用于安全漏洞报告识别的少样本学习

Muhammad Laiq

传播、质量管理与信息系统系，中瑞典大学，厄斯特松德校区，瑞典
muhammad.laiq@miun.se

arXiv:2601.02971v1 [cs.SE] 6 Jan 2026

摘要。 为了缩短软件系统的漏洞暴露窗口，必须及时识别安全漏洞报告。用于分类缺陷报告并识别其中安全漏洞报告的传统机器学习（ML）技术，高度依赖大量标注数据。然而，实践中安全漏洞报告的数据集往往十分稀缺，导致模型性能不佳，且在真实场景中的适用性有限。本研究提出一种基于少样本学习的技术，利用有限的标注数据有效识别安全漏洞报告。我们采用先进的少样本学习框架 SetFit，该框架结合了句子变换器、对比学习和参数高效微调。模型使用规模较小的缺陷报告标注数据集进行训练，并评估其将这些报告分类为与安全相关或与安全无关的能力。我们的方法在最佳情况下取得了 0.865 的曲线下面积，在所有受评估的数据集上均优于传统机器学习技术（基线）。这凸显了 SetFit 在有效识别安全漏洞报告方面的潜力。基于 SetFit 的少样本学习为使用传统机器学习技术识别安全漏洞报告提供了一种很有前景的替代方案。该方法能够以最少的标注工作量高效开发模型，因此特别适用于标注数据稀缺的场景。

关键词： 少样本学习 · SetFit · 问题分类 · 缺陷报告分类 · 安全漏洞报告。

1 引言

软件项目会通过 JIRA[1]，、Bugzilla[2]，和 GitHub Issues[3] 等漏洞跟踪系统收到大量缺陷报告。这些报告类型各异，涵盖功能增强请求、可用性问题、性能问题和安全漏洞。在各种缺陷报告中，安全相关缺陷报告尤为关键，因为它们可能使系统遭受利用。未能及时识别并处理此类漏洞，

[1] https://www.atlassian.com/software/jira/templates/bug-tracking [2] https://www.bugzilla.org/ [3] https://github.com/features/issues

可能会造成严重后果。手动审查缺陷报告以识别安全相关缺陷报告既耗时又容易出错，对于可能收到数千份缺陷报告的大型项目而言，也无法扩展 [25]。

文献中已提出若干自动化技术（例如， [20,28, 8, 22]），以协助识别安全漏洞报告，尤其是基于传统机器学习（ML）技术的方法。这些技术利用历史缺陷报告训练监督式机器学习模型，以预测新报告是否与安全问题相关。传统机器学习技术存在显著局限。具体而言，这些方法需要大量标注训练数据，而在实践中，安全漏洞报告通常很少有此类数据 [19, 7]。此外，安全报告与非安全报告之间的类别不平衡也会进一步影响模型性能，往往导致预测结果有偏差或不可靠。

为克服安全漏洞报告领域的数据稀缺问题，近期研究[7] 探索了使用大语言模型，尤其是生成式预训练 Transformer（GPT）模型对缺陷报告进行分类，以识别安全漏洞报告。尽管基于GPT的模型在许多自然语言处理/理解任务中表现出良好前景，但其识别安全漏洞报告的性能低于传统机器学习技术[7]。

我们在这些工作的基础上，探索一种旨在通过少样本学习应对标注数据有限这一挑战的方法。具体而言，我们提出使用 SetFit（句子变换器微调） [12] 来识别安全漏洞报告。SetFit 是最先进的少样本学习框架，它将句子变换器与对比学习及参数高效微调相结合。研究表明，在标注数据有限的场景中，SetFit 在类似任务上 [4]表现出良好的效果。

本文通过研究将一种少样本学习方法 SetFit 应用于安全漏洞报告识别任务（即对缺陷报告进行分类），作出了贡献。我们在四个广泛使用的安全漏洞报告数据集（即 Ambari、Camel、Wickey 和 Derby）上，对 SetFit 的有效性进行了实证评估。研究结果表明，SetFit 的曲线下面积最高可达 0.865，在所有评估数据集上均优于传统机器学习技术（基线）[7]。

本文其余部分安排如下。第 2 节介绍本研究的方法。第 3 节介绍研究结果。第 4 节讨论研究发现，并描述识别安全漏洞报告的相关工作。第 5 节介绍本研究的效度威胁，第 6 节总结全文。

