

噪声可容忍的软件缺陷预测特征选择方法



滕俊元 高猛 郑小萌 江云松

北京控制工程研究所 北京 100190

(tengjunyuan@sunwiseinfo.com)

摘要 通过对缺陷数据集进行挖掘,缺陷预测模型能够提前预测出被测软件中的缺陷模块,帮助测试人员实现更有针对性的测试,而普遍存在的数据集标签噪声会影响预测模型的性能。已有的特征选择方法很少对噪声可容忍性进行针对性设计,同时在主流的具有噪声容忍能力的特征选择框架中策略选取只能依靠经验手动执行,难以在软件工程实践中得到应用。鉴于此,文中提出一种噪声可容忍的软件缺陷预测特征选择方法 NTFES (Noise Tolerable FEature Selection),即通过 Bootstrap 抽样技术生成多个自助样本集,在自助样本集上基于近似马尔可夫毯将特征进行分组并采用两种启发式特征选择策略从每个组中选出候选特征,随后利用遗传算法在候选特征空间中搜索最优特征子集。为了验证 NTFES 方法的有效性,选择了 NASA MDP 软件项目集作为实验对象并对标签注入噪声以获得带有噪声标签的数据集,通过控制标签噪声比例对 NTFES 方法以及其他基准方法(如 FULL,FCBF,CFS)进行了比较。实验结果表明:在可接受的标签噪声比例下,NTFES 方法不仅具有更高的分类性能,还具有更好的噪声可容忍性。

关键词: 软件测试;软件缺陷预测;特征选择;标签噪声;噪声可容忍

中图分类号 TP391

Noise Tolerable Feature Selection Method for Software Defect Prediction

TENG Jun-yuan,GAO Meng,ZHENG Xiao-meng and JIANG Yun-song

Beijing Institute of Control Engineering,Beijing 100190,China

Abstract Software defect prediction can identify defective modules in advance by mining the defect datasets,helping testers to achieve more targeted testing. However,the ubiquity of label noise in the datasets affects the performance of the prediction model. Few feature selection methods have been used to specifically design noise tolerance. In addition,the strategy selection in the mainstream noise tolerable feature selection framework can only be performed manually based on human experience,which is difficult to be applied in software engineering. In view of this,this paper proposes a novel method NTFES (noise tolerable feature selection). In particular,NTFES first generates multiple Bootstrap samples by Bootstrap sampling method. Then it divides the original features into different groups on Bootstrap samples by approximate Markov blanket and selects candidate features from each group based on two heuristic feature selection strategies. Sequently it uses genetic algorithm (GA) to search the optimal feature subset in the candidate feature space. To verify the effectiveness of the proposed method,this paper chooses NASA MDP dataset,and inject label noises simultaneously to imitate noisy datasets. Then it compares NTFES with other classical baseline methods,such as FULL,FCBF and CFS,by controlling the ratio of label noises. The experimental results show that the proposed method has the advantages of achieving higher classification performance and has better noise tolerable while the ratio of label noises is acceptable.

Keywords Software testing,Software defect prediction,Feature selection,Label noise,Noise tolerable

随着软件产品的数量、复杂性和规模急剧增加,软件系统变得日趋庞大和难以驾驭,软件的开发和集成变得越来越复杂,用户对软件可信性的要求也越来越高,软件产品质量直接

影响着任务成败甚至人员的生命安全。软件测试作为保证软件质量、提升软件可信性的重要手段,目前仍然以手工测试为主,通过设计大量的测试用例来覆盖所有程序执行路径,以期

到稿日期:2020-10-28 返修日期:2021-03-15 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家自然科学基金(61802017);装备预研领域基金项目(61400020407)

This work was supported by the National Natural Science Foundation of China(61802017) and Equipment Pre-Research Field Fund Project(61400020407).

通信作者:高猛(gaomeng@sunwiseinfo.com)

发现程序中存在的各种缺陷。然而这种测试方法对人的能力、经验依赖较大,很难满足高质量软件研制任务的要求。软件缺陷预测作为有效提升测试效果的辅助方法之一,通过对软件缺陷历史数据进行充分挖掘和分析,训练出相应的缺陷预测模型,从而提前预测出被测软件中的缺陷模块。该方法能够帮助测试人员实现更有针对性的测试,因此在软件工程领域具有重要的意义^[1]。

软件缺陷预测使用带标签的缺陷数据集来训练预测模型,而在获取软件缺陷数据集过程中,通常会遇到标签噪声问题,主要原因包括:1)在测试时往往不能发现全部的软件缺陷,致使软件测试存在漏报,即有缺陷的软件模块被标记为无缺陷;2)在测试时往往存在由测试人员疏忽造成的误分类问题,Herzig等^[2]的研究发现,在7000多份问题报告中有33.8%的问题报告存在错误分类;3)在软件缺陷数据集收集时往往采用手工方式对标签进行标注,导致误标记问题存在。

缺陷数据集所对应的标签质量对于预测效果至关重要,标签噪声的存在会影响预测模型的性能。虽然已有学者对特征选择方法的噪声可容忍性进行了分析^[3-5],但是目前的特征选择方法很少对噪声可容忍性进行针对性设计,同时在主流的具有噪声容忍能力的特征选择框架中策略的选取只能依靠经验手动执行,难以在软件工程实践中得到应用。

针对上述问题,本文提出一种噪声可容忍的软件缺陷预测特征选择方法 NTFES。具体来说,我们通过 Bootstrap 抽样技术生成多个自助样本集,在自助样本集上设计了特征分组方法以及候选特征选择策略并得到候选特征序列,该序列内的特征均为典型特征,之后设计了最优特征子集评价准则,通过遗传算法在候选特征序列中进行搜索以获得最优特征子集。在实验评估中,我们选择了 NASA MDP 软件项目集作为实验对象,给标签注入噪声后获得带有噪声标签的数据集,并将 NTFES 与其他基准方法进行对比,验证了 NTFES 方法的有效性。

本文的主要贡献可总结如下:

(1)提出了一种噪声可容忍的特征选择方法 NTFES,首先我们在 Bootstrap 抽样的基础上设计了一种新颖的候选特征选取策略,然后结合遗传算法设计了最优特征子集评价准则。

(2)通过在 NASA MDP 数据集上进行实验评估来验证 NTFES 方法的有效性。实验结果表明:在可接受的标签噪声比例下,NTFES 方法相比于其他特征选择方法具有更高的分类性能和更好的噪声可容忍性。

本文第 1 节介绍了相关研究工作;第 2 节给出了 NTFES 方法的实现细节;第 3 节介绍了实验设计方案并给出了实验评估结果;最后总结全文并展望未来。

1 研究背景和相关工作

1.1 软件缺陷预测中的特征选择方法

软件缺陷预测^[6-7]作为有效提升测试过程的辅助方法之一,依据软件缺陷历史数据信息并借助机器学习方法来训练预测模型,通过该模型来预测被测软件中潜在的缺陷模块,其

