ICS 03.120.01
CCS L 05

T/CICC

中 国 指 挥 与 控 制 学 会 团 体 标 准

T/ CICC 35019-2025

复杂智能系统测试性技术要求

Technical requirements for testability of complex intelligent systems

2025-11-20 发布　　2025-11-20 实施

中国指挥与控制学会　发布

目　　次

前　　言

本文件按照 GB/T 1.1－2020《标准化工作导则　第 1 部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由中国指挥与控制学会提出并归口。

本文件起草参与单位：北京航空航天大学、杭州市北京航空航天大学国际创新研究院（北京航空航天大学国际创新学院）、可靠性与环境工程技术国家级重点实验室、北京航空航天大学可靠性工程研究所、中国科学院声学研究所、中国船舶集团有限公司综合技术经济研究院、中国兵器工业软件工程与评测中心、中国电子科技集团公司信息科学研究院、北京智臻领航科技有限公司、中国航空研究院。

本文件主要起草人：杨顺昆、刘磊、郝程鹏、徐珞、王若、司昌龙、庞红彪、冯玲玲、王树泰、郝威巍、包超鹏、冯润玉、代国良、李乐晓、段峙宇、林焱辉、姜巍、吴梦丹、周怡婧、蒋亮亮、王榆伟。

复杂智能系统测试性技术要求

1 范围

本文件适用于面向复杂智能系统的测试性评估、设计和测试。

本文件规定了智能系统通用质量特性中关于智能系统测试性的技术要求，涵盖了定性与定量两类指标，建立了智能系统可测试性的度量框架、评价方法与实施流程，用于描述复杂智能系统通用质量特性中有关智能系统测试性方面的相关指标以及测试方法。

本文件通过引入可观测性、可控性、可分解性、可隔离性、可重现性、可诊断性、可覆盖性、可自动化性及数据可测性等质量子特性，形成了系统化的测试性指标体系，用于指导智能系统的设计、开发、部署、运行与退役各阶段的测试性保障。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 25000　　系统与软件工程 系统与软件质量要求和评价

GB/T 35273　　信息技术安全 个人信息安全规范

GB/T 36344-2018　　信息技术 数据质量评价指标

GB/T 42018　　信息技术 人工智能 平台计算资源规范

GB/T 42018-2022　　智能制造 远程运维系统通用要求

GB/T 44662-2024　　健康管理 终端设备数据采集与传输协议

GB/T 45288.2-2025　　人工智能 大模型 第 1 部分：通用要求

GB/T 45087-2024　　人工智能 服务器系统性能测试方法

3 术语与定义

3.1

智能系统测试性　testability

系统在给定资源下被有效、充分、可重复测试的程度。

3.2

智能系统可观测性　observability

系统内部状态及行为通过日志、指标、可解释机制被外部探测的能力。

3.3

智能系统可控性　controllability

测试输入、参数、环境、随机种子等可被设置和复现实验条件的能力。

3.4

智能系统可分解性　decomposability

将整体系统拆分为可独立测试评估的阶段/模块能力。

3.5

智能系统可隔离性　isolability

测试执行与生产运行互不干扰，并在故障发生时能限制影响范围并快速恢复的能力。

3.6

智能系统可重复性　　reproducibility

在相同配置与资源条件下，可得到在统计容差内一致的训练/推理结果的能力。

3.7

智能系统可诊断性　　diagnosability

快速定位数据、特征、结构、资源、对抗等部分存在的缺陷，并定位性能退化根因的能力。

3.8

智能系统可覆盖性　　coverability

建立和量化数据、场景、特征、模型内部行为、攻击空间等多维测试覆盖的能力。

3.9

智能系统可自动化性　automatability

可自动化性是指测试活动在流水线（CI/CD）中自动执行与报告的程度。

3.10

智能系统数据可测性　data testability

对数据质量、偏见、漂移、标签一致性及生成工艺进行系统化测试与持续监控的能力。

3.11

对抗样本　　adversarial example

对输入进行微小扰动仍保持人类感知语义却诱使模型产生错误输出的样本。

3.12

偏见　　bias/fairness

算法输出在不同受保护群体之间的不合理差异。

3.13

隐私　　privacy

个人所具有的控制或影响与健康相关信息的权限，涉及由谁收集和存储、由谁披露。

[来源：GB/T 44662-2024，3.2]

3.14

测试对象层　　test object layer

数据、特征、模型、服务/API、集成系统、运行与监控层。

3.15

人工智能平台　　artificial intelligence platform

为人工智能应用提供各类资源的软硬件系统。

[来源：GB/T 42018-2022，3.1]

3.16

模型压缩　　model compression

通过算法优化减少机器学习模型的参数量或计算复杂度，以提升其在资源受限环境（如边缘设备）中的部署效率。

3.17

容错机制　　fault-tolerant mechanism

在硬件或软件发生故障时，系统通过冗余设计、状态备份或动态切换保持服务连续性的技术方案。

3.18

备件　　spare parts

为保证失效部件或设备得到替换，所预先准备的，能够提供正常功能的部件或设备。

[来源：GB/T 42136-2022，3.3]

