ISSN 1673-9418 CODEN JKYTA8
Journal of Frontiers of Computer Science and Technology
1673-9418/2016/10(01)-0043-13
doi: 10.3778/j.issn.1673-9418.1505089

E-mail: fcst@vip.163.com
http://www.ceaj.org
Tel: +86-10-89056056

基于实例迁移的跨项目软件缺陷预测*

毛发贵,李碧雯,沈备军+
上海交通大学 软件学院,上海 200240

Cross-Project Software Defect Prediction Based on Instance Transfer*

MAO Fagui, LI Biwen, SHEN Beijun+
School of Software, Shanghai Jiao Tong University, Shanghai 200240, China
+ Corresponding author: E-mail: bjshen@sjtu.edu.cn

MAO Fagui, LI Biwen, SHEN Beijun. Cross-project software defect prediction based on instance transfer. Journal of Frontiers of Computer Science and Technology, 2016, 10(1): 43-55.

Abstract: Cross-project defect prediction is considered as an effective means for solving the data shortage early in the project. Unfortunately, the performance of cross-project defect prediction is generally poor largely because of project variation. Focusing on this issue, this paper proposes a cross-project defect prediction approach based on instance transfer. The approach uses transfer learning and boosting technology to extract and transfer the training dataset high-related with target dataset from other projects, and builds a stronger combined classification model. The experimental results on PROMISE datasets show that, the proposed approach is superior to single-source single-target boosting methods with higher precision and recall; and in early phase with short data, it can achieve similar or better prediction results than intra-project approach with rich data.
Key words: cross-project defect prediction; transfer learning; instance-based transfer; boosting

摘 要:跨项目软件缺陷预测是解决项目初期缺陷预测缺乏数据集的有效途径,但是项目间的差异性降低了预测准确率。针对这一问题,研究提出了基于实例迁移的跨项目缺陷预测方法。该方法采用迁移学习和自适应增强技术,从其他项目数据集中提取并迁移转化出与目标数据集关联性高的训练数据集,训练出更有效的预测模型。使用PROMISE数据集进行了对比实验,结果表明所提出的新方法有效避免了单源单目标缺陷预

* The National Natural Science Foundation of China under Grant No. 61472242 (国家自然科学基金); the National Basic Research Program of China under Grant No. 2015CB352203 (国家重点基础研究发展计划(973计划)).
Received 2015-05, Accepted 2015-07.
CNKI网络优先出版:2015-08-11, http://www.cnki.net/kcms/detail/11.5602.TP.20150811.1513.001.html

测两极分化问题,获得了更高的预测准确率和查全率;在目标项目数据集不足的情况下,能达到甚至超过数据集充足时项目内缺陷预测的预测效果。

关键词:跨项目缺陷预测;迁移学习;基于实例的迁移;自适应增强

文献标志码:A **中图分类号:**TP311.5

1 引言

自20世纪70年代以来,软件缺陷预测技术一直是研究人员和软件开发者非常关注的研究课题之一[1]。一个好的软件缺陷预测技术能提前发现与锁定软件缺陷,从而缩减开发周期,提高软件质量。缺陷发现得越早,其修复的成本就越低,因此在软件开发上游活动中应用软件缺陷预测技术能获得更大的价值。但是由于数据集的缺乏,使得项目初期的缺陷预测遇到冷启动的问题,无法构建一个有效的预测模型。一种可能的解决方法是使用其他项目数据建立预测模型,预测目标项目的软件缺陷,即跨项目软件缺陷预测[2]。

和项目内缺陷预测相比,跨项目缺陷预测具有以下3个优势[2]:

(1)项目内缺陷预测依靠于软件项目前期活动的数据,需要花费大量时间在数据处理与特征提取上。而跨项目缺陷预测可以直接使用已提取的各类其他项目特征数据。

(2)一些项目在初期没有足够的项目数据,无法建立缺陷预测模型。而跨项目缺陷预测技术不受项目限制,存在大量的可用其他项目数据集。

(3)软件技术更新快,大项目由于周期长,其数据存在过时隐患。而跨项目缺陷预测,可以选择性挑选最新的软件仓库数据进行处理,利用潜在相关联系保证跨项目缺陷预测的效果。

然而,跨项目缺陷预测的最终输出结果较于项目内部的缺陷预测,精确度较低。其主要原因是,不同软件项目的预测特征与数据集存在差异性,若利用大量不相关数据进行预测,会影响最终结果的准确性。

本文从项目差异性问题出发,将迁移学习技术引入跨项目缺陷预测,对自适应增强算法进行多源改进,提出了基于实例迁移的跨项目缺陷预测方法。它通过自适应迭代和加权,从其他项目数据集中提取并迁移转化出与目标数据集关联性高的训练数据集,训练出更有效的预测模型。

本文主要贡献包括:

(1)跨项目的实例迁移技术。以目标项目数据样本和其他项目数据集为基础,通过基于实例的迁移学习,进行跨项目缺陷预测建模。实验表明,在目标项目数据集不足的情况下,实例迁移技术能使跨项目缺陷预测达到甚至超过项目内缺陷预测的效果。

(2)MergeTrAdaBoost模型训练算法。利用聚类和数据筛选算法,从大量训练数据中提取与目标数据集相关性高的数据子集,再采用自适应增强技术,训练出有效的预测模型。

(3)MultiTrAdaBoost模型训练算法。基于迭代适应,多次训练自适应单源单目标模型;基于训练结果错误率,降低相关性低的数据实例权重以及错误率高的单源单目标模型权重,形成多源最优训练模型。

实验结果表明MergeTrAdaBoost和MultiTrAdaBoost适应于跨项目缺陷预测多训练集的情况,获得了更高的预测准确率和查全率。

本文组织结构如下:第2章综述和分析了跨项目软件缺陷预测的相关工作;第3章提出了基于实例迁移的跨项目缺陷预测方法;第4章对新方法进行了实验及对比分析;第5章进行总结。

2 相关工作

近年来研究者们从数据集处理、项目相似性分组、自适应预测模型等角度研究跨项目软件缺陷预测方法和技术,取得了一定的成果。

2.1 数据预处理

研究者通过转换数据分布与筛选相关数据等方式,降低项目数据差异性。

Briand等人[3]最早探索跨项目缺陷预测,通过线性回归和MARS(multivariate adaptive regression splines)

Table 1 Analysis of related works

表1 相关技术分析

技术	优点	缺点
数据预处理	基于数据过滤，可以增加关联数据比例，有效降低误报率；	过滤后的数据与目标项目之间仍然存在差异，没有深入分析特征，忽略了数据潜在关联；
	基于数据转换，可降低数据集分布差异	目前尝试较少，预测效果有待改进
项目相似性分组	可以有效定位相似项目，并使用相似项目进行缺陷预测	分组技术较不成熟；以项目为分组力度，忽视了项目数据集中存在无关数据的可能性
自适应预测模型	通过遗传算法，自适应获取数据并得到较好预测效果	自适应过程运算时间长，效率较低，不适合存在大量其他项目数据的情况

