文章编号 1004-924X(2019)02-0469-10

# 基于迁移学习的跨公司航天软件缺陷预测

哈清华[1,2],刘大有[1,3*],陈 媛[2],刘 逻[2]
(1.吉林大学 计算机科学与技术学院,吉林 长春 130012;
2.中国科学院 长春光学精密机械与物理研究所,吉林 长春 130033;
3.吉林大学 符号计算与知识工程教育部重点实验室,吉林 长春 130012)

**摘要**:为提高航天软件测试的效率和质量,针对同公司航天软件数量少、研制周期长的特点,提出了一种跨公司航天软件缺陷预测方法。从航天软件背景信息复杂、规模大、功能独立等特征出发,提出基于静态分类缺陷预测的模型构建思想。引入迁移学习方法,利用最近邻分类器和数据引力模型,对训练数据的分布特征进行修正,提高训练数据与目标数据的相似性;为提高模型的泛化能力以适应目标数据的多样性,提出在训练数据中加入少量目标数据用于模型训练。将该方法在实际工程中进行应用,实验结果表明,与已有软件缺陷预测方法相比,该方法在保持较低误报率(不高于 0.3)的情况下可有效提高召回率(接近 0.6),整体可信度得到有效增强(G-measure 超过 0.6),方法稳定度高,泛化能力较强;本方法在实际工程中对测试规模影响可控,测试效率得到提高。

**关 键 词**:缺陷预测;迁移学习;最近邻分类器;数据引力;朴素贝叶斯
**中图分类号**:TB853.29 **文献标识码**:A **doi**:10.3788/OPE.20192702.0469

# Approach to cross-company spacecraft software defect prediction based on transfer learning

HA Qing-hua[1,2], LIU Da-you[1,3*], CHEN Yuan[2], LIU Luo[2]

(1. *College of Computer Science and Technology, Jilin University, Changchun* 130012, *China*;
2. *Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Science, Changchun* 130033, *China*;
3. *Key Laboratory of Symbolic Computation and Knowledge Engineering for the Ministry of Education, Jilin University, Changchun* 130012, *China*)
* *Corresponding author, E-mail: liudy@jlu.edu.cn*

**Abstract**: In order to improve the efficiency and quality of aerospace software testing, an approach to cross-company aerospace software defect prediction was proposed, especially for the scarcity of within-company software and the long cycle of development. Considering the complexity, large scale, and independent functions of aerospace software, the idea of building a defect prediction model based on static classification was proposed. In this paper, the transfer learning method was introduced. Using the nearest neighbor classifier and data gravity model, the distribution characteristics of training data

**收稿日期**:2018-06-27;**修订日期**:2018-09-06.
**基金项目**:国家自然科学基金资助项目(No.61502198,No.61572226,No.61472161)

were corrected to improve the similarity between training data and target data . In order to improve the generalization ability of the model to adapt to the diversity of target data , a small amount of target data was added to the training data for model training . The approach was applied to the test for aerospace software testing . The results of application show that , compared with existing software defect prediction methods , the proposed method can effectively improve the recall rate (close to 0 .6) with a low false alarm rate (not higher than 0 .3). The overall credibility is effectively enhanced (G-measure is over 0 .6), and the method has high stability and strong generalization ability . This method can control the test scale in practical projects and improve testing efficiency .

**Key words**: defect prediction ; transfer learning ; nearest neighbor classifier ; data gravity ; naive Bayes

## 1 引　言

据统计航天、航空飞行器的事故原因中软件占了其中的 60%[1-2]。软件缺陷预测技术已成为当前软件研究领域的热点。早在上个世纪 70 年代,软件缺陷预测技术就已开始发展,很多缺陷预测模型被提出并取得了一定的成功[3-4]。但航天领域的软件具有软件规模大、生命周期长、同公司的软件数量少等特点,导致使用同公司数据进行训练的机器学习方法难以在航天软件缺陷预测中有效使用。如何利用不同公司的软件数据来构建通用的软件缺陷预测模型,以实现跨公司软件缺陷预测已成为航天软件缺陷预测研究的重点[5]。

跨公司缺陷预测的核心问题在于不同公司的数据无法满足独立同分布假设,进而导致传统的机器学习方法失效。目前主要有两类方法来解决数据的分布差异问题。一类从数据的有效性出发,依据选定的评价方法对测试数据进行筛选,删除无效数据,提高数据的分布共性[6-7]。另一类从数据的相似性出发,通过相似性评价方法依据训练数据与目标数据的相似程度对训练数据赋权,以提高数据的分布共性[8-10]。以上两类方法虽然解决了训练数据内部的分布差异,并在一定程度上修正了训练数据与目标数据之间的分布差异,在实验中也获得了积极成果。但当训练数据与目标数据的分布差异较大时,现有的方法则会出现错误率大幅升高,预测模型失效的情况,导致在实际工程应用中的效果不理想。

