

# 实时欺诈检测系统软件测试覆盖报告

## 一、报告概述

### 1.1 报告目的

本报告旨在详细阐述实时欺诈检测交易系统的测试覆盖情况，评估测试工作的完整性和有效性，为系统的质量评估和后续改进提供依据。

### 1.2 系统概述

实时欺诈检测交易系统基于 Spring Boot 框架开发，使用 Java 语言编写。该系统主要用于实时监测交易数据，识别可能存在的欺诈行为，并及时采取相应的防范措施。系统具备数据接收、规则引擎、模型预测、报警通知等核心功能。

### 1.3 测试范围

本次测试覆盖了系统的主要功能模块，包括交易数据接入、欺诈规则匹配、风险评分计算、报警机制等，同时对系统的性能、兼容性等方面进行了部分测试。

## 二、测试环境

### 2.1 硬件环境

硬件设备	配置信息
服务器	ecs.ul-c1m1.xlarge 4 核 (vCPU) 4 GiB, 硬盘: ESSD 云盘 PL0 20GiB (2040 IOPS)
客户端	CPU: Intel Core i5 - 8400 @ 2.80GHz, 内存: 16GB, 硬盘: 1024GB SSD

### 2.2 软件环境

软件名称	版本信息
------	------

软件名称	版本信息
操作系统	服务器：Linux CentOS 8，客户端：Windows 10
数据库	MySQL 8.0
Java 运行环境	JDK 18
Spring Boot	2.2.5

## 三、测试方法与工具

### 3.1 测试方法

- 功能测试：**采用黑盒测试方法，依据系统需求规格说明书，对系统的各项功能进行验证，确保功能的正确性和完整性。
- 单元测试：**使用 JUnit 5 框架对系统中的各个独立单元（如服务层、DAO 层的方法）进行测试，保证单元代码的正确性。
- 集成测试：**将系统的各个模块进行集成，测试模块之间的接口和交互是否正常。
- 性能测试：**使用 JMeter 工具模拟大量并发用户请求，对系统的响应时间、吞吐量等性能指标进行测试。

### 3.2 测试工具

工具名称	用途
junit-jupiter	单元测试
Mockito	单元测试中的模拟对象
SpringBoot-Test	集成测试
JMeter	性能测试
Jacoco-maven-plugin	代码覆盖率测试

## 四、测试覆盖情况

4.1 功能测试覆盖

功能模块	测试用例数量	通过数量	通过率	覆盖情况说明
交易数据接入	20	20	100%	对多种格式和来源的交易数据接入进行了测试，大部分正常接入，但仍有部分特殊格式数据存在解析问题。
欺诈规则匹配	30	30	100%	涵盖了系统中预设的各类欺诈规则匹配测试，部分复杂规则在边界条件下存在匹配不准确的情况。
报警机制	10	10	100%	对邮件、日志等报警方式进行了测试，部分报警信息存在格式问题。

4.2 代码覆盖率

使用 JaCoCo 工具对系统代码进行覆盖率分析，结果如下：

覆盖率指标	覆盖率	详细说明
行覆盖率	80%	大部分业务逻辑代码被执行到，但部分异常处理和边缘情况的代码未被覆盖。
分支覆盖率	75%	条件判断语句的分支覆盖情况一般，一些复杂的逻辑分支未被完全覆盖。
类覆盖率	85%	大部分类被测试到，但部分工具类和配置类未被充分测试。

4.3 性能测试覆盖

性能指标	测试情况	达标情况
响应时间	在并发用户数为 1000 的情况下，平均响应时间为 500ms，最大响应时间为 1000ms。	满足系统设计要求（平均响应时间 ≤ 800ms，最大响应时间 ≤ 1500ms）

性能指标	测试情况	达标情况
吞吐量	并发用户数为 1000 时，系统吞吐量达到 2000 笔 / 秒。	满足系统设计要求（吞吐量 $\geq 1500$ 笔 / 秒）

## 五、未覆盖情况分析

### 5.1 功能未覆盖情况

- 部分特殊业务场景的交易数据处理未进行测试，如跨境交易中涉及多种货币和复杂汇率的情况。
- 系统与第三方支付接口的异常处理和兼容性测试不够全面。

### 5.2 代码未覆盖情况

- 部分代码逻辑由于测试数据难以构造，导致无法完全覆盖，如一些复杂的异常处理和边界条件。
- 部分工具类和配置类由于缺乏有效的测试入口，未被充分测试。

## 六、测试结论

### 6.1 测试总体评价

本次测试对实时欺诈检测交易系统的主要功能和性能进行了较为全面的覆盖，系统在大部分功能和性能指标上表现良好，但仍存在一些问题需要解决。功能测试中发现了部分功能的缺陷，代码覆盖率虽达到一定水平但仍有提升空间，性能测试基本满足系统设计要求。

### 6.2 建议

- 针对功能测试中发现的问题，及时修复并进行回归测试，确保系统功能的正确性和稳定性。
- 进一步完善测试数据，覆盖更多的特殊业务场景和边界条件，提高代码覆盖率。
- 加强对系统与第三方接口的兼容性测试，确保系统在各种环境下的正常运行。

## 七、附件

- 测试用例文档
- 代码覆盖率报告（JaCoCo 生成的 HTML 报告）
- 性能测试报告（JMeter 生成的测试结果）

以上报告阶段性产物，投产需要进一步完善，仅供参考。