



(12) 发明专利

(10) 授权公告号 CN 117056209 B

(45) 授权公告日 2024. 07. 26

(21) 申请号 202310933290.4

(22) 申请日 2023.07.27

(65) 同一申请的已公布的文献号

申请公布号 CN 117056209 A

(43) 申请公布日 2023.11.14

(73) 专利权人 人工智能与数字经济广东省实验室(广州)

地址 510330 广东省广州市海珠区新港东路2429号首层自编051房

专利权人 华南理工大学

(72) 发明人 钟昊阳 邹全义 陆璐 冼允廷

(74) 专利代理机构 广州科粤专利商标代理有限公司 44001

专利代理师 杨可维 劳剑东

(51) Int. Cl.

G06F 11/36 (2006.01)

G06N 3/04 (2023.01)

G06N 3/08 (2023.01)

(56) 对比文件

CN 112035345 A, 2020.12.04

CN 115146282 A, 2022.10.04

审查员 马桂新

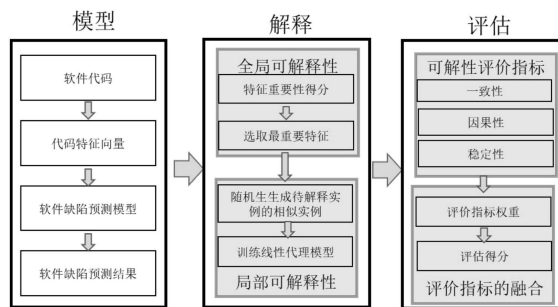
权利要求书3页 说明书9页 附图3页

(54) 发明名称

软件缺陷预测模型、解释方法以及量化评估方法

(57) 摘要

本发明公开一种软件缺陷预测模型、解释方法以及量化评估方法,首先,基于多维度代码特征构建软件缺陷预测模型,旨在捕捉软件缺陷的关键因素,从而提高模型的预测性能,其次,采用代理模型来拟合复杂的缺陷预测模型,通过使用代理模型,可以将缺陷预测模型的复杂性转化为更容易理解和解释的形式,能够更好地理解模型的决策过程和预测结果,最后,设计了多个可解释性评价指标,并构建了相应的评估方法,用于对可解释性效果进行评估,以确保评估结果的可靠性和有效性。本发明的有益效果:能够在软件缺陷预测模型中增加可解释性,并利用定量评估方法来衡量可解释性的效果,从而提供更可靠和可解释的软件缺陷预测模型解释机制。



1. 一种软件缺陷预测模型可解释性的量化评估方法,用于以下软件缺陷预测模型,以及解释方法,所述软件缺陷预测模型包括:将代码片段进行解析,生成代码中间表示;将所述代码中间表示输入向量训练模型进行学习,获得特征向量;检测所述代码片段的警告信息;提取所述警告信息的类型属性信息,并对每个所述警告信息进行切片,获取与所述警告信息相关的代码碎片,每个所述代码碎片表示为一组所述类型属性信息;所述特征向量和所述类型属性信息进行融合,构建综合特征向量;将所述综合特征向量和已标记的软件缺陷样本数据共同作为输入,使用深度神经网络构建并训练软件缺陷预测模型;所述解释方法包括以下步骤:根据所述软件缺陷预测模型获取基准参考预测值;为所述软件缺陷预测模型的特征创建变量分位数矩阵;根据所述变量分位数矩阵获取对应的预测矩阵;根据所述预测矩阵计算每个分位数的重要性得分值;将全部的所述重要性得分值合并为全局特征重要性得分值;从所述全局特征重要性得分值中将贡献最大的重要性得分值,并将所述重要性得分值对应的所述特征组成新的特征集合,将所述特征集合和所述软件缺陷预测模型组成待解释实例;以所述待解释实例为中心对所述特征集合的训练数据进行随机采样,得到所述待解释实例的局部解释目标函数;根据样本之间的距离来确定所述局部解释目标函数的权重系数;通过线性回归法对所述局部解释目标函数进行优化,得到代理模型;

其特征在于,包括以下步骤:

设置一致性评价指标,用于度量所述代理模型的解释结果和所述软件缺陷预测模型的解释结果之间的一致性;

设置因果性评价指标,用于判断各个特征的删除或扰动对所述因果性评价指标的输出值的影响是否超过预设值,若是,则认定所述特征与解释结果之间存在强因果效应;

设置稳定性评价指标,用于将所述代理模型和所述软件缺陷预测模型中拥有相同解释结果的实例定义为覆盖实例;

将所述一致性评价指标、所述因果性评价指标和所述稳定性评价指标进行归一化处理,得到归一化处理矩阵;

计算所述归一化处理矩阵的信息熵;

根据所述信息熵计算所述一致性评价指标、所述因果性评价指标和所述稳定性评价指标的相对熵;

根据所述相对熵的值,计算所述一致性评价指标、所述因果性评价指标和所述稳定性评价指标的权重;

各个所述权重分别和对应的所述归一化处理矩阵加权后得到综合评估分数。

2. 如权利要求1所述的软件缺陷预测模型可解释性的量化评估方法,其特征在于,所述变量分位数矩阵包括:

$$\mathbf{Z}_i = \begin{bmatrix} \mathbf{z}_{i,0} \\ \mathbf{z}_{i,\frac{1}{Q-1}} \\ \vdots \\ \mathbf{z}_{i,1} \end{bmatrix} = \begin{bmatrix} \bar{x}_1 & \bar{x}_2 & \dots & x_{i,0} & \dots & \bar{x}_p \\ \bar{x}_1 & \bar{x}_2 & \dots & x_{i,\frac{1}{Q-1}} & \dots & \bar{x}_p \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \bar{x}_1 & \bar{x}_2 & \dots & x_{i,1} & \dots & \bar{x}_p \end{bmatrix} \quad (1);$$

其中, $x_{i,q}$ 表示第 i 个特征对应的 q 个分位数, Q 表示分位数的总数, p 表示特征维度的总数, $q \in \{0, \frac{1}{Q-1}, \frac{2}{Q-1}, \dots, 1\}$;

所述预测矩阵包括:

$$\hat{\mathbf{y}}_i = \begin{bmatrix} \hat{y}_{i,0} \\ \hat{y}_{i,\frac{1}{Q-1}} \\ \vdots \\ \hat{y}_{i,1} \end{bmatrix} = \begin{bmatrix} f(\mathbf{z}_{i,0}) \\ f(\mathbf{z}_{i,\frac{1}{Q-1}}) \\ \vdots \\ f(\mathbf{z}_{i,1}) \end{bmatrix} \quad (2);$$