预测结果也逐渐成为测试人员开展针对性测试的重要依据。近年来,为了提升缺陷预测性能,国内外学者对特征选择方法进行了研究并取得了大量成果。Menzies等^[8]提出的缺陷预测方法以信息增益作为评价准则进行特征选择。Gao等^[9]针对大规模遗留软件系统提出了一种基于搜索的混合属性特征选择方法,该方法考虑了7种不同的特征排序技术以及3种不同的特征子集选择策略。Wang等^[10]利用6种不同的过滤式特征排序技术以及均值法、中值法这两种策略集成特征选择结果。Xu等^[11]提出的 MICHAC 方法利用最大信息系数过滤不相关特征,并利用分层聚类技术进行特征选择。Song等^[12]提出了通用软件缺陷预测框架,并在该框架中使用了包裹式特征选择方法。Xu等^[13]对32种不同的特征选择方法进行了多重比较和分类,发现基于不同数据集以及不同特征选择方法的预测效果排序并不一致。

文献[14-16]的研究工作与本文较为接近。其中,文献[14]利用对称不确定性以及近似马尔可夫毯技术快速过滤掉不相关特征并挑选非冗余的特征子集,但该方法存在过度删除冗余特征的缺点。文献[15]利用 Bootstrap 抽样技术对训练数据集进行自助采样,之后在采样数据集中通过中值法、均值法和指数法集成特征选择结果。文献[16]提出了一种基于 Fisher 得分和遗传算法的混合特征选择方法,该方法利用 Fisher 得分生成遗传算法初始种群。本文与这些研究的不同在于:本文基于 Bootstrap 抽样技术并结合多种特征排序技术以及遗传算法提出了由粗到精的特征选择方法,能够更有效地提高软件缺陷预测性能。

1.2 特征选择方法的噪声可容忍性

在特征选择方法的噪声可容忍性研究方面,可见的成果并不多。Liu等^[17]通过研究发现,在含有噪声的软件缺陷数据集上使用基于信息增益的评价准则仍然能够取得较好的分类效果。Rahman等^[18]通过研究发现,扩大软件缺陷数据集的规模能够有效缓解噪声对缺陷预测效果的影响。Kim等^[4]提出了关于噪声可容忍性的测量方法以及噪声识别技术 CLNI,并通过实证研究发现,当噪声总比例超过20%~35%时,预测性能会显著下降。Tantithamthavorn等^[5]通过研究发现,因软件缺陷误分类产生的噪声并不是随机发生的,并且不同的噪声注入方式对预测效果的影响并不一致。Liu等^[19]提出了具有一定噪声容忍能力的特征选择框架 FECS,该框架通过特征聚类技术以及3种不同的特征选择策略进行特征选择。

与以上研究工作不同,本文在提升特征选择方法的噪声可容忍性方面做了如下创新:1)采用 Bootstrap 抽样技术保证候选特征的多样性以及特征选择的抗噪声能力;2)在候选特征选择阶段,采用两种不同的特征选择策略,旨在使其具备更优的分类性能和噪声可容忍性;3)从定量的角度深入研究了噪声标签比例对 NTFES 方法以及基准方法的影响,并验证了本文方法的有效性。

2 NTFES 特征选择方法

本节对缺陷数据集的标签噪声问题进行了深入研究,提

出了一种噪声可容忍的特征选择方法 NTFES。首先描述该方法的基本框架,之后给出框架中软件缺陷自助样本集的生成、基于近似马尔可夫毯的特征分组、启发式候选特征选择以及基于遗传算法的最优特征选择的实现细节。

2.1 基本框架

本文首先基于 Bootstrap 抽样技术获得 N 个自助样本集,随后的特征分组以及候选特征选择均在自助样本集上进行。

图 1 给出了 NTFES 方法的基本框架。

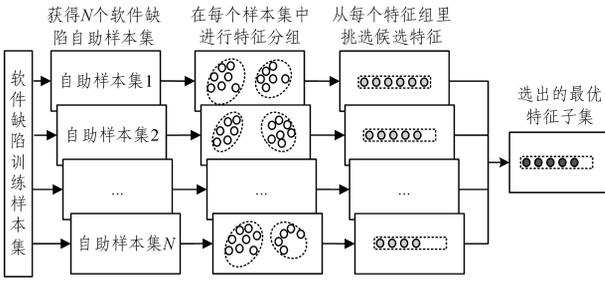


图 1 NTFES 方法的基本框架

Fig. 1 Basic framework of NTFES

NTFES 方法主要分为以下 4 个阶段:

(1) 软件缺陷自助样本集的生成

结合 Bootstrap 抽样技术,生成 N 个软件缺陷自助样本集,每个自助样本集规模和原样本集一样大。

(2) 基于近似马尔可夫毯的特征分组

本文基于近似马尔可夫毯模型的特征分组过程参考了 García-Torres 等^[20]的工作,但是在实现细节方面存在一定差异,具体内容将在 2.2.1 节进行描述。

(3) 启发式候选特征选择

完成特征分组后,我们在候选特征选择阶段采用两种不同的启发式策略,依次从每个特征组中选出典型特征,形成候选特征序列。

(4) 基于遗传算法的最优特征选择

遗传算法的输入是原样本集和候选特征序列。具体地,先基于候选特征序列计算每个特征被选中的概率,然后将计算后的结果用于算法的种群初始化,算法迭代结束后,适应度最高的个体所对应的特征集即为最优特征集。

2.2 方法描述

本节对 NTFES 方法进行说明,NTFES 的输入包括软件缺陷样本集、自助样本集的数目、相关度量指标以及两种启发式特征选择策略;输出是最优特征子集。Liu 等^[19]的研究工作表明:对称不确定性 SU 在软件缺陷预测中能够取得较好的预测性能,因此,本文方法采用对称不确定性 SU 度量特征(包括类别)之间的相关度。

NTFES 方法的执行过程如算法 1 所示。

算法 1 特征选择方法 NTFES

输入:软件缺陷样本集 D ,自助样本集的数目 N ,相关度量指标 SU ,特征选择策略 $S[2]$

输出:最优特征子集 S_{bf}

1. FOR $t=1$ to N DO
2. 生成一个大小为 T 的自助样本集 D_t

3. 借助 SU 计算自助样本集 D_t 上特征 f_i 与类别 y 之间的相关度 $SU(f_i, y)$
4. 借助 SU 计算自助样本集 D_t 上特征 f_i 与类别 f_j 之间的相关度 $SU(f_i, f_j)$
5. 根据 $SU(f_i, y)$ 和 $SU(f_i, f_j)$,将原始特征按近似马尔可夫毯模型进行划分得到特征组 FG
6. 根据策略 $S[2]$,从特征组 FG 中选择典型特征并更新到候选特征序列 C_{cf} 中
7. 根据 D 和 C_{cf} ,利用遗传算法得到最优特征子集 S_{bf}
8. RETURN S_{bf}

