

计算机软件工程中的数据挖掘技术应用研究

马瑞林 郭维靖

天津数字认证有限公司 天津 300060

摘要：随着软件规模扩大与复杂度提升，计算机软件工程各阶段产生海量异构数据，传统数据处理方式已无法满足智能化发展需求。本文阐述了软件工程与数据挖掘的核心理论及适配性，分析数据挖掘应用前提、数据准备流程及四大核心技术选型细节，探讨数据挖掘在软件工程需求分析、设计、编码、测试、维护各阶段的具体应用。研究表明，数据挖掘技术可有效优化工程流程、提升软件质量，为软件工程智能化发展提供可靠技术支撑。

关键词：计算机；软件工程；数据挖掘技术；各阶段应用

引言：在数字化转型加速的背景下，软件工程朝着规模化、复杂化方向发展，数据杂乱、分析低效、决策盲目等问题日益突出，严重影响软件开发效率与质量。数据挖掘作为人工智能与数据库技术的融合产物，具备自动处理海量异构数据、挖掘潜在价值的优势，与软件工程的适配性极高。基于此，本文结合天津数字认证有限公司的行业实践，系统研究数据挖掘技术在软件工程中的应用前提、关键技术及各阶段具体应用，旨在解决软件工程现存痛点，推动软件工程向智能化、高效化转型，为相关领域应用提供参考。

1 计算机软件工程中的数据挖掘技术概述

1.1 计算机软件工程核心理论

计算机软件工程是融合计算机科学、工程学与管理学的交叉学科，核心是采用系统化、规范化、可量化的方法开发和维护软件系统。其核心流程涵盖需求分析、设计、编码、测试、维护五个关键阶段，目标是提高软件质量、降低开发成本、提升开发效率。软件工程具有模块化、可复用性与可维护性特征，随着软件规模和复杂度提升，工程过程中会产生需求文档、代码片段、测试报告等海量数据，既为数据挖掘提供了丰富载体，也对数据处理分析能力提出更高要求。

1.2 数据挖掘技术基础

数据挖掘是从海量杂乱数据源中，通过算法提取隐藏有效信息、规律和知识的过程，是人工智能与数据库技术结合的重要产物。其核心流程包括数据收集、预处理、挖掘分析、结果评估四个环节，常用核心技术分为四类：关联规则挖掘、聚类分析、分类与预测、异常检测，分别用于发现数据关联、分类分组、趋势预测和异常识别。其核心优势是能自动处理海量异构数据，无需人工干预即可挖掘潜在价值，为软件工程智能化发展提供技术支撑。

1.3 数据挖掘与软件工程的适配性分析

数据挖掘与计算机软件工程高度适配，二者融合可实现优势互补。数据层面，软件工程各阶段产生的结构化与非结构化数据，可满足数据挖掘的数据源需求；目标层面，数据挖掘的智能化分析能力，能解决软件工程中数据杂乱、分析低效、决策盲目的痛点，助力各阶段优化。例如，数据挖掘可分析需求数据提升准确性，挖掘代码数据提前识别缺陷，二者深度融合是推动软件工程向智能化、高效化发展的重要路径，也是当前该领域的研究热点^[1]。

2 计算机软件工程中数据挖掘的应用前提与关键技术

2.1 应用前提与数据准备

数据挖掘技术在计算机软件工程中的有效应用，需依托明确的前提条件和规范的数据准备流程，具体内容如下：（1）应用前提。一是软件工程数据体系完善，明确各阶段的数据产出节点、类型及存储规范，确保数据可追溯、可提取；二是硬件与软件支撑到位，配备满足海量数据处理需求的服务器、存储设备，以及Python数据分析库、数据挖掘平台等适配工具；三是明确挖掘目标，结合实际场景确定核心方向（如缺陷检测、需求梳理等），避免盲目挖掘；四是数据安全合规，梳理核心代码、用户数据等敏感信息，建立加密、访问控制机制，确保挖掘过程符合行业规范。（2）数据准备。首先是数据收集，全面采集各阶段异构数据，包括测试用例执行结果、缺陷统计等结构化数据，以及需求文档、代码注释等非结构化数据，确保覆盖挖掘目标全场景；其次是数据预处理，对原始数据进行清洗、集成、转换、归约，剔除缺失值、异常值和重复数据，将非结构化数据转为可分析格式，统一数据标准；最后是数据验证，校验预处理后数据的完整性与准确性，确认其符合挖掘算法输入要求，为后续技术应用奠定基础^[2]。

