(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 110443045 A
(43)申请公布日 2019.11.12

(21)申请号 201910742264.7

(22)申请日 2019.08.13

(71)申请人 北京计算机技术及应用研究所
地址 100854 北京市海淀区永定路51号

(72)发明人 赵磊 贾琼 常承伟 刘滋润 杨枭 张宏星

(74)专利代理机构 中国兵器工业集团公司专利中心 11011
代理人 王雪芬

(51)Int.Cl.
G06F 21/57(2013.01)

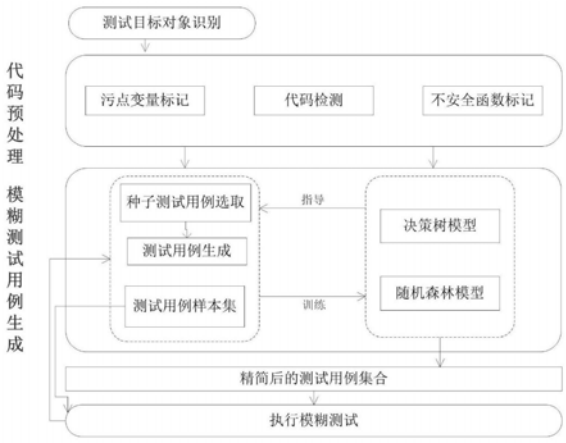
权利要求书2页 说明书5页 附图2页

(54)发明名称

一种基于机器学习方法的模糊测试用例生成方法

(57)摘要

本发明涉及一种基于机器学习方法的模糊测试用例生成方法，涉及信息安全领域。本发明对目前模糊测试技术存在的测试用例冗余问题优化设计，在面向源程序文件的模糊测试用例生成方面，通过在模糊测试用例生成前，标记识别程序对象中的污点变量和问题函数，结合对已有的种子用例生成、筛选技术，可提升模糊测试用例的有效性，降低模糊测试用例集合的冗余度。其中，在测试用例生成环节，结合机器学习，分析机器学习用于测试用例精简的可行性，得到机器学习的测试用例生成优化技术思路，采用机器学习的模型和算法，改进模糊测试流程中的测试用例生成环节，提升测试用例的生成效率，实现测试用例结合的去冗余，达到提高模糊测试流程智能化程度的目标。



CN 110443045 A

1.一种基于机器学习方法的模糊测试用例生成方法,其特征在于,包括代码预处理环节和模糊测试用例生成环节;

所述代码预处理环节是采用代码检测技术、污点标记技术和不安全函数标记技术,对待测软件程序进行预处理和分析,标识出可疑语句、污点变量和不安全函数这些疑似脆弱性代码;

所述模糊测试用例生成环节是利用机器学习中的决策树模型和随机森林模型寻找出种子用例集合,使用变异的方式产生大批量的测试用例,并使用训练后的分类预测器去除同类安全漏洞下多余的测试用例。

2.如权利要求1所述的方法,其特征在于,其中,所述代码预处理环节包括代码检测步骤、污点标记步骤和不安全函数标记步骤,所述代码检测步骤是进行逻辑错误和表达式类型的检测,污点标记步骤是进行程序变量的检查和标识,不安全函数标记步骤是标识代码检测步骤、污点标记步骤这两个步骤遗漏的疑似问题函数。

3.如权利要求2所述的方法,其特征在于,所述代码检测步骤通过静态语句标记技术,对待测软件程序只进行可疑语句标记操作。

4.如权利要求3所述的方法,其特征在于,所述污点标记步骤通过静态污点标记技术,对代码检测步骤输出的软件程序进行污点标记,标记时根据待测软件程序的数据输入格式和要检测的安全漏洞特征,对不信任的输入数据和程序中的可疑变量做标记。

5.如权利要求4所述的方法,其特征在于,所述不安全函数标记步骤识别并标记出污点标记步骤输出的软件程序中存在的不安全函数,对可能出现不安全调用方式的函数进行标识。

6.如权利要求5所述的方法,其特征在于,将代码检测、污点标记和不安全函数标记这三个步骤的执行顺序替换为并行处理,并对处理后的三个结果进行去重处理。

7.如权利要求6所述的方法,其特征在于,所述模糊测试用例生成环节具体包括确定种子用例的生成模式步骤和模糊测试用例生成步骤;