其中, $f(\cdot)$ 为软件缺陷预测模型。

3. 如权利要求2所述的软件缺陷预测模型可解释性的量化评估方法,其特征在于,所述重要性得分值包括:

$$\Delta_{i,q} = \frac{\hat{y}_{i,q} - f(\mathbf{x}_b)}{\sqrt{\text{var}(\hat{\mathbf{y}}_i)}} (\max(\hat{\mathbf{y}}_i) - \min(\hat{\mathbf{y}}_i)) \quad (3)。$$

4. 如权利要求3所述的软件缺陷预测模型可解释性的量化评估方法,其特征在于,所述局部解释目标函数包括:

$$\min_{\phi} L(\phi) + \sum_{k=1}^K p(\mathbf{t}_{j,k}) (f(\mathbf{x}_j) - g(\mathbf{t}_{j,k}))^2 \quad (4);$$

其中, $\mathbf{t}_{j,k} = \mathbf{t}_j + (1 - \lambda) \mathbf{t}'_{j,k}$, $\mathbf{t}'_{j,k}$ 为 $\mathbf{t}_j$ 的第 k 个最近邻的实例, $\mathbf{x}_j$ 为第 j 个实例, $p(\cdot)$ 为权重系数, $g(\cdot)$ 为所述代理模型, ϕ 为预测值;

所述权重系数包括:

$$p(\mathbf{t}_{i,k}) = \exp\left(\frac{-\|\mathbf{t}_{j,k} - \mathbf{t}_j\|^2}{\sigma^2}\right) \quad (5);$$

其中, $\mathbf{t}_j$ 为第 j 个代理实例, σ 为高斯函数的方差。

5. 如权利要求4所述的软件缺陷预测模型可解释性的量化评估方法,其特征在于,所述一致性评价指标包括:

$$PCM = \frac{1}{|C|} \sum_{c \in C} \frac{r_c}{|c|} - \frac{w_c}{|c|} \quad (6);$$

其中, c 表示标签, C 为表示标签, r_c 表示当标签为 c 时代理模型和所述软件缺陷预测模型的预测结果具有相同的数量, w_c 表示当标签为 c 时代理模型和所述软件缺陷预测模型的预测结果具有不同的数量;

所述因果性评价指标包括:

$$S = \frac{1}{M} \left| \sum_{i=1}^M g(t) - [g(T_i) + g(T'_i) + g(T_i^{\otimes})] \right| \quad (7);$$

其中, M 表示最重要的特征的个数, $T_i = [\bar{t}_1, \bar{t}_2, \dots, \bar{t}_{i-1}, t_i, \bar{t}_{i+1}, \dots, \bar{t}_M]$, $T'_i = [\bar{t}_1, \bar{t}_2, \dots, \bar{t}_{i-1}, 0, \bar{t}_{i+1}, \dots, \bar{t}_M]$ 和 $T_i^{\otimes} = [\bar{t}_1, \bar{t}_2, \dots, \bar{t}_{i-1}, t_i^{\times}, \bar{t}_{i+1}, \dots, \bar{t}_M]$, 其中 $t_i^{\times} = t_i + \text{rand}()$;

所述稳定性评价指标包括:

$$R = \frac{1}{r} \sum_{i=1}^r |g(t'_i) - g(t'^*_i)| \quad (8)。$$

6. 如权利要求5所述的软件缺陷预测模型可解释性的量化评估方法,其特征在於,所述权重的计算方法为:

$$w_i = \frac{\text{Re}(\mathbf{x}_i^*)}{\sum_{j=1}^3 \text{Re}(\mathbf{x}_j^*)} \quad (9)；$$

所述综合评估分数包括:

$$\text{score} = \sum_{i=1}^3 w_i \mathbf{x}_i^* \quad (10)。$$

软件缺陷预测模型、解释方法以及量化评估方法

技术领域

[0001] 本发明涉及软件可解释性评估技术领域,尤其涉及一种软件缺陷预测模型、解释方法以及量化评估方法。

背景技术

[0002] 软件缺陷预测是软件工程领域中一项重要的研究方向。它利用历史数据对软件开发过程进行挖掘和分析,通过设计与软件缺陷相关的度量指标,并对程序模块进行度量,再运用统计或机器学习技术构建缺陷预测模型,以实现目标软件模块的缺陷倾向、缺陷严重程度或缺陷数量分布等方面的预测。然而,尽管机器学习有着端到端等优点,但是由于多数机器学习的黑箱特点、以及当前对缺陷预测结果现象的理解不全面等事实,导致基于机器学习的软件缺陷预测模型无法提供清晰的解释和证据来支持其预测结果,从而限制了这些模型在实际应用中的有效性。

[0003] 中国发明专利申请CN114528221A公开了“一种基于异质图神经网络的软件缺陷预测方法”,该方法利用了软件仓库和缺陷日志数据中的信息,通过构建代码图和缺陷链,使用前缀树和带状态机进行节点跨域关联,将代码图中的节点与缺陷图中的节点关联起来。然后,对代码节点和缺陷节点生成表示向量,并通过异质图神经网络进行多层次的注意力聚合,从而实现内容和语义路径的信息传递。通过使用多层次的注意力聚合和知识图谱表示学习方法,可以更好地捕捉代码和缺陷之间的语义信息,提高检测的准确性。该方法存在的不足之处是:虽然从代码仓库中获取了代码的语义信息,但却舍弃了开源社区中的其他代码知识,如缺陷报告、修复记录等,导致所建立的缺陷检测模型学习到的知识不够充分,从而影响检测结果的精准性。其次,预测结果缺乏可解释性,难以解释模型对缺陷预测的依据,限制了方法的解释性。

[0004] 中国发明专利申请CN115629980A公开了一种名为“一种软件缺陷预测模型解释方法”,该方法旨在通过采用与模型无关的解释方法,提供对软件缺陷预测模型结果的解释,并结合改进的筛选策略,以满足软件缺陷预测的特定需求。通过该方法,用户能够更加清晰地理解模型的预测结果,并获得与软件缺陷相关的更多洞察。然而,需要指出的是,该方案未对提供的解释效果进行有效评估。因此,在实际应用该方法时,仍然需要进一步评估和验证,以确保解释结果的准确性和可信度。

