基于GAN数据增强的软件缺陷预测聚合模型

徐金鹏，郭新峰，王瑞波，李济洪

引用本文

徐金鹏，郭新峰，王瑞波，李济洪. 基于GAN数据增强的软件缺陷预测聚合模型[J]. 计算机科学，2023，50(12)：24-31.

XU Jinpeng, GUO Xinfeng, WANG Ruibo, LI Jihong. Aggregation Model for Software Defect Prediction Based on Data Enhancement by GAN [J]. Computer Science, 2023, 50(12): 24-31.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.221100171

基于 GAN 数据增强的软件缺陷预测聚合模型

徐金鹏[1]　郭新峰[1]　王瑞波[2]　李济洪[2]
1 山西大学自动化与软件学院　太原 030006
2 山西大学现代教育技术学院　太原 030006
(202123604008@email.sxu.edu.cn)

摘　要　在软件缺陷预测任务中,通常基于 C&K 等静态软件特征数据集,使用机器学习分类算法来构建软件缺陷预测(SDP)模型。然而,大多数静态软件特征数据集中缺陷数较少,数据集的类不平衡问题较为严重,导致学习到的 SDP 模型的预测性能较差。文中基于生成对抗网络(GAN),并利用 FID 得分筛选生成正例样本数据,增强正例样本量,然后在组块正则化 $m\times2$ 交叉验证($m\times2$BCV)框架下,通过众数投票法聚合多个子模型的结果,最终构成 SDP 模型。以 PROMISE 数据库下的 20 个数据集为实验数据集,采用随机森林算法构建 SDP 聚合模型。实验结果表明,与传统的随机上采样、SMOTE、随机下采样相比,所提 SDP 聚合模型的 F1 平均值分别提高了 10.2%,5.7%,3.4%,且 F1 的稳定性也得到相应提高;所提 SDP 聚合模型在 20 个数据集的评测中,有 17 个 F1 值最高。从 AUC 指标来看,所提方法与传统的采样方法没有明显差异。

关键词:生成对抗网络;数据增强;组块正则化交叉验证;软件缺陷预测;聚合模型
中图法分类号　TP311

Aggregation Model for Software Defect Prediction Based on Data Enhancement by GAN

XU Jinpeng[1],GUO Xinfeng[1],WANG Ruibo[2] and LI Jihong[2]
1 School of Automation and Software Engineering,Shanxi University,Taiyuan 030006,China
2 School of Modern Education Technology,Shanxi University,Taiyuan 030006,China

Abstract　In the task of software defect prediction,the machine learning classification algorithm is usually used to build a software defect prediction(SDP) model based on dataset with static software features such as C&K metrics. However,the number of defects in most datasets with static software metrics is small,the class imbalance in the dataset is serious,resulting in the low prediction performance of the model. Based on generation adversarial network(GAN),this paper uses FID score screening to generate positive sample data,enhances the amount of postitive data,and then aggregates the results of learned models by majority-voting,and finally build the SDP model based on block-regularized $m\times2$ Cross validation($m\times2$BCV). 20 datasets in PROMISE database are used as the experimental datasets,and random forest algorithm is used to build model. Experimental results show that,compared with the traditional random over-sampling,SMOTE,and random under-sampling,the average F1 values of the SDP aggregation model in the 20 datasets is increased by 10.2%,5.7%,and 3.4% respectively,and the stability of F1 is also improved accordingly. In 17 of the 20 datasets,the SDP aggregation models have the highest F1 values. From the AUC index,there is no significant difference between the proposed method and the traditional sampling method.

Keywords　Generative adversarial network,Data enhancement,Block-regularized $m\times2$ cross validation,Software defect prediction,Aggregation model

1　引言

近年来,由于软件结构规模和复杂度不断增加,软件质量保证面临着越来越大的挑战。随着机器学习的广泛应用,许多学者开始探索使用机器学习来构建软件缺陷预测(SDP)模型,以改进软件质量保证过程[1]。软件缺陷预测模型通常在静态代码特征(也称为静态软件度量)的数据集上训练得到模型,预测新的软件代码模块是否有缺陷。目前,基于静态代码特征的软件缺陷预测模型普遍性能不高且稳定性较差[2]。许多研究发现,这些 SDP 模型预测性能不佳的主要原因之一是 SDP 数据集中正例样本的比例较小[3-5],从机器学习角度来说,是因为大多数 SDP 数据集存在较为严重的类不平衡

到稿日期:2022-11-21　返修日期:2023-04-07
基金项目:国家自然科学基金青年科学基金(61806115)
This work was supported by the Young Scientists Fund of the National Natural Science Foundation of China(61806115).
通信作者:李济洪(lijh@sxu.edu.cn)

问题，即正例（有缺陷）样本的个数远远少于负例（无缺陷）样本的个数，训练得到的模型往往偏向于负例，导致模型对正例的预测性能较差且不稳定。

事实上，为缓解 SDP 建模中的数据类不平衡问题，一些学者已经使用若干重采样方法，如随机上采样（Random Over-Sampling，ROS）、SMOTE 和随机下采样（Random Under-Sampling，RUS）等来进行数据增强[6-8]。这些重采样方法具有简单易用、不依赖模型等优点，在许多 SDP 数据建模中被验证是有效的。然而 ROS 只是通过对正例样本的简单复制来达到与负例样本的均衡；RUS 是删除负例样本来达到与正例样本的均衡，这必然会失去相应样本的信息；SMOTE 是通过插值法生成新的正例样本，而新生成的正例样本与原样本很难保证来自相同的分布，这不利于模型预测性能的提升[9-10]。因此，学者们更希望根据已有的正例样本学习其分布，生成新的样本以提升模型的预测性能[11]，而不是通过简单的复制和插值来生成正例样本。

Goodfellow 等[12]提出的生成对抗网络（Generative Adversarial Network，GAN）是一种生成模型，它通过对抗神经网络可以生成与原样本分布几乎相同的新数据。有学者基于 GAN 新生成的图像来处理类不平衡的图像识别问题，识别的精度得到了较好的提升[13]。

在 SDP 数据集上进行机器学习算法建模时，能否也利用 GAN 来进行数据增强以提升模型预测性能？为此，本文基于 SDP 数据集，采用 GAN 生成新的正例样本，使得正、负例样本数量基本均衡，以提升模型的预测性能。