其中,所述确定种子用例的生成模式步骤是根据代码预处理标记出的问题语句、污点变量、不安全函数这些可疑代码构造测试用例的生成模式;

所述模糊测试用例生成步骤是根据种子用例的生成模式,对其中字段的取值种类进行组合,生成一批用于模糊测试的初始测试用例,将所述初始测试用例记为初始集合;然后,将所述初始集合中的初始测试用例输入到待测软件程序中,动态观测执行效果,记录下触发安全漏洞的情况,并根据触发效果判定种子用例的生成模式是否合理准确,在需要时进行生成模式的调整;使用机器学习中的决策树模型和随机森林模型构造分类预测器,并将调整后的用例生成模式作为分类依据,再对所述初始集合中的测试用例进行分类学习,训练出满足预设准确率要求的分类预测器;然后,根据修改后的用例生成模式,生成多批量的测试用例作为分类预测器的测试数据集合;接着,使用构造好的分类预测器对所述测试数据集合进行分类操作,去除安全漏洞触发效果相同的测试用例数据。

8.如权利要求7所述的方法,其特征在于,所述确定种子用例的生成模式步骤中,确定种子用例的生成模式格式如下:

<vars,exps,ops,func,judge_num,var_num,func_num,env,conf>

其中,vars指针,指向污点变量数组,用来存放污点变量,数组长度根据污点标记时记

录的污点变量个数而定;exps指针,指向表达式类型数组,用来存放可疑语句的类型数据,数组长度根据代码检测时记录的语句个数来确定;ops指针,指向运算符类型数组,用来存放可疑语句中存在的运算符类型;func指针,指向不安全库函数数组;judge_num指待测软件程序中存在的判定语句个数,所述判定语句个数只记录判定语句的个数,并不记录判定分支间的嵌套层次结构;var_num是指待测软件程序中的污点变量数量;func_num是指待测软件程序中已标记出的不安全函数的数量;env指针,指向环境信息字段;conf指针,指向配置信息字段。

9.如权利要求7所述的方法,其特征在于,使用机器学习中的决策树模型和随机森林模型构造分类预测器时,采用增加分类特征和调整参数的方法处理欠拟合问题。

10.如权利要求7所述的方法,其特征在于,使用机器学习中的决策树模型和随机森林模型构造分类预测器时,还对分类特征数进行限制,同时尽可能扩展训练数据集合以降低过拟合的概率。

一种基于机器学习方法的模糊测试用例生成方法

技术领域

[0001] 本发明涉及信息安全技术领域，具体涉及一种基于机器学习方法的模糊测试用例生成方法。

背景技术

[0002] 根据国家信息安全漏洞库(CNNVD)公布的安全漏洞数据统计，我国2018年公布的安全漏洞数量为23029个，与2017年的安全漏洞总数18586个相比，年增长率约为23.9%。与2017年的安全漏洞数量激增的趋势相比，2018年安全漏洞数量增速放缓。然而，这并不意味着安全漏洞预防工作已经取得了令人欣喜的成果，因为造成这一现象的主要原因是安全漏洞统计以及公布较以往更加分散化，大量的安全漏洞并没有经官方收录和公布。此外，安全漏洞逐渐被视为重要的战略资源，从而限制了安全漏洞公布的数量以及时间。网络安全研究中的一个核心内容就是安全漏洞相关的研究。安全漏洞的存在，使得恶意攻击者可以实现对网络空间中资源的非法访问，甚至破坏。在当今的网络生态系统中，由于安全漏洞造成的危害涉及范围越来越广，导致的结果越来越严重，众多研究机构和人员对安全漏洞检测技术开展了深入的研究工作。

[0003] 作为可以安全漏洞检测的有效手段，模糊测试技术对科研和生产都具有重要的安全防护意义。由于易用性高、成本较低以及检测效果较好等优势，模糊测试技术已成为目前业界常用的安全漏洞检测技术之一，针对网络协议、门户网站、关键信息系统等目标，已经开展了众多研究和应用。但目前模糊测试技术仍存在以下主要问题：

