深度学习在软件测试中的应用

北京邮电大学

邢颖

目录

1 深度学习大背景下软件工程的现状

背景——ChatGPT点燃的人工智能之火

火爆全球的ChatGPT

2022年11月30日，一经发布ChatGPT就像是一股热浪铺面而来，迅速爆火全球互联网圈，从中文互联网到国外互联网，我们都能看见ChatGPT用户的存在。

01

02

生成式AI的集大成之作

- 美国人工智能研究实验室OpenAI新推出的一种人工智能技术驱动的自然语言处理工具
- 拥有语言理解和文本生成能力，通过连接大量的语料库进而训练模型
- 不单是聊天机器人，还能进行撰写邮件、视频脚本、文案、翻译、代码等任务

背景——ChatGPT点燃的人工智能之火

- 随着ChatGPT的发布，点燃了人们对人工智能的热情，也同样引起了一场“人类将会被AI替代”的恐慌。
- 谁都知道现阶段人工智能难以落地，但几乎是所有人，相关的、不相关的领域都争先恐后的涌入这项技术的研究当中。
- 大家普遍人为即使现在还未在自己身上发送“机器替代人工”，但一旦出现这种可能性，它带来的变革将十分迅速。
- 谁都担心在这场变革中落后于他人，从而导致失去先机，甚至是被淘汰。

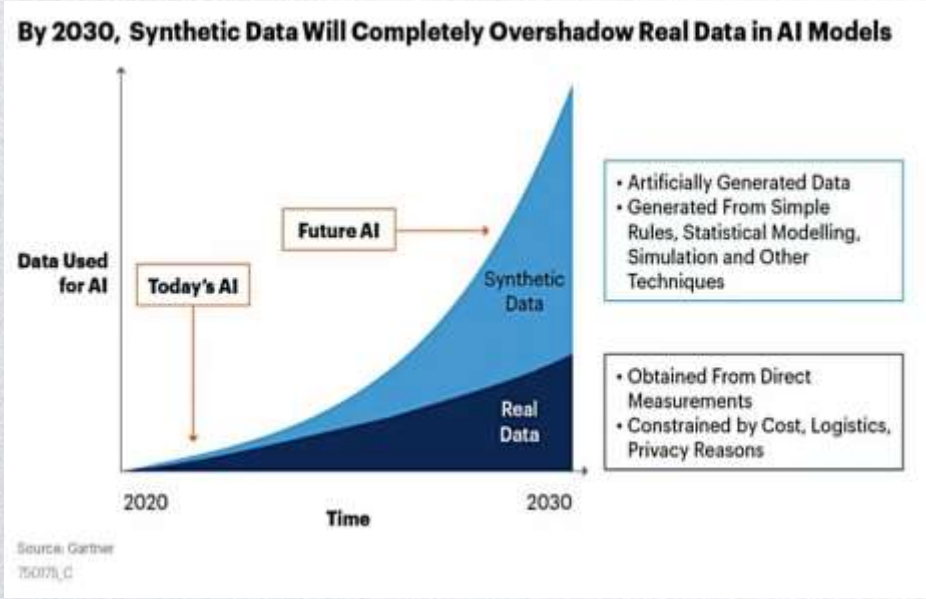
背景——ChatGPT点燃的人工智能之火

合成数据加速构建AI赋能的大型虚拟世界

Future for AIGC

➢Grand View Research预测，AI训练数据市场规模到2030年将超过86亿美元。

➢Gartner预测，到2024年用于训练AI的数据中有60%将是合成数据，到2030年AI模型使用的绝大部分数据将是人工智能合成的。



背景——软件测试在软件生命周期中的意义

- **确保软件质量**：软件测试可以帮助发现软件中的缺陷、错误和漏洞，并确保软件能够满足用户的需求和期望。通过测试，可以大大减少软件发布后出现问题的可能性，提高软件质量和可靠性。
- **降低开发成本**：软件测试可以帮助在软件开发早期发现问题，并在早期进行修复，从而避免在后期发现问题时需要付出更高的成本。此外，测试还可以帮助开发人员更好地理解用户需求，从而减少开发错误和重新工作的需要。
- **提高用户满意度**：通过测试，可以确保软件符合用户需求和期望，并且具有良好的性能和可用性。这将有助于提高用户对软件的满意度，从而增加用户忠诚度和重复使用率。

现状——自动化测试框架的痛点

虽然自动化测试框架可以提高测试效率和减少测试成本，但是在实践中，也存在一些痛点。

- **学习成本高**：需要一个具备测试开发能力的开发人员手动写测试代码。自动化测试需要编程技能和专业知识，因此对测试人员和开发人员的技能门槛较高。一些测试人员可能缺乏编程技能或没有足够的时间和资源来学习。
- **维护成本高**： 自动化测试框架需要不断更新和维护，以确保其与软件的最新版本兼容。这需要额外的时间和资源，并且可能需要专业的技能和知识。
- **跨平台**： 自动化测试框架可能需要在多个操作系统、浏览器和设备上运行，但是不同的平台可能存在兼容性问题，从而导致测试结果不准确或失败。

- **可靠性**：自动化测试框架需要在各种环境和情况下稳定运行，但是可能会受到一些因素的影响，如不稳定的测试环境、网络问题、系统故障等。
- **脚本维护**：自动化测试脚本需要不断更新和修改以适应软件的变化，但是这可能会导致测试脚本的维护成本很高，特别是当软件变化频繁时。