模型对开源项目进行跨项目缺陷预测建模，预测另一个开源项目。实验结果并不让人满意，但跨项目缺陷预测效果至少优于随机预测和一个简单的类级别预测模型。Cruz等人[4]尝试在预测前对数据分布进行调整，通过数据转换算法，获取分布相似的训练数据集和测试数据集，同时修剪数据集中异常的数据点，减少误报的发生。Nam等人发现源数据集与目标数据集的数据分布差异导致缺陷预测效果不理想[5]，提出了传递缺陷学习算法，通过传递成分分析[6]，转换源数据集数据，使得其接近目标数据集的数据分布，提高了跨项目缺陷预测的效果。

Turhan等人[7]提出了一种选择相邻数据并过滤不相干数据的方法，使得建立的跨项目缺陷预测模型的预测效果接近项目内缺陷预测。但是该方法受限于目标数据集实例数目，需要足够大小的目标数据集，不能运用在项目初期。Peters等人[8]改进了Turhan所提出的过滤技术，从源数据集出发，不断增量式查找与之相似的目标数据。对每一个源数据实例，若接近某个目标数据实例，则将源数据实例加入最终训练数据子集。

Tosun等人[9]利用过滤技术，成功利用NASA系统的数据来预测其他项目，证明了采用基于目标数据进行过滤挑选所获得的训练数据集，可以有效预测目标项目数据集的缺陷。

2.2 项目相似性分组

Jureczko和Madeyski[10]认为相似的项目之间预测效果会更好，通过对一个项目进行缺陷预测建模，能够将模型重用到其他组群中的项目上，并实验验证了这些预测模型的结果优于项目内缺陷预测。

Zhang等人[11]提出六元上下文因素组，定义了项目的相关属性，依据六元属性将项目分组，并建立了一个全局的跨项目缺陷预测框架。他们认为，对任意一个项目，可以依据其六元组找到相似的其他项目分组，并运用该组项目数据集进行缺陷预测。

2.3 自适应预测模型

研究者认为，跨领域技术的使用可以为解决问题提供新的想法。在大量的可选项目数据集中，选择一个大小适中、预测效果好的训练集，是一个重要的研究方向。Liu等人[12]引入基于搜索的选择策略，采用遗传算法逐步选择训练集数据，使用不同的机器学习方法对候选数据进行打分排序，选择最优数据集，得到了较为理想的实验结果。

2.4 小结

表1总结了上述技术的优点和不足，当前的跨项目缺陷预测相关研究仍然处于初期。

3 实例迁移与跨项目软件缺陷预测

3.1 跨项目软件缺陷预测方法概述

针对跨项目缺陷预测的挑战，本文从已有技术的不足点出发，吸取相关优点，采用基于实例的迁移学习技术，提出了一种有效的跨项目缺陷预测方法，如图1所示。整个预测过程以大量已标注的其他项目数据集和少量已标注的目标项目数据样本为输入，采用基于实例迁移的跨项目缺陷预测模型训练方法，训练得到适合目标项目的有效预测模型，针对目标项目数据集进行跨项目缺陷预测。

该方法不受限于特定的项目特征集，可以根据目标项目的特点任意选择所需的项目特征集合，从

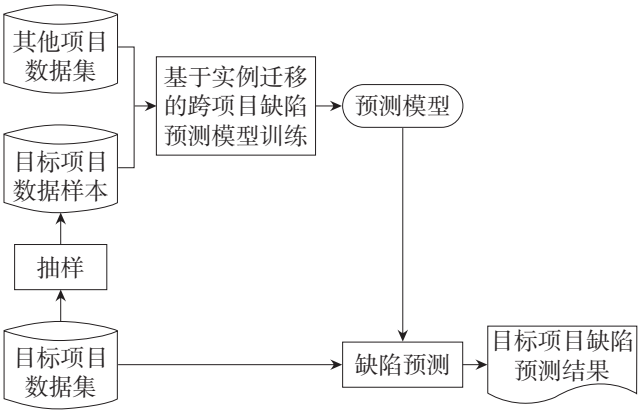

Fig.1 Cross-project defect prediction based on instance transfer

图1 基于实例迁移的跨项目缺陷预测

其他项目中提取相关特征的数据集,训练并获得一个适用于目标项目的缺陷预测模型。方法的本质是不断寻找项目间的相似性,增加数据实例间关联性的过程。通过筛选从其他项目中获得与目标项目关联性高的数据实例子集;通过数据迁移,使得最终获得的训练数据分布与目标数据分布相近。

预测模型训练是方法的核心,当前最有效的模型训练方法之一是自适应增强算法(Boosting)[13]。缺陷预测可以归纳为一个分类问题,即预测一个程序或模块是否有缺陷。自适应增强算法能够整合多个弱分类器,输出一个强分类器。本文结合迁移学习中基于实例的数据迁移思想,改进自适应增强训练模型,在训练数据过程中,多次迭代,基于数据对训练结果错误率的影响,降低相关性低的数据实例权重,侧重选择那些对目标项目分类有利的训练样本实例,最终输出一个分类能力较高的强分类器。以下各节将阐述基于实例迁移的预测模型训练。

3.2 TrAdaBoost算法的多源改进

项目数据集的差异性是导致跨项目缺陷预测结果不理想的重要原因,本文研究的出发点是如何从大量其他软件项目数据集中,提取适应于目标软件项目的跨项目缺陷预测训练集。

3.2.1 TrAdaBoost算法及其优缺点分析

Dai等人[14]扩展自适应增强算法到迁移学习中,提出了基于实例迁移的迭代加权TrAdaBoost(transfer AdaBoost)算法。在每次迭代中改变样本采样的权重,源领域中的无关样本权重被减弱,有利于模型训练的目标领域中的样本权重被加强。TrAdaBoost能够发现与目标数据集关联大的训练实例,并依据错误率定量地提高其在预测过程中的影响力;同时能整合多个弱分类器,有效减弱分类器的缺点,强化其优点。

然而,TrAdaBoost算法存在单源局限性。TrAdaBoost的输入为目标项目数据样本和单个其他项目训练数据。数据迁移结果的好坏,依赖于训练集与目标数据集的关联性,容易导致分类结果两极分化。训练集与目标数据集关联越大,则迁移效果越好。若两个数据集本身关联小,TrAdaBoost将修改大部分数据的权重,使得数据权重普遍较低,最终导致建立的分类模型预测能力较差。同时,若训练集空间某个分类值占比较低,则训练时该类的分类依据较少。