算法 1 首先在步骤 2 中基于 Bootstrap 抽样生成 N 个大小为 T 的自助样本集,通过集成方法能够保证候选特征的多样性^[21]。

随后 NTFES 在步骤 3—5 中利用近似马尔可夫毯模型得到特征分组,通过该方法得到的特征组呈现如下性质:1)同组内的特征冗余度高;2)不同组间的特征冗余度低。

接着 NTFES 在步骤 6 中通过两种启发式策略分别在每个特征组里选择候选特征,本文所采用的策略包括:1)选择类相关度最大的特征;2)选择 Fisher 得分最高的特征。通过以上特征选择策略的组合选取,进一步保证了特征选择结果的鲁棒性。

最后 NTFES 在步骤 7 中通过遗传算法得到最优特征子集,具体算法将在 2.2.3 节进行描述。与其他基于遗传算法的特征选择相比,NTFES 方法做了如下关键技术改进:1)通过 C_{cf} 提供的先验知识,为遗传算法提供更好的初始种群,有效提升了算法的计算效率;2)在个体适应度计算中综合考虑软件缺陷预测性能以及特征个数对特征子集的贡献,降低所选特征子集的冗余性。

2.2.1 基于近似马尔可夫毯的特征分组

近似马尔可夫毯理论^[14]认为:给定两个特征 F_i 和 $F_j (i \neq j)$ 以及类 C ,若存在 $SU(F_j, C) \geq SU(F_i, C)$ 且 $SU(F_i, F_j) \geq SU(F_i, C)$,那么说 F_j 为 F_i 的近似马尔可夫毯。其中,对称不确定性 SU 用于度量特征(包括类 C)之间的相关性,其定义为:

$$SU(X, Y) = 2 \left[\frac{IG(X|Y)}{H(X) + H(Y)} \right] \quad (1)$$

其中, $IG(X|Y)$ 表示 X 与 Y 之间的信息增益; $H(X)$ 表示 X 的熵。

与 García-Torres 等^[20]的工作不同的是,本文在特征分组过程中未删除不相关特征,目的是让所有特征都有机会参与到候选特征的选择中,避免了因标签噪声的影响导致优秀特征被误删的可能。

通过近似马尔可夫毯模型,我们可以将相似的特征划分到同一组,而不同组间的特征相关性低。具体的特征分组过程如算法 2 所示。

算法 2 基于近似马尔可夫毯的特征分组

输入:特征 f ,特征-特征相关性 $SU(f_i, f_j)$,特征-类相关性 $SU(f_i, y)$

输出:特征组 FG

1. 根据特征-类相关性 $SU(f_i, y)$,对特征进行降序排序
2. $t^* = 0$
3. FOREACH f_i s. t. $f_i \notin FG$ DO

```

4.  FGt = FGt ∪ {fi}
5.  FOREACH fj s. t. (fj ∉ FG) && (j > i) DO
6.    IF SU(fi, fj) ≥ SU(fj, y) THEN
7.      FGt = FGt ∪ {fj}
8.  FG = FG ∪ FGt
9.  t = t + 1
10. RETURN FG

```

在算法 2 中, 如果特征 f_j 的近似马尔可夫毯 (f_i) 存在, 则将 f_j 分到 f_i 所在特征组; 否则, 生成仅包含特征 f_j 的新特征组。相比于其他聚类方法, 使用近似马尔可夫毯模型对特征进行分组时不需要预先设定分组个数, 避免了人工设定分组数对分类效果的不利影响。

2.2.2 启发式候选特征选择

在候选特征选择阶段, 本文采用两种不同的选择策略依次从每个特征组中选出最典型的特征作为候选特征, 两种特征选择策略说明如下。

策略 1: 选择类相关度最大的特征

针对每个特征组, 依次计算组里每个特征与类的对称不确定性 SU , 选择值最大的特征作为候选特征。其中 SU 可通过式(1)来计算。

策略 2: 选择 Fisher 得分最高的特征

针对每个特征组, 依次计算组里每个特征的 Fisher 得分, 选择值最大的特征作为候选特征, 其中 Fisher 得分^[22]的计算公式为:

$$Fisher(f) = \frac{\sum_{i=1}^c n_i (\mu^i - \mu)^2}{\sum_{i=1}^c n_i (\sigma^i)^2} \quad (2)$$

其中, c 表示样本集中的类别总数, n_i 表示第 i 类样本总数, μ^i 表示第 i 类样本中特征 f 的均值, σ^i 表示第 i 类样本中特征 f 的方差, μ 表示特征 f 的均值。

下文给出相应的启发式候选特征选择算法, 如算法 3 所示。

算法 3 启发式候选特征选择

输入: 特征组 FG , 特征组个数 K , 特征选择策略 $S[2]$

输出: 候选特征序列 C_{cf}

```

1. FOR i=1 to K DO
2.   FOR j=1 to 2 DO
3.     根据策略  $S[j]$ , 从第  $i$  个特征组  $FG_i$  中选择典型特征  $f^*$ 
4.     IF  $f^* \notin C_{cf}$  THEN
5.        $C_{cf} = C_{cf} \cup \{(f^*, 1)\}$ 
6.     ELSE
7.        $C_{cf}[f^*] = C_{cf}[f^*] + 1$ 
8. RETURN  $C_{cf}$ 

```

本文通过示例对启发式候选特征的选择过程进行说明。假设样本集上含有 7 个特征, 特征组 $FG = \{FG_1: (f_1, f_3), FG_2: (f_2, f_5), FG_3: (f_4, f_6, f_7)\}$, 按照两种策略依次从特征组 FG_1 中选取的典型特征为 (f_1, f_1) , 从特征组 FG_2 中选取的典型特征为 (f_2, f_5) , 从特征组 FG_3 中选取的典型特征为 (f_7, f_7) , 那么最终得到的候选特征序列为 $C_{cf} = \{(f_1, 2), (f_2, 1), (f_5, 1), (f_7, 2)\}$ 。相比单一的特征选择策略, 其通过算法 3 获取的候选特征具有更优的分类性能。

2.2.3 基于遗传算法的最优特征选择

遗传算法作为一种随机搜索方法, 通过选择、交叉、变异等遗传操作模拟自然进化过程来搜索问题的近似最优解。本文借助遗传算法在候选特征空间中进行搜索以获得最优特征子集, 相比于在原始特征空间中搜索, 其大大加快了遗传算法的收敛速度; 另外借助候选特征序列提供的先验知识, 也保证了遗传算法的搜索效果。

(1) 编码方式

采用二进制编码方式^[23]对候选特征进行编码。假设候选特征个数为 L , 则可通过二进制编码得到一个长度为 L 的二进制符号串, 若第 i 位所代表的特征被选中则将该位设置为 1, 否则设置为 0。

