

软件用户手册 V1.0

XT-DDD-SUM

标记	数量	修改单号	签字	日期
编制		会签		
校对		标检		
审核		批准		
会签				

共 7 页

修订记录

版本号	修订状态	简要说明修订内容和范围	修订日期	修订人	批准日期

注：修订记录在体系文件发布后换版时使用，修订状态栏填写：A—增加，M—修改，D—删除

目 次

修订记录	2
1 范围	4
1.1 标识	4
1.2 系统概述	4
1.3 文档概述	4
2 引用文档	4
3 软件综述	4
3.1 软件应用	4
3.2 软件清单	5
3.3 软件环境	6
3.3.1 计算机设备要求	6
3.3.2 通信设备	7
3.3.3 支撑软件（操作系统与依赖）	7
3.3.4 软件组织和操作概述	7
3.4 意外事故及运行的备用状态和方式	9
3.5 保密性	9
3.6 帮助和问题报告	9
4 软件入门	10
4.1 软件的首次用户	10
4.1.1 熟悉设备	10
4.1.2 访问控制	10
4.1.3 安装和设置	10
4.2 启动	11
4.3 停止和挂起	13
5 使用指南	14
5.1 能力	14
5.2 约定	15
5.3 处理规程	16
5.3.1 数据导入与预览	16
5.3.2 模型选择与参数配置	17
5.3.3 执行分析与监控进度	17
5.3.4 结果查看与解读	18
5.3.5 报告导出与归档	19
5.3.6 实验回放	19
5.3.7 参数调优与模型对比	20
5.4 有关的处理	20
5.5 数据备份	21
5.6 错误、故障和紧急情况下的恢复	23
5.7 消息	25
5.8 快速参考指南	27
6 注释	29
6.1 术语与定义	29
6.2 模型公式	30
6.3 缩略语汇总	31
6.4 背景知识	31
6.5 相关标准与规范	32

1 范围

1.1 标识

本条应描述本文档所适用系统和软件的完整标识，适用时，包括其标识号、名称、缩略名、版本号和发布号。

- a) 标识号：ZDP-SYS-2024
- b) 标题：软件可靠性分析系统（Zero-Defect Prediction System）
- c) 缩略名：ZDP
- d) 版本号：V0.1.0

1.2 系统概述

本条应概述本文档所适用系统和软件的用途。

- a) 需方：软件质量保证部门、软件研发团队、可靠性工程部门
- b) 用户：可靠性工程师、质量保证工程师、软件测试人员、项目管理人员
- c) 开发方：ZDP（Zero-Defect Prediction）项目团队

本系统是一款专业的软件可靠性分析工具，基于 Python 科学计算生态及 PySide6 图形界面框架开发，为用户提供缺陷数据导入、多模型拟合分析、可视化诊断及报告导出等完整的可靠性评估工作流。系统支持多种国际认可的可靠性增长模型，包括 Jelinski-Moranda、Goel-Okumoto、S-Shaped、BP 神经网络、支持向量回归（SVR）、灰色模型（GM）及 EMD 混合模型等，能够对软件测试或运行过程中收集的故障数据进行科学建模与预测。

1.3 文档概述

本用户手册详细介绍了 ZDP（Zero-Defect Prediction，零缺陷预测）软件可靠性分析系统的功能特性、操作流程、使用方法和注意事项，旨在帮助用户快速掌握系统的安装配置、数据导入、模型分析、结果解读及报告生成等全流程操作。本手册适用于初次使用本系统的用户以及需要深入了解系统功能的高级用户。

本系统及配套文档不涉及国家秘密信息，属于企业内部技术资料，使用者应遵守相关保密协议，未经授权不得向第三方泄露或传播。

2 引用文档

本章应列出引用文档的编号、标题、编写单位、修订版及日期。

- a) 《GJB 438C-2019 军用软件开发文档通用要求》，国防科技工业委员会，2019 年
- b) 《GB/T 14394-2008 计算机软件可靠性和可维护性管理》，中国国家标准化管理委员会，2008 年
- c) 《IEEE 982.1-2005 IEEE Standard Dictionary of Measures of the Software Aspects of Dependability》，IEEE，2005 年
- d) 《软件可靠性模型及应用》，相关学术文献
- e) 《Python 软件基金会 Python 3.10+ 文档》，<https://docs.python.org/3/>
- f) 《PySide6/Qt for Python 官方文档》，The Qt Company，<https://doc.qt.io/qtforpython/>

3 软件综述

3.1 软件应用

ZDP（Zero-Defect Prediction，零缺陷预测）软件可靠性分析系统是专为软件测试与质量保证领域设计的专业可靠性建模与预测工具。系统预期用途包括：

1. 缺陷数据分析：对软件测试过程中收集的缺陷发现时间序列或故障间隔数据进行统计分析
2. 可靠性增长建模：应用多种经典及现代可靠性模型（如 Jelinski-Moranda、Goel-Okumoto、S-Shaped 等）进行拟合

3. 模型比较与优选：自动计算多项拟合优度指标（RMSE、MAE、R² 等），对模型进行排序，辅助选择最佳模型
4. 预测与趋势分析：基于历史数据预测未来缺陷发现趋势，评估软件可靠性增长态势
5. 可视化诊断：提供预测曲线、残差分析、U 图、Y 图等多种诊断图表，直观评估模型拟合质量
6. 专业报告生成：一键导出包含数据摘要、模型参数、拟合指标及诊断图表的 PDF 分析报告

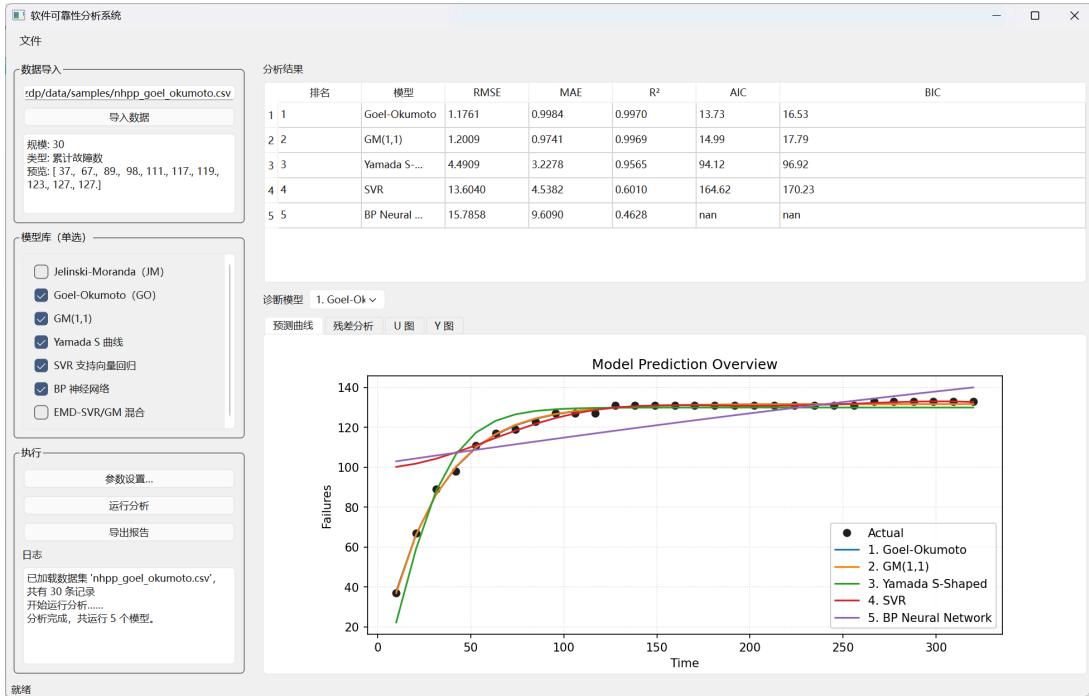


图 1 ZDP 系统主界面概览

本系统能够显著提高可靠性分析工作效率，减少手工计算误差，为软件质量评估与发布决策提供科学依据。主要改进和受益包括：

- 集成 7+ 种主流可靠性模型，一次分析多模型对比
- 智能识别数据类型（累计故障数/故障间隔），自动选择兼容模型
- 图形化操作界面（GUI）与命令行工具（CLI）双模式，适应不同使用场景
- 支持交叉验证（Walk-Forward Validation）评估模型泛化能力
- 可扩展插件架构，支持用户自定义模型注册

3.2 软件清单

使软件运行而必须安装的所有软件文件（含数据库、数据文件及保密性考虑）：

核心程序模块：

- `src/zdp/__init__.py` – 包初始化及公共 API
- `src/zdp/app.py` – GUI 应用入口
- `src/zdp/cli.py` – 命令行接口入口
- `src/zdp/_main_.py` – 模块直接执行入口

数据处理模块：

- `src/zdp/data/loader.py` – CSV/Excel 数据加载与列推断
- `src/zdp/data/dataset.py` – 数据集容器类（FailureDataset）
- `src/zdp/data/types.py` – 数据类型定义（故障间隔 TBF/累计故障数）

可靠性模型库：

- `src/zdp/models/jelinski_moranda.py` – Jelinski-Moranda 模型（TBF 数据）
- `src/zdp/models/goel_okumoto.py` – Goel-Okumoto NHPP 模型（累计故障数）
- `src/zdp/models/s_shaped.py` – S-Shaped 增长模型

- `src/zdp/models/gm.py` – 灰色模型 GM(1, 1)
- `src/zdp/models/bp_neural.py` – BP 神经网络模型
- `src/zdp/models/svr.py` – 支持向量回归模型
- `src/zdp/models/hybrid.py` – EMD-SVR/GM 混合模型
- `src/zdp/models/plugins.py` – 插件模型加载器

分析与服务模块:

- `src/zdp/services/analysis.py` – 模型编排与排序服务 (AnalysisService)
- `src/zdp/services/validation.py` – 交叉验证服务 (Walk-Forward)
- `src/zdp/services/intervals.py` – 预测区间计算
- `src/zdp/services/experiments.py` – 实验导出/导入 (ZIP 归档)

可视化模块:

- `src/zdp/visualization/plots.py` – 绘图函数 (预测曲线、残差、U 图、Y 图)
- `src/zdp/visualization/matplotlib_canvas.py` – Qt 集成的 Matplotlib 画布
- `src/zdp/visualization/utils.py` – 图表转 PNG 等工具函数

报告生成模块:

- `src/zdp/reporting/report_builder.py` – PDF 报告构建器 (基于 ReportLab)

图形界面模块:

- `src/zdp/gui/main_window.py` – 主窗口 (数据导入、模型选择、结果展示)
- `src/zdp/gui/parameters_window.py` – 参数配置窗口 (交叉验证、模型超参数)
- `src/zdp/gui/experiment_window.py` – 实验回放窗口

配置与依赖文件:

- `pyproject.toml` – 项目元数据、依赖声明、脚本入口配置
- `zdp.spec` – PyInstaller 打包配置 (Windows 单文件 exe)
- `zdp_entry.py` – PyInstaller 入口脚本 (含 multiprocessing 支持)

示例数据文件:

- `data/samples/nhpp_goel_okumoto.csv` – Goel-Okumoto 模型示例数据
- `data/samples/tbf_jm_synthetic.csv` – Jelinski-Moranda 模型 TBF 数据
- `data/samples/tbf_s_shaped.csv` – S-Shaped 模型示例
- `data/samples/field_weekly_counts.csv` – 现场周计数数据
- `data/samples/tbf_jm_nonmonotonic_solvable.csv` – JM 非单调可解数据

文档与脚本:

- `README.md` – 项目说明与快速开始指南
- `docs/manual/main.typ` – 用户手册源文件 (Typst 格式)
- `scripts/generate_sample_data.py` – 示例数据生成脚本

保密性说明: 本系统不内置任何敏感数据库或密钥文件, 用户导入的缺陷数据保存在用户本地, 系统不对外传输数据。

3.3 软件环境

描述硬件、软件、手工操作和其他资源。

3.3.1 计算机设备要求

- 处理器 (CPU) : x86_64 架构, 推荐 Intel Core i5 或 AMD Ryzen 5 及以上, 主频 2.0 GHz+
- 内存 (RAM) : 最低 4 GB, 推荐 8 GB 及以上 (大数据集或神经网络模型训练需更多内存)
- 硬盘空间: 系统安装需约 500 MB, 推荐预留 2 GB 用于数据、报告及临时文件
- 显示器: 分辨率不低于 1280×800, 推荐 1920×1080 以获得最佳图形界面体验
- 输入设备: 标准键盘、鼠标或触控板

3.3.2 通信设备

- 本系统为单机桌面应用，不强制要求网络连接
- 若需安装 Python 依赖包（开发环境），需要 Internet 连接访问 PyPI 镜像源
- PDF 报告导出及数据导入导出均为本地文件操作，无网络通信

3.3.3 支撑软件（操作系统与依赖）

- 操作系统：
 - Windows 10/11 (64 位)，推荐 Windows 10 20H2 及以上版本
 - 理论上支持 Linux (Ubuntu 20.04+、Fedora 35+) 和 macOS 11+，但主要测试环境为 Windows
- Python 运行时：
 - Python 3.10 及以上版本（系统基于 Python 3.10 开发）
 - 推荐使用 uv 包管理器（快速依赖安装）或官方 pip
- 核心依赖库（通过 pip/uv 自动安装）：
 - PySide6 ≥ 6.7 : Qt 图形界面框架
 - NumPy ≥ 1.26 : 数值计算基础库
 - Pandas ≥ 2.2 : 数据表处理
 - SciPy ≥ 1.11 : 科学计算与优化算法
 - Matplotlib ≥ 3.8 : 绘图与可视化
 - scikit-learn ≥ 1.4 : 机器学习算法 (SVR 等)
 - PyTorch ≥ 2.2 : 深度学习框架 (BP 神经网络)
 - ReportLab ≥ 4.1 : PDF 生成 (需中文字体支持)
 - statsmodels ≥ 0.14 : 统计模型工具
 - PyQtGraph ≥ 0.13 : 高性能绘图组件
- 字体支持 (Windows):
 - PDF 中文渲染需要系统字体：微软雅黑 (msyh.ttc)、宋体 (simsun.ttc) 或黑体 (simhei.ttf)
 - 系统会自动检测并注册可用字体，通常 Windows 系统已预装
- 开发与测试工具（可选）：
 - pytest ≥ 8.2 : 单元测试框架
 - ruff、black: 代码格式化与检查
 - PyInstaller ≥ 6.10 : 可执行文件打包

3.3.4 软件组织和操作概述

从用户角度看，ZDP 系统由以下逻辑部件组成：

1. 数据导入与预处理层：
 - 自动识别 CSV/Excel 文件格式（支持.csv、.txt、.tsv、.xls、.xlsx）
 - 智能推断时间列与值列（若未显式指定，根据列名匹配）
 - 自动判定数据类型：累计故障数 (Cumulative Failures) 或故障间隔 (TBF)
 - 若时间列缺失，自动生成序号时间轴 (1, 2, 3, ...)
2. 模型拟合与评估层：
 - 编排器 (AnalysisService) 根据数据类型自动选择兼容模型运行
 - 并发拟合多个模型（各模型独立计算）
 - 计算拟合优度指标：RMSE (均方根误差)、MAE (平均绝对误差)、R² (决定系数)
 - 可选交叉验证：基于滚动窗口 (Walk-Forward) 评估模型泛化能力，生成 cv_rmse、cv_mae 等指标
 - 按指定指标（默认 RMSE）升序排列模型，赋予排名 (Rank 1 为最佳模型)
3. 可视化与诊断层：
 - 预测曲线图：叠加显示所有成功拟合模型的预测曲线与实际数据点
 - 残差分析图：计算残差 (实际值-预测值)，绘制散点图与零线，评估偏差分布

- U 图（累计密度检验）：检验模型假设的时间分布是否符合均匀分布
 - Y 图（间隔分布检验）：评估故障间隔预测是否服从指数分布假设
 - 用户可通过下拉菜单切换“图表模型”，查看不同模型的诊断图
4. 报告生成层：
- 基于 ReportLab 生成 PDF 文档，包含：
 - ▶ 数据集摘要（记录数、数据类型、时间范围）
 - ▶ 模型排名表（参数、指标、排名）
 - ▶ 嵌入式图表（预测曲线、残差图等，以 PNG 格式嵌入）
 - 自动注册中文字体（宋体/微软雅黑），避免乱码
 - 导出文件可用于存档或对外展示分析结果
5. 实验管理层（可选）：
- 导出实验：将数据集、配置参数、分析结果打包为 ZIP 归档文件（experiment.zip）
 - 导入实验：加载历史分析记录，支持在不同机器上复现结果（不含模型 pickle，仅含结果 JSON）

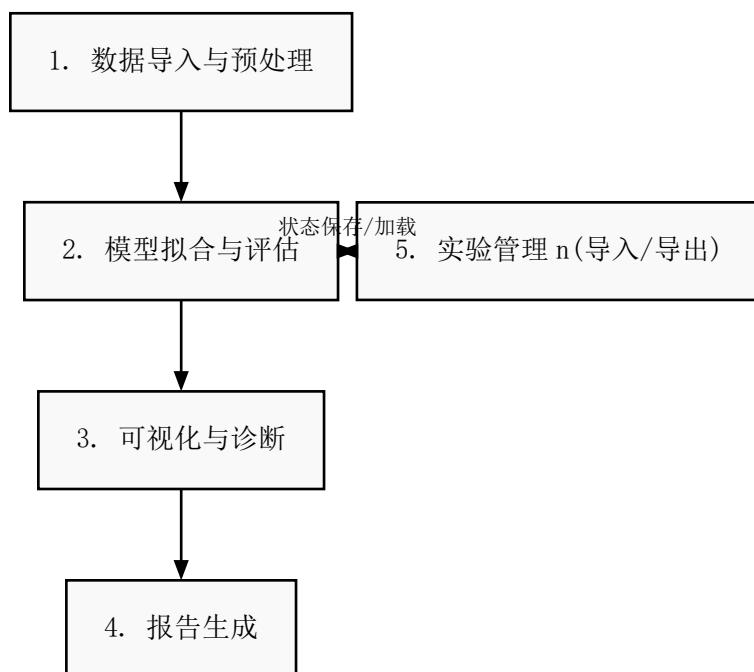


图 2 ZDP 系统逻辑分层架构图

性能特性：

- 小规模数据集（<1000 点）：拟合与排序通常在 5 秒内完成
- 大规模数据集（1000–10000 点）：BP 神经网络或 SVR 可能需 30 秒至 2 分钟
- 响应时间受数据规模、模型选择及硬件性能影响
- GUI 分析在后台线程运行，界面保持响应，进度条实时显示

可靠性措施：

- 模型拟合失败时不中断流程，自动跳过该模型并记录日志
- 数据格式异常时弹出明确错误提示，引导用户检查文件
- 实验导出包含配置备份，便于问题追溯
- 单元测试覆盖核心数据处理与模型拟合逻辑

监控措施：

- GUI 状态栏实时显示当前操作状态（如“分析完成”、“加载数据中...”）
- 进度条显示后台任务执行进度
- 命令行模式支持--verbose 参数（未来版本）输出详细日志

3.4 意外事故及运行的备用状态和方式

意外事故处理:

- 数据加载失败: 若导入的 CSV/Excel 文件格式错误、编码问题或缺失必需列, 系统弹出错误对话框, 提示具体问题 (如“未找到有效的值列”), 用户需检查文件格式后重新导入
- 模型拟合失败: 某些模型对数据有特定要求 (如 JM 模型要求 P 值满足有限解条件), 若条件不满足, 该模型自动跳过, 不影响其他模型运行, 结果表中不显示该模型
- 内存不足: 处理大数据集 (数万点) 或运行神经网络时若内存耗尽, Python 解释器可能崩溃或报 MemoryError, 建议关闭其他程序、减少数据点或增加物理内存
- GUI 无响应: 分析过程在后台线程运行, 但若任务长时间挂起 (如优化算法陷入循环), 可关闭程序窗口强制终止, 重启后重新加载数据

备用状态与方式:

- 命令行备用模式: 若 GUI 界面因显示驱动问题无法启动, 可使用 `zdp-cli` 命令行工具完成全流程分析 (数据加载、模型拟合、结果导出), 功能与 GUI 等价
- 离线运行: 系统不依赖网络连接, 所有计算均在本地进行, 适用于离线环境
- 数据备份: 建议用户定期备份原始数据文件 (CSV/Excel), 系统本身不自动备份用户数据
- 手工验证: 对于关键决策 (如软件发布), 建议人工抽查模型拟合结果, 结合残差图、U 图等诊断工具判断模型适用性, 避免盲目依赖排名

3.5 保密性

- 本系统为单机应用, 不涉及网络数据传输, 用户导入的缺陷数据保存在本地磁盘
- 生成的 PDF 报告包含用户数据内容, 应按企业保密规定妥善保管, 避免泄露测试数据或质量评估结果
- 系统源代码及配套文档属于企业内部技术资料, 未经授权不得向外部分发
- 实验导出文件 (ZIP) 包含完整数据集及分析结果, 传输或存储时应加密处理
- 插件模型功能允许加载外部 Python 代码, 使用第三方插件前应进行代码审查, 防止恶意代码执行

3.6 帮助和问题报告

系统内帮助:

- GUI 主窗口菜单栏提供“帮助 → 用户手册”(未来版本)快速访问本手册电子版
- 参数配置窗口各字段提供工具提示 (Tooltip), 鼠标悬停显示参数说明
- 命令行工具支持 `zdp-cli --help` 查看完整参数列表及用法说明

问题报告途径:

- 技术支持邮箱: `zdp-support@example.com` (示例, 实际部署时填写企业内部支持邮箱)
- 内部 Issue 跟踪: 企业 GitLab/Jira 系统提交 Bug 或功能建议
- 开发团队: 联系 ZDP Team 进行问题诊断或定制化需求讨论

问题报告应包含:

1. 操作系统版本 (Windows 10/11, 构建号)
2. Python 版本 (运行 `python --version` 获取)
3. 软件版本 (`zdp -version`, 当前为 0.1.0)
4. 详细复现步骤 (导入的数据文件格式、选择的模型、参数配置)
5. 错误信息截图或日志文件 (若程序崩溃, 可附加 Python traceback)
6. 期望行为描述

已知问题与解决方法:

- 中文路径乱码: 若 CSV 文件路径包含中文, 部分 Windows 系统可能出现编码问题, 建议使用英文路径或 UTF-8 编码保存文件
- PDF 字体缺失: 若生成的 PDF 中中文显示为方块, 检查 Windows/Fonts 目录是否存在 `simsun.ttc`、`msyh.ttc` 等字体文件

- PyTorch 初始化慢：首次运行 BP 神经网络模型时，PyTorch 需要初始化 CUDA 环境（若有 GPU）或 MKL 库，可能耗时 10-30 秒，属正常现象

4 软件入门

4.1 软件的首次用户

4.1.1 熟悉设备

电源与启动规程：

- 确保计算机正常供电并开机，进入 Windows 操作系统
- 显示器分辨率推荐设置为 1920×1080 或更高，以获得最佳界面显示效果
- 鼠标/触控板：用于点击按钮、选择模型、拖动窗口等操作
- 键盘：用于输入文件路径、参数值、搜索筛选等

屏幕能力与布局：

- ZDP 主窗口采用左右分栏布局：
 - 左侧：数据导入、模型选择、控制按钮、分析日志
 - 右侧：结果表格、图表展示（预测曲线、残差、U 图、Y 图）
- 窗口可自由拖动边界调整大小，分栏比例可拖动中间分隔线调节
- 支持最小化、最大化、关闭等标准窗口操作

光标定位与交互：

- 鼠标点击表格行可查看该模型的详细参数与指标
- 双击数据文件路径可快速选中复制
- 拖放文件到主窗口（未来版本支持）可快速导入数据

键盘布局与快捷键：

- Ctrl+O：打开文件对话框（导入数据）（未来版本）
- Ctrl+S：保存分析结果（导出报告）（未来版本）
- Ctrl+Q：退出程序（未来版本）
- F1：打开帮助文档（未来版本）

4.1.2 访问控制

口令获取与管理：

- 当前版本（V0.1.0）为单机应用，无需用户登录或口令认证
- 若企业部署时需增加访问控制，可配合 Windows 用户账户权限管理：
 - 将软件安装目录权限设置为管理员可读写、普通用户只读
 - 通过 Active Directory（AD）域控管理用户访问权限
- 未来企业版可能增加用户角色管理（管理员/分析员/只读用户），届时需要设置初始管理员密码

文件访问权限：

- 用户需要对导入的 CSV/Excel 文件所在目录有读权限
- 导出 PDF 报告时需要对目标目录有写权限
- 建议为 ZDP 创建专用工作目录（如 D:\ZDP_Work），统一存放数据文件与报告

4.1.3 安装和设置

方式一：开发环境安装（需要 Python）

1. 安装 Python 3.10+：
 - 访问 <https://www.python.org/downloads/>，下载 Windows 安装器
 - 运行安装程序，勾选“Add Python to PATH”
 - 验证安装：打开 PowerShell，输入 `python --version`，显示 Python 3.10.x 或更高版本
2. 安装 uv 包管理器（推荐，速度更快）：

```
pip install uv
```

或使用官方安装脚本（参见 <https://docs.astral.sh/uv/>）

3. 获取 ZDP 源代码：

- 从企业内部 Git 仓库克隆或解压发布包到本地目录（如 E:\zdp）

4. 安装依赖：

```
cd E:\\zdp
uv sync --all-extras
```

此命令会自动安装 pyproject.toml 中声明的所有依赖包（PySide6、NumPy、Pandas 等），首次运行需 5-10 分钟（取决于网络速度）

5. 验证安装：

```
uv run zdp --help      # 测试GUI启动
uv run zdp-cli --help  # 测试CLI工具
```

若无报错，安装成功

方式二：可执行文件安装（Windows 单文件版）

1. 获取安装包：

- 从企业发布服务器下载 zdp.exe（约 150-200 MB）
- 或由开发团队使用 PyInstaller 打包生成（运行 uv run pyinstaller zdp.spec）

2. 放置程序：

- 将 zdp.exe 复制到任意目录（如 C:\Program Files\ZDP\）
- 创建桌面快捷方式（可选）：右键 zdp.exe → 发送到 → 桌面快捷方式

3. 首次运行：

- 双击 zdp.exe，Windows 可能弹出 SmartScreen 警告，点击“仍要运行”
- 若程序无法启动，检查是否缺少 Visual C++ Redistributable（安装 VC++ 2015-2022 运行库）

配置与初始化设置：

- 软件无需额外配置文件，采用默认参数即可启动
- 首次启动会检测系统字体，注册中文 PDF 字体（自动完成）
- 若需自定义模型参数（如 BP 隐藏层节点数、SVR 核函数），在 GUI 主窗口点击“参数配置”按钮，或使用 CLI 的--bp-hidden、--svr-kernel 等参数

删除旧版本文件：

- 若从旧版本升级，建议先卸载：
 - 开发环境：删除旧版本目录，重新克隆代码并运行 uv sync
 - 可执行文件：删除旧的 zdp.exe，替换为新版本
- 注意备份历史分析结果（PDF 报告、实验 ZIP 文件）

输入参数与测试：

- 使用示例数据验证安装：

```
cd E:\\zdp
uv run zdp-cli data/samples/nhpp_goel_okumoto.csv --model go --report test_report.pdf
```

成功生成 test_report.pdf 表示安装正确

4.2 启动

提供开始工作的逐步规程、初始化设置、数据库 SQL 文件及问题检查单。

GUI 图形界面启动：

1. 开发环境启动:

```
# 方式1: 通过uv运行
cd E:\\zdp
uv run zdp

# 方式2: 通过Python模块运行
python -m zdp
```

2. 可执行文件启动:

- 双击 zdp.exe 图标

- 或在 PowerShell 中运行:

```
C:\\Path\\To\\zdp.exe
```

3. 启动过程:

- 程序启动后显示主窗口，标题为“软件可靠性分析系统”
- 左侧面板显示“数据导入”、“模型选择”、“控制”三个分组
- 右侧面板显示空白结果表格与图表区域
- 状态栏显示“就绪”或“Ready”

CLI 命令行启动:

```
# 基本用法 (自动推断列名)
uv run zdp-cli data.csv

# 指定时间列与值列
uv run zdp-cli data.csv --time-column t --value-column failures

# 仅运行指定模型
uv run zdp-cli data.csv --model jm --model go

# 导出PDF报告
uv run zdp-cli data.csv --report output.pdf

# 启用交叉验证
uv run zdp-cli data.csv --walk-forward --cv-horizon 2

# 查看完整参数列表
uv run zdp-cli --help
```

初始化设置检查:

- ✓ Python 版本 ≥ 3.10 (运行 `python --version`)
- ✓ 依赖包已安装 (PySide6、NumPy、Pandas 等)
- ✓ 系统字体可用 (Windows/Fonts 目录存在 simsun.ttc 或 msyh.ttc)
- ✓ 工作目录有读写权限 (用于临时文件与报告导出)

常见启动问题检查单:

问题现象	可能原因	解决方法
双击 zdp.exe 无反应	缺少 VC++运行库	安装 Microsoft Visual C++ Redistributable 2015–2022
报错 “ImportError: No module named PySide6”	依赖未安装	运行 <code>uv sync --all-extras</code> 重新安装依赖
GUI 窗口显示乱码	字体缺失或区域设置问题	检查系统语言设置为中文，安装中文字体包

报错“Python was not found”	Python 未添加到 PATH	重新安装 Python 并勾选“Add Python to PATH”
CLI 提示“zdp-cli: command not found”	脚本未注册	使用 <code>uv run zdp-cli</code> 代替直接调用
启动后立即崩溃	显卡驱动过旧	更新显卡驱动（尤其 Intel 核显），或使用 CLI 模式

数据库及初始化 SQL:

- 本系统不使用传统数据库（MySQL、PostgreSQL 等），所有数据通过 CSV/Excel 文件导入
- 无需执行 SQL 初始化脚本
- 未来版本若增加结果持久化功能，可能采用 SQLite 嵌入式数据库（自动创建）