[0004] (1)从所针对的目标范围来看，模糊测试技术的目标对象类型覆盖范围有待提高。例如，针对源代码程序文件的模糊器，所支持的源代码程序语言种类数一直都是制约其广泛应用的因素之一。此外，针对不同类型的软件程序，同一个模糊器经常出现检测水平的波动。

[0005] (2)测试用例生成环节一直都是模糊测试过程中至关重要的一个步骤，测试用例的有效性直接影响最后的测试结果是否准确。目前的主流模糊测试技术在测试用例生成阶段的用例冗余较严重、成本偏高，且易出现成本不可控等问题。

[0006] (3)在安全漏洞的类型覆盖方面，模糊测试技术仍有待提高。虽然目前的模糊测试技术可以有效地发现某些类型的安全漏洞，但是对一些特定的安全漏洞却无能为力。

发明内容

[0007] (一)要解决的技术问题

[0008] 本发明要解决的技术问题是：如何实现一种新的模糊测试用例生成方法，提升模糊测试用例的有效性，降低模糊测试用例集合的冗余度。

[0009] (二)技术方案

[0010] 为了解决上述技术问题，本发明提供了一种一种基于机器学习方法的模糊测试用例生成方法，包括代码预处理环节和模糊测试用例生成环节；

[0011] 所述代码预处理环节是采用代码检测技术、污点标记技术和不安全函数标记技术，对待测软件程序进行预处理和分析，标识出可疑语句、污点变量和不安全函数这些疑似脆弱性代码；

[0012] 所述模糊测试用例生成环节是利用机器学习中的决策树模型和随机森林模型寻找出种子用例集合，使用变异的方式产生大批量的测试用例，并使用训练后的分类预测器去除同类安全漏洞下多余的测试用例。

[0013] 优选地，其中，所述代码预处理环节包括代码检测步骤、污点标记步骤和不安全函数标记步骤，所述代码检测步骤是进行逻辑错误和表达式类型的检测，污点标记步骤是进行程序变量的检查和标识，不安全函数标记步骤是标识代码检测步骤、污点标记步骤这两个步骤遗漏的疑似问题函数。

[0014] 优选地，所述代码检测步骤通过静态语句标记技术，对待测软件程序只进行可疑语句标记操作。

[0015] 优选地，所述污点标记步骤通过静态污点标记技术，对代码检测步骤输出的软件程序进行污点标记，标记时根据待测软件程序的数据输入格式和要检测的安全漏洞特征，对不信任的输入数据和程序中的可疑变量做标记。

[0016] 优选地，所述不安全函数标记步骤识别并标记出污点标记步骤输出的软件程序中存在的不安全函数，对可能出现不安全调用方式的函数进行标识。

[0017] 优选地，将代码检测、污点标记和不安全函数标记这三个步骤的执行顺序替换为并行处理，并对处理后的三个结果进行去重处理。

[0018] 优选地，所述模糊测试用例生成环节具体包括确定种子用例的生成模式步骤和模糊测试用例生成步骤；

[0019] 其中，所述确定种子用例的生成模式步骤是根据代码预处理标记出的问题语句、污点变量、不安全函数这些可疑代码构造测试用例的生成模式；

[0020] 所述模糊测试用例生成步骤是根据种子用例的生成模式，对其中字段的取值种类进行组合，生成一批用于模糊测试的初始测试用例，将所述初始测试用例记为初始集合；然后，将所述初始集合中的初始测试用例输入到待测软件程序中，动态观测执行效果，记录下触发安全漏洞的情况，并根据触发效果判定种子用例的生成模式是否合理准确，在需要时进行生成模式的调整；使用机器学习中的决策树模型和随机森林模型构造分类预测器，并将调整后的用例生成模式作为分类依据，再对所述初始集合中的测试用例进行分类学习，训练出满足预设准确率要求的分类预测器；然后，根据修改后的用例生成模式，生成多批量的测试用例作为分类预测器的测试数据集合；接着，使用构造好的分类预测器对所述测试数据集合进行分类操作，去除安全漏洞触发效果相同的测试用例数据。

