

标记	数量	修改单号	签字	日期

Zero-Defect Prediction 软件可靠性分析系统 测试说明 V1.0

(ZDP 软件测试说明)

编制：_____

会签：_____

校对：_____

标检：_____

审核：_____

批准：_____

会签：_____

ZDP 软件项目组

目 次

1 范围 4

2 测试准备 4

2.1 硬件准备 4

2.2 软件准备 4

3 测试说明 4

3.1 功能测试 (GN) 5

3.1.1 数据导入功能测试 (GN_01) 5

3.1.2 模型拟合与参数估计测试 (GN_02) 5

3.1.3 可视化图表生成测试 (GN_03) 6

3.1.4 报告导出功能测试 (GN_04) 6

3.1.5 实验管理功能测试 (GN_05) 6

3.2 接口测试 (JK) 7

3.2.1 模型插件接口测试 (JK_01) 7

3.2.2 命令行接口测试 (JK_02) 7

3.3 边界测试 (BJ) 7

3.3.1 数据边界测试 (BJ_01) 7

3.3.2 模型参数边界测试 (BJ_02) 8

3.4 性能测试 (XN) 8

3.4.1 多模型并发执行性能测试 (XN_01) 8

3.4.2 大数据集处理性能测试 (XN_02) 8

3.5 安全性测试 (AQ) 9

3.5.1 输入数据安全性测试 (AQ_01) 9

3.6 兼容性测试 (JR) 9

3.6.1 操作系统兼容性测试 (JR_01) 9

4 附录 A: 功能测试用例集 10

4.1 A.1 数据导入功能测试用例 10

4.2 A.2 模型拟合功能测试用例 13

4.3 A.3 可视化图表生成测试用例 17

4.4 A.4 报告导出功能测试用例 19

4.5 A.5 实验管理功能测试用例 21

5 附录 B: 接口测试用例集 22

5.1 B.1 插件模型接口测试用例 22

5.2 B.2 命令行接口测试用例 23

6 附录 C: 边界测试用例集 24

6.1	C.1 数据边界测试用例	24
6.2	C.2 模型参数边界测试用例	26
7	附录 D: 性能测试用例集	28
7.1	D.1 并发性能测试用例	28
7.2	D.2 大数据集性能测试用例	29
8	附录 E: 安全性测试用例集	30
8.1	E.1 输入数据安全测试用例	30
9	附录 F: 兼容性测试用例集	32
9.1	F.1 操作系统兼容性测试用例	32
10	附录 G: 额外测试用例	34
10.1	G.1 可用性测试	34
10.2	G.2 回归测试	35
10.3	G.3 文档审查测试用例	36
11	附录 H: 测试数据集说明	38
11.1	数据集特征说明	38
12	附录 I: 测试环境配置清单	39
12.1	测试工具配置	39
12.2	测试环境硬件配置	39
13	测试用例统计汇总	40
13.1	测试覆盖率目标	40
13.2	测试执行策略	40

1 范围

本文档适用于 ZDP（Zero-Defect Prediction，零缺陷预测）软件可靠性分析系统，包含桌面端 GUI 和命令行 CLI 两种运行模式。

2 测试准备

2.1 硬件准备

序号	硬件名称	规格	设备编号
1	测试工作站	PC 机/Windows 10 64 位，CPU≥4 核，内存 ≥8GB	PC-TEST-001
2	备用测试机	PC 机/Windows 11 64 位，CPU≥4 核，内存 ≥16GB	PC-TEST-002

表 1 ZDP 软件可靠性分析系统需求硬件项

2.2 软件准备

序号	软件项名称	版本号
1	Windows 10/11 Professional	21H2/22H2
2	Python	3.10.x - 3.12.x
3	uv 包管理器	0.5.x
4	ZDP 测试版本	0.1.0
5	pytest 测试框架	8.2.x
6	Microsoft Excel / WPS	2016+

表 2 ZDP 软件可靠性分析系统的软件项

3 测试说明

本次测试设计共设计 126 个测试用例，详细测试用例设计见附录。

3.1 功能测试（GN）

3.1.1 数据导入功能测试（GN_01）

测试项名称	数据导入功能
测试项标识	GN_01
测试项描述	该测试项主要完成 CSV/Excel 数据文件的导入功能测试，包括故障数据类型自动推断（累计故障数 vs 故障间隔 TBF）、时间列可选导入、数据格式验证等。支持的文件格式：.csv, .xlsx, .xls。
测试方法	等价类划分法、边界值分析法、猜错法。
测试用例设计	见附录 A.1

表 3 数据导入功能测试

3.1.2 模型拟合与参数估计测试（GN_02）

测试项名称	可靠性模型拟合
测试项标识	GN_02
测试项描述	验证软件支持的 8 种可靠性模型的参数估计准确性和收敛性： 1. Goel-Okumoto 模型（GO） 2. Jelinski-Moranda 模型（JM） 3. S-Shaped 模型（S-Shaped） 4. BP 神经网络模型（BP） 5. 支持向量回归模型（SVR） 6. EMD-SVR/GM 混合模型（Hybrid） 7. GM(1,1) 灰色预测模型（GM） 8. 第三方插件模型（Plugin）
测试方法	基准测试法、误差分析法。使用标准测试数据集验证模型输出结果与理论值的偏差在可接受范围内（RMSE < 5%）。
测试用例设计	见附录 A.2

表 4 模型拟合功能测试

3.1.3 可视化图表生成测试 (GN_03)

