基于互补视图划分与融合的软件缺陷多视图深度学习方法

姚永芳[1] 丁永昌[2] 陈林君[2] 荆晓远[1,2*]
YAO Yongfang DING Yongchang CHEN Linjun JING Xiaoyuan

摘 要 软件缺陷预测是软件工程领域的研究热点，其核心目标是在软件发布前，利用已有度量信息提前识别高风险缺陷模块。然而，不同项目的特征类型存在显著差异，现有多视图深度学习方法虽能联合利用多类度量特征，但多数在融合阶段采用固定权重，默认所有样本对各视图的依赖程度一致，与实际场景中样本对不同视图（如改动规模、演化频率等）的依赖异质性相悖，且在视图质量波动时难以保证性能稳定性。针对上述问题，文章提出一种基于互补视图划分与融合的软件缺陷多视图深度学习方法。首先将过程度量划分为两个语义互补的子视图，通过视图级贡献分配机制为每个样本动态调整融合权重；同时引入与贡献分配相协调的判别性约束，提升特征表示的聚合性与区分度。 因未引入额外采样步骤，性能提升主要源于表示学习与结构设计的优化。在 AEEEM 数据集上的实验表明，所提方法在 *F*-measure 和 AUC 指标上整体优于多种对照方法，且在类别不平衡较明显的项目中提升更为显著，验证了其在稳定性和适用性上的优势。

关键词 软件缺陷预测；多视图深度表示学习；深度学习

doi：10.3969/j.issn.1672-9528.2026.04.005

0 引言

软件缺陷预测（software defect prediction, SDP）[1-2] 是软件质量保证的重要环节，其目标是在软件发布前利用历史数据构建预测模型，提前识别潜在缺陷，从而降低维护成本、提升软件可靠性。大量研究表明，软件系统的维护与测试阶段往往消耗超过 50% 的开发成本，而其中缺陷修复又占据了较大比例。因此，如何在开发早期通过有效的预测手段识别潜在缺陷模块，已成为提升软件质量和降低成本的关键问题。

随着软件工程的发展，研究者逐渐认识到，不同类型的软件度量数据能够从不同角度刻画软件质量特征。例如，产品度量（如代码规模、复杂度）[3] 可揭示结构性风险，过程度量（如提交频率、修改历史）[4] 能够反映开发行为与演化模式，语义度量（如静态分析与代码嵌入特征）则可补充上下文信息。这些异质特征为缺陷预测提供了丰富的信息来源。然而，在实际应用中仍存在两个核心挑战：

（1）度量数据往往稀缺或不完整，导致预测模型难以充分训练；

（2）不同视角的数据质量参差不齐，简单融合可能掩盖高价值特征，甚至引入噪声，从而降低预测性能。

为缓解数据不足的问题，近年来深度多视图学习 [5] 逐渐应用于软件缺陷预测。该类方法能够从多源数据中提取互补信息，从而提升模型的判别力。例如，MTMV 方法 [6] 利用多源静态分析结果构建多视图特征，以捕捉缺陷位置分布的不确定性和语义差异；MTDP 方法 [7] 则借助神经网络模型自动学习不同粒度和维度的标签信息，通过迁移学习为缺陷预测提供更多标注数据。多视角特征融合显著改善了传统单视角模型的性能，为缺陷预测提供了新的发展方向。然而，现有多视图方法大多依赖固定融合策略，在不同样本间使用相同的权重分配方式。这一策略忽视了视图信息量的动态差异：当某一视图包含大量噪声或其区分能力较弱时，仍然可能被

1. 广东石油化工学院计算机学院和石化装备智能安全广东省重点实验室 广东茂名 525000
2. 武汉大学计算机学院 湖北武汉 430000
[基金项目] 国家自然科学基金面上项目“基于类不平衡深度特征学习的石化动设备故障信号分类研究”（62176069）；广东省自然科学基金面上项目“软件缺陷预测关键技术和新方法研究”（2023A515012653）；广东省教育厅创新团队“石化生产网络安全与大数据分析控制创新团队”（2020KCXTD014）；2019 年广东省重点学科项目“石化工业安全生产深度感知与智能监控技术研究”；国家自然科学基金面上项目“人类群体决策引导的多视图表示学习方法和石化动静装备故障诊断与剩余寿命预测应用研究”（62576115）

