doi:10.11835/j.issn.1000-582X.2025.02.002

基于机器学习的软件缺陷预测研究

喻 皓[1],张 莹[2],李 倩[3],姜立标[4a,5],尚云鹏[4b]
(1.广汽埃安新能源汽车股份有限公司研发中心,广州 511400; 2.星河智联汽车科技有限公司,广州 510335;
3.工业和信息化部电子第五研究所,广州 510463; 4.广州城市理工学院 a.机械工程学院与机器人学院;
b.工程研究院,广州 510800; 5.华南理工大学 机械与汽车工程学院,广州 510641)

摘要:在机器学习技术逐渐渗透到各个领域的背景下,软件开发流程中的软件测试非常重要,面对在软件缺陷预测过程中出现的类别不平衡和准确性问题,提出一种基于监督学习的解决方案,采用样本平衡技术,结合合成少数类过采样技术(synthetic minority over-sampling technique,SMOTE)与编辑最近邻(edited nearest neighbor,ENN)算法,对局部加权学习(local weight learning,LWL)、J48、C4.8、随机森林、贝叶斯网络(Bayes net,BN)、多层前馈神经网络(multilayer feedforward neural network,MFNN)、支持向量机(supported vector machine,SVM)以及朴素贝叶斯(naive Bayes key,NB-K)等多种算法进行测试。这些算法被应用于NASA数据库的3个不同数据集(KK1, KK3, PK2),并对其效果进行详细比较分析。研究结果显示,结合了SMOTE和ENN的随机森林模型在处理类别不平衡问题方面展现出高效且避免过拟合的优势,为解决软件缺陷预测中的类别不平衡提供了一种有效的解决方案。

关键词:软件缺陷预测;机器学习;类不平衡;XGBoost;随机森林
中图分类号:TP391 **文献标志码:**A **文章编号:**1000-582X(2025)02-010-12

Research on software defect prediction based on machine learning

YU Hao[1], *ZHANG Ying*[2], *LI Qian*[3], *JIANG Libiao*[4a,5], *SHANG Yunpeng*[4b]
(1. GAC Aion New Energy Automobile Co., Ltd., Guangzhou 511400, P. R. China; 2. Syncore Autotech Co., Ltd., Guangzhou 510335, P. R. China; 3. The Fifth Research Institute of Electronics, Ministry of Industry and Information Technology, Guangzhou 510463, P. R. China; 4a. School of Mechanical Engineering and Robotics; 4b. Institute of Engineering Research, Guangzhou City University of Technology, Guangzhou 510800, P. R. China; 5. School of Mechanical & Automotive Engineering, South China University of Technology, Guangzhou 510641, P. R. China)

Abstract: With the gradual penetration of machine learning technology into various fields, software testing in the software development process is very important. Software defect prediction faces class imbalance problem and accuracy issue. This paper proposes a supervised learning-based software prediction method for solving these two core problems. The method adopts sample balancing technique, combined with synthetic minority over-sampling technique(SMOTE) and edited nearest neighbor(ENN) algorithm, to test local weight learning(LWL), J48, C4.8,

收稿日期:2024-04-20
基金项目:国家自然科学基金(61602345)。
Supported by National Natural Science Foundation of China(61602345).
作者简介:喻皓(1983—)男,高级工程师,主要从事电机设计方向研究,(E-mail)yuhao@gacne.com.cn。
通信作者:李倩,女,高级工程师,(E-mail)lq@ceprei.biz。

random forest, Bayes net(BN), multilayer feedforward neural network(MFNN), supported vector machine(SVM), and naive Bayes key(NB-K). These algorithms are applied to three different datasets (KK1, KK3 and PK2) in the NASA database and their effects are compared and analyzed in detail. The results show that the random forest model combining SMOTE and ENN exhibits high efficiency and avoiding overfitting in dealing with class imbalance problems, which provides an effective way to solve the problem in software defect prediction.
Keywords: software defect prediction; machine learning; class imbalance; XGBoost; random forest

软件开发需要大量人员协同完成,由于人员的工作背景、工作能力、思维方式等都存在差异[1],造成开发中潜在和不可预测的错误,即存在软件缺陷。软件缺陷包括:错误(error)、故障(fault)、失效(failure)等[2]。

软件缺陷预测模型旨在依据软件历史缺陷数据预测未来可能出现的缺陷。预测的准确性与缺陷数据的质量密切相关。软件缺陷预测在软件工程中具有重要意义:1)软件缺陷预测能有效缩短开发周期,用尽可能少的资源在最短时间开发出可靠的软件;2)优化测试资源,提高缺陷检测效率;3)提高软件质量[3]。

本项研究采用了来自美国国家航空航天局(national aeronautics and space administration,NASA)的缺陷样本集,包括KK1、KK3和PK2这3个数据集,评估8种不同的监督学习算法在解决软件缺陷预测中类不平衡问题的性能。这些算法为:局部加权学习(local weighted learning,LWL)、J48决策树、C4.5决策树、随机森林(random forest,RF)、贝叶斯信念网络(Bayesian belief network,BBN)、NB-K算法、多层前馈神经网络(multilayer feedforward neural network,MFNN)以及支持向量机(support vector machine,SVM)。研究通过一系列实验,对比这些算法在处理类不平衡问题时的表现,结果显示,随机森林算法在处理类不平衡问题方面具有更出色的效果。

1　国内外相关研究及问题

机器学习作为人工智能和数据科学的核心,是当今各项技术中发展最快的热点领域[4],目的是通过经验使学习性能有所提高[5]。目前,机器学习广泛应用于各领域,如:医疗保健、金融、零售、旅游及社交媒体等[6]。根据美国航空航天局(NASA)喷气推进实验室(jet propulsion laboratory,JPL)的研究人员2001年在《科学》杂志上发表的研究文章,机器学习在航天研究所发挥的作用日益显著。2006年,全球首个机器学习系也在美国卡内基梅隆大学成立。目前,较为热门的机器学习方法主要分为集成学习[7]、深度学习[8]、迁移学习[9]及多标记学习[10]等。在经济和社会领域,有一个著名的二八原则,即“帕累托原则”,它指出大约80%的结果往往是由20%的原因所导致。这一原则同样适用于软件缺陷预测领域,在软件开发过程中,大约80%的软件缺陷往往是由20%的关键因素引起。软件缺陷预测方面造诣很深的专家Boehm[11]提出,80%的缺陷存在于20%的软件模块之中,又称为Pareto原则。这一发现帮助软件开发团队更加高效地识别和修复潜在的软件缺陷,提高软件质量和可靠性。目前,基于机器学习的预测方法已经成为软件开发领域中不可或缺的核心技术,广泛应用于各种软件开发和维护过程,极大提高软件质量和开发效率。研究人员通过预测模型,找出这20%的模块作重点测试和研究,通过深入分析和挖掘数据中的隐含信息,找出缺陷,并总结潜在的规律和模式,为研究者们带来全新的视角和思考方式。