在 SDP 建模中，本文采用带有数据切分约束的 m 次 2 折的交叉验证，称为组块正则化交叉验证（简记为 $m\times2$ BCV）[14]。m×2 BCV 方法有许多优良特性，它既适用于模型评估与选择，又可以在选到最优模型的基础上，以众数投票来构建 SDP 聚合模型。理论和实验已经证明聚合模型可以提高模型的预测性能和稳定性[15]。Wang[16]的研究表明，$m\times2$ BCV 适用于模型比较，因此，本文采用 $m\times2$ BCV 下传统重采样方法 ROS，RUS 和 SMOTE 在 SDP 模型上进行对比。

本文的 SDP 机器学习算法选取随机森林（Random Forest，RF）。在软件缺陷预测的相关研究中已经验证了 RF 在绝大多数数据集上的性能优于一般的分类算法（如逻辑回归、支持向量机、决策树等）[17]。鉴于本文的目的是验证 GAN 数据增强是否能提升 SDP 模型的预测性能，因此本文选用了预测性能较好的 RF 算法，相应的基线模型选取了 ROS，RUS 和 SMOTE 重采样下的 RF。

本文的模型预测性能评价指标为 F1 和 AUC。Wang 等[18]在 $m\times2$ BCV 下导出了 F1 的分布函数，因此本文以 F1 的非对称置信区间长度来评价模型预测性能的稳定性，具体公式详见 6.2 节。

本文选取 PROMISE 下的 20 个数据集为实验数据集，每个数据集包含 20 个静态特征（见表 1）。由于 GAN 模型一般处理服从高斯分布的样本，而每个 SDP 数据集静态特征取值多数为正整数，因此，本文对每个静态特征先进行对数变换，再进行标准化处理。

本文的实验结果表明，基于 GAN 对软件缺陷数据中的正例样本进行数据增强，可以提升 SDP 模型的预测性能和稳定性。

本文第 2 章介绍 SDP 相关研究；第 3 章介绍组块正则化交叉验证方法（$m\times2$BCV）；第 4 章详细介绍基于 GAN 的数据增强的 SDP 聚合模型框架方法；第 5 章进一步介绍在 $m\times2$BCV 下基于 GAN 的 3 种数据生成算法；第 6 章给出了实验结果分析以及结论；最后总结全文。

表 1 SDP 数据集中的 C&K 度量

Table 1 C&K metrics in SDP dastset

度量名	含义
WMC	类的加权方法数
DIT	类在继承树中的深度
NOC	类在继承树中的孩子节点数
CBO	与该类存在耦合关系的类的数目
RFC	该类可以调用的外部方法数
LCOM	方法的不一致性度量
LCOM3	方法的不一致性度量，不同于 LCOM
LOC	代码总行数
DAM	数据访问度量
MOA	聚合的度量
MFA	函数抽象的度量
CAM	类方法数
IC	继承耦合
CBM	方法之间的耦合
AMC	平均方法复杂度
Ca	有多少其他类调用了特定类
Ce	有多少其他类被特定类调用
Max(CC)	方法的最大 McCabe 圈复杂度值
Avg(CC)	方法的平均 McCabe 圈复杂度值
NPM	公共方法树

2 相关研究工作

SDP 建模可以被看作是对有缺陷样本和无缺陷样本进行分类的二分类问题。多年来，许多学者探索使用成熟的机器学习算法来构建 SDP 模型，如决策树、支持向量机、随机森林、神经网络等[19-23]。然而，在大多数 SDP 数据集中，有缺陷的数据往往占整个数据集的 10%～30%，存在较为严重的类不平衡问题，这为 SDP 模型的训练带来了困难[24]。许多研究也表明，类别不均衡是导致传统的机器学习算法预测性能不佳的主要因素之一。

鉴于此，一些学者提出了代价敏感学习、重采样等方法[25]来缓解类不平衡问题的影响。代价敏感学习通过提高对正例样本误分类的代价使分类器能更好地识别正例样本，但代价函数需要先验知识进行人为赋值。而重采样方法由于简单易用而得到了广泛使用，如 SDP 建模中广泛使用的 ROS，RUS 和 SMOTE。从提升模型预测性能的效果来看，3 种重采样方法各有利弊，总体来说，RUS 要略好于 ROS 和 SMOTE。然而这些重采样方法仅使用复制、删除、插值的方式，使得数据类别形式上达到均衡以利于模型学习，但对数据集中的正、负例样本分布的信息挖掘不够深入[26-27]。

生成对抗网络(GAN)算法的出现,给数据增强带来了新的思路。GAN 通过逼近真实样本的分布来生成新样本,先在图像生成领域了取得良好的效果,之后引起了机器学习其他领域的广泛关注。Qian 等[28]将 GAN 应用到跨项目的软件缺陷预测领域上,将源代码转化成抽象语法树,将筛选后的抽象语法树节点通过 CBOW 模型转化为词向量,使用 BLSTM 作为特征提取器,利用 GAN 中的鉴别器将目标特征的分布逼近原项目特征的分布,从而进行跨项目 SDP 预测。实验结果表明 GAN 的优势非常明显。但这种方法需要大量抽象语法树的处理,从其实验选择的数据来看,基本是缺陷率超过 50%的数据集,而且其生成新数据的环节使用了测试集中特征的信息,即测试集的信息参与了模型学习训练环节。Sheng 等[29]将 GAN 和 CNN 结合起来,提出了 ADCNN 模型来进行基于抽象语法树节点的跨项目软件缺陷预测,其在 F1, PofB20 等评价指标上优于其他方法。GAN 算法的稳定性相对较差,为此,本文提出了其他策略来提升基于 GAN 的 SDP 模型的稳定性。

根据集成学习的理论,将多个模型的预测结果通过众数投票聚合能够提升预测性能和稳定性,例如传统的 Bagging 方法。事实上,Bagging 是基于 Bootstrap 抽样训练得到多个模型,再来投票聚合模型的预测结果,但是 Bootstrap 一般需要较多的抽样次数(如大于 200 次),不太适合耗时的 GAN 算法。另外,Bagging 是在 Bootstrap 有放回抽样之后以剩下的样本作为验证集来评估模型,该验证集可能与原样本的分布(如类别标签分布)差异较大,会导致模型预测性能评测准确度较差。因此,本文设计了一种优化的数据重抽样(子抽样)方案,对提升投票聚合模型的效率和稳定性有十分重要的意义。

本文采用一种被称为组块正则化交叉验证($m\times$2BCV)的数据重抽样方法。与大多数直接套用 $m\times$2BCV 不同,本文先在静态特征上对相对较少的正例样本采用 GAN 来生成新的样本数据,再进行软件缺陷预测建模,最后对 $m\times$2BCV 得到的 $2m$ 个子模型,构建投票聚合模型。在生成新样本时引入了生成样本与原样本的的差异度量 FID 得分,以筛选出更好的新生成样本。实验结果表明,基于 GAN 的数据增强的 SDP 投票聚合模型具有更佳的预测性能和稳定性。

