

# 雷达软件测试知识图谱构建技术研究与应用

王 杰<sup>1</sup> 郝 明<sup>1</sup> 柳 溪<sup>1</sup> 秦 剑<sup>1</sup>  
WANG Jie HAO Ming LIU Xi QIN Jian

## 摘 要

为充分利用雷达软件测试用例与其他测试知识的关系,进一步优化雷达软件测试用例推荐效果,文章提出了一种基于知识图谱的雷达软件测试智能推荐方法。依据雷达测试本体关系模型,从现有的雷达软件测试管理系统中抽取测试知识构建雷达软件测试知识图谱,以实体特征属性文本的语义相似度匹配为基础,结合知识推荐和协同过滤推荐技术建立雷达软件测试用例推荐模型,并使用该模型开展测试用例推荐设计和测试会话生成。相较于传统基于语义相似度匹配推荐方法,文章将图谱关系信息及用户推荐用例的交互图融合到一个图空间,融合协同过滤算法优势和知识图谱信息,同时发现高阶的关系信息,克服了经典传统的协同过滤算法需要大量的用户行为数据,即需要大量的冷启动数据缺点,同时提高推荐算法的预测召回率和归一化折损累计增益指标。

## 关键词

知识图谱; 实体抽取; 关系抽取; 软件测试; 雷达软件; KGAT 算法

doi: 10.3969/j.issn.1672-9528.2025.02.004

## 0 引言

随着软件及其相关领域技术的不断成熟和发展,各种军事装备、系统以及平台的软件化进程也在不断深化和普及,软件及其相关产品逐渐成为影响整个军事装备产品质量的重要因素<sup>[1]</sup>。雷达在军事电子主导的高科技战争中一直占据核心位置。因此,加强对雷达软件产品测试研究将对提升整个雷达产品质量具有重要意义。随着人工智能、大数据等互联网信息技术飞速发展,并逐渐被应用于测试领域,对于软件测试验证质量和效率的提升提供了新思路<sup>[2]</sup>。知识图谱以结构化的形式描述客观世界中概念、实体及其关系,将各类信息表达成更接近人类认知世界的形式,提供了一种更好的组织、管理和理解海量信息的能力<sup>[3]</sup>。通过构建雷达软件测试知识图谱,能够帮助测试人员检索、获取各类软件测试信息并开展分析,挖掘如测试缺陷、测试用例间的潜在关系,并为后续软件测试用例推荐等智能化技术研究提供基础。

软件测试用例推荐研究的重点是用例的匹配检索<sup>[4]</sup>。现有研究大多数通过构建测试用例本体等方法改进推荐用例的检索算法,但大部分仅从用例自身复用检索角度进行设计,未充分考虑专业领域用例本体与其他知识本体的关系,降低了对具体项目的针对性和适用性,影响在工程实际应用中的效果。同时,由于传统软件测试用例推荐库大多采用关系型数据库作为底层数据存储,当被测软件项目数量快速增加时,使用传统的推荐库设计架构已逐渐无法满足清晰描述这些不

同项目间知识关联关系的要求,涉及到更加复杂的测试知识关系查询需求,因此对整个软件测试资产库的数据检索效率提出了更高的要求。

基于以上分析,本文在软件测试用例推荐设计中引入知识图谱相关技术,根据构建的雷达软件测试本体关系模型,依托现有的雷达软件测试管理系统软硬件平台,快速构建雷达软件测试知识图谱,设计一种以实体特征属性文本和关联节点快速相似度计算为基础的软件测试知识匹配方法,并利用知识检索和协同过滤推荐等技术手段开展基于知识图谱的雷达软件测试用例推荐应用。该方法充分发挥了图形数据库在关系检索性能上的优势,能够有效降低测试知识库的更新复杂度,将测评项目的软件需求、软件测试用例和软件缺陷等知识和关联关系,以及相关人员的测试工作档案以图谱形式直观展示,从而为雷达软件测试用例推荐应用的进一步研究提供了新的解决思路。

## 1 雷达软件测试知识图谱构建技术

知识图谱的构建技术包含数据准备、知识抽取、知识融合等方面。雷达软件测试知识图谱技术架构是知识图谱构建方法和实现方案的具体表示,包含从数据来源到数据加工、存储的整个过程。雷达软件测试知识图谱逻辑架构是知识图谱的基础,包括模式层和数据层两部分,模式层在数据层之上,存储经过提炼的知识,雷达软件测试知识图谱数据层主要包含数据的提取及数据的存储。雷达软件测试知识图谱的可视化展示是通过界面对存储的知识进行可视化展示,如图