测试项名称	图表生成与展示
测试项标识	GN_03
测试项描述	验证系统生成以下 4 类可视化图表的完整性和准确性： 1. 预测曲线对比图（显示所有成功运行的模型） 2. 残差分析图（可选择特定模型） 3. U 图（拉普拉斯检验统计量） 4. Y 图（算术平均检验统计量）
测试方法	视觉验证法、像素对比法、数据一致性检查。
测试用例设计	见附录 A.3

表 5 可视化功能测试

3.1.4 报告导出功能测试 (GN_04)

测试项名称	PDF 报告导出
测试项标识	GN_04
测试项描述	验证系统能够导出包含中文内容的 PDF 测试报告，报告应包含：数据概览、模型排名表、拟合参数表、预测曲线图、统计检验结果。中文字体自动注册（微软雅黑/宋体/黑体），避免乱码。
测试方法	输出验证法、文件完整性检查。
测试用例设计	见附录 A.4

表 6 报告导出功能测试

3.1.5 实验管理功能测试 (GN_05)

测试项名称	实验导出/导入
测试项标识	GN_05
测试项描述	验证实验结果可以导出为 ZIP 压缩包（包含原始数据 CSV、配置 JSON、排名结果汇总），并支持在不同机器上重新导入以复现分析过程。
测试方法	数据完整性验证、跨平台兼容性测试。
测试用例设计	见附录 A.5

表 7 实验管理功能测试

3.2 接口测试（JK）

3.2.1 模型插件接口测试（JK_01）

测试项名称	第三方模型插件加载
测试项标识	JK_01
测试项描述	验证 ZDP 系统通过 Python entry_points 机制加载第三方模型插件的能力。插件需实现 ReliabilityModel 基类接口，支持 fit(), predict(), clone() 等方法。
测试方法	集成测试法、接口契约验证。
测试用例设计	见附录 B.1

表 8 插件模型接口测试

3.2.2 命令行接口测试（JK_02）

测试项名称	zdp-cli 命令行工具
测试项标识	JK_02
测试项描述	验证 CLI 工具的参数解析、数据加载、模型运行、结果输出等功能。测试场景包括：指定模型运行、批量数据处理、报告自动生成。
测试方法	黑盒测试法、命令行自动化测试。
测试用例设计	见附录 B.2

表 9 CLI 命令行接口测试

3.3 边界测试（BJ）

3.3.1 数据边界测试（BJ_01）

测试项名称	异常数据处理
测试项标识	BJ_01
测试项描述	验证系统对以下边界情况的鲁棒性： 1. 空数据集 2. 单点数据（N=1） 3. 极小数据集（N=2 3） 4. 超大数据集（N>10000） 5. 缺失值处理 6. 非单调累计故障数 7. 负数故障间隔 8. 浮点数精度边界
测试方法	边界值分析法、异常路径测试。
测试用例设计	见附录 C.1

表 10 数据边界处理测试

3.3.2 模型参数边界测试 (BJ_02)

测试项名称	模型配置参数边界
测试项标识	BJ_02
测试项描述	验证各模型在极端参数配置下的表现： <ul style="list-style-type: none">BP 神经网络：隐藏层神经元数 [1, 200]，学习率 [1e-5, 1.0]SVR：C 参数 [0.1, 1000]，epsilon [1e-4, 1.0]Hybrid：分解层数 [1, 10]，平滑度 [0.0, 1.0]
测试方法	参数敏感性分析、极值测试。
测试用例设计	见附录 C.2

表 11 模型参数边界测试

3.4 性能测试 (XN)

3.4.1 多模型并发执行性能测试 (XN_01)

测试项名称	多线程模型并发
测试项标识	XN_01
测试项描述	验证 GUI 使用 QThread 异步执行模型分析时的响应性能和线程安全性。测试同时运行 8 个模型时的耗时、CPU 占用率、内存占用。
测试方法	性能分析法、资源监控。
测试用例设计	见附录 D.1

表 12 并发性能测试

3.4.2 大数据集处理性能测试 (XN_02)

测试项名称	海量数据处理
测试项标识	XN_02
测试项描述	验证系统处理超大数据集（1 万 10 万故障点）时的响应时间和内存消耗。重点测试：数据加载时间 < 5 秒，模型拟合时间 < 60 秒，可视化渲染时间 < 10 秒。
测试方法	负载测试法、压力测试。
测试用例设计	见附录 D.2

表 13 大数据集性能测试

3.5 安全性测试 (AQ)

3.5.1 输入数据安全性测试 (AQ_01)

测试项名称	恶意输入防护
测试项标识	AQ_01
测试项描述	验证系统对恶意构造的数据文件的防护能力：特殊字符注入、超长字段、畸形 CSV/Excel、编码攻击等。确保不会导致程序崩溃或执行非预期代码。
测试方法	模糊测试法、异常注入。
测试用例设计	见附录 E.1

表 14 数据安全验证测试

3.6 兼容性测试 (JR)

3.6.1 操作系统兼容性测试 (JR_01)

测试项名称	Windows 多版本兼容
测试项标识	JR_01
测试项描述	验证 ZDP 在不同 Windows 版本上的兼容性：Windows 10 (1809+), Windows 11 (21H2+)。包括 PyInstaller 打包的单个文件 exe 可执行程序的运行测试。
测试方法	多环境验证测试。
测试用例设计	见附录 F.1