[0005] 综上,现有的软件缺陷预测模型、预测模型的解释方法,以及解释方法的评估方法均存在各自的不足,再有,由于上述三者并非针对彼此进行统一设计,各种特征和指标之间缺乏可靠的可解释逻辑,导致解释方法的可解释性的可靠性不高。

发明内容

[0006] 针对上述问题,本发明提出一种软件缺陷预测模型、解释方法以及量化评估方法,解决背景技术中指出的至少一个问题。

[0007] 为解决上述技术问题,本发明第一方面提出一种软件缺陷预测模型,包括以下步

骤:

- [0008] 将代码片段进行解析,生成代码中间表示;
- [0009] 将所述代码中间表示输入向量训练模型进行学习,获得特征向量;
- [0010] 检测所述代码片段的警告信息;
- [0011] 提取所述警告信息的类型属性信息,并对每个所述警告信息进行切片,获取与所述警告信息相关的代码碎片,每个所述代码碎片表示为一组所述类型属性信息;
- [0012] 所述特征向量和所述类型属性信息进行融合,构建综合特征向量;
- [0013] 将所述综合特征向量和已标记的软件缺陷样本数据共同作为输入,使用深度神经网络构建并训练软件缺陷预测模型。
- [0014] 本发明第二方面提出一种基于代理模型软件缺陷预测模型解释方法,用于上述的软件缺陷预测模型,包括以下步骤:
- [0015] 根据所述软件缺陷预测模型获取基准参考预测值;
- [0016] 为所述软件缺陷预测模型的特征创建变量分位数矩阵;
- [0017] 根据所述变量分位数矩阵获取对应的预测矩阵;
- [0018] 根据所述预测矩阵计算每个的分位数的重要性得分值;
- [0019] 将全部的所述重要性得分值合并为全局特征重要性得分值;
- [0020] 从所述全局特征重要性得分值中将贡献最大的重要性得分值,并将所述重要性得分值对应的所述特征组成新的特征集合,将所述特征集合和所述软件缺陷预测模型组成待解释实例;
- [0021] 以所述待解释实例为中心对所述特征集合的训练数据进行随机采样,得到所述待解释实例的局部解释目标函数;
- [0022] 根据样本之间的距离来确定所述局部解释目标函数的权重系数;
- [0023] 通过线性回归法对所述局部解释目标函数进行优化,得到代理模型。
- [0024] 在一些实施方式中,所述变量分位数矩阵包括:

$$[0025] \quad \mathbf{Z}_i = \begin{bmatrix} \mathbf{z}_{i,0} \\ \mathbf{z}_{i,\frac{1}{Q-1}} \\ \vdots \\ \mathbf{z}_{i,1} \end{bmatrix} = \begin{bmatrix} \bar{x}_1 & \bar{x}_2 & \dots & x_{i,0} & \dots & \bar{x}_p \\ \bar{x}_1 & \bar{x}_2 & \dots & x_{i,\frac{1}{Q-1}} & \dots & \bar{x}_p \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \bar{x}_1 & \bar{x}_2 & \dots & x_{i,1} & \dots & \bar{x}_p \end{bmatrix} \quad (1);$$

- [0026] 其中, $x_{i,q}$ 表示第 i 个特征对应的 q 个分位数, Q 表示分位数的总数, p 表示特征维度的总数, $q \in \{0, \frac{1}{Q-1}, \frac{2}{Q-1}, \dots, 1\}$;

- [0027] 所述预测矩阵包括:

$$[0028] \quad \hat{\mathbf{y}}_i = \begin{bmatrix} \hat{y}_{i,0} \\ \hat{y}_{i,\frac{1}{Q-1}} \\ \vdots \\ \hat{y}_{i,1} \end{bmatrix} = \begin{bmatrix} f(\mathbf{z}_{i,0}) \\ f(\mathbf{z}_{i,\frac{1}{Q-1}}) \\ \vdots \\ f(\mathbf{z}_{i,1}) \end{bmatrix} \quad (2);$$

- [0029] 其中, $f(\cdot)$ 为软件缺陷预测模型。
- [0030] 在一些实施方式中,所述重要性得分值包括:

[0031] $\Delta_{i,q} = \frac{\hat{y}_{j,q} - f(\mathbf{x}_b)}{\sqrt{\text{var}(\hat{\mathbf{y}}_i)}} (\max(\hat{\mathbf{y}}_i) - \min(\hat{\mathbf{y}}_i)) \quad (3)。$

[0032] 所述局部解释目标函数包括：

[0033] $\min_{\phi} L(\phi) + \sum_{k=1}^K p(\mathbf{t}_{j,k}) (f(\mathbf{x}_j) - g(\mathbf{t}_{j,k}))^2 \quad (4)；$

[0034] 其中, $\mathbf{t}_{j,k} = \mathbf{t}_j + (1-\lambda) \mathbf{t}'_{j,k}$, $\mathbf{t}'_{j,k}$ 为 $\mathbf{t}_j$ 的第 k 个最近邻的实例, $\mathbf{x}_j$ 为第 j 个实例, $p(\cdot)$ 为权重系数, $g(\cdot)$ 为所述代理模型, ϕ 为预测值。

[0035] 在一些实施方式中,所述权重系数包括：

[0036] $p(\mathbf{t}_{i,k}) = \exp\left(\frac{-\|\mathbf{t}_{j,k} - \mathbf{t}_j\|^2}{\sigma^2}\right) \quad (5)；$

[0037] 其中, $\mathbf{t}_j$ 为, σ 为。

[0038] 本发明第三方面提出一种软件缺陷预测模型可解释性的量化评估方法,用于上述的基于代理模型软件缺陷预测模型解释方法,包括以下步骤：

[0039] 设置一致性评价指标,用于度量所述代理模型的解释结果和所述软件缺陷预测模型的解释结果之间的一致性；

[0040] 设置因果性评价指标,用于判断各个特征的删除或扰动对所述因果性评价指标的输出值的影响是否超过预设值,若是,则认定所述特征与解释结果之间存在强因果效应；