(2) 初始种群的生成

不同于传统的初始种群生成策略, 本文基于候选特征序列提供的先验知识生成初始种群。具体地, 假设候选特征序列中的特征个数为 L , 特征 f_i 对应的候选次数为 n_i , 初始种群规模为 M , 按照式(3)计算个体中特征 f_i 对应的基因位为 1 的概率, 并根据计算后的概率生成 M 个个体 g_1, g_2, \dots, g_M 。

$$P(i=1) = \frac{n_i}{\sum_{j=1}^L n_j} \quad (3)$$

(3) 适应度函数的选择

本文将软件缺陷预测 AUC 值作为评估所选特征子集的准则。在构造适应度函数时, 我们选择经典的朴素贝叶斯作为分类器并将预测结果的 AUC 值作为适应度函数的一部分。另外, 为了进一步降低所选特征之间的冗余性, 我们在适应度函数中增加了惩罚项, 以控制选入特征子集的特征个数。适应度函数如式(4)所示。

$$f(X) = AUC(X) + \lambda \sqrt{1/N(X)} \quad (4)$$

其中, $AUC(X)$ 表示使用个体 X 对应的特征子集进行缺陷预测的 AUC 值; $N(X)$ 表示个体 X 对应的特征子集中的特征个数; λ 表示惩罚参数, 用于控制选入特征子集中的特征数目, 本文将 λ 设置为 0.3。

(4) 遗传算子选择

本文的选择算子采用最佳保留选择法, 交叉算子采用单点交叉, 变异算子采用基本位变异, 终止条件为达到最大代数或连续 5 次保持所选特征子集不变。

下文给出了基于遗传算法的最优特征选择的具体算法, 如算法 4 所示。

算法 4 基于遗传算法的最优特征选择

输入: 软件缺陷样本集 D , 候选特征序列 C_{cf}

输出: 最优特征子集 S_{bf}

```

1. 根据  $C_{cf}$ , 利用式(3)生成的  $M$  个个体组成初始种群  $G$ 
2. 根据  $D$  和  $G$ , 利用式(4)计算每个个体的适应度, 并根据适应度进行选择操作
3. 采用单点交叉算子进行交叉操作
4. 采用基本位变异进行变异操作, 生成新的种群  $G^*$ 
5.  $G = G^*$ 
6. 判断是否满足终止条件, 若不满足, 则转到步骤 2 继续执行; 否则输出最优特征子集  $S_{bf}$ 

```

3 实验评估

在本节的实验评估中,对 NTFES 方法的有效性进行验证,验证过程中需要回答以下两个实验问题。

RQ1:相比基准方法,NTFES 是否能在无标签噪声和有标签噪声的情况下均能提高缺陷预测模型的性能?

RQ2:相比基准方法,NTFES 能否在受标签噪声影响的情况下有效降低预测模型的性能损失?

本节的实验评估框架如图 2 所示。

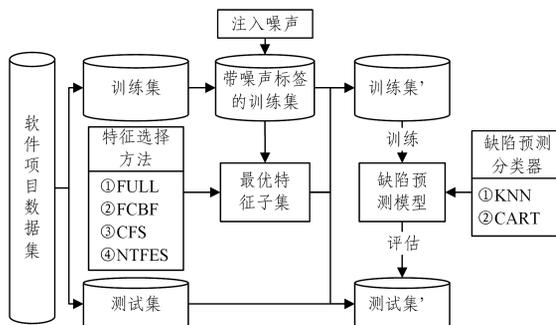


图 2 实验评估框架示意图

Fig. 2 Schematic diagram of experimental evaluation framework

为了评估 NTFES 方法,本文选取了 3 种基准方法进行对比实验,分别为:1) FULL 方法,即使用全部特征来构建预测模型;2) FCBF 方法^[14],即使用对称不确定性 SU 以及近似马尔可夫毯过滤掉不相关的特征并挑选非冗余的特征子集;3) CFS 方法^[24],即使用特征-类相关度以及特征-特征相关度挑选最优特征子集。另外,在缺陷预测模型方面,选取了经典的 K 最近邻(KNN)以及决策树(CART)作为本节实验评估的分类器。

在实验评估过程中,采用 10×10 折交叉验证的方式^[19]生成训练集和验证集,并通过在训练集中注入不同比例的标签噪声,观察缺陷预测模型的性能变化情况,进而评估在不同比例标签噪声的影响下各种特征选择方法的有效性。

3.1 数据集

本文实验数据选择来自 NASA MDP(Metric Data Program)公布的 11 个软件项目数据集¹⁾。表 1 列出了 NASA MDP 数据集的基本信息。

表 1 NASA MDP 数据集的基本信息

Table 1 Basic information of NASA MDP dataset

数据集	编程语言	特征数	样本数	缺陷率 (所占百分比)
CM1	C	37	327	12.84
JM1	C	21	7 782	21.49
KC1	C++	21	1 183	26.54
KC3	java	39	194	18.56
MC1	C/C++	38	1 988	2.3
MC2	C	39	125	35.2
MW1	C	37	253	10.67
PC1	C	37	705	8.65
PC3	C	37	1 077	12.44
PC4	C	37	1 287	13.75
PC5	C++	38	1 711	27.53

为了获得带有不同比例标签噪声的数据集,我们采用如下方式给标签注入噪声:

(1)根据给定的噪声比例 ω 和数据集 T ,从该数据集的 N 个训练总样本中随机选择 K 个样本,其中 $K = \lfloor N * \omega \rfloor$;

(2)对于选出的 K 个样本中的每一个样本,将其原始标签进行翻转。

3.2 评价指标

本文采用 AUC 和 RLA 两种评价指标对 NTFES 方法进行评估,其中 AUC 用于度量模型预测的性能,RLA 用于度量模型预测的稳定性。

(1) AUC (Area Under ROC Curve):广泛用于评估类不平衡的数据集预测性能的优劣,其定义为 ROC 曲线所覆盖的区域面积,取值范围为 $[0, 1]$ 。该值越接近 1,表示模型预测的性能越好;该值越接近 0,表示模型预测的性能越差。

(2) RLA^[25] (Relative Loss of Accuracy):衡量预测模型在注入噪声前后 AUC 的损失情况,可用于评估不同缺陷预测模型受到噪声影响的情况下预测性能的稳定性。计算结果越小,表示模型预测的稳定性越好;计算结果越大,表示模型预测的稳定性越差,其计算公式如式(5)所示:

$$RLA_{x\%} = \frac{AUC_{0\%} - AUC_{x\%}}{AUC_{0\%}} \quad (5)$$

其中, $RLA_{x\%}$ 表示注入噪声比例为 $x\%$ 时的 RLA 指标, $AUC_{0\%}$ 表示未注入噪声时预测模型的 AUC 指标, $AUC_{x\%}$ 表示注入噪声比例为 $x\%$ 时的 AUC 指标。