3 组块正则化交叉验证

Wang 等[30]最早为模型比较而设计了组块正则化交叉验证(简记为 $m\times$2BCV)方法,在其设计理念中贯穿了模型训练与验证同等重要、训练集与验证集分布要一致的思想。$m\times$2BCV 方法的特点是在给定数据集上不使用随机切分,而用带有约束的切分,实施 m 次 2 折交叉验证。

具体来说,假定一个包含 n 个样本点的二分类数据集为 $D_n=\{x_i,y_i\}(i=1,2,\cdots,n)$,其中,$x_i$ 为第 i 个样本点的特征,y_i 为第 i 个样本点的类别,$y_i\in\{0,1\}$。将数据集 D_n 进行 m 次对半切分为训练集与验证集,记为 $\{(D_i^{(t)},D_i^{(v)}),i=1,2,\cdots,m\}$;实施 m 次 2 折交叉验证,切分得到的训练集 $D_i^{(t)}$ 和验证集 $D_i^{(v)}$ 要满足以下正则化(约束)条件:

1)对于不同组切分中的任意两个训练集,重叠样本的数目应该基本相同,满足条件为 $|D_i^{(v)}\cap D_j^{(t)}|\approx 4/n(\forall i\neq j)$。

2)任意一组训练集 $D_i^{(t)}$ 和验证集 $D_i^{(v)}$ 之间的经验分布尽可能接近。

事实上,训练集与验证集分布相同是统计机器学习建模的一个基本假设。对类不平衡较为严重的 SDP 数据集,在传统的随机切分下,训练集 $D_i^{(t)}$ 和验证集 $D_i^{(v)}$ 之间的经验分布差异通常会较大,而在 $m\times$2BCV 下的带约束(正则化)切分,分布相同能够得到较好的保持。

Wang 等[30]给出了 $m\times$2BCV 数据切分的构造算法,该算法先将数据切分为多个小块,然后按一定的规则组合构造出满足上述约束条件的 m 组 2 折交叉验证。同时其也证明了 $m\times$2BCV 下的泛化误差估计具有方差最小性,更适用于数据建模中的模型比较,并给出了 m 的选取方法,推荐使用 $m>16$,本文中选取 $m=19$。

不同于传统交叉验证(如 5 折和 10 折)建模,在 $m\times$2 BCV 中,可以训练得到 $2m$ 个模型(下称子模型)。在验证集上对 $2m$ 个子模型进行验证,因此数据集中每个样本点都有 m 个预测标签;然后采用众数投票来聚合 $2m$ 个子模型的预测结果,得到该样本点的预测标签(称为投票聚合模型);最终将整个数据集上样本点的预测结果和样本点的真实类别进行比较,从而评估模型性能。理论和实验已经证明聚合模型可以提高模型的预测性能和稳定性[31]。

4 SDP 聚合模型框架和构建步骤

本文以 $m\times$2BCV 训练得到的 $2m$ 个子模型来构建一个软件缺陷预测聚合模型,其中所有子模型的算法均采用随机森林,模型总体框架如图 1 所示,主要步骤如下:

1)数据增强:$m\times$2BCV 切分得到每一组数据($D_i^{(t)}$, $D_i^{(v)}$),先对 $D_i^{(t)}$ 中的正例样本使用 GAN 生成新的正例样本,记为 $D_i^{(G)}$,为了使得新生成的 $D_i^{(G)}$ 与 $D_i^{(t)}$ 的正例样本分布更为接近,本文引入了 FID 度量,详细过程见下一章。

2)模型训练与验证:将新生成的每个 $D_i^{(G)}$ 融入到 $m\times$2BCV,增加正例,减少负例,使得最终训练集上正例和负例样本数相同,然后利用随机森林算法训练和验证模型,最终得到 $2m$ 个预测模型,而在验证集中每个样本点可以有 m 个预测结果。

3)模型评估和选择:对每个样本点,用其 m 个预测结果的众数投票的方法进行聚合,得票多者为该点最终的预测结果。具体来说,在 m 个预测结果中,预测为 $\hat{y}=1$ 的次数大于 $m/2$,则该点预测为 1;反之,若 $\hat{y}=0$ 的次数多,则预测为 0。最后根据整个数据集 D_n 上的预测结果,采用某个评价指标对模型进行评估和比较,在候选参数集中选出配置最好的一组参数,记为 OPT。

4)形成众数投票聚合模型:以上述配置最好的一组参数,使用步骤(2)中训练好的 $2m$ 个子模型,最终构建出众数投票聚合模型(记为 M_{opt}),即对一个新的软件模块,将 $2m$ 个子模型预测的最多数结果,作为聚合模型 M_{opt} 的预测结果。

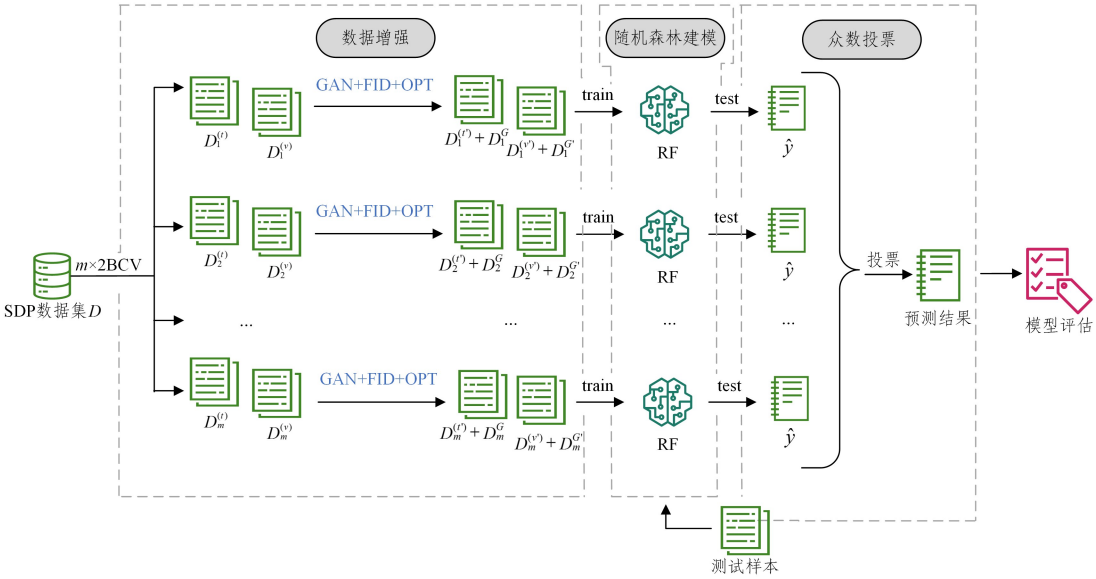

图 1 基于 GAN 数据增强的软件缺陷预测聚合模型

Fig. 1 Software defect prediction aggregation model based on GAN data enhancement