[0041] 设置稳定性评价指标,用于将所述代理模型和所述软件缺陷预测模型中拥有相同解释结果的实例定义为覆盖实例；

[0042] 将所述一致性评价指标、所述因果性评价指标和所述稳定性评价指标进行归一化处理,得到归一化处理矩阵；

[0043] 计算所述归一化处理矩阵的信息熵；

[0044] 根据所述信息熵计算所述一致性评价指标、所述因果性评价指标和所述稳定性评价指标的相对熵；

[0045] 根据所述相对熵的值,计算所述一致性评价指标、所述因果性评价指标和所述稳定性评价指标的权重；

[0046] 各个所述权重分别和对应的所述归一化处理矩阵加权后得到综合评估分数。

[0047] 在一些实施方式中,所述一致性评价指标包括：

[0048] $PCM = \frac{1}{|C|} \sum_{c \in C} \frac{r_c}{|c|} - \frac{w_c}{|c|} \quad (6)；$

[0049] 其中, c 表示标签, C 为表示标签, r_c 表示当标签为 c 时代理模型和所述软件缺陷预测模型的预测结果具有相同的数量, w_c 表示当标签为 c 时代理模型和所述软件缺陷预测模型的预测结果具有不同的数量。

[0050] 所述因果性评价指标包括：

[0051] $S = \frac{1}{M} \left| \sum_{i=1}^M g(t) - [g(T_i) + g(T_i) + g(T_i^{\otimes})] \right| \quad (7)；$

[0052] 其中, M 为表示最重要的特征的个数, $T_i = [\bar{t}_1, \bar{t}_2, \dots, \bar{t}_{i-1}, t_i, \bar{t}_{i+1}, \dots, \bar{t}_M]$, $T_i' = [\bar{t}_1, \bar{t}_2, \dots, \bar{t}_{i-1}, 0, \bar{t}_{i+1}, \dots, \bar{t}_M]$ 和 $T_i^{\otimes} = [\bar{t}_1, \bar{t}_2, \dots, \bar{t}_{i-1}, t_i^{\otimes}, \bar{t}_{i+1}, \dots, \bar{t}_M]$, 其中 $t_i^{\otimes} = t_i + \text{rand}()$ ；

[0053] 在一些实施方式中,所述稳定性评价指标包括:

$$[0054] \quad R = \frac{1}{r} \sum_{i=1}^r |g(t_i') - g(t_i^*)| \quad (8)。$$

[0055] 在一些实施方式中,所述权重的计算方法为:

$$[0056] \quad w_i = \frac{\text{Re}(\mathbf{x}_i^*)}{\sum_{j=1}^3 \text{Re}(\mathbf{x}_j^*)} \quad (9)；$$

[0057] 所述综合评估分数包括:

$$[0058] \quad \text{score} = \sum_{i=1}^3 w_i \mathbf{x}_i^* \quad (10)；$$

[0059] 本发明第四方面提出一种基于代理模型的软件缺陷预测模型解释装置,包括存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机程序,所述处理器执行所述计算机程序时实现上述方法的步骤。

[0060] 本发明第五方面提出一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,所述计算机程序被处理器执行时实现上述方法的步骤。

[0061] 本发明的有益效果为:能够在软件缺陷预测模型中增加可解释性,并利用定量评估方法来衡量可解释性的效果,从而提供更可靠和可解释的软件缺陷预测模型解释机制,有助于提高人们对于模型预测的理解和信任度,并为软件开发和维护过程中的决策提供有益的指导,具体请参阅实施例一至三。

附图说明

[0062] 图1为本发明的总体框架图;

[0063] 图2为本发明实施例一公开的软件缺陷预测模型的构建流程示意图;

[0064] 图3为本发明实施例二公开的基于代理模型的软件缺陷预测模型解释方法的流程示意图;

[0065] 图4为本发明实施例三公开的软件缺陷预测模型可解释性的量化评估方法的流程示意图;

[0066] 图5为本发明实施例四公开的基于代理模型的软件缺陷预测模型解释方法的结构示意图。

具体实施方式

[0067] 为使本发明的目的、技术方案及优点更加清楚、明确,下面结合附图和具体实施方式对本发明的内容做进一步详细说明。可以理解的是,此处所描述的具体实施例仅仅用于解释本发明,而非对本发明的限定。另外还需要说明的是,为了便于描述,附图中仅示出了与本发明相关的部分而非全部内容。

[0068] 以下实施例一至三分别提出一种软件缺陷预测模型、一种基于代理模型的软件缺陷预测模型解释方法,以及一种软件缺陷预测模型可解释性的量化评估方法,用于预测目标软件是否存在缺陷和对预测结果提供解释,并对解释进行量化评估,如图1所示,本发明以“模型—解释—评估”为框架开展研究,以下是对该实施例的详细描述。

[0069] 实施例一

[0070] 本实施例提出了一种软件缺陷预测模型,图2所示,包括以下步骤:

[0071] S11,将代码片段进行解析,生成代码中间表示。

[0072] 软件工程领域,中间表示(Intermediate Representation)如抽象语法树(Abstract Syntax Tree,AST)、控制流图(Control Flow Graph,CFG)和程序依赖图(Program Dependency Graph,PDG)等起着重要作用。利用Joern工具解析代码片段进行解析生成代码中间表示。

[0073] S12,将代码中间表示输入向量训练模型进行学习,获得特征向量。

[0074] 具体的,在S12中,对于AST,将AST进一步划分为语句级树的方法,将语句级树转化为词向量并利用双向循环神经网络进行学习,从而获得源代码的语义特征向量 v_1 。对于CFG和PDG,采用了Node2vec方法对控制流图(CFG)和程序依赖图(PDG)中的节点进行随机游走,得到图的节点表示。这些节点表示被输入到Sent2vec模型中进行训练,以获得控制流图的控制流特征向量 v_2 和程序依赖图的程序依赖特征向量 v_3 。

[0075] S13,检测代码片段的警告信息。

[0076] 在S13中,可使用静态扫描工具进行检测的方法,从而获得代码中存在的警告信息,通过整理每个软件代码片段的警告,形成警告集 $W = \{w_1, w_2, \dots, w_r\}$ 。这些警告信息能够揭示潜在的缺陷和代码质量问题。

[0077] S14,提取警告信息的类型属性信息,并对每个警告信息进行切片,获取与警告信息相关的代码碎片,每个代码碎片表示为一组类型属性信息。

[0078] 在一示例中,针对S13中的每个警告信息,提取了类型属性信息,并利用Joana对每个警告进行切片,以获取与警告相关的代码片段。最终,每个警告被表示为一组属性,表示为 $w = [attr_1, attr_2, \dots, attr_n]$ 。

