electronics

Article

Landmark-Based Domain Adaptation and Selective Pseudo-Labeling for Heterogeneous Defect Prediction

Yidan Chen [1] and Haowen Chen [2,*]

[1] National and Local Joint Laboratory of Cyberspace Security Technology, Zhengzhou 450001, China; 18638180100@139.com
[2] School of Cyber Science and Engineering, Information Engineering University, Zhengzhou 450001, China
* Correspondence: hwc_zzu@126.com

Abstract: Cross -project defect prediction (CPDP) is a promising technical means to solve the problem of insufficient training data in software defect prediction. As a special case of CPDP, heterogeneous defect prediction (HDP) has received increasing attention in recent years due to its ability to cope with different metric sets in projects. Existing studies have proven that using mixed-project data is a potential way to improve HDP performance, but there remain several challenges, including the negative impact of noise modules and the insufficient utilization of unlabeled modules. To this end, we propose a landmark-based domain adaptation and selective pseudo-labeling (LDASP) approach for mixed-project HDP. Specifically, we propose a novel landmark-based domain adaptation algorithm considering marginal and conditional distribution alignment and a class-wise locality structure to reduce the heterogeneity between both projects while reweighting modules to alleviate the negative impact brought by noise ones. Moreover, we design a progressive pseudo-label selection strategy exploring the underlying discriminative information of unlabeled target data to further improve the prediction effect. Extensive experiments are conducted based on 530 heterogeneous prediction combinations that are built from 27 projects using four datasets. The experimental results show that (1) our approach improves the *F1-score* and *AUC* over the baselines by 9.8–20.2% and 4.8–14.4%, respectively and (2) each component of LDASP (i.e., the landmark weights and selective pseudo-labeling strategy) can promote the HDP performance effectively.

Keywords: heterogeneous defect prediction; mixed project; domain adaptation; pseudo-label

check for updates

Citation: Chen, Y.; Chen, H. Landmark-Based Domain Adaptation and Selective Pseudo-Labeling for Heterogeneous Defect Prediction. *Electronics* **2024**, *13*, 456. https://doi.org/10.3390/electronics13020456

Academic Editors: Seifedine Kadry, Isah A. Lawal and Sahar Yassine

Received: 5 December 2023
Revised: 14 January 2024
Accepted: 18 January 2024
Published: 22 January 2024

1. Introduction

Software defect prediction technology can identify which modules (e.g., files, classes, and functions) are more likely to be defective in software projects, thereby helping developers/maintainers allocate testing resources reasonably [1,2]. However, this process often suffers from the lack of training data [3]. In order to overcome this, researchers came up with the idea of introducing external and well-labeled projects as the training data, that is, cross-project defect prediction (CPDP) [4–6]. Over the past decade, CPDP has brought about widespread attention in the field of software engineering and has made remarkable progress in alleviating the distribution difference between source and target projects' data. Conventional CPDP always holds the assumption that source and target projects have the same metrics. Therefore, if the assumption is invalid (i.e., both projects have different metrics), existing CPDP methods cannot be directly applied to defect prediction. In response to this special case of CPDP, researchers have proposed heterogeneous defect prediction (HDP) [7,8] and designed corresponding solutions to deal with the different metrics and distributions of both projects.

HDP focuses on defect prediction across those projects in which metrics are partially or completely different and thus is often viewed as a special case of CPDP. Its greatest challenge is how to overcome the obstacle brought by different metrics while reducing the distribution discrepancy between source and target data. The current related work also focuses on solving the subproblems (e.g., class imbalance and linear inseparability) derived from HDP to further improve the prediction effect without considering the labeled data in the target project.

Actually, in the early stage of software development, a small amount of labeled data (i.e., training target data) and a large amount of unlabeled data (i.e., test target data) often coexist in the target project [9]. Moreover, Turhan et al. proved the effectiveness of the combination of "within" and "cross-project" data (i.e., mixed-project data) for defect prediction [6]. Inspired by this, several mixed-project HDP methods [10–12] have been proposed to fully utilize labeled target data and show the potential to improve predictive performance. However, research on the HDP with mixed-project data is still in the initial stage, and a series of issues (e.g., negative impact of noise modules and insufficient utilization of unlabeled data) have not been fully considered and resolved. In this paper, we further explore and address issues relevant to HDP with mixed-project data for improving the prediction performance.

1.1. Motivation

1.1.1. Negative Impact of Noise Modules

The original intention of cross-project defect prediction is to improve the predictive effect of the target project by introducing external projects with sufficient historical data, but this is not the case in reality. Recent research [13] has pointed out that treating all project modules equally in the CPDP process may lead to data redundancy. In other words, equally treated modules commonly play different roles in cross-project defect prediction. Some of them (i.e., noise modules) may even inhibit learning and then weaken prediction performance. As a special case of CPDP, a similar issue also exists in HDP. The left part of Figure 1 displays an ideal distribution of the source and target data where modules can be well separated by category. Noise modules make it hard to decide the classification boundary for source and target data, as shown in the right part of Figure 1.

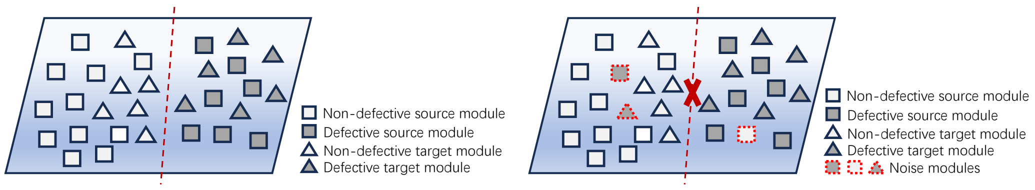
Non-defective source module
Defective source module
Non-defective target module
Defective target module
Non-defective source module
Defective source module
Non-defective target module
Defective target module
Noise modules

Figure 1. Illustration of the negative impact of noise modules.

Table 1 provides statistics on prediction combinations under different situations to illustrate the negative impact of noise modules (for detailed experimental settings, refer to Section 4.3). The WPDP method only uses labeled target data to build the classifier for predicting the remaining unlabeled ones. CLSUP is a mixed-project HDP method that trains the predictor with source data and labeled target data. As shown in this table, there are 21.5% and 20.2% prediction combinations in which WPDP performs better than CLSUP regarding the *F1-score* and *AUC*, respectively. This means that the source data are not helpful for prediction across projects in this situation (i.e., an invalid cross-project prediction combination). Therefore, we believe that noise modules must exist with negative impacts on predictions in the source project. To the best of our knowledge, current mix-project HDP methods usually consider all the modules of source and target projects equally important, which is not conducive to highlighting the effect of the relevant modules while eliminating the negative impact of the noise versions. Therefore, it is meaningful to tackle the above problem to investigate how to focus on the relevant modules while ignoring the noise versions.

Table 1. Statistics for the comparison between WPDP and CLSUP.

Situation	Number of Prediction Combinations	
	F1-Score	***AUC***
WPDP performs better than CLSUP	114	107
WPDP performs worse than CLSUP	416	423
% of invalid cross-project prediction combinations	21.5%	20.2%

1.1.2. Insufficient Utilization of Unlabeled Modules

In the existing HDP research, the unlabeled modules of the target project are commonly used to match the data distributions of both projects. It can alleviate the heterogeneity between the source and target projects so that the prediction model can be better adapted to the target data. However, this utilization of unlabeled target data does not fully explore the latent discriminant information within it, which makes it difficult to improve the classification ability of prediction models significantly. Guided by a small number of labels, semi-supervised learning can utilize a large number of unlabeled samples to improve learning performance and avoid wasting data resources. It solves the problems of the weak generalization ability of supervised learning methods when there are few labeled samples and the inaccuracy of unsupervised learning methods when there is a lack of labeled samples [14]. As an effective semi-supervised method, pseudo-label learning [15] is beneficial for determining the classification boundary in low-density regions, thereby improving the model's performance. Considering this, we further investigate how to utilize the pseudo-labels of unlabeled data from the target project to enhance the prediction effect.

1.2. Contribution

In this paper, we propose a landmark-based domain adaptation and selective pseudo-labeling (LDASP) approach for improving HDP with mixed-project data. The detailed contributions are summarized below:

- The proposal of a novel landmark-based domain adaptation algorithm that considers marginal and conditional distribution alignment and a class-wise locality structure to reduce the heterogeneity between both projects, reweighting modules to alleviate the negative impact brought about by the noise modules.
- The proposal of a progressive pseudo-label selection strategy exploring the underlying discriminative information of unlabeled target data to further improve the prediction effect.
- Extensive experiments are conducted on 27 projects from four datasets to demonstrate that our approach provides significantly better performance when compared to state-of-the-art methods, verifying the effectiveness of each component of it.

2. Related Work

In this section, we introduce the developments of heterogeneous defect prediction and domain adaptation that are relevant to this work. We first review the current HDP studies. Then, we summarize the concept of domain adaptation and detail the landmark-based domain adaptation methods.

2.1. Heterogeneous Defect Prediction

Based on the learning process of predictors, we roughly divide the existing HDP methods into two categories, i.e., conventional and mixed-project HDP methods, which correspond to different application scenarios.

2.1.1. Conventional HDP Methods

Conventional HDP methods are suitable for the scenario where the source project is well labeled and the target project is unlabeled. They train the defect predictor with labeled source data and unlabeled target data. To date, most of the existing related work belongs to this category. Jing et al. [8] were among the first to identify the HDP problem and propose an improved canonical correlation analysis (CCA) method for alleviating the heterogeneity between source and target projects. Meanwhile, Nam and Kim [16] proposed a metric selection and matching method for HDP that can remove redundant metrics for the source project and then match up the metrics of both projects (one-to-one) based on metric similarity. Based on these two methods, a series of related works [11,17–19] has emerged, and considerable progress has been made.

In order to deal with the class imbalance and linear inseparability issues, Li et al. successively proposed an ensemble multiple kernel correlation alignment (EMKCA) method [17] and a cost-sensitive transfer kernel canonical correlation analysis (CTKCCA) method [18]. Later, they also extended EMKCA to a two-stage ensemble learning framework by combining multiple predictors [20]. Similar to the idea of [16], Yu et al. [21] proposed a novel feature matching and transfer (FMT) method to select one-to-one feature pairs for heterogeneous source and target projects based on the "distance" between distribution curves. Tong et al. [19] designed a kernel spectral embedding transfer ensemble (KSETE) method to improve the prediction effect. This method first alleviates the class imbalance problem by performing the synthetic minority over-sampling technique (SMOTE) on the source project. Then, it combines kernel spectral embedding and transfer learning to find a latent common metric space for source and target data. Eventually, the prediction results are decided by ensemble learning. In addition, multi-view learning [22,23], multi-source transfer learning [24], and deep generation network [25] are introduced to improve the performance of the conventional HDP method.