4.3 停止和挂起

如何停止或中断使用，以及判断正常结束的标志。

正常退出 GUI:

1. 点击窗口右上角“×”关闭按钮
2. 或通过菜单栏“文件 → 退出”（未来版本）
3. 系统弹出确认对话框：“是否保存当前分析结果？”
 - 点击“是”：弹出文件保存对话框，导出 PDF 报告或实验 ZIP
 - 点击“否”：直接退出，分析结果丢失
 - 点击“取消”：返回主窗口，继续工作
4. 窗口关闭，进程终止，内存资源释放

强制终止:

- 若程序无响应（如模型拟合陷入死循环），可使用以下方法：
 - ▶ Windows 任务管理器（Ctrl+Shift+Esc）→ 找到“zdp.exe”或“python.exe”→ 右键“结束任务”
 - ▶ PowerShell 中使用 `Stop-Process -Name zdp` 命令
- 强制终止可能导致临时文件未清理，建议重启后手工删除临时目录（%TEMP%\zdp）

中断 CLI 命令:

- 运行命令行分析时，按 Ctrl+C 发送中断信号
- 程序捕获中断后打印“Analysis interrupted by user”并退出
- 已生成的中间结果（如部分模型拟合完成）不会保存

挂起与后台运行:

- GUI 程序不支持最小化到托盘后台运行，最小化窗口后仍占用前台进程
- CLI 命令不支持挂起（Ctrl+Z 在 Windows PowerShell 中行为不一致），建议使用后台任务（Start-Job）或重定向输出
- 长时间分析建议使用 CLI 模式配合日志重定向：

```
uv run zdp-cli large_data.csv --report out.pdf > analysis.log 2>&1
```

正常结束标志:

- GUI:
 - ▶ 状态栏显示“分析完成”，进度条消失
 - ▶ 结果表格填充模型排名与指标数据
 - ▶ 图表区域显示预测曲线等可视化结果
 - ▶ 分析日志输出“Analysis completed. X models ranked.”
- CLI:
 - ▶ 终端输出排名表格（ASCII 表格格式）
 - ▶ 若指定--report 参数，打印“Report saved to: /path/to/output.pdf”

- 退出代码为 0 (运行 `echo $LASTEXITCODE` 检查)
- 异常结束标志:
 - 弹出错误对话框 (GUI) 或打印红色错误信息 (CLI)
 - 退出代码非 0 (CLI)
 - 日志中包含 “Error”、“Exception”、“Failed” 等关键词

5 使用指南

5.1 能力

简述事务、菜单、功能之间的关系。

ZDP 系统提供完整的软件可靠性分析工作流，核心能力包括：

数据导入与预处理：

- 支持 CSV、Excel (.xls/.xlsx)、TSV 等常见表格格式
- 自动推断时间列 (关键词：time、t、timestamp) 与值列 (关键词：failures、tbf、interval)
- 自动判定数据类型：
 - 累计故障数 (Cumulative Failures): 数值单调递增
 - 故障间隔 (Time Between Failures, TBF): 相邻故障发生的时间差
- 缺失时间列时自动生成序号时间轴 (1, 2, 3, ...)

模型拟合与分析：系统内置 7 种主流可靠性模型，分类如下：

模型名称	适用数据类型	参数	特点
Jelinski-Moranda (JM)	TBF	N0 (初始缺陷数)、 Φ (故障率)	经典缺陷消耗模型，假设每次修复减少 1 个缺陷
Goel-Okumoto (GO)	累计故障数	a (最终缺陷数)、b (发现率)	NHPP 模型，指数增长，广泛应用
S-Shaped	累计故障数	a、b	S 型增长曲线，适合学习效应明显的场景
GM(1, 1)	累计故障数	a、b	灰色模型，适合小样本数据
BP 神经网络	累计故障数	hidden_size、epochs、lr	非线性拟合能力强，需训练较长时间
支持向量回归 (SVR)	累计故障数	kernel、C、epsilon	机器学习方法，泛化能力较好
EMD 混合模型	累计故障数	SVR 参数	经验模态分解+SVR/GM，处理非平稳数据

模型自动筛选：

- 系统根据数据类型自动跳过不兼容模型 (如 TBF 数据不运行 BP、SVR)
- 拟合失败的模型 (如 JM 无有限解) 自动排除，不影响其他模型

指标计算与排序：每个模型计算以下拟合优度指标：

- RMSE (均方根误差): 预测值与实际值差异的平方根均值，越小越好
- MAE (平均绝对误差): 预测值与实际值差异的绝对值均值
- R² (决定系数): 拟合优度，取值 0-1，越接近 1 表示拟合越好
- MSE (均方误差): RMSE 的平方
- MAPE (平均绝对百分比误差): 百分比形式的相对误差

交叉验证 (Walk-Forward) 额外提供：

- `cv_rmse`、`cv_mae`、`cv_r2`: 基于滚动窗口的泛化指标，更准确评估预测能力

系统按指定指标 (默认 RMSE) 升序排列模型，Rank 1 为最佳模型。

可视化诊断： 提供 4 类诊断图表，辅助判断模型适用性：

1. 预测曲线图：叠加所有模型预测曲线与实际数据散点，直观对比拟合效果
2. 残差分析图：散点图显示残差（实际-预测），零线为理想拟合，评估偏差分布及异方差性
3. U 图（累计密度检验）：检验模型假设的时间分布是否均匀，点应均匀分布在 45° 线附近
4. Y 图（间隔分布检验）：检验故障间隔是否服从指数分布，点应线性分布

用户可通过“图表模型”下拉菜单切换查看不同模型的诊断图（默认显示最佳模型）。

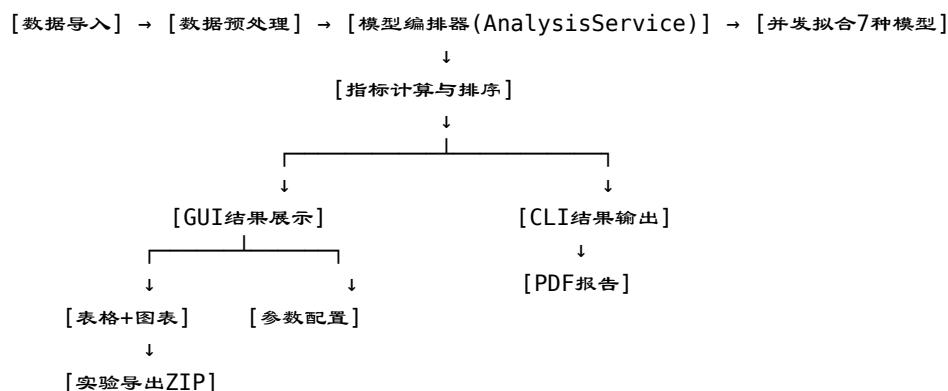
报告导出：

- 一键生成 PDF 分析报告，包含：
 - 数据集摘要（样本数、数据类型、时间范围）
 - 模型排名表（参数、指标、排名）
 - 嵌入式高清图表（预测曲线、残差图等）
 - 生成时间、分析配置等元信息
- 支持中文显示（自动注册系统字体）

实验管理（高级功能）：

- 导出实验：将数据集、配置、结果打包为 ZIP 文件（实验归档/复现）
- 导入实验：加载历史分析记录，通过“文件 → 实验回放”菜单打开

功能模块关系图：



5.2 约定

描述颜色、铃声、缩略语及命名规则。

界面颜色约定：

- 蓝色：主要操作按钮（“开始分析”、“导入数据”）
- 绿色：成功状态提示（“分析完成”）
- 红色：错误提示、警告信息（“数据加载失败”）
- 灰色：禁用状态控件（分析进行中时按钮置灰）
- 黄色：高亮选中项（表格行选中）

图表颜色约定：

- 黑色实心点：实际观测数据
- 彩色曲线：各模型预测曲线（每个模型自动分配不同颜色）
- 红色虚线：零线（残差图中）或参考线（U 图、Y 图中的理想分布线）

铃声与提示音：

- 系统不使用声音提示，所有消息通过弹窗或状态栏文字显示
- Windows 系统可能在错误对话框弹出时播放默认错误提示音（由操作系统控制）

缩略语：

- ZDP: Zero-Defect Prediction（零缺陷预测）
- TBF: Time Between Failures（故障间隔）

- NHPP: Non-Homogeneous Poisson Process (非齐次泊松过程)
- RMSE: Root Mean Square Error (均方根误差)
- MAE: Mean Absolute Error (平均绝对误差)
- R²: Coefficient of Determination (决定系数)
- MAPE: Mean Absolute Percentage Error (平均绝对百分比误差)
- MSE: Mean Square Error (均方误差)
- JM: Jelinski–Moranda (JM 模型)
- GO: Goel–Okumoto (GO 模型)
- GM: Grey Model (灰色模型)
- BP: Back Propagation (反向传播神经网络)
- SVR: Support Vector Regression (支持向量回归)
- EMD: Empirical Mode Decomposition (经验模态分解)
- CV: Cross-Validation (交叉验证)
- GUI: Graphical User Interface (图形用户界面)
- CLI: Command-Line Interface (命令行界面)
- PDF: Portable Document Format (便携式文档格式)

文件命名规则:

- 数据文件: 建议使用英文字母、数字、下划线, 如 `project_week1_failures.csv`
- 报告文件: 建议包含日期与项目名, 如 `zdp_report_ProjectX_20241227.pdf`
- 实验导出: 默认命名为 `experiment_YYYYMMDD_HHMMSS.zip`

列名推断规则: 系统根据以下关键词 (不区分大小写) 自动识别列:

- 时间列: time、t、timestamp、elapsed、duration
- TBF 值列: tbf、interval、delta、interfailure、tbfs、dt
- 累计故障数列: cumulative、failures、failure、count、n、nt、mt

若列名不匹配, 系统选择第一个数值列作为值列, 第二个数值列作为时间列 (若存在)。

模型标识符 (CLI 参数):