表 15 跨平台兼容性测试

4 附录 A：功能测试用例集

4.1 A.1 数据导入功能测试用例

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	GN_01_001		
测试用例名称	导入标准 CSV 累计故障数文件		
测试类型	功能测试		
测试说明	验证系统能正确导入包含时间列和累计故障数列的 CSV 文件，并自动识别为累计故障数类型		
序号	输入步骤	预期输出	实际输出
1	选择文件 nhpp_goel_okumoto.csv, 指定时间列 t, 值列 failures	成功加载，数据集类型 显示为“累计故障数”， 数据点数=50	
2	点击“运行分析”按钮， 选择 GO 模型	模型成功拟合，参数 a≈100, b≈0.02	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	GN_01_002		
测试用例名称	导入故障间隔 TBF 数据文件		
测试类型	功能测试		
测试说明	验证系统能正确识别故障间隔数据并转换为累计故障数		
序号	输入步骤	预期输出	实际输出
1	选择文件 tbf_jm_synthetic.csv, 指定时间列 interval, 值列 tbf	成功加载，数据集类型 显示为“故障间隔 (TBF)”，自动推断数据 类型正确	
2	查看数据预览面板	显示 TBF 原始数据，同 时提供累计故障数视图	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	GN_01_003		
测试用例名称	导入无时间列的数据文件		
测试类型	功能测试		
测试说明	验证系统在缺少时间列时自动生成序号作为时间轴		
序号	输入步骤	预期输出	实际输出
1	准备只有单列故障数的 CSV 文件，不指定时间列	成功加载，时间轴自动赋值为 1, 2, 3, ..., N	
2	在图表中查看横坐标	横坐标显示为整数序号	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	GN_01_004		
测试用例名称	导入 Excel 格式文件		
测试类型	功能测试		
测试说明	验证系统支持 .xlsx 和 .xls 格式文件的导入		
序号	输入步骤	预期输出	实际输出
1	准备包含多个工作表的 Excel 文件，选择目标工作表	成功解析并加载指定工作表的数据	
2	验证数据完整性	数据行数、列名与 Excel 中一致	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	GN_01_005		
测试用例名称	导入包含缺失值的数据文件		
测试类型	边界测试		
测试说明	验证系统对包含 NaN 或空白单元格的数据的处理		
序号	输入步骤	预期输出	实际输出
1	准备含有 3 个缺失值的 CSV 文件	系统弹出警告提示，提示缺失值位置	
2	选择“跳过缺失行”或“填充为 0”选项	数据预处理后成功加载，缺失值按选项处理	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	GN_01_006		
测试用例名称	导入非单调递增的累计故障数		
测试类型	异常测试		
测试说明	验证系统对非法累计故障数序列的检测能力		
序号	输入步骤	预期输出	实际输出
1	准备累计故障数中存在回退的数据文件（如：10, 15, 12, 20）	系统检测到非单调性，弹出错误提示：“累计故障数必须单调非递减”	
2	用户选择“转换为 TBF 模式”	系统将数据转换为故障间隔模式并成功加载	

4.2 A.2 模型拟合功能测试用例

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	GN_02_001		
测试用例名称	Goel-Okumoto 模型拟合测试		
测试类型	功能测试		
测试说明	验证 GO 模型在标准 NHPP 数据上的拟合精度		
序号	输入步骤	预期输出	实际输出
1	加载 nhpp_goel_okumoto.csv, 运行 GO 模型	参数 $a=99.8 \pm 1$, $b=0.0205 \pm 0.001$, RMSE < 3.0	
2	查看预测曲线	拟合曲线平滑通过数据 点, 残差随机分布	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	GN_02_002		
测试用例名称	Jelinski-Moranda 模型拟合测试		
测试类型	功能测试		
测试说明	验证 JM 模型在 TBF 数据上的表现		
序号	输入步骤	预期输出	实际输出
1	加载 tbf_jm_synthetic.csv, 运行 JM 模型	参数 $N_0 \approx 50$, $\phi \approx 0.002$, 模型收敛成 功	
2	对比 GO 模型排名	JM 模型在 TBF 数据上 RMSE 优于 GO 模型	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	GN_02_003		
测试用例名称	S-Shaped 模型拟合测试		
测试类型	功能测试		
测试说明	验证 S-Shaped 模型对 S 型故障累计曲线的拟合能力		
序号	输入步骤	预期输出	实际输出
1	加载 <code>tbf_s_shaped.csv</code> , 运行 S-Shaped 模型	参数估计成功, 曲线呈现 S 型特征	
2	验证早期阶段斜率	早期故障发现率较低, 中期加速, 后期饱和	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	GN_02_004		
测试用例名称	BP 神经网络模型训练测试		
测试类型	功能测试		
测试说明	验证 BP 模型的训练收敛性和预测精度		
序号	输入步骤	预期输出	实际输出
1	配置隐藏层神经元数=10, 学习率=0.01, 训练轮数=500	训练过程无异常, 损失函数单调下降	
2	运行 BP 模型拟合	仅对累计故障数类型数据运行成功, TBF 数据自动跳过	
3	验证预测精度	BP 模型 RMSE 在前 3 名内	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	GN_02_005		
测试用例名称	SVR 支持向量回归模型测试		
测试类型	功能测试		
测试说明	验证 SVR 模型的参数优化和泛化能力		
序号	输入步骤	预期输出	实际输出
1	配置 C=100, epsilon=0.1, kernel= 'rbf'	参数配置成功，模型开始训练	
2	运行 SVR 模型	仅对累计故障数数据运行，训练时间 < 30 秒	
3	验证非线性拟合能力	SVR 能捕捉复杂的非线性故障增长趋势	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	GN_02_006		
测试用例名称	EMD-SVR/GM 混合模型测试		
测试类型	功能测试		
测试说明	验证混合模型的分解-重构-预测能力		
序号	输入步骤	预期输出	实际输出
1	配置 EMD 分解层数=5, 平滑度=0.7	模型成功分解数据为 IMF 分量	
2	运行混合模型	对高频分量使用 SVR, 低频分量使用 GM 平滑	
3	验证预测精度	混合模型在复杂数据上 RMSE 优于单一模型	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	GN_02_007		
测试用例名称	GM(1, 1) 灰色预测模型测试		
测试类型	功能测试		
测试说明	验证 GM 模型的独立运行能力		
序号	输入步骤	预期输出	实际输出
1	加载累计故障数数据， 选择 GM 模型	模型成功执行灰色累加 生成，计算发展系数	
2	验证预测趋势	GM 模型擅长捕捉指数增 长趋势	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	GN_02_008		
测试用例名称	模型自动排名测试		
测试类型	功能测试		
测试说明	验证系统按 RMSE 升序自动排名所有模型		
序号	输入步骤	预期输出	实际输出
1	同时运行所有 8 个模型	所有兼容模型成功运行， 不兼容模型自动跳过	
2	查看排名表	表格按 RMSE 升序显示， 第 1 名模型标记为“最 佳模型”	
3	验证指标一致性	RMSE, MAE, R ² 等指标 与手工计算结果一致	