In summary, the above HDP methods utilize the labeled source data and unlabeled target data in the learning process to reduce the heterogeneity between both projects effectively, showing considerable performance. However, they are not designed for a situation with a small number of labeled target data, and thus, the supervised (label) information within it cannot be utilized reasonably in predictions. In this paper, we design a novel approach that can use labeled target data to improve the discriminative ability of predictors while eliminating the heterogeneity between source and target projects.

2.1.2. Mixed-Project HDP Methods

Mixed-project HDP methods focus on the scenario of the labeled source project and the target project with a small number of labeled modules. They combine heterogeneous source data and a small number of labeled target data (i.e., training target data) to build the defect predictor. Among the current research on HDP, only a few studies provide solutions for using mixed-project data to improve prediction performance. Li et al. [11] first proposed a cost-sensitive label and structure-consistent unilateral projection (CLSUP) method. This method combines domain adaptation and cost-sensitive learning techniques to learn the projection matrix from source to target data while introducing the misclassification cost to alleviate the impact of class imbalance. In order to enhance the quality of training data, Li et al. [10] proposed a multi-source selection-based manifold discriminant alignment method. Its core component is an improved manifold discriminant alignment (MDA) algorithm that learns transformation matrices for source and target data to make their distributions closer and have a favorable classification ability. The recent work [12] by Niu et al. was an extension of MDA that first applies a sampling technique to handle the class imbalance problem and then uses the kernel manifold discriminant alignment algorithm to overcome the linear inseparability issue. Extensive experiments on 13 projects from three public datasets demonstrate its state-of-the-art prediction performance.

Overall, the current mixed-project HDP methods focus primarily on using limited labeled target data reasonably in the learning process while considering common issues, such as class imbalance and linear inseparability. Although showing promising results, they ignore the negative impact of noise modules on prediction across projects, and the underlying discriminative information in unlabeled target data is not fully excavated. In this paper, we propose a novel landmark-based domain adaptation and selective pseudo-labeling approach for mixed-project HDP to address these limitations.

2.2. Domain Adaptation

Domain adaptation has been an important research direction of transfer learning and gained a lot of success in a wide range of tasks, such as computer vision and natural language processing [26]. The core idea of domain adaptation is to learn a feature extractor that makes the data distributions of both domains aligned so that the knowledge learned from the source domain can be applied to the target one. To this end, researchers attempt to introduce different methods (e.g., marginal, conditional, and joint distributions) to measure and optimize the discrepancy between different domains [27–29]. Unfortunately, the effectiveness of transfer learning is not always guaranteed due to the unsatisfying basic assumptions, which causes negative transfer, meaning the source knowledge decreases the learning performance in the target domain [30].

In order to deal with the negative transfer issue, several sample reweighting methods [28,31,32] are proposed to bring the source distribution closer to the target one; hence, this may help eliminate the impact of the negative transfer on learning performance in the target domain. These methods select the most relevant samples (i.e., landmarks) to match the data distributions by reweighting the samples from the source or both domains. Aljundi et al. [31] designed a novel unsupervised domain adaptation approach based on the subspace alignment and selection of landmarks similarly distributed between both domains. For the heterogeneous domain adaptation with the semi-supervised setting, Tsai et al. [32] proposed a cross-domain landmark selection method to project the source data into the feature subspace of the target domain. Specifically, this method considers reducing the marginal and conditional distribution discrepancies simultaneously while selecting landmarks from both domains through the learning weights for the samples. Inspired by this, we introduce the idea of landmark-based domain adaptation to alleviate the negative impact of noise modules in this work.

Unlike the landmark-based domain adaptation methods mentioned above, we consider more comprehensive factors, including marginal and conditional distribution alignment, and class-wise locality structure preservation to further improve the effect of domain adaptation. Furthermore, we also design a progressive selection strategy to raise the quality of used pseudo-labels instead of the direct utilization of all pseudo-labels.

3. Approach

In this section, we first introduce the setting of HDP with mixed-project data; the notations used in this paper are provided in Section 3.1. Sections 3.2 and 3.3 provide detailed descriptions of our improved landmark-based domain adaptation method and the proposed progressive pseudo-label selection strategy, respectively.

3.1. Problem Statement

The only difference from the conventional setting (i.e., labeled source and unlabeled target projects with different metric sets) is the existence of a small number of labeled target modules in the mixed-project data setting. Specifically, the labeled source project can be defined as $S = \{x_s^i, y_s^i\}_{i=1}^{n_s} = \{X_S, Y_S\}$, in which x_s^i refers to the ith module of the source project, and y_s^i is the corresponding label (i.e., defective or non-defective). Similarly, the unlabeled and labeled parts of the target project can be defined as $T_U = \{x_u^i, y_u^i\}_{i=1}^{n_u} = \{X_U, Y_U\}$ and $T_L = \{x_l^i, y_l^i\}_{i=1}^{n_l} = \{X_L, Y_L\}$ separately. In this task, the source project data (S) and training target data (T_L) will be combined to build the predictor and further identify the labels (Y_U) of the test target data (T_U).

3.2. Landmark-Based Domain Adaptation

In this section, we first present the basic structure of the objective function that is used to match the distributions between the source and target data. Then, we introduce the modified objective function by considering landmark weights to alleviate the negative impact of the noise modules.

3.2.1. Matching the Distributions of Source and Target Data

Generally, there is a significant difference between the data distributions of heterogeneous source and target projects. This will make the predictor trained using source data not be able to adjust to the target data well. To this end, we learn the transformation matrix, $\mathbf{A} \in \mathbb{R}^{d_s \times m}$, for the source data to project it into the metric subspace of the target data with m dimensions. Specifically, a principal component analysis (PCA) is first used to project target data into the metric subspace of the target data with m dimensions. The maximum mean discrepancy is an effective method to measure the marginal distribution difference between two domains, which does not assume any parametric form for the distributions. Hence, we employ it to calculate the distance between the distributions of both projects and define the term of marginal distribution alignment, $E_M(\mathbf{A}, S, T_L, T_U)$, as follows:

$$E_M(\mathbf{A}, S, T_L, T_U) = \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \mathbf{A}^{\mathrm{T}} x_s^i - \frac{1}{n_l + n_u} \left(\sum_{j=1}^{n_l} \widehat{x}_l^j + \sum_{j=1}^{n_u} \widehat{x}_u^j \right) \right\|^2, \tag{1}$$

where $\widehat{x}_u$ and $\widehat{x}_l$ refer to the unlabeled and labeled target modules processed via PCA.

Ref. [28] proposed a transfer learning algorithm that jointly optimizes marginal and conditional distributions, and the authors verified its effectiveness in reducing the distribution discrepancy through extensive experiments. As the labeled modules in both projects can be used to measure the conditional distribution discrepancy, we also define the term of conditional distribution alignment $E_C(\mathbf{A}, S, T_L)$ as shown below:

$$E_C(\mathbf{A}, S, T_L) = \sum_{c=1}^{C} \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \mathbf{A}^{\mathrm{T}} x_s^{i,c} - \frac{1}{n_l} \sum_{j=1}^{n_l} \widehat{x}_l^{j,c} \right\|^2 + \frac{1}{n_s^c n_l^c} \sum_{i=1}^{n_s^c} \sum_{j=1}^{n_l^c} \left\| \mathbf{A}^{\mathrm{T}} x_s^{i,c} - \widehat{x}_l^{j,c} \right\|^2, \tag{2}$$

where the former is designed to match the conditional distributions between the source and target data approximately, and the latter further aggregates the source and target modules with the same category in the target metric subspace.

In order to preserve the structure of the transformed source data, we impose an additional class-wise locality constraint [29] on the projected source data and define the class-wise locality-preserving term $E_S(\mathbf{A}, S)$ as shown below:

$$E_S(\mathbf{A}, S) = \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} w_{ij} \left\| \mathbf{A}^{\mathrm{T}} x_s^i - \mathbf{A}^{\mathrm{T}} x_s^j \right\|^2, \tag{3}$$

where w_{ij} is defined as follows:

$$w_{ij} = \begin{cases} exp\left(-\left\| x_s^i - x_s^j \right\|^2 / \delta^2\right) & if \{x_s^i, x_s^j\} \in \mathbf{X}_S^c \\ 0 & otherwise \end{cases} \tag{4}$$

In summary, the final objective function can be integrated based on Equations (1)–(3) as follows:

$$\min_{\mathbf{A}} E_M(\mathbf{A}, S, T_L, T_U) + E_C(\mathbf{A}, S, T_L) + E_S(\mathbf{A}, S) + \lambda \|\mathbf{A}\|^2, \tag{5}$$

where $\|\mathbf{A}\|^2$ is the regularization term that controls the complexity of $\mathbf{A}$ to avoid over-fitting. λ is the parameter of the regularization term. Here, the transformation matrix $\mathbf{A}$ can be optimized by minimizing Equation (5).

3.2.2. Matching the Distributions of Source and Target Data Based on Landmarks

Although the objective function in Equation (5) considers reducing the discrepancy between both projects from different aspects, it still believes that all modules are equally important and thus ignores the negative impact brought about by noise modules. As discussed in Section 1.1.1, this may inhibit the learning process of the transformation matrix and lead to poor performance. In order to solve this problem, we weight the modules of the source and target projects and view the modules with nonzero weights as landmarks. Based on this thought, the objective function in Equation (5) can be redefined as shown below:

$$\begin{aligned} &\min_{\mathbf{A},\boldsymbol{\alpha},\boldsymbol{\beta}} E_M(\mathbf{A},S,T_L,T_U,\boldsymbol{\alpha},\boldsymbol{\beta}) + E_C(\mathbf{A},S,T_L,T_U,\boldsymbol{\alpha},\boldsymbol{\beta}) \\ &\quad + E_S(\mathbf{A},S,\boldsymbol{\alpha}) + \lambda\|\mathbf{A}\|^2, \\ &\quad s.t. \quad \alpha_i^c,\beta_i^c \in [0,1], \quad \frac{\alpha^{c\mathrm{T}}1}{n_S^c} = \frac{\beta^{c\mathrm{T}}1}{n_T^c} = \theta, \end{aligned} \tag{6}$$

where $\boldsymbol{\alpha} = [\alpha^1;\ldots;\alpha^c;\ldots\alpha^C] \in \mathbb{R}^{n_s}$ represents the weights of the source modules. Specifically, α^c represents the module weights belonging to the category c in the source project. Because a small number of labeled target modules can provide discriminative information, all of them in T_L are considered landmarks, and their weights are fixed to 1, whereas the unlabeled ones in the target project are weighted using $\boldsymbol{\beta}$. Similarly, $\boldsymbol{\beta} = [\beta^1;\ldots;\beta^c;\ldots\beta^C] \in \mathbb{R}^{n_u}$ represents the weights of the unlabeled modules in the target project, where β^c specifically refers to the weights of the target modules that are predicted as category c (pseudo-label). $\theta \in [0,1]$ controls the ratio of the source data and target test data used for distribution adaptation.