2.2 核心应用技术选型

2.2.1 关联规则挖掘技术

关联规则挖掘技术核心是挖掘软件工程中不同数据项的关联关系，常用算法包括Apriori算法、FP-Growth算法，应用细节如下：（1）算法选型原则，优先选用FP-Growth算法，其无需多次扫描数据集，能高效处理代码片段、缺陷记录等海量数据，降低时间复杂度；小型项目需求文档分析等数据量较小场景，可选用Apriori算法，兼顾简单易实现的优势。（2）技术实现细节，首先确定最小支持度和最小置信度阈值，筛选有效关联规则；其次将代码缺陷等级、测试用例执行结果等数据项离散化处理，便于规则挖掘；最后筛选关联规则，剔除无效项，保留与工程场景相关的有效规则。

2.2.2 聚类分析技术

聚类分析技术核心是按相似性对软件工程海量数据分组，无需预先定义类别，常用算法包括K-Means算法、层次聚类算法，细节如下：（1）算法选型原则，代码特征、测试数据等高维数据场景，优先选用K-Means算法，其计算高效、可扩展性强；需求文档分类、缺陷分层等需明确数据层次关系的场景，选用层次聚类算法。（2）技术实现细节，首先提取数据特征，如代码行数、注释率及测试时长、缺陷数量等；其次通过肘部法则、轮廓系数法确定最优K值；最后用轮廓系数评估聚类效果，确保同类数据相似度高、异类数据相似度低。

2.2.3 分类与预测技术

分类与预测技术核心是通过历史数据建模，对未知数据分类或趋势预测，常用算法包括决策树、支持向量机（SVM）、逻辑回归，具体细节如下：（1）算法选型原则，缺陷、需求类型分类等场景优先选用决策树算法，模型可解释性强、无需数据归一化；代码缺陷预测等高维、非线性数据场景选用SVM，泛化能力强且适配小样本；测试用例通过/失败预测等二分类场景选用逻辑回归，计算高效、结果直观。（2）技术实现细节，按7:3比例划分训练集与测试集，用训练集建模；通过交叉验证、网格搜索优化算法参数，提升模型准确率；将测试集输入模型验证，确保准确率达80%以上，满足工程实际需求。

2.2.4 异常检测技术

异常检测技术核心是识别偏离正常模式的异常数据，常用算法包括孤立森林、DBSCAN算法、One-ClassSVM，具体细节如下：（1）算法选型原则，代码异常、测试数据异常等海量数据场景，优先选用孤立森林算法，计算快、内存占用低；系统运行日志等具有密度

特性的数据场景，选用DBSCAN算法，无需预设异常阈值；高维数据小样本异常检测场景，选用One-ClassSVM算法。（2）技术实现细节，结合工程场景，通过历史异常数据校准异常阈值，避免误检、漏检；对数据进行特征工程，提取CPU使用率、代码语法规范度等反映正常模式的特征；通过算法分析数据、标记异常，并初步判断异常原因，为问题排查提供支撑^[3]。

3 数据挖掘技术在软件工程各阶段的具体应用

3.1 在需求分析阶段的具体应用

需求分析阶段是软件工程的起始环节，核心是明确用户需求、梳理需求逻辑，数据挖掘技术主要应用于需求数据的整理、分析与筛选，具体应用细节如下：（1）需求数据收集与规整，通过数据挖掘技术采集用户反馈、市场调研数据、行业规范文档、同类项目需求案例等多渠道数据，利用文本挖掘技术将用户访谈记录、留言反馈等非结构化需求描述，转换为结构化数据，提取功能、性能、安全等核心需求关键词，剔除无效、重复信息，建立标准化需求数据库。（2）需求关联与冲突识别，采用FP-Growth等关联规则挖掘算法，分析不同需求间的关联关系，识别强关联需求，辅助需求梳理；同时通过异常检测技术，识别需求描述中的矛盾点、冲突点，标记异常需求并反馈，减少需求歧义。（3）需求优先级初步划分，基于用户关注度、实现难度、商业价值等需求数据特征，采用聚类分析算法将需求划分为核心、重要、一般三类，为后续排序提供数据支撑，提升划分客观性，减少人工主观判断误差。

3.2 在设计阶段的具体应用

设计阶段核心是将需求转化为可落地的软件架构与设计方案，数据挖掘技术主要应用于设计模式复用、设计缺陷预判，具体应用细节如下：（1）设计模式挖掘与复用，采集同类项目设计文档、架构图等数据，利用聚类分析、关联规则挖掘技术，提取单例、工厂等成熟设计模式，建立标注适用场景、核心逻辑的设计模式数据库；当前项目设计中，通过相似度匹配算法检索适配设计模式，直接复用成熟方案，降低设计难度。（2）设计缺陷提前识别，收集架构冗余、模块耦合过高、接口设计不合理等历史设计缺陷数据，建立缺陷特征库，采用决策树等分类与预测算法，分析当前设计方案的模块结构、接口设计、数据流向，标记潜在缺陷及类型，为设计优化提供方向。（3）设计参数优化，针对模块划分粒度、接口响应时间阈值等关键设计参数，通过数据挖掘分析历史项目设计参数与运行效果的关联数据，挖掘最优参数范围，为当前参数设定提供参考，避免参数不合