3.3 实验验证

NTFES 在特征选择过程中用到了 Bootstrap 抽样技术以及遗传算法,因此需要设置的参数包括自助采样数、种群规模、最大迭代次数、交叉概率以及变异概率。另外,本文实验阶段用到的 KNN 和 CART 分类器均采用默认参数设置。在实验验证环节,我们根据算法调参结果对本文方法各参数进行设置(见表 2)。

表 2 NTFES 方法的参数设置

Table 2 Parameter settings in NTFES

算法	参数名称	参数值
Bootstrap	自助采样数	60
	种群规模	100
	最大迭代次数	40
GA	交叉概率	0.5
	变异概率	0.2

本文的 NTFES 方法借助机器学习工具库 Scikit-Learn 和遗传算法库 DEAP 编程实现,而其他基准方法借助特征选择库 Scikit-Feature^[26] 编程实现。实验的硬件环境配置为 Windows 10 Intel Core i7-8550@1.80GHz 8.00GB RAM。

3.3.1 RQ1:模型预测性能分析

为了回答 RQ1,本次实验首先向训练集标签注入噪声,噪声注入比例依次设置为 0%,10%,20%,30%,然后分别基于 KNN 和 CART 分类器,在同一噪声标签比例下评估 NTFES 方法与其他基准方法在 11 个 NASA 数据集上的 AUC 指标(见表 3 和表 4)。

¹⁾ <http://mdp. ivv. nasa. gov>

表3 不同噪声标签比例下各特征选择方法的 AUC 指标(基于 KNN 分类器)

Table 3 AUC of each feature selection method under different noise label ratios (based on KNN)

噪声比例	特征选择方法	软件项目数据集											占优次数
		CM1	JM1	KC1	KC3	MC1	MC2	MW1	PC1	PC3	PC4	PC5	
0%	FULL	0.488	0.546	0.551	0.505	0.499	0.649	0.497	0.493	0.530	0.524	0.557	1
	FCBF	0.532	0.537	0.577	0.501	0.500	0.530	0.514	0.496	0.550	0.619	0.562	0
	CFS	0.487	0.535	0.551	0.505	0.508	0.650	0.513	0.498	0.530	0.499	0.564	1
	NTFES	0.534	0.545	0.578	0.507	0.510	0.598	0.554	0.549	0.556	0.657	0.588	9
10%	FULL	0.480	0.544	0.544	0.492	0.488	0.585	0.548	0.498	0.551	0.511	0.565	0
	FCBF	0.508	0.533	0.558	0.487	0.489	0.550	0.487	0.498	0.532	0.506	0.559	0
	CFS	0.480	0.534	0.546	0.491	0.490	0.589	0.548	0.496	0.554	0.519	0.566	1
	NTFES	0.529	0.545	0.562	0.582	0.500	0.575	0.618	0.531	0.556	0.636	0.590	10
20%	FULL	0.502	0.537	0.545	0.506	0.495	0.604	0.492	0.493	0.544	0.540	0.567	0
	FCBF	0.530	0.523	0.542	0.504	0.478	0.541	0.510	0.480	0.527	0.529	0.566	0
	CFS	0.519	0.537	0.541	0.506	0.478	0.612	0.492	0.493	0.540	0.533	0.534	0
	NTFES	0.560	0.541	0.557	0.556	0.508	0.673	0.594	0.529	0.599	0.650	0.572	11
30%	FULL	0.482	0.522	0.527	0.455	0.529	0.572	0.576	0.513	0.553	0.525	0.556	0
	FCBF	0.490	0.516	0.519	0.528	0.487	0.495	0.466	0.495	0.555	0.540	0.545	0
	CFS	0.480	0.522	0.522	0.455	0.515	0.563	0.579	0.515	0.557	0.529	0.551	2
	NTFES	0.564	0.533	0.544	0.628	0.541	0.610	0.512	0.551	0.541	0.640	0.588	9

表4 不同噪声标签比例下各特征选择方法的 AUC 指标(基于 CART 分类器)

Table 4 AUC of each feature selection method under different noise label ratios (based on CART)

噪声比例	特征选择方法	软件项目数据集											占优次数
		CM1	JM1	KC1	KC3	MC1	MC2	MW1	PC1	PC3	PC4	PC5	
0%	FULL	0.508	0.571	0.605	0.599	0.650	0.572	0.608	0.682	0.612	0.721	0.657	3
	FCBF	0.521	0.545	0.645	0.472	0.582	0.545	0.544	0.543	0.551	0.590	0.598	0
	CFS	0.543	0.563	0.576	0.632	0.574	0.554	0.651	0.663	0.582	0.618	0.618	1
	NTFES	0.562	0.524	0.656	0.615	0.692	0.607	0.689	0.637	0.587	0.728	0.675	7
10%	FULL	0.494	0.569	0.567	0.577	0.548	0.635	0.579	0.585	0.596	0.658	0.613	2
	FCBF	0.418	0.533	0.639	0.468	0.526	0.565	0.496	0.524	0.506	0.527	0.591	0
	CFS	0.493	0.552	0.565	0.554	0.507	0.644	0.608	0.643	0.558	0.594	0.617	1
	NTFES	0.533	0.526	0.651	0.614	0.591	0.607	0.613	0.656	0.608	0.657	0.665	8
20%	FULL	0.554	0.542	0.565	0.625	0.534	0.587	0.510	0.583	0.518	0.645	0.596	3
	FCBF	0.564	0.528	0.638	0.518	0.473	0.495	0.455	0.519	0.515	0.493	0.579	0
	CFS	0.428	0.541	0.548	0.577	0.564	0.542	0.510	0.578	0.539	0.564	0.578	0
	NTFES	0.593	0.533	0.654	0.609	0.583	0.553	0.568	0.640	0.554	0.688	0.658	8
30%	FULL	0.534	0.520	0.547	0.655	0.478	0.581	0.571	0.508	0.519	0.581	0.603	0
	FCBF	0.617	0.524	0.540	0.480	0.543	0.532	0.574	0.507	0.500	0.551	0.532	1
	CFS	0.480	0.528	0.544	0.574	0.533	0.512	0.536	0.562	0.527	0.535	0.548	0
	NTFES	0.557	0.531	0.630	0.689	0.583	0.585	0.624	0.615	0.542	0.612	0.640	10

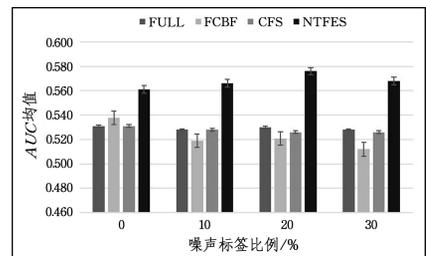
表3和表4记录的是不同噪声标签比例下各特征选择方法在每个数据集上进行 10×10 折交叉验证时所有的AUC结果。最后一列表示在11个NASA数据集上某一特征选择方法AUC指标的占优次数。具体来说:

(1)若基于KNN分类器,虽然特征选择方法在不同数据集或不同噪声标签比例上的性能表现各有差异,但从AUC占优次数来说,NTFES方法在不同噪声标签比例下的AUC占优次数分别为9,10,11,9,表现显著优于其他3种基准方法。

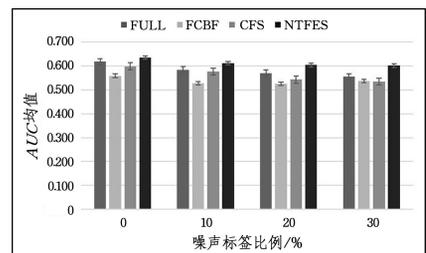
(2)若基于CART分类器,从AUC占优次数来说,NTFES方法在不同噪声标签比例下的AUC占优次数分别为7,8,8,10,表现依然显著优于其他3种基准方法。

(3)从AUC占优次数来说,NTFES方法在KNN分类器上的表现要优于在CART分类器上的表现。

图3为各特征选择方法在不同噪声标签比例下分别使用KNN和CART分类器在11个NASA数据集上进行缺陷预测的AUC均值对比图。从图3可以看出,标签注入噪声前后,对于不同分类器,使用NTFES方法进行缺陷预测的AUC均值均高于其他3种基准方法。这表明本文方法对于提升含有噪声标签的软件缺陷预测性能是非常有效的。



(a) 基于 KNN



(b) 基于 CART

图3 标签噪声对 AUC 指标的影响

Fig. 3 Influence of label noise on AUC

更进一步,我们随机选取了数据集 KC3, MW1 以及 MC2

作为实验对象,通过控制标签噪声注入比例在 0%~85% 范围内变化,观察给定模型性能的变化情况。本次实验中设定

标签噪声注入比例变化步长为 5%,图 4 和图 5 分别基于 KNN 和 CART 分类器给出了本次实验的结果。

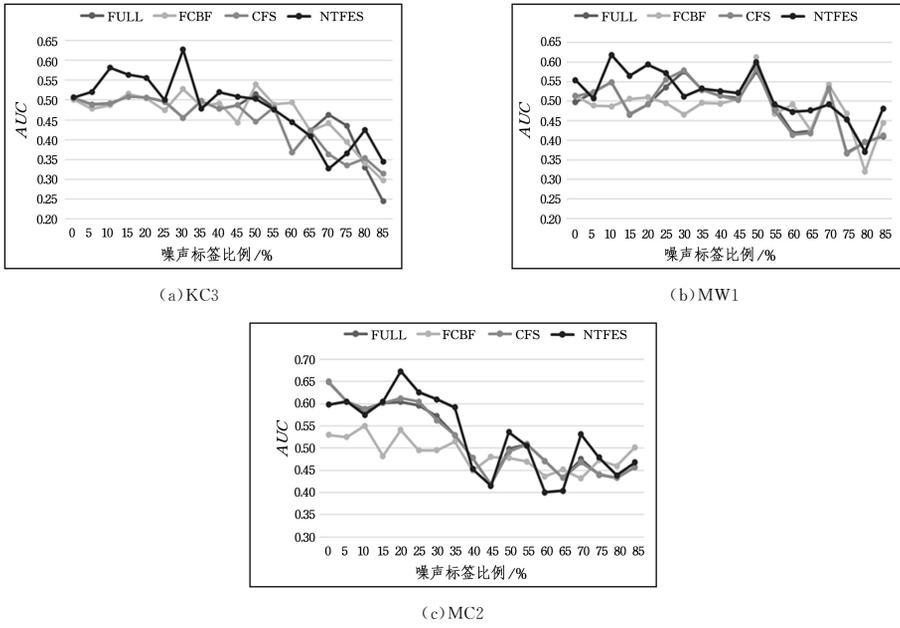


图 4 预测精度 V. S. 噪声标签比例(基于 KNN 分类器)

Fig. 4 Prediction accuracy V. S. noise label ratio (based on KNN)

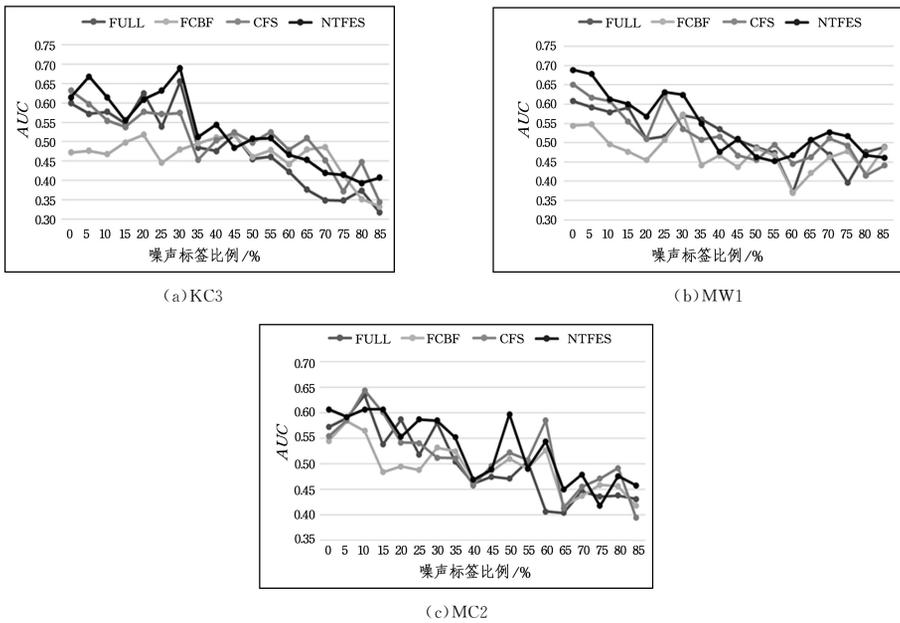


图 5 预测精度 V. S. 噪声标签比例(基于 CART 分类器)

Fig. 5 Prediction accuracy V. S. noise label ratio (based on CART)

从图 4 可以看出,当选择 KNN 作为分类器时,随着标签噪声注入比例的增加,各特征选择方法的缺陷预测性能都有所下降。在噪声标签占比不超过 45% 的情况下,NTFES 方法的 AUC 指标普遍高于其他基准方法;从 AUC 平均值的角度看,NTFES 方法在数据集 KC3, MW1 以及 MC2 上的 AUC 平均值分别为 0.537, 0.550, 0.575, 均高于其他基准方法;当噪声标签占比超过 45% 时,所有特征选择方法在预测性能方面的损失都比较显著。