Based on the module weights $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ defined above, the extended marginal distribution alignment term $E_M(\mathbf{A},S,T_L,T_U,\boldsymbol{\alpha},\boldsymbol{\beta})$ can be computed as follows:

$$E_M(\mathbf{A},S,T_L,T_U,\boldsymbol{\alpha},\boldsymbol{\beta}) = \left\| \frac{1}{\theta n_s}\sum_{i=1}^{n_s}\alpha_i\mathbf{A}^{\mathrm{T}}x_s^i - \frac{1}{n_l+\theta n_u}\Big(\sum_{j=1}^{n_l}\widehat{x}_l^j + \sum_{j=1}^{n_u}\beta_j\widehat{x}_u^j\Big)\right\|^2. \tag{7}$$

In order to match the conditional distribution of the source and target data, we use labeled project data (S and T_L) to train the classification model and make predictions of the unlabeled module $\widehat{x}_u^i$ to generate its corresponding pseudo-label $\widetilde{y}_u^i$. With the assistance of the obtained pseudo-labels, the unlabeled module T_U can be assigned specific categories. Therefore, the extended conditional distribution alignment term $E_C(\mathbf{A},S,T_L,T_U,\boldsymbol{\alpha},\boldsymbol{\beta})$ can be defined as follows:

$$E_C(\mathbf{A},S,T_L,T_U,\boldsymbol{\alpha},\boldsymbol{\beta}) = \sum_{c=1}^{C} E_{cond}^c + \frac{1}{e^c}E_{embed}^c, \tag{8}$$

where $e^c = \theta n_s^c n_l^c + \theta n_l^c n_u^c + \theta^2 n_u^c n_s^c$. E_{cond}^c and E_{embed}^c are separately defined below:

$$E_{cond}^c = \left\| \frac{1}{\theta n_s^c}\sum_{i=1}^{n_s^c}\alpha_i\mathbf{A}^{\mathrm{T}}x_s^{i,c} - \frac{1}{n_l^c+\theta n_u^c}\Big(\sum_{j=1}^{n_l^c}\widehat{x}_l^{j,c} + \sum_{j=1}^{n_u^c}\beta_j\widehat{x}_u^{j,c}\Big)\right\|^2, \tag{9}$$

$$\begin{aligned} E_{embed}^c =& \sum_{i=1}^{n_s^c}\sum_{j=1}^{n_l^c}\left\|\alpha_i\mathbf{A}^{\mathrm{T}}x_s^{i,c} - \widehat{x}_l^{j,c}\right\|^2 + \sum_{i=1}^{n_l^c}\sum_{j=1}^{n_u^c}\left\|\widehat{x}_l^{i,c} - \beta_j\widehat{x}_u^{j,c}\right\|^2 \\ &+ \sum_{i=1}^{n_u^c}\sum_{j=1}^{n_s^c}\left\|\beta_i\widehat{x}_u^{i,c} - \alpha_j\mathbf{A}^{\mathrm{T}}x_s^{j,c}\right\|^2. \end{aligned} \tag{10}$$

It can be seen that Equation (8) is essentially extended by imposing the unlabeled data T_U and its corresponding pseudo-labels $\{\widetilde{y}_u^i\}_{i=1}^{n_u}$ to Equation (2). With the module weight

$\boldsymbol{\alpha}$ in the source project, the class-wise locality-preserving term $E_S(\mathbf{A}, S, \boldsymbol{\alpha})$ can be defined as follows:

$$E_S(\mathbf{A}, S, \boldsymbol{\alpha}) = \frac{1}{\theta^2 n_s^2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} w_{ij} \left\| \alpha_i \mathbf{A}^{\mathrm{T}} x_s^i - \alpha_j \mathbf{A}^{\mathrm{T}} x_s^j \right\|^2, \tag{11}$$

where w_{ij} can be referred to in Equation (4).

3.2.3. Solution

Because the objective function in Equation (6) is a non-convex joint optimization problem with respect to $\mathbf{A}$, $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$, we employed the iterative optimization method described in Ref. [32] to learn the transformation matrix, $\mathbf{A}$, and landmark weights, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, alternately.

(1) Optimizing $\mathbf{A}$

In order to learn the transformation matrix $\mathbf{A}$, we first consider the landmark weights $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ as the constant term. Then, we compute the first-order derivative of Equation (6) with respect to $\mathbf{A}$ and let it be equal to zero. Finally, the closed-form solution of $\mathbf{A}$ can be obtained as follows:

$$\mathbf{A} = (\lambda \mathbf{I}_{d_s} + \mathbf{X}_S \mathbf{H}_S \mathbf{X}_S^{-1})^{-1} (\mathbf{X}_s (\mathbf{H}_L \widehat{\mathbf{X}}_L^{\mathrm{T}} + \mathbf{H}_U \widehat{\mathbf{X}}_U^{\mathrm{T}})), \tag{12}$$

where $\mathbf{I}_{d_s}$ is the identity matrix with d_s dimensions. $\mathbf{X}_S \in \mathbb{R}^{d_s \times n_s}$, $\widehat{\mathbf{X}}_L \in \mathbb{R}^{m \times n_l}$, and $\widehat{\mathbf{X}}_U \in \mathbb{R}^{m \times n_u}$ represent the source data, training target data, and test target data, respectively. The element $(\mathbf{H}_S)_{i,j}$ in $\mathbf{H}_S \in \mathbb{R}^{n_s \times n_s}$ refers to the derivative coefficient associated with ${x_s^i}^{\mathrm{T}} x_s^j$. A similar explanation can be applied to $\mathbf{H}_L \in \mathbb{R}^{n_s \times n_l}$ and $\mathbf{H}_U \in \mathbb{R}^{n_s \times n_u}$.

(2) Optimizing $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$

Given the transformation matrix $\mathbf{A}$, Equation (6) can be rewritten as:

$$\begin{aligned} \min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \quad & \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}} \mathbf{K}_{S,S} \boldsymbol{\alpha} + \frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}} \mathbf{K}_{U,U} \boldsymbol{\beta} - \boldsymbol{\alpha}^{\mathrm{T}} \mathbf{K}_{S,U} \boldsymbol{\beta} \\ & + \mathbf{k}_{S,L}^{\mathrm{T}} \boldsymbol{\alpha} + \mathbf{k}_{U,L}^{\mathrm{T}} \boldsymbol{\beta} \\ & s.t. \quad \alpha_i^c, \beta_i^c \in [0,1], \quad \frac{\alpha^{c\mathrm{T}} 1}{n_S^c} = \frac{\beta^{c\mathrm{T}} 1}{n_T^c} = \theta, \end{aligned} \tag{13}$$

where the element $(\mathbf{K}_{S,S})_{i,j}$ in $\mathbf{K}_{S,S} \in \mathbb{R}^{n_s \times n_s}$ denotes the correlation coefficient associated with $(\mathbf{A}^{\mathrm{T}} x_s^i)^{\mathrm{T}} \mathbf{A}^{\mathrm{T}} x_s^j$. A similar explanation can be applied to the element $(\mathbf{K}_{U,U})_{i,j}$ in $\mathbf{K}_{U,U} \in \mathbb{R}^{n_u \times n_u}$ and the element $(\mathbf{K}_{S,U})_{i,j}$ in $\mathbf{K}_{S,U} \in \mathbb{R}^{n_s \times n_u}$. The element $(\mathbf{k}_{S,L})_i$ in $\mathbf{k}_{S,L} \in \mathbb{R}^{n_s}$ denotes the sum of coefficients of $(\mathbf{A}^{\mathrm{T}} x_s^i)\widehat{x}_l^1, (\mathbf{A}^{\mathrm{T}} x_s^i)\widehat{x}_l^2, \ldots, (\mathbf{A}^{\mathrm{T}} x_s^i)\widehat{x}_l^{n_l}$. A similar explanation can be applied to the element $(\mathbf{k}_{U,L})_i$ in $\mathbf{k}_{U,L} \in \mathbb{R}^{n_u}$. Based on the above formulations, the solution of Equation (13) is transformed into the following quadratic programming problem that can be resolved by using existing tools (i.e., the quadprog function in MATLAB R2023b).

$$\min_{z_i \in [0,1], \mathbf{Z}^{\mathrm{T}} \mathbf{V} = \mathbf{W}} \quad \frac{1}{2} \mathbf{Z}^{\mathrm{T}} \mathbf{B} \mathbf{Z} + \mathbf{b}^{\mathrm{T}} \mathbf{Z}, \tag{14}$$

where $\mathbf{Z} = [\boldsymbol{\alpha}; \boldsymbol{\beta}]$, $\mathbf{B} = [\mathbf{K}_{S,S}, -\mathbf{K}_{S,U}; -\mathbf{K}_{S,U}^{\mathrm{T}}, \mathbf{K}_{U,U}]$, $\mathbf{b} = [-\mathbf{k}_{S,L}; \mathbf{k}_{U,L}]$,

$$\mathbf{W} \in \mathbb{R}^{1 \times 2C} \quad with \quad (\mathbf{W})_c = \begin{cases} \theta n_S^c & if \quad c \leq C \\ \theta n_S^{c-C} & if \quad c > C \end{cases},$$

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_S & \mathbf{0}_{n_s \times C} \\ \mathbf{0}_{n_u \times C} & \mathbf{V}_U \end{bmatrix} \in \mathbb{R}^{(n_s + n_u) \times 2C} \quad with$$

$$(\mathbf{V}_S)_{ij} = \begin{cases} 1 & if \quad x_s^i \in class \quad j \\ 0 & otherwise \end{cases},$$

$$(\mathbf{V}_U)_{ij} = \begin{cases} 1 & if \quad x_u^i \quad is \quad predicted \quad as \quad class \quad j \\ 0 & otherwise \end{cases}.$$

After obtaining the transformation matrix, $\mathbf{A}$, and landmark weights, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, we can utilize the transformed source data and training target data by combining them with landmark weights to train the classification model, which is used to predict the pseudo-labels $\{\widetilde{y}_u^i\}_{i=1}^{n_u}$ of the test target data $\{\widehat{x}_u^i\}_{i=1}^{n_u}$.