[0021] 优选地，所述确定种子用例的生成模式步骤中，确定种子用例的生成模式格式如下：

[0022] <vars,exps,ops,func,judge_num,var_num,func_num,env,conf>

[0023] 其中，vars指针，指向污点变量数组，用来存放污点变量，数组长度根据污点标记时记录的污点变量个数而定；exps指针，指向表达式类型数组，用来存放可疑语句的类型数据，数组长度根据代码检测时记录的语句个数来确定；ops指针，指向运算符类型数组，用来存放可疑语句中存在的运算符类型；func指针，指向不安全库函数数组；judge_num指待测

软件程序中存在的判定语句个数，所述判定语句个数只记录判定语句的个数，并不记录判定分支间的嵌套层次结构；var_num是指待测软件程序中的污点变量数量；func_num是指待测软件程序中已标记出的不安全函数的数量；env指针，指向环境信息字段；conf指针，指向配置信息字段。

[0024] 优选地，使用机器学习中的决策树模型和随机森林模型构造分类预测器时，采用增加分类特征和调整参数的方法处理欠拟合问题。

[0025] 优选地，使用机器学习中的决策树模型和随机森林模型构造分类预测器时，还对分类特征数进行限制，同时尽可能扩展训练数据集合以降低过拟合的概率。

[0026] (三)有益效果

[0027] 本发明对目前主流模糊测试技术存在的测试用例冗余问题开展优化设计，在面向源程序文件的模糊测试用例生成方面，通过在模糊测试用例生成之前，标记识别程序对象中的污点变量和问题函数，结合对已有的种子用例生成、筛选等技术，可提升模糊测试用例的有效性，降低模糊测试用例集合的冗余度。其中，在测试用例生成环节，本发明结合机器学习，分析机器学习用于测试用例精简的可行性，得到机器学习的测试用例生成优化技术思路，采用机器学习的模型和算法，改进模糊测试流程中的测试用例生成环节，提升测试用例的生成效率，实现测试用例结合的去冗余，达到提高模糊测试流程智能化程度的目标。

附图说明

[0028] 图1为本发明基于机器学习的模糊测试用例方法原理框图；

[0029] 图2为本发明的方法中代码检测流程图；

[0030] 图3为本发明的方法中污点标记流程图；

[0031] 图4为本发明的方法中不安全函数标记流程图。

具体实施方式

[0032] 为使本发明的目的、内容、和优点更加清楚，下面结合附图和实施例，对本发明的具体实施方式作进一步详细描述。

[0033] 针对目前主流模糊测试技术存在的三种典型问题，本发明对目前主流模糊测试技术存在的测试用例冗余问题开展优化设计，在面向源程序文件的模糊测试用例生成方面，通过在模糊测试用例生成之前，标记识别程序对象中的污点变量和问题函数，结合对已有的种子用例生成、筛选等技术，可提升模糊测试用例的有效性，降低模糊测试用例集合的冗余度。

[0034] 测试用例生成环节是模糊测试方法的核心环节，测试用例的有效性直接影响模糊测试结果的准确性。由于传统的模糊测试技术在测试用例生成阶段，采用在待测程序的输入空间中随机取值，导致测试用例集合对程序的覆盖率较低，最终的测试结果始终难以令人满意。本发明结合机器学习，分析机器学习用于测试用例精简的可行性，得到机器学习的测试用例生成优化技术思路，采用机器学习的模型和算法，改进模糊测试流程中的测试用例生成环节，提升测试用例的生成效率，实现测试用例结合的去冗余，达到提高模糊测试流程智能化程度的目标。

