrapper methods, is adopted. The paper employs a neural network as a classifier and grid search for hyper-parameter tuning. A study (Sharma, U., & Sadam, R. 2023) proposes an approach to improve cross-project software defect prediction by utilizing data from source projects to identify defective software modules in target projects. The method involves two stages: transformation and feature selection. The transformation phase aims to reduce the difference in data distribution between the source and target projects, while the feature selection phase removes irrelevant features using filter-based techniques. Evaluation on four datasets from the AEEEM archive compares the proposed method with four existing methods, demonstrating the highest F1 score and indicating its effectiveness and robustness for cross-project software defect prediction. Another study (Aziz, N., & Aftab, S. 2021). focuses on improving the accuracy of software defect prediction across projects, a challenging task due to project variance. The proposed approach combines feature selection, transfer learning, and ensemble learning to address issues such as feature correlation, data distribution, and model diversity. Evaluation on six benchmark datasets compares the method with existing approaches, reporting superior performance in terms of F-measure, G-mean, and

AUC. A new framework for inter-project defect prediction (HCPDP) is proposed, comprising three phases: feature classification and selection, metric matching, and defect prediction. HCPDP can handle projects with different characteristics and metrics by selecting relevant characteristics and matching them across projects. Evaluation on different datasets from three open-source projects compares HCPDP with within-project defect prediction (WPDP) and XG improvement classification baseline model, showing comparable performance to WPDP and the XG enhancement providing the best results among used classifiers.
The technique is compared with state-of-the-art methods on the PROMISE dataset, demonstrating superior performance on various evaluation metrics.
In order to improve the performance of software defect prediction models using transfer learning and proposes a framework called APOPT that automatically optimizes parameters for different transfer learning methods based on various evaluation metrics. The technique of cross-project mutation predictive testing, which leverages machine learning models to predict the outcome of running mutants without actually running them. The study evaluates the effectiveness and efficiency of the technique using a large dataset of real-world designs and 1.2 million mutants, demonstrating high accuracy and significant time savings compared to traditional mutation tests.
Moreover, improving the accuracy of software defect prediction by extracting semantic features from source code. The proposed method involves code preprocessing, semantic feature extraction, and defect feature extraction, achieving higher F-measures than existing methods in all evaluated projects. The challenge of predicting software defects in diverse projects with high-dimensional and heterogeneous data. The paper proposes dimensionality reduction algorithms such as PCA, LDA, and t-SNE, showing that some methods, like PCA and t-SNE, significantly improve model accuracy and recall compared to using raw data or random projections. Another article (Vijayaraj, N., & Ravi, T. N, 2021) tackles the problem of CPDP, introducing a training data selection method that considers defect label mismatch and project similarity. The method filters out irrelevant projects and balances bad labels across selected projects, demonstrating improved CPDP model performance in terms of accuracy, recall, precision, and F-measurement. The challenge of CPDP is addressed by proposing a new method called DeepCPDP, which combines features generated by deep learning and manually built features to reduce feature mismatch and deployment differences between different projects. The proposed method outperforms state-of-the-art baselines in terms of accuracy, recall, F-measure, and AUC. In the paper (Nawaz, Z., Aftab, S., & Anwer, F. 2017). , the authors address the heterogeneous cross-project defect prediction (CPDP) problem. They propose a model consisting of three parts: feature classification and selection, metric matching, and binary classification. The model uses metric selection and matching strategies to select relevant features from source projects and match them to target projects. A gradient-boosted classifier is employed for binary defect classification. The approach aims to handle challenges like feature mismatch, data imbalance, and transfer learning issues, improving the accuracy and reliability of defect prediction models. Another paper (Vijayaraj, N., & Ravi, T. N, 2021) introduces the MSCPDP approach to CPDP. MSCPDP considers the distribution of multiple data at the same time and uses a weighted clustering algorithm and feature selection algorithm to reduce the difference in data distribution between different projects. The method incorporates a feature selection algorithm based on mutual information to select the most relevant features for defect prediction. Evaluation on two public datasets shows that MSCPDP outperforms existing methods on F1 and AUC metrics. In the paper (Aziz, N., & Aftab, S. 2021)., the authors propose a method for predicting software defects in projects without historical data by leveraging data from other projects. Particle swarm optimization is used for feature selection from source terms, and feature-dependent Naive Bayes is employed to build a predictive model considering feature dependencies. The method aims to improve the accuracy and recall of CPDP and reduce distribution mismatch between projects. Evaluation on four public datasets supports the effectiveness and robustness of the proposed method. An algorithm utilizes clustering and active learning to filter and label representative data in both the target and source projects. The goal is to create a balanced dataset across projects for

building a defect prediction model. The proposed three-step model includes feature classification and selection, metric matching, and binary classification. The study evaluates the impact of feature extraction on model performance using three machine learning classifiers and benchmark datasets, concluding that feature extraction significantly improves prediction accuracy for certain dataset pairs, with the GBM classifier providing the best performance. The SLA+ algorithm to enhance the performance of cross-project heterogeneous defect prediction. SLA+ improves upon the selective learning algorithm (SLA) by selecting the source dataset with the greatest similarity to the target dataset and utilizing one or more intermediate datasets to bridge the gap between source and target domains. The method also incorporates feature selection to reduce dataset dimensionality, achieving comparable results to state-of-the-art methods in most cases.

A paper (Aziz, N., & Aftab, S. 2021) introduces Balanced Distribution Adaptation (BDA) as a method to adapt source data to target data by minimizing distribution differences. BDA assigns different weights to target entries based on the importance of source samples and employs a balancing strategy to address class imbalance. Evaluation on four public datasets using six metrics demonstrates that BDA achieves the best performance on most parameters, indicating its effectiveness and robustness for CPDP. A framework is proposed that utilizes feature selection and embedding techniques to extract relevant information from source code and crash reports. Deep learning models and transfer learning methods are then employed to train and predict fault persistence in crash cases across multiple projects. Experimental results from four open-source projects support the claims of improved accuracy and efficiency. The paper (Jahanshahi, H., Cevik, M., & Başar, A, 2021) introduces a model named CPDDPM (Cross-Project Dynamic Defect Prediction Model) to predict the number of defects in crowd-sourced testing. CPDDPM uses data from multiple source projects to build an initial prediction model and dynamically updates the model based on feedback from crowd-sourced testers. Experimental results show that CPDDPM achieves higher prediction accuracy and lower prediction error compared to other existing models. Another study (Sharma, U., & Sadam, R. 2023) investigates the use of different machine learning classifiers to predict software defects in different projects. The paper introduces seven combinatorial algorithms that combine multiple classifiers to improve the accuracy and cost-effectiveness of defect prediction. The composite algorithm is compared to the CODEP Logistic technique, demonstrating improvements in various parameters. To improve CPDP by using a random scaling algorithm for feature selection from source design data. Decision trees are then employed to predict defects in the target design using a simplified dataset and a Naive Bayes classifier. The method is reported to be superior to existing CFS methods. A new method for CPDP, utilizing two different feature selection strategies based on maximum information coefficients and a binary particle swarm optimization algorithm. The approach aims to overcome distribution differences between source and target terms. Evaluation on different software projects shows that the method outperforms comparable techniques on various performance metrics. A research-based feature selection method is proposed, using a genetic algorithm to select relevant and transferable features across projects. The method defines a fitness function considering classification accuracy and the similarity of characteristics between the source and target elements. Experiments on various open-source software projects compare the proposed method with four existing methods. Another study (Sharma, U., & Sadam, R. 2023) introduces the BurakMHD method, combining the Burak filter and Mahalanobis distance to filter out irrelevant or noisy data from multiple sources. Evaluation on a bug prediction dataset with 38 software projects demonstrates that BurakMHD achieves significant improvements in precision, recall, F-measure, and AUC-ROC compared to other methods. The approach is reported to reduce the size of training data without compromising prediction accuracy. A CPDP is to predict software defects across projects during the unit testing phase. CPDP uses singular value decomposition (SVD) to reduce the dimensionality and variance of different project data and employs a weighted voting scheme to combine forecasts from multi-source projects. The authors present the ARRAY method (Adaptive Triple Feature Weighted Transfer Naive Bayes) as