3.3. Progressive Pseudo-Label Selection

In the initial stage of learning, the transformation matrix $\mathbf{A}$, and the landmark weights $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, cannot be fully optimized, which may degrade the performance of the classification model and further lead to low-quality pseudo-labels. For this reason, we designed a progressive strategy for pseudo-label selection that selects the corresponding pseudo-label subset in each iteration for the optimization of the objective function. First, we define $\{(\widehat{x}_u^i, \widetilde{y}_u^i, p(\widetilde{y}_u^i|\widehat{x}_u^i))\}_{i=1}^{n_u}$, where $p(\widetilde{y}_u^i|\widehat{x}_u^i)$ denotes the probability of predicting the label of $\widehat{x}_u^i$ as $\widetilde{y}_u^i$. In the kth iteration, we then select the top $n_u k/T$ high-probability modules to optimize $\mathbf{A}$, $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$. T is the total number of iterations. In order to avoid the situation that selective pseudo-labels belong to the same category, we separately select the top $n_u^c k/T$ high-probability modules for each category, c, where n_u^c denotes the number of modules predicted as category c in the test target data.

Algorithm 1 describes the detailed process of our approach. Before iteration, we first preprocess the source and target data (lines 3–4). Then, we project the target data to the metric subspace with m dimensions by using principal component analysis (line 5). The projection matrix $\mathbf{A}$ and the pseudo-labels of the test target data $\{\widetilde{y}_u^i\}_{i=1}^{n_u}$ are initialized (lines 6–7). For each iteration, we first select the pseudo-labels and their corresponding modules and then use these to update the transformation matrix $\mathbf{A}$ and landmark weights $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ sequentially (lines 9–11). Next, given the $\mathbf{A}$, $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$, the pseudo-labels of the test target modules can be updated (line 12). When the iteration is over, we regard the pseudo-labels of the test target modules as the final prediction results.

Algorithm 1 Pseudo code of LDASP

```
1:  Input: source data $S = \{x_s^i, y_s^i\}_{i=1}^{n_s}$, training target data $T_L = \{x_l^i, y_l^i\}_{i=1}^{n_l}$, test target
    data$T_U = \{x_u^i\}_{i=1}^{n_u}$; dimension of metric subspace $m$; $\theta$; iteration number $T$.
2:  Output: predicted labels $\{y_u^i\}_{i=1}^{n_u}$ of test target data $\{x_u^i\}_{i=1}^{n_u}$.
3:  Remove the duplicated modules and the modules with missing metric values for $S$ and
    $\{T_L, T_U\}$;
4:  Use z-score normalization to scale the data from $S$ and $\{T_L, T_U\}$;
5:  Project target data $\{x_l^i\}_{i=1}^{n_l}$ and $\{x_u^i\}_{i=1}^{n_u}$ into the metric subspace with $m$ dimensions;
6:  Initialize the transformation matrix $\mathbf{A}$ via Equation (5);
7:  According to Section 3.3, initialize the pseudo-labels of test target data $\{\widetilde{y}_u^i\}_{i=1}^{n_u}$;
8:  for $k = 1$ to $T$ do
9:      For each category $c$, select $n_u^c k/T$ pseudo-labels with the highest probability from
        $\{\widetilde{y}_u^i\}_{i=1}^{n_u}$ and their corresponding modules;
10:     Update the transformation matrix $\mathbf{A}$ according to Equation (12);
11:     Update landmark weights $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ according to Equation (14);
12:     Update the pseudo-labels of test target data $\{\widetilde{y}_u^i\}_{i=1}^{n_u}$ based on the description of
        Section 3.3;
13: end for
```

4. Experiment Setup

In this work, experiments will be conducted based on the following research questions to verify the effectiveness of our approach.

- RQ1: How effective is LDASP in HDP with mixed-project data?
- RQ2: What is the effect of each component of LDASP on improving HDP performance?

4.1. Dataset and Evaluation Indicator

For a comprehensive evaluation, we selected 27 projects from four widely used public datasets, including NASA [33], AEEEM [34], PROMISE [35], and JIRA [36]. Table 2 displays the dataset details, where the last column lists what object a module specifically refers to in the corresponding project. It can be seen that the four datasets contain different numbers of metrics, which accords with the HDP setting.

Table 2. Details of datasets.

Dataset	Project	# of Metrics	# of Total Modules	# of Defective Modules (%)	Prediction Granularity
NASA	CM1	37	327	42 (12.84%)	Function
	MW1	37	253	27 (10.67%)	
	PC1	37	705	61 (8.65%)	
	PC3	37	1077	134 (12.44%)	
	PC4	37	1287	177 (13.75%)	
AEEEM	EQ	61	324	129 (39.81%)	Class
	JDT	61	997	206 (20.66%)	
	LC	61	691	64 (9.26%)	
	ML	61	1862	245 (13.16%)	
	PDE	61	1497	209 (13.96%)	
PROMISE	ant-1.7	20	745	166 (22.28%)	Class
	camel-1.4	20	872	145 (16.6%)	
	ivy-2.0	20	352	40 (11.36%)	
	jedit-4.0	20	306	75 (24.51%)	
	log4j-1.0	20	135	34 (25.19%)	
	poi-2.0	20	314	37 (11.78%)	
	tomcat-6.0	20	858	77 (8.97%)	
	velocity-1.6	20	229	78 (34.06%)	
	xalan-2.4	20	723	110 (15.21%)	
	xerces-1.3	20	453	69 (15.23%)	
JIRA	activemq-5.0.0	65	1884	293 (15.55%)	File
	derby-10.5.1.1	65	2705	383 (14.16%)	
	groovy-1.6.0.beta1	65	821	70 (8.53%)	
	hbase-0.94.0	65	1059	218 (20.59%)	
	hive-0.9.0	65	1416	283 (19.99%)	
	jruby-1.1	65	731	87 (11.9%)	
	wicket-1.3.0.beta2	65	1763	130 (7.37%)	

In order to evaluate the overall performance of the competing methods, we employed the frequently used indicators, including the *F1-score* and *AUC*. The *F1-score* [8] is denoted as the harmonic means of *recall* (probability of detection) and *precision*. According to the confusion matrix in Table 3, the *F1-score* can be calculated as $2 \times \frac{recall \times precision}{recall + precision}$, where *recall* is defined as $tp/(tp + fn)$, and *precision* is defined as $tp/(tp + fp)$. The *AUC* is the area under the receiver operating characteristic curve, which is plotted in a two-dimensional space with pf as the x-coordinate and pd as the y-coordinate. The *AUC* is a well-known indicator for the comparison of different models [5,16,37–40] because it is unaffected by class imbalance and is independent of the prediction threshold. The higher the *AUC* is, the better the performance of the prediction model. When the *AUC* is 0.5, this means the performance of a random predictor [41]. Moreover, Lessmann et al. [38] and Ghotra et al. [37] suggested using the *AUC* for better cross-dataset comparability. Hence, we select the *AUC* as one of the indicators.

Table 3. Four kinds of defect prediction results.

	Actual Defective	Actual Non-Defective
Predict defective	*tp*	*fp*
Predict non-defective	*fn*	*tn*

4.2. Experimental Design

This section provides the detailed experimental design for each research question.

4.2.1. RQ1: How Effective Is LDASP in HDP with Mixed-Project Data?

For RQ1, we compared LDASP for three types of related methods (as shown below) to evaluate its performance.

WPDP method: This method uses labeled data to build the prediction model that is then employed to determine whether the remaining unlabeled modules are defective or non-defective. Because the training and test data come from the same project, considerable prediction results can be generally obtained when the labeled modules are sufficient.

Unsupervised method: Chen et al. [42] found that the mainstream unsupervised prediction methods perform better than HDP ones for traditional and effort-aware indicators through extensive experiments. Therefore, we also chose the unsupervised methods SC [5] and ManualDown [43], which show considerable performance, as baselines. SC is a spectral clustering-based unsupervised method and achieves a better prediction effect over supervised models. ManualDown is a simple and effective unsupervised method based on the number of code lines, and it performs better than most current CPDP methods through extensive experiments. Note that the threshold of ManualDown was set as 50%, according to the suggestion of the original paper [43].

HDP method: Three mixed-project HDP methods, i.e., CLSUP [11], sMDA [10], and DSKMDA [12], were selected to compare with our approach. CLSUP not only utilizes the labeled data within source and target projects during the metric transformation but also imposes the costs of misclassification for the class imbalance problem. sMDA and DSKMDA are different extension versions of the manifold discriminant alignment algorithm [44]. In addition, we also selected a conventional HDP method, i.e., an ensemble method based on aligned metric representation (EAMR) [23], to verify the effectiveness of our approach.

In order to evaluate the statistical significance of the performance difference between both methods, we employed the nonparametric Wilcoxon signed-rank test [45] at the confidence level of 95%, and we report the results of the Win/Tie/Lose of LDASP vs. each baseline from the previous studies [6,46–48]. Moreover, we used the nonparametric effect size to test Cliff's delta (δ with values in $[-1, 1]$) [49] and measure the level of difference between the performance of the competitors. Table 4 presents the mappings of the δ values to the corresponding effectiveness levels.

Table 4. Mapping Cliff's delta values to the effectiveness levels.

Cliff's Delta (δ)	Effectiveness Levels
$\delta < 0.147$	Negligible (N)
$0.147 \leq \delta < 0.33$	Small (S)
$0.33 \leq \delta < 0.474$	Medium (M)
$0.474 \leq \delta$	Large (L)

4.2.2. RQ2: What Is the Effect of Each Component of LDASP on Improving HDP Performance?

In order to investigate RQ2, we constructed a series of variants based on the proposed LDASP, as shown in Table 5. This table lists the compared methods and their corresponding explanations. For example, LDASP$_{noMarginal}$ denotes the variant of our approach without the term of the marginal distribution alignment in the objective function, i.e., Equation (6); DASP refers to the variant of our approach without considering the landmark weights for the source and target modules, which is equivalent to the setting of $\alpha = 1$ and $\beta = 1$. In the experiments related to RQ2, we compared LDASP with four variants for all evaluation indicators. Furthermore, the nonparametric Wilcoxon signed-rank test [45] at a confidence level of 95% and the nonparametric effect size test of Cliff's delta were conducted between LDASP and each variant.

