文章编号:1671-4598(2024)08-0123-06 DOI:10.16526/j.cnki.11-4762/tp.2024.08.018 中图分类号:TP273.5.1 文献标识码:A

基于动态分布适应网络的跨项目缺陷预测

章树卿[1,2], 周世健[1], 毛敬恩[1,2], 樊 鑫[1,2]
(1. 南昌航空大学 软件学院, 南昌 330038; 2. 南昌航空大学 软件测评中心, 南昌 330038)

摘要: 在软件缺陷预测中, 跨项目缺陷预测是基于源项目的标记数据来训练模型, 并预测当前正在开发的目标项目的缺陷; 然而, 两个不同项目数据之间的分布差异往往限制了跨项目缺陷预测模型的能力; 由于源域和目标域的数据通常来自不同的分布, 因此现有方法主要集中于适应跨域边缘或条件分布; 在实际应用中, 现有方法无法定量评估边缘分布和条件分布的重要性, 这将导致传输性能不理想; 论文提出了一种基于动态分布适应网络的跨项目缺陷预测方法来解决分布差异问题, 它利用迁移学习能够定量评估每个分布的相对重要性; 论文对来自 3 个公共数据集的 24 个项目进行了实验, 以验证所提出的方法; 结果表明, 平均而言在 *AUC* 和 F_1 分数上分别比所有基线方法高出至少 1.3% 和 5.7%; 这表明所提出的方法具有良好的性能特点。

关键词: 迁移学习; 跨项目; 动态分布; 深度学习; 定量评估

Cross-Project Defect Prediction Based on Dynamic Distributed Adaptive Networks

ZHANG Shuqing[1,2], ZHOU Shijian[1], MAO Jingen[1,2], FAN Xin[1,2]
(1. School of Software, Nanchang Hangkong University, Nanchang 330038, China;
2. School Testing and Evaluation Center, Nanchang Hangkong University, Nanchang 330038, China)

Abstract: In software defect prediction, cross-project defect prediction is based on training models by using the labeled data from the original project, and predicts the defects of the current development target project. However, the data distribution differences between two different projects often limit the ability of cross-project defect prediction models. As the data from the source domain and target domain usually come from different distributions, existing methods mainly adapt to cross-domain edges or conditional distributions. In practical application, existing methods are unable to quantitatively evaluate the importance of marginal and conditional distributions, which leads to unsatisfactory transmission performance. This paper proposes a cross-project defect prediction method based on a dynamic distribution adaptation network to address the distribution difference, the transmission learning is used to quantitatively evaluate the relative importance of each distribution. the proposed method is verified by the experiments on 24 projects from 3 public datasets. The results show that, on average, the proposed method is superior to all baseline methods, the proposed method reaches the *AUC* and F_1 scores by at least 1.3% and 5.7%, respectively. This indicates that the proposed method has good performance characteristics.

Keywords: transfer learning; cross-project; dynamic distribution; deep learning; quantitative evaluation

0 引言

随着互联网时代的到来, 现如今软件已经遍布世界各地, 无论是工业化还是生活化, 人们都或多或少依赖相关软件产品, 然而, 任何软件都不是完美的, 其源代码不可避免地存在缺陷, 一旦该产品发生故障, 轻则影响用户的体验, 重则产生大规模的经济损失。因此, 如果能在软件开发的早期阶段准确地发现缺陷, 将大大节省开发人员的时间与精力, 并提高软件的质量。软件缺陷预测 (SDP, software defect prediction)[1-3]被用来缓解这个问题。

SDP 中的跨项目缺陷预测方法 (CPDP, cross project defect prediction) 利用已经标记的数据集, 将外部项目的标记数据应用于具有有限或无标记训练数据的项目中。通过这种方式, 研究人员可以利用已有的数据来训练模型, 并预测新项目中的潜在缺陷。文献 [4] 提出了一个两阶段跨项目预测模型解决了 TCA+ 的不稳定问题。文献 [5] 引入了基于协作过滤的源项目选择 (CFPS) 技术进行源项目选择, 并验证了 CPDP 源项目选择的可行性、重要性和有效性。文献 [6] 提出了一种过滤器方法 (BurakMHD 过滤器) 来选择 CPDP 中的相关训练数据, 提高了 CPDP 的性能。文献 [7] 使用核双支持向量机 (KTSVM) 来实现域适应 (DA) 来匹配不同项目的训练数据分布, 从而解决了训练数据分布的差异。文献 [8] 提出了一种流形嵌入分布自适应 (MDA) 技术来缩小各种特征子空间中的分布差距。文献 [9] 提出了一种新颖的 HCPDP 方法来弥合目标数据集和源数据集之间的差距。

传统的迁移学习一般假设边缘分布和条件分布权重相

收稿日期:2024-03-05; 修回日期:2024-03-20。
作者简介:章树卿(1997-),男,硕士。
引用格式:章树卿,周世健,毛敬恩,等. 基于动态分布适应网络的跨项目缺陷预测[J]. 计算机测量与控制,2024,32(8):123-128,137.