赋予较高权重，导致融合表示的判别性下降。因此，在视图质量不均衡或任务场景复杂的情况下，这类方法往往难以保证稳定的预测性能。

针对上述问题，本文提出基于互补视图划分与融合的软件缺陷多视图深度学习方法。该方法在多视图特征映射框架中引入视图级贡献分配机制[8]，能够根据输入样本的特征分布自适应分配各视图的贡献权重，从而在不同场景下自动突出信息量更高的视图；同时，在判别性建模阶段加入贡献分配引导的特征分离约束，在保持类内特征紧凑性的同时进一步扩大类间差异，以增强融合表示的判别能力。值得强调的是，本文方法不依赖任何采样技术，其性能提升完全来自结构优化与表示学习能力的增强，从而保证与现有方法的公平对比。本文的主要贡献如下：

（1）提出视图级贡献分配融合策略，能够针对不同样本动态分配各视图的贡献权重，实现更灵活的多视图特征融合；

（2）设计贡献分配引导的判别性约束，增强融合表示的类内紧凑性与类间分离度，从而提升预测精度与泛化能力；

（3）在不依赖采样技术的前提下，通过结构与优化机制改进实现性能提升，保证了与现有方法的公平性；

（4）在软件缺陷标准数据集 AEEEM 上开展实证研究，结果表明所提方法在多项指标上优于主流多视图深度学习方法。

1 方法

1.1 总体框架

本文方法共包含三个相互衔接的部分，依次为多视图特征构建模块、视图编码与融合模块和判别性表示优化模块，具体情况如下：

（1）多视图特征构建模块：从单源过程度量中划分出两个语义互补的子空间；

（2）视图编码与融合模块：两个子视图分别进入独立编码器，得到潜在表示后，再通过贡献分配机制完成融合；

（3）判别性表示优化模块：在分类损失之外，再加入额外约束，使得到的融合表示更利于区分类别。

这一框架并没有通过重新采样来改造原始数据分布，而是把改进重点放在表示建模本身。因此，性能变化更多来自模型如何看待特征、组织特征以及使用特征，而不是来自数据层面的重新平衡。

1.2 多视图特征构建

在只提供单源过程度量的情况下，如果直接把全部特征送入同一个模型，往往很难体现不同特征群体的作用差异。为此，本文采用特征子空间划分的方式构造多视图输入。主要包含两个视图：

视图 A 描述规模与复杂度，主要包含 churnLinesAdded、linesDeleted、maxChurn、entropy 等指标。这些特征更接近代码改动本身，能够反映改动幅度、集中程度以及复杂性变化。

视图 B 主要描述频率与协作，主要包含 hanrevisionCount、bugFixCount、authorCount、ownerCount 等指标。相较于视图 A，更强调演化过程中的活动频率、修复记录以及参与者行为。采用这样的划分方式，原因并不复杂。一是同一视图内部的特征语义更接近，能减少彼此之间的不协调；二是两个视图关注的是软件过程的不同侧面，一个偏向“改了多少、改得多复杂”，另一个偏向“改得多频繁、谁在参与”，二者存在天然互补；三是显式划分后，模型更容易分别学习不同特征群体的表示，而不是在一开始就被混杂输入牵着走。因此，即便数据只来自单一来源，也仍然可以通过结构化划分获得多视图建模的好处。

1.3 视图级贡献分配机制

为解决不同样本在不同视图上的信息量差异，本文在融合阶段引入视图级贡献分配机制。具体而言，假设编码器输出的两个视图表示分别为 h_i 和 h_j，则注意力权重计算为：

$$\alpha_i = \frac{\exp\left(f\left(h_i\right)\right)}{\sum_{j=1}^{2}\exp\left(f\left(h_j\right)\right)} \tag{1}$$

式中：$f(\cdot)$ 为由多层感知机实现的可学习打分函数。

融合表示为：

$$h = \sum_{i=1}^{2}\alpha_i \cdot h_i \tag{2}$$

该机制可针对每个样本动态调整各视图的贡献，使信息量更高的视图在预测中获得更大权重。

1.4 贡献分配引导的判别性约束

在分类监督信号之外，本文引入注意力引导的特征分离约束，以增强类间可分性。设$\mathcal{H}_y$表示标签为y的融合特征集合，每个样本包含两个视图的编码表示 h_1 和 h_2。

