ISCAS 中国科学院软件研究所学术年会'2022
暨计算机科学国家重点实验室开放周

学术论文

MultiCode: A Unified Code Analysis Framework based on Multi-type and Multi-granularity Semantic Learning

段旭 吴敬征 杜梦男 罗天悦 杨牧天 武延军

吴敬征 电话：18910958184 邮箱： jingzheng08@iscas.ac.cn

背景与动机

近年来，深度学习技术发展迅速。越来越多的代码分析方法使用该技术来提升准确率并降低人力开销。这类方法能够从大量现有代码数据中学习潜在规律，并根据学习到的规律对未知代码数据进行预测。与传统的基于规则的方法相比，这类方法不依赖大量的专家知识，并且能够覆盖复杂多变的代码情况。基于上述优点，基于深度学习的代码分析方法在工业界和学术界得到了广泛的研究。

在这些研究中，大多数方法针对特定的代码分析任务，引入针对性的设计，以达到更好的模型性能。在工业领域，该现象会使开发者在开发涉及多需求的代码分析平台时，面临开发开销大、模型集成困难、可扩展性受限等问题。例如，开发者在开发能够实现漏洞检测、代码克隆检测等任务的通用代码分析平台时，需要实现不同的模型，这通常涉及很多步骤，例如特征建模、编码器设计、解码器设计、损失函数设计等，并且由于模型的不同，该过程通常需要重复多次。此外，在集成不同的模型时，开发者需要实现不同的接口，管理不同的模型。当新需求出现时，开发者需要根据不同任务的特征重构模型。甚至当遇到一些数据不足的任务时，由于模型参数无法复用，预测性能难以达到可用的水平。

- **额外的开发开销**
  - (特征建模 -> 编码器设计 -> ... -> 损失函数设计) *x n*
- **模型集成困难**
  - 不同接口实现、模型管理、模型测试、...
- **可扩展性受限**
  - 重新实现模型、参数不同导致数据不足时性能较差、...

挑战

为解决上述问题，关键在于对代码的语义信息进行学习，从而对不同任务中需要的特征进行建模。然而，代码中的语义信息有多种类型，以及多种粒度。例如，在检测漏洞时，我们需要对控制依赖和数据依赖进行建模，以发现从Source到Sink的危险路径。在检测相似代码时，我们需要对语法结构进行建模，以揭示代码的各个组成部分是如何嵌套的。

代码语义信息

•类型:
- 语法结构
- 控制流
- 控制依赖
- 数据依赖
- ...

•粒度:
- 语句
- 函数
- 文件
- ...

使用示例

识别相似代码

发现缺失的输入验证

识别函数意图

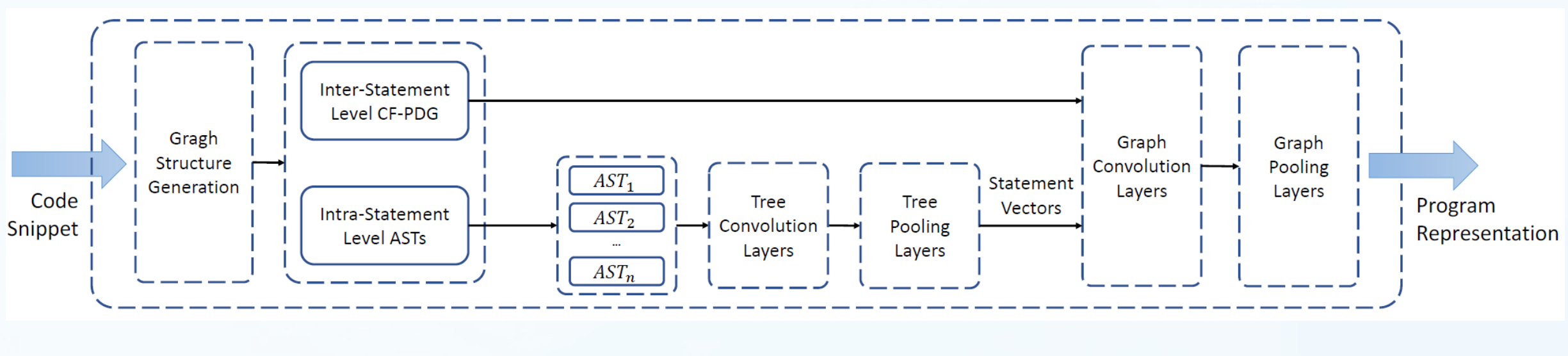
发现函数调用中不当的参数

...