3.19

任务　　task

被调度的训练或推理对象。

[来源：GB/T 45288.1-2025，3.3]

3.20

性能指标　　performance indicator

用于评估人工智能服务器系统实现效果的度量。

[来源：GB/T 45087-2024，3.19]

4　缩略语

下列缩略语适用于本文件。

AI——Artificial Intelligence 人工智能；

CPU——Central Processing Unit 中央处理器；

GPU——Graphics Processing Unit 图形处理器；

NPU——Neural Processing Unit 神经网络处理器；

CI/CD——Continuous Integration / Continuous Delivery 持续交付持续部署；

TI——Testability Index 测试性指标；

API——Application Programming Interface 应用程序编程接口；

TPR——True Positive Rate 真正率；

FGSM——Fast Gradient Sign Method 快速梯度符号法；

PGD——Projected Gradient Descent 投影梯度下降；

CNN——Convolutional Neural Network 卷积神经网络；

RNN——Recurrent Neural Network 循环神经网络；

LSTM——Long Short-Term Memory 长短期记忆网络；

GNN——Graph Neural Network 图神经网络；

RL——Reinforcement Learning 强化学习；

GAN——Generative Adversarial Network 生成对抗网络；

VAE——Variational Autoencoder 变分自编码器；

LLM——Large Language Model 大语言模型；

SHAP——SHapley Additive exPlanations 沙普利加性解释；

LIME——Local Interpretable Model-agnostic Explanations 局部可解释模型无关解释。

5　测试对象类型与测试层次

针对不同测试对象，其执行测试活动见表 1。

表 1 测试对象类型与测试活动映射表格

测试对象类型	范围定义	测试关注点	典型测试活动
数据层	包含原始数据、特征工程处理结果、标注标签等	数据合法性、完整性、一致性、偏见与代表性	数据来源审计、数据质量测试、偏见与公平性分析、分布覆盖率度量
模型层	包含模型结构、参数权重、可解释性接口	结构正确性、参数有效性、可解释性、一致性	权重检查与验证、模型可解释性接口调用测试、鲁棒性与安全性评估
应用层	包含对外提供推理或数据访问的接口	接口契约一致性、鉴权机制、流量控制	API 测试、权限测试、限流策略测试、性能与稳定性测试

表 1 测试对象类型与测试活动映射表格（续）

测试对象类型	范围定义	测试关注点	典型测试活动
运行环境层	包含硬件加速器、容器、任务调度系统等	硬件兼容性、虚拟环境一致性、调度可靠性	硬件驱动适配测试、容器镜像一致性验证、调度策略压力测试
系统层	包含 AI 模型与外部业务逻辑、缓存、消息队列等的集成	模块间数据交互正确性、延迟与吞吐、容错性	系统集成测试、接口边界测试、消息一致性测试

5.1 数据层

数据层测试旨在确保输入数据的合法性、质量和代表性。测试过程应覆盖数据来源审计、隐私保护检测（脱敏和加密措施）、数据质量度量，并通过统计分析识别偏见风险和长尾群体，建立数据分布与覆盖率基线，以支持后续的公平性和鲁棒性评估。

5.2 模型层

模型层测试重点在于验证模型结构设计与实现的正确性、参数权重的有效性以及可解释性接口的可靠性。应分析检查网络层、算子和参数规模是否符合设计要求，执行权重完整性与一致性校验，调用可解释性接口验证结果输出的稳定性与准确性，同时在该层进行初步鲁棒性、安全性和性能评估。

5.3 应用层

应用层测试关注对外接口的正确性、安全性和性能。测试应包括 API 一致性检查、鉴权与权限管理验证、限流与配额策略验证，以及高并发场景下的吞吐、延迟与可用性评估，确保在不同负载条件下服务稳定可靠。

5.4 运行环境层

运行环境测试旨在确保模型与系统在目标硬件与软件环境中的正常运行，包括 GPU/TPU 等硬件加速器的兼容性验证、驱动与固件的适配测试、容器镜像一致性检查，以及调度策略在不同任务分配和资源竞争条件下的稳定性与公平性评估。

5.5 系统层

系统层测试聚焦 AI 模型与外部业务逻辑、数据库、缓存、消息队列等模块的集成与交互。应通过系统集成测试验证数据传递的正确性、消息一致性、接口边界条件的处理能力，同时评估整体系统在高负载、网络抖动和模块故障情况下的容错能力和恢复机制。

