基于深度学习的空指针引用缺陷检测系统的设计与实现

罗辉

2018 年 5 月

中图分类号：TQ028.1

UDC分类号：540

基于深度学习的空指针引用缺陷检测系统的设计与实现

作 者 姓 名　罗辉

学 院 名 称　软件学院

指 导 教 师　田东海教授

答辩委员会主席　计卫星教授

申 请 学 位　工学硕士

学 科 专 业　软件工程

学位授予单位　北京理工大学

论文答辩日期　2018 年 5 月

Design and implementation of null pointer reference defect detection system based on deep learning

Candidate Name:	Hui Luo
School or Department:	Software Institute
Faculty Mentor:	Prof. Donghai Tian
Chair, Thesis Committee:	Prof. **
Degree Applied:	Master of Science
Major:	Software engineering
Degree by:	Beijing Insititute of Technology
The Date of Defence:	June，2018

基于深度学习的空指针引用缺陷检测系统的设计与实现

北京理工大学

研究成果声明

本人郑重声明：所提交的学位论文是我本人在指导教师的指导下进行的研究工作获得的研究成果。尽我所知，文中除特别标注和致谢的地方外，学位论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京理工大学或其它教育机构的学位或证书所使用过的材料。与我一同工作的合作者对此研究工作所做的任何贡献均已在学位论文中作了明确的说明并表示了谢意。

特此申明。

作者签名：_____　　签字日期：_____

关于学位论文使用权的说明

本人完全了解北京理工大学有关保管、使用学位论文的规定，其中包括：① 学校有权保管、并向有关部门送交学位论文的原件与复印件；② 学校可以采用影印、缩印或其它复制手段复制并保存学位论文；③ 学校可允许学位论文被查阅或借阅；④ 学校可以学术交流为目的, 复制赠送和交换学位论文；⑤ 学校可以公布学位论文的全部或部分内容（保密学位论文在解密后遵守此规定）。

作者签名：_____　　导师签名：_____

签字日期：_____　　签字日期：_____

摘要

空指针引用是程序中比较常见的缺陷之一，研究表明该缺陷在编译后的程序中大量存在，因此给软件的稳定性带来很大威胁。出于对检测效率和精度的平衡，现有工具在工作原理和检测范围等方面各不相同，无法全面检测该类缺陷，大量的误报也降低了此类工具的实用价值。本课题基于 Soot 框架实现一种 Java 代码空指针引用缺陷检测工具，弥补了同类工具在某些方面的不足。此外还提出一种多个工具交叉验证的空指针引用缺陷检测方法，并在 SonarQube 平台上进行实践。

关键词： 空指针引用；静态检测；交叉验证；

Abstract

In order to exploit ⋯⋯.

Key Words: shape memory properties; polyurethane;textile;synthesis;application

目录

第 1 章　绪论

1.1　研究背景

随着互联网的迅猛发展，各种应用软件日益增多，软件规模越来越大，但是这些软件背后的软件安全及其稳定性的问题也日益突出。特别是当下人工智能技术的快速成熟，软件的智能化和自动化程度都上升到了新的高度，在越来越多的无人场景下，如无人驾驶，智慧医疗等领域，软件正逐步完全替代人类在一些重要领域的工作，这就对软件的安全性 [1] 提出了更高的要求。

任何的软件设计或者编码产生的安全漏洞 [2]，都可能成为潜在的安全隐患，可能会给社会带来巨大的损失，信息安全问题早就成为人们日益关注的焦点 [3]。近年来，重大网络安全事件层出不穷，如 2017 年 5 月，史上规模最大的一次勒索病毒攻击事件爆发，全球近百个国家的网络遭遇 Wannacry 病毒 [4] 的攻击，电脑被该病毒感染后文件会被加密锁定，支付黑客索要的赎金后才能解密恢复，受攻击对象甚至包括医院、高校等公益性机构。2013 年 6 月，前美国中情局 (CIA) 雇员斯诺登曝出一项由美国国家安全局 (NSA) 实现的棱镜计划 [5]，震惊全球。据斯诺登披露的资料显示，NSA 通过植入恶意软件感染了全球超过 5 万台计算机，用于窃取敏感信息；