a transfer learning technique that adjusts feature weights and class priors of a Naive Bayes classifier based on feature similarity between the source and target datasets. The approach outperforms other methods on F-measure, G-average, and AUC-ROC on various datasets, addressing challenges like data heterogeneity, distribution mismatch, and feature selection.

In the paper (Sharma, U., & Sadam, R. 2023), the authors propose the Correlation Metric Selection Correlation Alignment Selection (CMSCA) approach for improving CPDP performance. CMSCA has two phases: relevance measure selection and relevance alignment. The first phase selects a subset of parameters highly correlated with defects in both source and target projects, discarding irrelevant or redundant parameters. The second phase adjusts the distribution of selected indicators by minimizing the distance between individual correlation matrices. Evaluation on five publicly available defect datasets shows that CMSCA achieves better or comparable prediction accuracy compared to the baseline model without requiring parameter selection or tuning. A paper (Sheng, L., Lu, L., & Lin, J, 2020) introduces a CPDP method combining source selection and transfer defect learning (TDL). The method addresses challenges like data sparsity, data imbalance, and differences in data distribution. The three-phase approach includes source selection, feature extraction, and TDL. Source selection is based on the similarity of defect distribution and feature distribution between source and target elements. Feature extraction involves static code metrics and semantic features derived from word embedding. TDL incorporates an improved TrAdaBoost algorithm for transferring defect knowledge from source to target projects. Evaluation on open-source projects, comparing with six baseline methods, shows that the proposed method outperforms the baseline in terms of AUC, F1 score, and G-mean. In the paper (Lei, T., Xue, J., & Han, W, 2020) , the authors propose an unsupervised learning model for predicting software defects in different projects with varying characteristics and data distribution. The model clusters source and target designs into a common latent space using a projection matrix, and a quantum crow search optimization algorithm selects the best subset of features. Evaluation on real-world datasets and comparison with six existing clustering models demonstrate that the proposed model achieves higher accuracy, F-measure, G-mean, and AUC while avoiding local convergence and over fitting. Another paper (Vijayaraj, N., & Ravi, T. N, 2021) suggests a comprehensive feature extraction method by selecting static code metrics and process metrics from GitHub projects with at least 10,000 commits. The assumption is that more commits lead to more bugs, and different metrics capture various aspects of software quality. Four machine learning algorithms (random forests, support vector machines, k-nearest neighbors, and naive Bayes) are used to build defect prediction models. Evaluation on different cross-project datasets indicates that the proposed method outperforms existing methods in terms of precision, recall, F-measure, and AUC-ROC. A feature transfer method (FTM) aiming to reduce distribution differences between source and target projects using deep neural networks. The method also proposes a weighted loss function (WLF) to correct class imbalance. Evaluation on 12 open-source projects from the PROMISE repository, comparing with four basic methods, demonstrates that the proposed method achieves better performance in terms of AUC, F1-score, and G-averages, showcasing the effectiveness and robustness of the feature and function transfer methods. The ADCNN (Adversarial Discriminative Convolutional Neural Network) approach is proposed, consisting of three parts: feature extractor, classifier, and discriminator. Adversarial learning is employed to train ADCNN, where the feature extractor generates transferable features invariant to the project domain. Evaluation on 10 benchmark projects from four open-source software systems, comparing with TCA+, PTA, DIP, and other related CPDP methods, indicates that ADCNN outperforms other methods in most projects in terms of F-measure, AUC, and PofB20. A study (Lei, T., Xue, J., & Han, W, 2020) presents the complex Fuzzy method to improve the performance of CPDP models. This approach addresses challenges such as data heterogeneity and imbalance by using metrics that measure the complexity of instances in the source and target projects. Metrics include code size, cyclomatic complexity, cohesion,coupling, and defect density. The method assigns each instance a membership value based on its similarity to the target project and its defects, selecting the most

relevant and defective instances for CPDP model training. The proposed complexity-based feature selection technique captures heterogeneity and imbalance in the data, enhancing the selection of relevant and defective instances for CPDP.

3. METHODOLOGY

The main purpose of this paper is to comprehensively review and analyze the role of feature selection techniques in predicting software defects across projects. Specifically, this paper aims to identify and classify various feature selection methods used in CPDP studies. This paper evaluates the effectiveness of these feature selection methods in improving prediction performance and the impact of different datasets, evaluation metrics, and machine learning algorithms used in CPDP studies.

A study (Sheng, L., Lu, L., & Lin, J, 2020) proposes a CPDP method based on feature selection and transfer learning. The method consists of two steps: feature selection and model building. Feature selection aims to reduce feature heterogeneity by selecting the most relevant and informative features for defect prediction. The feature selection method used in this paper is the MIC (Maximal Information Coefficient) method. This is a measure of the strength of a linear or nonlinear relationship between two variables. The MIC method can filter out redundant features and reduce the dimensionality of data. The paper uses the MIC method to select the most relevant and informative features for defect prediction. The paper points out that the MIC method can achieve better results than other feature selection methods, such as correlation-based feature selection (CFS) and information gain (IG). Model building aims to overcome data distribution mismatch by transferring knowledge from source projects to target projects. The paper used the TrAdaboost algorithm, a transfer learning technique that assigns higher weights to samples misclassified by multiple classifiers to build a predictive model. This paper evaluates the proposed method on some open-source projects in the PROMISE repository. The paper compares the proposed method with four existing methods: Naive Bayes (NB), Support Vector Machines (SVM), Transfer Component Analysis (TCA) and Transfer Naive Bayes (TNB). The paper reports that the proposed method can achieve better results in terms of AUC (area under the curve) and F1 score compared to existing methods. This paper also analyzes the impact of different factors, such as feature selection threshold, number of source projects, and number of boosting iterations, on the performance of the proposed method.

A software tool called MSCPDPLab, which stands for Multi-Source CPDP Laboratory. The tool is based on MATLAB, a programming language and environment for numerical computation and visualization. The tool implements a variety of software defect prediction methods, the task of identifying software modules that may contain faults or errors. The MSCPDPLab tool provides a user-friendly interface and comprehensive documentation for software defect prediction using transfer learning-based methods. The tool supports various types of data sources such as code metrics, process metrics, and text features. The tool also supports different types of transfer learning methods such as instance-based, feature-based, and model-based methods. The tool allows users to easily compare the results of different methods and evaluate their effectiveness. According to the paper, the feature selection technique used by MSCPDPLab is feature-based transfer learning. This technique aims to find a common feature space for different source and target domains, and then use the features of the source domain to train a defect prediction model for the target domain. This paper implements four feature-based transfer learning methods in MSCPDPLab: transfer component analysis (TCA), transfer subspace learning (TSL), L2, 1-norm transfer feature learning (TFL), and joint distribution adaptation (JDA). These methods use different strategies to reduce the distribution difference between source and target domains and select the most relevant features for defect prediction. The paper compares the performance of these methods with other baseline methods on multiple datasets and shows that feature-based transfer learning can improve the accuracy and recall of defect prediction models.