[0079] S15,特征向量和类型属性信息进行融合,构建综合特征向量。

[0080] 在一示例中,对于步骤S12中获得的语义特征向量 v_1 、控制流特征向量 v_2 和程序依赖特征向量 v_3 ,以及步骤S14中提取的类型属性信息 w 进行融合的方法,构建了一个综合特征向量 $x = [x_1, x_2, \dots, x_p]$ 。

[0081] S16,将综合特征向量和已标记的软件缺陷样本数据共同作为输入,使用深度神经网络构建并训练软件缺陷预测模型。

[0082] 在S16中,将融合后的特征向量作为输入,结合已标记的软件缺陷样本数据,使用深度神经网络方法(如随机森林)训练了一个软件缺陷预测模型 $f(\cdot)$,软件缺陷预测模型 $f(\cdot)$ 的评估和参数调优可以借助交叉验证等技术来进行。

[0083] S18,通过模型预测得到的软件缺陷概率或分类结果 $f(x)$ 。

[0084] 在本实施例中,基于多维度代码特征构建软件缺陷预测模型,通过收集目标软件与缺陷相关的警告信息(包括软件模块的特征、历史缺陷记录)等,用于训练软件缺陷预测模型,能够预测软件是否存在缺陷,上述代码特征的选择旨在捕捉软件缺陷的关键因素,从而提高模型的预测性能。

[0085] 实施例二

[0086] 本实施例提出了一种基于代理模型的软件缺陷预测模型解释方法,图3所示,用于实施例一所述的软件缺陷预测模型,包括以下步骤:

[0087] S21,根据软件缺陷预测模型获取基准参考预测值。

[0088] 具体的,在S21中,利用软件缺陷预测模型 $f(\cdot)$ 获取基准参考预测值 $\phi_0 = f(\bar{x})$ 。

[0089] S22,为软件缺陷预测模型的特征创建变量分位数矩阵。

[0090] 变量分位数矩阵 Z_i 包括:

$$[0091] \quad \mathbf{Z}_i = \begin{bmatrix} \mathbf{z}_{i,0} \\ \mathbf{z}_{i,\frac{1}{Q-1}} \\ \vdots \\ \mathbf{z}_{i,1} \end{bmatrix} = \begin{bmatrix} \bar{x}_1 & \bar{x}_2 & \dots & x_{i,0} & \dots & \bar{x}_p \\ \bar{x}_1 & \bar{x}_2 & \dots & x_{i,\frac{1}{Q-1}} & \dots & \bar{x}_p \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \bar{x}_1 & \bar{x}_2 & \dots & x_{i,1} & \dots & \bar{x}_p \end{bmatrix} \quad (1);$$

[0092] 其中, $Z_i \in \mathbb{R}^{Q \times p}$,该矩阵的一行可以表示为: $\mathbf{z}_{i,q} = [\bar{x}_1, \bar{x}_2, \dots, \bar{x}_{i-1}, x_{i,q}, \bar{x}_{i+1}, \dots, \bar{x}_p] \circ x_{i,q}$ 表示第*i*个特征对应的*q*个分位数,*Q*表示分位数的总数,*p*表示特征维度的总数, $q \in \{0, \frac{1}{Q-1}, \frac{2}{Q-1}, \dots, 1\}$ 。

[0093] S23,根据变量分位数矩阵获取对应的预测矩阵。

[0094] 预测矩阵 $\hat{\mathbf{y}}_i$ 包括:

$$[0095] \quad \hat{\mathbf{y}}_i = \begin{bmatrix} \hat{y}_{i,0} \\ \hat{y}_{i,\frac{1}{Q-1}} \\ \vdots \\ \hat{y}_{i,1} \end{bmatrix} = \begin{bmatrix} f(\mathbf{z}_{i,0}) \\ f(\mathbf{z}_{i,\frac{1}{Q-1}}) \\ \vdots \\ f(\mathbf{z}_{i,1}) \end{bmatrix} \quad (2);$$

[0096] 其中, $f(\cdot)$ 为软件缺陷预测模型。

[0097] S24,根据预测矩阵计算每个的分位数的重要性得分值。

[0098] 具体的,在S23的预测矩阵 $\hat{\mathbf{y}}_i$ 的基础上计算对应的分位数的要性得分值,重要性得分值包括:

$$[0099] \quad \Delta_{i,q} = \frac{\hat{y}_{i,q} - f(\mathbf{x}_b)}{\sqrt{\text{var}(\hat{\mathbf{y}}_i)}} (\max(\hat{\mathbf{y}}_i) - \min(\hat{\mathbf{y}}_i)) \quad (3)。$$

[0100] S25,将全部的重要性得分值合并为全局特征重要性得分值。

[0101] 根据前述的S21-S24的计算,特征 x_i 全局特征重要性得分值应表示为:

$$I_i = \frac{1}{Q} \sum |\Delta_{i,q}|。$$

[0102] S26,从全局特征重要性得分值中将贡献最大的重要性得分值,并将重要性得分值对应的特征组成新的特征集合,将特征集合和软件缺陷预测模型组成待解释实例。

[0103] 在步骤S26中,根据步骤S25获取的全局特征重要性得分值可解性方法目的是从为*p*个特征中贡献最大的*M*个重要特征组成新的特征集合 $t = [t_1, t_2, \dots, t_m]$,可以用 $(t_j, f(x_j))$ 代替待解释的实例 $(x_j, f(x_j))$ 。

[0104] S27,以待解释实例为中心对特征集合的训练数据进行随机采样,得到待解释实例的局部解释目标函数。

[0105] 局部解释目标函数包括:

$$[0106] \quad \min_{\phi} L(\phi) + \sum_{k=1}^K p(\mathbf{t}_{j,k}) (f(\mathbf{x}_j) - g(\mathbf{t}_{j,k}))^2 \quad (4);$$

[0107] 其中, $t_{j,k} = t_j + (1-\lambda) t'_{j,k}$, $t'_{j,k}$ 为 t_j 的第*k*个最近邻的实例, x_j 为第*j*个实例, $p(\cdot)$ 为权重系数, $g(\cdot)$ 为代理模型, ϕ 为预测值。

[0108] S28,根据样本之间的距离来确定局部解释目标函数的权重系数。

[0109] 在本实施例中,权重系数可以根据样本之间的距离远近来确定。通常情况下,距离越近的样本具有更高的权重,因此权重系数可表示为:

$$[0110] \quad p(\mathbf{t}_{i,k}) = \exp\left(\frac{-\|\mathbf{t}_{i,k} - \mathbf{t}_j\|^2}{\sigma^2}\right) \quad (5);$$