本文针对当前航天软件缺陷预测研究的问题,结合工程实际,提出从两个方面对现有缺陷预测方法加以改进。一方面,引入迁移学习方法,将两类数据处理方法进行综合应用改进以实现训练数据分布修正的最佳优化,提高模型的泛化能力;另一方面,针对工程应用中的实际情况,创新性的提出在训练中加入少量本公司内数据,力求不过多增加工作量的前提下,提高数据的独立分布特性,增强模型的个性化适应能力。进而提出了一种跨公司航天软件缺陷预测方法。通过实验结果可知该方法与已有方法相比,在保持较低误报率的前提下,可有效提高模型召回率。在实际工程应用中表明,该模型的泛化能力好,有较强的实用性,可有效提高工作效率。



## 2 航天软件缺陷预测建模

面向软件缺陷预测的模型按照自变量的类型可分为静态缺陷预测模型和动态缺陷预测模型[11-12]。区别主要在于静态缺陷预测模型关注软件代码本身自然属性变量的度量,而动态缺陷预测模型关注软件全生命周期属性变量的度量。按照因变量的类型可分为分类缺陷预测模型和回归缺陷预测模型[13-14]。前者用于分析软件的缺陷有无,后者分析软件的缺陷数量。

航天软件的主要特点为:(1)软件生命周期长,背景信息复杂,软件功能相对单一明确且严格独立。这导致采用动态缺陷预测模型时,影响因素分析复杂,建模难度大,模型标准难以统一,获取的模型通用性差。而静态缺陷预测模型可更好的满足模型通用化要求。(2)软件相对数量少,但规模大,功能划分清晰,模块相对独立。如果采用回归缺陷预测模型则准确性难以保证,效果不理想。而采用分类预测模型,将分析目标由配置项改为立足于软件的模块,则可以做到软件缺陷定位准确,达到有的放矢的效果,更符合航天软件的特点。

因此,本文采用静态分类缺陷预测模型思想进行方法的构建。主要由 4 个步骤组成。第一步,使用最近邻分类器算法[6](Nearest Neighbor Filtering,NNF)对训练数据进行过滤处理,剔除与目标数据分布差异较大的无效数据;第二步,使用数据引力方法[9,15](Data Gravitation,DG),依据数据的相似性对训练数据进行权值修订,进而达到修正数据分布特征的目的;第三步,在修正后的训练数据中加入少量的公司内数据进而合成最终的训练数据,并使用加权朴素贝叶斯方法获取预测模型;第四步,使用获得的软件缺陷预测模型用于目标数据的缺陷预测,获取预测结果。本方法的主要工作流程如图 1 所示。

## 3　跨公司航天软件缺陷预测方法

### 3.1　数据的定义

本文的研究对象为模块级软件(函数),采用的方法为基于软件静态度量元的数据分析。因此,本文中处理的数据即模块(如无特殊说明,任意模块均有 $n$ 个度量属性)可以用如下方式表示:

定义 1　数据 $x_i=\{a_{i1},a_{i2},\cdots,a_{in},c_i\}$,其中 $a_{ij}$ 表示数据 $x_i$ 的第 $j$ 个度量属性,$c_i$ 为数据 $x_i$ 的类别属性,本文主要研究软件模块的缺陷有无,因此 $c_i\in(T,F)$,$T$ 表示该数据有缺陷,$F$ 表示该数据无缺陷。

定义 2　数据集 $L=\{x_1,x_2,\cdots,x_n\}$,其中 $x_i$ 代表数据集中的某个数据。

### 3.2　最近邻分类器

根据以往的研究经验,在机器学习过程中,对模型效能影响最大的就是训练数据中存在的无效数据问题。而由于数据的分布差异巨大,导致在跨公司的软件缺陷预测中该问题更加突出。为解决此问题,本文采用 NNF 方法来清除无效数据。

本方法的核心思想在于从训练数据集中寻找与目标数据差异小的数据来构造新的训练数据集。其中任意两个数据的差异程度使用最近邻算法来评估,评价函数为两个数据的欧氏距离。

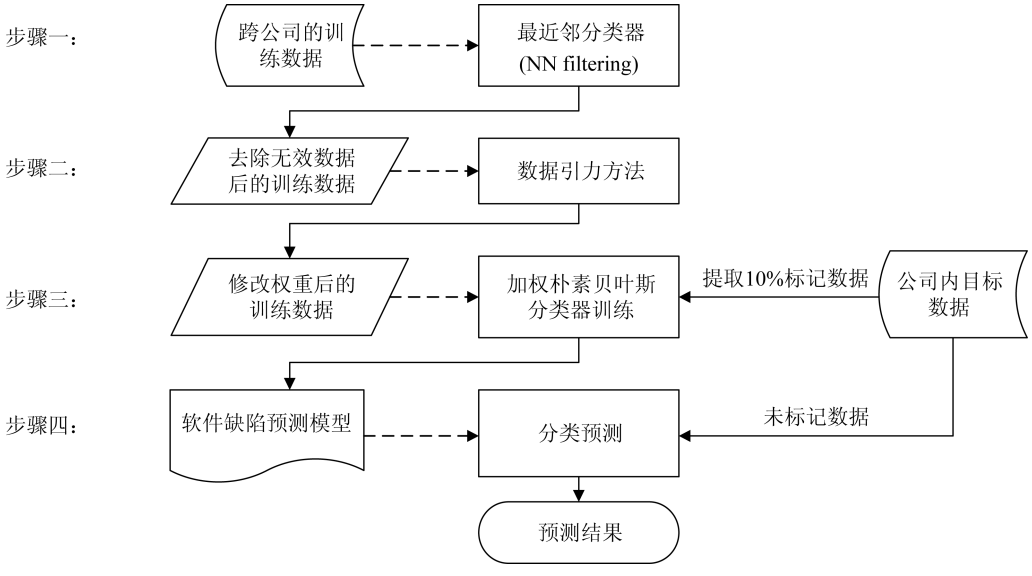


图 1　跨公司航天软件缺陷预测方法

Fig.1　A method of cross-company aerospace software defect prediction

任意数据 $x_i$ 与 $x_k$ 的欧氏距离 $d_{ik}$ 为:

$$d_{ik}=\sqrt{\sum_{j=1}^{n}(a_{ij}-a_{kj})^2}. \qquad (1)$$

本方法的详细处理过程参见算法 1。

算法 1　训练数据有效性评价算法

输入:训练数据集 $L_{cc}$、目标数据集 $L_{target}$

输出:有效训练数据集 $L_{nn}$

初始化:有效训练数据集 $L_{nn}$ 为空,选取近邻数量为 $k$。

步骤:

(1)从 $L_{target}$ 中取出一个数据 $x_i$,依次与 $L_{cc}$ 中的每个数据计算欧氏距离;

(2)在 $L_{cc}$ 中取与数据 $x$ 欧氏距离最近的 $k$ 个数据放入 $L_{nn}$；

(3)在 $L_{\text{target}}$ 中查找下一个数据 $x_{i+1}$，如果 $L_{\text{target}}$ 中还有未访问数据 $x_{i+1}$，跳转至步骤 1；

(4)对 $L_{nn}$ 进行整理，清除重复的数据，保证数据的唯一性；

(5)输出 $L_{nn}$，算法结束。

### 3.3 数据引力

通过第一步的处理虽可去除大量无效数据，但数据的分布差异并未修正。为减小训练数据与目标数据的分布差异，实现知识的转移学习，本文引入迁移学习方法来解决此问题。迁移学习的目的是将某个领域或任务上学习到的知识或模式应用到不同的但相关的领域或问题中。用于解决训练集与目标集之间的领域、任务或者数据分布的差异问题。而跨公司的软件缺陷预测问题属于典型的数据分布差异问题。

本文采用数据引力模型方法来评估训练数据与目标数据的相似性，通过相似性评价函数来赋予训练数据响应的权值，进而达到修正数据分布差异的目的。数据引力方法的主要思想为：通过在数据中模拟物理学中的万有引力方程，构造一种数据之间的“数据引力”，以实现对数据相似度的评价，“数据引力”越强则代表数据之间的相似度越高。

通过定义 1，计算可以获得训练数据集的数据属性值极值向量：$Max=\{max_1, max_2, \cdots, max_n\}$ 和 $Min=\{min_1, min_2, \cdots, min_n\}$ 其中，$max_i$ 和 $min_i$ 分别表示训练数据集中第 i 个属性的最大值和最小值。由此，依据属性独立性假设，可以定义某训练数据 $x_i$ 与目标数据集的相似度 $s_i$ 为：

$$s_i = \sum_{j=1}^{n} h(a_{ij}), \tag{2}$$

其中：$a_{ij}$ 为 $x_i$ 的第 $j$ 个度量属性值，$h(a_{ij})=\begin{cases}1, min_j \leqslant a_{ij} \leqslant max_j\\ 0, \text{others}\end{cases}$.

假设用属性来度量数据的重量(任意属性重量为 $M$)，用相似度来度量测试数据与目标数据集(数据集的数据量为 $k$)的距离，则有：某训练数据 $x_i$ 的质量 $m_1=s_iM$，目标数据集的质量 $m_2=knM$，$x_i$ 与目标数据集的距离 $r_i=n-s_i+1$。由此，依据万有引力公式，可定义训练数据 $t_i$ 的权重 $w_i$ 为：

$$w_i=\frac{m_1 m_2}{r_i^2}=\frac{knsiM^2}{(n-s_i+1)^2}\approx\frac{s_i}{(n-s_i+1)^2}. \tag{3}$$

根据定义可知，当训练数据与目标数据集相似度越高时，获得的权值 $w_i$ 也会相应更高，反之则会更低。通过式(3)，可对训练数据逐一进行权值赋值，实现数据分布的调整。

### 3.4 朴素贝叶斯分类器

根据已有的研究成果，在软件缺陷预测领域，朴素贝叶斯分类方法可获得不弱于甚至是强于其他分类方法的结果[5,8]。因此本文选取朴素贝叶斯方法作为缺陷预测分类器。

根据贝叶斯公式，在证据 $E=\{e_1, e_2, \cdots, e_n\}$ 的前提下，某一假设 $H$ 的后验概率为：

$$P(H \mid E)=\frac{P(H)}{P(E)}\prod_{i=1}^{n} P(e_i \mid H). \tag{4}$$

设 $L_{\text{train}}=\{x_1, x_2, \cdots, x_n\}$ 为训练数据集，$L_{\text{target}}=\{y_1, y_2, \cdots, y_m\}$ 为目标数据集，其中 $y_i=\{a_{i1}, a_{i2}, \cdots, a_{in}, c_i\}$。依据公式(4)，假定各属性独立无影响，且对结果具有相同权重，则有 $y_i$ 的类别属性 $c_i$ 为：

$$c_i = \operatorname{argmax} P(c \mid y_i) = \operatorname{argmax} \frac{P(c)\prod_{j=1}^{n} P(a_{ij} \mid c)}{\sum_{c\in(T,F)} P(c)\prod_{j=1}^{n} P(a_{ij} \mid c)}, \tag{5}$$