2 研究方法

本研究旨在评估小样本学习方法（即 SetFit）在识别安全漏洞报告方面的有效性。为实现这一目标，我们提出了以下总体研究问题：***SetFit*** 在识别

安全漏洞报告方面的表现如何？为回答这一研究问题，我们开展了对比实验 [1, 24]。

在以下各小节中，我们将介绍实验设计中的主要决策，包括自动化技术的选择、数据集的选择、评估方法的选择，以及性能评估指标的选择。

2.1 技术的选择

作为基线，我们选择了三种传统的监督式机器学习技术，这些技术曾在先前的安全漏洞报告识别研究中用作基线 [27]：支持向量机、逻辑回归和随机森林。此外，我们引入 SetFit [12]，将其作为识别安全漏洞报告的新基线。我们将把所提出的技术（SetFit）的性能与这三种传统机器学习技术进行比较。

2.2 数据集的选择

为了评估所提出的 SetFit 和传统机器学习技术，我们选取了常用于安全漏洞报告识别的数据集：Ambari、Camel、Derby 和 Wicket。这些数据集已被多项先前研究用作基准 [2, 26, 25],，其中包括 Francca 等人的近期研究 [7]。

表 1 概述了本研究使用的数据集。这四个数据集均来自开源项目，最初由 Wu 等人编译并提供 [25]。具体而言，Ambari[4] 是 Hadoop 管理平台，Camel[5] 是集成框架，Derby[6] 是关系数据库管理系统，而 Wicket[7] 是 Java Web 应用框架。

表 1：所选数据集中安全漏洞报告的数量及缺陷报告（BR）的总数。

项目	安全漏洞报告数	缺陷报告总数
Camel	58（10%）	580
Ambari	48（5.5%）	871
Derby	157（21.5%）	731
Wicket	43（6.5%）	663

[4] https://ambari.apache.org/ [5] http://camel.apache.org/ [6] https://db.apache.org/derby/ [7] https://wicket.apache.org/

2.3 评估方法

评估机器学习技术性能的方法有多种，包括交叉验证、在固定数据集上测试，以及基于时间或增量划分。本研究对传统机器学习技术和 SetFit 均采用五折交叉验证方法。与在固定数据集上测试相比，这种策略有助于减少潜在偏差，并提高结果的可靠性 [23]。为进行五折交叉验证，我们将数据集划分为五个大小相等的折。每轮迭代中，我们使用其中四折训练模型，并在剩余的一折上测试。该过程重复五次，使每轮迭代都能使用不同的测试集。最后，我们对每种技术在所有五折上的结果取平均值。

2.4 性能评估指标

本研究使用的四个数据集（见表 1）均存在数据不平衡。因此，为评估所选技术在不平衡数据上的性能，我们采用了被认为对数据不平衡具有稳健性的指标，即受试者工作特征（ROC）的曲线下面积（AUC） [10]和马修斯相关系数（MCC） [18]。AUC 值通过绘制 ROC曲线确定，该曲线使用真正率（TPR）和假正率（FPR）。

$$TPR = \frac{TP}{TP + FN} \tag{1}$$

$$FPR = \frac{FP}{FP + TN} \tag{2}$$

马修斯相关系数（MCC）衡量由混淆矩阵得出的预测值与实际值之间的相关性，也可根据混淆矩阵计算得出。MCC 的取值范围为 -1 到 1，其中 1 表示分类完全正确，-1 表示分类完全错误。MCC 为 0 意味着模型的表现不优于随机猜测。除了底层指标曲线下面积（AUC）和马修斯相关系数（MCC）之外，我们还测量精确率、召回率和 F分数。

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$F - score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{6}$$

2.5 数据预处理

与先前的工作 [7]，类似，本研究使用缺陷报告的描述来训练模型，以识别安全缺陷报告。我们对缺陷报告的描述采用标准的数据预处理方法，包括删除特殊字符、数字和链接。

对于传统机器学习技术，我们使用词频-逆文档频率（TF-IDF）将文本数据转换为稀疏矩阵 [6]。对于 SetFit 模型，我们使用了 allmpnet-base-v2，这是嵌入领域表现最佳的预训练模型之一 [4]。

2.6 实现细节

我们使用 sklearn 库 [5]来应用选定的传统机器学习技术（支持向量机、逻辑回归和随机森林）。我们通过使用 sklearn 库进行网格搜索，确定这些技术的最佳参数。首先，我们使用网格搜索调整所选每种传统机器学习技术的参数，然后使用最佳参数构建模型。