尽管全球的研究人员正积极致力于软件缺陷预测,并取得一定进展,但仍然存在许多挑战。主要包括:

1)类别不平衡:缺陷数据在整体数据集中的比例较低,缺陷样本与无缺陷样本的数量差异大,有时甚至存在数量级的差异;

2)预测结果的可靠性:尽管模型可能将所有选定样本预测为无缺陷的准确率高达99%,而当预测缺陷样本时,准确率仅为1%。显然,这种预测结果在实际应用中是不可信的。

3)缺陷存在对开发维护和成本预估的影响:确定软件模块中是否存在缺陷,属于1个二元分类问题。若能在软件开发周期的初级阶段辨识出含有缺陷的模块,将大幅度降低软件故障的出现频率。这有助于节省成本和提高客户满意度。因此,预测软件模块的缺陷倾向性变得尤为关键。

2 本文模型

2.1 软件缺陷的预测过程

在软件工程领域,预测软件缺陷是课题研究的关键。如图1所示,依据不同的监督机制,软件缺陷预测被分为3大类别:有监督缺陷预测、无监督缺陷预测以及半监督缺陷预测。有监督缺陷预测依赖于标注的数据集,通过模型训练识别潜在缺陷。无监督缺陷预测不依赖于标注,通过挖掘数据的内在结构识别异常或潜在缺陷。半监督缺陷预测融合了有监督和无监督的方法,利用少量的标注数据与大量的未标注数据进行缺陷预测。

半监督缺陷预测进一步细分为基于聚类的缺陷预测和基于排序的缺陷预测。基于聚类的缺陷预测方法通过将数据点分组,识别出具有相似特征的簇,发现潜在的缺陷簇。这种方法通常适用于数据集中缺陷分布不均匀的情况。基于排序的缺陷预测方法则侧重于对数据点进行排序,将最有可能包含缺陷的实例排在前面,提高预测的准确性。这种方法通常依赖于一些启发式规则或模型对数据进行排序。通过这2种方法,半监督缺陷预测能够在标签数据有限的情况下,提升缺陷检测的效率和准确性。

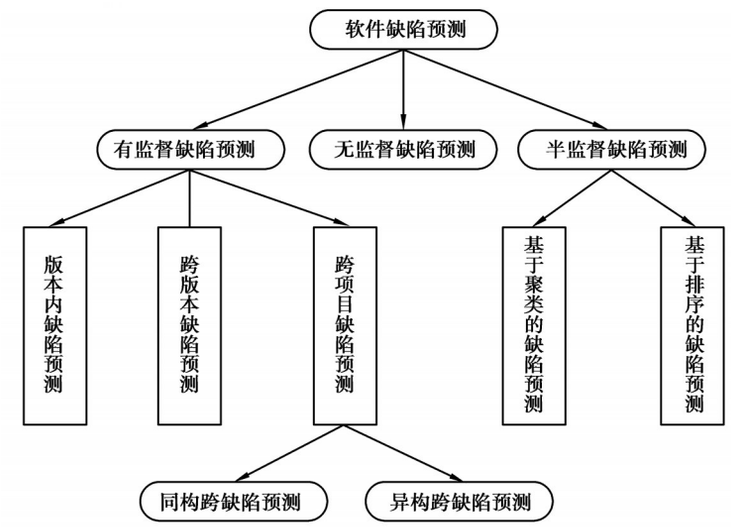

图1 软件缺陷预测研究结构图

Fig. 1 Structure diagram of software defect prediction research

软件缺陷预测流程如图2所示,该过程利用历史数据作为训练集来训练模型。在预测模型中对新数据执行测试,得出预测结果。在大多数情况下,训练集由已知类别的数据构成,而测试集则由未知类别的数据组成,这一过程属于有监督的预测方法范畴。

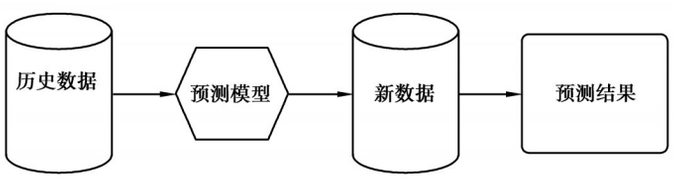

图2 软件缺陷预测过程

Fig. 2 Software defect prediction process

软件度量元在软件开发项目中不仅是衡量项目进展和质量的工具,更是预测项目未来发展趋势的关键手段。通过收集和分析各种度量数据,项目管理者可以更好地理解项目的当前状态,预测可能出现的问题,提前采取相应措施确保项目的顺利进行。这些度量元包括代码复杂度、缺陷密度、开发效率等,它们共同构成了全面的评估体系,帮助团队在软件开发过程中做出明智的决策。具体包括以下方面:

1)度量元是基于对历史代码库和开发过程中特定属性的数值进行挖掘和分析后得到的数据;

2)通过对度量元的大量整合数据进行深入分析和综合评估,有效预测软件中潜在的缺陷。度量元包括:代码复杂度、代码覆盖率、代码重复率等多种指标,这些指标能够全面反映软件的质量和稳定性。通过对这些度量元数据的收集和整合,利用统计学和机器学习方法,建立预测模型,提前发现软件中的潜在问题。这种方法不仅可以提高软件开发的效率,还能显著提升软件的质量和可靠性;