其中先验概率 $P(c)$ 和条件概率 $P(a_{ij} \mid c)$ 均可通过训练数据计算获取。由于使用的训练数据具有权值，因此根据文献[8]，加权 $P(c)$ 的计算公式如式(6)：

$$P(c)=\frac{1+\sum_{k=1}^{n} w_k I(c_k, c)}{n_c+\sum_{k=1}^{n} w_k}, \tag{6}$$

其中：$w_k$ 为训练数据 $x_k$ 的权重，$c_k$ 为 $x_k$ 的类别属性值，$I(c_k, c)$ 为选择函数，当 $c_k=c$ 时函数值为 1，否则为 0，$n_c$ 为类别的总数，本文中为 2。依据公式 6 可推知，条件概率 $P(a_{ij} \mid c)$ 的计算公式如式(7)：

$$P(a_{ij} \mid c)=\frac{1+\sum_{k=1}^{n} w_k I(c_k, c) I(a_{kj}, a_{ij})}{n_j+\sum_{k=1}^{n} w_k I(a_{kj}, a_{ij})}, \tag{7}$$

其中：$a_{kj}$ 为训练数据 $x_k$ 的第 $j$ 个度量属性值，$n_j$ 为 $L_{\text{train}}$ 中第 $j$ 个属性的不同取值数量。

## 4　实验分析

为综合考量本方法在实际工程实践中的效果,本次实验共分为有效性实验和效率实验两部分进行设计。其中有效性实验用于验证本方法在工程实践中可否做到有效预测软件缺陷;而效率实验用于验证本方法在工程实践中对工作量、工作时间,工作结果的影响是否可控、可接受。

### 4.1　实验数据

有效性实验使用的训练数据是从 PROMISE 网站的公共数据集中收集而来,虽然所有数据均为美国国家航空航天局(NASA)的实际工程项目,但 7 个数据集来自不同的开发团队,可视为跨公司数据集。目标数据来自国内某科研机构,均为我国航天工程项目,共计 10 个软件,10 个软件的开发人员也来自于不同的开发团队,同样具有跨公司特征。有效性实验数据情况参见表 1。

度量元的选取并不是本次研究的主要内容,且现有的研究认为度量元的选取并没有最优方案[11],因此结合以往研究[11-12],本文选取数据集的公共度量元作为本次研究的输入变量。度量元选择情况见表 2。

**表 1　有效性实验数据集**

Tab.1　Validation data set

| 训练数据 | | | 目标数据 | | |
|---|---|---|---|---|---|
| 软件名 | 实例数 | 用途 | 软件名 | 实例数 | 用途 |
| kc1 | 2 109 | 数据存储与管理 | TGZ | 216 | 航天相机控制 |
| kc2 | 522 | 数据存储与管理 | THZ | 189 | 航天相机控制 |
| kc3 | 194 | 数据存储与管理 | JBZ | 176 | 航天相机控制 |
| pc1 | 705 | 卫星飞行控制 | TGR | 69 | 航天载荷温控 |
| cm1 | 327 | 航天载荷 | THR | 56 | 航天载荷温控 |
| mc2 | 125 | 视频导航 | JBR | 85 | 航天载荷温控 |
| mw1 | 253 | 实验控制 | TGC | 59 | 成像控制 |
| | | | THC | 27 | 成像控制 |
| | | | XYC | 35 | 成像控制 |
| | | | KZK | 112 | 卫星通讯控制 |

**表 2　实验度量元**

Tab.2　Experimental metric element

| 序号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 度量元 | Branch count | Code and comment loc | Comment loc | Cyclomatic complexity | Design complexity | Halstead difficulty | Halstead effort | Halstead error | Halstead length | Halstead time | Halstead volume | Total operands | Total operators | Unique operands | Unique operators | Executable loc | Total loc |

工作效率实验主要考察本方法在工程实践中对工作的实际影响,因此主要选取了较为具有代表性的 1 个项目的数据,该数据与有效性实验中的目标数据的来源相同,也为国内某科研机构的航天工程项目。属于同 1 个项目的 4 个软件,均由各自独立的开发团队完成,其数据情况参见表 3。

**表 3　效率实验数据集**

Tab.3　Efficiency data set

| 软件名 | 实例数 | 用途 |
|---|---|---|
| HXZK | 374 | 航天相机控制 |
| HXCX | 102 | 成像控制 |
| HXR | 47 | 航天载荷温控 |
| HXTX | 133 | 通讯控制 |



### 4.2　评价指标

本次实验在有效性实验和效率实验中分别采用两套指标进行实验评估:

4.2.1　有效性实验指标

缺陷预测方法应是有效的,这体现在发现的缺陷是否充分,误报率是否够低等。为实现以上目标的估算,需要首先计算 4 个数值。

TP(True Positive)为有缺陷的模块被分类正确的数量;FN(False Negative)为有缺陷的模块被分类错误的数量;TN(True Negative)为无缺陷的模块被分类正确的数量;FP(False Positive)为无缺陷的模块被分类错误的数量。

本文引入 3 个指标用于评价本方法的有效性。

召回率 PD,用以评估数据中缺陷模块发现的概率:

$$PD=\frac{TP}{TP+FN}. \tag{8}$$

误报率 PF,用以评估数据中无缺陷模块被错误分类的概率:

$$PF=\frac{FP}{FP+TN}. \tag{9}$$

G-measure,用以评估 PD 和 PF 值的综合表现的度量值(该值越大分类器的效果越好):

$$G\text{-}measure=\frac{2PD(1-PF)}{PD+(1-PF)}. \tag{10}$$

4.2.2　工作效率实验指标

在实际工程实践中,好的缺陷预测方法不应对工作效率有较大的不利影响,否则该方法不具有工程意义,因此本方法对工作效率的影响是也是关键考虑因素。本文选取测试用例数量($N$)、测试时间($T$)和缺陷发现数量($Bug$)三个度量值(三个指标依次反应了工作量、工作用时和工作效果)用于评估。

同时引入 2 个复合指标,评价本方法对工作效率的综合影响:

单位时间内的缺陷发现量 PTN,用以评估测试工作在单位时间内可发现的问题数量,本文选取 10 h 作为基本单元时间:

$$PTB=\frac{Bug}{T}\times 10. \tag{11}$$

单位用例的缺陷发现量 PNB,用以评估测试工作在单位用例数量内可发现的问题数量,本文选取 100 个用例作为基本单元用例:

$$PNB=\frac{Bug}{N}\times 100. \tag{12}$$

### 4.3　实验过程及结果分析

4.3.1　有效性实验

为验证本方法的有效性,本文选取了 3 个代表性方法与本文提出的方法进行比较,各方法的介绍见表 4。

**表 4　实验方法**

Tab.4　Experimental method

| 方法名称 | 处理过程 |
|---|---|
| NB<br>(朴素贝叶斯分类器) | 该方法基于贝叶斯理论,使用中不对数据进行预处理,直接利用全部的训练数据进行模型训练,再用于目标数据的分类。 |
| NN-filtering<br>(最近邻分类器) | 根据训练数据与目标数据的距离,选择 $k$ 个最近邻数据组成训练数据集,再使用 NB 方法进行训练和分类。 |
| TNB<br>(迁移朴素贝叶斯) | 根据训练数据与目标数据的相似度对训练数据赋权,再使用 NB 方法对赋权后的训练数据进行训练和分类 |
| NTW<br>(本文提出的方法) | 对训练数据进行坏数据剔除和数据赋权双重处理,再对目标数据抽取 10% 进行标记后放入训练数据,将混合后的训练数据使用 NB 方法进行训练和分类。 |

实验中对数据有两次预处理操作,首先为解决数据值分布过大的问题,在实验开始之前,对所有的训练数据和目标数据使用 log 函数进行过滤,该方法的有效性在以往研究中得到了证明并强烈推荐使用[6]。

其次,为提高数据在迁移过程中的泛化能力以及解决数值型数据在 NB 方法中的适应性,本文在执行数据训练前采用 MDL 离散化方法[9]对训练数据进行离散处理,再进行数据的迁移和训练操作。

为消除随机误差带来的影响,本次实验中每次随机抽取 90% 的训练数据和 10% 的目标数据组合成新的训练数据用于生成模型和进行判定,并重新抽样数据进行实验反复执行 10 次,以覆盖所有的训练数据和目标数据。

对实验的结果按照指标 PD、PF、G-measure 的计算方法进行统计并取 10 次运行结果的均值,最终整体实验结果见表 5。对实验结果按照 10 个软件和 4 种方法分别进行均值、最大值、最小值、样本方差统计,其结果见表 6。