Table 5. Competing methods in RQ2.

Method	Description
LDASP$_{noMarginal}$	Removing the term of marginal distribution alignment
LDASP$_{noConditional}$	Removing the term of conditional distribution alignment
LDASP$_{noLocality}$	Removing the term of class-wise locality preserving
DASP	Without considering the landmark weights, i.e., $\alpha = 1$ and $\beta = 1$
LDAP	Using all pseudo-labels without selection

4.3. Evaluation Protocol

In the experiments, we constructed heterogeneous prediction combinations based on all the projects and used the predictor trained using the source data to predict the target project. For each prediction combination (i.e., source⇒target), we chose one project from the 27 total projects as the target. Another project from the datasets that did not contain the target project was selected as the source. Supposing that the target project is selected from AEEEM, the source project should come from NASA, PROMISE, or JIRA. In this way, 530 prediction combinations can be constructed based on these 27 projects from four datasets. Considering that there exists a small amount of labeled data in the target project, we randomly split 10% of the target modules for the training target data, and the remaining 90% were regarded as the test target data, as in Refs. [10,11]. Each prediction combination was executed 20 times, and we ultimately report the average indicator results of the competing methods on each target project.

4.4. Parameter Setting

For the parameters of LDASP, we set the metric subspace dimension m, the number of iterations T, and θ as $\lfloor min(d_s, d_t)/2 \rfloor$, 5, and 0.5, respectively. For each competing method, we used the logistic regression (LR) classifier to conduct the predictions. This is mainly because LR has been applied in extensive SDP research and has exhibited better classification ability than other classifiers. The LR classifier is implemented by LIBLINEAR [50] and adopts the parameter setting "−s 0" (i.e., logistical regression) and "−b 1" (i.e., no bias term added) suggested in Ref. [51].

5. Experimental Result

This section reports and analyzes the experimental results for each research question and further summarizes the corresponding conclusions.

5.1. Results for RQ1

Tables 6 and 7 report the mean results of the competing methods for all the target projects, in which the best result for each project is marked in bold. The "Average" presents the overall mean results across the 27 target projects.

- Our approach obtains the best results for the *F1-score* and *AUC* for more than half (i.e., 15 out of 27 and 14 out of 27) of the target projects. Moreover, it also achieves the best overall performance for each indicator in terms of the "Average" results.
- When compared with the mixed-project HDP methods (i.e., CLSUP, sMDA, and DSKMDA), our approach separately improves the *F1-score* and *AUC* by 9.8–20.2% and 5.3–11.3% in terms of the overall mean results. This is mainly because the learned landmark weights eliminate the negative impact brought about by the noise modules to some extent; meanwhile, selective pseudo-labeling further promotes prediction performance.
- When compared with the conventional HDP method (i.e., EAMR), our approach makes improvements of 15.8% and 4.8% on the *F1-score* and *AUC*, respectively. As a conventional HDP method without using training target data, EAMR even shows better performance than the mixed-project HDP baselines (i.e., CLSUP, sMDA, and DSKMDA). However, an over-reliance on the label information of the source data also prevents its performance for further promotion.
- When compared with the unsupervised methods (i.e., SC and ManualDown), our approach improves the average *F1-score* and *AUC* by at least 10.6% and 11.6%, respectively. The possible reason is that the lack of labeled modules results in unsupervised methods being unable to obtain reliable discriminative information and, thus, inhibit the improvement in prediction performance.
- When compared with the WPDP method, our approach separately improves the average *F1-score* and *AUC* by 18.7% and 14.4%, which means that the effective utilization of labeled source data can increase the prediction effect on the target project.

Table 6. Comparison results for *F1-score*.

Target	WPDP	SC	Manual Down	EAMR	CLSUP	sMDA	DSKMDA	Ours
CM1	0.249	**0.335**	0.282	0.279	0.274	0.299	0.281	0.326
MW1	0.206	**0.303**	0.247	0.272	0.248	0.271	0.253	0.301
PC1	0.217	0.245	0.236	0.252	0.228	0.250	0.222	**0.268**
PC3	0.288	0.345	0.316	0.332	0.348	0.364	0.296	**0.380**
PC4	**0.410**	0.329	0.328	0.401	0.361	0.367	0.271	0.380
EQ	0.548	0.557	**0.619**	0.594	0.558	0.539	0.448	0.581
JDT	0.544	**0.591**	0.466	0.553	0.562	0.566	0.532	0.590
LC	0.273	0.337	0.217	0.288	0.314	0.321	0.243	**0.355**
ML	0.333	0.318	0.311	0.344	0.364	0.344	0.347	**0.392**
PDE	0.325	0.369	0.333	0.372	0.366	0.374	0.358	**0.385**
ant-1.7	0.606	0.620	0.431	0.532	0.613	0.617	0.583	**0.701**
camel-1.4	0.439	0.481	0.342	0.343	0.484	0.487	0.348	**0.522**
ivy-2.0	0.337	0.367	0.281	0.381	0.339	0.377	**0.445**	0.423
jedit-4.0	0.544	0.508	0.457	0.518	0.550	0.551	0.545	**0.628**
log4j-1.0	0.624	0.577	0.574	0.580	0.667	0.625	0.536	**0.712**
poi-2.0	0.673	**0.753**	0.573	0.299	0.693	0.721	0.278	0.751
tomcat	0.318	0.384	0.199	0.326	0.361	0.356	0.383	**0.420**
velocity-1.6	0.487	0.529	0.532	0.498	0.508	0.537	0.504	**0.549**
xalan-2.4	0.318	0.364	0.349	0.402	0.350	0.346	**0.461**	0.386
xerces-1.3	0.264	0.331	0.317	**0.406**	0.302	0.334	0.367	0.366
activemq-5.0.0	0.441	0.518	0.471	0.497	0.495	0.495	**0.593**	0.550
derby-10.5.1.1	0.482	0.553	0.506	0.434	0.501	0.502	0.182	**0.596**
groovy-1.6.0.beta1	0.260	0.113	0.249	0.306	0.301	0.308	0.301	**0.338**
hbase-0.94.0	0.269	0.303	0.271	**0.513**	0.303	0.322	0.406	0.337
hive-0.9.0	0.517	0.509	0.550	0.509	0.505	0.508	0.419	**0.556**
jruby-1.1	0.339	0.380	0.416	0.462	0.389	0.390	**0.596**	0.420
wicket-1.3.0.beta2	0.368	0.453	0.404	0.278	0.393	0.384	0.352	**0.468**
Average	0.396	0.425	0.381	0.406	0.421	0.428	0.391	**0.470**
Win/Tie/Lose	26/0/1	23/3/1	24/2/1	22/2/3	27/0/0	27/0/0	22/1/4	-
N/S/M/L ($\delta > 0$)	0/0/0/26	0/0/1/22	0/2/0/24	2/5/3/14	0/0/0/27	0/0/3/24	0/0/0/22	-
N/S/M/L ($\delta < 0$)	0/0/0/1	3/0/1/0	0/0/0/1	0/3/0/0	0/0/0/0	0/0/0/0	1/0/0/4	-

The best results for each project and "Average" are marked in bold.

Table 7. Comparison results for *AUC*.

Target	WPDP	SC	Manual Down	EAMR	CLSUP	sMDA	DSKMDA	Ours
CM1	0.590	0.642	**0.688**	0.601	0.620	0.655	0.624	0.687
MW1	0.579	0.679	0.729	0.656	0.638	0.682	0.609	**0.731**
PC1	0.651	0.634	0.761	0.705	0.672	0.735	0.657	**0.763**
PC3	0.642	0.672	0.726	0.702	0.733	0.762	0.700	**0.782**
PC4	**0.801**	0.635	0.706	0.757	0.721	0.750	0.632	0.776
EQ	0.626	0.630	0.680	**0.764**	0.722	0.689	0.644	0.743
JDT	0.778	0.761	0.780	0.789	0.803	0.806	0.761	**0.821**
LC	0.656	0.716	0.634	0.718	0.771	0.764	0.640	**0.801**
ML	0.671	0.627	0.690	0.687	0.737	0.698	0.695	**0.761**
PDE	0.645	0.671	0.722	0.726	0.725	0.737	0.713	**0.746**
ant-1.7	0.810	0.746	0.596	0.783	0.828	0.828	0.820	**0.891**
camel-1.4	0.689	0.689	0.580	0.653	0.746	0.748	0.664	**0.768**
ivy-2.0	0.694	0.678	0.648	0.790	0.736	0.765	0.795	**0.821**
jedit-4.0	0.726	0.630	0.623	0.744	0.740	0.744	0.739	**0.802**
log4j-1.0	0.679	0.618	0.568	**0.809**	0.769	0.739	0.751	0.804
poi-2.0	0.846	0.823	0.780	0.655	0.847	0.873	0.620	**0.890**
tomcat	0.692	0.706	0.482	0.786	0.778	0.770	0.789	**0.837**
velocity-1.6	0.725	0.713	**0.822**	0.683	0.753	0.792	0.649	0.797
xalan-2.4	0.596	0.613	0.645	0.740	0.640	0.635	**0.788**	0.668
xerces-1.3	0.632	0.701	**0.812**	0.731	0.701	0.760	0.646	0.792
activemq-5.0.0	0.662	0.679	0.757	**0.820**	0.729	0.726	0.814	0.771
derby-10.5.1.1	0.706	0.708	0.777	0.783	0.724	0.719	0.663	**0.809**
groovy-1.6.0.beta1	0.588	0.356	0.653	**0.760**	0.690	0.711	0.600	0.744
hbase-0.94.0	0.688	0.715	**0.797**	0.792	0.745	0.782	0.635	0.791
hive-0.9.0	0.662	0.577	0.649	**0.776**	0.637	0.633	0.604	0.706
jruby-1.1	0.652	0.668	0.793	**0.857**	0.719	0.733	0.849	0.759
wicket-1.3.0.beta2	0.639	0.695	0.689	**0.798**	0.683	0.687	0.743	0.723
Average	0.679	0.666	0.696	0.743	0.726	0.738	0.698	**0.777**
Win/Tie/Lose	26/0/1	27/0/0	20/3/4	17/7/3	27/0/0	26/1/0	23/2/2	-
N/S/M/L ($\delta > 0$)	0/0/1/25	0/0/0/27	3/0/0/20	3/3/1/12	0/0/0/27	1/1/4/21	0/1/1/22	-
N/S/M/L ($\delta < 0$)	0/0/0/1	0/0/0/0	0/1/0/3	4/0/2/2	0/0/0/0	0/0/0/0	0/1/0/2	-

The best results for each project and "Average" are marked in bold.