由于软件的复杂性随着软件的规模和数量不断增大，软件开发的规模和数量不断增大，软件开发的难度也在增大，导致在开发过程中存在某些不确定性的错误或缺陷。另一方面软件开发人员的水平参差不齐，即使是富有经验的开发者在开发过程中也难以避免引入一些错误或缺陷。如图 1.1 所示，Eclipse3.0.1 中也会存在着一些明显的空指针引用错误 [6]。公开数据显示，对于有经验的程序员编写的代码，每 1000 行就有 50-250 个错误，平均每 1000 行会有 100 个缺陷，即使是经过软件故障控制管理培训的软件工程师，平均每 1000 行代码中存在 50 个故障 [7]。因此，整个行业在关注软件如何提升生产力的同时，也更加注重提高软件源代码编写的质量，加大了对源代码的检测力度，以期及早发现代码中潜在的安全隐患，避免或减少因为软件缺陷带来的损害。据统计，现在在软件开发总成本中，投入到软件测试中的资源约占到 25% 到 50%[8]，并贯穿在软件生命周期的各个阶段。

在数量繁多的软件缺陷中，空指针引用缺陷是相对比较常见的缺陷类型，广泛存在于不同的编程语言中。同时，它也是最影响软件系统可靠性和稳定性的故障之一。

```
// Eclipse 3.0.1
if (in == null)
  try {
     in.close();
  } catch (IOException e) {}
```

图 1.1　Eclipse3.0.1 中存在的空指针引用缺陷

通过人们对故障的总结，人们发现常见的比较大型的软件故障有数组越界，资源泄漏，空指针引用等，其中空指针引用问题出现的尤为频繁，根据 coverity 公司 2009 年针对 280 个开源项目的故障分析报告，空指针引用在所有类故障中所占比例为 27.81%，是所占比例最高的故障 [11]。

中国国家信息安全漏洞库 (CNNVD) 统计，2013 年共发现空指针引用引发的漏洞共 35 个，这些漏洞存在于操作系统、服务器应用程序等软件系统中，漏洞类型有拒绝服务、代码注入、信息泄露、一区溢出、数字错误等。这些漏洞一旦被恶意攻击者利用, 可能导致系统崩溃，服务器程序可能会拒绝服务, 或者机密信息泄露，这都将严重的影响软件的运行以及系统的安全。根据对国内航空航天、武器装备、金融、电信等数千万行国产软件应用 DTSC 的测试报告统计，在所有故障类缺陷中，空指针引用缺陷大约会占到 30% 左右，空指针引用缺陷的密度大致是 0.3/KLOC。

由此可见，空指针引用缺陷的清除对于程序的稳定性和安全性都具有巨大价值，而针对空指针引用缺陷的检测技术研究也就具备了重大的意义。

1.2　国内外研究现状及发展趋势

软件缺陷检测相关的研究几乎是伴随着软件的产生而出现的，随着程序设计语言的发展，软件缺陷类型也越来越多。通过人们对故障的总结，人们发现常见的比较大型的软件故障有数组越界，资源泄漏，空指针引用等，其中空指针引用问题出现的尤为频繁，根据 coverity 公司 2009 年针对 280 个开源项目的故障分析报告，空指针引用在所有类故障中所占比例为 27.81%，是所占比例最高的故障 [11]。

以 Java 语言为例，Null 关键字的广泛使用是 Java 代码中产生 NPE（Null Pointer Exception）故障的直接原因，Haidar Osman[12] 等人开发了 NullTracker 工具对 810 个开源 Java 项目进行简单的数据流分析，追踪代码中空指针检查语句的分布情况，以发现程序开发者使用 null 关键字的时机和目的，结果表明在所有的条件判断语句中，用

来进行空指针检查的语句平均占比为35%，类成员没有初始化，方法返回null值，以及向方法中传递null值是引发NPE的最常见的因素。其中，71%的空指针检查语句用来保证方法调用返回值的安全性。由于空指针检查语句的频繁使用，可能导致程序运行时的开销增加2%-10%，不仅如此，空指针检查的频繁使用还会降低代码的可读性和可维护性，而一旦缺失了这种检查，程序的稳定性便无法得到保障。