对于 SetFit，我们遵循 Hugging Face 的官方文档[8]，使用 sentence-transformers/all-mpnet-base-v2 创建模型。

为了使用经典机器学习技术和 SetFit 进行实验，我们使用了一台配备 Windows 11 64 位操作系统、Intel (R) Core (TM) Ultra 9（16 核）、64 GB RAM 和 NVIDIA GeForce RTX 4060 笔记本电脑 GPU（8 GB）的机器。

3 结果与分析

为了回答提出的研究问题（即，***SetFit*** 在识别安全漏洞报告方面表现如何？），我们评估了所选基线方法（支持向量机、逻辑回归和随机森林）以及我们提出的方法（即 SetFit）的性能。我们使用四个安全漏洞报告数据集（即 Camel、Ambari、Derby 和 Wicket，见表 1）评估了这些方法。下文将介绍这些方法的评估结果。

表 2 和图 1 展示了本研究的结果。在所有数据集中，SetFit 在几乎所有指标上始终优于基线方法。

Camel 项目： 在 Camel 数据集中，SetFit 取得了最高分，曲线下面积为 0.807，马修斯相关系数为 0.733。这显著优于最强的基线模型逻辑回归，其曲线下面积为 0.772，马修斯相关系数为 0.688。此外，SetFit 取得了最高的 F分数（0.737）和召回率（0.620），表明其能够正确识别安全相关缺陷报告。与基线模型不同，这些结果表明 SetFit 能够有效平衡精确率和召回率。

[8] https://huggingface.co/docs/setfit/index

表 2：逻辑回归（LR）、支持向量机（SVM）、随机森林（RF）和 SetFit 在识别安全漏洞报告方面的表现。曲线下面积（AUC）和马修斯相关系数（MCC）。

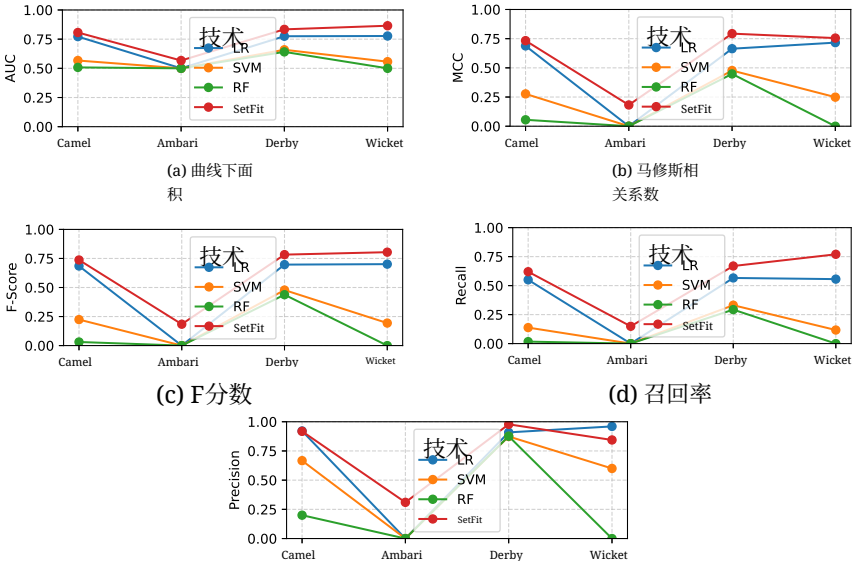
数据集	技术	AUC	MCC	F分数	精确率	召回率
Camel	逻辑回归（基线）	0.772	0.688	0.685	**0.921**	0.550
	支持向量机（基线）	0.567	0.277	0.224	0.667	0.138
	随机森林（基线）	0.508	0.055	0.031	0.200	0.017
	SetFit（我们的）	**0.807**	**0.733**	**0.737**	0.918	**0.620**
Ambari	逻辑回归（基线）	0.500	0.000	0.000	0.000	0.000
	支持向量机（基线）	0.500	0.000	0.000	0.000	0.000
	随机森林（基线）	0.500	0.000	0.000	0.000	0.000
	SetFit（我们的）	**0.567**	**0.182**	**0.184**	**0.310**	**0.149**
Derby	逻辑回归（基线）	0.775	0.664	0.697	0.909	0.566
	支持向量机（基线）	0.659	0.477	0.478	0.874	0.331
	随机森林（基线）	0.641	0.448	0.438	0.876	0.293
	SetFit（我们的）	**0.834**	**0.793**	**0.783**	**0.978**	**0.669**
Wicket	逻辑回归（基线）	0.777	0.716	0.701	**0.960**	0.556
	支持向量机（基线）	0.557	0.249	0.194	0.600	0.117
	随机森林（基线）	0.500	0.000	0.000	0.000	0.000
	SetFit（我们的方法）	**0.865**	**0.755**	**0.804**	0.844	**0.770**