[0111] 其中, $\mathbf{t}_j$ 为, σ 为。

[0112] S29,通过线性回归法对局部解释目标函数进行优化,得到代理模型。

[0113] 在本实施例中,S21-S25通过获取软件缺陷预测模型的重要性得分值,得到每个特征对于预测结果的贡献度,从而获取全局可解释性,然后S26-S29对待解释实例进行随机扰动生成一组近似实例。使用生成的数据集和相应的预测结果,训练一个解释模型。因此,通过分析代理模型的参数、特征重要性或决策路径,可以更好地理解原始模型在给定实例上的预测结果。

[0114] 其中,全局可解释性是通过特征重要性分析揭示预测模型在做出预测时所依据的关键特征,帮助理解模型的决策过程。局部可解释性则是通过随机生成与待解释的实例相似的实例集,训练代理模型。代理模型可以是一个简单的线性模型。

[0115] 实施例三

[0116] 本实施例提出了一种软件缺陷预测模型可解释性的量化评估方法,用于实施例二所述的基于代理模型软件缺陷预测模型解释方法,包括以下步骤:

[0117] S31,设置一致性评价指标,用于度量代理模型的解释结果和软件缺陷预测模型的解释结果之间的一致性。

[0118] 一致性评价指标包括:

$$[0119] \quad PCM = \frac{1}{|C|} \sum_{c \in C} \frac{r_c}{|c|} - \frac{w_c}{|c|} \quad (6);$$

[0120] 其中, c 表示标签, C 为表示标签, r_c 表示当标签为 c 时代理模型和所述软件缺陷预测模型的预测结果具有相同的数量, w_c 表示当标签为 c 时代理模型和所述软件缺陷预测模型的预测结果具有不同的数量。

[0121] S32,设置因果性评价指标,用于判断各个特征的删除或扰动对因果性评价指标的输出值的影响是否超过预设值,若是,则认定特征与解释结果之间存在强因果效应。

[0122] 因果性评价指标包括:

$$[0123] \quad S = \frac{1}{M} \left| \sum_{i=1}^M g(t) - [g(T_i) + g(T_i') + g(T_i^\otimes)] \right| \quad (7);$$

[0124] 其中, M 表示最重要的特征的个数, $T_i = [\bar{t}_1, \bar{t}_2, \dots, \bar{t}_{i-1}, t_i, \bar{t}_{i+1}, \dots, \bar{t}_M]$, $T_i' = [\bar{t}_1, \bar{t}_2, \dots, \bar{t}_{i-1}, 0, \bar{t}_{i+1}, \dots, \bar{t}_M]$ 和 $T_i^\otimes = [\bar{t}_1, \bar{t}_2, \dots, \bar{t}_{i-1}, t_i^\otimes, \bar{t}_{i+1}, \dots, \bar{t}_M]$,其中 $t_i^\otimes = t_i + \text{rand}()$,若某一个特征的删除和扰动对预测结果发生变化较大,则该特征与预测结果之间的因果效应越强。

[0125] S33,设置稳定性评价指标,用于将代理模型和软件缺陷预测模型中拥有相同解释结果的实例定义为覆盖实例。

[0126] 稳定性评价指标包括:

[0127] $R = \frac{1}{r} \sum_{i=1}^r |g(t'_i) - g(t_i^*)|$ (8)。

[0128] 具体的,即代理模型 $g(t)$ 和软件缺陷预测模型 $f(x)$ 预测结果相同的实例成为覆盖实例。 $t'_i = t_i \pm \text{rand}()$ 表示 t_i 扰动实例。

[0129] S34,将一致性评价指标、因果性评价指标和稳定性评价指标进行归一化处理,得到归一化处理矩阵。

[0130] 在一示例中,将其转化为相同的量级或范围 $[0, 1]$,得到一化处理矩阵 $\mathbf{X}^* = [\mathbf{x}_1^*, \mathbf{x}_2^*, \mathbf{x}_3^*]$ 。

[0131] S35,计算归一化处理矩阵的信息熵。

[0132] 该信息熵可表示为 $E(\mathbf{x}^*) = -\sum_{i=1}^n p_i \log(p_i)$,其中, p_i 表示是指标值 $\mathbf{x}_i^*$ 在样本 $\mathbf{x}^*$ 中出现的概率。信息熵的作用是衡量概率分布的不确定性或信息量。

[0133] S36,根据信息熵计算一致性评价指标、因果性评价指标和稳定性评价指标的相对熵。

[0134] 相对熵用于衡量各个指标之间的相对重要性。相对熵的计算公式为:

$$[0135] \quad \text{Re}(\mathbf{x}^*) = \frac{E(\mathbf{x}_{\max}^*) - E(\mathbf{x}^*)}{E(\mathbf{x}_{\max}^*) - E(\mathbf{x}_{\min}^*)}$$

[0136] 其中, $E(\mathbf{x}_{\max}^*)$ 和 $E(\mathbf{x}_{\min}^*)$ 分别是指标的最大信息熵和最小信息熵。

[0137] S37,根据相对熵的值,计算一致性评价指标、因果性评价指标和稳定性评价指标的权重。

[0138] 权重的计算方法为:

$$[0139] \quad w_i = \frac{\text{Re}(\mathbf{x}_i^*)}{\sum_{j=1}^3 \text{Re}(\mathbf{x}_j^*)} \quad (9)。$$

[0140] S38,各个权重分别和对应的归一化处理矩阵加权后得到综合评估分数。

[0141] 综合评估分数包括:

$$[0142] \quad \text{score} = \sum_{i=1}^3 w_i \mathbf{x}_i^* \quad (10)。$$

[0143] 在本实施例中,首先是可解释性评价指标的设计,其次是多维度评价指标的融合。构建了一致性、因果性、稳定性三个维度的评价指标,并通过有效融合建立了评估方法,对代理模型的解释性进行了全面评估,帮助判断解释的准确性和可信度。

[0144] 实施例四

[0145] 参阅图5,本实施例提供的基于代理模型软件缺陷预测模型解释装置包括处理器、存储器以及存储在该存储器中并可在所述处理器上运行的计算机程序,例如基于代理模型软件缺陷预测模型解释程序。该处理器执行所述计算机程序时实现上述实施例步骤,例如图3所示的步骤。