"Win/Tie/Lose" and "N/S/M/L" in Tables 6 and 7 exhibit the results of the nonparametric Wilcoxon signed-rank test and the nonparametric effect size test, respectively. From these tables, we can obtain the following observations:

- In terms of "Win/Tie/Lose", our approach obtains significant performance improvements for most projects when compared to the competing methods. This is especially true for EAMR, which had the best performance among the baselines; our approach wins the comparisons against the other approaches, with significant advantages in 22 out of 27 and 17 out of 27 projects, respectively, according to the *F1-score* and *AUC*.
- In terms of "N/S/M/L" ($\delta > 0$), our approach achieves non-negligible performance improvements for more than 20 projects when compared to the competing methods, except for the best baseline EAMR. Even compared to EAMR, our approach could still obtain non-negligible performance improvements for more than half of the cases (i.e., 22 out of 27 and 16 out of 27 projects).

Answer to RQ1: Our approach shows the best overall performance among all the competitors in terms of the *F1-score* and *AUC*. Moreover, its performance improvements are statistically significant in general.

5.2. Results for RQ2

Figure 2 displays the prediction results of each method for the *F1-score* and *AUC*, in which the horizontal line and diamond in a box denote the median and mean results, respectively. This figure shows that our approach generally obtains the best median and mean results on both indicators, whereas the degree of improvement varies, which indicates the positive effect of each component of LDASP on defect prediction. As shown in this figure, the improvements in our approach compared to LDASP$_{noMargin}$, LDASP$_{noCondition}$, and DASP are visible, but the improvements over LDASP$_{noLocal}$ and LDAP are slight.

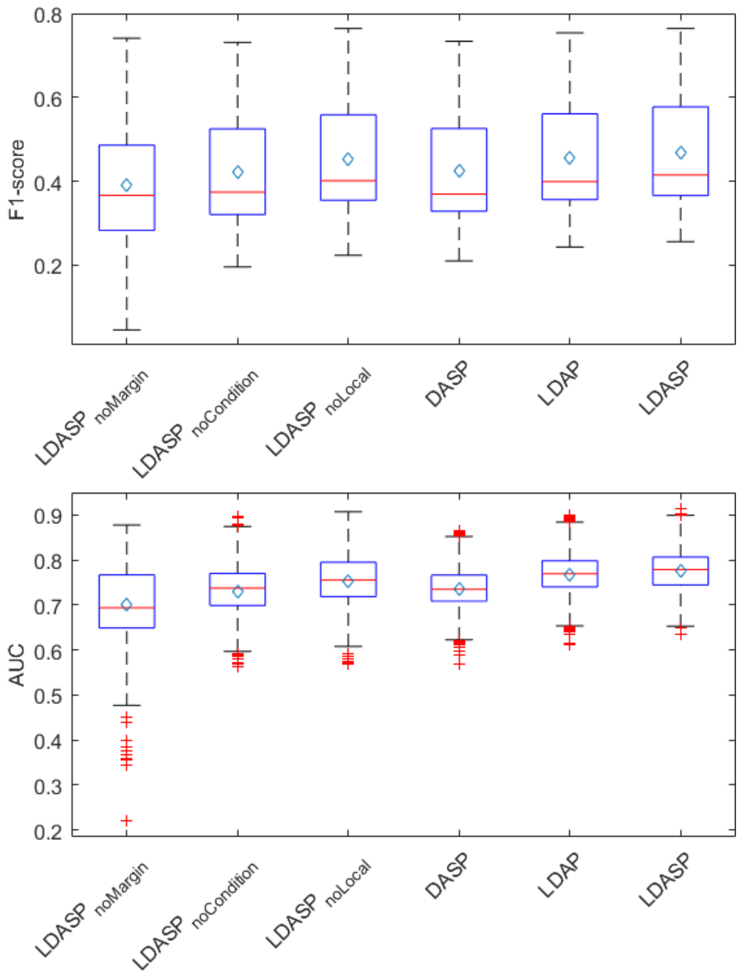

Figure 2. Comparison results of *F1-score* and *AUC*.

To further illustrate the performance difference, Table 8 provides the statistical significance test and effect size test results of LDASP versus other competitors. Based on the observation of Table 8, we can conclude the following:

- When compared with LDASP$_{noMargin}$, our approach wins the comparisons of both indicators in almost all projects (i.e., 21 out of 27 and 25 out of 27 projects). In other words, LDASP$_{noMargin}$ shows significant performance degradation without the marginal distribution alignment term. Moreover, in terms of N/S/M/L ($\delta > 0$), our approach obtains non-negligible performance improvements in the *F1-score* and *AUC* on 22 out of 27 projects, respectively. This is because the marginal distribution alignment can reduce the discrepancy between the source and target data. Therefore, a failure to consider marginal distribution alignment may result in the insufficient reduction in the data distribution difference between both projects, which causes the inadaptation of the predictor to target data.
- When compared with LDASP$_{noCondition}$, our approach maintains consistent performance improvements in general. In terms of Win/Tie/Lose, our approach wins the comparisons in all projects on the *F1-score* and *AUC*. Furthermore, the N/S/M/L ($\delta > 0$) rows show that our approach obtains large performance improvements in both indicators on almost all projects, except for the medium improvement in the AUC on one project. Thus, removing the conditional distribution alignment term weakens the prediction effect significantly. The possible reason is that conditional

distribution alignment can alleviate the distribution difference in the modules from the same category in different projects, thereby enhancing the classification ability of predictors in target data.

- When compared with LDASP$_{noLocal}$, our approach wins the comparisons of both indicators in 14 out of 27 and 19 out of 27 projects, respectively. Furthermore, regarding N/S/M/L ($\delta > 0$), our approach achieves the non-negligible performance improvements in the *F1-score* and *AUC* in more than half of the projects. This illustrates that although preserving the class-wise locality brings about limited improvements, it still shows a significantly positive effect on defect predictions in general.
- When compared with DASP, our approach improves both indicators significantly in most projects in terms of Win/Tie/Lose and N/S/M/L ($\delta > 0$). Especially for the *AUC*, our approach wins the competition against DASP and achieves non-negligible performance improvements in all projects. Based on the above observation, we can learn that the learned landmark weights are conducive to the improvement in predictions by highlighting the relevant modules and mitigating the negative transfer of noise modules.
- When compared with LDAP, our approach wins the comparisons while achieving non-negligible performance improvements for more than half of the cases, which demonstrates that the proposed progressive pseudo-labeling strategy effectively reduces the introduction of unreliable pseudo-labels that may hinder the learning process.

Table 8. Comparison results against LDASP.

Test	Indicator	LDASP$_{noMargin}$	LDASP$_{noCondion}$	LDASP$_{noLocal}$	DASP	LDAP
Win/Tie/Lose	*F1-score*	22/1/4	27/0/0	14/6/7	20/6/1	19/6/2
	AUC	22/2/3	27/0/0	19/5/3	27/0/0	21/6/0
N/S/M/L ($\delta > 0$)	*F1-score*	0/0/0/22	0/0/0/27	3/5/3/6	4/10/6/4	4/14/3/2
	AUC	0/1/0/22	0/0/1/26	4/10/2/6	0/3/8/16	7/19/0/0

Answer to RQ2: According to the comparison results for the *F1-score* and *AUC*, each component of our approach can promote HDP performance effectively.

6. Discussion

In this section, we further discuss the performance of LDASP. Specifically, Section 6.1 explains its effectiveness. Section 6.2 investigates the impact of different percentages of the training target data on the prediction effect. Section 6.3 analyzes the sensitivity of the parameters, and Section 6.4 provides the threats to the validity of LDASP.

6.1. Effectiveness of LDASP

In order to illustrate the effectiveness of the proposed approach further, we visualized the data distributions processed by LDASP. The t-SNE (t-distributed stochastic neighbor embedding) [52] was employed to reduce the dimensions of the source and target data in the metric subspace for easy display. By taking ant-1.7⇒jurby-1.1 as an example, Figures 3 and 4 show the visualization results when the iteration time equals 1 and 5. In Figure 3, LDASP initializes the transformation matrix $\mathbf{A}$ without considering landmark weights and pseudo-labels. It can be seen that defective modules do not gather together very well, which leads to poor separability in the metric space. After multiple iterations, the modules of different categories have good separability, as shown in Figure 4. We can observe that defective and non-defective modules gather in the upper and lower parts of the coordinate system, respectively, and are easy to separate linearly. In summary, our proposed approach can match the data distributions of the source and target projects effectively while enhancing the classification ability of the transformed data.

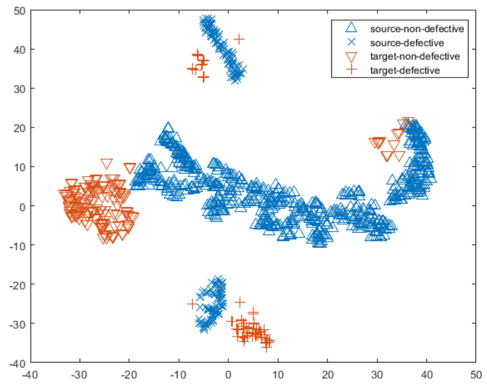

Figure 3. Distributions of source and target data processed by LDASP when the iteration number equals 1.

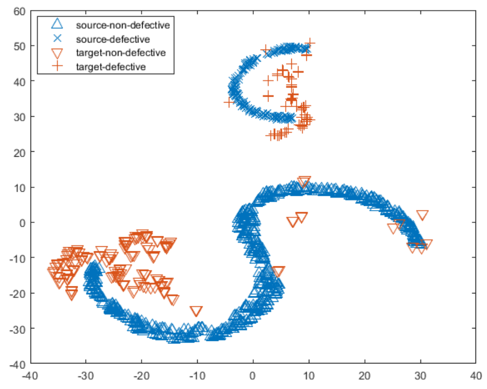

Figure 4. Distributions of source and target data processed by LDASP when the iteration number equals 5.