Ambari 项目： Ambari 数据集呈现了一个尤其具有挑战性的场景：所有基线模型均完全失效，曲线下面积和马修斯相关系数分别为 0.500 和 0.000。SetFit 虽然表现并不高（曲线下面积为 $= 0.567$，马修斯相关系数为 $= 0.182$），却是唯一能够从数据中识别出某种模式的技术。这些 Ambari 数据集的结果与先前关于识别安全漏洞报告的研究 [7]一致。

Derby 项目： 在 Derby 数据集中，SetFit 在所有指标上均优于基线。其曲线下面积得分最高，达到 0.834，马修斯相关系数最高为 0.793。此外，SetFit 的召回率显著提升，得分为 0.669，高于逻辑回归的 0.566。其精确率也保持在较高水平，为 0.978。这些改进使其 F分数显著提高至 0.783，表明 SetFit 的整体分类性能更优。

Wicket 项目： Wicket 项目的结果进一步表明，SetFit 在识别安全漏洞报告方面表现出色。其曲线下面积（0.865）、马修斯相关系数（0.755）和 F分数（0.804）均为最佳。值得注意的是，SetFit 的召回率超过了表现强劲的基线逻辑回归（0.770 对 0.556），同时保持了较高的精确率（0.844）。这种平衡很有用，尤其是在漏掉安全漏洞可能造成严重代价的情况下。

总之，评估结果（见表 2 和图 1）表明，与基线相比，SetFit 即使在可用标注数据极少的情况下，也能有效识别安全漏洞报告。它在各种数据集上表现稳定，这表明基于 SetFit 的少样本学习方法很有望用于这项任务。传统基线往往难以应对，尤



(a) 曲线下面积

(b) 马修斯相关系数

(c) F分数

(d) 召回率

(e) 精确率

图 1：所评估技术（逻辑回归 (LR)、支持向量机 (SVM)、随机森林 (RF) 和 SetFit）在识别安全漏洞报告方面的曲线下面积、马修斯相关系数、F分数、召回率和精确率。

其是在数据有限的场景下，而 SetFit 仅凭少数几个示例便能实现泛化，这一能力尤为有利。

4 讨论

本节讨论我们的研究发现，以及有关识别安全漏洞报告的相关工作。

缺陷报告可依据多种标准进行分类，包括其有效性、优先级，或其是否涉及安全问题或非安全问题。因此，许多研究探讨了缺陷报告分类的不同方面 [17]。例如，既有研究探讨了基于报告有效性的分类 [16, 15]，也研究了不同问题类型的识别，例如功能请求、用户问题和文档相关任务 [13, 11]。此外，其他研究还探讨了对缺陷报告进行分组或聚类，以发现共同的根本原因 [14, 21]。与之不同的是，本研究聚焦于对缺陷报告进行分类，以区分与安全相关的报告和非安全相关的报告。

已有多项研究探讨了使用自动化技术对缺陷报告进行分类，以识别安全漏洞报告，例如 [20, 28, 8, 22]。在评估的自动化技术中，传统机器学习技术占据主导地位，包括逻辑回归、

支持向量机、随机森林、K近邻和朴素贝叶斯。近期，Francca 等人 [7] 探讨了使用大语言模型（LLMs），特别是基于GPT的模型，对缺陷报告进行分类以识别安全漏洞报告。尽管这些模型日益流行且具备通用能力，他们的研究[7] 发现，在这一场景下，基于GPT的模型表现不如传统机器学习技术。

在缺陷报告分类的相关研究中，报告会被划分为缺陷、问题、功能请求和文档。Colavito等人采用了基于 SetFit 的小样本学习方法。他们使用 200 份经过标注的缺陷报告训练模型，并在另外 200 份报告上进行评估，取得了 0.832 的 F1-micro分数。研究结果表明，即使训练数据有限，SetFit 仍能取得有竞争力的表现，从而解决了大型标注数据集难以获取的问题；而传统机器学习技术通常需要这类数据集。受其研究启发，我们在本研究中采用 SetFit 来识别安全相关缺陷报告。