[0146] 示例性的,所述计算机程序可以被分割成一个或多个模块/单元,所述一个或者多个模块/单元被存储在所述存储器中,并由所述处理器执行,以完成本发明。所述一个或多个模块/单元可以是能够完成特定功能的一系列计算机程序指令段,该指令段用于描述所述计算机程序在所述基于代理模型软件缺陷预测模型解释装置中的执行过程。

[0147] 所述基于代理模型软件缺陷预测模型解释装置可以是桌上型计算机、笔记本、

掌上电脑及云端服务器等计算设备。所述基于代理模型的软件缺陷预测模型解释装置可包括,但不仅限于,处理器、存储器。本领域技术人员可以理解,图5仅仅是基于代理模型的软件缺陷预测模型解释装置的示例,并不构成基于代理模型的软件缺陷预测模型解释装置的限定,可以包括比图示更多或更少的部件,或者组合某些部件,或者不同的部件,例如所述基于代理模型的软件缺陷预测模型解释装置还可以包括输入输出设备、网络接入设备、总线等。

[0148] 所称处理器可以是中央处理单元(Central Processing Unit,CPU),还可以是其他通用处理器、数字信号处理器(Digital Signal Processor,DSP)、专用集成电路(Application Specific Integrated Circuit,ASIC)、现成可编程门阵列(Field Programmable Gate Array,FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

[0149] 所述存储器可以是所述基于代理模型的软件缺陷预测模型解释装置的内部存储单元,例如基于代理模型的软件缺陷预测模型解释装置的硬盘或内存。所述存储器也可以是所述基于代理模型的软件缺陷预测模型解释装置的外部存储设备,例如所述基于代理模型的软件缺陷预测模型解释装置上配备的插接式硬盘,智能存储卡(SmartMedia Card,SMC),安全数字(Secure Digital,SD)卡,闪存卡(Flash Card)等。进一步地,所述存储器还可以既包括所述基于代理模型的软件缺陷预测模型解释装置的内部存储单元也包括外部存储设备。所述存储器用于存储所述计算机程序以及所述基于代理模型的软件缺陷预测模型解释装置所需的其他程序和数据。所述存储器还可以用于暂时地存储已经输出或者将要输出的数据。

[0150] 实施例五

[0151] 本实施例提供了一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,所述计算机程序被处理器执行时实现实施例二所述方法的步骤。

[0152] 所示计算机可读介质可以是任何可以包含、存储、通信、传播或传输程序以供指令执行系统、装置或设备或结合这些指令执行系统、装置或设备而使用的装置。计算机可读介质的更具体的示例(非穷尽性列表)包括以下:具有一个或多个布线的电连接部(电子装置),便携式计算机盘盒(磁装置),随机存取存储器(RAM),只读存储器(ROM),可擦除可编程只读存储器(EPROM或闪速存储器),光纤装置,以及便携式光盘只读存储器(CDROM)。另外,计算机可读介质甚至可以是可在其上打印所述程序的纸或其他合适的介质,例如通过对纸或其他介质进行光学扫描,接着进行编辑、解译或必要时以其他合适方式进行处理再以电子方式获得所述程序,然后将其存储在计算机存储器中。

[0153] 上述实施例只是为了说明本发明的技术构思及特点,其目的是在于让本领域内的普通技术人员能够了解本发明的内容并据以实施,并不能以此限制本发明的保护范围。凡是根据本发明内容的实质所做出的等效的变化或修饰,都应涵盖在本发明的保护范围内。