3.2.2 TrAdaBoost算法的改进

针对TrAdaBoost算法的单源局限性问题,本文的改进措施是,输入多个项目数据集,增加潜在相关数据的数目,这也正符合跨项目缺陷预测的背景。输入多个其他项目数据集用于模型训练,在增加潜在相关联数据数量的同时,也造成了最终分类误报率的增加。一种解决方式是通过筛选增加关联性高的数据的比例。然而关联性低的数据不代表没有影响力,因此另一种方式是基于自适应模型,对每个数据和每个迭代输出的分类器加权。综上,本文提出了两种TrAdaBoost算法的改进方案。

方案1 训练集合并与预处理(merge-source TrAdaBoost,MergeTrAdaBoost)。合并筛选数据,获得大小合适且与目标项目集关联性高的训练数据集,进行基于实例迁移的预测模型训练。

方案2 多源自适应演化(multi-source TrAdaBoost,MultiTrAdaBoost)。通过多次单源单目标训练,加权合并,形成强训练模型。

3.3 MergeTrAdaBoost算法

训练数据集与目标数据集的关联性低会导致TrAdaBoost算法分类效果变差,MergeTrAdaBoost在模型训练前根据目标数据集,从多个其他项目数据集中筛选出相关性大的训练数据子集。

MergeTrAdaBoost算法主要分为两个阶段：数据合并（merge）和数据迁移（transfer）。数据合并结果的好坏直接影响后期训练结果的准确度和时间效率。使用*k*-邻近算法会限制训练集样本个数，在跨项目缺陷预测背景下，存在大量其他项目训练数据。相对的，参照的目标数据样本只有少量。为了解决上述问题，从其他项目训练集出发，逐个判断，提取邻近于目标项目数据样本的训练实例。

MergeTrAdaBoost算法在TrAdaBoost算法的基础上进行筛选，以筛选后的其他项目数据集作为训练源，训练得到模型，并对目标项目数据集进行跨项目缺陷预测。算法的具体步骤包括：

（1）基于目标项目数据集 T_s，使用*k*-means对其他项目训练集 T_d 进行分类。假设其他项目训练集实例个数为 n，则 k 的值为 $n/10$，保证每个簇至少包含10个训练数据实例和1个目标项目数据实例。

（2）删除不包含数据的集合簇。

（3）对于每个簇中的训练数据 I，找到最接近的测试实例。若存在邻近测试实例，则将 I 加入候选训练集 T_d'。

3.4 MultiTrAdaBoost算法

多源自适应演化MultiTrAdaBoost是一个多次迭代的集成算法。本文采用Yao等人[15]改进的多源迁移算法MTrA，集成多个其他项目训练集，通过多次迭代，对目标项目数据集进行训练。每次迭代，从其他项目训练源中选择与目标项目数据集最相关的训练数据进行迁移学习，保证迁移后的知识与目标项目数据的相关性大。然而，MTrA直接忽略了其他项目数据源的影响，因此本文在MTrA的基础上，为与目标项目数据集相关性较弱的训练集实例添加权重，最大限度上利用其他项目训练数据集。

在TrAdaBoost算法基础上改进后的MultiTrAdaBoost算法的基本步骤包括：

（1）对于任意一个其他项目训练数据集 T_{d_k} 和目标项目数据集 T_s，合并获得训练集 T_k。

（2）每次循环，基于 k 个其他项目数据集。

（2.1）通过基本分类算法Learner，学习 T_k 并构建分类器 h_t^k。

（2.2）计算分类器的错误率，并针对错误率为 k 个分类器添加权重 w_i^k。

（2.3）针对本次迭代 k 个分类器的权重，合并分类器 h_t。

（3）基于循环所得分类器 h_t，整合获得强分类器 $h_f(x)$。

MultiTrAdaBoost算法的优点在于其考虑到所有其他项目训练集的影响，并为其影响力添加权重，不会忽视所有可用迁移知识。

4 实验

本文采用公开的PROMISE数据集中的数据进行实验，对比和分析基于TrAdaBoost、MergeTrAdaBoost和MultiTrAdaBoost的跨项目缺陷预测的能力。

4.1 实验数据和评估指标

本文按照项目的不同规模，从PROMISE数据集中选择了9个项目进行实验：

（1）小规模项目for08、pbeans2和sys，它们的实例数据分别为29、51和65条。

（2）中规模项目luc20、poi20和jedit43，它们的实例数据分别为195、314和492条。

（3）大规模项目prop6、xal24和cam16，它们的实例数据分别为660、723和965条。

缺陷预测算法选择了常用的朴素贝叶斯算法（naïve Bayes，NB）和决策树算法j48。

本文采用F度量（F-measure）、G度量（G-measure）和ROC曲线下面积（area under the ROC curve，AUC）作为预测评估指标。

为了保证评估过程中训练数据和测试数据相互独立，本文使用 k 折交叉方法，随机打乱数据集，并将数据集分为 N 份，取其中1份作为测试数据，其他 $N-1$ 份作为训练数据。为了减少偶然性造成的数据特异性，评估过程将重复 M 次，最终输出平均值。

4.2 TrAdaBoost实验

首先使用TrAdaBoost进行基于单目标单训练集迁移的跨项目预测实验。基于NB和j48，TrAdaBoost算法缺陷预测结果的F度量、G度量和AUC的指标值如表2至表7所示。

Table 2 F-measure of TrAdaBoost cross-project defect prediction based on NB

表2 基于NB的TrAdaBoost跨项目缺陷预测F度量指标

Project	for08	pbeans2	sys	luc20	poi20	jedit43	prop6	xal24	cam16
for08	—	0.907	0.894	0.853	0.907	0.907	0.915	0.907	0.875
pbeans2	0.717	—	0.717	0.788	0.717	0.717	0.717	0.717	0.717
sys	0.797	0.846	—	0.805	0.797	0.797	0.849	0.797	0.797
luc20	0.641	0.658	0.627	—	0.492	0.286	0.680	0.447	0.567
poi20	0.833	0.842	0.857	0.821	—	0.851	0.823	0.837	0.836
jedit43	0.001	0.001	0.001	0.001	0.001	—	0.001	0.001	0.001
prop6	0.851	0.649	0.792	0.660	0.861	0.827	—	0.843	0.845
xal24	0.752	0.728	0.773	0.749	0.706	0.811	0.752	—	0.783
cam16	0.773	0.671	0.682	0.749	0.749	0.749	0.753	0.764	—