**表 5　有效性实验结果**

Tab.5　Results of validity experiment

| 序号 | 软件名 | 指标 | NB | NN-filtering($k$=10) | TNB | NTW($k$=100) |
|---|---|---|---|---|---|---|
| 1 | TGZ | PD | 0.211 4 | 0.491 4 | **0.651 4** | 0.640 0 |
| | | PF | 0.268 3 | 0.414 6 | 0.341 5 | **0.243 9** |
| | | G-measure | 0.328 1 | 0.534 3 | 0.655 0 | **0.693 2** |
| 2 | THZ | PD | 0.240 9 | 0.328 5 | **0.562 0** | 0.547 4 |
| | | PF | **0.269 2** | 0.538 5 | 0.480 8 | 0.288 5 |
| | | G-measure | 0.362 3 | 0.383 8 | 0.539 8 | **0.618 8** |
| 3 | JBZ | PD | 0.117 3 | 0.549 4 | **0.648 1** | 0.623 5 |
| | | PF | **0.142 9** | 0.285 7 | 0.285 7 | 0.357 1 |
| | | G-measure | 0.206 3 | 0.621 1 | **0.679 6** | 0.633 0 |
| 4 | TGR | PD | 0.041 7 | 0.145 8 | 0.354 2 | **0.562 5** |
| | | PF | **0.095 2** | 0.047 6 | 0.142 9 | 0.333 3 |
| | | G-measure | 0.079 7 | 0.252 9 | 0.501 2 | **0.610 2** |
| 5 | THR | PD | 0.162 2 | 0.216 2 | 0.270 3 | **0.432 4** |
| | | PF | **0.157 9** | 0.421 1 | 0.526 3 | 0.473 7 |
| | | G-measure | 0.272 0 | 0.314 8 | 0.344 2 | **0.474 8** |
| 6 | JBR | PD | 0.102 9 | 0.352 9 | **0.529 4** | 0.441 2 |
| | | PF | **0.176 5** | 0.529 4 | 0.470 6 | 0.352 9 |
| | | G-measure | 0.183 0 | 0.403 4 | **0.529 4** | 0.524 6 |
| 7 | TGC | PD | 0.449 0 | 0.653 1 | 0.632 7 | **0.755 1** |
| | | PF | 0.300 0 | **0.100 0** | **0.100 0** | 0.300 0 |
| | | G-measure | 0.547 1 | **0.756 9** | 0.743 0 | 0.726 5 |
| 8 | THC | PD | 0.136 4 | 0.272 7 | 0.181 8 | **0.409 1** |
| | | PF | **0.000 0** | 0.200 0 | 0.200 0 | 0.200 0 |
| | | G-measure | 0.240 0 | 0.406 8 | 0.296 3 | **0.541 4** |
| 9 | XYC | PD | 0.000 0 | 0.407 4 | 0.333 3 | **0.592 6** |
| | | PF | 0.125 0 | 0.125 0 | 0.000 0 | 0.000 0 |
| | | G-measure | 0.000 0 | 0.556 0 | 0.500 0 | **0.744 2** |
| 10 | KZK | PD | 0.067 4 | 0.427 0 | **0.719 1** | 0.696 6 |
| | | PF | **0.087 0** | 0.304 3 | 0.217 4 | 0.304 3 |
| | | G-measure | 0.125 6 | 0.529 2 | **0.749 5** | 0.696 1 |

**表 6　统计结果**

Tab.6　Statistical results

| 指标 | PD | | | | PF | | | | G-measure | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 均值 | 最大值 | 最小值 | 样本方差 | 均值 | 最大值 | 最小值 | 样本方差 | 均值 | 最大值 | 最小值 | 样本方差 |
| NB | 0.1529 | 0.449 0 | 0.000 0 | 0.016 2 | 0.162 2 | 0.300 0 | 0.000 0 | 0.008 9 | 0.234 4 | 0.547 1 | 0.000 0 | 0.024 3 |
| NN-filtering ($k$=10) | 0.384 4 | 0.653 1 | 0.145 8 | 0.023 9 | 0.296 6 | 0.538 5 | 0.047 6 | 0.031 3 | 0.475 9 | 0.756 9 | 0.252 9 | 0.023 0 |
| TNB | 0.4882 | 0.719 1 | 0.181 8 | 0.035 2 | 0.276 5 | 0.526 3 | 0.000 0 | 0.031 2 | 0.553 8 | 0.749 5 | 0.296 3 | 0.024 1 |
| NTW ($k$=100) | 0.570 0 | 0.755 1 | 0.409 1 | 0.013 4 | 0.285 4 | 0.473 7 | 0.000 0 | 0.015 4 | 0.626 3 | 0.744 2 | 0.474 8 | 0.008 2 |

通过对数据分析可知，NB 方法虽然有最低的 PF 值，但同时又有最低的 PF 值和最低的 G-measure 值，这使得该方法在工程上不具有实践意义。NN-filtering 方法虽然在 PD 值和 G-measure 值上获得了一定的性能提升，但 PD 均值仍小于 0.5，不利于工程实践。TNB 方法和 NTW 方法在统计中获得了明显好于其他两个方法的表现，在保持了 PF 值可接受(小于 0.3)的情况下，有效提高了 PD 值；NTW 方法与 TNB 方法相比，在获得了相似 PF 值的情况下，进一步提高了 PD 值，并获得了最大 G-measure 值；同时通过分析样本方差值可见，NTW 方法在不同的数据集上的性能偏差最小，模型稳定性最好，这在需要面对各种复杂背景情况的工程实践中具有十分重要的积极意义。

4.3.2　效率实验

工作效率实验中的测试人员均为同一软件测评中心的测评人员，该测评中心具有中国合格评定国家认可委员会(CNAS)的认定资质。测试方法为动态黑盒，本次实验将测试人员分为 4 组，A 组按照传统的测试方法对 4 个软件进行测试；B、C、D 组先使用缺陷预测方法对被测软件进行缺陷预测，在依据预测的结果对被测软件进行更有针对性的测试用例设计。其中 B 组使用 NN-filtering 方法、C 组使用 TNB 方法、D 组使用本文提出的 NTW 方法。测试前，除将有效性实验的测试结果提供给各测试组，以利于测试实施外，不再提供其他支持或指导意见。本次测试使用测试时间($T$，单位：h)、设计的测试用例数($N$，单位：个)、缺陷发现数($Bug$，单位：个)、三个指标对实验进行度量，其实验度量结果见表 7。使用公式(11)和公式(12)对 PTB 和 PNB 指标进行统计，结果见表 8。