1. 南京电子技术研究所 江苏南京 210039

1 所示。雷达软件测试知识图谱的构建思路以结构化数据为起点, 逐步进行研究和扩展。以 HP ALM 等数据为源头, 开发接口工具对历史数据进行读取和导出, 提取出数据的本体、关系和属性元素, 然后通过数据清洗和整理, 为数据打上唯一特征标识, 对实体进行对齐, 然后将数据导入 Neo4j 等知识图谱平台中, 进行可视化展示并定期更新。

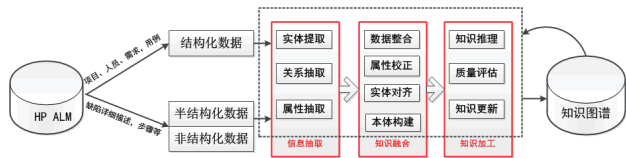


图 1 雷达软件测试实体关系模型

## 1.1 数据准备

通过软件代码静态分析工具对编码规范进行检查、软件代码安全漏洞检测工具开展软件安全漏洞检查、软件测试仿真工具对嵌入式软件所有重要接口进行针对性的缺陷检测、仿真工具和代码分析工具对可编程逻辑器件代码进行缺陷检测、界面自动化测试工具对操作界面开展自动化测试,借助应用软件生命周期管理平台(application lifecycle management)实现全流程软件测试的数据存储。依托软件测试管理平台管理 18 个领域近千个软件测试项目,包含软件测试需求、测试用例、测试问题等数据,其中软件测试用例超过 30 万条、测试问题超过 3 万个、文档审查和软件质量问题上千条、各类统计数据近千项。以上测试数据为雷达软件知识图谱构建提供数据来源,如图 2 所示。

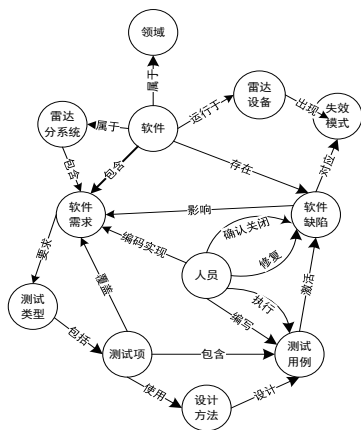


图2 雷达软件测试实体关系模型

## 1.2 知识抽取

知识抽取是实现自动化构建大规模知识图谱的重要技术,其目的在于从不同源、不同结构的数据中进行知识提取并存入知识图谱中。知识抽取的数据源可以是结构化数据(如链接数据库、数据库)、半结构化数据(如网页中的表格、列表)或者非结构化数据(即纯文本数据)。面向不同类型的数据源,知识抽取涉及的关键技术和需要解决的技术难点有所不同。

具体知识抽取包括以下任务:

### (1) 实体抽取

从文本中检测出命名实体,并将其分类到预定义的类别中,例如雷达、预警机、天线、阵面等。一般情况下,实体抽取是知识抽取其他任务的基础。

常见的实体抽取算法包括基于规则的方法、基于统计的方法、基于机器学习的方法等<sup>[4]</sup>。考虑到基于机器学习的方法是一种比较常用的算法,本文采取基于机器学习的方法开展实体抽取工作,其主要流程包括:

数据预处理：对原始数据进行清洗、分词、词性标注等处理，生成训练数据集；

特征提取：对训练数据集中的每个样本提取一组特征，如词频、TF-IDF、词向量等；

模型训练：将特征和标签输入到机器学习模型中进行训练，得到一个分类器：

实体抽取：使用训练好的分类器对新的文本数据进行分类，提取出有意义的知识信息。

## (2) 关系抽取

关系抽取是知识抽取的重要子任务之一，面向非结构化文本数据，关系抽取从文本中抽取两个或者多个实体之间的语义关系，例如，从句子“雷达一般由发射机、天线、接收机、显示器、天线控制设备和电源设备等组成”中识别出实体“雷达”和“发射机”“天线”“接收机”之间具有“父子”关系。

关系抽取与实体抽取密切关系,一般在实体抽取后,再抽取实体之间可能存在的关系。本部分采取基于监督学习的关系抽取方法<sup>[5]</sup>。

基于监督学习的关系抽取方法将关系转换为分类问题，在大量标注数据的基础上，训练有监督学习模型进行关系抽取。利用监督学习方法进行关系抽取的一般步骤包括预定义关系的类型、人工标注数据、设计关系识别所需的特征，一般根据实体所在句子的上下文计算获得选择分类模型（如支持向量机、神经网络和朴素贝叶斯等）、基于标注数据训练模型、对训练模型进行评估。