A paper (Zou, J., Li, Z., Liu, X., & Tong, H, 2023) addresses the class imbalance problem in heterogeneous CPDP, a technique for predicting software defects across projects using different sets of metrics. According to the paper, the feature selection technique used is an optimization method for the class imbalance problem in heterogeneous cross-project defect prediction. The paper

describes the technique as follows: The technique consists of three steps: feature selection, feature transformation, and defect prediction. The feature selection step uses a genetic algorithm (GA) to select the best subset of features from the source and target projects. The feature transformation step uses principal component analysis (PCA) to reduce dimensionality and align the features of the source and target projects. The defect prediction step uses a support vector machine (SVM) to classify defects in the target project based on the transformed features. The paper claims that the technique can handle the class imbalance problem and improve the accuracy of heterogeneous cross-project defect prediction.

Another paper (Kalaivani, N., & Beena, R, 2022) proposes the use of search-based selection (SBS) methods, an optimization method for finding the best subset of training data from a pool of candidate projects according to some fitness function. This paper shows that feature sets have a significant impact on the performance of SBS methods and that using only static code metrics can lead to poor results. The paper proposes to use a combination of static code metrics, process metrics, and change metrics as the feature set. The validation dataset selection method had no significant impact on the performance of the SBS method, and both random and stratified methods produced similar results. This article recommends the random method because it is simpler and faster. The size of the training data has a significant impact on the performance of the SBS method, and better results can be obtained with a variable size than with a fixed size. This article recommends using a variable size because it can adapt to the characteristics of each target project. The fitness function has a significant impact on the performance of the SBS method, and better results can be obtained using the F-measure than using the accuracy. The paper recommends using F-measure because it balances precision and recall.

A study (Lei, T., Xue, J., & Han, W, 2020) proposes a CPDP method that combines source selection and transfer learning, called MZTCA+. Source selection is the process of selecting the source projects that are most relevant to the target project based on certain criteria such as similarity or diversity. Transfer learning is the process of adapting source data to target data by enhancing feature transferability. MZTCA+ consists of four steps: 1) source selection based on manifold ranking and cluster analysis, 2) feature extraction based on manifold learning, 3) feature selection based on correlation analysis; 4) defect prediction based on ensemble learning. The paper evaluates MZTCA+ on many open-source projects and compares it with several baseline methods such as Naive Bayes, Logistic Regression and TCA+. The feature selection technique used in this paper is correlation analysis. Correlation analysis is a method of measuring the strength and direction of the relationship between two variables. The paper utilizes correlation analysis to select features that are highly correlated with defect labels and less correlated with each other. Therefore, this paper aims to reduce feature dimensionality and redundancy and improve defect prediction performance.

A paper (Zou, J., Li, Z., Liu, X., & Tong, H, 2023) addresses the problem of CPDP, which aims to use data from source projects to predict software defects in target projects. According to the paper, existing CPDP methods face two major challenges: data distribution mismatch and class imbalance. A data distribution mismatch means that the source and target projects have different feature spaces and statistical characteristics. Class imbalance means that the number of defective and non-defective instances is highly skewed, which makes it difficult to learn a balanced classifier. This paper proposes a new approach to overcome these challenges using transfer learning and class imbalance learning techniques. Transfer learning is a method for transferring knowledge from one domain to another, while class imbalance learning is a method for dealing with imbalanced datasets. This paper presents the two main components of the method: a few-shot oversampling technique for transfer learning (TOMO) and a feature-weighted transfer naive Bayesian (FWTNB) method. TOMO is a technique that uses a clustering-based algorithm to generate synthetic minority instances in source projects that are similar to target projects. FWTNB is a method that uses a Naive Bayesian classifier to assign different weights to different features based on their relevance to the target project. The paper evaluates their method on some public defect datasets and compares it with existing state-of-the-art CPDP methods. The paper uses two metrics to measure the performance of the CPDP model: G-Measure and MCC. G-measure is

the harmonic mean of precision and recall, while MCC is the correlation coefficient between predicted and actual labels. The paper reports significant improvements in G-Measure and MCC metrics, showing that their method can effectively predict software defects across different projects. The paper also analyzes the impact of different parameters and settings on its method, such as the number of clusters, the number of synthetic instances, and the feature selection method. The feature selection technique they used is called Feature Weighted Transfer Naive Bayes (FWTNB). The technique uses a Naive Bayesian classifier to assign different weights to different features based on their relevance to the target project. The authors claim that the technique can reduce the data distribution mismatch between source and target projects and improve the performance of CPDP models. The paper also mentioned that they used an information gain-based feature selection method to filter out irrelevant features before applying FWTNB. They compared their method to other feature selection techniques, such as principal component analysis (PCA), linear discriminant analysis (LDA), and correlation-based feature selection (CFS).

There is another study (Zou, J., Li, Z., Liu, X., & Tong, H, 2023) that proposes a hybrid feature selection method combining random forest and recursive feature elimination cross-validation (RF-RFE-CV) to select the most important features of CPDP. The technique aims to select the most important features for CPDP by ranking features according to importance scores and iteratively eliminating the least important features. The paper claims that the technique can reduce the dimensionality of the feature space and improve the performance of classifiers. The paper also applies a convolutional neural network (CNN) as a classifier, using selected features to predict defects in target projects. This paper evaluates the proposed method on 14 versions of four open-source software projects, which are multiclass, meaning they have more than two classes of flaws. The paper reports that the proposed method achieved an average prediction accuracy of 78% across all versions, as measured by AUC (a measure of the trade-off between true positive rate and false positive rate). The paper concludes that the proposed method is efficient and robust for CPDP in multi-class datasets and outperforms several existing methods.

A paper (Shabib Aftab, et al., 2018) poses the problem of defect prediction and explains the challenges of applying CPDP to different projects. This article is about software defect prediction, a technique for identifying and prioritizing the parts of a software system that are most likely to contain defects. The paper proposes a new method called heterogeneous defect prediction (HDP), which can transfer knowledge projects from one software system to another without requiring a common metric. This paper also introduces the idea of HDP and outlines the contributions and organization of this paper. The paper describes the details of HDP, which consists of three steps: data preprocessing, feature extraction, and defect prediction. This paper explains how HDP uses a novel feature extraction technique based on deep neural networks to handle heterogeneous data sources and metrics. The paper also introduces the algorithm and architecture of HDP. The paper also compares HDP to CPDP, the previous state-of-the-art CPDP method. The paper claims that HDP can achieve better performance than CPDP in some cases, but also acknowledges the limitations and challenges of HDP.