- jm 或 jelinski-moranda: Jelinski–Moranda
- go 或 goel-okumoto: Goel–Okumoto
- gm: 灰色模型 GM(1, 1)
- s 或 s-shaped: S-Shaped
- bp: BP 神经网络
- svr: 支持向量回归
- hybrid: EMD 混合模型

5.3 处理规程

5.3.1 数据导入与预览

操作步骤:

1. 启动 ZDP 主窗口
2. 点击左侧“数据导入”分组中的“导入数据”按钮
3. 弹出文件选择对话框, 浏览到数据文件目录
4. 选择 CSV 或 Excel 文件 (支持.csv、.txt、.tsv、.xls、.xlsx 格式), 点击“打开”
5. 系统自动加载数据并执行以下操作:
 - 推断时间列与值列 (若未显式指定)
 - 判定数据类型 (累计故障数或 TBF)
 - 在“数据预览”文本框中显示前 10 行数据
 - 路径字段显示文件完整路径
6. 检查预览数据是否正确, 若列识别错误可关闭程序, 在 CLI 中使用--time-column 和--value-column 参数手动指定

用户输入:

- 文件路径（通过对话框选择）
- 可选：显式指定列名（CLI 模式）

输出消息:

- 成功：“数据加载成功：100 条记录（累计故障数）”
- 失败：“数据加载失败：未找到有效的值列，请检查文件格式”

帮助工具:

- 工具提示（Tooltip）：鼠标悬停在“导入数据”按钮上显示支持的文件格式
- 示例数据：系统自带 data/samples/ 目录下 5 个示例文件，可用于测试

5.3.2 模型选择与参数配置

操作步骤:

1. 数据导入成功后，左侧“模型选择”分组显示 7 个复选框：
 - Jelinski-Moranda（TBF 数据专用）
 - Goel-Okumoto（累计故障数专用）
 - S-Shaped
 - 灰色模型 GM(1, 1)
 - BP 神经网络（累计故障数专用）
 - 支持向量回归（累计故障数专用）
 - EMD 混合模型（累计故障数专用）
2. 默认情况下，系统根据数据类型预勾选兼容模型：
 - TBF 数据：仅勾选 JM、S-Shaped
 - 累计故障数：勾选 GO、S-Shaped、GM、BP、SVR、Hybrid
3. 用户可手动勾选/取消勾选复选框，选择需要运行的模型
4. 若需调整高级参数（如 BP 隐藏层节点数、SVR 核函数），点击“参数配置”按钮
5. 弹出参数配置窗口，提供以下设置：
 - 交叉验证：启用 Walk-Forward 验证、最小训练集大小、预测步长
 - 预测区间：启用预测区间带、置信水平 α （如 0.05 表示 95% 置信区间）
 - BP 参数：隐藏层节点数、训练轮数、学习率、动量、训练集划分比例
 - SVR 参数：核函数（RBF/Poly/Linear/Sigmoid）、惩罚系数 C、 ϵ -管宽度
 - Hybrid 参数：SVR 核函数、C、 ϵ （用于 EMD 混合模型）
6. 调整参数后点击“确定”保存，关闭参数窗口

用户输入:

- 模型复选框选中状态（多选）
- 高级参数值（数值型）

输出消息:

- 状态栏显示：“已选择 5 个模型”

参数说明（工具提示）:

- BP 隐藏层节点数：神经网络中间层神经元数量，推荐 16–64，过大可能过拟合
- SVR 核函数：RBF（径向基，通用），Poly（多项式），Linear（线性），Sigmoid（S 型）
- 交叉验证最小训练集：设为 0 时自动选择数据集的 70%，建议 \geq 数据总量的一半
- 预测步长：每次验证向前预测的点数，1 表示单步预测，2 表示预测未来 2 个时间点

5.3.3 执行分析与监控进度

操作步骤:

1. 数据导入并选择模型后，点击左侧“控制”分组中的“开始分析”按钮
2. 按钮立即置灰变为“分析中…”，防止重复点击
3. 进度条出现并滚动（无法显示精确百分比，因为各模型耗时不同）
4. 系统在后台线程依次执行：

- 为每个选中模型创建实例
 - 调用 `model.fit(dataset)` 拟合参数
 - 计算 RMSE、MAE、R² 等指标
 - 若启用交叉验证，执行 Walk-Forward 切分与验证
 - 若启用预测区间，计算置信带
5. 分析日志实时输出到左下角日志框（如“正在拟合 Goel-Okumoto...”）
 6. 拟合失败的模型跳过并记录警告（如“Jelinski-Moranda 无有限解，已跳过”）
 7. 所有模型完成后，进度条消失，按钮恢复为“开始分析”
 8. 状态栏显示：“分析完成，共 5 个模型排名”

输出消息：

- 进度中：“正在拟合模型：Goel-Okumoto (2/5)”
- 完成：“分析完成。最佳模型：Goel-Okumoto (RMSE=2.35)”
- 异常：“分析失败：数据点数不足，至少需要 3 个数据点”

帮助工具：

- 进度条：显示任务进行中状态（虽不显示精确百分比，但滚动表示未卡死）
- 日志框：记录每个模型的拟合状态，便于问题定位

5.3.4 结果查看与解读

操作步骤：

1. 分析完成后，右侧“结果与图表”区域自动刷新
2. 结果表格：
 - 列名：排名 (Rank)、模型名称 (Model)、RMSE、MAE、R²、参数
 - 行排序：按 RMSE 升序（或指定的排序指标）
 - 点击表头可切换排序（未来版本）
 - 双击某行可在图表中高亮该模型曲线（未来版本）
3. 预测曲线图（第一个选项卡）：
 - 黑色散点：实际观测数据
 - 多条彩色曲线：各模型预测曲线（图例显示模型名与 RMSE）
 - 横轴：时间（或序号）
 - 纵轴：累计故障数或故障间隔（根据数据类型）
4. 残差分析图（第二个选项卡）：
 - 横轴：时间
 - 纵轴：残差（实际值-预测值）
 - 红色虚线：零线（理想情况）
 - 散点：残差分布，随机分布在零线附近为佳，系统性偏离表示模型偏差
5. U 图（第三个选项卡）：
 - 横轴：理论累计分布（均匀分布）
 - 纵轴：实际累计故障时间
 - 红色虚线：45° 理想线
 - 点应沿直线分布，偏离表示时间假设不成立
6. Y 图（第四个选项卡）：
 - 横轴：序号
 - 纵轴：标准化间隔 (-ln(1-U))
 - 红色虚线：理想指数分布
 - 线性关系良好表示间隔符合指数分布假设
7. 使用“图表模型”下拉菜单切换查看不同模型的残差/U/Y 图（默认显示 Rank 1 最佳模型）

结果解读要点：

- RMSE：越小越好，表示平均误差小；建议选择 RMSE 最小的模型
- R²：越接近 1 越好，>0.9 为优秀拟合，0.7–0.9 为良好，<0.7 需谨慎使用

- 残差图：随机散布为佳，若呈现趋势（如先负后正）表示模型系统性偏差
- U 图：点越贴近 45° 线越好，S 型弯曲表示模型假设与数据不符
- Y 图：点越线性越好，曲线或阶跃表示指数分布假设不成立
- 参数合理性：如 G0 模型的 a（最终缺陷数）应 > 当前观测到的最大累计故障数

5.3.5 报告导出与归档

操作步骤：

1. 分析完成并确认结果无误后，点击左侧“控制”分组中的“导出报告”按钮
2. 弹出文件保存对话框，选择保存路径与文件名（默认 zdp_report.pdf）
3. 系统调用 ReportBuilder 生成 PDF 文档，包含：
 - 标题页：分析报告、生成时间、软件版本
 - 数据集摘要：样本数、数据类型、时间范围、最大值/最小值
 - 模型排名表：以表格形式展示 Rank、模型名、参数、指标
 - 图表页：嵌入预测曲线、残差图（高清 PNG, 300 DPI）
 - 诊断说明：简要解释各指标含义（未来版本）
4. 生成过程显示进度对话框（约 5-15 秒，取决于图表数量）
5. 完成后弹出提示：“报告已保存至：D:\Reports\zdp_report.pdf”
6. 状态栏显示：“PDF 报告导出成功”

CLI 模式导出：

```
uv run zdp-cli data.csv --model go --model jm --report output.pdf
```

命令执行后自动生成 output.pdf，无需交互

实验归档（高级用户）：

1. 在主菜单栏选择“文件 → 导出实验”（未来版本）或使用 CLI 参数：

```
uv run zdp-cli data.csv --export-experiment experiment.zip
```

2. 系统将以下内容打包为 ZIP：

- dataset.csv：原始数据（或归一化后的数据）
- config.json：分析配置（模型列表、参数、交叉验证设置）
- results.json：排名结果、指标、参数

3. ZIP 文件可用于：

- 项目归档存档
- 在其他机器上复现分析（通过“实验回放”功能）
- 审计追溯（验证历史分析结果）

输出文件：

- *.pdf：PDF 报告（10-50 MB，取决于图表数量）
- *.zip：实验归档文件（<1 MB，不含图像）

帮助工具：

- 报告预览（未来版本）：导出前预览 PDF 内容
- 自定义报告模板（未来版本）：企业 Logo、页眉页脚定制

5.3.6 实验回放

操作步骤（加载历史分析）：

1. 主菜单选择“文件 → 实验回放”
2. 弹出实验回放窗口，点击“打开实验”按钮
3. 选择之前导出的 experiment.zip 文件
4. 系统解析 ZIP 内容并显示：
 - 数据集摘要（样本数、时间范围）
 - 配置摘要（模型列表、参数）
 - 排名表格（与原始分析结果一致）