3)度量元在提升开发效率和降低项目成本方面起重要作用。通过引入度量元,开发团队能准确地评估项目进度,识别潜在风险和瓶颈,采取相应措施优化工作流程。度量元的使用使项目管理更加透明和可量化,团队成员可清晰了解自己的任务和目标,提高工作效率。此外,度量元还帮助项目管理者更好地分配资源,避免浪费。通过对项目各个阶段的详细分析,管理者可以合理安排人力和物力,确保每个环节都能高效运行,不仅缩短了开发周期,还显著降低了整体成本。

2.2　软件缺陷预测模型

绝大多数软件缺陷预测模型都是基于机器学习算法构建的。尤其是在缺陷数据充足的情况下,这些模型往往倾向于使用常规的机器学习方法。图 3 展示了基于监督学习预测模型的具体实例。

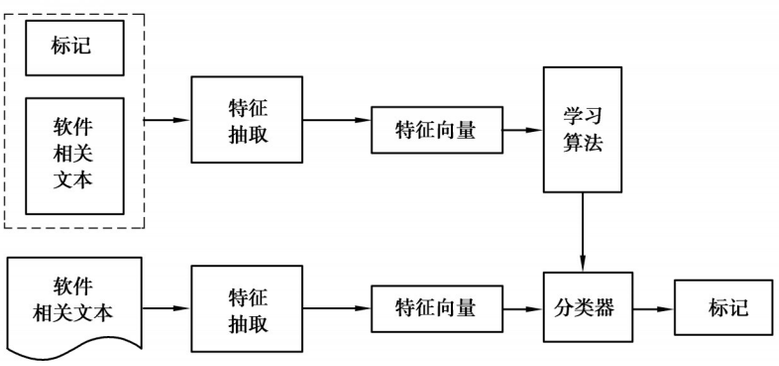

图 3　基于监督学习的预测模型

Fig. 3　Supervised learning based prediction model

2.3　软件缺陷预测二分类和不平衡分类问题

在软件缺陷预测领域,选择合适的评价指标至关重要。由于软件缺陷预测常常面临类别不平衡等问题,一些常规指标可能无法准确反映分类器的实际效能[12]。接下来,将对软件缺陷预测中几种常用的评价指标进行阐述。

软件缺陷预测可以被视为二分类问题,预测结果分为软件中存在缺陷和不存在缺陷 2 种情况[13-16]。采用基于混淆矩阵得出的性能评估指标,如表 1 所示。混淆矩阵包含 4 种基本值:真正例数(true positive,TP)、假正例数(false positive,FP)、真反例数(true negative,TN)、假反例数(false negative,FN)。

表 1　混淆矩阵

Table 1　Confusion matrix

真实值	预测值	
	Negative	Positive
Negative	TN	FP
Positive	FN	TP

在软件缺陷预测这一领域,类不平衡问题始终是难以克服的挑战,存在缺陷的模块与无缺陷模块之间的比例极其悬殊,这种不平衡现象使分类任务变得异常复杂。在面对这种数据分布不均的情况时,大多数标准的分类算法会遇到极大困难。由于数据集中某一类别的样本数量远远超过另一类别,导致算法在学习过程中偏向于多数类,从而忽视了少数类的重要性。这种偏向性使算法难以达到预期性能,影响预测结果的准确

性和可靠性。因此,如何有效解决类不平衡问题,成为软件缺陷预测领域待解决的关键问题。

2.4 代价敏感学习

代价敏感学习是一种先进的机器学习方法,它在模型训练过程中考虑了不同错误分类所带来的不同代价。在现实世界的应用场景中,某些类型的错误分类带来的后果和代价远远高于其他类型的错误。例如,在医疗诊断领域,将一个实际有某种疾病的患者误诊为健康(即假阴性错误)的代价可能会非常高,导致患者错过最佳治疗时机,甚至危及生命。相反,将健康的人误诊为患有某种疾病(即假阳性错误)虽然也会带来一定代价,比如患者的精神压力和不必要的医疗检查,但其代价通常被认为比假阴性错误要低许多。

代价敏感学习的核心是通过为不同类型的错误分配不同权重或代价,优化模型性能,根据预先设定的代价矩阵调整模型训练过程,使模型在学习过程中更关注那些代价较高的错误类型。这种方法适用于那些错误代价不均等领域,传统的机器学习方法可能无法达到最优性能[17]。例如,在金融欺诈检测中,将欺诈行为误判为合法交易(假阴性错误)可能会给金融机构带来巨大经济损失,而将合法交易误判为欺诈(假阳性错误)虽然也会带来一定的操作成本,但其代价通常要低得多。因此,代价敏感学习可以通过为不同类型的错误分配不同的代价权重,优化模型性能,减少高代价错误的发生。同样,在疾病诊断领域,通过为不同类型的诊断错误分配不同代价,模型更加关注那些可能导致严重后果的错误,提高诊断的准确性和可靠性。此外,在网络安全领域,检测网络入侵时,将真正的入侵行为误判为正常行为(假阴性错误)可能会导致严重的安全漏洞,而将正常行为误判为入侵行为(假阳性错误)虽然会带来一定误报,但其代价通常要低得多。因此,通过代价敏感学习,可以优化入侵检测模型的性能,减少高代价错误的发生,提高整个网络安全系统的防护能力。在大多数学习任务中,样本权重通常是相等的,只有在特定任务中,样本才会有不同权重[18-22]。常见的代价敏感学习方法包括将普通分类模型转换为代价敏感模型的元代价(metacost),通过最小化期望代价修改训练样本的标签,并在新的模型上继续学习,这个过程被称为“元学习”。在元代价方法中,底层分类器作为黑盒不受影响,其优点是对具体使用的分类器没有依赖性。此外,通过改变训练集中各类别的频率反映错误分类的代价。样本分布的改变有时会影响算法性能,因此,需要采取分层抽样(stratification)来保持训练集的代表性。通过多次集中抽样训练集获得多个模型,计算测试样本中不同类别的概率,这些概率来源于多个模型的代价敏感决策[23]。

获取测试样本的分数误差代价,最终确定类标记。类标记的选择旨在最小化代价,典型的做法是采用集成学习技术提升学习效果。上述介绍的几种方法是应对分类不平衡问题的常见策略,需要在特定的预测模型中应用。同时,鉴于分类不平衡的影响,预测模型需要进行相应调整。