Table 3 G-measure of TrAdaBoost cross-project defect prediction based on NB

表3 基于NB的TrAdaBoost跨项目缺陷预测G度量指标

Project	for08	pbeans2	sys	luc20	poi20	jedit42	prop6	xal24	cam16
for08	—	0.117	0.656	0.623	0.117	0.117	0.667	0.117	0.109
pbeans2	0.315	—	0.315	0.660	0.315	0.315	0.315	0.315	0.315
sys	0.239	0.636	—	0.675	0.239	0.239	0.563	0.239	0.239
luc20	0.580	0.670	0.608	—	0.560	0.470	0.675	0.528	0.589
poi20	0.661	0.577	0.597	0.484	—	0.584	0.528	0.444	0.611
jedit43	0.044	0.044	0.044	0.044	0.044	—	0.044	0.044	0.044
prop6	0.444	0.492	0.456	0.535	0.527	0.322	—	0.314	0.519
xal24	0.572	0.539	0.501	0.493	0.569	0.529	0.486	—	0.629
cam16	0.520	0.534	0.578	0.440	0.423	0.432	0.509	0.454	—

Table 4 AUC of TrAdaBoost cross-project defect prediction based on NB

表4 基于NB的TrAdaBoost跨项目缺陷预测AUC指标

Project	for08	pbeans2	sys	luc20	poi20	jedit43	prop6	xal24	cam16
for08	—	0.500	0.700	0.667	0.500	0.500	0.717	0.500	0.467
pbeans2	0.500	—	0.500	0.677	0.500	0.500	0.500	0.500	0.500
sys	0.500	0.678	—	0.688	0.500	0.500	0.640	0.500	0.500
luc20	0.641	0.657	0.628	—	0.561	0.473	0.683	0.530	0.578
poi20	0.685	0.639	0.671	0.676	—	0.646	0.603	0.574	0.655
jedit43	0.500	0.500	0.500	0.500	0.500	—	0.500	0.500	0.500
prop6	0.564	0.521	0.544	0.529	0.625	0.514	—	0.604	0.606
xal24	0.571	0.582	0.585	0.550	0.571	0.604	0.558	—	0.655
cam16	0.592	0.580	0.596	0.548	0.556	0.544	0.575	0.562	—

从表2和表5中可以看出:当训练集数目远大于测试集时,预测效果准确率较好。例如for08为目标项目时,由于其本身实例数据较少,可以有效利用大数据集进行预测分析并获得高准确率。对于大部分目标项目,基于NB和j48,TrAdaBoost可以有效地进行迁移,并获得较好的跨项目缺陷预测F度量指标,F度量值都在0.7以上。同时,j48比NB的分类效果相对更佳。少量目标项目的跨项目预测F度量值小于0.5。以luc20和jedit43为例分析其原因,可以发现luc20和jedit43的缺陷率分别为46.7%和2.2%,

Table 5 F-measure of TrAdaBoost cross-project defect prediction based on j48

表5 基于j48的TrAdaBoost跨项目缺陷预测F度量指标

Project	for08	pbeans2	sys	luc20	poi20	jedit43	prop6	xal24	cam16
for08	—	0.907	0.858	0.907	0.907	0.907	0.907	0.841	0.907
pbeans2	0.717	—	0.717	0.717	0.717	0.717	0.717	0.673	0.717
sys	0.797	0.750	—	0.797	0.797	0.797	0.797	0.797	0.797
luc20	0.697	0.758	0.703	—	**0.863**	**0.297**	**0.648**	**0.371**	0.697
poi20	0.814	0.861	0.874	0.814	—	**0.701**	**0.852**	**0.874**	0.836
jedit43	0.943	0.906	0.934	0.908	**0.884**	—	**0.792**	**0.770**	0.786
prop6	0.866	0.884	0.884	0.888	**0.877**	**0.732**	—	**0.896**	0.875
xal24	0.836	0.852	0.848	0.841	0.853	0.742	0.837	—	0.853
cam16	0.787	0.792	0.792	0.823	0.800	0.728	0.807	0.808	—

Table 6 G-measure of TrAdaBoost cross-project defect prediction based on j48

表6 基于j48的TrAdaBoost跨项目缺陷预测G度量指标

Project	for08	pbeans2	sys	luc20	poi20	jedit43	prop6	xal24	cam16
for08	—	0.117	0.105	0.117	0.117	0.117	0.117	0.102	0.117
pbeans2	0.315	—	0.315	0.315	0.315	0.315	0.315	0.359	0.315
sys	0.239	0.511	—	0.239	0.239	0.239	0.239	0.239	0.239
luc20	0.664	0.713	0.702	—	0.757	0.498	0.667	0.498	0.687
poi20	0.679	0.641	0.629	0.681	—	0.718	0.620	0.594	0.655
jedit43	0.443	0.326	0.539	0.326	0.187	—	0.299	0.398	0.182
prop6	0.537	0.537	0.524	0.552	0.560	0.582	—	0.573	0.573
xal24	0.654	0.630	0.617	0.591	0.605	0.675	0.578	—	0.632
cam16	0.598	0.596	0.588	0.628	0.616	0.659	0.572	0.588	—

Table 7 AUC of TrAdaBoost cross-project defect prediction based on j48

表7 基于j48的TrAdaBoost跨项目缺陷预测AUC指标

Project	for08	pbeans2	sys	luc20	poi20	jedit43	prop6	xal24	cam16
for08	—	0.500	0.450	0.500	0.500	0.500	0.500	0.433	0.500
pbeans2	0.500	—	0.500	0.500	0.500	0.500	0.500	0.465	0.500
sys	0.500	0.560	—	0.500	0.500	0.500	0.500	0.500	0.500
luc20	0.701	0.756	0.702	—	0.684	0.500	0.662	0.500	0.696
poi20	0.692	0.685	0.684	0.670	—	0.701	0.586	0.632	0.698
jedit43	0.563	0.442	0.642	0.532	0.466	—	0.481	0.421	0.433
prop6	0.639	0.655	0.631	0.664	0.631	0.580	—	0.707	0.630
xal24	0.687	0.679	0.670	0.700	0.664	0.684	0.637	—	0.664
cam16	0.639	0.668	0.649	0.686	0.659	0.653	0.638	0.643	—

jedit43没有足够的依据来分类缺陷较多的luc20，从而导致最终分类结果不理想。综上，跨项目缺陷预测需要足够多的训练样本，因此可以使用多个项目集作为训练集。

从表3和表6中可以看出：50%以上目标项目误分类现象严重，G度量值低于0.5。当训练集数目远大于测试集（目标项目for08、pbeans2和sys）时，误报率增加，并使得最终G度量值较低。

从表4和表7中的AUC值同样可以看出，误报率影响了最终的分类结果。如果能够有效减少造成误

报率的数据实例，就能进一步提高G度量和AUC的指标。这也从另一角度证明了选择训练集的重要性，即需要选择关联性较高的训练数据来减少误报率。

值得注意的是，表2中，基于NB用任何数据集预测jedit43得到的F度量值都为0.01。而表5中，基于j48进行缺陷预测的结果甚至超过0.9。这是因为jedit43的缺陷率为2.2%，使得贝叶斯概率接近0。而使用决策树，可以有效判断实例所在的分支，不受实例缺陷率的影响。故而这个特殊现象将不作为验证MergeTrAdaBoost和MultiTrAdaBoost效果的依据。