2 深度学习给软件测试带来的机遇和挑战

机遇、技术、融合、前景

机遇

生成式AI取得算法突破，AIGC进入应用爆发期，创造巨大经济价值。

技术

Diffusion、GPT-3、CLIP等深度学习模型已经相继成熟，内容生产模式过渡到AI辅助内容生成阶段。

融合

软件测试在形成新的创新范式同时，需要借助深度学习的加持，完成面向严肃领域的创意设计与智能软件平台与系统。

前景

当前AIGC正经历一个渗透率快速提升的阶段，为深度学习行业打开全新的成长空间。

既是机遇也是挑战——深度学习对软件测试领域的影响

- 缩短测试时间
- 提高软件测试的精确度
- 非功能性测试
- 降低测试成本

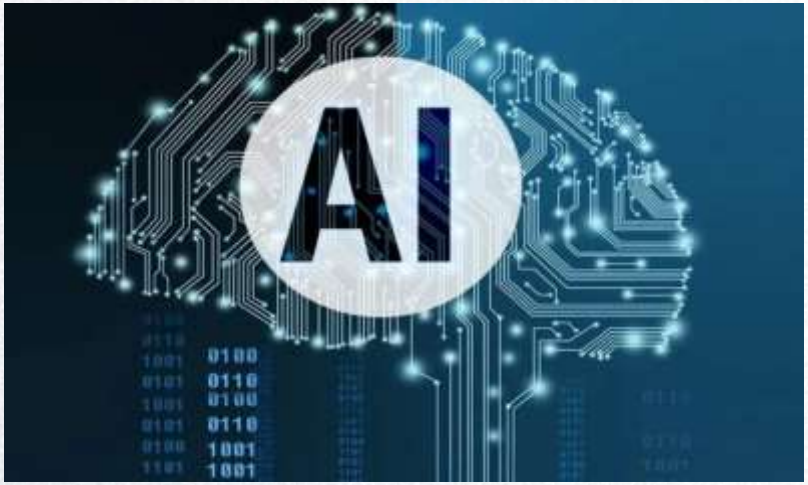
大势所趋，不可避免

总之，深度学习将会对软件测试领域产生巨大的影响。在未来的软件测试中，深度学习将不可避免地成为必不可少的工具，以帮助企业提高软件质量并优化开发流程。

同时，测试人员也需要学习更多的计算机科学知识和技能来充分反思机器学习算法的输出，并在检查测试结果时发挥人类思维的重要作用。

机遇——深度学习在软件测试领域的广泛应用

- **自动化测试脚本生成**：使用自然语言处理技术将测试需求转换为自动化测试脚本，从而降低测试人员和开发人员的技能门槛。这可以大大提高测试效率和减少测试成本。
- **自动化测试用例生成**：使用 深 度 学习技术学习软件的行为和性能，并根据学习结果自动生成测试用例。这可以帮助测试人员发现更多的缺陷和漏洞，并提高软件的质量。
- **缺陷预测和分析**：使用 深 度 学习技术对软件的历史数据进行分析，预测未来可能出现的缺陷和漏洞。这可以帮助测试人员提前发现和解决问题，从而提高软件的可靠性和稳定性。



- **自动化测试结果分析**：使用 深 度 学习技术对自动化测试结果进行分析和归纳，从而帮助测试人员快速识别和解决问题。这可以提高测试效率并加快软件发布速度。

挑战——深度学习带来的挑战

- **数据准备和清洗**：深度学习算法需要大量的数据来训练和学习，但是这些数据可能存在噪音和错误。因此，测试人员需要花费大量的时间和精力来准备和清洗数据，以确保生成式AI的准确性和可靠性。
- **可解释性**：深度学习算法往往难以解释其决策过程，这使得测试人员很难理解它是如何生成测试用例和脚本的。这也可能导致测试结果难以验证和复现。
- **软件复杂性**：现代软件系统往往非常复杂，涉及到多个层次和组件。这使得测试用例和脚本的生成变得更加困难，并且可能需要使用更复杂的算法和技术来处理。

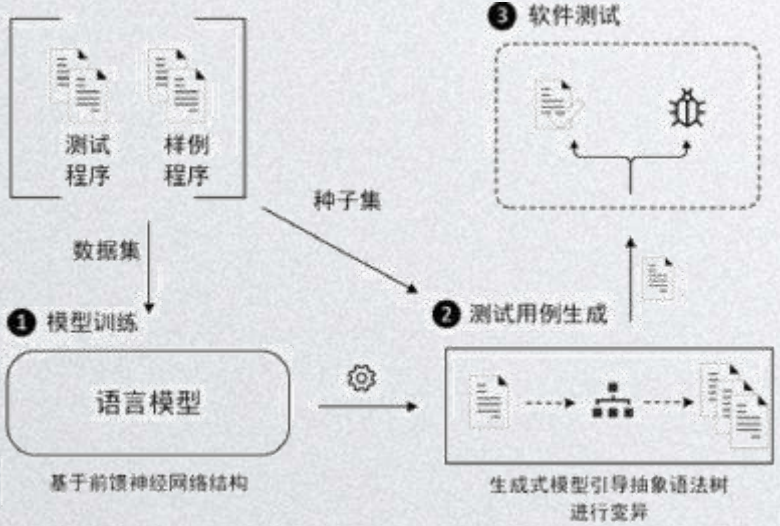
- **缺陷定位和修复**：深度学习算法可以帮助测试人员快速发现缺陷和漏洞，但是定位和修复这些问题可能需要更多的人工干预和专业知识。

3 深度学习在软件测试的应用

1. 测试用例自动生成

深度学习可以通过分析历史测试用例与结果,自动学习测试用例的模式与规律,然后生成新的测试用例。这可以规避测试用例的遗漏,扩充用例的覆盖面。

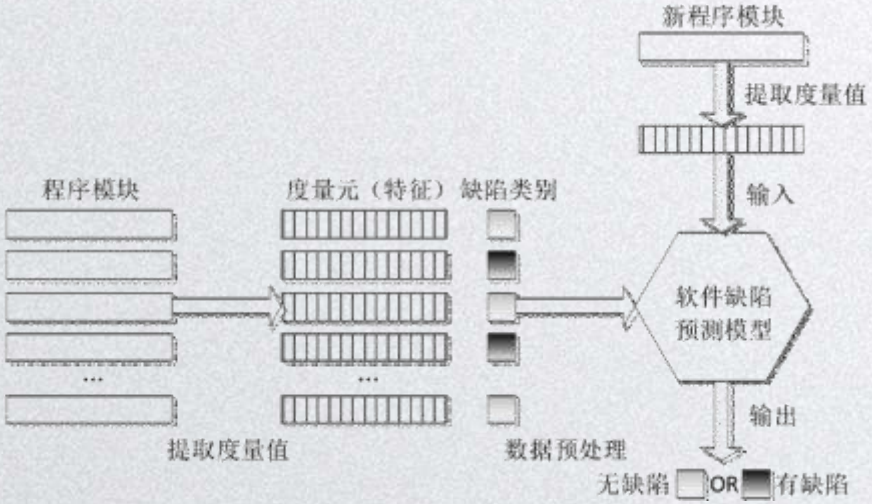
采用神经网络可以生成符合语法与结构的测试用例。这需要输入大量的正样本测试用例与负样本非测试用例进行训练。