3 实 验

3.1 实验数据集

在实验研究中,选取了来自美国航空航天局(NASA)提供的开放数据库中的数据集[24-29],该数据库主要针对银河系外的天体进行研究。实验过程中,采用了8种不同的监督学习算法,包括局部加权学习法、J48、C4.8、随机森林、贝叶斯网络、多层前馈神经网络(multi-layer feedforward neural network,MLFNN)、支持向量机以及NB-K[30]。这些算法被应用于3个不同的数据集(KK1、KK3和PK2)上,通过训练和研究,对这些算法在处理数据集的表现进行了详细的分析和评估。

在实验过程中,特别关注了这些算法在处理不平衡分类问题上的效率和效果[31]。对这些核心比较结果,进行更细致和深入考量,更好理解每种算法在处理不平衡分类问题时的优势和局限性。有助于在未来研究中选择更适合的算法,提高数据处理的准确性和效率。

3.2 实验平台和数据选取

实验选用了几款特定的软件平台来完成任务。首先,研究使用Weka软件,其版本号为3.4。Weka是一个功能强大的数据挖掘工具,开源且免费,适合用于学习和研究。尽管它的功能数量有限,但提供了一整套全面解决方案,使用户轻松进行各种数据挖掘任务。此外,还使用MATLAB软件,版本号为R2019b。MATLAB是一个广泛应用于工程和科学领域的高性能数值计算和可视化软件。它在数据预处理和分析方

面表现出色,能处理各种复杂的数据处理任务[32-35]。为确保实验的顺利进行,配备了 1 台性能优越的笔记本电脑。这台笔记本电脑配置了 Intel Core i52.40 GHz 处理器和 16 GB 的内存,能够提供强大的计算能力和足够的内存支持,应对各种复杂的数据处理和分析任务。考虑到 NASA 数据库的高度安全性,特别选择了 Weka 作为数据挖掘工具。Weka 不仅开源且免费,而且功能全面,非常适合用于研究[36-37]。

尽管在不同数据集中,软件度量的数量存在差异,但这些度量都是针对特定方法设计的。研究综合考虑了多种因素,在编程语言、数据模块等基础上,从 NASA 数据库中精心挑选了 3 个具有代表性的测试数据集:KK1、KK3、PK2。选择这些数据集是基于对不同编程语言和数据模块的深入分析和评估,如表 2 所示,详细列出了每个数据集的特征和度量指标。

表 2　NASA 数据库中选出的测试数据集

Table 2　Test datasets selected from the nasa database

编号	DATASET	*F*	*I*	*D*	*ND*	*DR*/%	*SS*/ K
1	KK1	24	2 093	328	1 778	15.81	24
2	KK3	36	454	47	419	9.47	7
3	PK2	35	5 462	24	5 439	0.52	29

在表 2 中,度量的数量用“*F*”表示,模块的数量用“*I*”表示,存在缺陷的模块数量用“*D*”表示,无缺陷的模块数量用“*ND*”表示,数据集的缺陷率用“*DR*”表示,软件的规模大小用“*SS*”表示。

此外在机器学习领域,数据预处理是关键步骤,通常包括处理数据中的缺失值、标准化和规范化等操作。在本实验部分,鉴于使用的数据集包含数值型数据,采取以下处理顺序:数据离散化、数据降维处理、处理数据不平衡问题。数据离散化可以通过 Weka 软件的图形用户界面完成。而数据降维则涉及使用评估器 CfsSubsetEval 和搜索方法 BestFirst,这些工具的使用同样可以通过 Weka 的 GUI 指导进行。

大多数采样技术仅专注于某一特定类型的样本,这意味着要么对多数类样本进行欠采样,要么对少数类样本进行过采样。在现实应用中,样本类别不平衡和过拟合问题普遍存在。研究者们提出了结合欠采样和过采样的方法。这种方法能够克服 SMOTE 算法在处理少数类样本时由于与多数类样本重叠而导致的分类困难。数据清洗技术在这里发挥了关键作用,能够有效处理重叠样本。

3.3　实验方案

在实际应用中,处理数据不平衡问题时,有多种技术可以采用,其中 2 种较为常用且效果显著的方法分别是 SMOTE 结合 ENN 和 SMOTE 结合 Tomek Links。这 2 种方法在处理流程上具有高度相似性:先利用 SMOTE 技术对数据集进行过采样,增加少数类的样本数量,减少类别不平衡对模型性能的影响。然而,它们在后续的处理步骤中采用了不同的策略,即 SMOTE+ENN 方法使用了 ENN 算法进行下采样,而 SMOTE+Tomek Links 方法则采用了 Tomek Links 算法进行下采样。

具体来说,SMOTE+ENN 方法在处理过程中,通过 SMOTE 技术生成新的少数类样本,利用 ENN 算法对过采样后的数据集进行清洗,去除那些与最近邻多数类样本不一致的样本。这种方法在清除重叠样本方面表现出色,能够有效提高数据集质量,减少噪声和冗余数据的影响。而 SMOTE+Tomek Links 方法则通过 Tomek Links 算法识别并移除那些位于多数类和少数类边界上的样本,这些样本可能会导致分类器的决策边界模糊不清。SMOTE+Tomek Links 方法旨在提高分类器的决策边界清晰度,提升模型的分类性能。

研究者通过一系列实验验证了 SMOTE+ENN 算法在处理数据不平衡问题上的优越性。SMOTE+ENN 算法不仅能够有效提高少数类的分类精度,还能在一定程度上提高整体的分类性能。因此,在研究中,选择使用 SMOTE+ENN 算法来处理软件缺陷预测数据集的不平衡问题,获得更好的预测结果。

在研究的实验部分,选用了上述 7 种分类算法以执行数据分类测试。Chen 等[38]通过实验验证了朴素贝叶斯(Naïve Bayes,NB)分类器的优越性能。研究采用 NB-K 算法,这是一种基于核估计器的改进版朴素贝叶斯分类器,实验在 Weka 3.9 平台上执行,使用默认参数。为了确保评估结果的可靠性和准确性,采用了十折交叉验证的方法,即将数据集分为 10 份,其中 9 份用于训练模型,1 份用于测试模型性能,重复 10 次,取平