5 $m\times2$BCV 下基于 GAN 的数据增强算法

GAN 算法由生成器和鉴别器两部分组成，生成器使生成的新样本分布逼近真实样本的分布，鉴别器是一个二分类器，用来判别新样本与真实样本是否来自相同分布。训练过程重复上述步骤，通过反向传播算法更新参数，相互竞争对抗，直至生成器可以生成以假乱真的样本，使鉴别器无法判别。不同于一般的 GAN 算法，在数据生成中，本文只生成 SDP 数据集中的正例样本，缓解正例样本较少而带来的类不平衡问题。

在 $m\times2$BCV 架构下，本文基于 GAN 的数据增强的 3 种算法如下：

算法 1（记为 GAN*） 对 $m\times2$BCV 中的每一组（$D_i^{(t)}$，$D_i^{(v)}$），GAN 生成与 $D_i^{(t)}$ 中正例样本个数且分布相同的 $D_i^{(G)}$（只生成一次）并加入训练集，即构成 $D_i^{(t)}+D_i^{(G)}$，再随机抽样将 $D_i^{(t)}$ 中的负例样本个数缩减到 $D_i^{(t)}$ 中正例样本个数的两倍，记为 $D_i^{(t')}$，使得训练样本中正例与负例样本数相同，以 $D_i^{(t')}+D_i^{(G)}$ 来训练模型，以 $D_i^{(v)}$ 为验证集来评估。反过来，对 $D_i^{(v)}$ 的处理方式也一样，相应地以 $D_i^{(v')}+D_i^{(G')}$ 来训练模型，以 $D_i^{(t)}$ 为验证集来评估。

上述方法适用于缺陷率小于 1/3 的数据集，当数据集缺陷率大于或等于 1/3 时，只需将生成的 $D_i^{(G)}$ 的个数调整为与原训练集上负例个数与正例个数之差相同就可。

算法 2（记为 GAN*+FID） 与算法一基本相同，在使用 GAN 数据生成多次 $D_i^{(G)}$（本文实验中为 5 次），选与 $D_i^{(t)}$ 中正例样本的分布最接近的一组，这里的分布最接近使用 FID 来度量，详见式(1)。对于负例样本的选择也采用类似的方法，但应遵循数据增强后训练集上负例样本和正例样本数目比例为 1。

$$FID(r,g)=\|\mu_r-\mu_g\|^2+Tr(\boldsymbol{\Sigma}_r+\boldsymbol{\Sigma}_g-2(\boldsymbol{\Sigma}_r\boldsymbol{\Sigma}_g)^{1/2}) \tag{1}$$

其中，μ_r 是原样本的均值，μ_g 是新生成样本的均值，$\boldsymbol{\Sigma}_r$ 是原样本的协方差矩阵，Σ_g 是新生成样本的协方差矩阵。

事实上，许多学者发现 GAN 生成的样本波动性较大[32-33]，易造成模型预测性能不稳定，因此，本文引入 FID 得分来衡量新生成样本的分布与原样本分布的相似度，并选择使用 FID 最小的一组正例样本。起初 FID 在图像识别领域被用来衡量生成图片和真实图片的距离，之后也被用来作为 GAN 生成图片质量的一种评价标准[34]。

算法 3（记为 GAN*+FID+OPT） 算法 3 在算法 2 基础上增加了超参数（本文的超参数特指采样率）调优过程（最优采样率记为 OPT）。具体来说，先使用在 $m\times2$BCV 中少量的切分（比如 3×2BCV 下以微平均 F1 为指标）做采样率的优选，以最优采样率做完整的 $m\times2$BCV（$m=19$），再构建软件缺陷预测聚合模型。这里的采样率是指新生成的正例样本与原正例样本数目的比值，本文选用的采样率候选集合为{1.2，1.3，1.4，1.5，1.6，1.7，1.8，1.9，2.0，2.1，2.2}。

算法 1 和算法 2 中对应的采样率（数据集缺陷率小于 1/3）均为 2.0。事实上，研究发现模型的预测性能与新生成正例样本的个数有关[35]，增加采样率的优选过程可以进一步验证不同的采样率对 SDP 模型的 F1 指标的影响。

6 实验结果与分析

本章对第 5 章中提出的 3 种数据增强算法的有效性进行实验验证，验证所提出的 GAN*，GAN*+FID，GAN*+FID+OPT 这 3 种数据增强算法下，基于 $m\times2$BCV 得到的 SDP 聚合模型的预测性能和稳定性是否可以得到提升？为此，本文选取了 3 种基线重采样方法（ROS，SMOTE，RUS）进行对比。

6.1 数据集

本文实验选用了 PROMISE 数据库下广泛使用的 20 个数据集[36]，每个数据集有 20 个静态特征（C&K 度量），度量的详细信息如表 1 所列。在选取过程中，本文剔除了缺陷率在 50%以上的数据集，选用的 20 个数据集的软件模块数量范围在 195～965 之间，缺陷率范围为 8.97%～48.19%，数据集的详细统计信息如表 2 所列。

为了适应 GAN 数据生成，本文先对这 20 个数据集的

特征进行预处理，即对每个特征先做 $\log(X+1)$ 变换，然后对每一个特征除以其样本标准差。

表 2 选用的 PROMISE 数据库中的 20 个数据集的基本信息

Table 2 Basic statistics of twenty software defect datasets in PROMISE

项目	版本	模块数	缺陷模块数	缺陷率%	项目	版本	模块数	缺陷模块数	缺陷率%
tomcat		857	77	8.97	jedit	4.0	306	75	24.51
prop	6	660	66	10.00	jedit	4.1	312	79	25.32
ivy	2.0	352	40	11.36	synapse	1.1	222	60	27.03
jedit	4.2	367	48	13.08	jedit	3.2	272	90	33.09
xalan	2.4	723	110	15.21	synapse	1.2	256	86	33.59
xerces	1.3	453	69	15.23	velocity	1.6	229	78	34.06
xerces	1.2	440	71	16.14	camel	1.2	608	216	35.53
camel	1.4	872	145	16.63	xalan	2.6	885	411	46.44
camel	1.6	965	188	19.48	lucene	2.0	195	91	46.67
ant	1.7	745	166	22.28	xalan	2.5	803	387	48.19