通过实体抽取和关系抽取获得的实体 - 关系 - 实体三元组后, 可构建如图 3 所示的雷达软件测试知识图谱。



图3 雷达软件测试知识图谱示意图

### (3) 知识融合

知识融合将不同来源、不同格式、不同质量的知识融合到知识图谱中。新的知识加入雷达软件测试知识图谱前,需要进行知识图谱的实体匹配(Entity Matching)。即将新建的知识与雷达软件测试图谱中已有的知识进行实体、属性和关系的关联,排除可能存在的概念实体和属性冲突,实现对相同语义实体的共指消解和消歧<sup>[6]</sup>。常见的知识融合技术包含:

① 实体融合:将来自不同数据源的相同或相似实体进行匹配和融合,生成一个唯一的实体表示。

② 关系融合:将来自不同数据源的相同或相似关系进行匹配和融合,生成一个准确的关系表示。

知识图谱的匹配方法一般分为基于快速相似度计算、基于规则和基于分治3种方法。考虑到现有雷达软件测试知识图谱的规模和特点,本文在实体属性类型分块的基础上选择了一种基于加权的实体特征属性文本的CBOW(continuous bag-of-words)模型自然语言处理(NLP)算法,快速开展各种软件测试知识实体匹配。

对于输入语义文本,经语义关键词提取和英文缩略语翻译后,得到该语句的中文关键词序列 $S = \{\omega_1, \omega_2, \dots, \omega_n\}$ ,利用CBOW模型的训练输出文件可以获得每个关键词的词向量值,再将句子中的所有词向量求均值后即得到该属性文本的句子向量 $\text{vec}(S)$ 。

给定两个待匹配的实体 $u$ 和 $v$ ,两者具有相同的文本属性集合 $A = \{A_1, A_2, \dots, A_n\}$ ,通过计算 $u$ 、 $v$ 对共同属性 $A_j$ ( $0 \leq n \leq N$ )描述文本的句子向量 $S_{u,j}$ 和 $S_{v,j}$ 间的余弦值,可以得到实体 $u$ 和 $v$ 在该属性 $A_j$ 上的相似度 $\text{sim}_{\text{attr}}(u_j, v_j)$ 。

$$\text{sim}_{\text{attr}}(u_j, v_j) = \frac{\text{vec}(S_{u,j}) \cdot \text{vec}(S_{v,j})}{|\text{vec}(S_{u,j})| |\text{vec}(S_{v,j})|} \quad (1)$$

考虑到不同实体特征属性值对相似度计算结果的影响程度不同,为每个特征属性的匹配计算结果设置不同的影响因子 $D_j$ 并进行综合加权修正后求平均,最终得到实体 $u$ 和 $v$ 的属性综合相似度。

$$\text{sim}_{\text{attr}}(u, v) = \frac{1}{n} \sum_j D_j \text{sim}_{\text{attr}}(u_j, v_j) \quad (2)$$

## 2 雷达软件测试知识图谱应用

近年来,深度推荐模型算法(如DeepFM)模型暴露出一个缺点:将训练数据(用户交互数据)的特征进行独立建模,未充分考虑交互数据之间的关系<sup>[7]</sup>,使模型不足以从用户的行为中提取基于属性的协同信息;而经典传统的协同过滤算法(collaborative filtering, CF)需要大量的用户行为数据,即需要大量的冷启动数据,并且难以给出合理的推荐解释。为解决上述问题,本文提出KGAT算法(knowledge graph

attention network for recommendation),将图谱关系信息及用户选择用例的交互图融合到一个图空间,可以融合CF信息和KG信息,同时发现高阶的关系信息。

KGAT是一个基于图神经网络的推荐系统模型<sup>[8]</sup>,用于学习需求和用例之间的复杂交互关系,下面给出KGAT模型的数学描述。

假如有 $N$ 个需求和 $M$ 个用例,其之间的交互可以用一个二部图表示,节点集合 $V = V_d \cup V_i$ ,其中 $V_d$ 表示需求节点集合, $V_i$ 表示用例节点集合。同时,每个节点都有一个对应的嵌入向量,其中 $u_i$ 表示第 $i$ 个需求的嵌入向量, $v_j$ 表示第 $j$ 个用例的嵌入向量。