均值提高评估质量。此外,为了进一步精确评估算法性能,进行了10次十折交叉验证,取所有结果的平均值,确保评估结果的稳定性和可靠性。

3.4 实验分析

通过对这些数据集进行深入分析,得到一系列有价值的结果。为更好地展示结果,将它们分别整理并展示在不同的表格中。具体来说,KK1数据集的实验结果被详细记录在表3中,KK3数据集的实验结果展示在表4中,而PK2数据集的实验结果则展示在表5中。通过对比3个表格数据,清晰看到不同数据集在实验中的表现和差异。有助于理解各个数据集的特性,为后续的数据分析和模型优化提供重要的参考依据。

表3 不同数据集(KK1)的实验结果

Table 3 Experimental results for different datasets (KK1)

数据集	算法							
	LWL	J48	C4.5	RF	BBN	MF	SMO	NB
A	0.593 1	0.599 2	0.593 3	0.603 7	0.632 3	0.612 6	0.611 8	0.622 3
T	0.648 1	0.634 2	0.606 3	0.639 7	0.628 3	0.627 6	0.622 8	0.589 3
F	0.336 1	0.382 1	0.358 2	0.323 3	0.364 7	0.391 6	0.375 8	0.374 6
P	0.626 6	0.617 2	0.687 1	0.707 3	0.668 5	0.638 4	0.615 8	0.667 7
R	0.659 8	0.635 4	0.619 4	0.668 2	0.667 3	0.613 5	0.642 1	0.651 1
f	0.614 1	0.595 8	0.616 4	0.643 5	0.662 2	0.597 8	0.561 4	0.581 7
M	0.284 3	0.286 6	0.266 3	0.327 9	0.289 7	0.243 4	0.264 9	0.279 5
U	0.656 1	0.637 3	0.722 2	0.678 3	0.710 5	0.696 9	0.609 7	0.660 5
C	0.620 6	0.582 2	0.669 5	0.682 1	0.688 5	0.626 2	0.591 1	0.647 3

尽管所用的测试集在航空航天中也有所应用,软件的可靠性很高,但是通过刘文英等[39]研究发现,在NASA原数据集中存在一些问题,比如样本重复、数据不相同等[40-41]。因此,为了确保数据的准确性和可靠性,对NASA数据集中的原始数据执行了数据清洗操作。这些操作包括:去除重复记录、填补缺失值、纠正错误的数据以及过滤掉不相关的数据点。通过这些步骤,生成经过净化的版本,命名为NASA-cleaned。为了进一步确保数据的质量和实验的准确性,选择了这个经过清洗的NASA-cleaned版本数据集进行后续实验和研究。通过使用这个经过处理的数据集,结果更加可靠,为相关领域的研究提供坚实基础。

研究将有效数字定为5位,部分数据简化处理。根据表3和图4展示的数据,在KK1数据集上,不同算法的性能差异并不显著。在软件缺陷预测领域,精确度(Precision)和召回率(Recall)是2个核心指标。观察表3可知,召回率较高的算法包括贝叶斯信念网络(BBN)、局部加权学习(LWL),其AUC值达到了0.715 2。随机森林与贝叶斯信念网络(BBN)的F-measure值非常接近,分别为0.639 5和0.646 2,均高于C4.5算法的0.610 4。MCC值揭示了算法处理不平衡分类问题的能力,表现最佳的算法是随机森林、贝叶斯信念网络(BBN)和局部加权学习(LWL),其中,随机森林的MCC值高达0.301 9,表明这3种算法在处理不平衡数据分类问题上具有很好的适应性。综合考量这些关键指标,得出结论:在KK1数据集上,随机森林算法的整体表现最为卓越。

图5展示了KK1的接收者操作特征(receiver operation creature,ROC)曲线,可以看出RandomForest、C4.5以及贝叶斯信念网络算法的性能优于其他算法。综合考虑ROC曲线和精确率-召回率(precision-recall curve, PRC)曲线(见图6),可以观察到RandomForest算法在这些算法中表现最佳,而C4.5和BBN算法的性能则较为接近。基于ROC和PRC曲线的分析结果表明,在KK1的评估中,当度量值较少而实例较多时,RandomForest算法的性能最为出色。

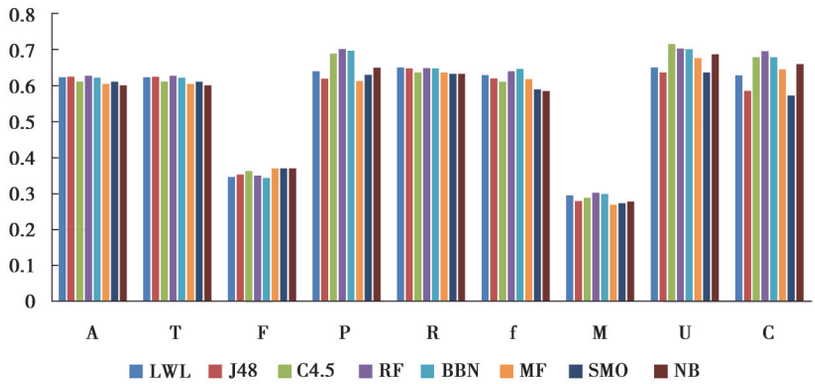

图 4　算法性能图(KK1)

Fig. 4　Algorithm performance graph (KK1)

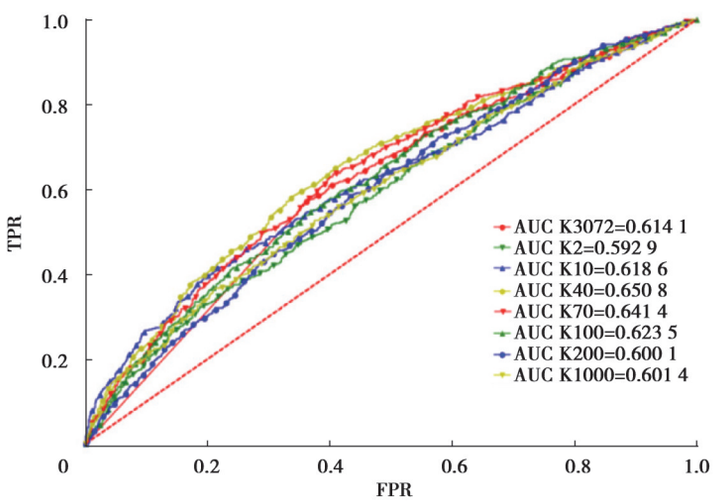

图 5　ROC 比较图(KK1)