4.3 A.3 可视化图表生成测试用例

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	GN_03_001		
测试用例名称	预测曲线对比图测试		
测试类型	功能测试		
测试说明	验证系统生成包含所有成功模型的预测曲线对比图		
序号	输入步骤	预期输出	实际输出
1	运行 5 个成功的模型	预测曲线图显示 5 条不同颜色的曲线	
2	验证图例完整性	图例包含所有模型名称和对应颜色	
3	验证坐标轴标签	横轴：“时间 / Time”，纵轴：“累计故障数 / Cumulative Failures”	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	GN_03_002		
测试用例名称	残差分析图测试		
测试类型	功能测试		
测试说明	验证残差图的正确性和模型选择功能		
序号	输入步骤	预期输出	实际输出
1	在“图表模型”下拉框选择 G0 模型	残差图切换为 G0 模型的残差分布	
2	验证残差计算	残差 = 实际值 - 预测值，零线清晰显示	
3	切换到 BP 模型	残差图实时更新为 BP 模型的残差	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	GN_03_003		
测试用例名称	U 图（拉普拉斯检验）测试		
测试类型	功能测试		
测试说明	验证 U 图统计量的计算和可视化		
序号	输入步骤	预期输出	实际输出
1	选择最佳模型，生成 U 图	U 统计量曲线位于 $[-2, 2]$ 区间内表示模型稳定	
2	验证置信区间标注	图中标注 ± 2 标准差置信线	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	GN_03_004		
测试用例名称	Y 图（算术平均检验）测试		
测试类型	功能测试		
测试说明	验证 Y 图的正确性		
序号	输入步骤	预期输出	实际输出
1	生成 Y 图	Y 统计量围绕 0 线波动	
2	验证趋势分析	Y 值持续偏离 0 表示模型存在系统性偏差	

4. 4 A. 4 报告导出功能测试用例

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	GN_04_001		
测试用例名称	PDF 报告完整性测试		
测试类型	功能测试		
测试说明	验证导出的 PDF 报告包含所有必要内容		
序号	输入步骤	预期输出	实际输出
1	运行完整分析后，点击“导出 PDF 报告”	生成 PDF 文件，大小约 2-5 MB	
2	打开 PDF，检查章节结构	包含：封面、数据概览、模型排名表、参数表、图表、结论	
3	验证中文显示	所有中文文本正常显示，无乱码	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	GN_04_002		
测试用例名称	报告图表嵌入测试		
测试类型	功能测试		
测试说明	验证图表以 PNG 格式正确嵌入 PDF		
序号	输入步骤	预期输出	实际输出
1	检查 PDF 中的预测曲线图	图表清晰，分辨率≥300 DPI	
2	验证图表标题和坐标轴	标题、坐标轴标签完整显示	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	GN_04_003		
测试用例名称	多页报告生成测试		
测试类型	功能测试		
测试说明	验证长报告的自动分页功能		
序号	输入步骤	预期输出	实际输出
1	运行 8 个模型，导出报告	报告自动分页，每页布局合理	
2	验证页码	页脚显示页码：“第 X 页 / 共 Y 页”	

4.5 A.5 实验管理功能测试用例

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	GN_05_001		
测试用例名称	实验导出 ZIP 文件测试		
测试类型	功能测试		
测试说明	验证实验结果导出为 ZIP 压缩包的完整性		
序号	输入步骤	预期输出	实际输出
1	运行分析后，点击“导出实验”，选择保存路径	生成 .zip 文件，文件名包含时间戳	
2	解压缩 ZIP 文件	包含 3 个文件： dataset.csv, config.json, results.json	
3	验证文件内容	dataset.csv 为原始数据，config.json 包含模型配置， results.json 包含排名结果	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	GN_05_002		
测试用例名称	实验导入重现测试		
测试类型	功能测试		
测试说明	验证导入实验 ZIP 文件后能重现分析过程		
序号	输入步骤	预期输出	实际输出
1	点击“导入实验”，选择之前导出的 ZIP 文件	系统成功解压并加载实验数据	
2	验证数据一致性	加载的数据集与原始数据完全一致	
3	重新运行分析	模型排名和参数估计结果与原始实验一致	