等，但无法评估这两个分布的相对重要性，以往的方法主要关注边缘分布差异而忽略了条件分布差异，这将导致性能不尽如人意。文献［10］通过迁移学习使源项目和目标项目的特征分布彼此相似，并利用粒子群优化算法综合考虑多个源项目来预测目标项目。文献［11］通过合并多个源项目来增加源数据集的数量以减少负传输的影响并升级分类器的性能。文献［12］提出了一种迁移倾向算法（TSboostDF），该算法考虑了 CPDP 的知识转移和类不平衡。文献［13］引入迁移学习技术从特征工程的角度减少项目之间的差异。文献［14］提出了一种基于特征选择和迁移学习的度量补偿软件缺陷预测方法。文献［15］提出了一个结合特征迁移和集成学习的 CPDP 模型，包括特征迁移和分类两个阶段。文献［16］提出了多源迁移学习（3SW-MSTL）的三阶段加权框架来解决条件分布差异和多源数据问题。目前对于迁移学习的研究大多是集中在类不平衡以及源数据和目标数据之间的分布差异等方面，文献［17］提出的 BDA 方法首次给出了边缘分布和条件分布的定量估计。然而，其并未解决平衡因子 μ 的精确计算问题，对源数据与目标数据集之间分布差异的优化研究将是论文的主要内容。文献［18］提出了一种动态自适应分布的迁移学习（DDA）方法，该方法在 BDA 上做了改进，使其能定量 μ，达到更好的效果。

为了最小化两个项目之间数据的边缘和条件分布差异[19-20]，论文提出了一种基于动态分布适应网络的跨项目缺陷预测方法（CPDP-DDAN，cross-project defect prediction based on dynamic distributed adaptive networks）。DDAN 是一个针对领域适应中边缘分布和条件分布的相对重要性的定量评估框架。具体来说，DDAN 能够通过计算域之间的散度[21]来动态学习分布权重。然后，可以评估两种分布的相对重要性，进而可以用来学习更多可转移的特征表示。该方法能够不断迭代优化动态重要性学习和特征学习，最终达到提升模型的效果。CPDP-DDAN 可以通过交叉验证（Cross-Validation）确定自适应因子 μ 最优的取值来解决数据分布差异问题。结合 DDAN 迁移学习方法，可以更好地调整边缘分布和条件分布来解决类不平衡问题。论文对 3 个基准数据集的 24 个项目数据进行了足够的实验，以评估 DDAN 的有效性。实验结果表明 DDAN 在 *AUC* 上平均提升了 1.3%，在 F_1 上平均提升了 5.7%。

1 研究方法

1.1 动态分布适应网络

动态分布适应的目的是定量评估在域适应中对齐边缘（P）和条件（Q）分布的重要性。这里引入了一个自适应因子来动态调整这两个分布的重要性。形式上，动态分布对齐 $\overline{D}_f$ 定义为：

$$\overline{D}_f(\Omega_s,\Omega_t)=(1-\mu)\,D_f(P_s,P_t)+\mu\sum_{(c=1)}^{C}D_f^{(c)}(Q_s,Q_t)\quad(1)$$

其中：$\mu\in[0,1]$ 为自适应因子，$c\in\{1,\cdots,C\}$ 为类别指示符。$D_f(P_s,P_t)$ 为边缘分布对齐，$D_f^{(c)}(Q_s,Q_t)$ 为 c 类的条件分布对齐。

当 $\mu\to 0$ 时，表示源域和目标域之间的分布距离较大。因此，边缘分布对齐显得更为重要。当 $\mu\to 1$ 时，意味着域之间的特征分布相对较小，使得每个类的分布占主导地位。因此，条件分布对齐就变得更为重要。当 $\mu\to 0.5$ 时，两种分布均按照现有方法进行同等处理[22]。