5. 点击“重新分析”按钮可基于相同配置重新运行（需要原始数据集）
6. 点击“导出报告”可将回放结果导出为 PDF

用途：

- 审计：验证历史分析结果是否可复现
- 学习：查看他人的分析配置与结果
- 对比：加载多个实验文件，对比不同配置的效果（未来版本）

5.3.7 参数调优与模型对比

典型工作流：

1. 初步分析：使用默认参数运行所有兼容模型，查看排名
2. 诊断检查：检查最佳模型的残差图、U 图、Y 图，判断是否存在系统性偏差
3. 参数调优（若初步结果不理想）：
 - BP 模型：增加 hidden_size（如从 16 调到 64）、增加 epochs（如从 100 调到 500）
 - SVR 模型：尝试不同核函数（RBF→Poly），调整 C（增大以减少欠拟合，减小以减少过拟合）
 - 交叉验证：减小 cv_horizon（从 5 调到 1）以获得更保守的泛化评估
4. 重新分析：点击“开始分析”再次运行，对比新旧结果
5. 模型对比：
 - 在结果表格中对比不同模型的 RMSE、R²
 - 在预测曲线图中直观对比拟合效果（曲线是否贴近实际点）
 - 在残差图中对比残差分布（哪个模型的残差更随机）
6. 选型决策：
 - 若某模型 RMSE 最小但 R² 不高，可能数据噪声较大
 - 若多个模型 RMSE 接近，优先选择参数少、可解释性强的模型（如 G0 优于 BP）
 - 若交叉验证 RMSE 显著高于训练 RMSE，表示过拟合，建议简化模型或增加数据

高级技巧：

- 数据预处理：若原始数据波动剧烈，可在 Excel 中先做平滑处理（移动平均）
- 分段建模：若数据呈现明显阶段性（如前期快速增长、后期平缓），可分段分析
- 集成预测：取多个模型预测值的加权平均（需手工实现，系统未内置）

5.4 有关的处理

描述批处理、脱机处理或后台处理。

批处理分析： 系统支持通过 CLI 工具进行批量数据分析，适用于需要处理多个数据集的场景。

批处理脚本示例（PowerShell）：

```
# 批量分析多个数据文件
$dataFiles = Get-ChildItem "D:\failure_data" -Filter *.csv
foreach ($file in $dataFiles) {
    $reportName = $file.BaseName + "_report.pdf"
    uv run zdp-cli $file.FullName --report "D:\reports\$reportName"
    Write-Host "已完成: $($file.Name)"
}
```

批处理配置：

- 可将模型选择、参数配置等写入脚本，实现标准化分析流程
- 使用--model 参数指定统一的模型集，如--model go --model jm --model s
- 使用--walk-forward 统一启用交叉验证
- 输出日志重定向到文件：uv run zdp-cli data.csv > analysis.log 2>&1

脱机处理：

- 系统为纯离线桌面应用，无需联网即可完整运行
- 数据文件、报告文件均保存在本地磁盘
- 适合在安全隔离环境（如涉密实验室）中使用

- 若需在无 Python 环境的机器上运行，可使用 PyInstaller 打包的 zdp.exe 单文件版本

后台处理：

- GUI 后台线程：分析任务在 QThread 中运行，主界面保持响应，用户可查看日志或准备下一次分析
- CLI 后台任务（PowerShell）：

```
# 使用Start-Job在后台运行
Start-Job -ScriptBlock {
    uv run zdp-cli "D:\\large_data.csv" --report "D:\\report.pdf"
}
Get-Job # 查看任务状态
Receive-Job -Id 1 # 获取输出
```

- 后台处理注意事项：

- 大数据集 (>10000 点) 建议使用 CLI 后台模式，避免占用 GUI
- 神经网络训练 (BP 模型) 可能需 5-10 分钟，适合后台运行
- 后台任务无进度条，建议结合日志监控 (`tail -f analysis.log`, Windows 使用 `Get-Content -Wait`)

并行处理（未来版本）：

- 当前版本模型拟合为串行执行（一个接一个）
- 未来版本计划支持多进程并行拟合，利用多核 CPU 加速（需处理 PyTorch 的 multiprocessing 兼容性）

定时任务：

- 配合 Windows 任务计划程序（Task Scheduler）可实现定期自动分析：
 1. 打开“任务计划程序”
 2. 创建基本任务，触发器设为“每周一上午 9 点”
 3. 操作设为“启动程序”，程序路径填写 PowerShell 脚本路径（调用 `zdp-cli`）
 4. 保存任务，系统将自动定期执行分析并生成报告

5.5 数据备份

描述备份规程及恢复数据的准备。

数据备份对象：

1. 原始数据文件 (CSV/Excel)：
 - 位置：用户指定的数据目录（如 D:\failure_data\）
 - 重要性：最高，丢失无法恢复
 - 建议备份频率：每日或每次修改后
2. 分析报告 (PDF)：
 - 位置：用户指定的报告输出目录
 - 重要性：中等，可从原始数据重新生成
 - 建议备份频率：每周或项目节点
3. 实验归档 (ZIP)：
 - 位置：实验导出目录
 - 重要性：高，用于审计与复现
 - 建议备份频率：每次重要分析后
4. 配置文件（未来版本）：
 - 位置：用户主目录/.zdp/config.json
 - 重要性：低，主要为界面偏好设置
 - 建议备份频率：不定期

备份方式：

方式一：手工复制

```

# 创建备份目录
New-Item -ItemType Directory -Path "D:\\ZDP_Backup\\2024-12-27"

# 复制数据文件
Copy-Item "D:\\failure_data\\*" -Destination "D:\\ZDP_Backup\\2024-12-27\\data\\" -Recurse

# 复制报告
Copy-Item "D:\\reports\\*.pdf" -Destination "D:\\ZDP_Backup\\2024-12-27\\reports\\"

```

方式二：自动备份脚本

```

# backup_zdp.ps1
$backupRoot = "D:\\ZDP_Backup"
$timestramp = Get-Date -Format "yyyyMMdd_HHmmss"
$backupDir = "$backupRoot\\$timestramp"

# 创建目录结构
New-Item -ItemType Directory -Path "$backupDir\\data" -Force
New-Item -ItemType Directory -Path "$backupDir\\reports" -Force

# 复制文件
Copy-Item "D:\\failure_data\\*" -Destination "$backupDir\\data\\" -Recurse
Copy-Item "D:\\reports\\*.pdf" -Destination "$backupDir\\reports\\"
Copy-Item "D:\\experiments\\*.zip" -Destination "$backupDir\\experiments\\"

# 清理30天前的备份
Get-ChildItem $backupRoot | Where-Object {$_.LastWriteTime -lt (Get-Date).AddDays(-30)} | Remove-Item -Recurse

```

方式三：云存储同步

- 使用 OneDrive、百度网盘等云存储客户端，将数据目录与报告目录设为同步文件夹
- 优点：自动增量同步、版本历史、跨设备访问
- 注意：若涉及保密数据，需使用企业内部私有云或加密存储

方式四：Git 版本控制（适合数据文件较小的场景）

```

cd D:\\failure_data
git init
git add *.csv
git commit -m "Initial commit"
# 后续每次修改
git add -A
git commit -m "Update failure data"

```

数据恢复规程：

场景一：误删除数据文件

1. 检查回收站，若文件在回收站中，右键“还原”
2. 若已清空回收站，从最近备份目录恢复：

```

Copy-Item "D:\\ZDP_Backup\\2024-12-27\\data\\project_data.csv" -Destination "D:\\failure_data\\"

```

3. 若无备份，尝试使用数据恢复软件（Recuva、EaseUS 等）扫描磁盘

场景二：数据文件损坏

1. 尝试用记事本打开 CSV 文件，检查是否有乱码或截断
2. 若是 Excel 文件损坏，尝试“打开并修复”功能（Excel → 打开 → 选择文件 → 下拉菜单选“打开并修复”）
3. 从备份恢复最近可用版本

场景三：系统崩溃或磁盘故障

1. 将备份磁盘/云存储中的文件下载到新机器
2. 重新安装 ZDP 系统（按“软件入门”章节步骤）
3. 将恢复的数据文件导入 ZDP，验证完整性
4. 若有实验归档 ZIP，通过“实验回放”功能验证历史分析结果一致性

备份验证：

- 定期（如每月）随机抽取备份文件，尝试用 ZDP 加载，确保备份可用
- 检查备份文件大小与原始文件一致
- 对于关键数据集，计算文件哈希值（MD5/SHA256）并记录，恢复后验证一致性：

```
Get-FileHash "D:\\failure_data\\project_data.csv" -Algorithm SHA256
```

容灾建议：

- 遵循“3-2-1 备份原则”：3 份副本、2 种存储介质、1 份异地备份
- 至少每周完整备份一次，每日增量备份
- 对于关键项目，在分析前后各备份一次
- 定期演练数据恢复流程，确保团队成员熟悉操作

5.6 错误、故障和紧急情况下的恢复

详细的重启与连续性运行规程。

常见错误与恢复：

错误 1：数据加载失败

- 现象：弹出错误对话框“数据加载失败：未找到有效的值列”
- 原因：CSV 文件格式错误、编码问题、列名无法识别
- 恢复步骤：
 1. 用 Excel 或记事本打开数据文件，检查是否有空行、非数值字符
 2. 确认至少有一列包含数值数据（故障数或间隔）
 3. 若列名不标准，使用 CLI 模式显式指定：

```
uv run zdp-cli data.csv --value-column "DefectCount" --time-column "Week"
```

4. 若是编码问题（中文乱码），用 Excel 另存为 UTF-8 编码的 CSV

错误 2：模型拟合失败

- 现象：分析完成后某模型未在结果表中出现，日志显示“Jelinski-Moranda 无有限解”
- 原因：数据不满足模型数学条件（如 JM 模型的 P 值条件）
- 恢复步骤：
 1. 这是正常行为，不是软件故障
 2. 该模型自动跳过，不影响其他模型运行
 3. 若所有模型均失败，检查数据点数是否过少（<3 个点）
 4. 尝试使用其他模型（如 G0、S-Shaped 通常兼容性较好）

错误 3：程序无响应（假死）

- 现象：点击按钮无反应，窗口标题显示“（无响应）”，进度条卡住
- 原因：后台优化算法陷入长时间计算或死循环
- 恢复步骤：
 1. 等待 5 分钟，某些大数据集或神经网络训练确实耗时长
 2. 若超过 10 分钟仍无响应，按 Ctrl+Alt+Del 打开任务管理器
 3. 找到 zdp.exe 或 python.exe 进程，右键“结束任务”
 4. 重新启动程序，尝试减少模型数量（先只运行 G0、JM 等快速模型）
 5. 若反复卡死，检查数据规模（数万点可能需分批处理）