因此，有以下两个关键问题需要回答：
•1）如何在代码中表达语义信息？
•2）如何设计神经网络模型学习上述语义信息？

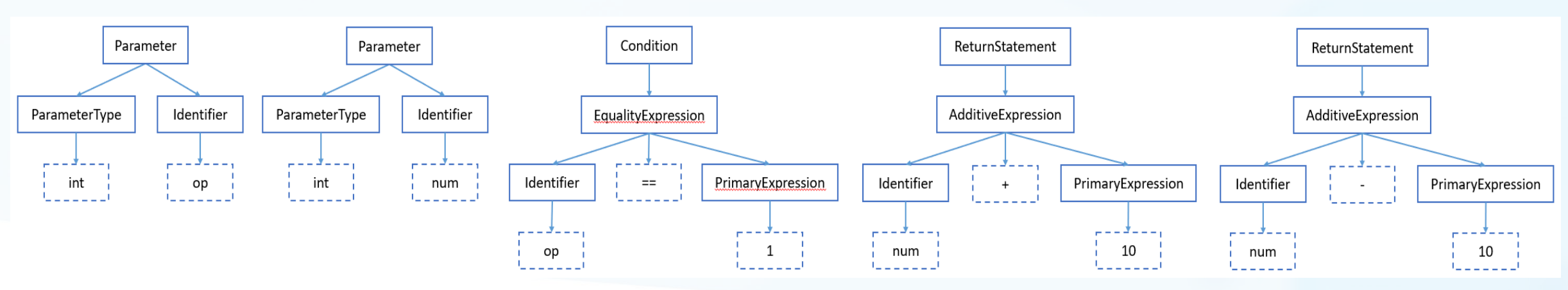
方法设计

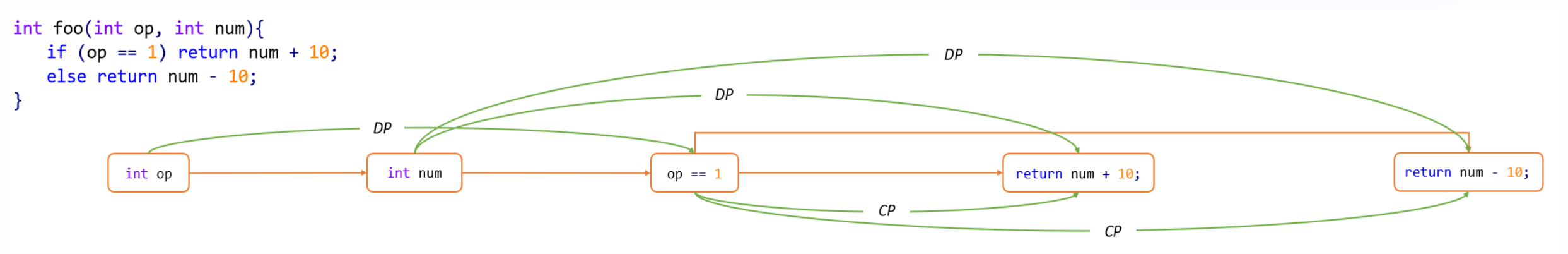
为了解决上述问题，我们提出了一个统一的代码分析框架——MultiCode，其特点如下：
•使用树结构和图结构来表达不同类型和不同粒度的语义。
•使用树卷积神经网络和图卷积神经网络分别处理不同的结构。
•不同粒度的语义信息通过自下而上的方式学习。
•能够作为编码器简单地适配到不同的代码分析任务中。



MultiCode的总体框架可以分为三个步骤：

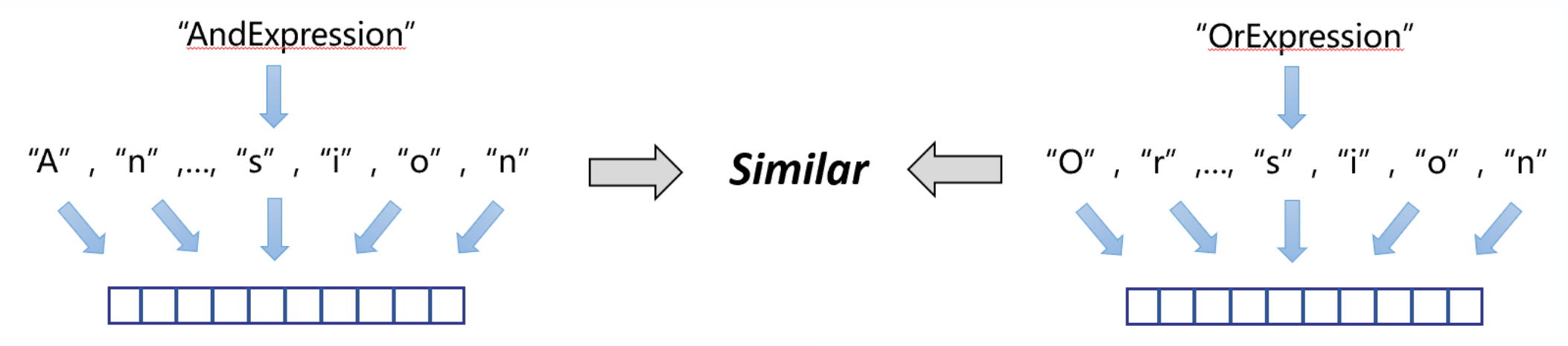
步骤1：将代码段转换为语句内AST和语句间CF-PDG。其中，语句内AST对语句内的语法结构建模，语句间CF-PDG对语句间的控制流、控制依赖、数据依赖建模。