类内紧凑性：最小化同类样本与类中心的距离为：

$$\mathrm{mu}_y = \frac{1}{|\mathcal{H}_y|}\sum_{(h_1,h_2)\in\mathcal{H}_y}\left(\alpha_1 h_1 + \alpha_2 h_2\right) \tag{3}$$

类间分离度：最大化不同类中心之间的距离为：

$$\mathcal{C}_{\text{inter}} = \sum_{y\neq y'}\max\left(0,\, m - \left\|\mu_y - \mu_{y'}\right\|_2\right) \tag{4}$$

式中：m 为间隔参数。注意力权重 α_i 会对类中心计算过程进行加权，使判别性优化更符合样本的视图信息分布。

最终损失函数为：

$$\mathcal{C} = \mathcal{C}_{\text{cls}} + \lambda_1\mathcal{C}_{\text{intra}} + \lambda_2\mathcal{C}_{\text{inter}} \tag{5}$$

式中：$\mathcal{C}_{\text{cls}}$为交叉熵分类损失；λ_1、λ_2 为权重系数。

2 实验分析

2.1 实验样本对象

为充分验证本文方法在单源多视图条件下的有效性与鲁棒性，选取AEEEM数据集[9]中的EQ（Equinox）与LC（Lucene）两个子项目作为实验对象（如表1所示）。两者均采用过程度量（变更级特征），不包含LOC等产品度量字段。

表1 AEEEM数据集的具体情况

项目	模块	缺陷模块	缺陷率/%	不平衡率	语言	度量特征
EQ	324	129	39.8	0.7		
JDT	997	206	20.7	0.3		
LC	691	64	9.3	0.1	Java	61
ML	1 862	245	13.2	0.2		
PDE	1 497	209	14.0	0.2		

为保证结果的权威性与可比性，实验遵循以下设定：

（1）不进行任何采样处理（不过采、不过欠），保持原始类分布；

（2）项目内评测：EQ与LC分别独立建模与验证；

（3）10折交叉验证：分层抽样，固定随机种子，报告均值与标准差。

2.2 实验评价指标

本次工作采用软件缺陷检测领域的主流指标*F*-measure、AUC指标进行评价。

具体指标计算为：

F-measure（F_1分数）是精确率（Precision）和召回率（Recall）的调和平均数：

$$\text{Precision}=\frac{\text{TP}}{\text{TP}+\text{FP}},\quad \text{Recall}=\frac{\text{TP}}{\text{TP}+\text{FN}} \tag{6}$$

$$F_1=\frac{2\cdot\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}=\frac{2\text{TP}}{2\text{TP}+\text{FP}+\text{FN}} \tag{7}$$

式中：TP代表真阳性；FP代表假阳性；FN代表假阴性。

AUC（Area Under the ROC Curve）是ROC曲线下的面积，等价于随机选取一个正样本和一个负样本时，分类器把正样本排在负样本前的概率。公式为：

$$\text{TPR}=\frac{\text{TP}}{\text{TP}+\text{FN}} \tag{8}$$

$$\text{FPR}=\frac{\text{FP}}{\text{FP}+\text{TN}} \tag{9}$$

$$\text{AUC}\approx\sum_{i=1}^{n-1}(\text{FPR}_{i+1}-\text{FPR}_i)\cdot\frac{\text{TPR}_{i+1}+\text{TPR}_i}{2} \tag{10}$$

式中：TPR代表真正率；FPR代表假正率。

2.3 实验配置

2.3.1 硬件与软件环境

实验在常规科研工作站上完成，处理器为Intel Core i7系列，内存为16~32 GB。软件环境包括Python 3.10+、PyTorch 2.x、scikit-learn以及常用数值处理与可视化库。为避免偶然性干扰，实验统一设置随机种子。

2.3.2 单源多视图构建

在仅有过程度量[10]的前提下，采用语义划分的“特征子空间划分”单源多视图方案：

（1）视图A（规模/复杂度）：包含churn强度、增删行数、最大变更量、熵等，反映改动强度与复杂性；

（2）视图B（频率/协作）：包含修订次数、缺陷修复次数、作者/提交者数量、文件年龄等，刻画演化频率与团队协作。

2.3.3 训练流程与超参数

（1）预处理：对两个视图分别进行标准化（按训练折均值/方差）；不做采样与类别重加权。

（2）模型结构：双编码器（各视图独立MLP编码至相同维度d），视图级注意力软融合，分类头为两层MLP。

（3）损失函数：交叉熵$\mathcal{C}_{\text{cls}}$+判别性约束（类内紧凑$\mathcal{C}_{\text{intra}}$与类间间隔$\mathcal{C}_{\text{inter}}$）。