本文使用最大平均差异（MMD）[21]来计算域之间的分布散度。边缘分布距离和条件分布距离可以分别计算为：

$$D_f(P_s,P_t)=\|[f(z_s)]-E[f(z_t)]\|_{H_K}^2\quad(2)$$

$$D_f^{(c)}(Q_s,Q_t)=\|E[f_s^{(c)}]-E[f(z_t^{(c)})]\|_{H_K}^2\quad(3)$$

那么，DDA 可以表示为：

$$\overline{D}_f(\Omega_s,\Omega_t)=(1-\mu)\,\|E[f(z_s)-E[f(z_t)]\|^2\,H_K+\mu\sum_{c=1}^{C}\|E[f(z_s^{(c)})]-E[f(z_t^{(c)})]\|_{H_K}^2\quad(4)$$

通过利用域的全局和局部结构来计算 μ（即 $\hat{\mu}$）。这里采用 A-distance[21]作为基本测量。A-distance 被定义为构建线性分类器来区分两个域（即二元分类）的误差。形式上，可以将 $\in(h)$ 表示区分两个域 Ω_s 和 Ω_t 的线性分类器 h 的误差。那么，A-distance 可以定义为：

$$d_A(\Omega_s,\Omega_t)=2[1-2\in(h)]\quad(5)$$

接着，使用上面的方程直接计算边缘 A-distance，记为 d_M。对于条件分布之间的 A-distance，使用 d_c 表示第 c 类的 A-distance。它可以计算为 $d_c=d_A(D_s^{(c)},D_t^{(c)})$，其中 $D_s^{(c)}$ 和 $D_t^{(c)}$ 分别表示 Ω_s 和 Ω_t 中 c 类的样本。请注意，d_M 表示边缘差异，而 $sum_{c=1}^{c}d_c$ 表示所有类别的条件差异。本文的假设是域分歧是由边缘分布和条件分布引起的。因此，$d_M+\sum_{c=1}^{C}d_c$ 可以代表整个散度。最终，μ 可以估计为：

$$\hat{\mu}=1-\frac{d_M}{d_M+\sum_{c=1}^{C}d_c}\quad(6)$$

源域中标记样本的数量通常远大于目标域中的标记样本数量。因此，为了解决这种不平衡的分类问题，本文对目标域进行上采样，以使样本大小几乎相同。我们还注意到这个上采样过程是随机的，因此重复这个步骤几次以获得平均 μ 值。

采用 DDA，DDAN 的学习目标可以表示为：

$$f=\min_{\Theta}\sum_{i=1}^{n}J(f(x_i^s),y_i^s)+\lambda\overline{D}_f(\Omega_s,\Omega_t)+\rho R_f(\Omega_s,\Omega_t)\quad(7)$$

其中：$J(\cdot,\cdot)$ 是交叉熵损失函数，$\Theta=\{\boldsymbol{w},b\}$ 包含神经网络的权重和偏差参数。由于 DDAN 基于深度神经网络，因此不使用整个域数据，而是按照小批量随机梯度下降（SGD）训练过程使用批量数据。因此，动态分布适应仅在批次之间计算，而不是在整个域之间计算。

1.2 CPDP-DDAN 模型构建

图 1 展示了本文提出的 CPDP-DDAN 方法的整体框架。

加载数据：首先，对源数据集和目标数据集进行读取，将其中的样本作为深度神经网络的输入。其次，接着利用深度模型 ResNet 网络从输入中提取高级特征。

模型训练：在训练过程中，使用动态分布对齐（DDA）

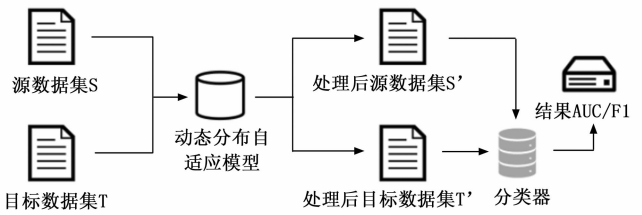

图 1　CPDP-DDAN 方法框架图

来对齐源域和目标域特征，通过动态调整 μ 的值，以及在损失函数中对 MMD 和分类损失进行加权组合，可以实现在训练过程中逐渐减小源域和目标域之间的差异。

模型测试：将经过模型处理后的训练集和测试集进行划分，对模型在测试集上进行测试，计算测试集上的损失值和输出。

分类：最后，采用线性核的支持向量机（SVM）对新的数据进行分类，根据测试集上的预测结果用评价指标来评估效果。

2　实验设置

2.1　数据集

在该小节中，本文使用 AEEEM、NASA 和 PROMISE[23-24]3 个公开数据集，其中一共包含 24 个项目。表 1 列出了所有这些存储库的详细摘要。AEEEM、NASA 和 PROMISE 中的每个项目分别取 61、21、20 个指标。