A study (Vijayaraj, N., & Ravi, T. N, 2021) proposes a new classification framework called Hybrid Inducer Ensemble Learning (HIEL), which combines multiple inducers (such as decision trees, support vector machines, and neural networks) to create an ensemble model of CPDP. This paper also proposes three project-specific performance metrics, such as Percentage Perfectly Cleaned (PPC), Percentage Not Perfectly Cleaned (PNPC), and False Omission Rate (FOR), to evaluate the benefits of the CPDP model for software projects. This paper evaluates the proposed model and performance metrics using data from PROMISE, NASA, and AEEEM repositories. The paper claims that HIEL outperforms many existing CPDP models in terms of F-measure and AUC. The paper also shows that HIEL can reduce the cost, service time, and failure rate of software projects through accurate predictions. The feature selection technique used in their proposed model is called Feature Ranking and Selection (FRS). The technique is a filtering method that ranks features based on their relevance to the target variable and selects the top k features for classification tasks. The paper uses the information gain metric to measure the relevance of each feature and uses the

optimal number of features (ONF) algorithm to determine the optimal value of k. The paper claims that FRS can improve the performance of CPDP models by reducing the dimensionality and noise of the feature space. The paper also compares FRS with other feature selection techniques such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Correlation-based Feature Selection (CFS), showing that FRS outperforms -curves under F-measure and area.

Another paper (Vijayaraj, N., & Ravi, T. N, 2021) presents a novel method for CPDP based on cognitive metrics and sampling boosting. Cognitive measures are derived from the cognitive complexity of software code, reflecting the difficulty of human developers understanding and modifying the code. Sampling boosting is a technique combining sampling and boosting to deal with class imbalance and distribution mismatch in CPDP. Feature selection techniques consist of two different strategies: one is non-iterative and the other is iterative. The non-iterative strategy is called MIC_SM_FS, which selects features that are important and have a similar distribution to the corresponding target features. Feature importance is measured using the maximum information coefficient (MIC), and feature distribution similarity is calculated using 10 statistical metrics. The iterative strategy is called BPSO_FS, which uses the binary particle swarm optimization (BPSO) algorithm to select representative features of CPDP. This paper evaluates the proposed method on many open-source projects and compares it with six existing CPDP methods.

In this paper (Sheng, L., Lu, L., & Lin, J, 2020), we propose a new model for CPDP based on two-stage feature importance amplification (TFIA). CPDP is a method of using labelled data from other projects to build models to predict defects in the current project. TFIA is a technique that aims to reduce the problems of class imbalance and different distribution of data between projects through filtering, feature selection, and classification. The authors describe the details of their proposed model, which consists of two stages: a domain adaptation stage and a classification stage. In the domain adaptation stage, the authors use an Euclidean distance-based filtering method to select the most similar source and target instances and then apply correlation-based feature selection and greedy best-first search to amplify features with strong correlations. The authors use random forest as a classifier to further amplify the importance of highly correlated features and build a model sensitive to them. They conduct ablation experiments and comparative experiments to demonstrate the effectiveness and efficiency of their model. They used the AEEEM database as a benchmark dataset for CPDP. They compare the model against six baseline methods and report various performance metrics such as accuracy, precision, recall, F-measure, and AUC. The results show that TFIA achieves significant improvements over CPDP on all metrics. The authors also analyzed the impact of different components of TFIA on CPDP performance and provided some insights for future research directions. The feature selection techniques used in this paper are correlation-based feature selection and greedy best-first search. This technique aims to amplify the importance of features that are strongly correlated with class labels and reduce redundancy among features. The authors explain that this technique can help improve the performance of CPDP by selecting the most relevant and informative features from source and target projects. The authors claim that this feature selection technique can help TFIA achieve better results compared to other methods that use different techniques such as information gain and chi-square.

A paper (Kalaivani, N., & Beena, R, 2022) proposes a new method for heterogeneous cross-project defect prediction (HCPDP), which is the task of predicting software defects by utilizing data from different projects with different distributions and metrics. This paper addresses two major challenges in HCPDP: class imbalance and shallow feature learning. Class imbalance refers to the situation where the number of defective and non-defective instances is not equal, thus affecting the performance of the classifier. Shallow feature learning refers to the limitation that traditional machine learning algorithms cannot capture the deep semantic features of software indicators. This paper proposes a hybrid deep learning classifier of the Improved Synthetic Minority Oversampling Technique (ISMOTE) and Golden Eagle Optimized Siamese Neural Network (GEO-SNN) for HCPDP. ISMOTE is an enhanced version of SMOTE, a widely used oversampling method that generates

synthetic instances for minority classes. GEO-SNN is a combination of Siamese Neural Network (SNN), a deep learning model that can learn similarity between pairs of instances, and Golden Eagle Optimization (GEO), a Meta heuristic algorithm that can optimize SNN parameter algorithms. The feature selection technique used in this paper is metric matching, a technique that uses Spearman rank correlation to align features according to the ranks of the source and target datasets. Z-score normalization applies metric matching, a scaling method that transforms features into a standard normal distribution. Metric matching aims to reduce the heterogeneity between source and target datasets with different distributions and metrics. Metric matching can improve the performance of defect prediction by selecting the most relevant and informative features for each dataset.

A paper (Sheng, L., Lu, L., & Lin, J, 2020) proposes a framework called BSLDP, which stands for Bidirectional Long Short-Term Memory Networks with Self-Attention Mechanism, for deep learning-based CPDP. The main components of BSLDP are ASL: Semantic Extractor, which uses a bidirectional LSTM network with a self-attention mechanism to generate semantic vectors of source code files. ASL can handle variable-length source code files and capture long-term dependencies and local features of the code. BSL: A classification algorithm that uses a balanced sampling strategy and a logistic regression model to build a defect prediction model from the semantic vectors of source and target projects. BSL can handle the problem of class imbalance and the problem of differences in data distribution between projects.

The paper evaluates BSLDP on a publicly available dataset called PROMISE, which contains 34 projects from different domains and languages. The paper compares BSLDP to four state-of-the-art methods: TCA+, Burak, CLAMI, and CPDP-CNN. The paper uses the F1 score as the evaluation index, which is the harmonic mean of precision and recall. The paper reports that BSLDP outperforms other methods by 14.2%, 34.6%, 32.2%, and 23.6% on average, respectively, in terms of F1 scores. The paper uses a feature selection technique called ASL (semantic extractor), using a bidirectional long short-term memory network with a self-attention mechanism. ASL is a deep learning model that can extract semantic information from source code files and generate semantic vectors for them. ASL can handle variable-length source code files and capture long-term dependencies and local features of the code. The paper concludes that BSLDP is an efficient and robust CPDP framework that can exploit the semantic information of source code files and overcome the challenges of data heterogeneity and class imbalance.

A novel approach to transfer knowledge from source projects with abundant labelled data to target projects with little or no labelled data using adversarial learning. The method consists of three parts: feature extractor, classifier and discriminator. The feature extractor maps the data of the two projects into a common feature space. A classifier predicts defect labels based on the extracted features. The discriminator tries to distinguish source and target data based on the features, while the feature extractor tries to fool the discriminator by making the features indistinguishable. By optimizing these components in an adversarial manner, the method can align the feature distributions of different projects and reduce domain differences. This can improve the performance of defect prediction models for target projects. The paper evaluates the proposed method on some open-source projects and compares it with several state-of-the-art baselines. The results show that the method achieves significant improvements in accuracy, precision, recall and score. This paper uses a feature selection technique called discriminative adversarial feature learning (DAFL). The technique is based on the idea of adversarial learning, a type of machine learning that involves two competing models: a generator and a discriminator. The generator tries to create fake data that looks like real data, while the discriminator tries to distinguish real data from fake data. By training these models in an adversarial manner, they can both improve performance and learn useful features from the data.