5 附录 B：接口测试用例集

5.1 B.1 插件模型接口测试用例

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	JK_01_001		
测试用例名称	第三方插件模型加载测试		
测试类型	接口测试		
测试说明	验证通过 <code>entry_points</code> 加载外部模型的能力		
序号	输入步骤	预期输出	实际输出
1	创建测试插件包，在 <code>pyproject.toml</code> 中注册 <code>entry_point</code>	插件成功注册到 <code>zdp.models</code> 命名空间	
2	在 CLI 中运行 <code>zdp-cli data.csv --model my_plugin</code>	插件模型被正确加载并执行拟合	
3	验证插件结果	插件模型返回的 <code>ModelResult</code> 对象包含所需字段	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	JK_01_002		
测试用例名称	插件接口契约验证测试		
测试类型	接口测试		
测试说明	验证插件必须实现的接口方法		
序号	输入步骤	预期输出	实际输出
1	创建缺少 <code>fit()</code> 方法的插件	加载时抛出 <code>TypeError</code> 异常	
2	创建缺少 <code>clone()</code> 方法的插件	多线程执行时抛出错误	
3	创建完整实现的插件	插件正常加载和运行	

5.2 B.2 命令行接口测试用例

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	JK_02_001		
测试用例名称	CLI 基本参数解析测试		
测试类型	接口测试		
测试说明	验证 zdp-cli 命令行参数解析的正确性		
序号	输入步骤	预期输出	实际输出
1	运行 <code>zdp-cli data.csv --time-column t --value-column failures</code>	成功加载数据，列名解析正确	
2	运行 <code>zdp-cli data.csv --model go --model jm</code>	仅运行指定的 GO 和 JM 模型	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	JK_02_002		
测试用例名称	CLI 报告导出参数测试		
测试类型	接口测试		
测试说明	验证 <code>--report</code> 参数生成 PDF 报告		
序号	输入步骤	预期输出	实际输出
1	运行 <code>zdp-cli data.csv --report output.pdf</code>	分析完成后自动生成 <code>output.pdf</code> 文件	
2	验证文件存在	<code>output.pdf</code> 文件成功创建在当前目录	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	JK_02_003		
测试用例名称	CLI 错误处理测试		
测试类型	异常测试		
测试说明	验证 CLI 对错误输入的处理		
序号	输入步骤	预期输出	实际输出
1	运行 <code>zdp-cli non_existent.csv</code>	输出错误信息：“文件未找到”，退出码=1	
2	运行 <code>zdp-cli data.csv --model invalid_model</code>	输出错误信息：“未知模型”，列出可用模型	

6 附录 C：边界测试用例集

6.1 C.1 数据边界测试用例

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	BJ_01_001		
测试用例名称	空数据集测试		
测试类型	边界测试		
测试说明	验证系统对空数据集的处理		
序号	输入步骤	预期输出	实际输出
1	尝试加载空 CSV 文件 (0 行数据)	系统弹出错误提示：“数据集为空，至少需要 2 个数据点”	
2	验证系统状态	系统保持稳定，不崩溃	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	BJ_01_002		
测试用例名称	单点数据测试		
测试类型	边界测试		
测试说明	验证系统对单点数据的处理		
序号	输入步骤	预期输出	实际输出
1	加载仅包含 1 个数据点的文件	系统提示：“数据点过少 (N=1)，无法进行可靠性分析”	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	BJ_01_003		
测试用例名称	极小数据集测试（N=2~3）		
测试类型	边界测试		
测试说明	验证系统在极小数据集上的行为		
序号	输入步骤	预期输出	实际输出
1	加载 N=2 的数据集	系统允许加载，但警告： “数据点过少，模型拟合 结果可能不可靠”	
2	运行 G0 模型	模型尝试拟合，但可能 无法收敛或 R ² 很低	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	BJ_01_004		
测试用例名称	超大数据集测试（N>10000）		
测试类型	性能测试		
测试说明	验证系统处理超大数据集的能力		
序号	输入步骤	预期输出	实际输出
1	生成 N=15000 的故障数据	数据加载时间 < 5 秒	
2	运行所有模型	所有模型在 120 秒内完成拟合	
3	验证内存占用	峰值内存 < 2GB	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	BJ_01_005		
测试用例名称	负数故障间隔测试		
测试类型	异常测试		
测试说明	验证系统对负数 TBF 的检测		
序号	输入步骤	预期输出	实际输出
1	准备包含负数 TBF 的数据文件（如：-5，10，15）	系统检测到异常，弹出错误：“故障间隔不能为负数”	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	BJ_01_006		
测试用例名称	浮点数精度边界测试		
测试类型	边界测试		
测试说明	验证系统对极小/极大浮点数的处理		
序号	输入步骤	预期输出	实际输出
1	准备包含 $1e-10$ 和 $1e10$ 数量级的数据	数据成功加载，无精度溢出错误	
2	运行模型拟合	模型能正确处理不同数量级的数据	