Fig. 5　ROC comparison chart (KK1)

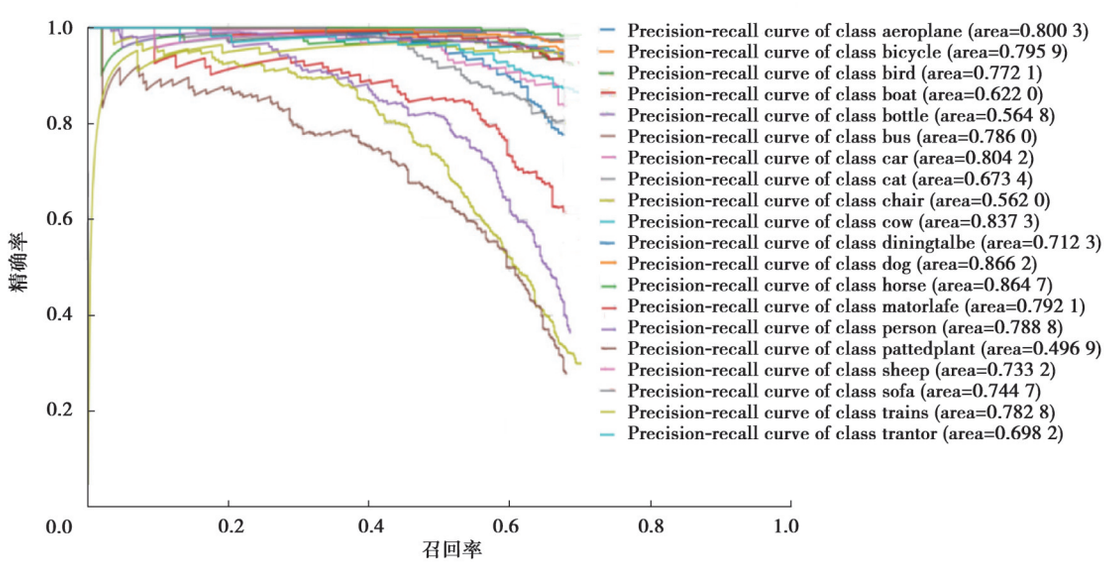

图 6　PRC 比较图(KK1)

Fig. 6　PRC comparison chart (KK1)

根据表4和图7的深入分析，研究发现KK3数据集相较于KK1数据集，在不同算法性能上的差异更显著。在所评估的8个分类算法中，NB-K、BBN和J48算法在召回率方面的表现尤为出色。特别是NB-K算法，其召回率高达0.842 1，显示出其在识别正例时的卓越能力。这意味着在这些算法中，NB-K能够更有效地捕捉到实际存在的正例，减少漏报情况。此外，这3种算法在PRC（精确率-召回率曲线）值方面也领先于其他5种算法，NB-K算法的PRC值更是达到了0.821 5。从F-measure值的分析来看，J48算法与NB-K算法在性能上非常接近，两者的F-measure值分别为0.828 0和0.826 9，均略高于BBN算法的0.825 2。MCC值（Marthius相关系数）反映了算法对不平衡分类问题的处理能力，其中NB-K、BBN和J48算法的MCC值位列前3，表明这3种算法在处理不平衡分类数据时具有较强优势。

表4 不同数据集(KK3)的实验结果

Table 4 Experimental results for different datasets (KK3)

数据集	算法							
	LWL	J48	C4.5	RF	BBN	MF	SMO	NB
A	0.648 1	0.738 5	0.601 8	0.639 6	0.766 70	0.621 1	0.656 5	0.744 2
T	0.648 1	0.731 5	0.615 8	0.614 6	0.715 70	0.651 1	0.639 5	0.777 2
F	0.348 5	0.270 1	0.324 1	0.392 3	0.223 04	0.296 2	0.401 8	0.296 1
P	0.693 9	0.711 5	0.629 9	0.665 8	0.744 20	0.650 8	0.646 6	0.742 6
R	0.628 5	0.732 5	0.680 5	0.650 8	0.710 10	0.674 9	0.602 5	0.697 1
f	0.678 6	0.721 5	0.676 8	0.642 1	0.722 20	0.610 5	0.593 6	0.732 9
M	0.419 6	0.447 4	0.256 0	0.247 2	0.514 30	0.396 1	0.312 8	0.530 5
U	0.689 9	0.703 5	0.707 2	0.624 7	0.676 60	0.686 8	0.657 5	0.722 9
C	0.629 4	0.759 2	0.611 2	0.701 3	0.690 30	0.690 2	0.605 5	0.747 5

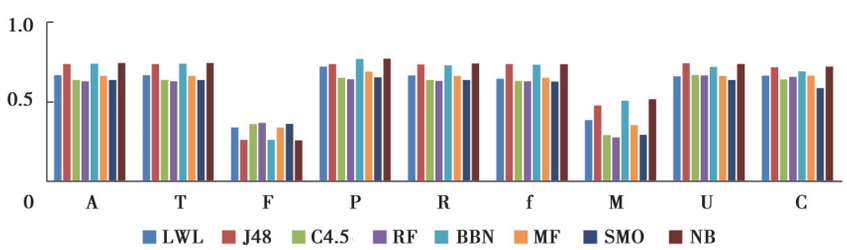

图7 算法性能图(KK3)

Fig. 7 Algorithm performance graph (KK3)

在对图8中展示的算法性能统计数据进行深入分析后，研究得出结论，所选取的几种算法均表现出色。从表5的数据来看，通过综合运用PRC曲线、ROC曲线等多种评估方法进行详细分析后，发现BBN和J48算法在PRC曲线上的表现存在不稳定性。具体来说，这些算法在PRC曲线上的表现显示出较大波动，未能确保单调一致性。这种波动和不一致性可能会对算法的可靠性产生影响。因此，在KK3数据集上，NB-K算法的卓越性能尤为显著，其在PRC曲线和ROC曲线上均表现出较高的稳定性和一致性，从而在整体性能评估中较为突出。