Another paper (Lei, T., Xue, J., & Han, W, 2020) proposes a new method called ALTRA, which combines active learning and TrAdaBoost, utilizes labeled and unlabeled data from different projects, and reduces the negative effects of domain differences and class imbalances. The paper evaluates ALTRA on 10 large open-source projects from different domains and compares it with seven state-of-the-art baselines. The results show that

ALTRA achieves better performance on both F1 and AUC metrics. The paper also analyzes the effectiveness of different components of ALTRA, such as Burak filters, active learning strategies under uncertainty, class imbalance learning methods, and the TrAdaBoost algorithm. This paper provides some insights into how these components can help improve CPDP. The paper uses a feature selection method called the Burak filter, which is a variant of the ReliefF algorithm. The Burak filter is designed to handle imbalanced datasets and reduce the noise sensitivity of the original ReliefF algorithm. The paper claims that Burak filters can be ordered by weights, which are computed based on the distance between instances of different classes, to efficiently select the most relevant features for CPDP. The paper also compares the Burak filter with other feature selection methods such as chi-square, information gain, and correlation-based feature selection and the results show that the Burak filter achieves better results in terms of both F1 and AUC metrics.
A paper by (Bhat, N. A., & Farooq, S. U. 2023) compares different defect prediction methods, such as code metrics, process metrics, defect a priori, and entropy metrics, and evaluates their performance in both intra-project and cross-project environments. Feature selection techniques are based on information gain and correlation-based feature selection, measuring the correlation and non-redundancy of features for defect prediction. The technique uses a wrapper approach to evaluate the performance of features with machine learning models and cross-validation experiments. The technique performs sequential floating forward selection (SFFS), adding and removing features to a subset until no more improvements can be made. This paper uses different evaluation scenarios and metrics such as precision, recall, F-measure, accuracy and area under the curve (AUC) to measure the effectiveness of defect prediction methods. The paper claims that process metrics, which capture change and activity during software development, are the best predictors of software defects, followed by churn and entropy metrics, which capture source code complexity and instability. The paper also claims that previous defect metrics (using historical defect information for software modules) and single metric approaches (using only one type of metric) are the worst predictors of software defects. This paper argues that intra-project and cross-project environments have different characteristics, such as project size, domain, language, and defect distribution, and that results from one environment cannot be generalized to the other.
An article by (Lei, T., Xue, J., & Han, W, 2020) discusses JIT error prediction for mobile applications, which aims to improve testing efficiency by identifying error-prone code changes. It evaluates their relevance in the mobile environment. They analyzed 14 different open source Android applications and the dataset contains 42,543 submissions. The study employed information gain for feature selection, balanced by SMOTE and employed various classifiers and ensemble techniques. It answers key research questions: Identify relevant metrics, including code changes, proliferation, history, and experience, Different classifiers affect predictive performance; naive Bayes performs well, while ensemble methods show limited improvement, Due to the ever-evolving nature of mobile applications, cross-project JIT bug prediction models are extremely valuable. Performance metrics include F-Measure, MCC, and AUC-ROC, and are statistically validated. The study provides insights into mobile-specific error predictions, highlighting the potential for timely defect detection in dynamic mobile development environments.
There is another study (Reddy, J. M, et al., 2022) that proposes a new method for CPDP, which aims to improve prediction accuracy by reducing the variance between projects. The method focuses on feature selection and distance-weighted instance transfer. The authors propose the WCM-WTrA model, which combines similarity-based feature selection with advanced TrAdaBoost transfer learning. The framework of the model involves two stages: feature selection and transfer learning. In feature selection, features are selected based on similarity and importance scores. Distance-weighted transfer learning involves the calculation of initial weights based on distance and a weight supplementary factor for adjusting weight updates. The Multi-WCM-WTrA model extends this approach to multi-source forecasting. Experiments on the AEEEM and ReLink datasets show significant improvements over existing methods. Overall, this paper proposes an innovative approach for CPDP that addresses feature

differences and enhances knowledge transfer through weighted learning.

The paper (Kalaivani, N., & Beena, R, 2022) discusses CPDP using a genetic algorithm-based feature selection method (GAFS) in a cross-project environment. As software plays an increasingly important role in everyday life, ensuring its reliability is critical. CPDP is used to predict defects in new projects with limited historical data using relevant project data. However, data distribution differences and class imbalance affect the effectiveness of CPDP. GAFS is proposed to enhance CPDP by using a genetic algorithm for feature selection and the EasyEnsemble method to alleviate class imbalance. GAFS consists of two stages: feature selection and ensemble training. In feature selection, genetic algorithms adaptively search for optimal feature subsets based on historical data integration. In ensemble training, multiple Naive Bayesian classifiers are built and the final model is built through ensemble learning. Experimental results on AEEEM and Promise datasets show that GAFS is superior in improving F1-score and MCC compared to other methods. The paper's approach demonstrates great potential for enhancing software defect prediction for new projects. The feature selection method used in this paper is called "genetic algorithm-based feature selection" (GAFS). GAFS employs a genetic algorithm for feature selection, which is a meta-heuristic search technique. It involves two key stages: In the feature selection stage, GAFS utilizes a genetic algorithm to search for the best subset of features that can maximize model performance. It integrates training results from a source project dataset (with historical data) and a target project dataset (with limited or no historical data). The purpose of this phase is to iteratively evaluate different combinations of features based on their performance on the validation set to find the best subset of features. Ensemble training phase: After selecting the best feature subset, GAFS uses ensemble learning to further enhance the predictive ability of the model. The EasyEnsemble method is used to solve the class imbalance problem. Build multiple Naive Bayes classifiers using a randomly undersampled training set. These base classifiers are then aggregated via ensemble learning to form the final predictive model.

Another paper (Ozturk, M. M, 2021) introduces a new method "CPDP method based on manifold feature transformation" to address the challenges of software defect prediction. Traditional methods struggle when dealing with new projects that lack labelled data. CPDP is proposed, which utilizes labelled data of similar projects for training. However, different data distributions between source and target projects may affect the prediction accuracy. The proposed method involves transforming the original feature space into a manifold space to bridge the distribution gap. A detailed framework outlines the process: transforming source and target project data, building defect prediction models, and evaluating results using the F1 metric. Experimental studies on the Relink and AEEEM datasets demonstrate the effectiveness of the proposed method in reducing distribution variance and improving prediction performance. The method improves existing CPDP methods by addressing the feature distortion problem and outperforms baseline and popular CPDP methods, demonstrating its potential in real-world software defect prediction scenarios.