2. 测试结果的智能分析

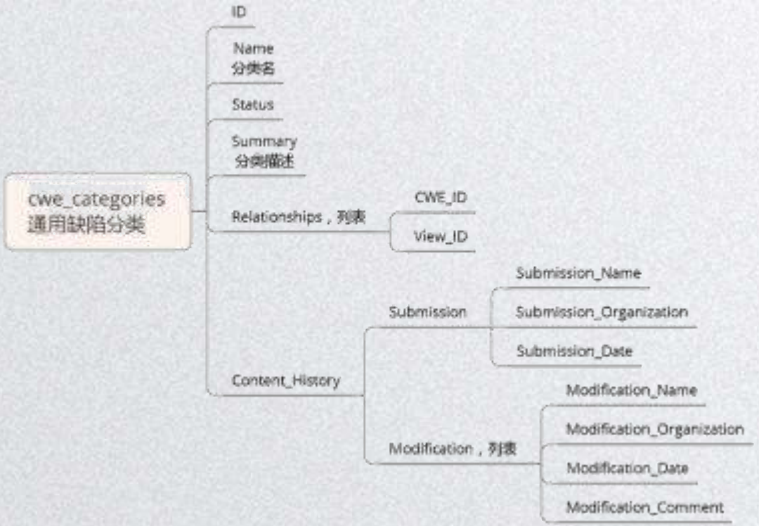
深度学习可以对大量的测试结果与日志进行深入分析,找出测试缺陷的模式与规律,产生测试报告与缺陷列表,帮助测试人员快速定位问题。

通过对历史测试结果与缺陷报告的深度学习,可以建立软件缺陷的检测模型,然后自动分析新测试结果中的缺陷。这可以识别新的缺陷模式和类型。



3. 智能化漏洞信息生态开发的方向与构想

这部分所需要解决的主要问题是如何应对种类繁多，且可能层出不穷的各种漏洞知识信息以及新的组件与漏洞信息的连接关系。因此如何设计一个框架以满足开源组件来满足知识图谱构建的自动性和可扩展性需求是本研究点所探讨的主要问题。基于开发的漏洞信息知识图谱，可以实现组件信息、漏洞信息进行查询功能，实现子图检索、路径查找等功能

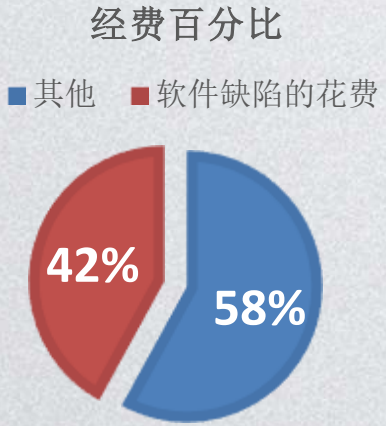


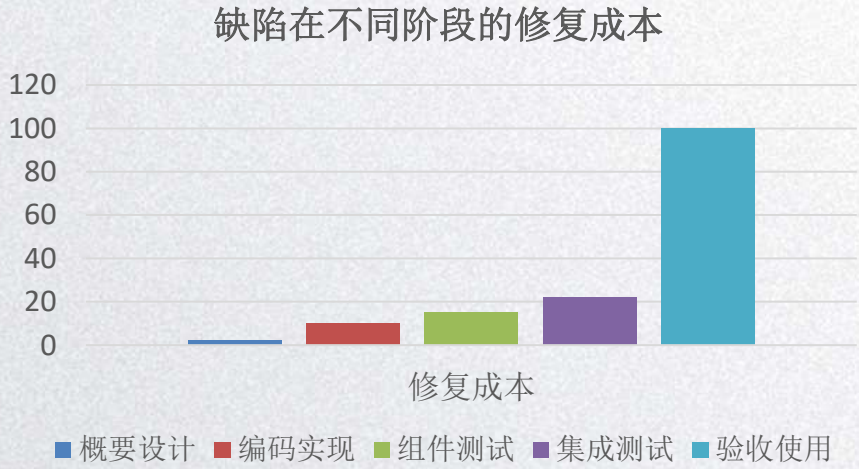
4 基于深度学习的测试结果的智能分析

测试结果智能分析的目的和意义

➢ 软件缺陷需要被尽快检验并修复

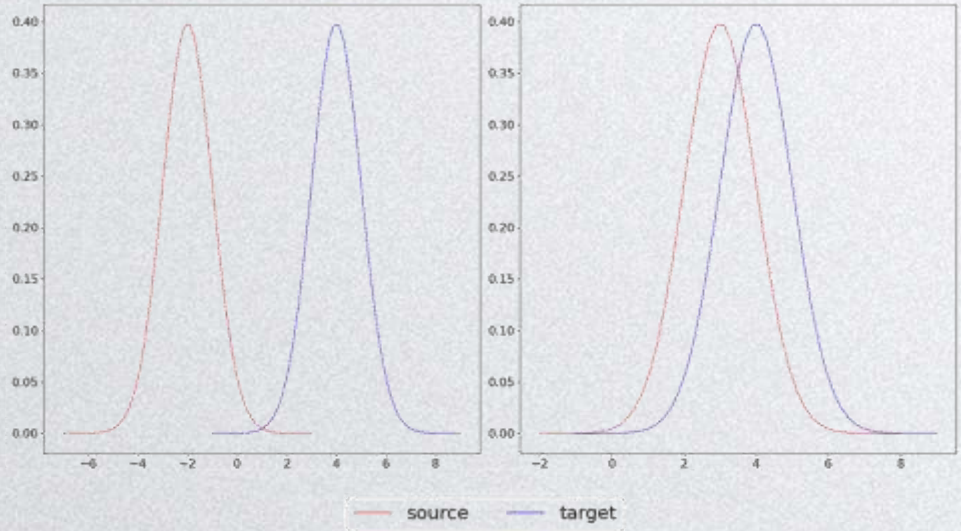
- 据统计，在软件产品中，大约**42%**的经费用于与软件缺陷相关的工作。
- 缺陷**越早**被发现，需要的修复成本就**越低。**





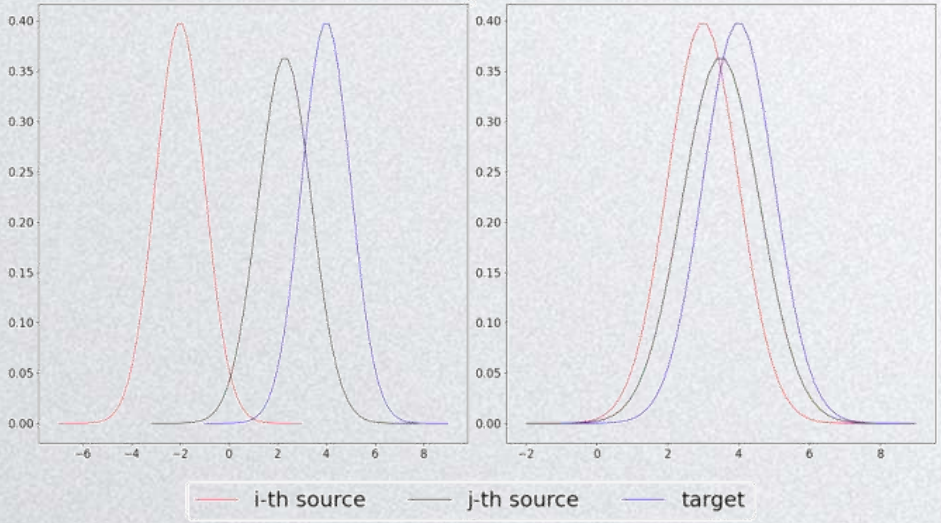
测试结果智能分析的目的和意义

- **软件缺陷预测缺乏足够的数据支撑**
- 在实际的软件开发中，需要进行预测的项目可能是一个新启动项目或项目内标签数据较少，无法直接使用传统的缺陷预测方法。
- **跨项目缺陷预测方法是在一个带标签源项目上训练，在目标项目上进行预测。**