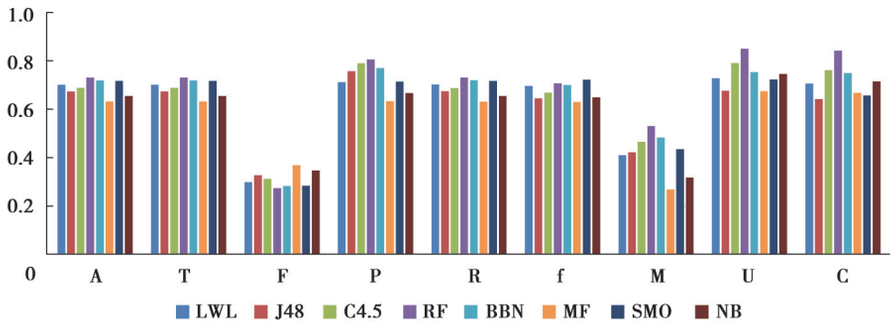

图 8　算法性能图(PK2)

Fig. 8　Algorithm performance graph (PK2)

表 5　不同数据集(PK2)的实验结果

Table 5　Experimental results for different datasets (PK2)

数据集	算法							
	LWL	J48	C4.5	RF	BBN	MF	SMO	NB
A	0.642 8	0.674 5	0.711 6	0.700 5	0.746 6	0.646 9	0.767 5	0.630 6
T	0.714 8	0.732 5	0.693 6	0.764 5	0.724 6	0.594 9	0.681 5	0.711 6
F	0.361 4	0.312 6	0.365 4	0.334 9	0.325 4	0.423 8	0.336 2	0.377 4
P	0.643 4	0.747 2	0.853 5	0.782 1	0.756 8	0.619 1	0.686 3	0.709 6
R	0.735 4	0.681 2	0.709 5	0.788 5	0.719 1	0.590 6	0.690 8	0.592 3
f	0.705 8	0.621 5	0.645 1	0.672 8	0.743 4	0.579 4	0.701 5	0.586 3
M	0.370 2	0.409 4	0.490 6	0.520 1	0.460 5	0.306 5	0.429 1	0.276 5
U	0.770 7	0.633 8	0.774 4	0.781 5	0.723 8	0.641 8	0.725 6	0.808 5
C	0.670 5	0.587 7	0.815 4	0.824 3	0.794 5	0.643 8	0.629 4	0.655 6

4　结　论

在深入探讨软件缺陷预测领域普遍存在类别不平衡问题时,笔者提出一种基于随机森林算法的解决策略。为全面评估该策略的有效性,研究将其与7种其他先进的机器学习算法进行了对比实验。这些算法包括局部加权学习法、J48、C4.8、贝叶斯网络、多层前馈神经网络、支持向量机以及NB-K算法。

在进行实验时,对多种算法在处理类别不平衡数据集的性能进行了深入分析和评估。研究关注了这些算法在分类精度、召回率、FF_1分数以及ROC曲线下面积(AUC)等关键性能指标上的表现。通过综合评估,全面掌握每种算法在处理不平衡数据集时的优势和局限性。在对比实验中,随机森林算法结合SMOTE和ENN的集成模型表现尤为出色。SMOTE技术通过合成新的少数类样本,有效缓解数据集中的不平衡问题,而ENN则通过编辑过程进一步提升数据集质量。这种集成模型不仅提高了少数类的识别率,还保持对多数类的良好分类性能,整体上提升分类的准确性和鲁棒性。

最终研究结果表明,在处理类别不平衡问题时,SMOTE+ENN+随机森林集成模型相较于其他算法,展现出更优越的性能和效果。这一发现不仅为软件缺陷预测领域提供新的思路和方法,也为其他涉及类别不平衡问题的分类任务提供宝贵参考。通过这种综合性方法,进一步推动机器学习在实际应用中的发展,特别是在那些数据不平衡现象较为普遍的领域。

参考文献

[1] Wang X, Cui Y, Duan Y. Enhancing software defect prediction using extreme randomized trees[J]. Journal of Software

Engineering, 2022, 12(3): 139-144.

[2] Rajbahadur G K, Wang S W, et al. Feature selection and ensemble learning for software defect prediction: a case study[J]. Journal of Systems and Software, 2023, 192: 2245-2261.

[3] Wang S, Liu T, Tan L. Improving software defect prediction with transfer learning[J]. IEEE Transactions on Software Engineering, 2022, 48(4): 1234-1245.

[4] Chen B, Ma L, Hu J. An improved multi-label classification method based on SVM with delicate decision boundary[J]. International Journal of Innovative Computing Information & Control Ijicic, 2010, 6(4): 1605-1614.

[5] Chen L, Fang B, Shang Z, et al. Tackling class overlap and imbalance problems in software defect prediction[J]. Software Quality Journal, 2018, 26(1): 97-125.

[6] Lin J, Liang L, Han X, et al. Cross-target transfer algorithm based on the volterra model of SSVEP-BCI[J]. Tsinghua Science and Technology, 2021, 26(4): 505-522.

[7] 宿晨, 徐华, 崔鑫, 等 . 一种处理不均衡多分类问题的特征选择集成方法[J]. 重庆大学学报, 2022, 45(5): 125-134.
Su C, Xu H, Cui X, et al. A feature selection ensemble method for imbalanced multi-class classification problems[J]. Journal of Chongqing University, 2022, 45(5): 125-134. (in Chinese)

[8] Erray W, Hacid H. A new cost sensitive decision tree method application for mammograms classification[J]. IJCSNS International Journal of Computer Science and Network Security, 2006, 6(2): 130-138.

[9] 余晓娜, 黄亮, 陈朋弟 . 基于 Segnet 网络和迁移学习的全景街区影像变化检测[J]. 重庆大学学报, 2022, 45(11): 100-107.
Yu X N, Huang L, Chen P D. Change detection of panoramic street images based on Segnet network and transfer learning[J]. Journal of Chongqing University, 2022, 45(11): 100-107. (in Chinese)

[10] Flowers S. Software failure[J]. International Journal of Information Management the Journal for Information Professionals, 1996, 17(5): 14.

[11] Boehm B, Clark B, Horowitz E, et al. Cost models for future software life cycle processes: COCOMO 2.0[J]. Annals of Software Engineering, 1995, 1: 57-94.

[12] Ji C, Li Y. A neighborhood synthesis-based oversampling method for software defect prediction[J]. Software Engineering and Applications, 2023, 12(6): 930-939.