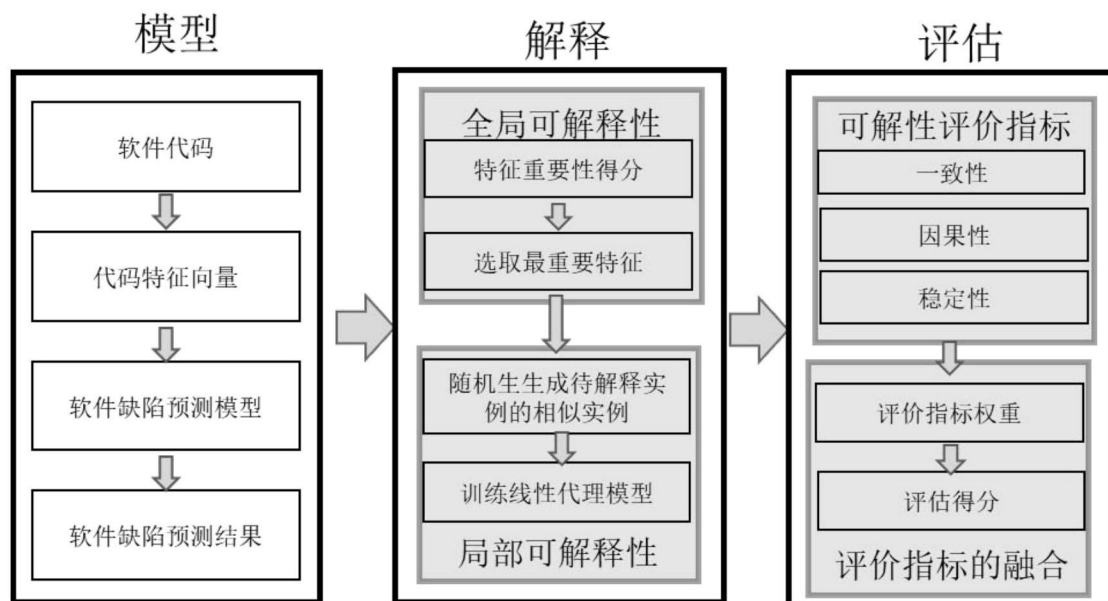


图1

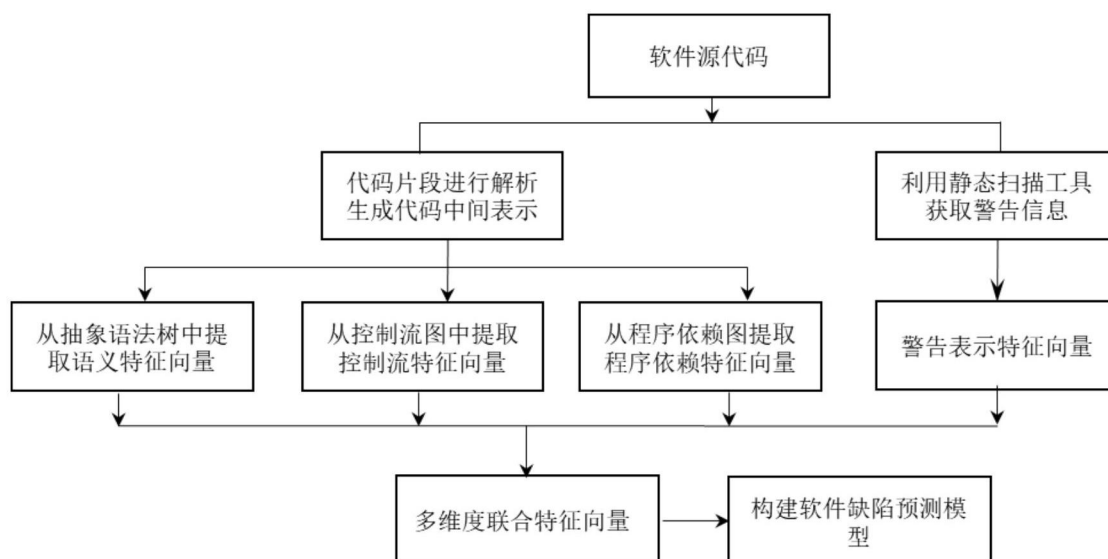


图2

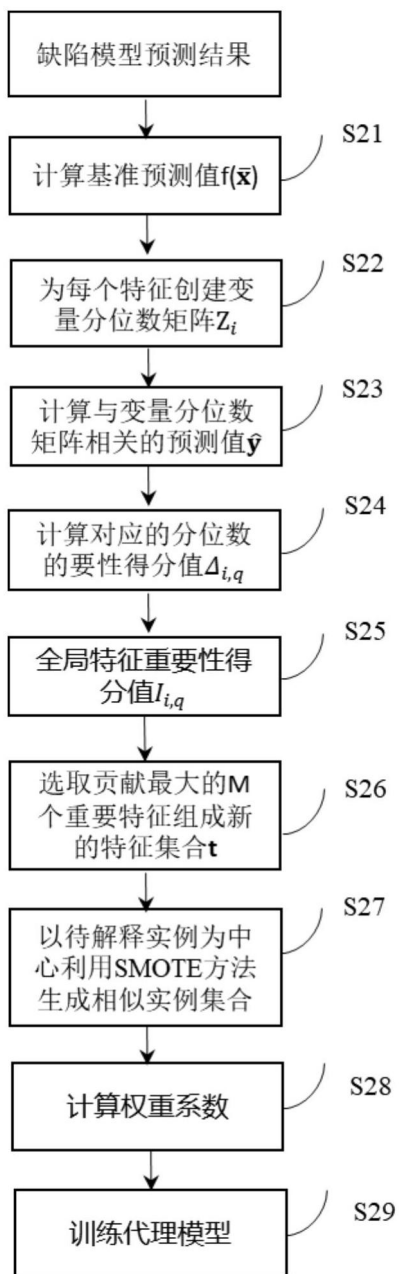


图3

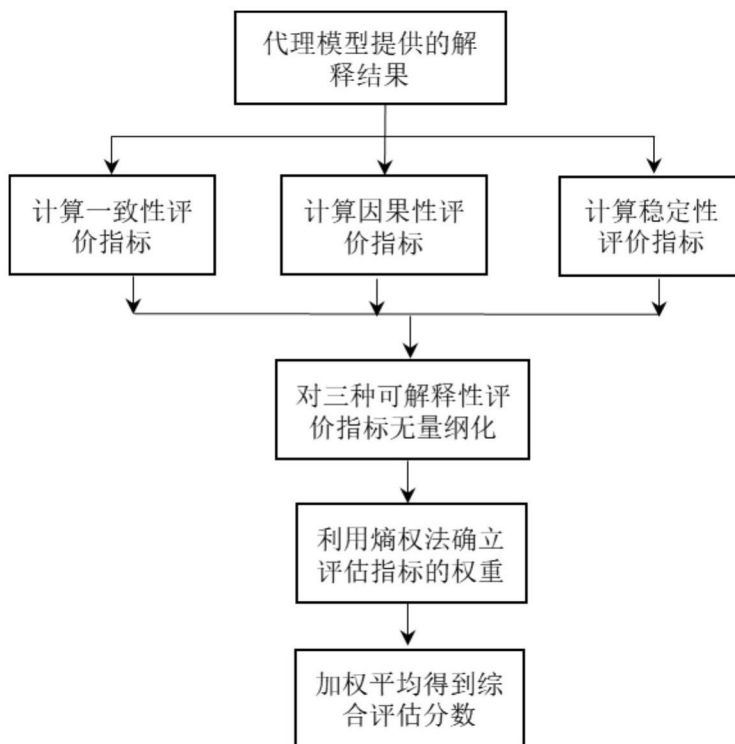


图4

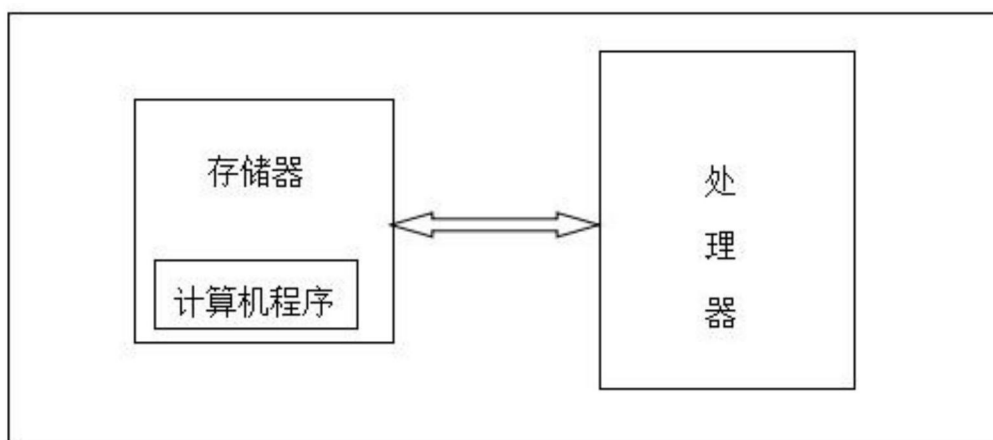


图5