6.2 模型预测性能评价指标

本文使用 F1 值和 AUC 来度量软件缺陷预测模型的性能，F1 值的定义如下：

$$F1=\frac{2\times P\times R}{P+R} \tag{2}$$

其中，准确率 P 为预测正确的有缺陷模块数目与所有预测为有缺陷模块数目的比值；召回率 R 为预测正确的有缺陷模块数目与原有的有缺陷模块数目的比值。

AUC 为 ROC 曲线下与坐标轴围成的面积，它量化了分类器的分类能力。若 $AUC=0.5$，代表该分类器的性能与随机猜测无异，AUC 越接近 1，代表该分类器的性能越好。

除此之外，本文还使用了 F1 值估计的后验置信区间来评估模型预测性能的稳定性[16]。置信区间长度越短，则模型预测性能的稳定性越好。F1 值估计的后验置信区间及长度的计算式如下：

$$F1_{\alpha}=\frac{1}{1+\frac{1}{2}Be'_{1-\alpha}} \tag{3}$$

$$CI=[F1_{\alpha/2},F1_{1-\alpha/2}] \tag{4}$$

$$CIL=F1_{1-\alpha/2}-F1_{\alpha/2} \tag{5}$$

其中，$Be'_{1-\alpha}$ 表示 Beta-prime 分布；CI(Confidence interval)表示 $F1$ 的后验置信区间，CIL 表示 $F1$ 值的后验置信区间的长度；α 为置信水平，本文实验中选取 $\alpha=0.05$。

6.3 实验设置

本文实验都在 $m=19$ 的组块正则化交叉验证下进行模型比较，分类器采用随机森林。随机森林和 3 种基线重采样方法(ROS，SMOTE，RUS)分别来自 scikit-learn 和 imlearn 包，均采用缺省参数。

GAN 中的生成器和鉴别器使用了全连接层、正则化层和 LeakRelu 激活函数，在鉴别器网络的最后一层使用 Sigmoid 激活函数，训练中的 $batch\_size=6$，$epoch=30$。

本文使用 Linux 环境下的 Pytorch 深度学习框架，Python 版本为 3.7，Cuda 版本为 11.4，采用 NVIDIA Tesla V100S-PCIE-32GB 进行加速。

6.4 结果对比分析

6.4.1 GAN* 可以提高 SDP 模型的 F1 值及稳定性

本节给出了算法 1(GAN*)与 ROS，SMOTE，RUS 这 3 种重采样的对比结果，表 3 中加粗的数据表示最优结果。从实验结果中可以得出：

1)相比 ROS，SMOTE，RUS，在表 3 的 20 个数据集中，有 12 个数据集在 GAN* 下的 F1 值达到最优，且在 20 个数据集上的 GAN* 的 F1 平均值为 54.9%，分别比 ROS，SMOTE，RUS 方法高出 8%，3.5%，1.2%，说明 GAN* 对提高模型的 F1 值是有效的。

2)在表 3 中，GAN* 的 F1 置信区间长度最短的次数为 15 次，明显优于 3 个基线方法。GAN* 的 F1 置信区间平均长度为 0.159，相比 3 个基线中最好的 RUS 的平均长度缩短了 0.7%，说明 GAN* 对提高 F1 值的稳定性也是有效的。

3)从表 4 中可以看出，20 个数据集上 GAN* 的 AUC 与 ROS，SMOTE，RUS 基本没有差别。

表 3 6 种数据增强方法在 20 个数据集上的 F1 值及置信区间长度

Table 3 F1 values and confidence interval lengths of six methods on twenty datasets

数据集	ROS		SMOTE		RUS		GAN*		GAN*+FID		GAN*+FID+OPT	
	F1	*CI-L*	*F1*	*CI-L*	*F1*	*CI-L*	*F1*	*CI-L*	*F1*	*CI-L*	*F1*	*CI-L*
ant-1.7	0.543	0.140	0.594	0.128	0.588	0.124	0.605	0.122	0.614	0.119	**0.617**	**0.111**
camel-1.2	0.462	0.133	0.501	0.126	0.522	0.121	0.468	0.131	0.472	0.133	**0.536**	**0.116**
camel-1.4	0.237	0.164	0.300	0.162	0.396	0.148	0.424	0.142	**0.454**	0.136	0.437	**0.130**
camel-1.6	0.237	0.140	0.347	0.138	0.377	0.131	0.390	0.129	0.415	0.124	**0.454**	**0.108**
ivy-2.0	0.321	0.299	0.317	0.283	0.410	0.264	0.443	**0.214**	**0.463**	0.226	0.456	0.217
jedit-3.2	0.647	0.167	**0.685**	**0.153**	0.681	0.154	0.674	0.159	0.659	0.163	0.675	0.156
jedit-4.0	0.541	0.200	0.571	0.192	0.583	0.189	0.593	0.186	0.605	0.186	**0.687**	**0.172**
jedit-4.1	0.604	0.189	0.631	0.179	0.647	0.170	0.667	0.170	**0.671**	0.171	0.658	**0.167**
jedit-4.2	0.358	0.280	**0.523**	0.246	0.485	0.230	0.504	0.211	0.492	0.210	0.508	**0.208**
lucene-2.0	0.652	0.158	0.629	0.165	0.651	0.163	0.644	0.161	**0.664**	**0.152**	0.663	0.154
prop-6	0.312	0.220	0.301	0.190	0.343	0.197	0.385	**0.165**	0.386	0.169	**0.391**	**0.165**
synapse-1.1	0.574	0.225	0.602	0.209	0.625	0.200	0.613	0.209	0.627	0.212	**0.635**	**0.193**