表 1　24 个数据集

数据集	项目	样本	特征	缺陷率/%
AEEEM	EQ	324	61	39.81
	JDT	997	61	20.66
	Lucene	691	61	9.26
	Mylyn	1862	61	13.16
	PDE	1497	61	13.96
NASA	CM1	327	37	12.84
	JM1	7782	21	21.49
	KC1	1183	21	26.54
	MW1	253	37	10.67
	PC1	705	37	8.65
	PC2	745	36	2.15
	PC3	1077	37	12.44
PROMISE	Ant-1.3	125	20	16
	Camel-1.6	965	20	19.48
	Ivy-2.0	352	20	11.36
	Jedit-4.1	312	20	25.32
	Log4j-1.2	205	20	92.2
	Poi-2.0	314	20	11.78
	Prop-6	660	20	10
	Synapse-1.2	269	20	31.97
	Tomcat	858	20	8.97
	Velocity-1.4	196	20	75
	Xalan-2.4	723	20	15.21
	Xerces-1.2	440	20	16.14

2.2　实验环境和测量评估

本次实验的硬件配置为一台带有 Intel i5 酷睿处理器、16 G 内存和 GTX3060 显卡的台式机，操作系统为 Windows10，使用到的开发语言为 Python 及其相关机器学习与深度学习模块 Scikit-learn 和 PyTorch，开发工具为 PyCharm。

本研究使用两种目前广泛使用的指标来评估模型的性能，即 AUC 和 F_1。这两种常见指标在许多 CPDP 方法中被使用。

表 2 为混淆矩阵，是一种用于总结分类模型的表格。真正类率（TPR）是指预测正确的正类在实际全部正类中的频率（相当于召回率），假正类率（FPR）是指预测错误的正类在实际全部负类中的频率，公式如下：

$$TPR = R = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN} \tag{8}$$

ROC 表示不同分类阈值下的 TPR 和 FPR 构成的曲线。AUC 即 ROC 曲线下的面积，面积越大，则分类模型效果越优。

精准率是指模型正确预测正类的频率。召回率是指在所有实际的正类标签中，模型正确识别正类的频率，公式如下：

$$Precise = P = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = R = \frac{TP}{TP + FN} \tag{10}$$

F_1 表示精确率和召回率的调和平均值，调和均值 F_1 可以整体上反映算法性能：

$$F_1 = \frac{2 * P * R}{P + R} = \frac{2TP}{2TP + FP + FN} \tag{11}$$

AUC 和 F_1 的取值范围为 0～1，AUC 和 F_1 的值越大，表明预测性能越好。

表 2　混淆矩阵

项目		真实值	
		Positive	Negeative
预测值	Positive	TP	FP
	Negeative	FN	TN

2.3　比较方法

为了评估 CPDP-DDAN 的有效性，本文将其与一系列代表性的 CPDP 方法进行比较，包括 TCA+[20]、CTKC-CA[25]、ALTRA[26]。TCA+[20]使用了最先进的迁移学习方法 TCA 来使源项目和目标项目中的特征分布相似，并提出了 TCA+，它为 TCA 选择了合适的标准化方法。CTKC-CA[25]提出了一种基于特征的迁移学习方法，可以利用缺陷类和无缺陷类的不同误分类成本来缓解类不平衡问题。ALTRA[26]使用主动学习来选择有代表性的未标记模块，并在第一阶段依靠专家来标记这些模块，然后使用 TrAdaBoost 分别确定源项目和目标项目中标记模块的权重，并使用加权变量支持向量机来构建下一次迭代的模型，这样可以缓

解数据分布差异问题。

2.4 评估设置

本文使用了来自 3 个公开数据集的 24 个项目作为基准数据集。为了评估 CPDP 模型的性能，这里选择相同数据集中的一个项目作为源数据，再选择该数据集中的另一个项目作为目标数据，共获得了 194 种组合。对于每种组合，本文从源数据集中随机采样数据，根据帕累托法则将 80% 的实例作为训练数据进行训练。为了评估 CPDP 方法的性能，我们重复了上述步骤 30 次，并计算了相同数据集下所有组合的平均性能。