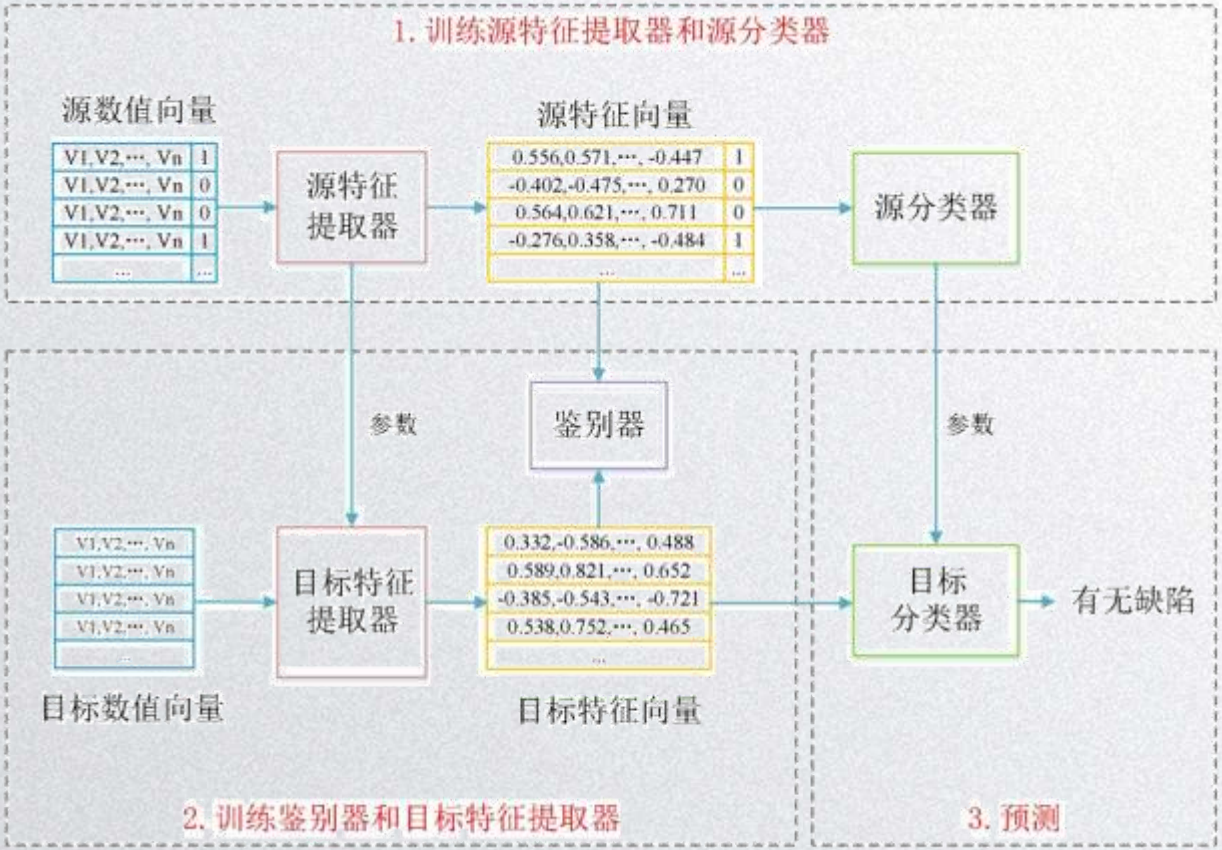


测试结果智能分析的目的和意义

- **在实际进行跨项目软件缺陷预测时，常常有多个带标签项目可以使用**
- 跨项目软件缺陷预测模型在与目标项目相似的项目上训练会产生更好的结果。实际操作中，不同项目的数据分布可能相差很大。
- **考虑到不同项目之间的分布差异，多源域适应可以解决这一问题。**



基于 GAN 的跨项目软件缺陷预测模型



基于 GAN 的跨项目软件缺陷预测模型

基于抽象语法树和CBOW的源代码特征向量获取

基于 GAN 的跨项目软件缺陷预测模型

基于抽象语法树和CBOW的特征提取

```
Package Com;
public class A{
   public static void main(String[]args){
      int i = 0;
      while( i < 10 ) {
        i ++;
      }
      System.out.println(i)
   }
}
```