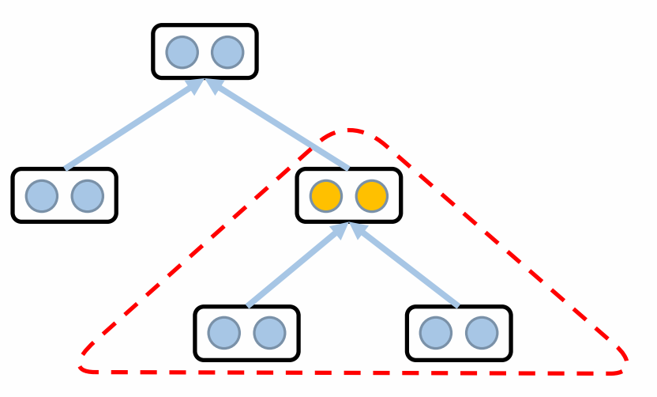


```
int foo(int op, int num){
    if (op == 1) return num + 10;
    else return num - 10;
}
```

步骤2：使用树卷积神经网络学习语句内AST的语义，得到语句的向量表示。在学习之前，使用PACE算法根据token字符独热编码的线性组合对节点初始特征进行编码。由于AST中相似语义token具有相似字符，该方法能够在编码token语义的同时避免OOV问题。

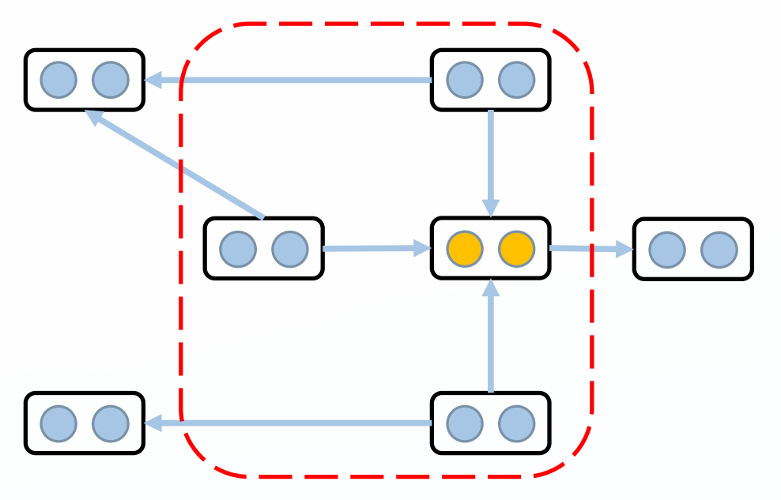


此后，使用基于树的卷积根据子节点特征更新父节点特征，并使用基于树的池化对所有节点表示进行聚合。



$$\vec{h}_0' = \sigma(\sum_{i=0}^{n} \boldsymbol{W}_i \vec{h}_i + \boldsymbol{b})$$

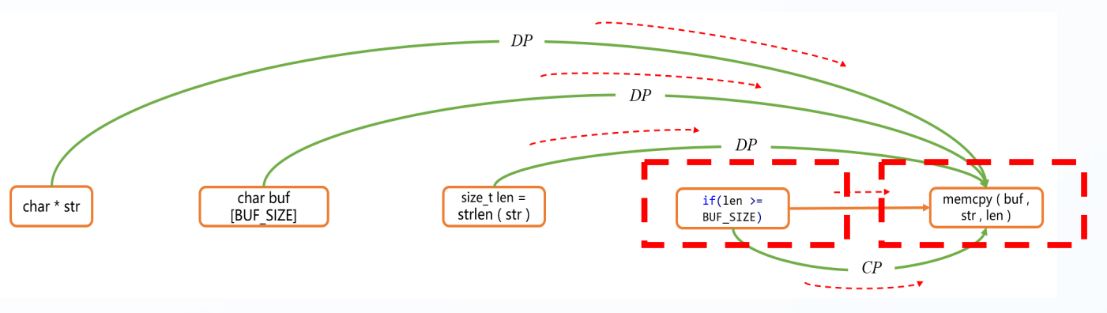
步骤3：基于语句向量表示，使用图卷积神经网络学习语句间CF-PDG的语义信息，得到代码段的向量表示。具体地，在每次卷积时，根据邻接节点的表示更新中心节点的表示，并使用基于图的池化将所有节点的表示聚合为代码段的表示。