2.5 统计显著性检验和效应量检验

Scott-Knott ESD 测试是一种统计检验方法，其被广泛应用于分析某些方法是否优于其他方法，并且可以生成这些方法的全局排名，Scott-Knott ESD 测试使用层次聚类分析将不同的方法分为显著不同的组，其中效应大小不可忽略，它保证了同组方法之间没有统计学显著的性能差异，不同组方法之间存在统计学显著的性能差异。Scott-Knott ESD 检验可以得出处理平均值的排序，同时确保：组内差异可以忽略不计；组间的差异不可忽略。

3 实验结果与分析

3.1 与基线方法的比较

表 3、表 4 表示了 CPDP-DDAN 在 *AUC*、F_1 中和其他基线方法的比较结果。最佳值以粗体显示。从表中可以注意到：CPDP-DDAN 有至少一半的项目优于其他基线方法，在 Promise 数据集上，AUC 得到了最大值 0.655，可能是因为 Ivy-2.0 项目的数据分布非常适合 CPDP-DDAN 方法，并且相比于其他基线，性能提高了 1.3% ～ 7.1%；在 NASA 数据集上，F_1 得到了最大值 0.501，可能是因为 MW1 项目的缺陷数量相对来说较少，和其他基线相比，性能提高了 5.7%～ 12.6%。

图 2、图 3 以箱型图形式分别显示了不同 CPDP 方法在 *AUC* 和 F_1 方面的效果。顶部的横线表示非异常范围内的最大值、方框内的横线表示中位数，叉表示平均值，底部的横线表示非异常范围内的最小值。在图中，最大值、中位数、平均数和最小值的值越高意味着两项指标的表现越好，这里舍弃了异常值，从图中可以看出：对于图 2，和所有基线方法相比，CPDP-DDAN 的平均数、中位数和最小值均为最优，尽管最大值略低于 TCA+，但是整体看来效果依旧最好；对于图 3，和所有基线方法相比，CPDP-DDAN 的最大值、平均数和中位数均为最优，尽管最小值略低于 ALTRA，但是整体看来效果依旧最好。

图 4、图 5 以组合图形式分别显示了不同 CPDP 方法在 *AUC* 和 F_1 方面的效果。在图中，3 种基线方法均以柱形图表示，每个项目的左边柱形表示 TCA+方法，中间柱形表示 CTKCCA 方法，右边柱形表示 ALTRA 方法，CPDP-DDAN 以折线图表示。从图中可以看出在大多数项目里，经过 CPDP-DDAN 处理后得到的结果最好。

表 3 CPDP-DDAN 和基线的 *AUC* 比较

目标项目	TCA+	CTKCCA	ALTRA	CPDP-DDAN
EQ	0.632	0.628	0.636	0.568
JDT	0.565	0.642	0.601	0.606
Lucene	0.633	0.634	0.530	0.614
Mylyn	0.596	0.59	0.539	0.599
PDE	0.652	0.613	0.594	0.642
CM1	0.582	0.629	0.592	0.631
JM1	0.551	0.605	0.525	0.598
KC1	0.528	0.539	0.573	0.532
MW1	0.575	0.626	0.597	0.654
PC1	0.624	0.598	0.582	0.603
PC2	0.617	0.572	0.574	0.654
PC3	0.537	0.585	0.566	0.598
Ant-1.3	0.627	0.604	0.604	0.622
Camel-1.6	0.56	0.523	0.531	0.573
Ivy-2.0	0.627	0.529	0.617	0.655
Jedit-4.1	0.634	0.606	0.492	0.649
Log4j-1.2	0.539	0.596	0.508	0.548
Poi-2.0	0.559	0.629	0.593	0.638
Prop-6	0.562	0.587	0.58	0.62
Synapse-1.2	0.587	0.625	0.562	0.596
Tomcat	0.619	0.606	0.537	0.566
Velocity-1.4	0.541	0.599	0.618	0.551
Xalan-2.4	0.657	0.589	0.481	0.627
Xerces-1.2	0.611	0.629	0.584	0.635
平均值	0.592	0.599	0.567	0.607
提升/%	2.5	1.3	7.1	/