（4）优化设置：Adam，学习率1e-3，权重衰减1e-5，批大小64；每折最多40轮，验证早停（耐心8轮）。

（5）交叉验证：StratifiedKFold(n_splits=10, shuffle=True, random_state=42)。

2.4 实验结果与对比

本次工作选择传统模型KNN[11]、LR[12]、RF[13]、SVM[14]以及主流模型A2NN[15]、SMOTUNED[16]、DEEPER[17]、CBS+[18]作为对照实验组。

从*F*-measure看，本文方法在五个项目上都处于较优水平。例如，在EQ项目中，本文方法取得0.77，高于最优基线SVM的0.71，提升了0.06，说明在特征条件相对较好的情况下（不平衡率0.7），动态融合仍能继续挖掘有用信息。在LC项目中，本文方法达到0.58，相比LR的0.45有明显提升。相较于EQ、LC的类别不平衡更突出，这说明本文所提出的方法在复杂分布下仍保持了较好的识别能力。PDE项目中，本文方法为0.51，相比LR和SMOTUNED的0.33提升幅度更大，这一结果表明方法对于判别性约束对稀疏缺陷样本场景优势明显。JDT项目中，本文方法达到0.68，也高于多数组对照方法。ML项目中，本文方法与最佳采样基线持平，但由于没有依赖采样操作，这一结果依然具有参考价值，如表2所示。

在AUC结果也反映出类似趋势。EQ、LC、PDE和ML四个项目上，本文方法均取得最高值；JDT项目虽然没有达到最优，但差距不大。将两类指标对比，发现该方法的优势并不只体现在某一个特定项目上，而是在多个不同分布条件下都保持了较稳定的表现。尤其在LC与PDE这类缺陷样本较少、分布偏斜更明显的场景中，提升更加突出，这从侧面说明按样本动态分配视图贡献是有必要的。

进一步比较宏平均结果，本文方法的*F*-measure和AUC

表 2 每个项目的 *F*-measure、AUC 值比较结果

每个项目的 *F*-measure 值比较结果（* 为本次工作方法）									
Project	KNN[11]	LR[12]	RF[13]	SVM[14]	A2NN[15]	SMOTUNED[16]	DEEPER[17]	CBS+[18]	*
EQ	0.70	0.63	0.67	0.71	0.59	0.67	0.65	0.65	0.77
LC	0.36	0.45	0.38	0.40	0.38	0.35	0.32	0.26	0.58
PDE	0.20	0.33	0.26	0.23	0.14	0.33	0.31	0.31	0.51
JDT	0.51	0.58	0.60	0.46	0.25	0.55	0.45	0.58	0.68
ML	0.31	0.28	0.32	0.41	0.15	0.42	0.34	0.39	0.42
各项目 AUC 值的比较结果（* 为本次工作方法）									
Project	KNN	LR	RF	SVM	A2NN	SMOTUNED	DEEPER	CBS+	*
EQ	0.75	0.71	0.73	0.76	0.69	0.72	0.71	0.70	0.84
LC	0.62	0.67	0.65	0.66	0.70	0.70	0.69	0.65	0.78
PDE	0.55	0.60	0.58	0.56	0.66	0.64	0.61	0.61	0.76
JDT	0.69	0.72	0.74	0.67	0.72	0.73	0.68	0.75	0.73
ML	0.60	0.58	0.60	0.63	0.65	0.69	0.65	0.70	0.79

相较基线分别提升了 27.6% 和 12.1%。这说明不是单纯在少数项目上“碰巧有效”，而是在不同规模、不同难度和不同不平衡水平的数据集上都保持了相对一致的竞争力。即便个别指标并非绝对最佳，其整体稳定性和不依赖采样的特点，仍然使该方法具有较好的实用意义。

3 结语

本文围绕软件缺陷预测中的多视图融合问题展开研究[19-21]。针对固定融合难以反映样本差异、不同视图贡献不均以及不平衡数据下性能波动较大的情况，本文提出了一种基于互补视图划分与融合的软件缺陷多视图深度学习方法。该方法先对单源过程度量进行语义划分，形成两个互补子视图；再利用视图级贡献分配机制调整融合过程；同时借助判别性约束改善表示质量。