6.2 C.2 模型参数边界测试用例

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	BJ_02_001		
测试用例名称	BP 神经网络极端参数测试		
测试类型	边界测试		
测试说明	验证 BP 模型在极端参数配置下的稳定性		
序号	输入步骤	预期输出	实际输出
1	配置隐藏层神经元数=1（最小值）	模型可运行，但拟合能力受限，RMSE 较高	
2	配置隐藏层神经元数=200（最大值）	模型训练时间显著增加，但不崩溃	
3	配置学习率= $1e-5$ （极小值）	训练收敛极慢，可能需要增加训练轮数	
4	配置学习率=1.0（极大值）	训练可能震荡，损失函数不收敛	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	BJ_02_002		
测试用例名称	SVR 参数边界测试		
测试类型	边界测试		
测试说明	验证 SVR 模型在极端参数下的表现		
序号	输入步骤	预期输出	实际输出
1	配置 C=0.1（最小惩罚系数）	模型欠拟合，预测精度低	
2	配置 C=1000（最大惩罚系数）	模型可能过拟合，训练数据拟合完美但泛化差	
3	配置 epsilon=1e-4（极小容忍度）	模型对噪声敏感	
4	配置 epsilon=1.0（极大容忍度）	模型过于平滑，丢失细节	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	BJ_02_003		
测试用例名称	Hybrid 模型分解参数边界测试		
测试类型	边界测试		
测试说明	验证混合模型在极端分解参数下的稳定性		
序号	输入步骤	预期输出	实际输出
1	配置 EMD 分解层数=1（最小值）	模型退化为近似单一模型	
2	配置 EMD 分解层数=10（最大值）	分解计算时间增加，但不崩溃	
3	配置平滑度=0.0（无平滑）	模型保留所有高频噪声	
4	配置平滑度=1.0（完全平滑）	模型过度平滑，丢失有用信息	

7 附录 D：性能测试用例集

7.1 D.1 并发性能测试用例

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	XN_01_001		
测试用例名称	GUI 多线程并发测试		
测试类型	性能测试		
测试说明	验证 GUI 使用 QThread 异步执行的响应性		
序号	输入步骤	预期输出	实际输出
1	在 GUI 中加载数据，勾选全部 8 个模型，点击“运行分析”	GUI 界面保持响应，不卡死，进度条实时更新	
2	在分析运行期间尝试操作其他界面元素	能够最小化/最大化窗口，但分析按钮显示为“运行中...”不可再次点击	
3	验证并发执行时间	8 个模型并发执行总耗时 < 45 秒（单线程约 120 秒）	
4	监控 CPU 占用率	CPU 使用率达到 70-90%，充分利用多核	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	XN_01_002		
测试用例名称	线程安全性测试		
测试类型	性能测试		
测试说明	验证多线程执行时的数据一致性和无竞态条件		
序号	输入步骤	预期输出	实际输出
1	连续 10 次运行相同数据和模型组合	每次结果完全一致，参数误差 < 1e-6	
2	验证模型 clone() 机制	每个线程使用独立的模型实例，无状态共享	

7.2D.2 大数据集性能测试用例

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	XN_02_001		
测试用例名称	10000 点数据集性能测试		
测试类型	性能测试		
测试说明	验证系统处理 1 万故障点的性能		
序号	输入步骤	预期输出	实际输出
1	生成 N=10000 的故障数据，加载到系统	加载时间 < 3 秒，内存占用 < 500MB	
2	运行 GO 模型	拟合时间 < 5 秒	
3	运行 BP 模型（最耗时）	训练时间 < 60 秒	
4	生成预测曲线图	matplotlib 渲染时间 < 8 秒	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	XN_02_002		
测试用例名称	100000 点超大数据集极限测试		
测试类型	压力测试		
测试说明	测试系统在极限数据规模下的表现		
序号	输入步骤	预期输出	实际输出
1	生成 N=100000 的故障数据	数据生成完成，文件大小约 5MB	
2	尝试加载数据	加载时间 < 15 秒，或弹出警告：“数据集过大，建议采样”	
3	如果加载成功，运行简单模型（GO）	拟合时间 < 30 秒，或提示用户采样后分析	
4	监控峰值内存	峰值内存 < 4GB（64 位系统）	

8 附录 E：安全性测试用例集

8.1 E.1 输入数据安全测试用例

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	AQ_01_001		
测试用例名称	特殊字符注入测试		
测试类型	安全测试		
测试说明	验证系统对包含特殊字符的数据文件的防护		
序号	输入步骤	预期输出	实际输出
1	准备包含 SQL 注入式字符串的 CSV（如：'; DROP TABLE--）	系统作为纯文本处理，不执行任何代码	
2	准备包含 XSS 脚本的数据（如：<script>alert('xss')</script>）	系统忽略或转义 HTML 标签，GUI 显示为纯文本	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	AQ_01_002		
测试用例名称	超长字段测试		
测试类型	安全测试		
测试说明	验证系统对超长字符串字段的处理		
序号	输入步骤	预期输出	实际输出
1	准备包含 10000 字符长度列名的 CSV	系统截断或拒绝加载，弹出警告：“列名过长”	
2	准备包含 1MB 单个单元格数据的文件	系统拒绝加载该行，或限制单元格大小	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	AQ_01_003		
测试用例名称	畸形文件测试		
测试类型	安全测试		
测试说明	验证系统对损坏或畸形文件的容错性		
序号	输入步骤	预期输出	实际输出
1	准备列数不一致的 CSV（第 1 行 3 列，第 2 行 5 列）	系统检测到格式错误，提示：“CSV 格式不规范”	
2	准备损坏的 Excel 文件（二进制篡改）	pandas 抛出异常，系统捕获并提示：“文件损坏，无法读取”	
3	准备非 UTF-8 编码的 CSV（GBK 编码）	系统自动检测编码并转换，或提示用户选择编码	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	AQ_01_004		
测试用例名称	路径遍历攻击测试		
测试类型	安全测试		
测试说明	验证系统对文件路径的安全处理		
序号	输入步骤	预期输出	实际输出
1	尝试在文件名中包含 ../ 等路径遍历符号	系统规范化路径，阻止访问父目录	
2	尝试访问系统敏感文件（如 /etc/passwd）	系统限制只能访问用户指定的工作目录	