[13] Gray D, Bowes D, Davey N, et al. Using the support vector machine as a classification method for software defect prediction with static code metrics[C]//11th Engineering Applications of Neural Networks International Conference. Berlin: Springer Berlin Heidelberg, 2009: 223-234.

[14] Zhang L, Shen Y, Zhu Y. Improved SMOTE for software defect prediction[J]. Computer Engineering and Design, 2023, 44(10): 2965-2972.

[15] Javaid N, Gul H, Baig S, et al. Using GANCNN and ERNET for detection of non technical losses to secure smart grids[J]. IEEE Access, 2021, 9: 98679-98700.

[16] Jiang Y, Li M, Zhou Z H. Software defect detection with ROCS[J]. Journal of Computer Science & Technology, 2011, 26(2): 328-342.

[17] Li L, Hu Q, Wu X, et al. Exploration of classification confidence in ensemble learning[J]. Pattern Recognition, 2014, 47(9): 3120-3131.

[18] Liaw A, Wiener M. Classification and regression by random forest[J]. R News, 2002, 23(23): 76-93.

[19] Lorenzo G, Guglielmo I, Luigi I P, et al. Logistic red flags in mass-casualty incidents and disasters: a problem-based approach[J]. Prehospital and disaster medicine, 2022, 37(3): 285-292.

[20] Lu Y, Ma W, Dong X, et al. Differentiate Xp11.2 translocation renal cell carcinoma from computed tomography images and clinical data with resNet-18 CNN and XGBoost[J]. IEEE Transactions on Software Engineering, 2023, 136(1): 347-362.

[21] ManchalaP, BisiM. Diversity base dim balance learning approach for software fault prediction using machine learning models[J]. Applied Soft Computing, 2022, 35(2): 124-292.

[22] Mccabe T J. A complexity measure[J]. IEEE Transactions on Software Engineering, 2006, 2(4): 308-320.

[23] Mjolsness E, Decoste D. Machine learning for science: state of the art and future prospects[J]. Science, 2001, 293(5537): 2051-2055.

[24] Bhutamapuram U S, Sadam R. Software defect prediction using wrapper-based dynamic feature selection[J]. Software: Practice and Experience, 2024, 54(10): 1234-1245.

[25] Wang S, Li Y, Guo S. Improving software defect prediction on NASA data using feature selection and ensemble learning[J].

Expert Systems with Applications, 2022, 93: 285-297.

［26］Zhang Y, Li H, Wang X. Just-in-time software defect prediction method for non-functional requirements using NASA data[J]. Journal of Software, 2023, 14(3): 456-467.

［27］Shepperd M, Song Q, Sun Z, et al. Data quality: some comments on the NASA software defect datasets[J]. IEEE Transactions on Software Engineering, 2013, 39(9): 1208-1215.

［28］Khan T, Faisal M. Performance evaluation of software defect prediction with NASA datasets using machine learning and data balancing techniques[J]. International Journal of Information Technology, 2023, 15: 2147-2160.

［29］Agrawal A, Malhotra R. Cross-project defect prediction for open-source software using NASA datasets[J]. International Journal of Information Technology, 2022, 14: 587-601.

［30］Liu J, Li Y, Wang S. Naive Bayes-based software defect prediction with feature selection[J]. Expert Systems with Applications, 2024, 126: 12345-12356.

［31］Wang T, Zhang Z, Jing X, et al. Multiple kernel ensemble learning for software defect prediction[J]. Automated Software Engineering, 2016, 23(4): 569-590.

［32］Xie G, Xie S, Peng X, et al. Prediction of number of software defects based on SMOTE[J]. International Journal of Performability Engineering, 2021, 17(6): 1001-1010.

［33］Yang M, Wang X, Ma L, et al. A hybrid optimization algorithm for structural balance model based on influence between nodes and community quality[J]. Swarm and Evolutionary Computation, 2022, 69(1): 101-114.

［34］Zhou Z H, Zhang M L. Solving multi-instance problems with classifier ensemble based on constructive clustering[J]. Knowledge & Information Systems, 2007, 11(2): 155-170.

［35］Zhou Z H, Zhang M L, Huang S J, et al. Multi-instance multi-label learning[J]. Artificial Intelligence, 2008, 176(1): 2291-2320.

［36］陈勇, 徐超, 何炎祥, 等. 基于编译优化的软件缺陷预测研究[J]. 电子学报, 2021, 49(2): 216-224.
Chen Y, Xu C, He Y X, et al. Research on software defect prediction based on compilation optimization[J]. Acta Electronica Sinica, 2021, 49(2): 216-224.（in Chinese）

［37］葛建新. 我国软件测试项目管理的重要作用[J]. 价值工程, 2014, 33(19): 204-205.
Ge J X. The important role of software testing project management in our country[J]. Value Engineering, 2014, 33(19): 204-205.（in Chinese）

［38］Chen J, Hu K, Yang Y. Enhancing software defect prediction using naive Bayes and transfer learning[J]. IEEE Access, 2022, 10: 12345-12356.

［39］刘文英, 林亚林, 李克文, 等. 一种软件缺陷不平衡数据分类新方法[J]. 山东科技大学学报(自然科学版), 2021, 40(2): 84-94.
Liu W Y, Lin Y L, Li K W, et al. A new classification method for imbalanced software defect data[J]. Journal of Shandong University of Science and Technology (Natural Science Edition), 2021, 40(2): 84-94.（in Chinese）

［40］于巧, 姜淑娟, 张艳梅, 等. 分类不平衡对软件缺陷预测模型性能的影响研究[J]. 计算机学报, 2018, 41(4): 809-824.
Yu Q, Jiang S J, Zhang Y M, et al. Research on the impact of class imbalance on the performance of software defect prediction models[J]. Chinese Journal of Computers, 2018, 41(4): 809-824.（in Chinese）

［41］张博, 史忠植, 赵晓非, 等. 一种基于跨领域典型相关性分析的迁移学习方法[J]. 计算机学报, 2015, 38(7): 1326-1336.
Zhang B, Shi Z Z, Zhao X F, et al. A transfer learning method based on cross-domain canonical correlation analysis[J]. Chinese Journal of Computers, 2015, 38(7): 1326-1336.（in Chinese）

（编辑　侯　湘）