通过数据分析可见，在工程实践中不管采用了何种缺陷预测方法，测试发现的缺陷数量都有提高，测试活动的整体质量得到了提升。与不进行缺陷预测的方法相比，采用 NN-filtering 方法的测试活动其 PTB 和 PNB 值都有降低，测试的效率难以保证；采用 TNB 方法和 NTW 方法的测试活动，其 PTB 和 PNB 值得到了提高，有明显的提升测试效率的效果。

**表 7　效率实验结果**

Tab.7　Results of efficiency experiment

| 软件名 | 指标 | A | B | C | D |
|---|---|---|---|---|---|
| HXZK | $T$ | 260 | 347 | 288 | 311 |
| | $N$ | 692 | 945 | 865 | 825 |
| | $Bug$ | 45 | 52 | 61 | 63 |
| HXCX | $T$ | 132 | 181 | 140 | 149 |
| | $N$ | 248 | 318 | 284 | 273 |
| | $Bug$ | 15 | 17 | 20 | 24 |
| HXR | $T$ | 74 | 83 | 79 | 105 |
| | $N$ | 96 | 107 | 121 | 104 |
| | $Bug$ | 12 | 15 | 15 | 14 |
| HXJK | $T$ | 165 | 221 | 170 | 182 |
| | $N$ | 271 | 378 | 320 | 346 |
| | $Bug$ | 27 | 33 | 35 | 43 |

**表 8**　PTB 与 PNB 的计算结果

Tab .8　Results of PTB and PNB

| 软件名 | 指标 | A | B | C | D |
|---|---|---|---|---|---|
| HXZK | PTB | 1 .731 | 1 .499 | 2 .118 | 2 .026 |
| | PNB | 6 .503 | 5 .503 | 7 .052 | 7 .636 |
| HXCX | PTB | 1 .136 | 0 .939 | 1 .429 | 1 .611 |
| | PNB | 6 .048 | 5 .346 | 7 .042 | 8 .791 |
| HXR | PTB | 1 .622 | 1 .807 | 1 .899 | 1 .333 |
| | PNB | 12 .500 | 14 .019 | 12 .397 | 13 .462 |
| HXJK | PTB | 1 .636 | 1 .493 | 2 .059 | 2 .363 |
| | PNB | 9 .963 | 8 .730 | 10 .938 | 12 .428 |
| 平均值 | PTB | 1 .531 | 1 .435 | 1 .876 | 1 .833 |
| | PNB | 8 .754 | 8 .399 | 9 .357 | 10 .579 |

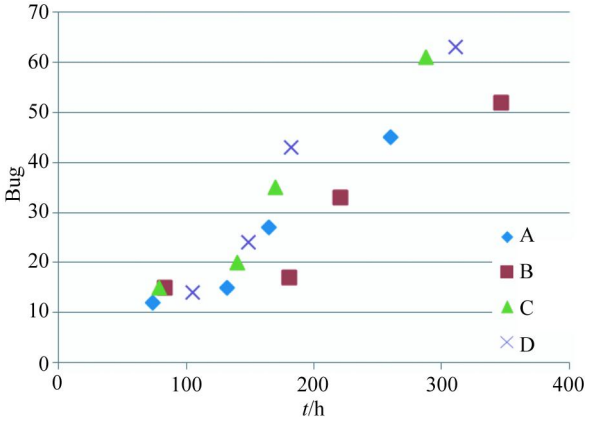


图 2　测试时间与测试缺陷

Fig .2　*T* and bug

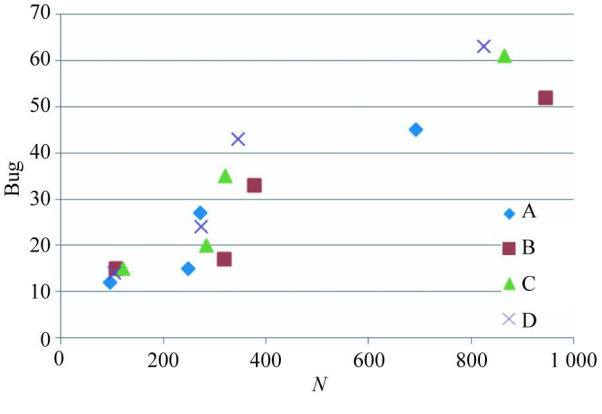


图 3　测试用例与测试缺陷

Fig .3　*N* and bug

图 2 和图 3 为本次测试的数据分布情况，其中图 2 为测试时间与测试问题数量的分布，图 3 为测试用例与测试问题数量的分布。根据数据特点可知，图中的数据越靠近左上越有更高的测试效率，反之越靠近右下测试效率越低。图中采用 TNB 和 NTW 方法的数据分布更加靠近左上，且随着被测软件规模的增加，NTW 方法具有更高的时间和用例效率。