通过比较传统机器学习基线（逻辑回归、支持向量机和随机森林）与我们提出的方法（即 SetFit），我们证明，SetFit 在所评估的数据集上始终优于这些技术。详情请参见表 2 和图 1。

我们的研究结果表明，SetFit 为识别安全相关缺陷报告提供了一种前景良好的解决方案。与通常需要大量标注数据集才能取得良好效果的传统机器学习技术不同，SetFit 即使仅使用数量有限的标注示例进行训练，也能实现良好性能。这一点对于识别安全漏洞报告尤其有价值，因为由于需要领域专业知识，相关标注数据既稀缺又难以获取。

5 有效性威胁

结果的可推广性： 在本研究中，我们使用了多个数据集（即 Camel、Ambari、Derby 和 Wicket，详见第 2.2 节），以提高研究结果的可推广性。尽管这些数据集来自不同的项目，但它们可能无法完全代表所有软件开发环境，尤其是专有项目。

评估方法的有效性： 可以采用不同的评估方法（例如交叉验证和在固定数据集上进行测试）来评估机器学习技术的性能。为尽量减少评估过程中的潜在偏差并提高结果的可靠性，我们采用了五折交叉验证方法（详见第 2.3 节）。该策略有助于减少实验偏差，并增强结果的可靠性 [23]。

评估指标的有效性：本研究使用的数据集（即 Camel、Ambari、Derby、Wicket，详见第 2.2 节）存在数据不平衡。为减少解读结果时的潜在偏差并提高结果的

可靠性，我们使用了被认为对数据不平衡具有稳健性的评估指标，即曲线下面积 [10] 和马修斯相关系数 [18]。

训练数据的可靠性：用于训练这些技术的数据可能会带来效度威胁。例如，缺陷报告可能被错误分类。为尽量降低这一风险，我们使用了四个广泛使用且经过人工验证的数据集 [25] 来评估所提出的安全漏洞报告识别技术。

6 结论与未来工作

本文研究了使用 SetFit（一种先进的少样本学习方法）识别安全漏洞报告。我们的方法表现最佳时，曲线下面积达到 0.865，在所有评估数据集上的表现均优于传统机器学习基线（逻辑回归、支持向量机和随机森林）。研究结果强调，利用预训练语言表示和对比学习有助于提升安全漏洞报告的识别效果，也展示了 SetFit 在此任务中的有效性。

总之，基于 SetFit 的少样本学习为识别安全漏洞报告提供了一种有前景的传统机器学习技术替代方案。这种方法只需极少的标注工作量，即可高效开发模型，因此特别适用于标注数据有限的情形。

未来的工作将重点使用更广泛的项目来评估所提出的技术，并纳入来自开源和闭源环境的数据。此外，我们还将研究使用 SetFit 识别安全漏洞报告时模型预测的可解释性。

利益披露。 作者声明，他们不存在已知的、可能影响本文所报告工作的竞争性财务利益或个人关系。

参考文献

1. Alpaydin, E.：《机器学习导论》。麻省理工学院出版社（2020） 2. Alqahtani, S.S.：使用 fastText 对安全漏洞报告进行分类。《国际信息安全期刊》**23**(2)，1347–1358（2024） 3. Antoniol, G., Ayari, K., Di Penta, M., Khomh, F., Guéhéneuc, Y.G.：这是缺陷还是功能增强？一种基于文本的变更请求分类方法。载于：协作研究高级研究中心会议：思想交流会议。第304–318页（2008） 4. Colavito, G., Lanubile, F., Novielli, N.：少样本学习在问题报告分类中的应用。载于：自然语言软件工程国际研讨会。第16–19页。IEEE（2023） 5. scikit-learn 开发者：scikit-learn：Python 中的机器学习。https://scikit-learn.org/stable/index.html，在线；访问于2025年10月6日。scikit-learn 开发者：TfidfVectorizer。
https://scikit-learn.org/stable/index.html，在线；访问于2025年6月10日