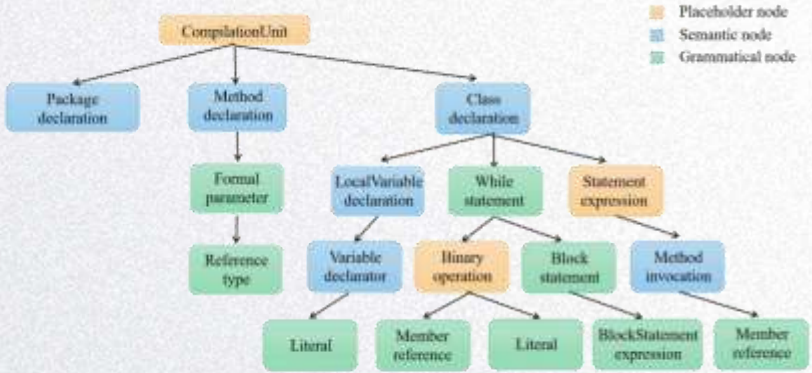
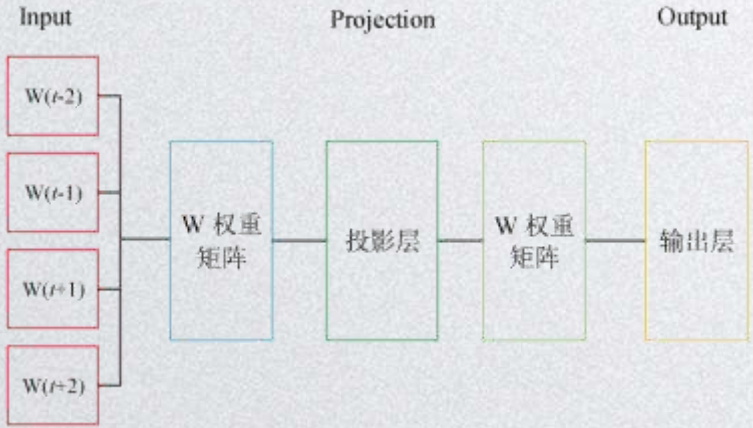
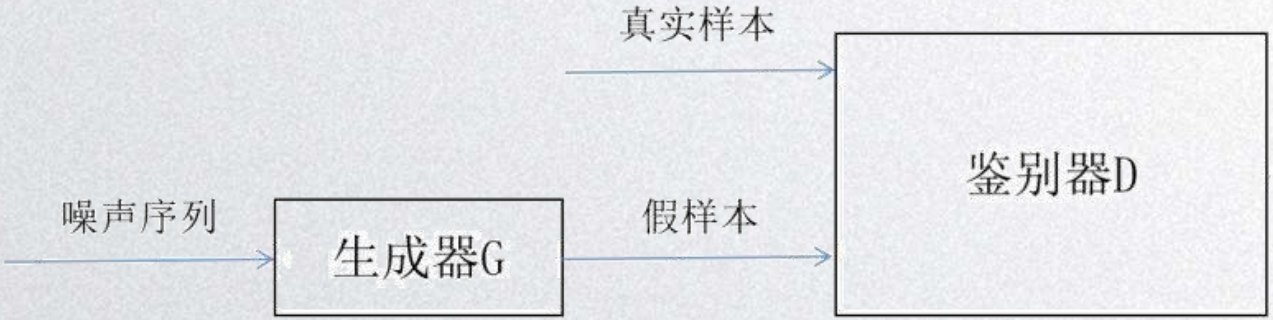


图 2　Java 代码片段及对应的抽象语法树



基于GAN的特征提取和数据迁移

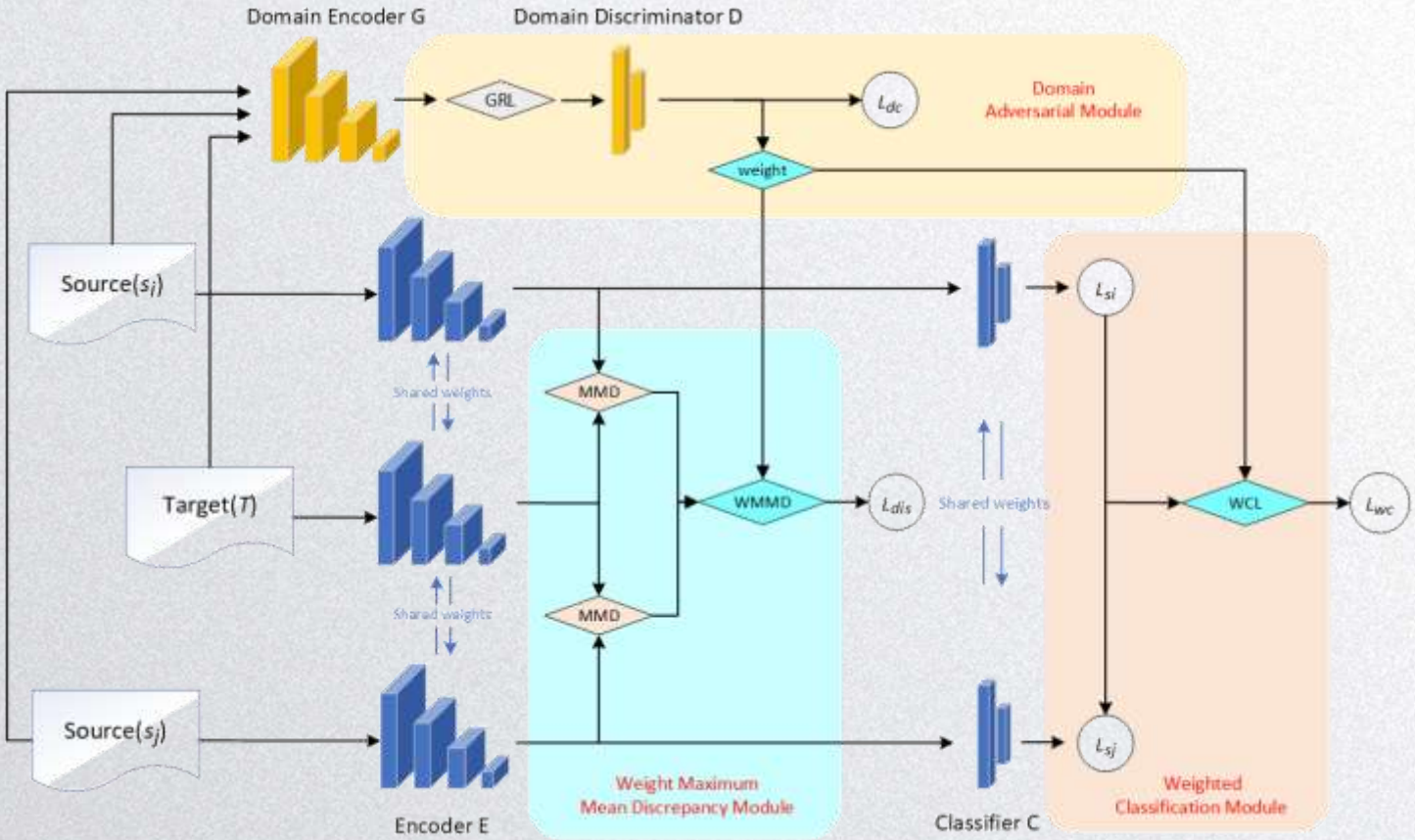


生成网络的目的是尽可能使得生成样本逼近真实样本，混淆判别网络,使判别网络无法识别假样本。而判别网络的目的则是甄别虚假样本，使其无法混入真实样本中。两者进行博弈，直至假样本分布与真实样本基本相同。