由于空指针引用故障在代码中广泛存在而又十分隐蔽，但是其一旦出现很大可能会导致程序崩溃，因此对程序的稳定性具有非常大的威胁。针对空指针引用缺陷的检测一直备受关注。

目前，代码缺陷检测的技术从大的层面上主要分为两大类，静态检测和动态检测。

动态检测 [9] 主要侧重于软件的性能、功能完善等方面，通过动态测试来进行漏洞的探测与发现，不仅仅要求测试人员对缺陷特性具有较深入的理解，测试过程中还需要大量测试用例，在目前软件规模愈加庞大，逻辑愈加复杂的情境下，这种方式必然带来大量的人力和物力的浪费。

静态检测 [10] 是指利用静态分析手段来探测程序中潜在缺陷的方法，不需要运行程序使用其他手段完成对程序结构分析的技术。相较于动态分析技术而言，静态分析成本较低，而且能有效的对代码中的缺陷进行精确定位。因此，对源代码的静态分析和缺陷检测是一个值得深入研究的方向。

1.3 本文研究内容

静态分析技术与工具具有较早发现缺陷、覆盖率高、低开销、自动化程序高等优点；同时静态分析技术也存在一定的局限性，不仅要在分析效率与精度中做出取舍，还需要在误报率与漏报率之间做出取舍。现有的一些静态检测工具（如Findbugs,PMD）都采用了不同的实现方法达到这样的平衡，由于采取的分析策略的不同，形成的检测结果也往往有较大差异。

基于当下Java代码空指针引用缺陷检测工具的特点，本文提出了利用交叉验证的方式整合不同工具的检测能力，以提高检测结果准确率的方法。研究内容如下：

(1) 开发多工具缺陷代码检测系统BIT-Detector，将多种Java空指针引用缺陷静态缺陷检测工具以插件的形式集成在SonarQube平台上，针对目标代码进行一次性地同时检测，将检测结果按照优先级排序，即缺陷被检测出的工具数量越多，排序越靠前，缺陷真实的可信度越高。

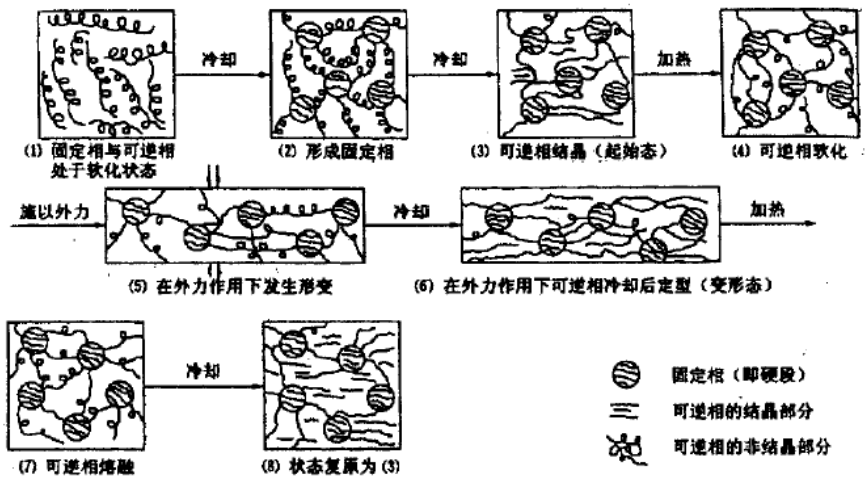

图 1.2　热塑性形状记忆聚氨酯的形状记忆机理示意图

(2) 提取缺陷代码特征，将缺陷代码的控制流图转换成向量，利用深度学习的方法搭建神经网络模型帮助选择目标代码适合的工具，进一步优化 BIT-Detector 的检测结果。

1.4　论文结构

1.4.1　形状记忆聚氨酯的形状记忆机理

形状记忆聚合物（SMP）是继形状记忆合金后在 80 年代发展起来的一种新型形状记忆材料[1]。形状记忆高分子材料在常温范围内具有塑料的性质，即刚性、形状稳定恢复性；同时在一定温度下（所谓记忆温度下）具有橡胶的特性，主要表现为材料的可变形性和形变恢复性。即“记忆初始态－固定变形－恢复起始态”的循环。

固定相只有物理交联结构的聚氨酯称为热塑性 SMPU, 而有化学交联结构称为热固性 SMPU。热塑性和热固性形状记忆聚氨酯的形状记忆原理示意图如图1.2所示

1.4.2　形状记忆聚氨酯的研究进展

首例 SMPU 是日本 Mitsubishi 公司开发成功的……。

表 1.1　水系聚氨酯分类

类别	水溶型	胶体分散型	乳液型
状态	溶解 ~ 胶束	分散	白浊
外观	水溶型	胶体分散型	乳液型
粒径 $/\mu m$	< 0.001	$0.001 - 0.1$	> 0.1
重均分子量	$1000 \sim 10000$	数千 ~ 20□	> 5000