（续表）

数据集	ROS		SMOTE		RUS		GAN*		GAN*+FID		GAN*+FID+OPT	
	F1	*CI-L*	*F1*	*CI-L*	*F1*	*CI-L*	*F1*	*CI-L*	*F1*	*CI-L*	*F1*	*CI-L*
synapse-1.2	0.646	0.172	0.690	0.157	0.685	0.152	0.637	0.173	0.675	0.164	**0.694**	**0.156**
tomcat	0.206	0.212	0.259	0.204	0.360	0.186	0.406	**0.156**	0.410	0.157	**0.412**	0.157
velocity-1.6	0.574	0.195	0.614	0.179	**0.640**	**0.177**	0.574	0.195	0.614	0.185	0.620	0.179
xalan-2.4	0.243	0.184	0.341	0.177	0.384	0.161	**0.484**	0.144	0.471	0.145	0.463	**0.142**
xalan-2.5	0.693	0.075	0.703	0.074	0.704	0.073	**0.713**	**0.072**	0.711	0.073	0.711	0.073
xalan-2.6	0.743	0.069	0.740	0.069	0.749	0.068	0.747	0.068	**0.753**	**0.064**	**0.753**	0.067
xerces-1.2	0.340	0.231	0.459	0.214	0.388	0.205	0.494	0.186	0.491	**0.183**	**0.509**	**0.183**
xerces-1.3	0.451	0.233	0.478	0.221	0.530	0.202	0.508	0.180	**0.533**	0.181	**0.533**	**0.179**
平均值	0.469	0.184	0.514	0.173	0.537	0.166	0.549	0.159	0.559	0.157	**0.571**	**0.151**
占优次数	0	0	2	1	1	1	2	4	6	3	**11**	**13**

表 4　6 种数据增强方法在 20 个数据集上的 AUC

Table 4　AUC values of six methods on twenty datasets

数据集	ROS	SMOTE	RUS	GAN*	GAN*+FID	GAN*+FID+OPT
ant-1.7	0.824	**0.825**	0.815	0.822	0.818	0.817
camel-1.2	0.659	0.650	0.664	0.662	0.656	**0.667**
camel-1.4	0.707	0.705	**0.714**	0.709	0.711	0.706
camel-1.6	0.708	0.717	**0.720**	0.708	0.719	0.712
ivy-2.0	0.771	**0.785**	0.774	0.784	0.782	**0.785**
jedit-3.2	0.830	0.836	0.840	**0.845**	0.836	0.839
jedit-4.0	0.797	0.799	**0.805**	0.796	0.795	0.798
jedit-4.1	0.807	0.816	0.814	0.818	**0.820**	0.818
jedit-4.2	0.820	0.830	0.833	0.834	**0.840**	0.835
lucene-2.0	0.742	0.736	0.740	0.728	0.741	**0.746**
prop-6	**0.708**	0.701	0.688	0.697	0.698	0.670
synapse-1.1	0.767	**0.768**	0.763	0.764	0.765	0.764
synapse-1.2	**0.811**	0.802	0.802	0.807	0.810	0.800
tomcat	0.808	0.802	0.803	0.801	**0.810**	**0.810**
velocity-1.6	0.754	**0.772**	0.762	0.763	0.768	**0.772**
xalan-2.4	**0.807**	0.786	0.796	0.789	0.795	0.792
xalan-2.5	0.762	0.759	0.758	**0.766**	0.760	0.762
xalan-2.6	0.844	0.846	0.845	0.844	0.847	**0.849**
xerces-1.2	**0.774**	0.763	**0.774**	0.762	0.766	0.767
xerces-1.3	0.830	0.825	0.833	**0.840**	0.826	0.825
平均值	0.7765	0.7761	0.7771	0.7769	**0.7781**	0.7767
占优次数	4	4	4	2	3	**5**

6.4.2　GAN*+FID 可以进一步提高 F1 值

在 GAN* 的基础上，采用多次生成数据，以与原样本数据的分布差异（以 FID 来度量）最小的那组数据作为最终使用的数据来进行数据增强的算法 2，简记为 GAN*+FID。从表 3 中可以看出，GAN*+FID 能够进一步提高模型的 F1 值，20 个数据集中有 15 个数据集的 F1 值高于 GAN*；20 个数据集上 GAN*+FID 的 F1 置信区间长度有 10 个小于或等于 GAN*，说明两者的稳定性相差不大。

6.4.3　GAN*+FID+OPT 可进一步提高 F1 值及稳定性

在引入 FID 的基础上进一步引入最优采样率（OPT）的算法 3，简记为 GAN*+FID+OPT。本文选用的采样率候选集合为{1.2，1.3，1.4，1.5，1.6，1.7，1.8，1.9，2.0，2.1，2.2}，表 5 列出了 20 个数据集中以 3×2BCV 选出的 OPT，从实验结果可以得出：

1）GAN*+FID+OPT 和 GAN*+FID 的 F1 值对比：从表 3 中可以看出，20 个数据集上 GAN*+FID+OPT 的 F1 值较大的次数为 14 次，GAN*+FID 较大的只有 5 次，GAN*+FID+OPT 具有明显优势；从平均值来看，GAN*+FID+OPT 高出 GAN*+FID 1.2%。

2）GAN*+FID+OPT 与 GAN*+FID 的 F1 的置信区间长度对比：20 个数据集上，GAN*+FID+OPT 长度较短的为 15 个，GAN*+FID 较短的为 5 个，20 个数据集上的平均长度比 GAN*+FID 的缩短了 0.6%。说明 GAN*+FID+OPT 的稳定性也有所提高。

3）3 种算法的 AUC 对比：从表 4 中可以看出，GAN*+FID+OPT，GAN*+FID，GAN* 在 20 个数据集上取得 AUC 最高的次数分别为 6，7，8，AUC 平均值相差不大。总体来说，3 种算法下的 AUC 指标基本没有差别。

表 5　数据集的最优采样率

Table 5　Optimal sampling rate of datasets

数据集	OPT	数据集	OPT
ant-1.7	1.4	prop-6	1.8
camel-1.2	1.3	synapse-1.1	1.7
camel-1.4	1.7	synapse-1.2	1.5
camel-1.6	1.3	tomcat	2.0
ivy-2.0	1.7	velocity-1.6	1.6
jedit-3.2	1.6	xalan-2.4	1.8
jedit-4.0	1.3	xalan-2.5	#
jedit-4.1	1.3	xalan-2.6	#
jedit-4.2	2.1	xerces-1.2	1.9
lucene-2.0	#	xerces-1.3	2.0

注：# 代表该数据集缺陷率超出了采样率候选集的调节范围，因此这 3 个数据集的处理方法和算法 2 一致。

6.4.4　GAN*+FID+OPT 明显优于基线模型

1）在表 3 中，GAN*+FID+OPT 和 ROS，SMOTE，RUS 进行对比的结果表明：20 个数据集上的 GAN*+FID+OPT 取 F1 值最大的次数为 17，F1 值的置信区间长度最短的次数为 18，明显优于 3 种基线模型；从 F1 的平均值来看，