➢ 多源域适应在跨项目软件缺陷预测应用的探究

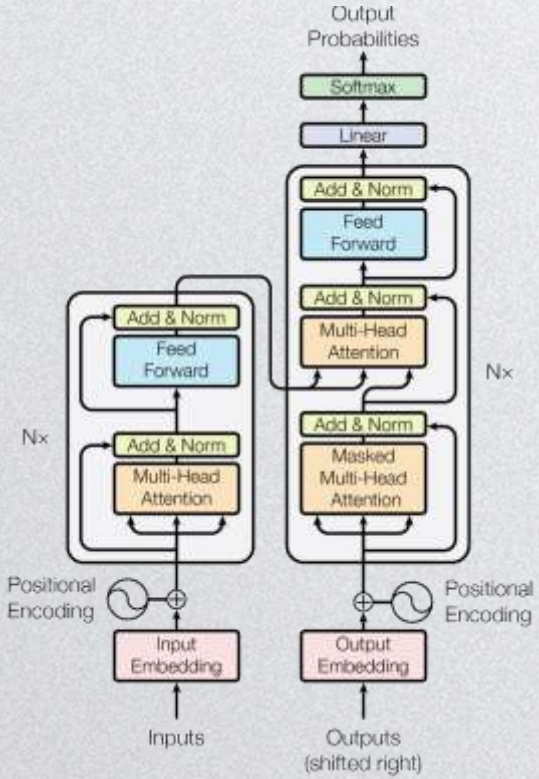
- 迁移学习是一种改善特征分布差异的有效方法。迁移学习对特征差异的改善效果在其他领域得到了充分的验证。
- 大多数迁移学习方法只涉及一个源域和一个目标域。然而，在实际应用中，常常存在多个可用的源域，这就需要对多个源域进行适应。多源域适应是一种迁移学习方法，旨在通过最小化多个源数据集和目标数据集之间的数据特征分布差异，从多个源学习预测模型。
- 本课题主要研究多源域适应在跨项目软件预测中的应用，通过适当的多源域适应方法达到从特征层面优化模型效果的目的。

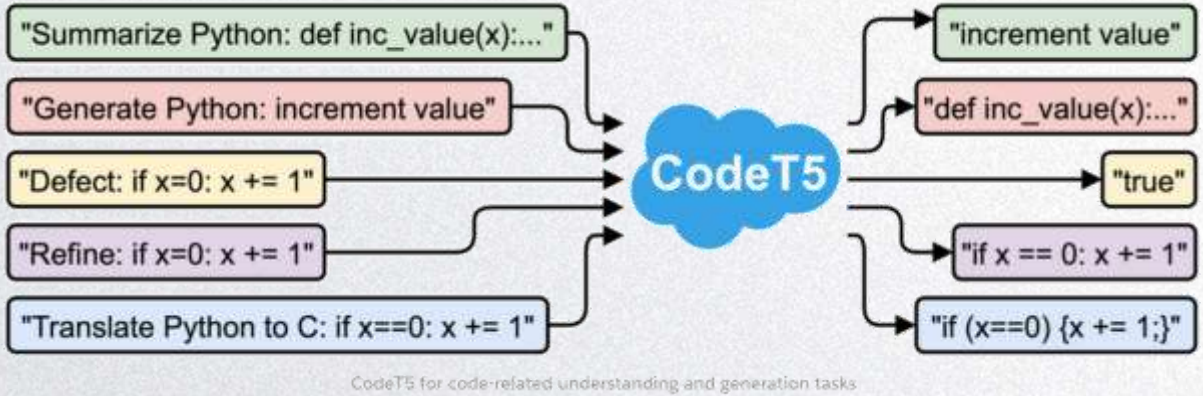
基于对抗训练和多源注意力的跨项目缺陷预测模型



基于多源域适应的跨项目缺陷预测

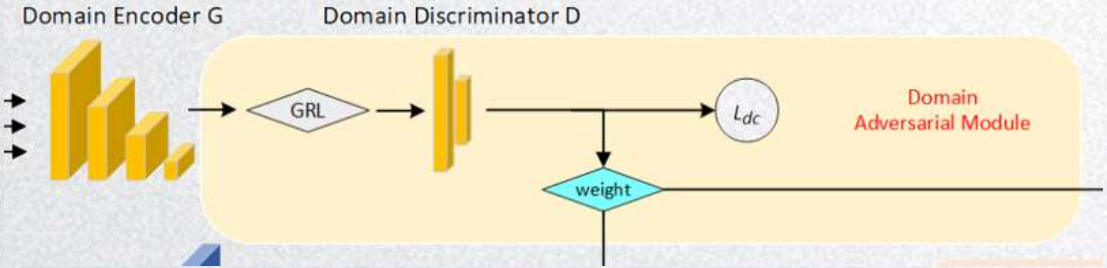
基于CodeT5的特征提取方法





CodeT5 for code-related understanding and generation tasks

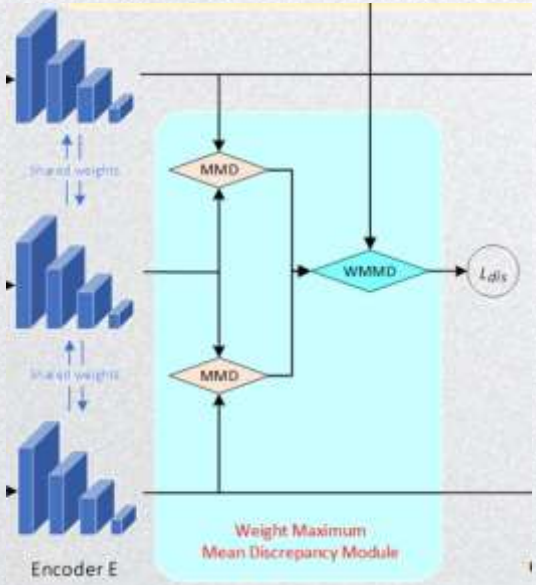
基于对抗学习的域相关性获取方法



$$\mathcal{L}_{dc} = -\lambda \mathcal{L}_d = \lambda \frac{1}{n} \sum_{i=1}^{n} d_i^{\top} \ln D\left(\mathrm{G}\left(X_i\right)\right)$$