从图 5 可以看出,当选择 CART 作为分类器时,随着标签噪声注入比例的增加,各特征选择方法的缺陷预测性能都

有所下降。但在噪声标签占比不超过 45% 的情况下,NTFES 方法的 AUC 指标普遍高于其他基准方法;从 AUC 平均值角度看,NTFES 方法在数据集 KC3, MW1 以及 MC2 上的 AUC 平均值分别为 0.592, 0.594, 0.565, 均高于其他基准方法;当噪声标签占比超过 45% 时,所有特征选择方法在预测性能方面的损失都比较显著。

通过以上分析,我们可以得出如下结论:

(1) 在未注入标签噪声的情况下,NTFES 方法在缺陷预测性能方面的表现优于其他特征选择算法,对于不同分类器,这种优势保持稳定;

(2)在注入标签噪声的数据集上,对于不同分类器,NTFES方法在预测性能方面均优于其他特征选择算法,随着注入标签噪声比例超过45%,这种优势将逐渐消失。这说明NTFES方法需要进一步的改进,以使其在高标签噪声(即噪声标签占比超过45%)的影响下仍能够取得更优的预测性能。

3.3.2 RQ2:模型预测稳定性分析

为了回答RQ2,本次实验依次控制标签噪声注入比例为10%,20%,30%,之后分别基于KNN和CART分类器计算NTFES方法与其他基准方法在11个NASA数据集上的RLA指标。表5和表6记录的是在不同噪声标签比例下分别选择KNN和CART作为分类器时各特征选择方法在每个

数据集上进行 10×10 折交叉验证时的RLA结果,最后一列表示在11个NASA数据集上某一特征选择方法的RLA指标的占优次数。

从表5和表6的实验结果可以看出:

(1)若基于KNN分类器,虽然特征选择方法在不同数据集或不同噪声标签比例上的RLA指标互有优劣,但从RLA占优次数来说,NTFES方法在不同噪声标签比例下的RLA占优次数分别为5,5,6,表现优于其他3种基准方法。

(2)若基于CART分类器,从RLA占优次数来说,NTFES方法在不同噪声标签比例下的RLA占优次数分别为5,6,6,表现依然显著优于其他3种基准方法。

表5 不同噪声标签比例下各特征选择方法的RLA指标(基于KNN分类器)

Table 5 RLA of each feature selection method under different noise label ratios (based on KNN)

噪声比例	特征选择方法	软件项目数据集											占优次数
		CM1	JM1	KC1	KC3	MC1	MC2	MW1	PC1	PC3	PC4	PC5	
10%	FULL	0.016	0.004	0.013	0.026	0.022	0.099	-0.103	-0.010	-0.040	0.025	-0.014	2
	FCBF	0.045	0.007	0.033	0.028	0.022	-0.038	0.053	-0.004	0.033	0.183	0.005	1
	CFS	0.014	0.002	0.009	0.028	0.035	0.094	-0.068	0.004	-0.045	-0.040	-0.004	3
	NTFES	0.009	0.000	0.028	-0.148	0.020	0.038	-0.116	0.033	0.000	0.032	-0.003	5
20%	FULL	-0.029	0.016	0.011	-0.002	0.008	0.069	0.010	0.000	-0.026	-0.031	-0.018	3
	FCBF	0.004	0.026	0.061	-0.006	0.044	-0.021	0.008	0.032	0.042	0.145	-0.007	0
	CFS	-0.066	-0.004	0.018	-0.002	0.059	0.058	0.041	0.010	-0.019	-0.068	0.053	3
	NTFES	-0.049	0.007	0.036	-0.097	0.004	-0.125	-0.072	0.036	-0.077	0.011	0.027	5
30%	FULL	0.012	0.044	0.044	0.099	-0.060	0.119	-0.159	-0.041	-0.043	-0.002	0.002	3
	FCBF	0.079	0.039	0.101	-0.054	0.026	0.066	0.093	0.002	-0.009	0.128	0.030	0
	CFS	0.014	0.024	0.053	0.099	-0.014	0.134	-0.129	-0.034	-0.051	-0.060	0.023	2
	NTFES	-0.056	0.022	0.059	-0.239	-0.061	-0.020	0.076	-0.004	0.027	0.026	0.000	6

表6 不同噪声标签比例下各特征选择方法的RLA指标(基于CART分类器)

Table 6 RLA of each feature selection method under different noise label ratios (based on CART)

噪声比例	特征选择方法	软件项目数据集											占优次数
		CM1	JM1	KC1	KC3	MC1	MC2	MW1	PC1	PC3	PC4	PC5	
10%	FULL	0.028	0.004	0.063	0.037	0.157	-0.110	0.048	0.142	0.026	0.087	0.067	2
	FCBF	0.198	0.022	0.009	0.008	0.096	-0.037	0.088	0.035	0.082	0.107	0.012	1
	CFS	0.092	0.020	0.019	0.123	0.117	-0.162	0.066	0.030	0.041	0.039	0.002	3
	NTFES	0.052	-0.004	0.008	0.002	0.146	0.000	0.110	-0.030	-0.036	0.098	0.015	5
20%	FULL	-0.091	0.051	0.066	-0.043	0.178	-0.026	0.161	0.145	0.154	0.105	0.093	3
	FCBF	-0.083	0.031	0.011	-0.097	0.187	0.092	0.164	0.044	0.065	0.164	0.032	1
	CFS	0.212	0.039	0.049	0.087	0.017	0.022	0.217	0.128	0.074	0.087	0.065	1
	NTFES	-0.055	-0.017	0.003	0.010	0.158	0.089	0.176	-0.005	0.056	0.055	0.025	6
30%	FULL	-0.051	0.089	0.096	-0.093	0.265	-0.016	0.061	0.255	0.152	0.194	0.082	1
	FCBF	-0.184	0.039	0.163	-0.017	0.067	0.024	-0.055	0.066	0.093	0.066	0.110	4
	CFS	0.116	0.062	0.056	0.092	0.071	0.076	0.177	0.152	0.095	0.134	0.113	0
	NTFES	0.009	-0.013	0.040	-0.120	0.158	0.036	0.094	0.035	0.077	0.159	0.052	6

通过以上分析,我们可以得出如下结论:在注入标签噪声的数据集上,NTFES方法在噪声可容忍性方面的表现优于其他特征选择算法,对于不同分类器,这种优势保持稳定。

3.4 有效性影响因素分析

本节从实验准备有效性以及实验设计有效性两个方面对可能影响实验结果的因素进行简要分析。实验准备有效性主要涉及数据集选取、噪声生成以及算法实现。本文选取了软件缺陷预测研究中常采用的NASA项目数据集,保证了研究结果具有一定的代表性。同时,依据文献[19]提出的标签噪声注入方式得到带有不同比例标签噪声的数据集。此外,本文实现的算法代码主要基于机器学习工具库Scikit-Learn、遗传算法库DEAP以及特征选择库Scikit-Feature,并设计了测试用例对算法实现的正确性进行验证,最大程度地保证分类器和特征选择方法实现的正确性。实验设计有效性主要涉及