7. França, H.L., Goseva-Popstojanova, K., Teixeira, C., Laranjeiro, N.：GPT并非万能灵药：使用GPT进行安全漏洞报告识别的表现与挑战。《信息与软件技术》第107778页（2025） 8. Goseva-Popstojanova, K., Tyo, J.：利用监督和无监督分类进行文本挖掘，以识别安全相关缺陷报告。载于：国际软件质量、可靠性与安全会议。第344–355页。IEEE（2018） 9. Herzig, K., Just, S., Zeller, A.：这不是缺陷，而是功能：错误分类如何影响缺陷预测。载于：国际软件工程会议。第392–401页。IEEE（2013） 10. Huang, J., Ling, C.X.：使用曲线下面积和准确率评估学习算法。《IEEE知识与数据工程汇刊》**17**(3)，299–310（2005） 11. Kallis, R., Di Sorbo, A., Canfora, G., Panichella, S.：工单标注器：机器学习驱动的问题分类。载于：2019年IEEE国际软件维护与演化会议（ICSME）。第406–409页。IEEE（2019） 12. Koch, G., Zemel, R., Salakhutdinov, R., 等：用于单样本图像识别的孪生神经网络。载于：国际机器学习会议深度学习研讨会。第2卷。里尔（2015） 13. Laiq, M.：用于问题报告分类的智能工具。载于：2023年IEEE/ACM第二届自然语言软件工程国际研讨会（NLBSE）。第13–15页。IEEE（2023） 14. Laiq, M., bin Ali, N., Börstler, J., Engström, E.：理解无效缺陷报告的数据驱动方法：一项工业案例研究。《信息与软件技术》**164**，107305（2023） 15. Laiq, M., bin Ali, N., Börstler, J., Engström, E.：缺陷报告分类机器学习技术的比较分析。《系统与软件杂志》第112457页（2025） 16. Laiq, M., Ali, N.b., Böstler, J., Engström, E.：在工业环境中尽早识别无效缺陷报告——一项案例研究。载于：国际以产品为中心的软件过程改进会议。第497–507页。施普林格（2022） 17. Laiq, M., Dobslaw, F.：问题报告分类的自动化技术：系统映射研究。arXiv预印本 arXiv:2505.01469（2025） 18. Matthews, B.W.：T4噬菌体溶菌酶预测与观测到的二级结构比较。《生物化学与生物物理学报》（BBA）—蛋白质结构**405**(2)，442–451（1975） 19. Ohira, M., Kashiwa, Y., Yamatani, Y., Yoshiyuki, H., Maeda, Y., Limsettho, N., Fujino, K., Hata, H., Ihara, A., Matsumoto, K.：高影响缺陷数据集：人工分类的问题报告。载于：软件仓库挖掘工作会议。第518–521页。IEEE（2015） 20. Peters, F., Tun, T.T., Yu, Y., Nuseibeh, B.：面向安全漏洞报告预测的文本过滤与排序。《IEEE软件工程汇刊》**45**(6)，615–631（2017） 21. Rahman, M.M., Khomh, F., Castelluccio, M.：为什么有些缺陷无法复现？——基于数据融合的实证调查。载于：2020年IEEE国际软件维护与演化会议（ICSME）。第605–616页。IEEE（2020） 22. Shu, R., Xia, T., Williams, L., Menzies, T.：通过超参数优化改进安全漏洞报告分类。arXiv预印本 arXiv:1905.06872（2019） 23. Witten, I.H., Frank, E.：数据挖掘：使用Java实现的实用机器学习工具与技术。《ACM SIGMOD记录》**31**(1)，76–77（2002） 24. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A., 等：软件工程实验研究，第236卷。施普林格（2012）

25. Wu, X., Zheng, W., Xia, X., Lo, D.：数据质量至关重要：关于安全漏洞报告预测中数据标签正确性的案例研究。IEEE软件工程汇刊 **48(7)**，2541–2556（2021） 26. Yokoyama, D., Nishiura, K., Monden, A.：识别问题报告中的安全漏洞：BERT、n-gram IDF 与 ChatGPT 的比较。载于：国际软件工程研究、管理与应用会议。第328–333页。IEEE（2024） 27. Zheng, W., Xun, Y., Wu, X., Deng, Z., Chen, X., Sui, Y.：安全漏洞报告分类的类别再平衡方法比较研究。IEEE可靠性汇刊 **70(4)**，1658–1670（2021） 28. Zhou, Y., Sharma, A.：从提交消息和缺陷报告中自动识别安全问题。载于：软件工程基础联合会议。第914–919页（2017）