SDP aims to improve software quality and testing efficiency by identifying defects early, while CPDP addresses data limitations by leveraging knowledge from source projects to predict defects in target projects. The proposed two-stage approach involves feature selection, category reweighting, and ensemble modelling. The method consists of extracting metrics from multiple source projects and one target project, followed by feature selection using the mRMR algorithm to reduce complexity and irrelevance. To address the data imbalance problem, a category reweighting technique is applied. The core of the method is an ensemble model combining AdaBoost and Random Forest (RF) to solve the overfitting and underfitting problems. The algorithmic framework is detailed, outlining steps such as data splitting, reweighting, and ensemble construction. The proposed method is carefully evaluated on 25 software projects, demonstrating a significant improvement in score over the state-of-the-art methods. This study contributes to the field by introducing a hybrid ensemble approach to CPDP, improving predictive accuracy while overcoming the challenges posed by differences in feature assignments between source and target projects. The application of mRMR, category reweighting, and ensemble learning embodies the overall strategy for effective CPDP.

DeepCPDP, a novel method for CPDP using deep learning. In this framework, labelled datasets from source projects are preprocessed to address class imbalance. Source code is parsed into a simplified Abstract Syntax Tree (SimAST), and token vectors are extracted while considering project-independent node types. The SimASTToken2Vec method is proposed for unsupervised token embedding, exploiting contextual similarity to learn vector representations. DeepCPDP employs a BiLSTM neural network to capture semantic features from labelled vectors to enhance defect prediction. The attention mechanism refines the feature weights, and the Logistic regression classifier builds the CPDP model. Ten large-scale projects are evaluated, demonstrating the superiority of DeepCPDP over existing methods. Key findings include the effectiveness of SimASTToken2Vec, the advantages of BiLSTM over CNNs, and the impact of attention mechanisms. DeepCPDP consistently outperforms baseline methods in CPDP, demonstrating its potential to improve software quality assurance.

Another study (Ozturk, M. M, 2021) addresses the critical issue of CPDP in software development. For this, it introduces a new method using multinomial classification. The authors emphasize the importance of early identification of defect-prone classes to improve testing efficiency and reduce costs. They emphasize the limitations of regression-based approaches and advocate multinomial classification because of its ability to provide severity information. This approach requires the use of ensemble models such as gradient boosting and random forests for data acquisition, preprocessing, and classification. Perform data encoding, normalization, and feature selection in preprocessing. Evaluation metrics include F-Measure, AUC-ROC, and Mean Precision (MAP). Hyper-parameter tuning is employed to optimize classifier performance. Through empirical experiments, the authors compare the power of multinomial classification in CPDP with within-project defect prediction (WPDP). This study highlights the feasibility of multiple classifications in CPDP and provides comparable results to WPDP. The contribution of this paper is to extend the application of multinomial classification to CPDP and demonstrate its effectiveness in identifying defect-prone categories in different projects, thereby improving software quality and cost efficiency.

A research paper (Ozturk, M. M, 2021) focuses on CPDP and provides a comprehensive analysis of optimizers and classifiers according to the nature of data classes (multiclass or binary). The study re-evaluates the underlying paper's approach, which initially treated data as binary classes using the Tera PROMISE cross-project defect repository. Through exploratory data analysis (EDA), it was found that the data is multiclass. This article introduces artificial neural network (ANN) filters and K-nearest neighbour (KNN) filters and compares their performance. A recursive feature elimination (RFE) technique is used as a search-based optimizer. The results show that the performance of the ANN filter is better than that of KNN, and the results of RFE are better than the method of the basic paper. By tuning the classifier for multi-class data, the study achieved a 30% improvement over the base paper's classifier. The obtained results are consistently enhanced by 40% to 60% across various filters, and this improvement is consistent when considering multiclass data. The proposed method demonstrates a significant performance advantage of CPDP over the methods in the underlying paper, especially when dealing with multi-class data. The "ckloc" function proved to be more effective than information gain (IG) and all functions. However, caution is advised regarding the variable results of IG features. The study acknowledges the impact of training set size on the success of the chosen method and suggests further exploration of the determinants affecting CPDP. Future directions include predicting classification categories for CPDPs, providing early defect predictions to reduce resource requirements, and using available training sets for deeper analysis.

The problem of just-in-time defect prediction (JITDP) across projects, aiming to identify defect-prone code changes across different projects. Cross-project JITDP is challenging due to the heterogeneity and diversity of software projects, which may have different characteristics, quality standards, and development practices. Existing cross-project JITDP models often suffer from low performance and poor portability. The paper proposes a new approach called FENSE, which stands for Feature-Based Ensemble Modeling for Cross-Project Software Defect Prediction. FENSE exploits project characteristics such as size,

complexity, activity, and defect rate to select and combine the most transferable cross-project JITDP models. FENSE consists of three steps: (1) feature extraction, which extracts project features from source projects and target projects; (2) model selection, which selects the most portable model according to the similarity of project features; (3) model combination, using The weighted voting scheme combines the selected models. The paper compares FENSE with six state-of-the-art cross-project JITDP models and shows that FENSE achieves significant improvements in score, AUC-ROC and G-mean.

Another article (Ozturk, M. M, 2021) addresses the problem of CPDP, which involves using data from other projects to predict defects in a software project. CPDP is challenging because different designs can have different properties and defect distributions. This article proposes a new data selection method for CPDP that considers both local and global characteristics of the source code. Local characteristics are indicators to measure the complexity, coupling and cohesion of each software module. Global characteristics are parameters that capture the change history and defect density of each design. The paper argues that both types of features are important to CPDPs because they can reflect similarities and differences between elements and modules. The document data selection method is divided into two stages: project selection and table selection. In the element selection phase, the document uses a clustering algorithm to group source elements according to their global characteristics and then selects the cluster most similar to the target element as a candidate element. In the module selection stage, we use a sorting algorithm to sort the modules in the candidates according to local characteristics, and then we select the top-k modules as training data for CPDP. The article points out that this method can select high-quality and relevant data for CPDP, thereby improving the prediction performance. The article also conducts experiments on some open-source projects of Apache, Eclipse, Mozilla and JBoss. The paper uses four metrics for evaluation: precision, recall, F-measure, and AUC-ROC. The paper compares the results with several basic methods, such as random selection, all selection, within-project prediction, and other existing CPDP methods. The article reports that the proposed method outperforms all basic methods on most metrics and domains and achieves an average improvement in F-measure and AUC-ROC compared to random selection. The paper also analyzes the impact of several parameters and characteristics on the performance of his method. The article concludes that the data selection method proposed for CPDP is effective and robust and can handle the heterogeneity and diversity of software projects. The paper also suggests some future work directions, such as integrating more features, applying more advanced machine-learning techniques, and exploring more applications of the CPDP.

The software defect prediction problem, i.e. the task of identifying defective software modules before release. Software defect prediction can help software developers allocate testing resources and improve software quality. However, software defect prediction is challenging due to the lack of sufficient data and the diversity of software projects. Therefore, this paper proposes to use CPDP, that is, to use data from other projects to build a defect prediction model. This paper focuses on the problem of feature selection in CPDP, that is, the process of selecting a subset of features relevant to defect prediction. Feature selection can reduce data dimensionality and noise, and improve the performance and interpretability of defect prediction models. This article compares several feature selection techniques, such as filters, wrappers, and built-in methods, and evaluates their performance using a Naive Bayes classifier. The article also proposes two new feature selection techniques: SBS (Sequential Backward Selection) and SBFS (Sequential Backward Floating Selection), which are based on the idea of removing features one by one until an optimal subset is found.