为学习需求和用例之间的交互关系,KGAT使用多层卷积神经网络(GCN)来学习节点的表示。具体来说,假设在第 $k$ 层GCN中,节点 $i$ 的表示向量为 $h_i^{(k)}$ ,则有以下更新规则:

$$h_i^{(k)} = \sigma \left( \sum_{j \in N(i)} \frac{1}{\sqrt{d_i d_j}} h_j^{(k-1)} W^{(k-1)} \right) \quad (3)$$

式中: $N(i)$ 表示节点 $i$ 的邻居节点集合; $d_i$ 表示节点 $i$ 的度数; $\sigma$ 表示非线性激活函数; $W^{(k-1)}$ 表示第 $k-1$ 层GCN参数。

KGAT采用注意力机制学习不同节点之间的重要性权重。具体来说,对于每个需求节点 $i$ ,KGAT将其邻居用例节点 $j$ 的嵌入向量 $v_j$ 和需求节点 $i$ 的嵌入向量 $u_i$ 拼接到一起,然后通过一层全连接网络 $f$ 得到一个注意力向量 $\alpha_i$ ,公式为:

$$\alpha_i = \text{softmax} (f([u_i; v_j], \theta)) \quad (4)$$

式中: $[u_i; v_j]$ 表示向量拼接操作; $\theta$ 表示全连接网络参数; $\alpha_i$ 表示需求节点 $i$ 对所有邻居用例节点的注意力权重。

最后,对于每个需求节点 $i$ ,KGAT将其邻居用例节点 $j$ 的表示向量 $h_j^{(K)}$ 按照注意力向量 $\alpha_i$ 进行加权平均,得到需求节点 $i$ 的最终表示向量 $z_i$ ,公式为:

$$z_i = \sum_{j \in N(i)} \alpha_i(j) h_j^{(K)} \quad (5)$$

式中: $K$ 表示图卷积网络(graph convolutional networks, GCN)的层数; $z_i$ 表示需求节点 $i$ 的最终表示向量,用于预测需求对未见过的用例的评分。

## 3 工程应用验证

### 3.1 知识融合算法比较

本文选取了8个典型雷达软件配置项测试数据作为实验输入,评估CBOW模型在知识图谱实体匹配算法检索中的应用性能,同时选用编辑距离计算(levenshtein distance)、JACCARD系数计算以及TF-IDF等中文文本相似度计算常用算法进行效果比对。这些项目覆盖雷达显控、数据处理、信号处理等多个分系统,各测试项目主要实体数量和知识图谱查询时间数据如表1所示<sup>[9]</sup>。

表 1 项目验证数据

| 项目<br>编号 | 测试知识实体数量 |     |      |     |      | 查询时间 /s |
|----------|----------|-----|------|-----|------|---------|
|          | 软件需求     | 测试项 | 测试用例 | 缺陷  | 合计   |         |
| P1       | 199      | 338 | 1981 | 140 | 2658 | 0.409   |
| P2       | 170      | 261 | 905  | 98  | 1434 | 0.233   |
| P3       | 83       | 146 | 1874 | 318 | 2421 | 0.434   |
| P4       | 159      | 258 | 1764 | 114 | 2295 | 0.337   |
| P5       | 158      | 296 | 2039 | 211 | 2704 | 0.432   |
| P6       | 159      | 258 | 1764 | 114 | 2295 | 0.336   |
| P7       | 307      | 496 | 1663 | 116 | 2582 | 0.391   |
| P8       | 472      | 687 | 2693 | 396 | 4248 | 0.721   |

### 3.2 基于测试需求知识匹配的测试用例推荐算法比较

在雷达软件测试知识图谱的模式层中，包含项目、软件需求、测试需求、测试用例和缺陷等实体，具体的模式层结构如表 2 所示。

表 2 雷达软件测试知识图谱模式层

| 实体   | 属性     | 关系                            |
|------|--------|-------------------------------|
| 测试项目 | 项目名称   |                               |
|      | 域名称    |                               |
|      | 项目序号   | 通过唯一标识的项目序号将测试项目与软件需求数据进行链接   |
| 软件需求 | 项目序号   |                               |
|      | 需求名称   |                               |
|      | 需求描述   |                               |
|      | 需求类型   |                               |
| 测试需求 | 软件需求序号 | 通过唯一标识的项目序号将软件需求与测试需求数据进行链接   |
|      | 软件需求序号 |                               |
|      | 项目序号   |                               |
|      | 需求名称   |                               |
|      | 需求描述   |                               |
|      | 需求类型   |                               |
| 测试用例 | 测试需求序号 | 通过唯一标识的测试需求序号将测试需求与测试用例数据进行链接 |
|      | 测试需求序号 |                               |
|      | 用例名称   |                               |
|      | 用例描述   |                               |
|      | 用例类型   |                               |
|      | 测试用例序号 | 通过唯一标识的测试用例序号将测试用例与软件缺陷数据进行链接 |