错误 4：内存不足

- 现象：程序崩溃，弹出“MemoryError”或系统提示内存不足

- 原因：数据集过大（>50000 点）或神经网络模型占用过多内存
- 恢复步骤：
 1. 关闭其他占用内存的程序（浏览器、Office 等）
 2. 重启计算机释放内存碎片
 3. 若数据确实很大，考虑数据降采样（如每 10 个点取 1 个）或分段分析
 4. 调整 BP 模型参数：减小 hidden_size（如从 64 降到 16）、减少 epochs
 5. 升级物理内存（推荐 16 GB 以处理大规模数据）

错误 5：PDF 生成失败

- 现象：点击“导出报告”后报错“ReportLab is required for PDF export”
- 原因：ReportLab 库未安装或字体缺失
- 恢复步骤：
 1. 重新安装依赖：`uv sync --all-extras`
 2. 检查是否在虚拟环境中，确保使用正确的 Python 环境
 3. 若提示字体缺失，检查 C:\Windows\Fonts 目录是否有 simsun.ttc 或 msyh.ttc
 4. 若系统为精简版 Windows，手动下载中文字体并安装

故障 6：图表无法显示

- 现象：右侧图表区域空白或显示错误图标
- 原因：Matplotlib 后端问题、显卡驱动过旧
- 恢复步骤：
 1. 更新显卡驱动（尤其 Intel 核显）
 2. 尝试切换 Matplotlib 后端（需修改代码或环境变量，高级用户）
 3. 使用 CLI 模式导出 PDF，通过 PDF 查看图表

紧急情况处理：

情况 1：系统崩溃导致分析中断

- 恢复规程：
 1. 重启 ZDP 程序
 2. 重新导入数据文件（系统无状态保存，需从头开始）
 3. 重新选择模型并配置参数
 4. 再次执行分析
 5. 若反复崩溃，怀疑数据问题，尝试用示例数据测试（确认软件本身无问题）

情况 2：磁盘空间不足

- 恢复规程：
 1. 检查磁盘可用空间（至少预留 2 GB）
 2. 清理临时文件：`%TEMP%\zdp_*`、`C:\Users\<user>\AppData\Local\Temp\`
 3. 删除不需要的旧报告 PDF 文件
 4. 若导出大量实验 ZIP，考虑移动到其他磁盘或云存储

情况 3：数据文件被锁定

- 现象：提示“文件被其他程序占用，无法读取”
- 恢复规程：
 1. 关闭所有打开该 CSV/Excel 文件的程序（Excel、记事本等）
 2. 若仍提示锁定，使用 Process Explorer 等工具查找占用进程并关闭
 3. 复制文件到新位置并重命名，尝试加载副本

连续性运行保障：

- 对于需要长时间稳定运行的批处理任务：
 1. 使用 CLI 模式代替 GUI（更稳定，内存占用小）
 2. 使用 `try-catch` 包裹脚本，失败时自动重试：

```

$maxRetries = 3
$attempt = 0
while ($attempt -lt $maxRetries) {
    try {
        uv run zdp-cli data.csv --report out.pdf
        break
    } catch {
        $attempt++
        Write-Host "尝试 $attempt 失败, 重试..."
        Start-Sleep -Seconds 10
    }
}

```

3. 监控日志文件，设置告警（如日志中出现“Error”关键词时发送邮件）
4. 定期检查进程状态，若进程异常退出则自动重启

灾难恢复：

- 若系统完全无法使用（如硬件故障、操作系统损坏）：
 1. 从备份恢复数据文件到新机器
 2. 重新安装 ZDP 系统（按“安装和设置”章节）
 3. 从实验归档 ZIP 中恢复历史分析配置
 4. 重新运行关键分析任务，对比结果一致性
 5. 更新备份策略，避免未来重复故障

5.7 消息

列出错误消息、诊断消息的含义及应对动作。

数据加载消息：

消息内容	类型	含义	应对动作
“数据加载成功：100 条记录（累计故障数）”	信息	数据正常导入，识别为累计故障数类型	无需操作，继续选择模型
“数据加载成功：50 条记录（故障间隔 TBF）”	信息	数据正常导入，识别为 TBF 类型	注意只能运行 JM 等 TBF 兼容模型
“数据加载失败：未找到有效的值列”	错误	CSV 文件无数值列或列名无法识别	检查文件格式，或使用 -value-column 显式指定
“数据加载失败：Input file contains no rows”	错误	文件为空或只有表头	检查文件内容，确保有数据行
“Unsupported file extension: .doc”	错误	文件格式不支持	转换为 CSV 或 Excel 格式
“文件路径不存在或无法访问”	错误	文件路径错误或权限不足	检查路径拼写，确认文件存在及可读

模型拟合消息：

消息内容	类型	含义	应对动作
“正在拟合 Goel-Okumoto...”	信息	模型拟合进行中	等待完成
“Goel-Okumoto 拟合完成，RMSE=2.35”	信息	模型成功拟合	查看结果表格与图表
“Jelinski-Moranda 无有限解，已跳过”	警告	JM 模型数学条件不满足 ($P \leq (n-1)/2$)	正常情况，尝试其他模型

“JM model requires at least 2 failures”	错误	数据点数不足	增加数据量或使用其他模型
“Optimization failed: maxfev exceeded”	警告	优化算法未收敛（迭代次数超限）	模型可能不适合该数据，检查拟合质量
“SVR training failed: X has 0 features”	错误	数据维度错误（内部逻辑问题）	报告给技术支持，可能是软件 Bug

分析结果消息：

消息内容	类型	含义	应对动作
“分析完成，共 5 个模型排名”	信息	所有选中模型拟合完成	查看结果并导出报告
“分析完成。最佳模型：Goel-Okumoto (RMSE=2.35)”	信息	最佳模型为 G0, RMSE 为 2.35	可直接使用该模型进行预测
“警告：所有模型均拟合失败”	错误	无任何模型成功	检查数据质量（点数、数值范围、类型）
“分析失败：数据点数不足，至少需要 3 个数据点”	错误	数据量太少无法建模	收集更多数据
“Analysis interrupted by user”	信息	用户按 Ctrl+C 中断 CLI 命令	重新运行或检查配置

PDF 导出消息：

消息内容	类型	含义	应对动作
“报告已保存至：D:\Reports\zdp_report.pdf”	信息	PDF 导出成功	打开 PDF 查看完整报告
“PDF 报告导出成功”	信息	状态栏提示导出完成	无需操作
“ReportLab is required for PDF export”	错误	ReportLab 库未安装	运行 uv sync --all-extras 重新安装
“无法创建 PDF：权限被拒绝”	错误	输出目录无写权限	更换输出路径或以管理员权限运行
“字体注册失败，PDF 中文可能显示为方块”	警告	系统缺少中文字体	安装微软雅黑或宋体字体文件

交叉验证消息：

消息内容	类型	含义	应对动作
“启用 Walk-Forward 验证，最小训练集=50”	信息	交叉验证已启用	查看 cv_rmse 等指标
“交叉验证跳过：数据点数不足”	警告	数据量<最小训练集要求	减小 cv_min_train 或增加数据
“cv_rmse=3.45 (泛化误差)”	信息	交叉验证 RMSE 为 3.45	对比训练 RMSE，评估过拟合程度

系统级消息：

消息内容	类型	含义	应对动作

“ImportError: No module named ‘torch’”	错误	PyTorch 未安装	运行 <code>uv sync --all-extras</code> 或 <code>pip install torch</code>
“MemoryError”	错误	内存耗尽	关闭其他程序、减少数据量、增加物理内存
“FileNotFoundException: [Errno 2] No such file or directory”	错误	文件路径错误	检查路径拼写，使用绝对路径
“Permission denied”	错误	权限不足	以管理员权限运行或更换目录
“程序已停止工作”	错误	程序崩溃（Windows 对话框）	重启程序，若反复崩溃报告技术支持

5.8 快速参考指南

常用功能键、控制序列或命令的快速索引。

GUI 操作快捷键（未来版本，当前版本主要通过鼠标操作）：

快捷键	功能	说明
Ctrl+O	打开数据文件	弹出文件选择对话框
Ctrl+S	保存报告	导出 PDF 到默认路径
Ctrl+E	导出实验	打包当前分析为 ZIP
Ctrl+R	开始分析	执行模型拟合
Ctrl+Q	退出程序	关闭主窗口
F1	帮助文档	打开用户手册
F5	刷新数据	重新加载当前数据文件

CLI 常用命令速查：

```
# 基本用法
uv run zdp-cli data.csv

# 指定列名
uv run zdp-cli data.csv --time-column t --value-column failures

# 选择特定模型
uv run zdp-cli data.csv --model go --model jm --model s

# 导出PDF报告
uv run zdp-cli data.csv --report output.pdf

# 启用交叉验证
uv run zdp-cli data.csv --walk-forward --cv-min-train 50 --cv-horizon 1

# 计算预测区间
uv run zdp-cli data.csv --prediction-interval-alpha 0.05

# 按交叉验证RMSE排序
uv run zdp-cli data.csv --walk-forward --rank-by cv_rmse

# 加载插件模型
uv run zdp-cli data.csv --include-plugins
```

```

# 导出实验归档
uv run zdp-cli data.csv --export-experiment experiment.zip

# 回放实验
uv run zdp-cli --load-experiment experiment.zip --report replay_report.pdf

# BP模型参数调优
uv run zdp-cli data.csv --model bp --bp-hidden 64 --bp-epochs 500 --bp-lr 0.005

# SVR模型参数调优
uv run zdp-cli data.csv --model svr --svr-kernel poly --svr-c 100 --svr-epsilon 0.001

# 查看帮助
uv run zdp-cli --help

```

模型标识符速查:

标识符	完整名称	适用数据类型	主要参数
jm	Jelinski-Moranda	TBF	N0 (初始缺陷数)、Φ (故障率)
go	Goel-Okumoto	累计故障数	a (最终缺陷数)、b (发现率)
s	S-Shaped	累计故障数	a、b
gm	GM(1, 1)	累计故障数	a、b (灰色参数)
bp	BP Neural Network	累计故障数	hidden_size、epochs、lr
svr	Support Vector Regression	累计故障数	kernel、C、epsilon
hybrid	EMD Hybrid	累计故障数	svr_c、svr_epsilon