Similar to Table 1, we also provide Table 9 to illustrate whether LDASP could reduce the negative impact of noise modules effectively. Given a source project and a target project, when the WPDP method performs better than the mixed-project HDP method, we think this is an invalid cross-project prediction combination because the source data play the role of degrading prediction performance. Therefore, the fewer invalid cross-project prediction combinations there are, the more effective the competing method is in dealing with the negative impact of the noise modules. From this table, we can find that among the competing mixed-project HDP methods, our approach has the fewest invalid cross-project prediction combinations on both indicators. The percentages of the invalid cross-project prediction combinations of our approach (i.e., 4.5% and 4.3%) are far below those of the other three methods, which demonstrates that LDASP improves the ability to eliminate the negative impact of noise modules effectively.

Table 9. Statistics for the comparison between WPDP and mixed-project HDP method.

Situation	**Number of Prediction Combinations (% of Invalid Cross-Project Prediction Combinations)**	
	F1-Score	***AUC***
WPDP performs better than CLSUP	114 (21.5%)	107 (20.2%)
WPDP performs better than sMDA	104 (19.6%)	85 (16.0%)
WPDP performs better than DSKMDA	228 (43.0%)	201 (37.9%)
WPDP performs better than ours	24 (4.5%)	23 (4.3%)

6.2. Effect of Different Percentages of Training Target Data

In our experiments, the percentage of training target data is fixed at 10%, as in Refs. [10,11]. In order to understand the performance of LDASP comprehensively, we explore its prediction effect under different percentages of training target data in this section. The proportion ranges from 10% to 90%, with a step size of 10%. We selected WPDP as the baseline that trains the predictor with training target data and used it to predict the test target data directly. Figure 5 presents the average indicator results of both methods under different percentages. We can see that with the growing percentage, the results of WPDP increase gradually, whereas those of LDASP remain basically stable. On the one hand, the prediction performance of WPDP improves with the increase in the training target data. On the other hand, the change in proportion has a limited impact on the performance of LDASP because the progressively selected pseudo-labels have provided relatively reliable and supervised information for it. Overall, LDASP performs better than WPDP for both indicators under most of the percentages. When the percentage is greater than about 60%, WPDP achieves a comparable prediction performance to LDASP. Hence, using WPDP directly can also obtain considerable prediction results as long as the training data within the target project are sufficient.

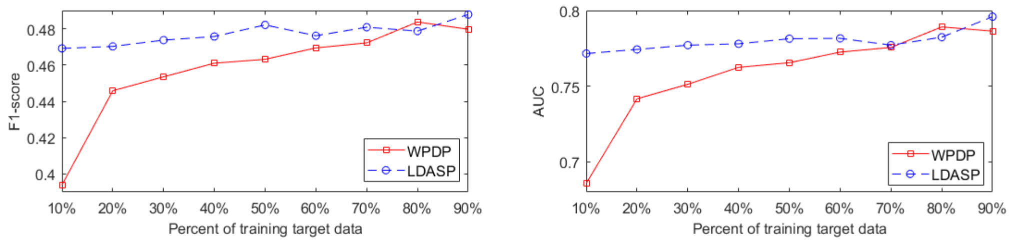

Figure 5. Comparison results between LDASP and WPDP under different percentages of training target data.

6.3. Sensitivity of Parameters

In order to assess the parameter sensitivity of LDASP, we investigate the impact of different hyperparameters, T and θ, on prediction performance. Figures 6 and 7 exhibit the average indicator results of LDASP under different values of T and θ, respectively.

(1) T is the iteration time that affects the convergence degree of the objective function. Normally, the more iterations there are, the easier it is for the objective function to converge. As shown in Figure 6, the prediction performance of LDASP generally keeps rising with the increase in T. When T is greater than 7, its indicator results tend to be stable. Moreover, the change in prediction performance is actually slight in terms of the difference between the indicator results of various T. Based on this, we can believe that the objective function has been fully optimized, although the setting of $T = 5$ is not optimal.

(2) θ is used to restrict the proportion of landmarks that participate in distribution adaptation. LDASP$_{\theta=1}$ treats all modules as having the same weight and corresponds to the optimization problem in Equation (5). As shown in Figure 7, LDASP$_{\theta=0}$ and LDASP$_{\theta=1}$ perform worse than LDASP with other θ values for both indicators. Furthermore, when θ is greater than 0, there is a significant improvement in the performance of LDASP. This indicates the positive impact of introducing unlabeled modules and their pseudo-labels on improving performance. In summary, our parameter setting of $\theta = 0.5$ is reasonable, although it cannot help LDASP reach the best prediction performance.

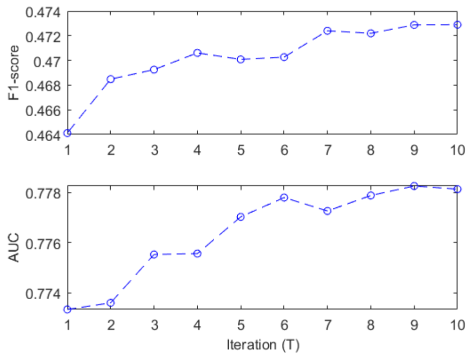

Figure 6. Results of LDASP for the *F1-scores* and *AUC* under different iteration numbers, *T*.

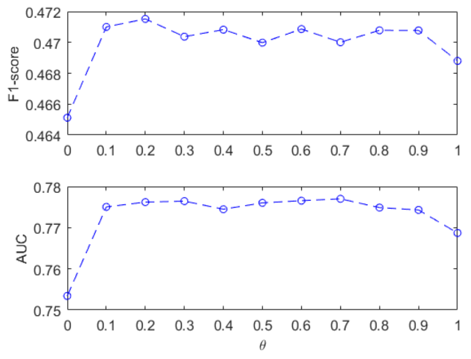

Figure 7. Results of LDASP for the *F1-scores* and *AUC* under different θ.

6.4. Threats to Validity

Construct validity relates to the evaluation methods used in this work. In order to simulate the case where there is a small number of labeled data in the target project, we randomly split 10% of the target modules to make the training target data, and the remaining 90% were regarded as the test target data. This operation has been adopted from Refs. [10,11] to evaluate the performance of HDP with mixed-project data, but the changing percentages may lead to different prediction results. Moreover, we chose the *F1-score* and *AUC*, which are widely applied in HDP studies, to measure the overall prediction performance of competing methods. Other indicators (e.g., *G-measure*, *MCC*, and *Popt*) will be considered in our future work to improve the evaluation from different aspects.

The potential threat to internal validity may come from the replication of baselines. We have carefully implemented the compared method, which is not open-source, according to the description of the original paper. However, differences between our implementation and the original code may still exist, thus leading to biases in the prediction results.

External validity refers to the generalizability of our experimental results. In the experiments, we chose 27 projects from four datasets to validate the effectiveness of the proposed approach. A total of 530 one-to-one prediction combinations were constructed to be tested. Therefore, the conclusions of this work may not be generalized to other datasets.

7. Conclusions

Combining "within" and "cross-project" data (i.e., mixed-project data) is an effective way of improving prediction performance when there are limited historical data in the target project. In this paper, we propose a novel approach based on landmark-based domain adaptation and selective pseudo-labeling for HDP using mixed-project data. We first construct the objective function that considers marginal and conditional distribution matching and class-wise locality constraints (for projected source data) simultaneously to alleviate the heterogeneity between both projects. In order to reduce the negative impact of noise modules, we introduce the landmark weights to be learned for labeled source and unlabeled target modules. Furthermore, we also design a pseudo-label selection strategy to progressively select the pseudo-labels with high confidence and the corresponding modules for the learning process. Extensive comparisons are conducted for 27 projects from four datasets, including NASA, AEEEM, PROMISE, and JIRA. Two widely used indicators (i.e., the *F1-score* and *AUC*) are employed to evaluate the overall performance of each method. The experimental results indicate that LDASP outperforms the compared methods, and this verifies the effectiveness of each component of it.

In future work, we plan to collect more experimental data from open-source projects to test our approach. Moreover, we will extend our approach to the context of effort-aware evaluation, which considers both the prediction performance and the amount of work required to check modules.

Author Contributions: Conceptualization, Y.C. and H.C.; methodology, Y.C.; software, H.C.; validation, Y.C. and H.C.; formal analysis, H.C.; investigation, Y.C.; data curation, H.C.; writing—original draft preparation, Y.C.; writing—review and editing, H.C.; visualization, Y.C. and H.C.; project administration, H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The research data are available on request from the corresponding author. The data are not publicly available due to their close relation to our future work.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Menzies, T.; Greenwald, J.; Frank, A. Data mining static code attributes to learn defect predictors. *IEEE Trans. Softw. Eng.* **2007**, *33*, 2–13. [CrossRef]
2. Hassan, A.E. Predicting faults using the complexity of code changes. In Proceedings of the 31st International Conference on Software Engineering, ICSE, Vancouver, BC, Canada, 16–24 May 2009; pp. 78–88.
3. Tosun, A.; Bener, A.; Turhan, B.; Menzies, T. Practical considerations in deploying statistical methods for defect prediction: A case study within the Turkish telecommunications industry. *Inf. Softw. Technol.* **2010**, *52*, 1242–1257. [CrossRef]
4. Turhan, B.; Menzies, T.; Bener, A.B.; Stefano, J.S.D. On the relative value of cross-company and within-company data for defect prediction. *Empir. Softw. Eng.* **2009**, *14*, 540–578. [CrossRef]
5. Zhang, F.; Zheng, Q.; Zou, Y.; Hassan, A.E. Cross-project defect prediction using a connectivity-based unsupervised classifier. In Proceedings of the 38th International Conference on Software Engineering, ICSE, Austin, TX, USA, 14–22 May 2016; pp. 309–320.
6. Xia, X.; Lo, D.; Pan, S.J.; Nagappan, N.; Xinyu, W. HYDRA: Massively compositional model for cross-project defect prediction. *IEEE Trans. Softw. Eng.* **2016**, *42*, 977–998. [CrossRef]
7. Nam, J.; Kim, S. CLAMI: Defect prediction on unlabeled datasets. In Proceedings of the 30th International Conference on Automated Software Engineering, ASE, Lincoln, NE, USA, 9–13 November 2015; pp. 1–12.
8. Jing, X.Y.; Wu, F.; Dong, X.; Qi, F.; Xu, B. Heterogeneous cross-company defect prediction by unified metric representation and CCA-based transfer learning. In Proceedings of the 10th Joint Meeting on Foundations of Software Engineering, FSE, Bergamo, Italy, 30 August–4 September 2015; pp. 496–507.
9. Turhan, B.; Misirli, A.T.; Bener, A. Empirical evaluation of the effects of mixed project data on learning defect predictors. *Inf. Softw. Technol.* **2013**, *55*, 1101–1118. [CrossRef]
10. Li, Z.; Jing, X.Y.; Zhu, X.; Zhang, H.; Xu, B.; Ying, S. On the multiple sources and privacy preservation issues for heterogeneous defect prediction. *IEEE Trans. Softw. Eng.* **2019**, *45*, 391–411. [CrossRef]
11. Li, Z.; Jing, X.Y.; Zhu, X. Heterogeneous fault prediction with cost-sensitive domain adaptation. *Softw. Test. Verif. Reliab.* **2018**, *28*, e1658. [CrossRef]
12. Niu, J.; Li, Z.; Chen, H.; Dong, X.; Jing, X. Data sampling and kernel manifold discriminant alignment for mixed-project heterogeneous defect prediction. *Softw. Qual. J.* **2022**, *30*, 917–951. [CrossRef]
13. Li, Z.; Niu, J.; Jing, X.; Yu, W.; Qi, C. Cross-Project Defect Prediction via Landmark Selection-Based Kernelized Discriminant Subspace Alignment. *IEEE Trans. Reliab.* **2021**, *70*, 996–1013. [CrossRef]
14. Van Engelen, J.E.; Hoos, H.H. A survey on semi-supervised learning. *Mach. Learn.* **2020**, *109*, 373–440. [CrossRef]
15. Lee, D.H. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In Proceedings of the ICML 2013 Workshop: Challenges in Representation Learning (WREPL), Atlanta, GA, USA, 16–21 June 2013.
16. Nam, J.; Kim, S. Heterogeneous defect prediction. In Proceedings of the 10th Joint Meeting on Foundations of Software Engineering, FSE, Bergamo, Italy, 30 August–4 September 2015; pp. 508–519.