总结上述实验的分析结果，验证了多源多目标改进算法的必要性：

(1)使用多个其他项目集作为训练集，能提高最终预测结果的准确度。

(2)有效筛选或迁移其他项目集数据，是基于实例迁移的跨项目缺陷预测模型训练减少误报率的前提。

4.3 MergeTrAdaBoost 和 MultiTrAdaBoost 实验

接下来，本文基于MergeTrAdaBoost和MultiTrAdaBoost算法进行跨项目缺陷预测。当选择其中1个项目作为目标项目时，其他8个项目数据则作为训练数据集。

表8展示了基于NB分类算法的MergeTrAdaBoost和MultiTrAdaBoost跨项目预测结果。采用MergeTrAdaBoost，F度量值中7/8的结果高于0.75，G度量值中7/8的结果高于0.5，AUC值都不低于0.5。采用MultiTrAdaBoost，F度量值中5/8的结果高于0.75，G度量值中6/8的结果高于0.5，AUC值都不低于0.5。

表9展示了基于j48分类算法的MergeTrAdaBoost和MultiTrAdaBoost跨项目预测结果。采用MergeTr-

Table 8 F-measure, G-measure and AUC of MergeTrAdaBoost and MultiTrAdaBoost based on NB

表8 基于NB的MergeTrAdaBoost和MultiTrAdaBoost预测的F度量、G度量和AUC

目标数据集	MergeTrAdaBoost			MultiTrAdaBoost		
	F-measure	*G-measure*	*AUC*	*F-measure*	*F-measure*	*AUC*
for08	0.907	0.560	0.500	0.907	0.117	0.500
pbeans2	0.788	0.315	0.677	0.815	0.571	0.638
sys	0.817	0.672	0.746	0.838	0.472	0.593
luc20	0.673	0.675	0.672	0.554	0.604	0.575
poi20	0.848	0.661	0.658	0.822	0.597	0.581
prop6	0.860	0.657	0.662	0.836	0.552	0.593
xal24	0.815	0.656	0.658	0.743	0.667	0.670
cam16	0.766	0.617	0.600	0.739	0.598	0.537

Table 9 F-measure, G-measure and AUC of MergeTrAdaBoost and MultiTrAdaBoost based on j48

表9 基于j48的MergeTrAdaBoost和MultiTrAdaBoost预测的F度量、G度量和AUC

目标数据集	MergeTrAdaBoost			MultiTrAdaBoost		
	F-measure	*G-measure*	*AUC*	*F-measure*	*G-measure*	*AUC*
for08	0.907	0.117	0.500	0.907	0.117	0.500
pbeans2	0.766	0.315	0.728	0.717	0.315	0.500
sys	0.797	0.239	0.653	0.797	0.239	0.500
luc20	0.739	0.694	0.736	0.672	0.673	0.674
poi20	0.892	0.719	0.761	0.881	0.713	0.648
jedit43	0.938	0.568	0.515	0.742	0.476	0.557
prop6	0.880	0.662	0.698	0.882	0.547	0.641
xal24	0.843	0.663	0.711	0.853	0.663	0.699
cam16	0.830	0.630	0.669	0.796	0.586	0.635

AdaBoost，F 度量值中 8/9 的结果高于 0.75，G 度量值中 6/9 的结果高于 0.5，AUC 值都不低于 0.5。采用 MultiTrAdaBoost，F 度量值中 6/9 的结果高于 0.75，G 度量值中 5/9 的结果高于 0.5，AUC 值都不低于 0.5。

图 2、图 3 和图 4 进一步对比了 TrAdaBoost、MergeTrAdaBoost 和 MultiTrAdaBoost 算法的跨项目缺陷预测结果的 F 度量、G 度量和 AUC 指标。图中前 8 项数据是基于 TrAdaBoost 进行单源单目标迁移训练获得的评估结果，后 2 项数据展示了通过 MergeTrAdaBoost 和 MultiTrAdaBoost 两种算法进行模型学习和预测所得的评估结果。以图 2 中 for08 为例，纵轴为 F 度量，横轴为训练对象。蓝色柱状代表以 j48 进行训练获得的 F 度量值，红色柱状代表以 NB 进行训练获得的 F 度量值。

由于基于 NB 训练 jedit43 时存在异常，故而对比实验不考虑其预测结果。

由图 2 可以看出，MergeTrAdaBoost 和 MultiTrAdaBoost 预测结果 F 度量值优于单源单目标 TrAdaBoost 训练模型，同时不会获得比 TrAdaBoost 更差的预测准确率。这是因为有足够的实例用于建模，并且两种新算法能对数据进行筛选加权，获得一个与目标数据关联性大的训练模型。