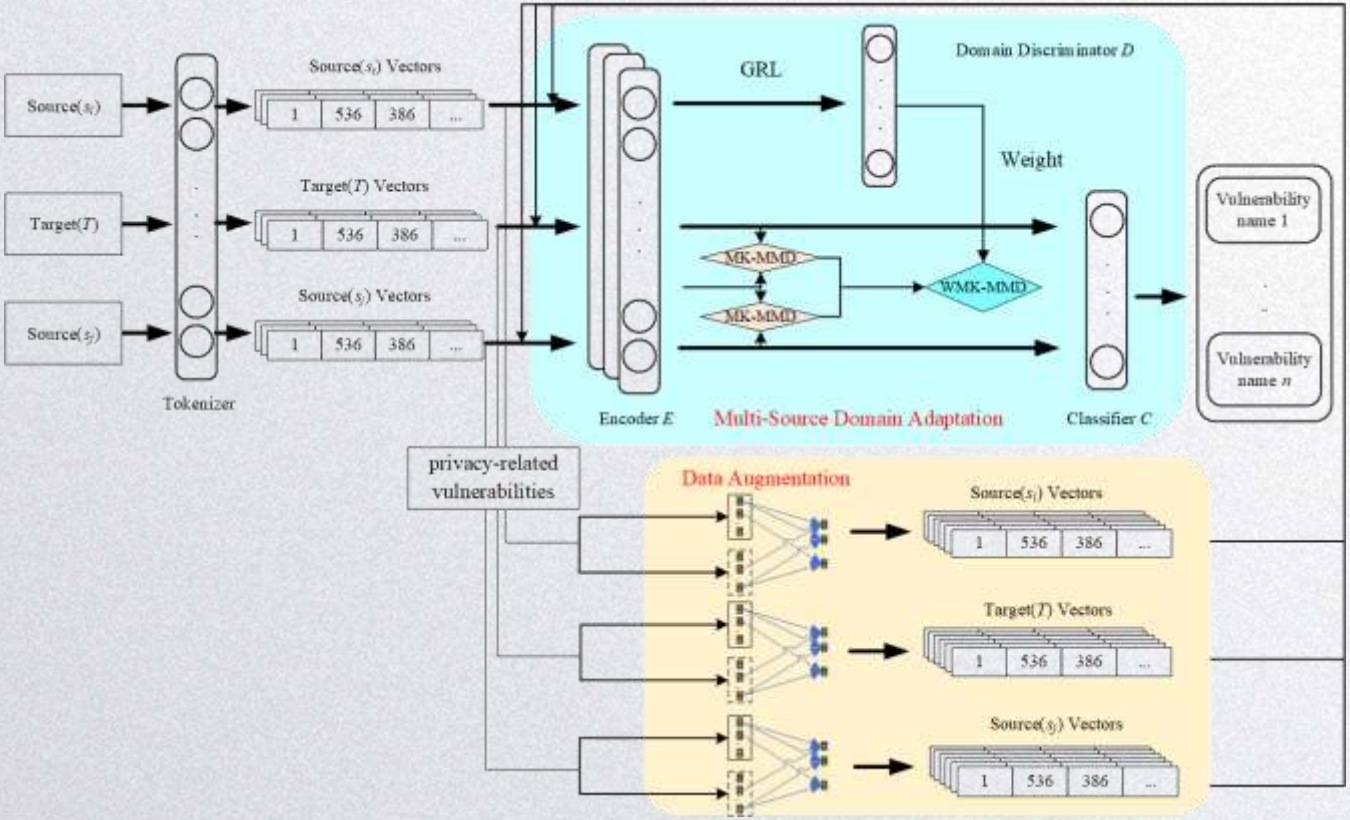
基于多源域适应的跨项目缺陷预测

基于加权最大平均差异的缩小特征分布差异方法

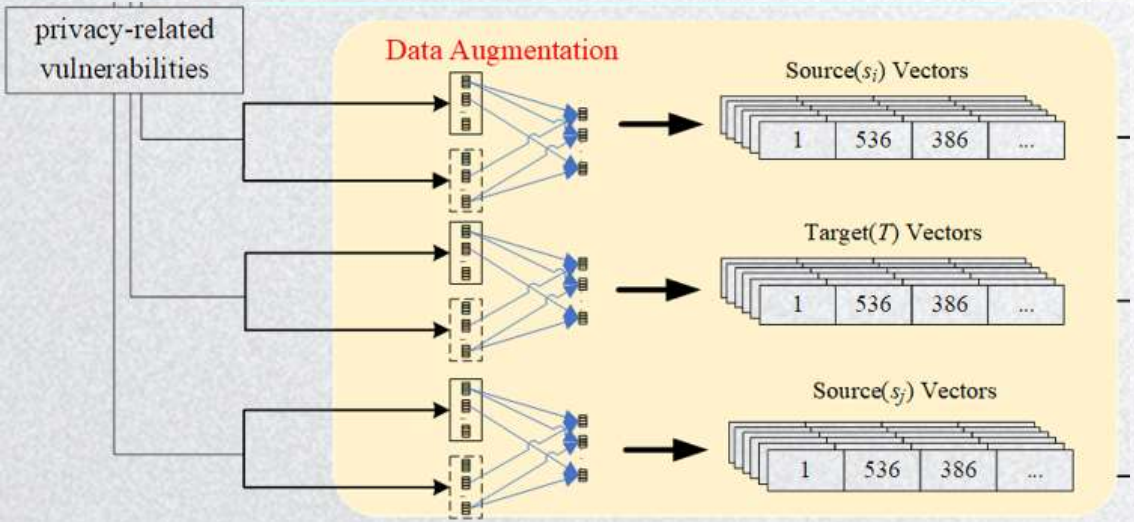


$$\sum_{j=1}^{M} w_j \left(\frac{1}{n_{s_j}^2} \sum_{i=1}^{n_{s_j}} \sum_{i'=1}^{n_{s_j}} k\left(X_i^{s_j}, X_{i'}^{s_j}\right) - \frac{2}{n_{s_j} n_t} \sum_{i=1}^{n_{s_j}} \sum_{h=1}^{n_t} k\left(X_i^{s_j}, X_h^t\right) + \frac{1}{n_t^2} \sum_{h=1}^{n_t} \sum_{h'=1}^{n_t} k\left(X_h^t, X_{h'}^t\right) \right) \tag{7}$$

基于增强多源域适应的跨项目缺陷预测模型



基于增强多源域适应的跨项目缺陷预测模型



针对隐私相关的缺陷，在缺陷名称预测的预处理阶段应用数据增强，通过扩大特定缺陷类型的样本量来增强现有数据集。

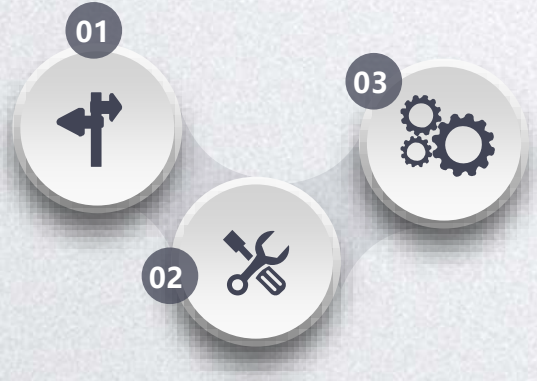
5 基于深度学习的智能漏洞信息生态开发

深度学习模型给漏洞挖掘带来的机遇

深度学习是一种基于人工神经网络的机器学习方法，其具有强大的表达能力和自动特征提取能力，在漏洞挖掘领域具有广阔的应用前景。

更准确

深度学习模型可以从海量的输入数据中自动学习出特征和模式，无需依赖人工定义的特征。模型能够更好地捕捉到隐藏在数据中的细微特征和规律，提高漏洞预测的准确性和可靠性。



更高效

深度学习模型可以并行处理大规模数据，并通过GPU等硬件加速技术快速进行训练和预测，大大提升了挖掘效率。此外，深度学习模型还能够利用分布式计算和云计算等技术，进一步加速漏洞挖掘的过程。