[0035] 基于机器学习的模糊测试用例生成方法原理框图如图1所示，该方法包括代码预

处理环节、模糊测试用例生成环节。

[0036] 为了提高模糊测试用例的有效性，目前针对模糊测试的优化技术都不同程度地对待测软件程序进行了程序分析。本发明延续智能模糊测试的技术路线，为了生成有效的模糊测试用例，需要对待测的软件程序进行代码预处理。本发明的代码预处理环节采用代码检测技术、污点标记技术和不安全函数标记技术，对待测软件程序进行预处理和分析，标识出可疑语句、污点变量和不安全函数等疑似脆弱性代码，而不进行过于深入的程序理论分析。其中，代码检测技术用于逻辑错误和表达式类型的检测，污点标记技术侧重于程序变量的检查和标识，而不安全函数标记技术用于标识前两种手段遗漏的疑似问题函数。通过这三种技术的结合使用，实现对待测软件程序中的变量、语句和函数三个层级的预处理。具体包括如下三个步骤：

[0037] (1)代码检测

[0038] 本发明采取的代码检测技术，通过静态语句标记技术，对待测软件程序只进行可疑语句标记操作，而不进行诸如路径检查、语义分析等过于深入的程序理论分析。具体流程如图2所示。

[0039] (2)污点标记

[0040] 本发明通过静态污点标记技术，在对代码检测输出的软件程序进行污点标记时，根据待测软件程序的数据输入格式和要检测的安全漏洞特征，对不信任的输入数据和程序中的可疑变量做标记，后续结合不安全函数标记技术，能够检测使用污点数据的不安全方式，而不通过跟踪污点数据的传播过程来揭示代码脆弱性所在的位置，具体流程如图3所示。

[0041] (3)不安全函数标记

[0042] 本发明通过采用不安全函数标记技术，识别并标记出污点标记输出的软件程序中存在的不安全函数，对可能出现不安全调用方式的函数进行标识。具体流程如图4所示。

[0043] 以上代码检测、污点标记和不安全函数标记三个步骤也可以并行处理，但处理后的三个结果还需要进行去重处理。

[0044] 模糊测试用例生成环节：为了在保证覆盖率的前提下完成测试用例数量的精简，本发明利用机器学习中的决策树模型和随机森林模型寻找出良好的种子用例集合，使用变异的方式产生大批量的测试用例，并使用训练后的分类器去除同类安全漏洞下多余的测试用例。具体包括如下两个步骤：

[0045] (1)确定种子用例的生成模式

[0046] 本发明根据代码预处理标记出的问题语句、污点变量、不安全函数等可疑代码构造测试用例的生成模式。针对待测的软件程序，经过代码预处理阶段的标记和信息记录，可疑语句、污点变量、不安全函数的位置和相互之间的关系均已被提取出来。这些数据用于生成种子用例的生成模式，提供诸如污点变量字段、不安全函数字段、可疑语句字段、运算符类型字段、表达式类型字段、判定分支数量字段、步骤生成模式的参考依据。故而，确定种子用例的生成模式格式如下：

[0047] <vars,exps,ops,func,judge_num,var_num,func_num,env,conf>

[0048] 其中，vars指针，指向污点变量数组，用来存放污点变量，数组长度根据污点标记时记录的污点变量个数而定；exps指针，指向表达式类型数组，用来存放可疑语句的类型数

据,数组长度根据代码检测时记录的语句个数来确定;ops指针,指向运算符类型数组,用来存放可疑语句中存在的运算符类型;func指针,指向不安全库函数数组;judge_num指待测软件程序中存在的判定语句个数(只记录判定语句的个数,并不记录判定分支间的嵌套层次结构);var_num是指待测软件程序中的污点变量数量;func_num是指待测软件程序中已标记出的不安全函数的数量;env指针,指向环境信息字段;conf指针,指向配置信息字段。

[0049] (2)模糊测试用例生成

[0050] 根据种子用例的生成模式,对其中字段的取值种类进行组合,生成一批用于模糊测试的初始测试用例,将所述初始测试用例记为初始集合;然后,将所述初始集合中的初始测试用例输入到待测软件程序中,动态观测执行效果,记录下触发安全漏洞的情况,并根据触发效果判定种子用例的生成模式是否合理准确,在需要时进行生成模式的调整;使用机器学习中的决策树算法和随机森林算法构造分类预测器,并将调整后的用例生成模式作为分类依据,再对所述初始集合中的测试用例进行分类学习,以此训练出满足准确率要求的分类预测器;然后,根据修改后的用例生成模式,生成多批量的测试用例作为分类预测器的测试数据集合;接着,使用构造好的分类预测器对所述测试数据集合进行分类操作,去除安全漏洞触发效果相同的测试用例数据,以此实现精简测试用例集合的目标。