1.4.3　水系聚氨酯及聚氨酯整理剂

水系聚氨酯的形态对其流动性，成膜性及加工织物的性能有重要影响，一般分为三种类型[1]，如表 1.1所示。

由于它们对纤维织物的浸透性和亲和性不同，因此在纺织品染整加工中的用途也有差别，其中以水溶型和乳液型产品较为常用。另外，水系聚氨酯又有反应性和非反应性之分。虽然它们的共同特点是分子结构中不含异氰酸酯基，但前者是用封闭剂将异氰酸酯基暂时封闭，在纺织品整理时复出。相互交联反应形成三维网状结构而固着在织物表面。……

第 2 章　相关工作

为了保证软件的可靠性和稳定性，在大量研究人员长期不懈地努力下，出现了很多针对软件缺陷的检测方法。这些方法可以在软件开发周期的不同阶段介入，检测的效率和效果也大不相同，最终涌现出了一批相对成熟的代码缺陷检测方法和工具。另一方面，随着软件数量的日益庞大，以及数据挖掘技术在各个研究领域的广泛应用，利用机器学习的方式来解决软件安全问题也逐渐成为了研究热点。

2.1　程序静态分析技术

静态分析技术即是在不运行程序，不依赖程序输入的情况下对程序代码进行分析的一项技术。这种技术有助于开发人员对代码结构的理解，同时也能检测潜在的安全缺陷 (如 SQL 注入)，运行时错误 (如空指针引用缺陷) 以及部分代码逻辑错误。它一般需要配合利用自动化工具执行分析。采用的技术有数据流分析，机器学习，语义精简等。可检测死锁，空指针，资源泄露，缓存区溢出，安全漏洞，竞态条件等软件缺陷。具有快速，准确，伸缩性强等特点，能够在代码开发阶段找到并修复多种问题，从而节省大量人力成本和时间。

针对空指针引用缺陷，研究人员已经利用静态分析技术做出了很多实践并取得了一定成果。

针对 Java 语言来说，Haidar Osman 等人利用数据流分析技术考察了 Java 语言空指针缺陷产生的情况，研究表明 Null 关键字的广泛使用是 Java 代码中产生 NPE（Null Pointer Exception）故障的直接原因。他们开发了 NullTracker[12] 工具对 810 个开源 Java 项目进行简单的数据流分析，追踪代码中空指针检查语句的分布情况，以发现程序开发者使用 null 关键字的时机和目的，结果表明在所有的条件判断语句中，用来进行空指针检查的语句平均占比为 35%，类成员没有初始化，方法返回 null 值，以及向方法中传递 null 值是引发 NPE 的最常见的因素。其中，71% 的空指针检查语句用来保证方法调用返回值的安全性。由于空指针检查语句的频繁使用，可能导致程序运行时的开销增加 2%-10%，不仅如此，空指针检查的频繁使用还会降低代码的可读性和可维护性，而一旦缺失了这种检查，程序的稳定性便无法得到保障。

研究人员利用静态分析技术在 Java 空指针引用缺陷上进行了大量的工作，产生

了很多检测空指针引用的工具和技术，这些技术可粗略的分为指针引用验证和空指针引用 [13] 缺陷检测两大类。前者侧重于如何验证程序中的指针是否为空。后者侧重于如何尽可能多的发现程序中的空指针引用。指针引用验证技术是基于需求驱动的思想[14]，一般是首先识别出指针，再沿着控制流后向的验证指针是否为空。空指针引用缺陷检测一般是在进行数据流分析 [15]、指针分析的基础上，根据一些规则基于控制流前向的检测。两者通常都需要进行数据流分析与指针分析。