实验结果表明，本文方法在 AEEEM[22-23] 数据集上的整体表现优于多数对照方法，在 *F*-measure 和 AUC 两项指标上都展现出较好的竞争力。由于方法没有改变原始数据分布，因此在公平性、可解释性和实际部署方面也具备一定优势。

后续工作仍可继续推进。例如，可以把该框架扩展到跨项目缺陷预测场景，考察其迁移能力；也可以引入产品度量、语义信息和静态分析结果，构建更丰富的多模态输入；此外，还可结合注意力分布与特征贡献分析，进一步增强模型解释性。

参考文献：

[1] 李杭桔 . 深度学习在软件缺陷检测交叉实验中的应用研究 [J]. 软件 , 2025, 46(3): 153-155.

[2]CHEN H W, JING X Y, LI Z Q,et al. An empirical study on heterogeneous defect prediction approaches[J]. IEEE transactions on software engineering, 2021, 47(12):2803-2822.

[3] 陈翔 , 顾庆 , 刘望舒 , 等 . 静态软件缺陷预测方法研究 [J]. 软件学报 , 2016, 27(1): 1-25.

[4] 杨丰玉 , 黄雅璇 , 周世健 , 等 . 结合多元度量指标软件缺陷预测研究进展 [J]. 计算机工程与应用 , 2021, 57(5): 10-24.

[5] 郭圣 , 仲兆满 , 李存华 . 基于深度自编码的多视图子空间聚类网络 [J]. 计算机工程与应用 , 2020, 56(17): 60-68.

[6]YANG M H, YANG S K, ERIC WONG W. Multi-objective software defect prediction via multi-source uncertain information fusion and multi-task multi-view learning[J]. IEEE transactions on software engineering, 2024, 50(8): 2054-2076.

[7]CHEN J Y, YANG Y T, HU K K, et al. Multiview transfer learning for software defect prediction[J]. IEEE access, 2019, 7: 8901-8916.

[8] 张宸嘉 , 朱磊 , 俞璐 . 卷积神经网络中的注意力机制综述 [J]. 计算机工程与应用 ,2021,57(20):64-72.

[9] 李莉 , 赵鑫 , 石可欣 , 等 . 结合特征对齐与实例迁移的跨项目缺陷预测 [J]. 计算机应用研究 ,2023,40(10):3091-3099.

[10] 兰雨晴 , 高静 . 软件质量度量和软件过程度量 [J]. 计算机系统应用 ,2003(9):69-72.

[11]GUO G D, WANG H, BELL D, et al. KNN model-based approach in classification[C]//On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. Berlin: Springer, 2003: 986-996.

[12]SAGAE K, TSUJII J I. Dependency parsing and domain adaptation with data-driven LR models and parser ensembles[J]. Trends in parsing technology. Berlin:Springer, 2010: 57-68.

[13]JING X Y, CHEN H W, XU B W. Other research questions of SDP[C]//Intelligent Software Defect Prediction. Singapore: Springer Nature Singapore. Berlin: Springer, 2024: 171-201.

[14]BAI Y Q, NIU B L, CHEN Y. New SDP models for protein homology detection with semi-supervised SVM[J]. Optimization, 2013, 62(4): 561-572.

（下转第 31 页）

成的高质量结构化特征，这些特征能够更准确地表征软件行为，从而在差异特征比对与缺陷匹配过程中有效降低误判与重复。实验证明，本文方法通过深度学习驱动的语义解析确保了测试用例的高质量与高覆盖度，为后续缺陷判定奠定了坚实基础；而基于深层特征表达的智能判定则在此基础上实现了高精准识别，共同保障了最终测试效果的全面提升。

3 结语

本文针对复杂软件系统中语义理解不足与隐性缺陷难以挖掘的挑战，构建了一种基于深度学习的全流程自动化测试框架。核心创新在于三个关键步骤：首先，设计了基于深度学习的语义解析与特征提取机制，能够从多源非结构化信息中精准捕捉语义依赖与动态参数约束，为生成高覆盖度测试用例奠定基础；其次，提出了融合动态负载与依赖关系的用例调度策略，通过量化优先级与实时资源感知，保障测试执行的高效稳定；最后，建立了基于加权余弦相似度的缺陷智能判定模型，利用深层特征实现缺陷的精准识别与分级。实验结果表明，本方法在缺陷发现率、类型识别准确率等关键指标上显著优于传统方法，尤其在挖掘隐性缺陷与控制误检漏检方面优势明显，验证了深度语义理解驱动测试全流程的有效性。

参考文献：

[1] 肖英剑，陈婷．基于深度学习的人工智能软件测试优化分析 [J]. 集成电路应用 ,2025,42(6):146-147.