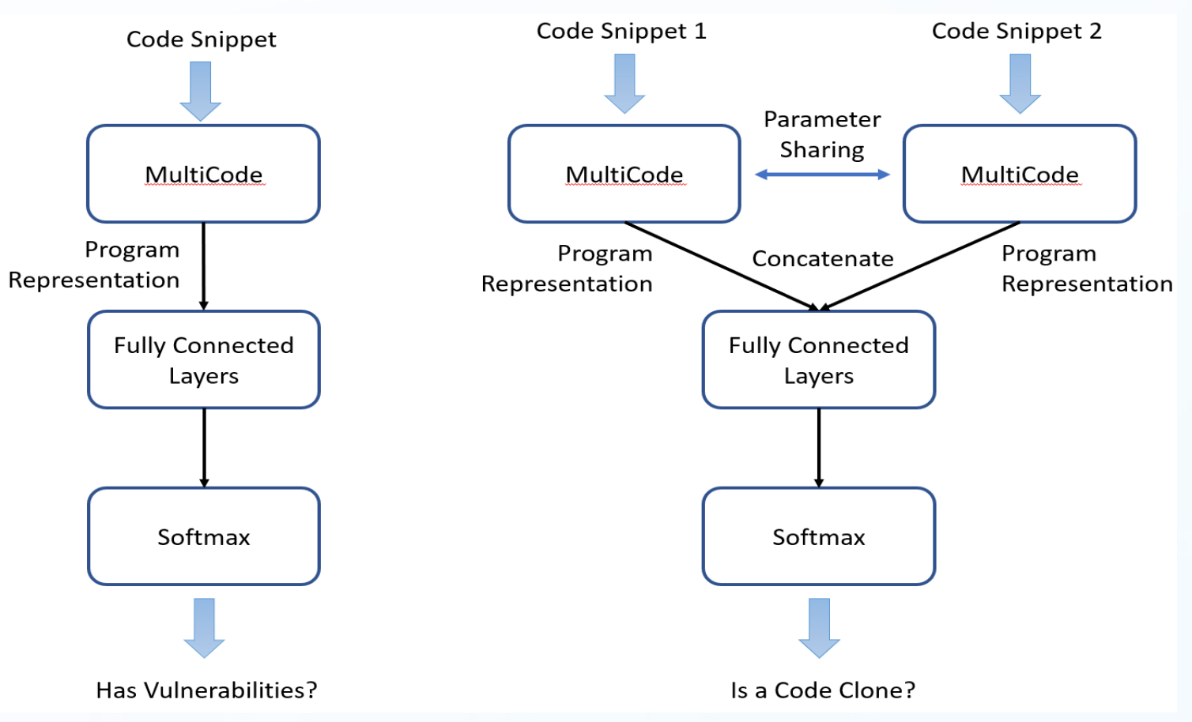
$$\vec{h}_0' = \sigma(\sum_{i=0}^{n} \alpha_{0i} \boldsymbol{W} \vec{h}_i)$$

在一些任务中，不同节点对于中心节点具有不同的重要性，因此，使用注意力机制捕获不同节点的重要性差异。例如在漏洞检测时，条件语句节点对于敏感API调用语句节点具有更高的重要性。

$$\alpha_{0i} = \frac{exp(\sigma(\vec{a}^T(\boldsymbol{W}\vec{h}_0 || \boldsymbol{W}\vec{h}_i || \vec{h}_{(0,i)})))}{\sum_{k=0}^{n} exp(\sigma(\vec{a}^T(\boldsymbol{W}\vec{h}_0 || \boldsymbol{W}\vec{h}_k || \vec{h}_{(0,k)})))},$$