表 4 CPDP-DDAN 和基线的 F_1 比较

目标项目	TCA+	CTKCCA	ALTRA	CPDP-DDAN
EQ	0.417	0.332	0.357	0.436
JDT	0.403	0.386	0.393	0.439
Lucene	0.385	0.373	0.341	0.324
Mylyn	0.342	0.36	0.372	0.359
PDE	0.346	0.364	0.368	0.317
CM1	0.431	0.354	0.397	0.464
JM1	0.453	0.342	0.401	0.47
KC1	0.458	0.39	0.406	0.428
MW1	0.43	0.404	0.412	0.501
PC1	0.422	0.346	0.397	0.482
PC2	0.394	0.367	0.422	0.453
PC3	0.424	0.378	0.403	0.465
Ant-1.3	0.372	0.324	0.348	0.326
Camel-1.6	0.418	0.366	0.338	0.466
Ivy-2.0	0.405	0.411	0.332	0.343
Jedit-4.1	0.433	0.392	0.442	0.486
Log4j-1.2	0.396	0.406	0.403	0.44
Poi-2.0	0.265	0.294	0.317	0.315
Prop-6	0.331	0.323	0.371	0.336
Synapse-1.2	0.326	0.321	0.318	0.368
Tomcat	0.439	0.398	0.33	0.448
Velocity-1.4	0.385	0.384	0.341	0.42
Xalan-2.4	0.328	0.396	0.332	0.361
Xerces-1.2	0.343	0.345	0.358	0.415
平均值	0.389	0.365	0.371	0.411
提升/%	5.7	12.6	10.8	/