评价指标的选取。由于选取的软件缺陷数据集具有类不平衡的特点,为了避免该影响因素,本文使用了AUC值作为评价指标,同时使用了文献[19]提出的RLA指标对模型预测的稳定性进行评估。

结束语 本文针对数据集标签中的噪声问题,提出了一种新颖的噪声可容忍特征选择方法NTFES,即基于近似马尔可夫毯在自助样本集上进行特征分组,随后提出两种启发式特征选择策略选出候选特征,然后利用遗传算法在候选特征空间中搜索最优特征子集。基于NASA MDP软件项目集对该方法的有效性进行了评估,结果表明:本文提出的方法不仅具有更高的分类性能,还具有更好的噪声可容忍性。由实验分析可知,随着注入标签噪声比例超过45%,特征选择方法的预测性能较差。因此,设计更加有效的特征选择策略来提高本文方法在高标签噪声环境下的缺陷预测性能,将是我们的

下一步的研究重点。另外,后续工作中我们也将建立面向航天领域的软件缺陷数据集,并结合真实场景下类标噪声的实际情况,从类标噪声假阳性和假阴性两个层面对本文方法的有效性和普适性进行验证。

参 考 文 献

- [1] CATAL C. Software fault prediction: A literature review and current trends [J]. *Expert Systems with Applications*, 2011, 38(4): 4626-4636.
- [2] HERZIG K, JUST S, ZELLER A. It's not a bug, it's a feature: How misclassification impacts bug prediction [C] // *Proceedings of the International Conference on Software Engineering*, San Francisco, USA, 2013: 392-401.
- [3] BOLON-CANEDO V, SANCHEZ-MARONO N, ALONSO-BETANZOS A. Feature selection for high dimensional data [J]. *Progress in Artificial Intelligence*, 2016, 5(2): 65-75.
- [4] KIM S, ZHANG H Y, WU R X, et al. Dealing with noise in defect prediction [C] // *Proceedings of the International Conference on Software Engineering*, Honolulu, USA, 2011: 481-490.
- [5] TANTITHAMTHAVORN C, MCINTOSH S, HASSAN A E, et al. The impact of mislabeling on the performance and interpretation of defect prediction models [C] // *Proceedings of the International Conference on Software Engineering*, Firenze, Italy, 2015: 812-823.
- [6] HALL T, BEECHAM S, BOWES D, et al. A systematic literature review on fault prediction performance in software engineering [J]. *IEEE Transactions on Software Engineering*, 2012, 38(6): 1276-1304.
- [7] CHEN X, GU Q, LIU W S, et al. Software defect prediction [J]. *Journal of Software*, 2016, 27(1): 1-25.
- [8] MENZIES T, GREENWALD J, FRANK A. Data mining static code attributes to learn defect predictors [J]. *IEEE Transactions on Software Engineering*, 2007, 33(1): 2-13.
- [9] GAO K H, KHOSHGOFTAAR T M, WANG H J, et al. Choosing software metrics for defect prediction: an investigation on feature selection techniques [J]. *Software Practice & Experience*, 2011, 41(5): 579-606.
- [10] WANG H J, KHOSHGOFTAAR T M, HULSE J V, et al. Metric selection for software defect prediction [J]. *International Journal of Software Engineering & Knowledge Engineering*, 2011, 21(2): 237-257.
- [11] XU Z, XUAN J F, LIU J, et al. MICHAC: defect prediction via feature selection based on maximal information coefficient with hierarchical agglomerative clustering [C] // *Proceedings of the 23rd International Conference on Software Analysis, Evolution and Reengineering*, Washington: IEEE Computer Society, 2016, 1: 370-381.
- [12] SONG Q B, JIA Z H, SHEPPERD M, et al. A general software defect-proneness prediction framework [J]. *IEEE Transactions on Software Engineering*, 2011, 37(3): 356-370.
- [13] XU Z, LIU J, YANG Z J, et al. The impact of feature selection on defect prediction performance: an empirical comparison [C] // *Proceedings of the 27th International Symposium on Software Reliability Engineering*, Washington: IEEE Computer Society, 2016: 309-320.
- [14] YU L, LIU H. Efficient feature selection via analysis of relevance and redundancy [J]. *Journal of Machine Learning Research*, 2004, 5(10): 1205-1224.
- [15] PES B, DESSI N, ANGIANI M. Exploiting the ensemble paradigm for stable feature selection: A case study on high dimensional genomic data [J]. *Information Fusion*, 2017, 35(C): 132-147.
- [16] ZHOU M. A hybrid feature selection method based on fisher score and genetic algorithm [J]. *Journal of Mathematical Sciences: Advances and Application*, 2016, 37: 51-78.
- [17] LIU S L, CHEN X, LIU W S, et al. FECAR: A feature selection framework for software defect prediction [C] // *Proceedings of the Annual Computer Software and Applications Conference*, Vasteras, Sweden, 2014: 426-435.
- [18] RAHMAN F, POSNETT D, HERRAIZ I, et al. Sample size vs. bias in defect prediction [C] // *Proceedings of the Joint Meeting of the European Software Engineering Conference and the Symposium on Foundations of Software Engineering*, Saint Petersburg, Russia, 2013: 147-157.
- [19] LIU W S, CHEN X, GU Q, et al. A noise tolerable feature selection framework for software defect prediction [J]. *Chinese Journal of Computers*, 2018, 41(3): 506-520.
- [20] GARCÍA-TORRES M, GÓMEZ-VELA F, MELIÁN-BATISTA B, et al. High-dimensional feature selection via feature grouping: A Variable Neighborhood Search approach [J]. *Information Sciences*, 2016, 326: 102-118.
- [21] LIU Y, CAO J J, DIAO X C, et al. Survey on Stability of Feature Selection [J]. *Journal of Software*, 2018, 29(9): 2559-2579.
- [22] DEVIJVER P A, KITTLER J. *Pattern recognition: a statistical approach [M]*. London: Prentice Hall, 1992.
- [23] VAFARIE H, DE JONG K A. Genetic algorithms as a tool for feature selection in machine learning [C] // *Proceedings of the 4th IEEE International Conference on Tools with AI*, Washington DC: IEEE Computer Society, 1992: 200-203.
- [24] HALL M A. *Correlation-based feature subset selection for machine learning [D]*. Hamilton, New Zealand: University of Waikato, 1999.
- [25] SÁEZ J A, GALAR M, LUENGO J, et al. Tackling the problem of classification with noisy data using Multiple Classifier Systems: Analysis of the performance and robustness [J]. *Information Sciences*, 2013, 247: 1-20.
- [26] LI J, CHENG K, WANG S, et al. Feature selection: A data perspective [J]. *ACM Computing Surveys (CSUR)*, 2017, 50(6): 1-45.



TENG Jun-yuan, born in 1985, master, senior engineer. His main research interests include embedded software testing and software engineering.



GAO Meng, born in 1982, master, senior engineer. His main research interests include embedded software testing and software engineering.