（1）MergeTrAdaBoost 和 MultiTrAdaBoost 获得的 F 度量值普遍高于 0.75，同时准确率优于单源单目标训练模型。

（2）对于大部分目标训练项目，MergeTrAdaBoost

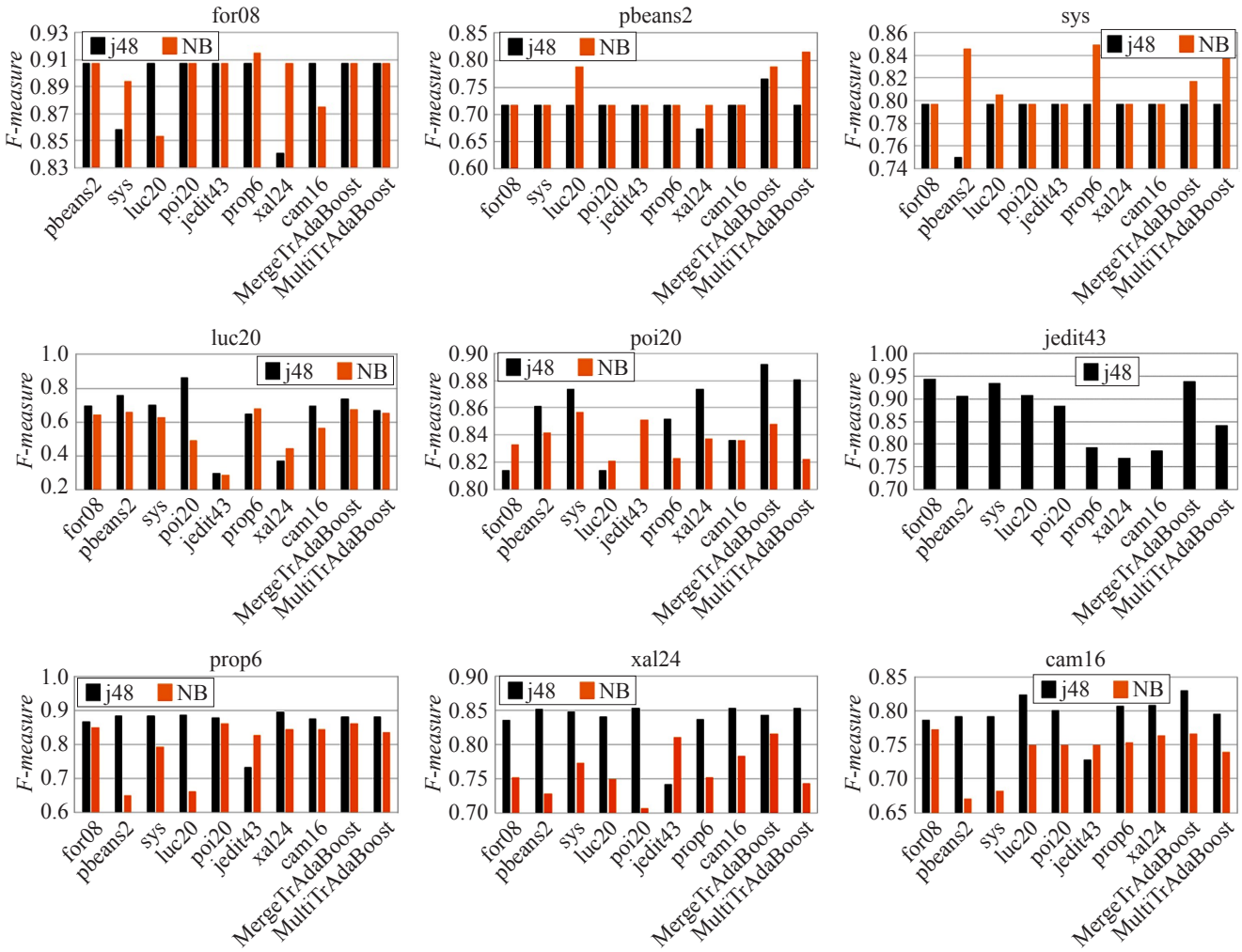

Fig.2 Comparison of F-measure

图 2 F 度量指标对比

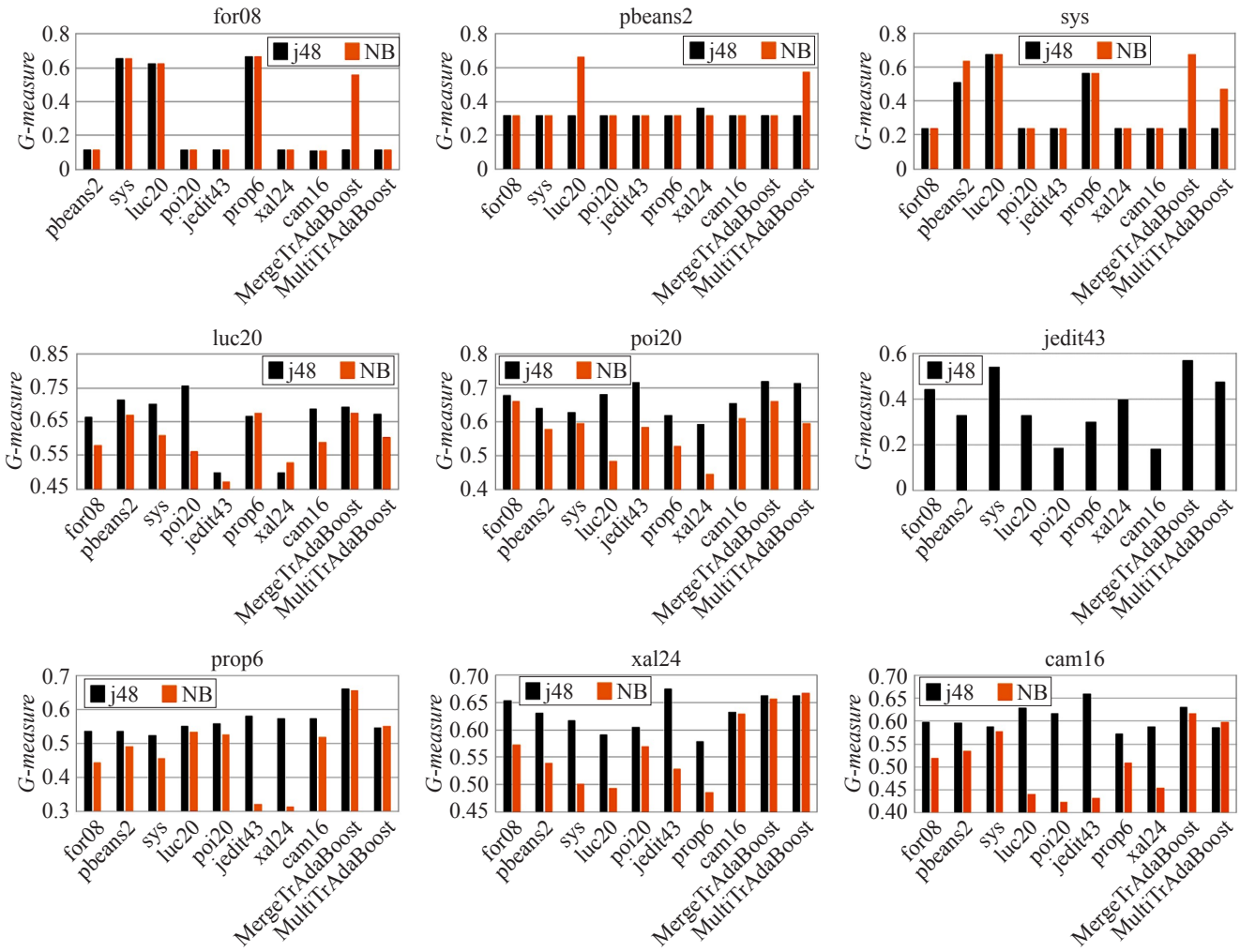

Fig.3 Comparison of G-measure

图3 G度量指标对比

的F度量值优于MultiTrAdaBoost。

(3)当训练集数据规模远大于目标数据,换句话说,当仅有少量目标样本数据时(for08、pbeans2和sys),NB所获得的F度量值较高;而当有更多参照样本时,j48分类精确度更优。

由图3可以看出,MergeTrAdaBoost和MultiTrAdaBoost预测结果G度量值优于大部分单源单目标TrAdaBoost训练模型,有效降低了误报率的影响。