指标含义速查:

指标	全称	取值范围	优劣判断
RMSE	Root Mean Square Error	[0, +∞)	越小越好
MAE	Mean Absolute Error	[0, +∞], [越小越好], [R ²], [Coefficient of Determination], [(-∞, 1]	越接近 1 越好
MAPE	Mean Absolute Percentage Error	[0%, +∞)	越小越好
cv_rmse	Cross-Validated RMSE	[0, +∞], [越小越好 (泛化能力)], [cv_r2], [Cross-Validated R ²], [(-∞, 1]	越接近 1 越好

文件格式速查:

格式	扩展名	支持导入	支持导出	说明
CSV	.csv, .txt	✓	-	逗号分隔值, UTF-8 编码推荐
TSV	.tsv	✓	-	制表符分隔值

Excel	.xls, .xlsx	✓	-	Microsoft Excel 工作簿
PDF	.pdf	-	✓	分析报告输出
ZIP	.zip	✓ (实验)	✓ (实验)	实验归档文件

常见问题速查：

问题	快速解决方案
启动报错 “Python not found”	安装 Python 3.10+ 并添加到 PATH
数据列识别错误	使用 -value-column 和 -time-column 显式指定
BP 模型训练太慢	减少 -bp-epochs (如改为 100) 或 hidden_size
PDF 中文显示方块	安装 Windows 中文字体 (simsun.ttc/msyh.ttc)
所有模型拟合失败	检查数据点数 ≥ 3 、数值非空、类型正确
内存不足崩溃	减少数据量、关闭其他程序、使用 CLI 模式
JM 模型无结果	正常，数据不满足 JM 解的存在条件
交叉验证 RMSE 远大于训练 RMSE	模型过拟合，简化模型或增加数据

系统要求速查：

- 操作系统：Windows 10/11 (64 位)
- Python：≥3.10
- 内存：≥4 GB (推荐 8 GB)
- 磁盘：≥2 GB 可用空间
- 依赖库：PySide6、NumPy、Pandas、SciPy、Matplotlib、scikit-learn、PyTorch、ReportLab

技术支持联系：

- 内部 Wiki: <https://github.com/4627488/zdp/wiki>
- Bug 报告: <https://github.com/4627488/zdp/issues>

6 注释

包括背景、术语、缩略语或公式。

6.1 术语与定义

软件可靠性 (Software Reliability)： 软件在规定的条件下、规定的时间内，完成规定功能的能力。通常用平均故障间隔时间 (MTBF) 或故障率 $\lambda(t)$ 量化。

可靠性增长 (Reliability Growth)： 软件在测试与调试过程中，随缺陷被发现和修复，可靠性逐步提升的过程。表现为故障间隔增大、累计故障数增长速率减缓。

累计故障数 (Cumulative Failures)： 从测试开始到时刻 t ，已发现的总故障次数，记为 $m(t)$ 或 $N(t)$ 。单调非递减函数。

故障间隔 (Time Between Failures, TBF)： 相邻两次故障发生的时间间隔，记为 $x_i = t_i - t_{i-1}$ 。JM 模型等基于 TBF 建模。

非齐次泊松过程 (Non-Homogeneous Poisson Process, NHPP)： 故障发现过程服从泊松分布，但强度函数 $\lambda(t)$ 随时间变化。G0 模型、S-Shaped 模型属于 NHPP 类。

均方根误差 (Root Mean Square Error, RMSE)： 预测值与实际值差异的平方均值的平方根，公式为：

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

其中 y_i 为实际值, \hat{y}_i 为预测值, n 为样本数。

决定系数 (Coefficient of Determination, R^2) : 衡量模型拟合优度的指标, 公式为:

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}} = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

其中 \bar{y} 为实际值均值。 R^2 接近 1 表示拟合优秀, 接近 0 表示模型无效。

残差 (Residual) : 实际值与预测值之差, 记为 $e_i = y_i - \hat{y}_i$ 。残差分析用于检验模型假设是否成立。

交叉验证 (Cross-Validation, CV) : 将数据集分为训练集与测试集, 用训练集拟合模型, 测试集评估泛化能力。本系统采用 Walk-Forward (滚动窗口) 方式, 模拟实际预测场景。

过拟合 (Overfitting) : 模型在训练数据上表现优秀, 但在新数据上泛化能力差。表现为训练 RMSE 远小于交叉验证 RMSE。

预测区间 (Prediction Interval) : 未来观测值的置信区间, 通常基于残差标准差构建, 如 95% 预测区间为 $[\hat{y} - 1.96\sigma, \hat{y} + 1.96\sigma]$ 。

6.2 模型公式

Goel-Okumoto 模型:

$$m(t) = a(1 - e^{-bt})$$

- a : 预期最终发现的缺陷总数 (渐近值)
- b : 缺陷发现率 (单位时间发现缺陷的能力)
- $m(t)$: 时刻 t 的累计故障数
- 适用于累计故障数单调递增、发现率递减的场景

Jelinski-Moranda 模型: 故障强度函数:

$$\lambda(t) = \varphi[N_0 - (i - 1)]$$

第 i 个故障间隔期望:

$$E(x_i) = \frac{1}{\varphi(N_0 - i + 1)}$$

- N_0 : 软件初始缺陷总数
- φ : 单个缺陷的故障率 (比例常数)
- x_i : 第 i 个故障间隔
- 假设每修复 1 个缺陷, 剩余缺陷数减 1, 故障强度线性递减

S-Shaped 模型:

$$m(t) = a(1 - (1 + bt)e^{-bt})$$

- 参数含义同 G0 模型, 但增长曲线呈 S 型
- 适合初期缓慢、中期快速、后期饱和的学习效应场景

灰色模型 GM(1, 1): 微分方程形式:

$$\frac{dx^{(1)}}{dt} + ax^{(1)} = b$$

解析解:

$$\hat{x}^{(1)}(k+1) = \left(x^{(0)}(1) - \frac{b}{a} \right) e^{-ak} + \frac{b}{a}$$

- $x^{(0)}$: 原始序列
- $x^{(1)}$: 一次累加序列 (AGO)
- a 、 b : 模型参数 (通过最小二乘估计)
- 适合小样本、短时间序列预测

6.3 缩略语汇总

缩略语	英文全称	中文译名
ZDP	Zero-Defect Prediction	零缺陷预测
TBF	Time Between Failures	故障间隔
NHPP	Non-Homogeneous Poisson Process	非齐次泊松过程
GO	Goel-Okumoto	戈尔-奥库莫托模型
JM	Jelinski-Moranda	杰林斯基-莫兰达模型
GM	Grey Model	灰色模型
BP	Back Propagation	反向传播 (神经网络)
SVR	Support Vector Regression	支持向量回归
EMD	Empirical Mode Decomposition	经验模态分解
RMSE	Root Mean Square Error	均方根误差
MAE	Mean Absolute Error	平均绝对误差
MAPE	Mean Absolute Percentage Error	平均绝对百分比误差
MSE	Mean Square Error	均方误差
R ²	Coefficient of Determination	决定系数
CV	Cross-Validation	交叉验证
PDF	Portable Document Format	便携式文档格式
CSV	Comma-Separated Values	逗号分隔值
GUI	Graphical User Interface	图形用户界面
CLI	Command-Line Interface	命令行界面
API	Application Programming Interface	应用程序编程接口
MLE	Maximum Likelihood Estimation	最大似然估计

6.4 背景知识

软件可靠性工程发展： 软件可靠性研究始于 20 世纪 70 年代，早期模型如 Jelinski-Moranda (1972)、Goel-Okumoto (1979) 基于缺陷消耗或泊松过程假设，广泛应用于航天、国防等高可靠性领域。21 世纪后，机器学习方法（神经网络、SVR）逐步引入，能够处理非线性、非平稳数据，但可解释性较传统模型弱。本系统同时集成经典模型与现代方法，兼顾理论严谨性与实用灵活性。

数据类型选择依据：

- 累计故障数：适合故障记录完整、时间戳准确的场景（如自动化测试日志、现场缺陷跟踪系统）
- 故障间隔：适合间隔时间明确、关注可靠性变化趋势的场景（如系统运行日志、间歇性测试）
- 两种类型可相互转换，但 JM 模型等专为 TBF 设计，GO 模型等专为累计数设计，需注意兼容性

模型选型建议：

- 数据量少 (<20 点)：优先 GM(1, 1)、GO、S-Shaped
- 数据量中等 (20–200 点)：GO、JM、S-Shaped、SVR

- 数据量大 (>200 点)：BP 神经网络、SVR、Hybrid
- 数据平稳单调：GO、S-Shaped
- 数据波动大：Hybrid (EMD 降噪)、SVR
- 需要可解释性：GO、JM (参数有明确物理意义)
- 仅需预测准确：BP、SVR (黑箱模型，精度可能更高)

未来发展方向：

- 集成更多模型 (如 Weibull、对数泊松、Musa-Okumoto 等)
- 支持深度学习模型 (LSTM、Transformer 时间序列预测)
- 增加模型可解释性工具 (SHAP 值、敏感性分析)
- 实现在线学习 (增量更新模型参数)
- 提供 REST API 接口，支持 CI/CD 集成

6.5 相关标准与规范

国际标准：

- IEEE 982.1-2005：软件可靠性度量标准字典
- ISO/IEC 25010：系统与软件质量模型 (含可靠性特性定义)
- IEC 61508：功能安全国际标准 (软件安全完整性等级)

国内标准：

- GB/T 14394-2008：计算机软件可靠性和可维护性管理
- GJB 438C-2019：军用软件开发文档通用要求
- GJB 5236-2004：软件可靠性预计模型及应用指南

参考文献：

- Musa, J. D., Iannino, A., Okumoto, K. (1987). Software Reliability: Measurement, Prediction, Application. McGraw-Hill.
- Lyu, M. R. (Ed.). (1996). Handbook of Software Reliability Engineering. IEEE Computer Society Press.
- Xie, M. (1991). Software Reliability Modelling. World Scientific.
- 张德平;软件系统可靠性分析基础与实践 (Fundamentals and Practice of Software System Reliability Analysis) 清华大学出版社