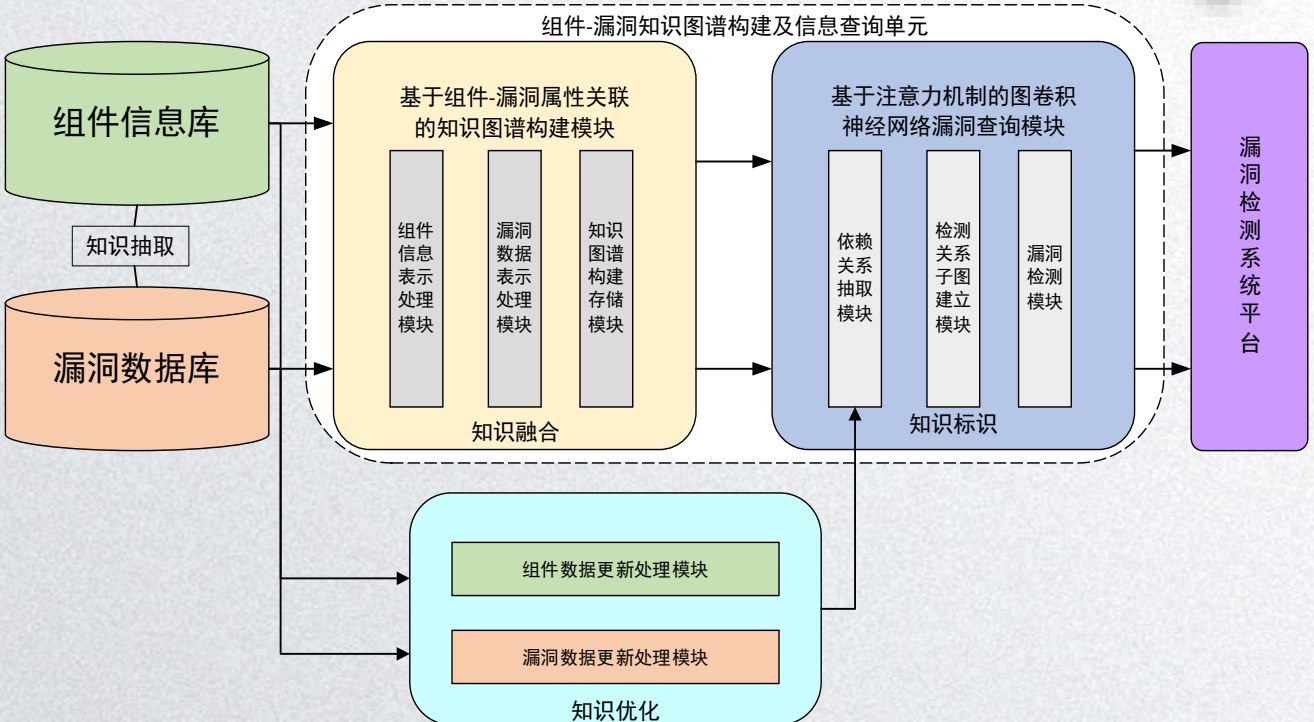
可迁移性

深度学习模型具有较强的可迁移性，即在一个领域或任务上训练好的模型可以迁移到另一个相关的领域或任务上，并且能够保持一定的泛化能力。因为漏洞的产生和利用与不同的软件系统和环境有关，而深度学习模型可以通过迁移学习的方式，利用已有的模型知识和预训练模型，在新环境中进行快速的漏洞挖掘。

这部分所需要解决的主要问题是如何应对种类繁多，且可能层出不穷的各种漏洞知识信息以及新的组件与漏洞信息的连接关系。因此如何设计一个框架以满足开源组件来满足知识图谱构建的自动性和可扩展性需求是本研究点所探讨的主要问题。基于开发的漏洞信息知识图谱，可以实现组件信息、漏洞信息进行查询功能，实现子图检索、路径查找等功能。

知识表示：对于不同的数据源，即开源组件本身与代码漏洞分别进行解析与整理，抽取出实体与关系，形成特定于该数据源的子知识图；

知识融合：在这些相互独立的子知识谱之间建立广泛关联、跨数据源的关系，并从中进一步提炼出更多的知识，从而形成最终的开源组件知识图谱。



知识图谱信息获取和更新策略

对各数据库更新信息的采集主要通过 RSS 订阅方式或下载更新列表的方式进行。RSS 是建立在 XML 基础上的一种信息传输方式，目前广泛使用的是 RSS 2.0，其中的内容都以 XML 文档结构模式有序的组织在一起，可以方便地提取其中有用的信息。而 CVE 提供了固定的更新发布地址，通过分析页面中的信息，提取需要的更新信息。

知识图谱生成和存储

本项目使用Neo4j图数据库生成并存储知识图谱。Neo4j支持导入CSV文件生成节点和关系的功能实现。

```
// write a bubble sort function
function bubble_sort(arr) {
    for(var i=0;i<arr.length;i++) {
        for(var j=1;j<arr.length-i;j++) {
            if(arr[j]>arr[j-1]) {
                // swap the values
                var temp = arr[j];
                arr[j] = arr[j-1];
                arr[j-1] = temp;
            }
        }
    }
    return arr;
}
```

漏洞检测模块

对于每个节点我们需要将不同类型的节点映射到同一纬度的特征空间下，使用维度相同的向量进行表示。利用图神经网络模型，对组件节点进行表示学习，学习到每个组件节点的低维向量表示。通过将节点特征与邻居节点的特征进行聚合，图神经网络能够将节点的上下文信息融合到节点表示中。

节点分类

使用图神经网络模型能够有效地处理图结构数据，对节点进行表示学习，捕捉图中节点之间的依赖关系。之后使用已标记的组件节点作为训练样本，进行节点分类任务，即判断组件节点是否存在漏洞。根据已有的标记信息来训练图神经网络模型，通过最终的节点表示进行分类预测。

THANK YOU FOR YOUR LISTENING.

谢谢聆听