[2] 杨梅芳．基于云计算的分布式软件测试方法 [J]. 信息与电脑 (理论版),2023,35(18):4-6.

[3] 杨洋，杨庆婷，彭培．基于 RRT 技术的应用软件集成测试方法 [J]. 长江信息通信 ,2023,36(6):63-65.

[4] 严文昊，缪慧宇，龚健，等．基于 SDL 技术的软件自动化测试方法研究 [J]. 无线互联科技 ,2024,21(21):100-102.

[5] 刘阳，寇力，杨基宏，等．基于语义匹配模型的软件测试用例共享和复用方法分析 [J]. 电子技术 ,2025,54(2):83-85.

[6] 吕虹，谢琳．基于深度学习的软件错误定位与修复方法研究 [J]. 信息记录材料 ,2024,25(9):129-131.

[7] 胡明，蔡传军，童绪军．基于深度神经网络的信息管理软件自动化测试方法研究 [J]. 佳木斯大学学报 (自然科学版),2024,42(5):30-33.

[8] 马玲玲．基于强化学习的自动化软件测试用例生成方法探索 [J]. 电脑编程技巧与维护 ,2025(9):56-58.

[9] 崔靖．基于深度学习的计算机应用软件自动化测试方法 [J]. 中阿科技论坛 (中英文),2025(6):97-100.

【作者简介】

王晓宇（1986—），女，河南淮阳人，硕士，助教，研究方向：计算机应用技术。

孔杰（1989—），男，河南焦作人，硕士，助教，研究方向：计算机视觉。

（收稿日期：2025-12-16 修回日期：2026-04-13）

（上接第 27 页）

[15]ARAR Ö F, AYAN K. Software defect prediction using cost-sensitive neural network[J]. Applied soft computing, 2015, 33: 263-277.

[16]AGRAWAL A, MENZIES T. Is "better data" better than "better data miners" ? on the benefits of tuning SMOTE for defect prediction[C]//ICSE'18: Proceedings of the 40th International Conference on Software Engineering. NewYork: ACM, 2018: 1050-1061.

[17]YANG X L, LO D, XIA X, et al. Deep learning for just-in-time defect prediction[C]//2015 IEEE International conference on software quality, reliability and security. Piscataway: IEEE, 2015: 17-26.

[18]HUANG Q, XIA X, LO D. Revisiting supervised and unsupervised models for effort-aware just-in-time defect prediction[J]. Empirical software engineering, 2019, 24: 2823-2862.

[19] 陈翔，王莉萍，顾庆，等．跨项目软件缺陷预测方法研究综述 [J]. 计算机学报 , 2018, 41(1): 254-274.

[20] 任泽裕，王振超，柯尊旺，等．多模态数据融合综述 [J]. 计算机工程与应用 , 2021,57(18):49-64.

[21] 化盈盈，张岱墀，葛仕明．深度学习模型可解释性的研究进展 [J]. 信息安全学报 , 2020,5(3):1-12.

[22] 纪晨辉，李英梅．一种邻域合成的软件缺陷预测过采样方法 [J]. 软件工程与应用 , 2023, 12: 930-939.

[23] 纪晨辉．软件缺陷预测中数据高维性和不平衡性问题处理研究 [D]. 哈尔滨 : 哈尔滨师范大学 ,2024.

【作者简介】

姚永芳（1975—），女，湖南常德人，硕士，助理研究员，研究方向：模式识别、人工智能、故障诊断、软件工程。

丁永昌（1997—），男，浙江金华人，博士研究生，研究方向：软件缺陷检测、人工智能。

陈林君（1996—），男，江西九江人，博士研究生，研究方向：类不平衡、模式识别。

荆晓远（1971—），通信作者（email:jingxy_2000@126.com），男，江苏南京人，博士，教授，研究方向：模式识别、人工智能、故障诊断、软件工程。

（收稿日期：2025-08-28 修回日期：2026-04-02）