[0051] 同时,为了解决本步骤中决策树学习模型和随机森林学习模型的泛化局限问题,本发明采用了一些方法对欠拟合与过拟合情况进行了限制和避免。一般来说,机器学习中的欠拟合问题是指学习模型在测试数据集合中的训练数据集的训练过程中无法取得令人满意的准确率,在测试数据集合上的表现也不好;而过拟合问题是指学习模型在训练数据集上可以取得较好的结果,甚至是100%的准确率,但是到了模型验证环节,却在测试数据集合上无法取得符合要求的结果。为了在一定程度上解决随机森林模型和决策树模型的欠拟合与过拟合问题,采用增加分类特征和调整参数的方法处理欠拟合问题;通过一定程度上降低模型的复杂度,并对分类特征数进行限制,同时尽可能扩展训练数据集合,降低过拟合的概率。

[0052] 以上所述仅是本发明的优选实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明技术原理的前提下,还可以做出若干改进和变形,这些改进和变形也应视为本发明的保护范围。

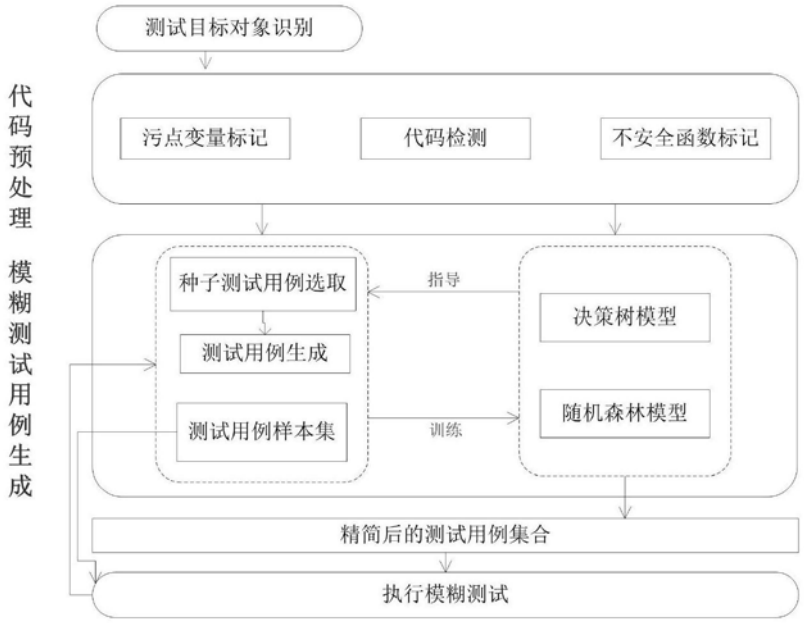

测试目标对象识别
代码预处理
污点变量标记
代码检测
不安全函数标记
模糊测试用例生成
种子测试用例选取
指导
决策树模型
测试用例生成
测试用例样本集
训练
随机森林模型
精简后的测试用例集合
执行模糊测试