9 附录 F：兼容性测试用例集

9.1 F.1 操作系统兼容性测试用例

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	JR_01_001		
测试用例名称	Windows 10 兼容性测试		
测试类型	兼容性测试		
测试说明	验证 ZDP 在 Windows 10 各版本上的运行情况		
序号	输入步骤	预期输出	实际输出
1	在 Windows 10 1809 (RS5) 上安装 Python 3.10 和依赖	依赖安装成功，无兼容性警告	
2	运行 GUI: <code>uv run zdp</code>	GUI 正常启动，PySide6 界面渲染正常	
3	运行 CLI: <code>uv run zdp-cli data.csv</code>	CLI 正常执行，输出结果正确	
4	在 Windows 10 22H2 (最新版) 重复测试	所有功能正常，性能无退化	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	JR_01_002		
测试用例名称	Windows 11 兼容性测试		
测试类型	兼容性测试		
测试说明	验证 ZDP 在 Windows 11 上的运行情况		
序号	输入步骤	预期输出	实际输出
1	在 Windows 11 21H2 上安装并运行	所有功能正常，界面适配 Windows 11 设计风格	
2	测试 DPI 缩放 (150%, 200%)	GUI 界面在高 DPI 下显示清晰，无模糊或错位	
3	在 Windows 11 22H2/23H2 重复测试	兼容性良好，无新增问题	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	JR_01_003		
测试用例名称	PyInstaller 打包 EXE 兼容性测试		
测试类型	兼容性测试		
测试说明	验证 PyInstaller 单文件 EXE 的兼容性		
序号	输入步骤	预期输出	实际输出
1	使用 <code>uv run pyinstaller --noconfirm zdp.spec</code> 打包	生成 <code>dist/zdp.exe</code> 单文件可执行程序，大小约 80-150MB	
2	在无 Python 环境的 Windows 10 机器上运行 <code>zdp.exe</code>	程序正常启动，功能完整	
3	验证多进程支持	<code>zdp_entry.py</code> 的 <code>multiprocessing.freeze_support()</code> 生效，无多进程错误	
4	在 Windows 11 上测试同一 EXE	跨版本兼容性良好	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	JR_01_004		
测试用例名称	Python 版本兼容性测试		
测试类型	兼容性测试		
测试说明	验证 ZDP 在不同 Python 版本上的运行情况		
序号	输入步骤	预期输出	实际输出
1	使用 Python 3.10 运行完整测试套件	所有测试通过	
2	使用 Python 3.11 运行	兼容性良好，性能略有提升	
3	使用 Python 3.12 运行	依赖库兼容，功能正常	
4	尝试使用 Python 3.9 运行	可能部分依赖不兼容，给出版本要求提示： <code>requires-python >= 3.10</code>	

10 附录 G：额外测试用例

10.1 G.1 可用性测试

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	KY_01_001		
测试用例名称	首次使用引导测试		
测试类型	可用性测试		
测试说明	验证新用户能快速上手		
序号	输入步骤	预期输出	实际输出
1	首次启动 GUI	显示欢迎界面或快速入门提示	
2	点击“加载示例数据”按钮	自动加载 data/samples/nhpp_goel_okumoto.csv	
3	点击“运行默认模型”按钮	自动选择常用模型（G0, JM, S-Shaped）并运行	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	KY_01_002		
测试用例名称	错误提示友好性测试		
测试类型	可用性测试		
测试说明	验证错误提示的可读性和指导性		
序号	输入步骤	预期输出	实际输出
1	加载格式错误的 CSV	错误提示清晰说明问题：“第 5 行缺少必需的列”，并建议解决方案	
2	运行不兼容的模型（TBF 数据 + BP 模型）	提示：“BP 模型仅支持累计故障数数据，已自动跳过”	

10.2 G.2 回归测试

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	HG_01_001		
测试用例名称	核心功能回归测试		
测试类型	回归测试		
测试说明	验证版本更新后核心功能无退化		
序号	输入步骤	预期输出	实际输出
1	使用基准数据集运行完整分析	模型参数与基准版本结果一致（误差 < 0.01%）	
2	对比生成的 PDF 报告	报告内容和格式与基准版本一致	
3	运行自动化测试套件： uv run pytest	所有单元测试和集成测试通过	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	HG_01_002		
测试用例名称	性能回归测试		
测试类型	回归测试		
测试说明	验证版本更新后性能无显著下降		
序号	输入步骤	预期输出	实际输出
1	使用 N=5000 的标准数据集，运行全部模型 10 次	平均执行时间与基准版本相比变化 < 5%	
2	监控内存峰值	内存占用与基准版本相比变化 < 10%	