GAN* +FID+OPT 相比 ROS,SMOTE,RUS 的 F1 平均值分别提高了 10.2%,5.7%,3.4%,F1 的置信区间的平均长度缩短了 3.3%,2.2%,1.5%,均优于基线模型。

2)从表 4 中可以看出,GAN* +FID+OPT 上的 AUC 平均值高于 ROS 和 SMOTE,略低于 RUS;GAN* +FID+OPT 上 AUC 占优次数略多于 3 种基线模型,总的来说无明显差异。

6.4.5 实验耗时对比

在 6.3 节详细说明的环境下,在 19×2BCV 框架下和 20 个 SDP 数据集上,4 种算法的耗时如表 6 所列。

表 6 19×2BCV 下 4 种算法的耗时

Table 6 Time consumed by four algorithms under 19×2BCV

算法	消耗时间/s
GAN*	1092
ROS	208
SMOTE	225
RUS	242

从表 6 可以看出,在 20 个数据集上,GAN* 算法的总耗时分别是 ROS,SMOTE,RUS 的 5.25 倍,4.85 倍,4.51 倍,GAN* 在 20 个数据集上的 F1 最优次数、F1 平均值、F1 稳定性全面领先于 ROS,SMOTE,RUS,同时,其消耗的时间成本也在可接受范围之内;若在 5 次生成中选取最小的 FID,则 GAN* +FID 实验耗时约为 5 * 1092=5460 s=1.52 h;若在候选集的 11 个候选采样率中选取最优的 OPT,则 GAN* +FID+OPT 实验耗时约为 11 * 5 * 1092=60060 s=16.68 h。总体而言,由于大多 SDP 的数据集样本数量较少,因此总耗时在可接受范围之内。

6.4.6 m×2BCV 优于 5 折和 10 折交叉验证

本文取 $m=19$,将 m×2BCV 与 5 折交叉验证(5-fold-CV)、10 折交叉验证(10-foldCV)在算法一下得到的聚合模型进行评估,实验结果如表 7 所列。

表 7 3 种交叉验证下的 F1 值及置信区间长度

Table 7 F1 value and confidence interval length under three kinds of cross validation

数据集	19×2BCV		5-fold CV		10-fold CV	
	F1	*CI-L*	*F1*	*CI-L*	*F1*	*CI-L*
ant-1.7	**0.605**	0.122	0.599	0.123	0.593	**0.121**
camel-1.2	**0.468**	0.131	0.393	0.131	0.459	**0.128**
camel-1.4	0.424	0.142	0.400	**0.134**	**0.425**	0.138
camel-1.6	0.390	0.129	0.389	0.125	**0.393**	**0.124**
ivy-2.0	**0.443**	**0.214**	0.411	0.223	0.377	0.228
jedit-3.2	**0.674**	**0.159**	0.663	0.164	0.663	0.164
jedit-4.0	**0.593**	**0.186**	0.526	0.187	0.573	0.187
jedit-4.1	**0.667**	0.170	0.633	0.174	0.667	**0.169**
jedit-4.2	0.504	0.211	**0.516**	**0.209**	0.444	0.219
lucene-2.0	0.644	0.161	**0.674**	**0.156**	0.663	0.158
prop-6	**0.385**	0.165	0.238	0.163	0.341	**0.158**
synapse-1.1	0.613	0.209	0.553	0207	**0.632**	**0.203**
synapse-1.2	0.637	0.173	**0.659**	**0.166**	0.642	0.170
tomcat	**0.406**	0.156	0.392	**0.154**	0.403	0.155
velocity-1.6	0.574	0.195	0.605	**0.181**	**0.620**	0.185
xalan-2.4	**0.484**	0.144	0.482	**0.143**	0.440	0.144
xalan-2.5	**0.713**	**0.072**	0.699	0.074	0.702	0.074
xalan-2.6	0.747	0.068	**0.755**	**0.066**	0.749	0.068
xerces-1.2	**0.494**	0.186	0.413	0.178	0.464	**0.184**
xerces-1.3	**0.508**	0.180	0.503	**0.178**	0.473	0.183
平均值	**0.549**	0.159	0.525	**0.157**	0.536	0.158

在 20 个数据集的 F1 值上,19×2BCV 有 15 个优于 5-foldCV 的结果,有 13 个优于 10-foldCV 的结果,3 种交叉验证的 F1 置信区间平均长度差异不大。因此基于 m×2BCV 投票聚合模型优于 5 折和 10 折交叉验证的模型。

总体来说,从模型的 F1 值以及稳定性角度来看,GAN* +FID+OPT 优于 GAN* +FID,GAN* +FID 优于 GAN*,3 种方式均优于传统采样的 ROS,SMOTE,RUS;从模型的 AUC 指标来看,3 种数据增强方式没有明显差异,且与传统的 ROS,SMOTE,RUS 相比也没有明显优势;从采用交叉验证方案来看,m×2BCV 投票聚合模型优于常用的 5 折和 10 折交叉验证。

结束语 本文针对基于静态特征的软件缺陷预测建模任务,在正例样本过少的情况下,提出了一种新的数据增强方法(GAN* +FID+OPT),并通过实验验证了该数据增强方法可以提高模型的预测性能以及稳定性。此外,实验也验证了本文提出的基于 m×2BCV 的投票聚合模型优于传统的 5 折和 10 折交叉验证建模方法。

今后可以将 m×2BCV 投票聚合模型拓展到跨项目软件缺陷预测和跨版本软件缺陷预测模型,并进一步开展新的数据增强方法来处理较为严重的类不平衡数据。

参 考 文 献

[1] LI L, REN Z K, SHI K X, et al. Cost Sensitive Boosting Software Defect Prediction Method [J]. Computer Engineering, 2022, 48(3): 175-180.

[2] LI Z, JING X Y, ZHU X, et al. Progress on approaches to software defect prediction[J]. IET Software, 2018, 12(3): 161-175.

[3] YU Q, JIANG S J. The Impact Study of Class Imbalance on the Performance of Software DefectPrediction Models [J]. Chinese Journal of Computers, 2018, 4: 809-824.

[4] SONG Q, GUO Y, SHEPPERD M, et al. A comprehensive investigation of the role of imbalanced learning for software defect prediction [J]. IEEE Transactions on Software Engineering, 2018, 45(12): 1253-1269.

[5] MALHOTRA R, KAMAL S. An empirical study to investigate oversampling methods for improving software defect prediction using imbalanced data[J]. Neurocomputing, 2019, 343: 120-140.