通过接口工具利用软件测试管理系统开放的接口函数从后台对 ALM 数据库进行访问和读取，对存储在 ALM 中的软件测试项目依次进行检索，并完成测试数据的导出。经过数据的检索、导出、加工处理和导入知识图谱平台，最终创建了 497 个项目实体节点，20 512 个软件需求实体节点，

50 042 个测试需求节点，165 813 个测试用例实体节点和 11 566 个软件缺陷数据节点，并建立各实体间的关系。

基于构建的雷达软件知识谱图，开展基于测试需求知识匹配的测试用例推荐算法研究。首先，定义推荐算法的 3 个衡量指标，分别为准确率、召回率和归一化折损累计增益<sup>[10]</sup>如下所示。

计算预测准确率 KA，即预测为真的样本中，预测正确的样本占比为：

$$KA = \frac{TP}{TP + FP} \quad (6)$$

计算预测召回率 KE，即预测为真并且正确的数量在实际为真的数量中的占比为：

$$KE = \frac{TP}{TP + FN} \quad (7)$$

式中：TP 为系统成功推荐的软件测试用例数据量；FP 为系统错误推荐的软件测试用例数据量；FN 为系统未推荐出来的满足推荐条件的软件测试用例数据量。

为了让高度关联的结果比一般关联度的结果更加影响最终的指标得分，有高度关联度的结果出现在更靠前的位置的时候，指标会更高<sup>[11]</sup>，构建归一化折损累计增益 NDCG 指标分别为：

$$NDCG = \frac{DCG}{IDCG} \quad (8)$$

$$DCG = \sum_{i=1}^p \frac{2^{rel_i}}{\log_2(i+1)}, \quad IDCG = \sum_{i=1}^{|REL|} \frac{2^{rel_i} - 1}{\log_2(i+1)} \quad (9)$$

式中：rel<sub>i</sub> 表示第 i 个位置的 item 是否是用户喜欢，rel<sub>i</sub>=0.1；REL 表示将原始的召回集合按照得分从大到小排序后的集合。

选择 Amazon-Book、Last-FM 和 Yelp2018 三个公开基准上开展大量试验，证明 KGAT 算法的有效性及其在理解高级关系重要性方面的可解释性<sup>[12]</sup>。

表 3 雷达软件测试知识图谱模式层

| 算法<br>名称  | Amazon-Book |         | Last-FM |         | Yelp2018 |         |
|-----------|-------------|---------|---------|---------|----------|---------|
|           | KE          | NDCG    | KE      | NDCG    | KE       | NDCG    |
| User-CF   | 0.114 5     | 0.077 1 | 0.072 5 | 0.114 5 | 0.052 4  | 0.064 6 |
| NMF- CF   | 0.136 6     | 0.091 3 | 0.082 9 | 0.121 4 | 0.066 0  | 0.081 2 |
| RippleNet | 0.113 6     | 0.091 3 | 0.079 3 | 0.123 7 | 0.066 5  | 0.082 4 |
| KGAT      | 0.148 7     | 0.101 6 | 0.086 5 | 0.131 5 | 0.072 2  | 0.087 7 |
| Improv    | 8.95%       | 10.05%  | 4.93%   | 5.77%   | 7.18%    | 5.54%   |

以表 3 中的需求知识实体集合为输入，分别选择 User-CF 和 KGAT 两种算法对所在单位雷达产品软件项目进行试验，其预测准确率 KA 和预测召回率 KE 如图 4、图 5 所示。