图1

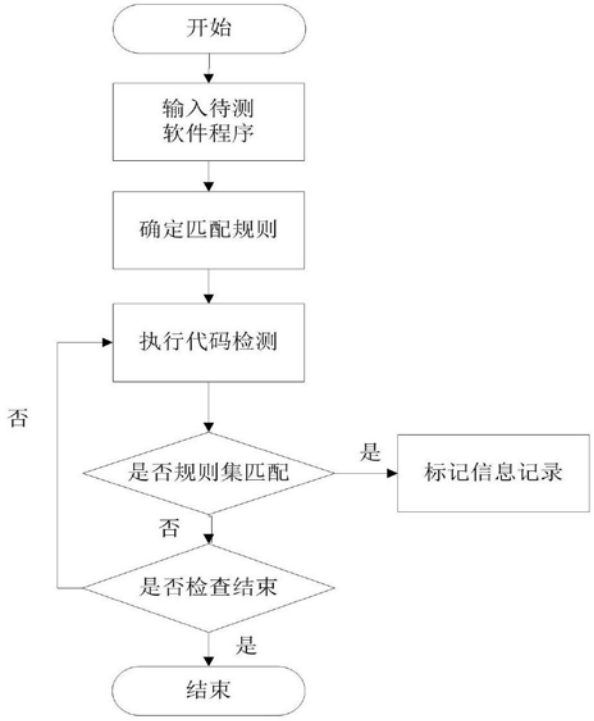

开始
输入待测
软件程序
确定匹配规则
执行代码检测
否
是否规则集匹配
是
标记信息记录
否
是否检查结束
是
结束

图2

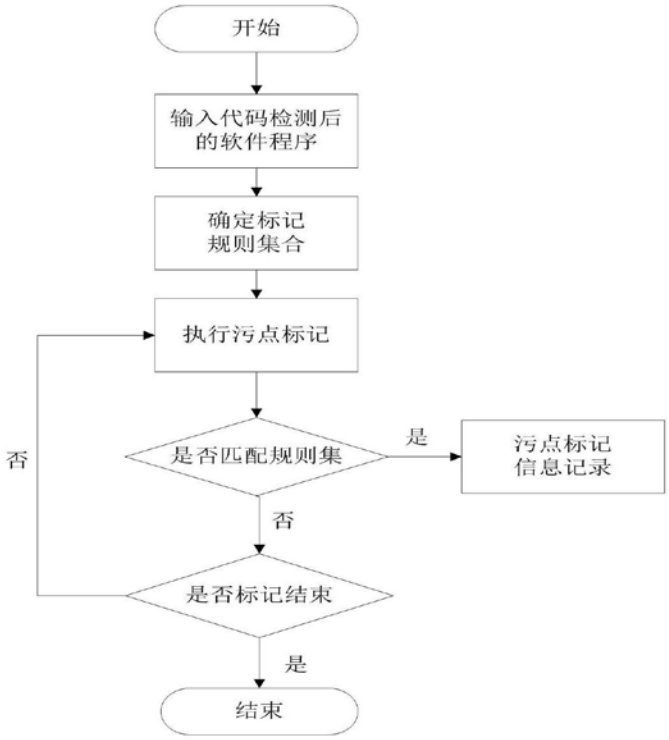

开始
输入代码检测后的软件程序
确定标记规则集合
执行污点标记
是否匹配规则集
是
污点标记信息记录
否
否
是否标记结束
是
结束

图3

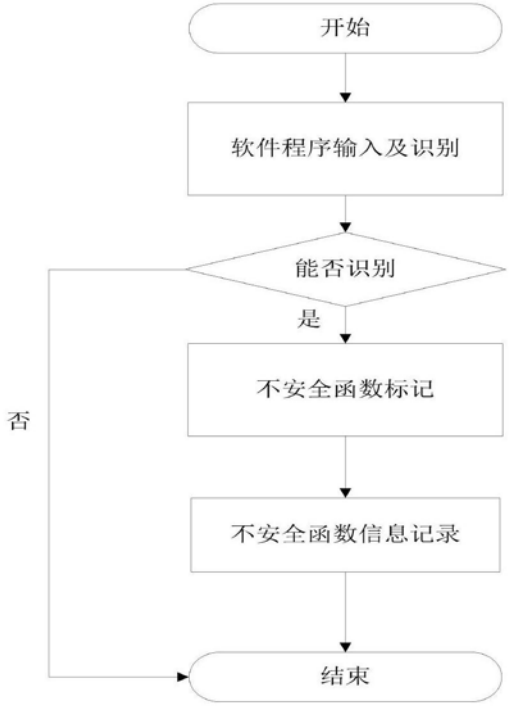

开始
软件程序输入及识别
能否识别
是
否
不安全函数标记
不安全函数信息记录
结束

图4