(1)MergeTrAdaBoost和MultiTrAdaBoost获得的G度量值普遍高于0.5,同时优于或不低于单源单目标训练模型。

(2)对于大部分目标训练项目,MergeTrAdaBoost训练模型所得的G度量值优于MultiTrAdaBoost。同时,对比9个目标训练项目,基于j48的MergeTrAdaBoost获得的G度量值相对高于NB。

(3)当训练集数据规模远大于目标数据时,例如for08、pbeans2和sys,NB所获得的G度量值较高,而j48无法降低误报率。

(4)对于TrAdaBoost、MergeTrAdaBoost和MultiTrAdaBoost,基于j48的自适应模型所获得的G度量值普遍较高。

由图4可以看出,MergeTrAdaBoost和MultiTrAdaBoost预测结果AUC值优于大部分单源单目标TrAdaBoost训练模型,有效降低了误报率的影响。

(1)MergeTrAdaBoost和MultiTrAdaBoost算法获得的AUC值普遍高于0.5。

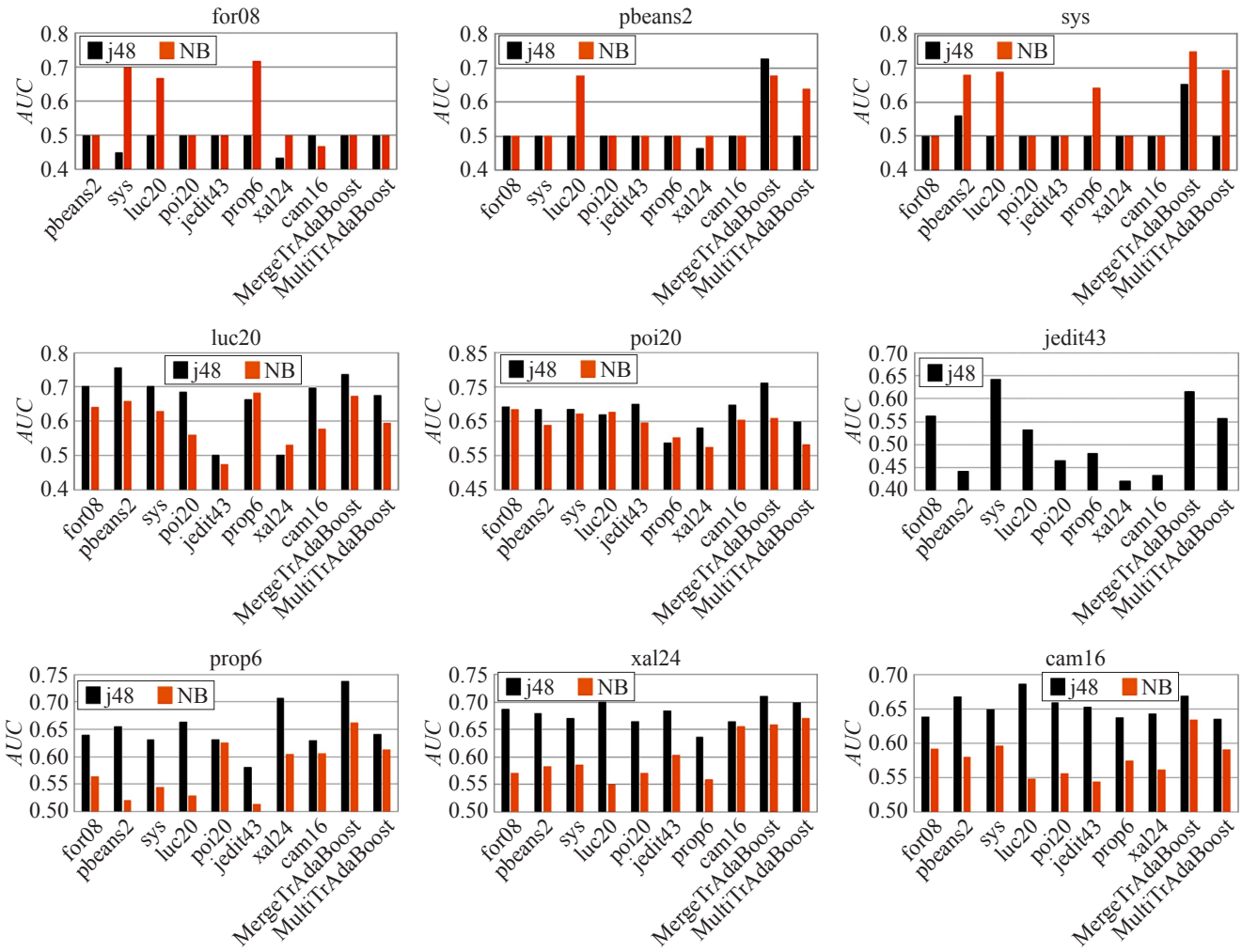

Fig.4 Comparison of AUC

图4 AUC指标对比

（2）对于大部分目标训练项目，MergeTrAdaBoost算法训练模型所得的AUC值优于MultiTrAdaBoost。同时对比9个目标训练项目，基于j48的MergeTrAdaBoost获得的AUC值相对高于NB。

综上，MergeTrAdaBoost和MultiTrAdaBoost算法适应于跨项目缺陷预测多训练集的情况，有效避免了TrAdaBoost算法单源单目标缺陷预测的两极分化问题。

4.4 与项目内缺陷预测的对比实验

进而，本文将基于实例迁移的跨项目软件缺陷预测与项目内软件缺陷预测进行对比实验。预测算法选取Naïve Bayes算法。

为了使实验更充分，增加了项目数。从PROMISE数据集选用的实验数据如下：

（1）其他项目数据集log11、ivy11、ant13、log10、syn10、ant14、luc20、vel14、vel15、syn11、vel16、poi15、ivy14、luc22、syn12、jedit32、ant15、jedit40、jedit41、poi20、cam10、ant16、jedit42、poi25、xer、xer12、poi30、xer13、xer14、cam12、prop6、xal24、xal25、cam14、xal26。

（2）目标项目数据集ckjm、wsp、skleb、szy、pb1、inter、kal、nier、for07、zuzel、for08、work、term、berek、sera、skar、pb2、pdf、elearn、sys。

实验结果如表10所示，从中可得出以下结论：在目标项目数据集不足的情况下，MergeTrAdaBoost算法和MultiTrAdaBoost算法都能达到甚至超过数据集充足时项目内缺陷预测的预测效果。