Another paper (Ozturk, M. M, 2021) addresses the problem of predicting software defects in different projects, which is challenging due to the heterogeneity of data sources and domain shifts between source and target projects. This paper presents a new method called MHCPDP, which stands for defect prediction in multisource heterogeneous crossover design. It consists of two main stages: feature extraction and knowledge transfer. In the feature extraction stage, documents learn intermediate features from different datasets using autoencoders, which can capture common and specific information for each project. In the

knowledge transfer stage, the document uses a multi-source transfer learning algorithm to reduce the negative impact of knowledge transfer from multiple source projects to the target project. This paper evaluates the proposed method on different datasets of four open-source software systems and compares it with several baseline methods. The results show that MHCPDP outperforms the baselines in terms of accuracy, recall, precision and F-measure. The paper also performed an ablation study to analyze the effectiveness of the individual components of the MHCPDP. This paper concludes that MHCPDP is a promising technique for software defect prediction in different projects and proposes some future directions for improvement.

A paper (Ghazal, T. M, et al., 2023) is a comparative study of different feature selection techniques for CPDP, a task of identifying faulty software modules in new projects using data from previous projects. This paper evaluates three types of feature selection techniques: filters, wrappers, and population search-based methods. Filtering methods use statistical measures to rank features and select the best ones. Wrapper methods use classifiers to evaluate features and select the best subset. Population search-based methods use metaheuristic algorithms to search for optimal features. This paper uses two software defect datasets: AEEEM and ReLink, which contain information about software modules, their functionality and their defect status. This paper applies various feature selection techniques to these datasets and builds predictive models using different classifiers such as Naive Bayes, Logistic Regression, Random Forest and Support Vector Machines. The paper uses metrics such as accuracy, precision, recall, F-measure, and area under the receiver operating characteristic curve (AUC) to measure the performance of predictive models. The paper also compares the computational cost of feature selection techniques in terms of execution time.

A paper by (Wang, W, et al.,2022) proposes a new cross-project software failure prediction model aimed at improving the quality of software products. The model uses two feature selection strategies to reduce the distribution gap between source and target projects, i.e., projects for which training data is provided and projects for which prediction is required, respectively. The first strategy is based on correlation analysis, selecting features that are highly correlated with the error proneness of the source project. The second strategy is based on learning to transfer and select features most relevant to the target project. The model then combines features selected from both strategies and applies a classifier to predict the error proneness of the target project. The paper evaluates the model based on 26 cross-sectional experiments on 8 software projects and compares it with several baselines and state-of-the-art methods. The results show that the model outperforms existing methods in terms of prediction accuracy, precision, recall, F-measure and AUC-ROC. The paper also performed statistical tests to confirm the significance of the improvements. The paper concludes that the proposed model is effective and robust for predicting software failures in different projects and can help software developers and managers improve product quality.

The paper uses a kernel-based method called KDA (Kernel Distribution Alignment) to align feature distributions in a common subspace. KDA can preserve the original data structure and reduce the dimensionality of features. Selection of neighbourhood instances: This step aims to select the most relevant and representative instances from the source projects to train the defect prediction model. The paper proposes a method called NIS (Neighborhood Instance Selection) to select instances close to the target feature in the aligned feature space. NIS can reduce noise and redundancy in source data and improve prediction accuracy. The paper evaluates the proposed method on two datasets: NASA and PROMISE. This paper compares the proposed method with a classic case-based method called TCA+ (Transferential Component Analysis Plus) and a within-project method called LR (Logistic Regression). This document uses two evaluation metrics: F-measure and AUC (area under the ROC curve). The paper reports that the proposed method achieves significant improvements in F-measure and AUC compared to TCA+ on both datasets and performs comparably to LR on the NASA dataset. This paper also conducts parameter analysis and case studies to prove the effectiveness and robustness of the method. This paper concludes that the proposed method can effectively solve the CPDP problem by aligning feature distributions and selecting neighbourhood

instances.

A study (Yuan, Z., Chen, X., Cui, Z., & Mu, Y. 2020) proposes a hybrid model for CPDP, a technique that uses data or models from other projects to predict defects in software projects. The hybrid model consists of two stages: Ensemble Learning (EL) and Genetic Algorithm (GA). EL is a method that combines multiple machine learning models to improve prediction performance. Genetic algorithms are a method of optimizing model parameters using evolutionary algorithms. This article uses datasets from the PROMISE repository, a publicly available set of software defect prediction datasets. This article applies the k-means clustering algorithm to a data set to create clusters based on the characteristics of similar projects. This paper then uses GA to train and test multiple EL models on each cluster to find the optimal parameters. The paper evaluates the performance of hybrid models using the F1 score, which is a measure of precision and recall. The paper points out that the F1 score of the hybrid model is 0.666, which is higher than the existing CPDP method. The article also states that hybrid models can improve software quality and reduce testing time by more accurately predicting defects.

An article by (Wen, W., Zhang, R, et al., 2022) proposes the use of the Population Stability Index (PSI) as a measure of domain differences between source and target elements. The PSI is calculated by comparing the histograms of each feature in the source and target projects and then summarizing the differences. The paper also proposes the use of adversarial methods to verify the results of the CPDP model. Adversarial methods involve training a classifier to distinguish source and target projects based on their characteristics and then using the classifier's accuracy as a measure of domain disagreement. The paper conducted experiments on some open-source Java projects, using four CPDP models: Naive Bayes (NB), Logistic Regression (LR), Random Forest (RF), and Support Vector Machine (SVM). This paper compares the performance of the CPDP model with a baseline model using only target project data and evaluates it using two metrics: false alarm rate (FAR) and recall. The paper reports that when the source and target projects are very similar, the CPDP model can improve FAR and reduce recall compared to the baseline model. The paper also reports that when combining designs from different origins, domain divergence is reduced and the performance of the CPDP model improves. The paper concludes that domain divergence is an important factor affecting the performance of the CPDP model and that PSI and adversarial methods are useful methods for measuring and verifying domain divergence.

4. CONCLUSION

Feature selection in CPDP occurs at the intersection of multiple research areas, including machine learning, software engineering, and data mining. It serves as a bridge connecting the unique challenges of software defect prediction with powerful feature selection techniques. Differences in project size, domain, language, and defect distribution introduce significant data heterogeneity across the project environment. Feature selection methods tailored for CPDP must address these challenges by identifying and selecting features that are relevant and suitable for different project environments. Many CPDP datasets contain a large number of features, which can lead to the curse of dimensionality. Feature selection methods effectively solve this problem by reducing the dimensionality of the data, thereby improving model performance and interpretability.

Selecting information-rich features not only reduces noise but also helps build models that can be effectively generalized across projects. This is critical for accurate defect prediction when target project historical data is limited. Various feature selection techniques have been explored in the context of CPDP, including filters, wrappers, genetic algorithms, and ensemble methods. The choice of technique depends on the target data set and the specific requirements and characteristics of the project. Numerous studies have shown that careful application of feature selection methods can significantly improve predictive performance metrics such as accuracy, precision, recall, F-measure, and AUC-ROC. Throughout our exploration, we have witnessed the ingenuity and diversity of approaches used to address the multiple challenges of the CPDP. Researchers devise innovative methods spanning the fields of data analysis, machine learning and statistical techniques. These methods address issues such as domain differences, class imbalance, and feature selection inherent in the CPDP problem. A

recurring theme in these studies is the importance of engineering and feature selection. Various feature selection techniques, including filters, wrappers, and population search-based methods, were used to extract the most relevant attributes for defect prediction. Among them, the Genetic Algorithm-based Feature Selection (GAFS) method and the Burak filter showed strong capabilities in improving the prediction accuracy and alleviating the class imbalance. Robust feature selection methods have been proposed to address the challenge of differences in data distribution and ensure that predictive models are adaptable to various project environments. As software development continues to evolve, the role of feature selection in CPDP is expected to evolve as well. Future research should explore advanced techniques, address specific challenges related to imbalanced datasets and class distributions, and investigate the potential for integrating additional features and machine learning innovations.