[6] NEZHADSHOKOUHI M M, MAJIDI M A, RASOOLZADEGAN A, et al. Software defect prediction using over-sampling and feature extraction based on Mahalanobis distance[J]. The Journal of Supercomputing, 2020, 76(1): 602-635.

[7] PAK C, WANG T, SU X, et al. An empirical study on software defect prediction using oversampling by smote[J]. International Journal of Software Engineering and Knowledge Engineering, 2018, 28(6): 811- 830.

[8] GOYAL S. Handling class-imbalance with KNN(neighbourhood) under-sampling for software defect prediction[J]. Artificial Intelligence Review, 2022, 55(3): 2023-2064.

[9] HAN H, WANG W Y, MAO B H, et al. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning [C]// International Conference on Intelligent Computing. Berlin: Springer, 2005: 878-887.

[10] LIU X Y, WU J, ZHOU Z H, et al. Exploratory undersampling for class-imbalance learning[J]. IEEE Transactions on Systems,

Man,and Cybernetics,Part B(Cybernetics),2008,39(2):539-550.

[11] KONNO T,IWAZUME M. Pseudo-feature generation for imbalanced data analysis in deep learning[C]// CoRR. 2018.

[12] GOODFELLOW I,POUGET-ABADIE J,MIRZA M,et al. Generative adversarial networks[J]. Communications of the ACM,2020,63(11):139-144.

[13] LI Z. Imbalanced Data Enhancement Algorithm Based on GAN and Its Application Research [D]. Beijing:Beijing Jiaotong University,2019.

[14] WANG R,WANG Y,LI J,et al. Block-regularized $m\times 2$ cross-validated estimator of the generalization error[J]. Neural Computation,2017,29(2):519-554.

[15] XUE Y. Confidence in Comparing Two Models with F1 MeasureBased on Block-regularized $m\times 2$ Cross Validation[C]// Proceedings of the AAAI Conference on Artificial Intelligence. 2023:1-8.

[16] WANG R. Research on Block-regularized Cross - Validation Methods for Comparing Supervised Algorithms [D]. Taiyuan: Shanxi University,2019.

[17] ALSAEEDI A,KHAN M Z. Software defect prediction using supervised machine learning and ensemble techniques:a comparative study[J]. Journal of Software Engineering and Applications,2019,12(5):85-100.

[18] WANG Y,LI J,LI Y,et al. Confidence Interval for F1 Measure of Algorithm Performance Based on Blocked 3×2 Cross-validation [J]. IEEE Transactions on Knowledge and Data Engineering,2014,27(3):651-659.

[19] HOSSEINI S,TURHAN B,GUNARATHNA D,et al. A systematic literature review and meta-analysis on cross project defect prediction[J]. IEEE Transactions on Software Engineering,2017,45(2):111-147.

[20] MENG F,CHENG W,WANG J,et al. Semi-supervised software defect prediction model based on tri-training[J]. KSII Transactions on Internet and Information Systems (TIIS), 2021,15(11):4028-4042.

[21] WANG K,LIU L,YUAN C,et al. Software defect prediction model based on LASSO-SVM[J]. Neural Computing and Applications,2021,33(14):8249-8259.

[22] MALOHTRA R,YADAV H S. An improved CNN-based architecture forwithin-project software defect prediction[M]// Soft Computing and Signal Processing. Springer, Singapore, 2021: 335-349.

[23] IBRAHIM D R,GHNEMAT R,HUDAIB A,et al. Software defect prediction using feature selection and random forest algorithm[C]// 2017 International Conference on New Trends in Computing Sciences(ICTCS). IEEE,2017:252-257.

[24] TANTITHAMTHAVORN C,HASSAN A E,MATSUMOTO K,et al. The impact of class rebalancing techniques on the performance and interpretation of defect prediction models[J]. IEEE Transactions on Software Engineering, 2018, 46(11): 1200-1219.

[25] HU M Y,HUANG H Y,XIANG Z H,et al. EnsembleModel for Software Defect Prediction[J]. Computer Science,2019,46(11): 176-180.

[26] ALI H,SALLEH M N M,SAEDUDIN R,et al. Imbalance class problems in data mining: a review[J]. Indonesian Journal of Electrical Engineering and Computer Science,2019,14(3):1560-1571.

[27] LEEVY J L,KHOSHGOFTAAR T M,BAUDER R A,et al. A survey on addressing high-class imbalance in big data[J]. Journal of Big Data,2018,5(1):1-30.

[28] QIAN Y,QIAN X M,GUAN Y,et al. A Cross-Project Defect Prediction Method Using Adversarial Learning[J]. Journal of Software2022,33(6):2097-2112.

[29] SHENG L,LU L,LIN J,et al. An adversarial discriminative convolutional neural network for cross-project defect prediction [J]. IEEE Access,2020,8:55241-55253.

[30] WANG R,LI J,YANG X,et al. Block-regularized repeated learning-testing for estimating generalization error[J]. Information Sciences,2019,477:246-264.

[31] YANG X,WANG Y,WANG R,et al. Ensemble Feature Selection With Block-Regularized mx2 Cross-Validation[J]. IEEE Transactions on Neural Networks and Learning Systems,2023, 34(9):6628-6641.

[32] ARJOVSKY M,BOTTOU L. Towards principled methods for training generative adversarial networks[J]. arXiv:1701.04862, 2017.

[33] LEI K,MARDANI M,PAULY J M,et al. Wasserstein GANs for MR imaging: from paired to unpaired training[J]. IEEE Transactions on Medical Imaging,2020,40(1):105-115.

[34] OBUKHOV A,KRASNYANSKIY M. Quality assessment method for GAN based on modified metrics inception score and Fréchet inception distance[C]// Proceedings of the Computational Methods in Systems and Software. Cham:Springer,2020: 102-114.

[35] DEL RIO S,BENÍTEZ J M,HERRERA F,et al. Analysis of data preprocessing increasing the oversampling ratio for extremely imbalanced big data classification[C]// 2015 IEEE Trustcom/BigDataSE/ISPA. IEEE,2015:180-185.

[36] WANG S,LIU T,TAN L,et al. Automatically learning semantic features for defect prediction[C]// 2016 IEEE/ACM 38th International Conference on Software Engineering(ICSE). IEEE, 2016:297-308.

XU Jinpeng, born in 1998, postgraduate. His main research interests include software defect prediction and deep learning.

LI Jihong, born in 1964, Ph. D, professor, Ph. D supervisor, is a member of China Computer Federation. His main research interests include deep learning, natural language processing and software defect prediction.

(责任编辑:何杨)