Salsa[16] 是一个致力于验证 Java 代码中指针引用安全性验证的工具，通过定制的数据表示形式进行前向数据流分析，通过对传播深度和数据流传播路径数量的简单限制来获得方法的可扩展性，同时依赖预先进行的必然别名分析来提高方法间数据流分析的准确性。由于一些空指针的引用需要经过多层方法调用链才有可能触发，这种验证方式会产生很多漏报的同时，效率也不理想。数据流分析技术具有十分灵活的特点，为了提高效率，Ravichandhran Madhavan[17] 等提出了一种过近似的最弱前置条件分析方法以验证 Java 程序中指针的安全性，该方法通过需求驱动的前向数据流分析大幅提升了单个引用的分析效率，该方法试图找到程序入口处可能满足被分析程序点的引用不安全的条件，如果存在这样的条件，则可以判定该引用不安全。此方法的数据流事实为有限的谓词集合，通过有选择地限制谓词集合的大小以及传播路径的数量，该方法可以做到低延迟的流敏感，上下文敏感的 sound 分析，利用 Wala[18] 程序分析框架，可以取得较好的验证引用安全的效果，但是过于追求针对单个引用的需求驱动分析，在对大规模代码中的引用进行批量分析时性能欠佳。

空指针检测相比于指针安全性验证更加具有实用性，而误报率和漏报率是检验工具实用性的重要指标，空指针检测工具大多不追求完美的正确率，而将较低的误报率和较高的召回率作为最重要的目标。

FindBugs[6][19] 是一个开源的针对 Java 代码的缺陷静态检测工具，通过分析 class 文件，在字节码层级进行简单的前向数据流分析，对程序中的每一个引用的是否为 null 值的不同情况，给定相应的标识从而在触发可能的空指针调用时给出不同的告警等级。对于指针引用 FindBugs 总结出了一些经验规则，对不可达路径、控制流汇合、指针赋值语句、断言等特定情况定制了专用的检测规则，在进行过程间分析时，其主要依赖特定故障模式以及用户编码时给出的注解来推断空指针是否可能发生，所以它只能检测出特定场景下的空指针引用。

检测工具 Xylem[20] 从每一个指针引用出发，进行基于需求驱动的后向数据流分

析，并将谓词作为数据流事实，目标是能够高效的检测出最重要的空指针引用，在进行分析时采取的是不完全可靠的分析方法，检测结果存在较多漏报。

北京邮电大学的杨睿 [21] 提出一种 Java 中空指针引用故障的静态检测方法，将空指针引用问题抽象为一类故障模型，并以故障模式状态机来形式化描述此类故障模型，然后根据故障状态机的创建条件及待检测代码的语义信息确定是否创建该类型的状态机，并将创建的状态机示例置于控制流图入口，根据数据流分析的结果对故障状态进行迭代以检测空指针引用问题。

中国矿业大学的姜淑娟 [22] 提出一种空指针异常自动定位方法，该方法结合程序的静态分析技术，利用程序运行时的堆栈信息指导程序切片，然后对得到的切片进行空指针分析及别名分析，得出引发空指针异常的可疑语句集合，最终给出错误定位报告。

总体来看，以上这些分析方法都有各自的优缺点，但是目前无法找到一种完美的静态检测方法可以兼顾缺陷检测的误报率和漏报率。这也正是静态代码分析的短处，不仅是将复杂的缺陷解释出来很困难，对于结果的高误报率，往往显得无能为力。

2.2 程序动态测试技术

2.3 代码缺陷检测工具介绍

2.4 深度学习技术在软件安全领域的应用

结论

（结论作为学位论文正文的最后部分单独排写，但不加章号。结论是对整个论文主要结果的总结。在结论中应明确指出本研究的创新点，对其应用前景和社会、经济价值等加以预测和评价，并指出今后进一步在本研究方向进行研究工作的展望与设想。结论部分的撰写应简明扼要，突出创新性。）本文采用……。（结论作为学位论文正文的最后部分单独排写，但不加章号。结论是对整个论文主要结果的总结。在结论中应明确指出本研究的创新点，对其应用前景和社会、经济价值等加以预测和评价，并指出今后进一步在本研究方向进行研究工作的展望与设想。结论部分的撰写应简明扼要，突出创新性。）

参考文献

[1] 姜敏, 彭少贤, 郦华兴. 形状记忆聚合物研究现状与发展 [J]. 现代塑料加工应用, 2005, 17 (2): 53–56.

附录 A　***

附录相关内容…

攻读学位期间发表论文与研究成果清单

[1] 高凌. 交联型与线形水性聚氨酯的形状记忆性能比较 [J]. 化工进展, 2006, 532 一 535.（核心期刊）

致谢

本论文的工作是在导师……。

作者简介

本人…。