17. Li, Z.; Jing, X.Y.; Zhu, X.; Zhang, H. Heterogeneous defect prediction through multiple kernel learning and ensemble learning. In Proceedings of the IEEE International Conference on Software Maintenance and Evolution, ICSME, Shanghai, China, 17–22 September 2017; pp. 91–102.
18. Li, Z.; Jing, X.Y.; Wu, F.; Zhu, X.; Xu, B.; Ying, S. Cost-sensitive transfer kernel canonical correlation analysis for heterogeneous defect prediction. *Autom. Softw. Eng.* **2018**, *25*, 201–245. [CrossRef]
19. Tong, H.; Liu, B.; Wang, S. Kernel Spectral Embedding Transfer Ensemble for Heterogeneous Defect Prediction. *IEEE Trans. Softw. Eng.* **2021**, *47*, 1886–1906. [CrossRef]
20. Li, Z.; Jing, X.; Zhu, X.; Zhang, H.; Xu, B.; Ying, S. Heterogeneous defect prediction with two-stage ensemble learning. *Autom. Softw. Eng.* **2019**, *26*, 599–651. [CrossRef]
21. Yu, Q.; Jiang, S.; Zhang, Y. A feature matching and transfer approach for cross-company defect prediction. *J. Syst. Softw.* **2017**, *132*, 366–378. [CrossRef]
22. Xu, Z.; Ye, S.; Zhang, T.; Xia, Z.; Pang, S.; Wang1, Y.; Tang, Y. MVSE: Effort-Aware Heterogeneous Defect Prediction via Multiple-View Spectral Embedding. In Proceedings of the International Conference on Software Quality, Reliability and Security, QRS, Sofia, Bulgari, 22–26 July 2019; pp. 10–17.
23. Chen, H.; Jing, X.; Zhou, Y.; Li, B.; Xu, B. Aligned metric representation based balanced multiset ensemble learning for heterogeneous defect prediction. *Inf. Softw. Technol.* **2022**, *147*, 106892. [CrossRef]
24. Wu, J.; Wu, Y.; Niu, N.; Zhou, M. MHCPDP: Multi-source heterogeneous cross-project defect prediction via multi-source transfer learning and autoencoder. *Softw. Qual. J.* **2021**, *29*, 405–430. [CrossRef]
25. Zhu, K.; Ying, S.; Ding, W.; Zhang, N.; Zhu, D. IVKMP: A robust data-driven heterogeneous defect model based on deep representation optimization learning. *Inf. Sci.* **2022**, *583*, 332–363. [CrossRef]
26. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [CrossRef]
27. Long, M.; Wang, J.; Ding, G.; Sun, J.; Yu, P.S. Transfer Feature Learning with Joint Distribution Adaptation. In Proceedings of the IEEE International Conference on Computer Vision, ICCV, Sydney, Australia, 1–8 December 2013; pp. 2200–2207.
28. Long, M.; Wang, J.; Ding, G.; Sun, J.; Yu, P.S. Transfer joint matching for unsupervised domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1410–1417.
29. Tsai, Y.H.H.; Yeh, Y.R.; Wang, Y.C.F. Heterogeneous domain adaptation with label and structure consistency. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Shanghai, China, 20–25 March 2016; pp. 2842–2846.
30. Zhang, W.; Deng, L.; Zhang, L.; Wu, D. A Survey on Negative Transfer. *IEEE/CAA J. Autom. Sin.* **2023**, *10*, 305–329. [CrossRef]
31. Aljundi, R.; Emonet, R.; Muselet, D.; Sebban, M. Landmarks-based kernelized subspace alignment for unsupervised domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Boston, MA, USA, 7–12 June 2015; pp. 56–63.
32. Tsai, Y.H.; Yeh, Y.; Wang, Y.F. Learning cross-domain landmarks for heterogeneous domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Las Vegas, NV, USA, 27–30 June 2016; IEEE Computer Society: Washington, DC, USA, 2016; pp. 5081–5090.
33. Shepperd, M.; Song, Q.; Sun, Z.; Mair, C. Data quality: Some comments on the NASA software defect datasets. *IEEE Trans. Softw. Eng.* **2013**, *39*, 1208–1215. [CrossRef]
34. D'Ambros, M.; Lanza, M.; Robbes, R. Evaluating defect prediction approaches: A benchmark and an extensive comparison. *Empir. Softw. Eng.* **2012**, *17*, 531–577. [CrossRef]
35. Marian, J.; Lech, M. Towards identifying software project clusters with regard to defect prediction. In Proceedings of the 6th International Conference on Predictive Models in Software Engineering, PROMISE, Timisoara, Romania, 12–13 September 2010; pp. 1–10.
36. Yatish, S.; Jiarpakdee, J.; Thongtanunam, P.; Tantithamthavorn, C. Mining software defects: Should we consider affected releases? In Proceedings of the 41st International Conference on Software Engineering, ICSE, Montreal, QC, Canada, 25–31 May 2019; pp. 654–665.
37. Ghotra, B.; McIntosh, S.; E. Hassan, A. Revisiting the Impact of Classification Techniques on the Performance of Defect Prediction Models. In Proceedings of the 37th International Conference on Software Engineering, ICSE, Florence, Italy, 16–24 May 2015; pp. 789–800.
38. Lessmann, S.; Baesens, B.; Mues, C.; Pietsch, S. Benchmarking classification models for software defect prediction: A proposed framework and novel findings. *IEEE Trans. Softw. Eng.* **2008**, *34*, 485–496. [CrossRef]
39. Ryu, D.; Choi, O.; Baik, J. Value-cognitive boosting with a support vector machine for cross-project defect prediction. *Empir. Softw. Eng.* **2016**, *21*, 43–71. [CrossRef]
40. Tantithamthavorn, C.; McIntosh, S.; E. Hassan, A.; Matsumoto, K. Automated parameter optimization of classification techniques for defect prediction models. In Proceedings of the 38th International Conference on Software Engineering, ICSE, Austin, TX, USA, 14–22 May 2016; pp. 321–332.
41. Rahman, F.; Posnett, D.; Devanbu, P.T. Recalling the "imprecision" of cross-project defect prediction. In Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering, ESEC/FSE, Cary, NC, USA, 11–16 November 2012; pp. 1–11.
42. Chen, X.; Mu, Y.; Liu, K.; Cui, Z.; Ni, C. Revisiting heterogeneous defect prediction methods: How far are we? *Inf. Softw. Technol.* **2021**, *130*, 106441. [CrossRef]

43. Zhou, Y.; Yang, Y.; Lu, H.; Chen, L.; Li, Y.; Zhao, Y.; Qian, J.; Xu, B. How far we have progressed in the journey? An examination of cross-project defect prediction. *ACM Trans. Softw. Eng. Methodol.* **2018**, *27*, 1–51. [CrossRef]
44. Wang, C.; Mahadevan, S. Heterogeneous domain adaptation using manifold alignment. In Proceedings of the 22nd International Joint Conference on Artificial Intelligence, IJCAI, Barcelona, Spain, 16–22 July 2011; pp. 1541–1546.
45. Hollander, M.; Wolfe, D.A. *Nonparametric Statistical Methods*; Wiley: Hoboken, NJ, USA, 1999; p. 74.
46. He, Z.; Shu, F.; Yang, Y.; Li, M.; Wang, Q. An investigation on the feasibility of cross-project defect prediction. *Autom. Softw. Eng.* **2012**, *19*, 167–199. [CrossRef]
47. Ma, Y.; Luo, G.; Zeng, X.; Aiguo, C. Transfer learning for cross-company software defect prediction. *Inf. Softw. Technol.* **2012**, *54*, 248–256. [CrossRef]
48. Chen, L.; Fang, B.; Shang, Z.; Tang, Y. Negative samples reduction in cross-company software defects prediction. *Inf. Softw. Technol.* **2015**, *62*, 67–77. [CrossRef]
49. Fu, W.; Menzies, T. Revisiting unsupervised learning for defect prediction. In Proceedings of the 11th Joint Meeting on Foundations of Software Engineering, ESEC/FSE, Paderborn, Germany, 4–8 September 2017; pp. 72–83.
50. Fan, R.E.; Chang, K.W.; Hsieh, C.J.; Wang, X.R.; Lin, C.J. LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.* **2008**, *9*, 1871–1874.
51. Nam, J.; Pan, S.J.; Kim, S. Transfer defect learning. In Proceedings of the 35th International Conference on Software Engineering, ICSE, San Francisco, CA, USA, 18–26 May 2013; pp. 382–391.
52. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.