理引发后续问题^[4]。

3.3 在编码阶段的具体应用

编码阶段是将设计方案转化为可执行代码的核心环节,数据挖掘技术主要应用于代码质量管控、代码复用,具体应用细节如下:(1)代码缺陷实时检测,通过静态代码分析工具采集代码片段,提取代码复杂度、注释率、语法规范性等特征,结合历史缺陷数据,采用孤立森林等异常检测算法,实时识别空指针异常、数组越界、逻辑错误等潜在缺陷,标记位置及类型并同步反馈开发人员,实现缺陷早发现、早修正。(2)代码复用挖掘,采集项目内部及同类项目优质代码片段,利用聚类分析技术按数据查询、权限验证、日志记录等功能模块分类,建立代码复用库;通过代码特征匹配,检索适配当前编码需求的片段,开发人员可基于复用代码修改,减少重复编码,提升效率。(3)编码规范合规性检查,建立命名规范、代码缩进、注释要求等编码规范特征库,利用数据挖掘技术对比当前代码与规范的差异,标记不合规片段并提醒整改,确保编码风格统一,降低后续维护难度。

3.4 在测试阶段的具体应用

测试阶段核心是验证软件功能、发现潜在缺陷,数据挖掘技术主要应用于测试用例优化、缺陷定位,具体应用细节如下:(1)测试用例自动生成与优化,采集历史测试用例、缺陷记录、代码特征数据,采用支持向量机等分类与预测算法,分析测试用例与缺陷的关联关系,挖掘高效用例特征,自动生成覆盖核心功能、高缺陷检出率的用户例;同时通过关联规则挖掘识别冗余用例,删减优化以降低测试成本。(2)缺陷定位与分类,采集缺陷描述、出现场景、触发条件等测试缺陷数据,利用聚类分析技术将缺陷分为功能、性能、兼容性等类别,明确分布规律;通过关联规则挖掘分析缺陷与代码模块、测试用例的关联,快速定位核心原因,辅助排查。(3)测试重点聚焦,基于历史测试数据,采用分类与预测算法,预测易出现缺陷的模块、功能,将测试资源优先分配至高风险区域,提升测试针对性,避免盲目测试,缩短测试周期。

3.5 在维护阶段的具体应用

维护阶段是软件工程的后续保障环节,核心是解决

软件运行问题、优化性能,数据挖掘技术主要应用于故障排查、维护需求梳理,具体应用细节如下:(1)系统故障快速排查,采集软件运行日志、故障记录及CPU使用率、内存占用、接口响应时间等监控数据,采用DBSCAN等异常检测算法,识别系统运行异常数据,标记内存骤升、接口超时等故障征兆;通过关联规则挖掘分析故障与运行参数、用户操作的关联,快速定位原因。(2)维护需求分类与优先级排序,采集用户反馈、故障记录、软件性能数据,利用聚类分析技术将维护需求分为故障修复、性能优化、功能升级三类;结合紧急程度、用户关注度,采用分类算法排序,辅助维护人员合理安排工作顺序。(3)潜在维护需求预测,基于历史维护数据、软件运行数据,采用分类与预测算法,预测未来可能出现的故障及需优化环节,提前制定维护计划,实现主动维护,减少故障对软件运行的影响^[5]。

结束语:本文围绕计算机软件工程中的数据挖掘技术应用展开全面研究,明确了应用前提与数据准备要点,细化了四大核心挖掘技术的选型与实现细节,梳理了其在软件工程各阶段的具体应用路径。研究结合实践场景,验证了数据挖掘技术对优化工程流程、提升软件质量的实际效用。未来可进一步优化算法适配性、降低应用成本,推动该技术在更多软件工程场景中落地,助力软件行业高质量、智能化发展。

参考文献

- [1]陈荣宝.计算机软件工程中的数据挖掘技术应用研究[J].软件,2025,46(7):97-99.
- [2]阿迪娅·扎曼别克.计算机软件工程中的数据挖掘技术应用研究[J].电子技术与软件工程,2022(10):183-186.
- [3]吕学进.计算机软件工程中数据挖掘技术应用[J].微型计算机,2026(1):157-159.
- [4]唐宁,金斌彬,潘玉.数据挖掘技术在计算机软件工程中的应用研究[J].中文科技期刊数据库(全文版)工程技术,2025(4):153-156.
- [5]李娜.数据挖掘技术在计算机软件工程中的应用分析[J].信息记录材料,2025,26(8):89-91.