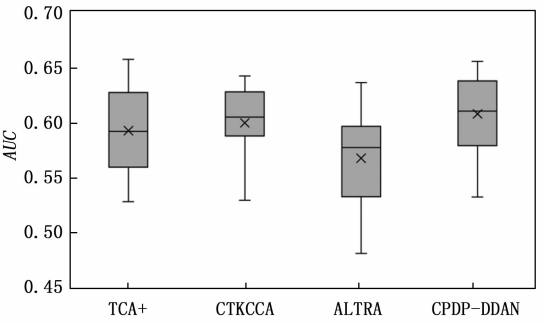

图 2　以箱形图表示不同 CPDP 方法的 *AUC* 结果

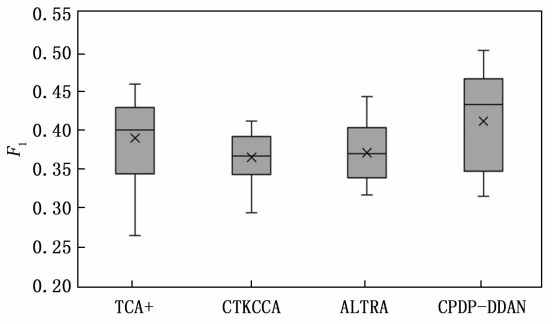

图 3　以箱形图表示不同 CPDP 方法的 F_1 结果

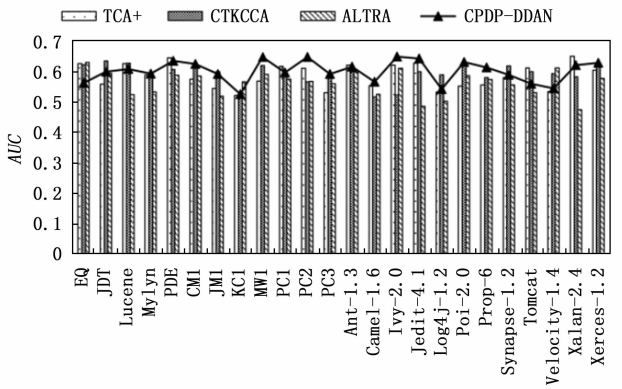

图 4　以组合图表示不同 CPDP 方法的 *AUC* 结果

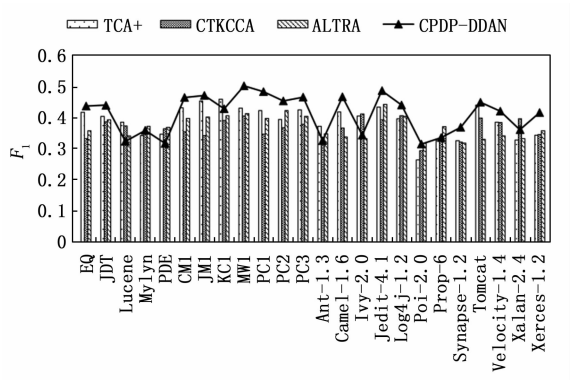

图 5　以组合图表示不同 CPDP 方法的 *AUC* 结果

本实验证明了与跨项目软件缺陷预测的其余 3 种方法相比，本文提出的 CPDP-DDAN 方法拥有更好的缺陷预测性能。

3.2　统计检验研究

图 6、图 7 分别用 *AUC* 和 F_1 表示了 CPDP-DDAN 与各种基线方法在 ScottKnott ESD 上的测试结果。图中用箱形图表示结果，以中间的横线表示平均值，由于第四节实验结果部分已经展示了各种基线方法的平均数，对于结果部分本文不做详细描述。从图 6 中可以看到，尽管本文方法的最大值低于 TCA+，最小值低于 CTKCCA，但是 CPDP-DDAN 方法依旧排名第一，CTKCCA 和 TCA+排第二，ALTRA 排在最后。从图 7 中可以看到，尽管本文方法的最小值略低于 ALTRA，但是 CPDP-DDAN 方法依旧排名第一，TCA+排第二，ALTRA 和 CTKCCA 排在最后。图 6、图 7 均表明 CPDP-DDAN 方法优于最先进的 CPDP 方法。ScottKnott ESD 测试还证实，CPDP-DDAN 方法在 *AUC* 和 F_1 方面始终名列前茅，这表明性能差异具有统计显著性，且效应大小不可忽略。

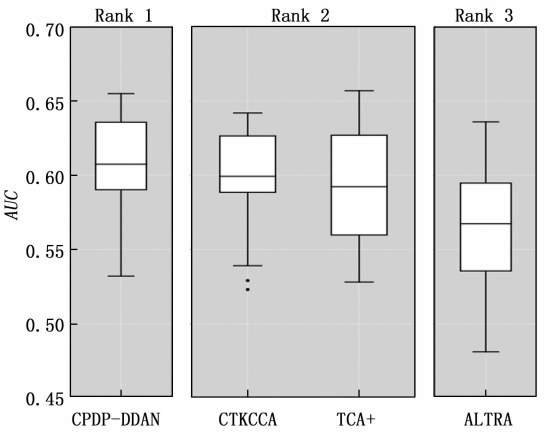

图 6　以 AUC 表示的 ScottKnott ESD 排名

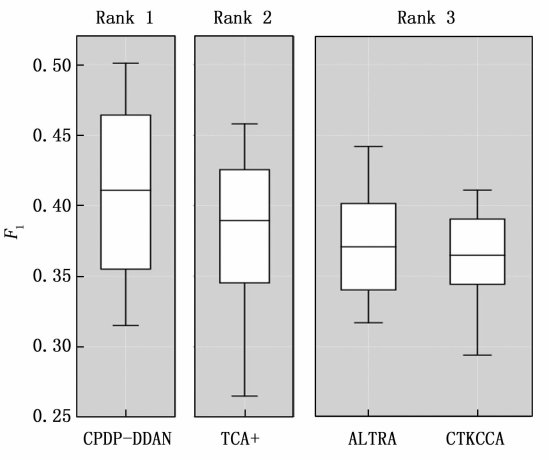

图 7　以 F_1 表示的 ScottKnott ESD 排名

4 结束语

本研究提出了一种基于动态分布自适应网络（CPDP-DDAN）的跨项目缺陷预测方法，旨在解决源项目和目标项目之间存在的数据分布差异问题。CPDP-DDAN 通过采用深度学习网络模型，将获取到的源数据集和目标数据集中的特征提取出来，接着在 DDAN 模型中利用最大均值差异 MMD 和动态平衡因子 μ 来构建损失函数，不断地对模型参数进行更新，最后将处理过后的数据通过 SVM 分类器进行评估预测。为了评估 CPDP-DDAN 方法的性能，本文在 3 个公开数据集一共 24 个项目上并使用 *AUC* 和 F_1 作为评估指标对实验进行了评估。实验结果表明，CPDP-DDAN 在预测有效性方面优于其他基线方法，并且具有不错的效果。这意味着 CPDP-DDAN 方法可以更准确地预测跨项目的缺陷情况。CPDP-DDAN 方法具有简单、有效等特点，这使得它可以帮助研究者在不同项目之间进行可靠的缺陷预测。CPDP-DDAN 是一种可行的新型 CPDP 方法，能够减少源项目和目标项目之间的数据分布差异，从而提高跨项目缺陷预测的性能。