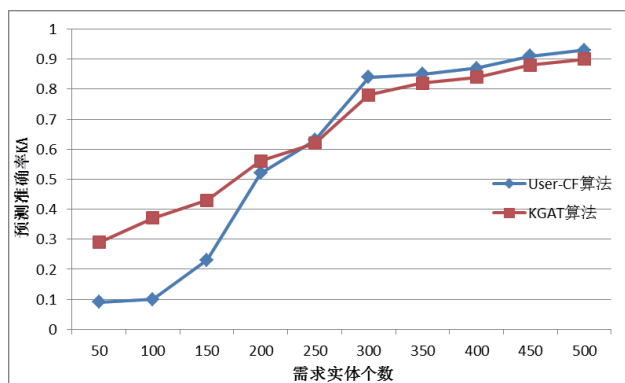


图4 不同需求实体个数下两种算法预测准确率比较

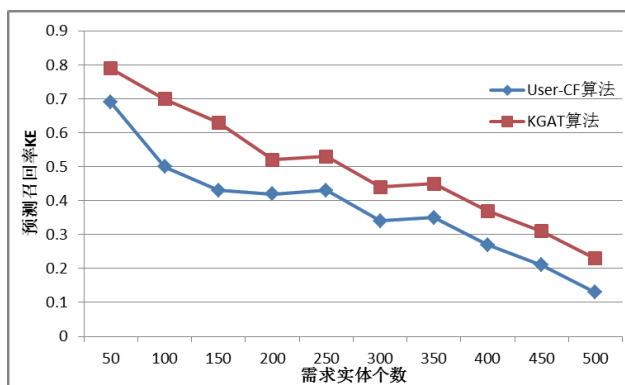


图5 不同需求实体个数下两种算法预测召回率比较

#### 4 结语

本文利用雷达软件测试管理系统数据库中存储有400余个历史测试项目记录数据，利用测试管理平台提供的开发接口，获取这些历史数据并建立雷达软件测试知识图谱，以此为基础提出一种基于知识图谱的雷达软件测试用例推荐方法。通过对所在单位近期使用基于雷达软件测试知识图谱开展用例推荐设计的20个测试项目进行复用情况统计，平均用例预测准确率到了90%，累计节约测试用例设计开销时间约为63%，提高了软件测试用例设计效率。

由于图谱推荐阶段存在复杂规则表达困难的情况，知识计算的主要对象多是二元关系，通常处理多元关系的方法是将其拆分为二元关系进行推理。然而，将多元关系拆分会损失结构信息，如何尽可能完整地利用多元关系中复杂的隐含信息进行推理仍有较多工作需要继续研究。

#### 参考文献:

- [1] 刘小齐, 杨根兴, 蔡立志, 等. 基于复用行为的测试用例本体描述和检索[J]. 计算机应用与软件, 2011, 28(10): 65-68.
- [2] 张吉祥, 张祥森, 武长旭, 等. 知识图谱构建技术综述[J]. 计算机工程, 2022, 48(3): 23-37.

- [3] 徐增林, 盛泳潘, 贺丽荣, 等. 知识图谱技术综述[J]. 电子科技大学学报, 2016, 45(4): 589-606.
- [4] WANG X, HE X N, CAO Y X, et al. KGAT: knowledge graph attention network for recommendation[C]//Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019: 950-958.
- [5] JI S X, PAN S R, CAMBRIA E, et al. A survey on knowledge graphs: representation, acquisition, and applications[J]. IEEE transactions on neural networks and learning systems, 2021, 33(2): 494-514.
- [6] WANG L, SUN C H, ZHANG C Y, et al. Application of knowledge graph in software engineering field: a systematic literature review[J]. Information and software technology, 2023: 107327.
- [7] FIENGO R, ADAMS A. Test and evaluation of radar systems operating in the modern electromagnetic spectrum[J]. IET radar, sonar & navigation, 2024, 18(11): 2189-2198.
- [8] JIN D S, WANG Z, LI M Y, et al. Construction and application of knowledge graph of domestic operating system testing[C]//Proceedings of the 4th International Conference on Computer Science and Software Engineering. New York: ACM, 2021: 268-273.
- [9] 柳溪. 探索式测试在雷达软件中的应用研究[J]. 现代雷达, 2016, 38(9): 86-91.
- [10] 李昊, 田峰敏. 雷达软件的安全性测试研究[J]. 信息化研究, 2011, 37(6): 29-33.
- [11] BAILAN O, ROSSI U, WANTENS A, et al. Verification of soft error detection mechanism through fault injection on hardware emulation platform[C/OL]//2010 International Conference on Dependable Systems and Networks Workshops (DSN-W). Piscataway: IEEE, 2010[2024-05-02]. <https://ieeexplore.ieee.org/document/5542611>.
- [12] WILLIAMS D, HERMAN T. Data quality management for industry 4.0: a survey[J]. Software quality professional, 2020, 22(2): 26-35.

#### 【作者简介】

王杰(1993—), 男, 安徽铜陵人, 博士研究生, 工程师, 研究方向: 软件工程。

(收稿日期: 2024-10-25)