## 5　结　论

本文从航天软件的特点出发，提出了一种基于静态分类缺陷预测模型思想的软件缺陷预测方法。该方法分为数据清理和模型训练两个部分。在数据清理过程中，首先通过采用 NN-filtering 分类方法，清除了训练数据中的无效数据，接着使用数据引力方法对数据进行迁移计算，实现了数据的分布特征修正；之后再模型训练过程中，针对工程实践的特点，本着进一步控制数据的样本偏差，提高模型泛化能力以适应大量不同的目标数据的目的，引入少量的公司内数据用于模型训练。最终使用获得的模型进行数据的分类预测。

通过有效性实验可知，本方法与传统的软件缺陷预测方法相比，在保持较低误报率（PF 值低于 0 .3）的前提下，可有效提高模型召回率（PD 值提高至接近 0 .6），预测结果的整体可信度得到了有效增强（G-measure 值提升至超过 0 .6）。同时，本方法在实验中的三项统计数据均获得了最小的样本方差，说明本方法的模型稳定度高，泛化能力强，可更好地适应复杂的数据环境。

通过效率实验可知，与不采用缺陷预测的常规方法相比，本方法可有效提高测试中发现的问题数量，提升测试质量。同时本方法对测试规模的影响可控，虽然测试时间和用例数量都有明显增加，但测试效率同时得到了提升（PTB 和 PNB 值都有提高），测试方法有效、可用，达到了预期效果。

**参考文献：**

[1]　王俊杰，沈湘衡，张波，等 .环境参数与状态参数融合的测试用例集约简方法[J].光学 精密工程，2009，17(7)：1678-1685 .

WANG J J，SHEN X H，ZHANG B，*et al*..Optimal test suite generation methods based on fusion of environment and state parameters [J]. *Opt. Preci-*

*sion Eng*., 2009, 17(7): 1678-1685.(in Chinese)

[2] 苗悦,王峰.敏捷成像卫星"逐级择优"在轨任务实时规划[J].光学 精密工程, 2018,26(1):150-160.
MIAO Y, WANG F. Optimize-by-priority on-orbit task real-time planning for agile imaging satellite [J]. *Opt. Precision Eng*., 2018,26(1): 150-160.(in chinese)

[3] MCCABE T J. A complexity measure [J]. *IEEE Transactions on Software Engineering*, 1976, SE-2(4): 308-320.

[4] CHIDAMBER S R, KEMERER C. A metrics suite for object oriented design [J]. *IEEE Transactions on Software Engineering*, 1994, 20(6): 476-493.

[5] 陈翔,王莉萍,顾庆,等.跨项目软件缺陷预测方法研究综述[J].软件学报, 2018, 41(1):254-274.
CHEN X, WANG L P, GU Q, *et al*.. A survey on cross-project software defect prediction methods [J]. *Journal of Software*, 2018, 41(1): 254-274.(in Chinese)

[6] 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..

[7] ZIMMERMANN T, NAGAPPAN N, GALL H, *et al*.. Cross-project defect prediction: a large scale experiment on data vs. domain vs. process [C]. *Proc. Joint Meeting of the European Software Engineering Conference and the ACM Sigsoft Symposium on the Foundations of Software Engineering*.,2009:91-100.

[8] LIU D, LIANG D. Three-way decisions in ordered decision system [J]. *Knowledge-Based Systems*, 2017,137(12):182-195.

[9] Ma Y, Luo G, Zeng X, *et al*.. Transfer learning for cross-company software defect prediction [J]. *Information & Software Technology*, 2012, 54(3):248-256.

[10] ZHANG, FENG, MOCKUS, *et al*.. Towards building a universal defect prediction model [J]. *Empirical Software Engineering*, 2016, 21(5):2107-2145.

[11] KAMEI Y, FUKUSHIMA T, MCINTOSH S, *et al*.. Studying just-in-time defect prediction using cross-project models [J]. *Empirical Software Engineering*, 2016, 21(5):2072-2106.

[12] ZHANG F, ZHENG Q, ZOU Y, *et al*.. Cross-project defect prediction using a connectivity-based unsupervised classifier [C]. *Ieee/acm, International Conference on Software Engineering. IEEE*, 2017:309-320.

[13] KHOSHGOGTAAR T M, CUKIC B, SELIYA N. An empirical assessment on program module-order models [J]. *Quality Technology & Quantitative Management*, 2007, 4(2):171-190.

[14] ZIMMERMANN T, PREMRAJ R, ZELLER A. Predicting defects for eclipse [C]. *International Workshop on Predictor MODELS in Software Engineering*, 2007. *Promise'07: ICSE Workshops. IEEE*, 2007:9.

[15] PENG L, YANG B, CHEN Y, *et al*.. Data gravitation based classification [J]. *Information Sciences An International Journal*, 2009, 179(6): 809-819.

**作者简介:**

**哈清华**(1983－),男,吉林长春人,回族,助理研究员,博士研究生,2007 年于哈尔滨工业大学获得硕士学位,主要从事软件测试方面的研究。E-mail: haqinghuaha@ hotmail.com

**导师简介:**

**刘大有**(1942－),男,河北人,博士生导师,教授,1981 年于吉林大学获得硕士学位。主要从事人工智能、数据挖掘的研究。E-mail:liudy@ jlu.edu.cn