**摘要**

软件缺陷定位通过定位程序中的缺陷所在位置为调试提供关键信息，从而提高软件调试和修复的效率，是一种重要的软件质量保障技术。在此前的研究中，基于频谱、变异和切片的软件缺陷定位方法已取得了较好的效果。然而，上述方法都存在大量语句可疑度相同的问题，需要测试人员人工确认，影响缺陷定位的效率和精确度。例如：使用Ochiai和DStar方法对Defects4J 中6个项目的395个存在缺陷的版本进行缺陷定位，都有近70%的版本存在多条语句可疑度并列排名第1。针对上述问题，本文提出了一种通过语句复杂度分析提升缺陷定位精确度的方法CBFL（Complexity-based fault localization）。首先，使用SBFL、MBFL等传统方法进行缺陷定位，得到程序中语句的可疑度排名；然后，针对排名靠前的语句进行复杂度分析，提取特征以及类别标签，构建数据集；最后，构造基于语句复杂度特征的缺陷倾向性分类模型，预测排名靠前的语句中更有可能存在缺陷的位置。基于上述方法，本文实现了基于语句复杂度分析的缺陷定位工具，并在真实的缺陷数据集 Defects4J 项目上进行了仿真实验，分别与基于频谱的缺陷定位方法DStar和基于变异的缺陷定位方法MUSE进行了对比，仿真结果表明本文所提出的方法具有更好的缺陷定位效果。

1. **引言**

近年来，随着互联网与信息技术的飞速发展，各种软件产品的数量和规模与日俱增。各行各业对软件产品的需求越来越迫切，也越来越复杂。当软件产品的数量和规模日趋庞大，发现和修复软件中缺陷的难度也就逐步增加。而软件中包含的缺陷数目与软件产品的质量息息相关，软件中包含的缺陷数量越多，软件可靠性面临的挑战就越大。为了保证软件的可靠性，在软件项目正式发布上线之前，通常要经过科学且系统的软件测试。软件开发及测试人员需要精准找到缺陷所在位置并进行修复，直至达到上线标准才可正式发布，这个过程往往会消耗掉大量的人力、物力与时间。而一个公认的事实，即手动调试不仅是耗时、冗长和昂贵的，而且也是相当容易出错的。

软件缺陷定位通过定位程序中的缺陷所在位置为调试提供关键信息，从而提高软件调试和修复的效率，是一种重要的软件质量保障技术。在此前的研究中，基于频谱、变异和切片的软件缺陷定位方法已取得了较好的效果。然而，这些方法都面临一个十分关键的挑战，即缺陷定位的效率和精度不高。例如，使用Ochiai和DStar方法对Defects4J 中6个项目的395个存在缺陷的版本进行缺陷定位，都有近70%的版本存在多条语句可疑度并列排名第1。当上述技术对测试软件进行缺陷定位后，结果往往存在大量语句可疑度相同，难以区分的问题。这时候的定位结果仍然需要耗费大量人力进行人工确认，而这无疑加重了软件开发过程中成本方面的负担。因此，研究出有效提升软件缺陷定位精确度的方法，用于提升软件缺陷定位工作的效率是十分必要的。

在本文中，我们提出了一种通过语句复杂度分析提升缺陷定位精确度的方法CBFL （Complexity-based fault localization）。通常复杂度被应用于数学和计算机导论中，用于计算可执行程序运行时所需要的时间资源和内存资源。而在本论文中，我们要提出一种新的语句复杂度，以程序中每一行代码为单位，计算其复杂度作为判断缺陷语句的条件。据我们所知，这样的系数在之前的研究中还未被提出过（尤其在故障定位的背景下）。

A.解决方案（PROPOSED SOLUTION）

本文实现了基于语句复杂度分析的缺陷定位工具，并在真实的缺陷数据集 Defects4J 项目上进行了仿真实验，对每个经过缺陷定位的程序结果进行筛选，提高定位的有效性。首先使用SBFL、MBFL等传统方法进行缺陷定位，得到程序中语句的可疑度排名；然后，针对排名靠前的语句进行复杂度分析，提取特征以及类别标签，构建数据集；最后，构造基于语句复杂度特征的缺陷倾向性分类模型，预测排名靠前的语句中更有可能存在缺陷的位置。我们通过使用*Einspect*@*n*（n=1,3）和EXAM来印证本文所提出方法的有效性。

***B.* CONTRIBUTIONS**

文的主要贡献可以总结如下。

**·** 提出一种新的方法CBFL，通过语句复杂度分析提升缺陷定位精确度。

**·** 提出一种新的语句复杂度计算方法，以程序中每一行代码为单位，提取语句特征并构造缺陷倾向性分类模型

**·** 在Defects4J 基准测试上对CBFL与SBFL和MBFL进行的实验评估。

本文的其余部分组织如下。第2节概述了基于语句复杂度分析的缺陷定位工具CBFL，并再次论述同一缺陷定位结果中可能包含多个可疑度相同的语句以及训练分类模型的目的。第3节介绍了我们的实验研究和模型评估，并把方法设计中每个实验步骤，陈述目标展示执行细节，根据评估标准与SBFL和MBFL进行对比。第4节展示了本文提出的方法如何提高了缺陷定位结果的有效性，以及方法的有效威胁和未来发展。第5节描述了与本文中提出的主题相关的工作。最后，我们在第6节中提供了我们对本文工作的结论。

**介绍背景，（为什么会有可疑度相同的问题**

**介绍本文贡献及组成**

1. **方法设计（Study Design）**

**介绍本文总体思路，再次说明效率问题**

**2.1第一步：使用SBFL做评估**

**使用SBFL,介绍SBFL原理和Dstar使用的公式**

**/使用MBFL，介绍MBFL原理和MUSE使用的公式**

**2.2第二步：结合产生数据构造模型**

**介绍机器学习的分类算法**

**介绍选择了决策树并进行训练，要达到一个什么效果**

1. **实验和评估（CFBL EFFECTIVENESS）**

**提出问题，并把方法设计中每个实验，陈述目标展示执行细节**

**研究性问题（Research Questions）**

**3.1数据集（Datasets）**

**Defects4J**

**生成的语句可疑度排名集合**

**3.2特征提取/数据预处理——**语句复杂度分析，提取度量元

**3.3平衡数据集，提出问题-引出解决方法——SMOTE**

**3.4**构建决策树分类器

**3.5介绍测量指标，**十折交叉验证

**3.6介绍对真实缺陷的适用性**

1. **讨论（DISCUSSION）**

**根据研究结果表明，CBFL对。。。有效，讨论结果的有效性和方法的局限性**

把CBFL作为补充测试技术

* 1. 威胁有效性**Threats to Validity**
  2. **局限和未来工作**

**数据集的局限性**

**未来是否可以把这个思想投入到APR工作中去**

1. **相关工作（RELATED WORK）**

**介绍最近的且比较相似的缺陷定位工作**

1. **结尾（CONCLUSION）**