|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 论文题目 | Software defect prediction with semantic and structural information of codes based on Graph Neural Networks | | | |
| Paper URL | https://www.sciencedirect.com/science/article/pii/S0950584922001677 | | | |
| Project URL | <https://github.com/XiaoYingJian/CGCN> | | | |
| 综述/背景介绍 | 发展状况 | 原因 | 意义 | 关键词（速记词汇、信息索引词汇） |
| 现有的研究通常使用手工设计的静态指标来构建预测器。然而，对某个项目表现良好的手工制作的特征在其他项目中可能表现不佳。源代码除了一系列代码度量的特征外，还具有丰富的语法和语义信息。  抽象语法树（AST）是源代码语法结构的抽象表示。  软件也可以被抽象成一个基于类之间依赖关系的粗粒度网络结构，即类依赖网络（CDN）。每个类被看作是一个节点，而类之间的依赖关系是有向边。  本文提出CGCN，它结合了CNN和GCN的优势，首先将源代码解析为AST和软件依赖网络。然后用CNN来捕捉AST中的语义信息，而用GCN来捕捉软件网络中的结构信息。然后，我们用不同的权重将这两类学习到的特征结合起来，得到融合特征。最后，融合特征与传统的手工制作的特征相连接，以训练一个预测器。 | | |  |
| 假设 |  | | |  |
| 方法描述(含图) | CGCN的工作流程可以分为三部分：   1. 生成内部特征   1.源代码解析为AST  2.形成token序列，映射成数字向量。  3.CNN生成内部特征。   1. 生成外部特征   1.构建类依赖网络  2.生成CDN的初始化节点属性  3.使用GCN生成外部特征   1. 融合特征   CNN的最大池化层和GCN最后一层卷积层合并。 | | |  |
| 实验设计 | 1. 数据集 PROMISE 2. Baseline模型 3. 传统度量 4. CNN 5. BiLSTM 6. Deepwalk 7. Node2defect 8. GCN2defect 9. GAT 10. 评估指标   准确率accuracy、精确率precision、召回率recall、F1值、AUC   1. 研究问题   RQ1:CGCN方法适合哪种任务（版本内，跨版本，或跨项目）？  RQ2:特征组合权重策略对提出的CGCN方法是否有重大影响？  RQ3:CGCN方法是否比基线模型表现得更好？ | | |  |
| 数据处理 | 输入 | 筛除特例 | 处理方式 | 关键词（速记词汇、信息索引词汇） |
|  |  |  |  |
| 结论 | 提出了CGCN方法，结合了CNN和GCN的优势，自动提取每个文件的内部语义信息和所有类文件之间的外部结构信息。给这两类特征分配了不同的权重，因为这两类特征对不同的项目可能有不同的贡献。结果表明，我们提出的方法比基线模型产生了更稳定和更好的结果。CGCN方法更适用于版本内预测。 | | | |
| 局限性分析 |  | | | |

2.论文总结：

**（论文名）：**基于图神经网络的代码语义和结构信息的软件缺陷预测

**（题目）**：Software defect prediction with semantic and structural information of codes based on Graph Neural Networks

**（论文URL）：**https://www.sciencedirect.com/science/article/pii/S0950584922001677

**（项目URL）：**<https://github.com/XiaoYingJian/CGCN>

**（总结）：**

本文提出了CGCN，它是一种结合了CNN和GCN优势的缺陷预测方法。源代码有丰富的语法和语义信息，忽视源代码的上下文信息可能会大大影响对程序的理解。本方法将提取源代码的语义和结构信息，以获得更准确的SDP。本方法首先将源代码解析为AST和软件依赖网络。然后用CNN来捕捉AST中的语义信息，而用GCN来捕捉软件网络中的结构信息。然后，我们用不同的权重将这两类学习到的特征结合起来，得到融合特征。最后，融合特征与传统的手工制作的特征相连接，以训练一个预测器。实验结果表明，CGCN方法比基线模型产生了更稳定和更好的结果，并且更适用于版本内预测。

在未来将更加关注跨版本和跨项目的缺陷预测，并加强模型的可转移性，以提高在跨版本和跨项目预测中的性能。并且将来将自动学习内部特征和外部特征的权重。

**（附图）：**



























                           