对于未来的工作，我们计划：1）使用更多的数据集与评估指标来比较本文方法；2）不同的分类器可能会带来不同的效果，考虑其他分类器对本文方法的影响；3）将 DDAN 方法扩展到异构迁移学习（HDP，Heterogeneous defect prediction）领域，并将其应用于更复杂的迁移学习情况。HDP 的第一个挑战是如何克服源项目和目标项目中包含的不同指标集。第二个挑战是源项目和目标项目的数据分布差异，这在 HDP 和 CPDP 中同时存在，因此，我们希望 DDAN 能够在 HDP 上产生更好的效果。

参考文献：

[1] LIMSETTHO N, BENNIN K E, KEUNG J W, et al. Cross project defect prediction using class distribution estimation and oversampling [J]. Information and Software Technology, 2018, 100: 87 - 102.

[2] WANG S, LIU T, NAM J, et al. Deep semantic feature learning for software defect prediction [J]. IEEE Transactions on Software Engineering, 2018, 46 (12): 1267 - 1293.

[3] NI C, XIA X, LO D, et al. Revisiting supervised and unsupervised methods for effort-aware cross-project defect prediction [J]. IEEE Transactions on Software Engineering, 2020, 48 (3): 786 - 802.

[4] LIU C, YANG D, XIA X, et al. A two-phase transfer learning model for cross-project defect prediction [J]. Information and Software Technology, 2019, 107: 125 - 136.

[5] SUN Z, LI J, SUN H, et al. CFPS: Collaborative filtering based source projects selection for cross-project defect prediction [J]. Applied Soft Computing, 2021, 99: 106940.

[6] BHAT N A, FAROOQ S U. An improved method for training data selection for cross-project defect prediction [J]. Arabian Journal for Science and Engineering, 2022: 1 - 16.

[7] JIN C. Cross-project software defect prediction based on domain adaptation learning and optimization [J]. Expert Systems with Applications, 2021, 171: 114637.

[8] SUN Y, JING X Y, WU F, et al. Manifold embedded distribution adaptation for cross-project defect prediction [J]. IET Software, 2020, 14 (7): 825 - 838.

[9] WU J, WU Y, NIU N, et al. MHCPDP: Multi-source heterogeneous cross-project defect prediction via multi-source transfer learning and autoencoder [J]. Software Quality Journal, 2021, 29 (2): 405 - 430.

[10] CHEN J, HU K, YANG Y, et al. Collective transfer learning for defect prediction [J]. Neurocomputing, 2020, 416: 103 - 116.

[11] WU J, WU Y, NIU N, et al. MHCPDP: multi-source heterogeneous cross-project defect prediction via multi-source transfer learning and autoencoder [J]. Software Quality Journal, 2021, 29 (2): 405 - 430.

[12] TANG S, HUANG S, ZHENG C, et al. A novel cross-project software defect prediction algorithm based on transfer learning [J]. Tsinghua Science and Technology, 2021, 27 (1): 41 - 57.

[13] LEI T, XUE J, WANG Y, et al. WCM-WTrA: A cross-project defect prediction method based on feature selection and distance-weight transfer learning [J]. Chinese Journal of Electronics, 2022, 31 (2): 354 - 366.

[14] CHEN J, WANG X, CAI S, et al. A software defect prediction method with metric compensation based on feature selection and transfer learning [J]. Frontiers of Information Technology & Electronic Engineering, 2022, 23 (5): 715 - 731.

[15] ZENG F, LIN W, XING Y, et al. A cross-project defect prediction model using feature transfer and ensemble learning [J]. Tehnicki Vjesnik, 2022, 29 (4): 1089 - 1099.

[16] BAI J, JIA J, CAPRETZ L F. A three-stage transfer learning framework for multi-source cross-project software defect prediction [J]. Information and Software Technology, 2022, 150: 106985.

[17] XU Z, PANG S, ZHANG T, et al. Cross project defect prediction via balanced distribution adaptation based transfer learning [J]. Journal of Computer Science and Technology, 2019, 34 (5): 1039 - 1062.

[18] WANG J, CHEN Y, FENG W, et al. Transfer learning with dynamic distribution adaptation [J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2020, 11 (1): 1 - 25.

[19] MA Y, LUO G, ZENG X, et al. Transfer learning for cross-company software defect prediction [J]. Information and Software Technology, 2012, 54 (3): 248 - 256.

（下转第 137 页）