In conclusion, this review highlights the critical role of feature selection in predicting software defects across projects. It highlights the importance of selecting relevant and adaptable features to alleviate data heterogeneity, reduce dimensionality, and improve the generalization ability of predictive models. The research findings and insights presented here provide a comprehensive understanding of the current state of research in this field while paving the way for future advances in the pursuit of more reliable and efficient software defect prediction in diverse software projects.

REFERENCES

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).

Aftab, S., Abbas, S., Ghazal, T. M., Ahmad, M., Hamadi, H. A., Yeun, C. Y., & Khan, M. A. (2023). A Cloud-Based Software Defect Prediction System Using Data and Decision-Level Machine Learning Fusion. Mathematics, 11(3), 632.

Aftab, S., Alanazi, S., Ahmad, M., Khan, M. A., Fatima, A., & Elmitwally, N. S. (2021). Cloud-Based Diabetes Decision Support System Using Machine Learning Fusion. Computers, Materials & Continua, 68(1).

Ahmed, U., Issa, G. F., Khan, M. A., Aftab, S., Khan, M. F., Said, R. A., ... & Ahmad, M. (2022). Prediction of diabetes empowered with fused machine learning. IEEE Access, 10, 8529-8538.

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. Int. J. Mod. Educ. Comput. Sci, 12(5), 29-40.

Aziz, N., & Aftab, S. (2021). Data mining framework for nutrition ranking: Methodology: SPSS modeller. International Journal of Technology, Innovation and Management (IJTIM), 1(1), 85-95.

Bhat, N. A., & Farooq, S. U. (2023). An empirical evaluation of defect prediction approaches in within-project and cross-project context. Software Quality Journal, 1-30.

Catolino, G., Di Nucci, D., & Ferrucci, F. (2019, May). Cross-project just-in-time bug prediction for mobile apps: An empirical assessment. In 2019 IEEE/ACM 6th International Conference on Mobile Software Engineering and Systems (MOBILESoft) (pp. 99-110). IEEE.

Daoud, M. S., Aftab, S., Ahmad, M., Khan, M. A., Iqbal, A., Abbas, S., ... & Ihnaini, B. (2022). Machine learning empowered software defect prediction system.

Daoud, M. S., Fatima, A., Khan, W. A., Khan, M. A., Abbas, S., Ihnaini, B., ... & Aftab, S. (2021). Joint Channel and Multi-User Detection Empowered with Machine Learning.

Ghazal, T. M., Abbas, S., Ahmad, M., & Aftab, S. (2022, February). An IoMT based Ensemble Classification Framework to Predict Treatment Response in Hepatitis C Patients. In 2022 International Conference on Business Analytics for Technology and Security (ICBATS) (pp. 1-4). IEEE.

Iqbal, A., & Aftab, S. (2019). A feed-forward and pattern recognition ANN model for network intrusion detection. International Journal of Computer Network and Information Security, 11(4), 19.

Iqbal, A., & Aftab, S. (2020). A Classification Framework for Software Defect Prediction Using Multi-filter Feature Selection Technique and MLP. International Journal of Modern Education & Computer Science, 12(1).

Iqbal, A., Aftab, S., Ali, U., Nawaz, Z., Sana, L., Ahmad, M., & Husen, A. (2019). Performance analysis of machine learning techniques on software defect prediction using NASA datasets. International Journal of Advanced Computer Science and Applications, 10(5).

Iqbal, A., Aftab, S., Ullah, I., Saeed, M. A., & Husen, A. (2019). A classification framework to detect DoS attacks. International Journal of Computer Network and Information Security, 11(9), 40-47.

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.

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.

Kalaivani, N., & Beena, R. (2022). Improved SMOTE and Optimized Siamese Neural Networks for Class Imbalanced Heterogeneous Cross Project Defect Prediction. International Journal of Intelligent Engineering & Systems, 15(2).

Lei, T., Xue, J., & Han, W. (2020). Cross-Project Software Defect Prediction Based on Feature Selection and Transfer Learning. In Machine Learning for Cyber Security: Third

International Conference, ML4CS 2020, Guangzhou, China, October 8–10, 2020, Proceedings, Part III 3 (pp. 363-371). Springer International Publishing.

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).

Nawaz, Z., Aftab, S., & Anwer, F. (2017). Simplified FDD process model. International Journal of Modern Education and Computer Science, 9(9), 53.

Omondiagbe, O. P., Licorish, S. A., & MacDonell, S. G. (2022, August). Negative Transfer in Cross Project Defect Prediction: Effect of Domain Divergence. In 2022 48th Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (pp. 1-4). IEEE.

Ozturk, M. M. (2021). complexFuzzy: A novel clustering method for selecting training instances of cross-project defect prediction. Computer Science, 22(1).

Rahman, A. U., Abbas, S., Gollapalli, M., Ahmed, R., Aftab, S., Ahmad, M., ... & Mosavi, A. (2022). Rainfall prediction system using machine learning fusion for smart cities. Sensors, 22(9), 3504.

Reddy, J. M., Muthukumaran, K., Shahriar, H., Clincy, V., & Sakib, N. (2022, June). Comprehensive Feature Extraction for Cross-Project Software Defect Prediction. In 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC) (pp. 450-451). IEEE.

Shabib Aftab, M. A., Hameed, N., Bashir, M. S., Ali, I., & Nawaz, Z. (2018). Rainfall prediction in Lahore City using data mining techniques. International journal of advanced computer science and applications, 9(4).

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.

Vijayaraj, N., & Ravi, T. N. (2021). Cross-Project Defect Prediction based on Cognitive Metrics Using Sampled Boosting. Annals of the Romanian Society for Cell Biology, 25(6), 7431-7440.

Wang, W., Zhao, H., Li, Y., Su, J., Lu, J., & Wang, B. (2022, November). Research on cross-project software defect prediction based on feature transfer method. In Proceedings of the 4th International Conference on Advanced Information Science and System (pp. 1-5).

Wen, W., Zhang, R., Wang, C., Shen, C., Yu, M., Zhang, S., & Gao, X. (2022). A Cross-Project Defect Prediction Model Based on Deep Learning With Self-Attention. IEEE Access, 10, 110385-110401.

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.

Zhu, Y., Zhao, Y., Yu, Q., & Chen, X. (2022). Cross-Project Defect Prediction Method based on Feature Distribution Alignment and Neighborhood Instance Selection. Journal of Internet Technology, 23(4), 761-769.

Zou, J., Li, Z., Liu, X., & Tong, H. (2023). MSCPDPLab: A MATLAB toolbox for transfer learning based multi-source cross-project defect prediction. SoftwareX, 21, 101286.