10.3 G.3 文档审查测试用例

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	WD_01_001		
测试用例名称	用户手册完整性审查		
测试类型	文档审查		
测试说明	验证用户手册覆盖所有功能模块		
序号	输入步骤	预期输出	实际输出
1	检查手册目录结构	包含：安装说明、快速入门、功能详解、CLI 使用、API 文档	
2	验证截图的时效性	所有截图与当前版本 GUI 一致	
3	核对命令行示例	所有 CLI 示例命令能成功执行	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	WD_01_002		
测试用例名称	README 文档准确性审查		
测试类型	文档审查		
测试说明	验证 README.md 的安装和使用说明准确性		
序号	输入步骤	预期输出	实际输出
1	按照 README 中的安装步骤在全新环境安装	安装过程顺利，无遗漏步骤	
2	执行 README 中的示例命令	所有示例命令输出与文档描述一致	
3	验证依赖版本范围	pyproject.toml 中的依赖版本与 README 描述一致	

软件名称	ZDP 软件可靠性分析系统		
测试用例标识	WD_01_003		
测试用例名称	API 文档审查		
测试类型	文档审查		
测试说明	验证 API 文档与代码实现的一致性		
序号	输入步骤	预期输出	实际输出
1	检查 <code>ReliabilityModel</code> 基类文档	<code>docstring</code> 描述与实际接口一致	
2	验证公开 API 的类型注解	所有公开函数/方法包含完整类型注解	
3	运行 <code>mypy</code> 类型检查	类型检查通过，无类型错误	

11 附录 H：测试数据集说明

本测试使用以下标准数据集（位于 `data/samples/` 目录）：

序号	文件名	数据类型	数据点数
1	nhpp_goel_okumoto.csv	累计故障数 (NHPP)	50
2	tbf_jm_synthetic.csv	故障间隔 (TBF)	40
3	tbf_jm_nonmonotonic_solvable.csv	故障间隔 (TBF)，非单调	35
4	tbf_s_shaped.csv	累计故障数 (S 型曲线)	60
5	field_weekly_counts.csv	现场周故障计数	52

表 16 测试数据集清单

11.1 数据集特征说明

nhpp_goel_okumoto.csv:

- 理想的 NHPP 过程数据，故障累计数符合指数增长规律
- 适用模型：GO, S-Shaped, BP, SVR, GM, Hybrid
- 基准参数： $a \approx 100$, $b \approx 0.02$

tbf_jm_synthetic.csv:

- 合成的 Jelinski-Moranda 过程数据，故障间隔递增
- 适用模型：JM, GO, S-Shaped
- 基准参数： $N_0 \approx 50$, $\phi \approx 0.002$

tbf_s_shaped.csv:

- S 型故障发现曲线，早期发现率低，中期加速，后期饱和
- 适用模型：S-Shaped, GO, BP, SVR
- 适合测试 S-Shaped 模型的拟合优势

field_weekly_counts.csv:

- 真实现场周故障计数数据（脱敏处理）
- 包含噪声和异常波动，适合测试模型鲁棒性
- 适用模型：所有模型

12 附录 I：测试环境配置清单

序号	包名	版本
1	PySide6	6.7.x
2	pandas	2.2.x
3	numpy	1.26.x
4	scipy	1.11.x
5	matplotlib	3.8.x
6	scikit-learn	1.4.x
7	torch	2.2.x
8	reportlab	4.1.x
9	pytest	8.2.x
10	pytest-qt	4.4.x

表 17 Python 依赖包版本

12.1 测试工具配置

代码格式化：

- black (line-length=100)
- ruff (代码质量检查)

类型检查：

- mypy (严格模式)

打包工具：

- PyInstaller 6.10.x
- uv 0.5.x (包管理器)

12.2 测试环境硬件配置

最低配置：

- CPU: Intel Core i5-8250U 或同等性能
- 内存: 8GB DDR4
- 硬盘: 10GB 可用空间
- 显示: 1920x1080 分辨率

推荐配置：

- CPU: Intel Core i7-10700 或 AMD Ryzen 7 3700X
- 内存: 16GB DDR4
- 硬盘: SSD 20GB 可用空间
- 显示: 2560x1440 分辨率

13 测试用例统计汇总

序号	测试类型	用例数量	覆盖模块
1	功能测试 (GN)	35	数据导入、模型拟合、可视化、报告、实验管理
2	接口测试 (JK)	5	插件接口、CLI 命令行
3	边界测试 (BJ)	9	数据边界、参数边界
4	性能测试 (XN)	4	并发性能、大数据集处理
5	安全性测试 (AQ)	4	输入安全、文件安全
6	兼容性测试 (JR)	4	操作系统、Python 版本、打包程序
7	可用性测试 (KY)	2	用户体验、错误提示
8	回归测试 (HG)	2	功能回归、性能回归
9	文档审查 (WD)	3	用户手册、README、API 文档
总计	126 个测试用例		

表 18 测试用例分类统计

13.1 测试覆盖率目标

- 代码行覆盖率：≥ 85%
- 分支覆盖率：≥ 75%
- 功能覆盖率：100%（所有用户可见功能）
- 模型覆盖率：100%（8 个可靠性模型）

13.2 测试执行策略

- 第一轮：冒烟测试（预计 2 天）
- 核心功能测试用例：GN_01, GN_02, GN_03
 - 快速验证主要流程是否可用
- 第二轮：全面测试（预计 5 天）
- 执行所有功能、接口、边界测试用例
 - 发现和记录缺陷
- 第三轮：专项测试（预计 3 天）
- 性能测试、安全测试、兼容性测试
 - 压力测试和极限场景验证
- 第四轮：回归测试（预计 2 天）
- 修复缺陷后的验证
 - 确保所有问题已解决
- 总测试周期：12 个工作日

— 测试说明文档结束 —