Table 10 G-measure of prediction results for different algorithms

表10 不同算法预测结果的G度量指标对比

数据集	项目内	MergeTrAdaBoost	MultiTrAdaBoost
ckjm	0	0.50	**0.70**
wsp	0.67	0.44	0.50
skleb	0.50	0.48	0.77
szy	0.67	0.49	0.53
pbl	0.63	0.36	0.55
inter	0	**0.79**	**0.77**
kal	0.63	0.44	**0.71**
nier	0.67	0.47	0.54
for07	0	**0.69**	**0.87**
zuzel	0.75	0.62	0.85
for08	0	0.12	0.12
work	0.47	**0.60**	**0.58**
term	0.63	0.48	**0.87**
berek	0.88	0.86	0.86
sera	0.76	0.52	**0.75**
skar	0.48	0.29	**0.71**
pb2	0.33	**0.57**	**0.67**
pdf	0.70	0.53	**0.75**
elearn	0	0.14	**0.82**
sys	0.69	0.57	**0.61**
平均	0.473	0.498	0.676

5 结束语

为了通过解决不同项目的差异性问题，提高跨项目缺陷预测精确度，降低预测结果误报率，本文研究提出了基于实例迁移的跨项目缺陷预测方法。该方法采用迁移学习和自适应增强技术，在训练模型阶段，多次迭代并对训练实例加权，降低相关性低的数据造成的影响，最终输出一个分类能力较高的强分类器。其中模型训练引入TrAdaBoost算法，为解决其单源单目标的局限性，提出两种改进方案：训练集合并预处理MergeTrAdaBoost算法和多源自适应演化MultiTrAdaBoost算法。

实验表明，MergeTrAdaBoost和MultiTrAdaBoost算法适应于跨项目缺陷预测多训练集的情况，有效避免了TrAdaBoost算法的单源单目标缺陷预测两极分化问题，能获得较于TrAdaBoost更好的预测结果。这是因为有足够的实例用于建模，并且两种改进算法能对实例进行筛选加权，获得一个与目标项目关联性大的缺陷预测训练模型。在克服项目开始时缺陷预测的冷启动问题的同时，MergeTrAdaBoost和MultiTrAdaBoost能够获得与传统项目内缺陷预测更优的预测效果。

下一步的研究工作主要针对以下两个方面：在更多的项目上进行实践，深入研究该类源项目和目标项目存在较好结果的原因，并改进预测模型；采用更多数据集，开展更多对比实验，寻找算法新的改进方向。

References:

[1] Wang Qing, Wu Shujian, Li Mingshu. Software defect prediction[J]. Journal of Software, 2008, 19(7): 1565-1580.

[2] Turhan B, Menzies T, Bener A, et al. On the relative value of cross-company and within-company data for defect prediction[J]. Empirical Software Engineering, 2007, 14(5): 540-578.

[3] Briand L C, Melo W L, Wust J. Assessing the applicability of fault-proneness models across object-oriented software projects[J]. IEEE Transactions on Software Engineering, 2002, 28(7): 706-720.

[4] Cruz A, Ochimizu K. Towards logistic regression models for predicting fault-prone code across software projects[C]//Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement, Lake Buena Vista, USA, Oct 15-16, 2009. Piscataway, USA: IEEE, 2009: 460-463.

[5] Nam J, Pan S J, Kim S. Transfer defect learning[C]//Proceedings of the 2013 International Conference on Software Engineering, San Francisco, USA, May 18-26, 2013. Piscataway, USA: IEEE, 2013: 382-391.

[6] Pan S J, Tsang I W, Kwok J T, et al. Domain adaptation via transfer component analysis[J]. IEEE Transactions on Neural Networks, 2010, 22(2): 199-210.

[7] Turhan B, Menzies T, Bener A B, et al. On the relative value of cross-company and within-company data for defect prediction[J]. Empirical Software Engineering, 2009, 14(5): 540-578.

[8] Peters F, Menzies T, Marcus A. Better cross company defect prediction[C]//Proceedings of the 10th International Workshop on Mining Software Repositories, San Francisco, USA, May 18-19, 2013. Piscataway, USA: IEEE, 2013: 409-418.

[9] Tosun A, Bener A B, Kale R. AI-based software defect predictors: applications and benefits in a case study[J]. AI Magzine, 2011, 32(2): 57-68.

[10] Jureczko M, Madeyski L. Towards identifying software project clusters with regard to defect prediction[C]//Proceedings of the 6th International Conference on Predictive Models in Software Engineering, Timisoara, Romania, Sep 12-13, 2010. New York, USA: ACM, 2012: 9.

[11] Zhang F, Mockus A, Keivanloo I, et al. Towards building a universal defect prediction model[C]//Proceedings of the 11th Working Conference on Mining Software Repositories, Hyderabad, India, May 31-Jun 1, 2014. New York, USA: ACM, 2014: 182-191.

[12] Liu Yi, Khoshgoftaar T M, Seliya N. Evolutionary optimization of software quality modeling with multiple repositories[J]. IEEE Transactions on Software Engineering, 2010, 36(6): 852-864.

[13] Freund Y, Schapire R E. A decision-theoretic generalization of on-line learning and an application to boosting[J]. Journal of Computer and System Sciences, 1995, 55(1): 119-139.

[14] Dai Wenyuan, Yang Qiang, Xue Guirong, et al. Boosting for transfer learning[C]//Proceedings of the 24th International Conference on Machine Learning, Corvallis, USA, Jun 20-24, 2007. New York, USA: ACM, 2007: 193-200.

[15] Yao Yi, Doretto G. Boosting for transfer learning with multiple sources[C]//Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, USA, Jun 13-18, 2010. Piscataway, USA: IEEE, 2010: 1855-1862.

附中文参考文献：

[1] 王青, 伍书剑, 李明树. 软件缺陷预测技术[J]. 软件学报, 2008, 19(7): 1565-1580.

MAO Fagui was born in 1992. He is an M.S. candidate at Shanghai Jiao Tong University. His research interests include software defect prediction and software quality, etc.

毛发贵(1992—)，男，江西萍乡人，上海交通大学硕士研究生，主要研究领域为软件缺陷预测，软件质量等。

LI Biwen was born in 1990. She received the M.S. degree in software engineering from Shanghai Jiao Tong University in 2015. Her research interests include software defect prediction and software quality, etc.

李碧雯(1990—)，女，江苏苏州人，2015年于上海交通大学软件工程专业获得硕士学位，目前在VMware公司任质量工程师，主要研究领域为软件缺陷预测，软件质量等。

SHEN Beijun was born in 1969. She received the Ph.D. degree in software engineering from Institute of Software, Chinese Academy of Sciences in 2002. Now she is an associate professor at Shanghai Jiao Tong University, and the senior member of CCF. Her research interests include software process and software repository mining, etc.

沈备军(1969—)，女，浙江慈溪人，2002年于中国科学院软件所软件工程专业获得博士学位，现为上海交通大学副教授，CCF高级会员，主要研究领域为软件过程，软件仓库挖掘等。近年来发表学术论文100余篇，主持多项国家及省部级项目。