步骤4：将MultiCode作为编码器适配于各种代码分析任务。例如，在后面加入全连接层和softmax实现分类，或者构建孪生结构网络实现双端输入的处理。

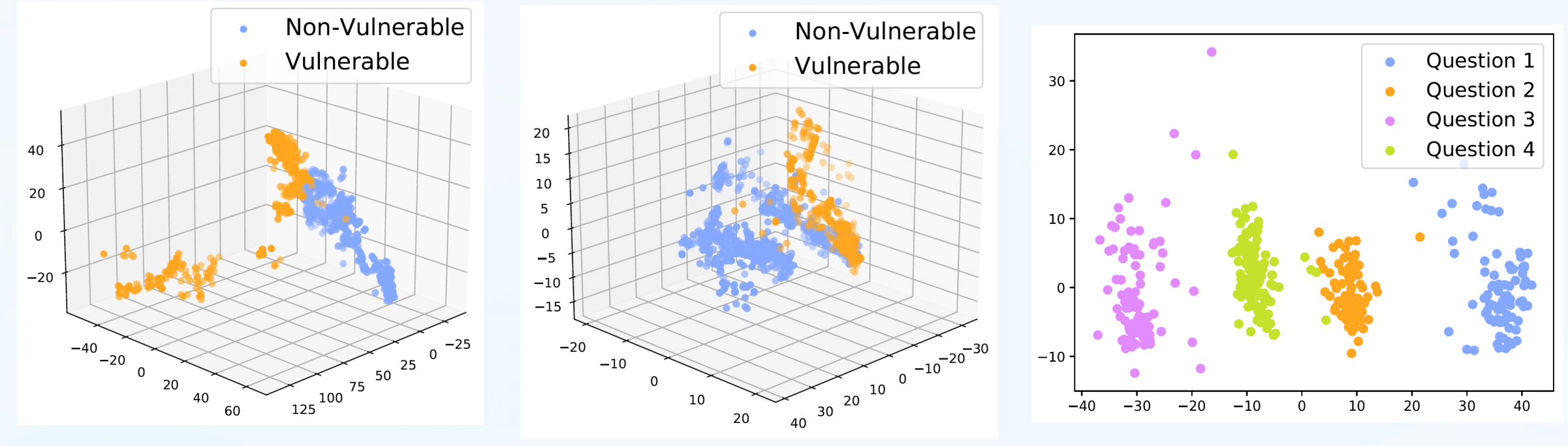


实验评估

本文在漏洞检测和代码克隆检测两个任务上对MultiCode进行评估。评估结果表明，MultiCode能够在两个任务中的三个数据集上取得比baselines更优的结果，其中在SARD-BE和SARD-RME数据集上的F1分数分别达到94.6%和92.5%，表明其能够学习多种漏洞的语义。MultiCode在OJClone数据集上的F1分数达到97.1%，表明其能够在一定程度上识别代码的语义。

Dataset	Tools	FPR (%)	FNR (%)	R (%)	P (%)	F1 (%)
SARD-BE	Flawfinder	82.0	11.5	88.5	38.4	53.6
	RATS	65.1	27.0	73.0	39.3	51.1
	TBCNN	25.2	41.4	58.6	60.9	59.7
	MultiCode	0.4	9.5	90.5	99.2	94.6
SARD-RME	Flawfinder	50.4	45.5	54.5	31.8	40.2
	RATS	38.5	55.2	44.8	33.4	38.3
	TBCNN	3.0	55.5	44.5	87.7	59.0
	MultiCode	6.1	1.7	98.3	87.4	92.5
OJClone	Deckard	0.7	92.0	8.0	92.1	14.7
	TBCNN	4.9	1.4	98.6	95.3	96.9
	MultiCode	1.4	4.4	95.6	98.6	97.1

此外，通过PCA降维对MultiCode输出的代码段表示进行可视化，发现不同类别的代码表示明显地聚集在不同簇中，表明MultiCode能够有效地捕获不同类别代码的语义差异。



(a) SARD-BE dataset *(b) SARD-RME dataset* *(c) OJClone dataset*

研究成果

本文发表在ISSRE 2021会议的Industry Track中，并被评为Best Practice Paper。

IEEE computer society IEEE

The 32nd International Symposium on Software Reliability Engineering (ISSRE 2021)

BEST PRACTICE PAPER

This certificate is awarded to:

Xu Duan

for

MultiCode: A Unified Code Analysis Framework based on Multi-type and Multi-granularity Semantic Learning

presented at the

ISSRE 2021

held in Wuhan, China from October 25th to October 28th, 2021

Zhi Jin, Xuandong Li, Jianwen Xiang
ISSRE 2021 General Chairs

Chunhui Yang, Valerio Formicola, Guangtai Liang
ISSRE 2021 Industry Program Committee Chairs