6 测试性指标体系

测试性指标体系见图 1。

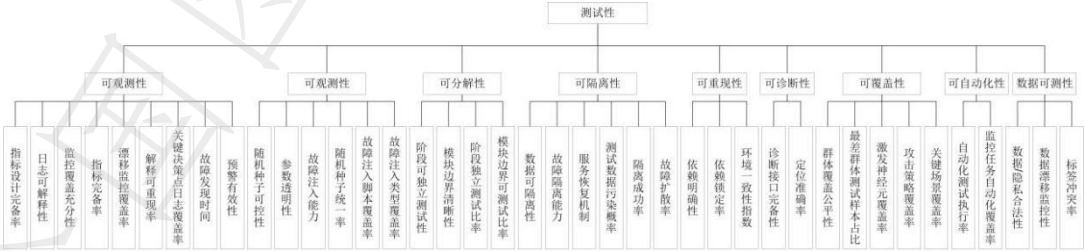

图 1 指标体系结构图

6.1 可观测性

可观测性是指对模型训练、推理运行时的内部状态、性能、漂移、解释结果的可获取与解析能力，以及故障或异常被发现与确认的效率。

6.2 可控性

可控性是指对输入、参数、随机性、资源与故障注入的可设定与复现能力。

6.3 可分解性

可分解性是指将整体系统拆分为可独立测试评估的阶段/模块能力。

6.4 可隔离性

可隔离性是指测试执行与生产运行互不干扰，并在故障发生时能限制影响范围并快速恢复的能力

6.5 可重现性

可重现性是指在相同配置与资源条件下，可得到在统计容差内一致的训练/推理结果的能力。

6.6 可诊断性

可诊断性是指快速定位数据、特征、结构、资源、对抗等部分存在的缺陷，并定位性能退化根因的能力。

6.7 可覆盖性

可覆盖性是指建立和量化数据、场景、特征、模型内部行为、攻击空间等多维测试覆盖的能力。

6.8 可自动化性

可自动化性是指测试活动在流水线（CI/CD）中自动执行与报告的程度。

6.9 数据可测性

数据可测性是指对数据质量、偏见、漂移、标签一致性及生成工艺进行系统化测试与持续监控的能力。

7 复杂智能系统测试性定性指标要求

7.1 可观测性定性指标要求

可观测性定性指标包括：

a) 指标设计完备性：系统应在设计阶段完成全面、体系化的监控指标规划，确保指标覆盖性能、可靠性、安全性、偏见与公平性等核心维度，无关键领域遗漏，并能准确反映系统健康度与业务目标；

b) 日志可解析性：系统日志必须采用标准化的结构化格式，确保日志内容机器可读、语义清晰；每条日志应包含唯一追踪标识，支持请求在跨服务、跨模块间的完整链路追踪与关联分析；

c) 监控覆盖充分性：系统须对所有核心业务特征、关键模型行为与决策边界实施实时、持续监控；监控机制应能及时捕捉异常波动与潜在风险，确保对系统核心功能与输出质量的有效洞察。

7.2 可控性定性指标要求

可控性定性指标包括：

a) 随机种子可控性：系统必须在所有随机性模块中支持全局随机种子的统一配置与管理，确保在相同输入与配置下，实验过程与结果具备完全的确定性与可复现性；

b) 参数透明性：系统所有关键超参数与配置项必须被完整、清晰地文档化，说明其定义、取值范围及对系统行为的影响；同时，这些参数应提供安全、便捷的配置接口，支持在不修改代码的前提下进行调整与生效；

c) 故障注入能力：系统测试环境须具备主动的、可控的故障注入机制，能够模拟数据链路、模型服务及底层基础设施的各类异常，以系统性地验证系统的容错能力、自愈能力与整体韧性。

7.3 可分解性定性指标要求

可分解性定性指标包括：

a) 阶段可独立测试性：系统架构须支持其各个生命周期阶段能够被隔离并独立地进行验证；每个阶段都应提供独立的测试入口、模拟数据和验证工具，确保无需依赖或完整运行上下游阶段即可完成该阶段的功能与性能测试；

b) 模块边界清晰性：系统的核心模块必须通过严格定义的接口进行交互，每个模块都应具备明确、稳定且文档化的输入/输出数据契约；模块间应实现高内聚、低耦合，确保其职责单一、依赖关系清晰，并能支持独立的开发、测试与部署。

7.4 可隔离性定性指标要求

可隔离性定性指标包括：

a) 数据可隔离性：智能系统必须实现测试数据与生产数据的严格物理或逻辑隔离，确保测试活动不会访问、污染或泄露任何生产数据，并建立明确的数据访问权限控制与审计机制；

b) 故障隔离能力：系统架构应设计有自动化的故障检测与熔断机制，当任一组件或模块发生故障时，系统能自动将其隔离，防止故障在系统中扩散，确保局部故障不影响整体服务的可用性；

c) 服务恢复机制：系统须具备快速、可靠的服务恢复能力，包括自动化的回滚机制，能在发布或升级失败时迅速恢复到上一个稳定版本、以及冗余部署策略，确保在单个服务实例或节点失效时，能无缝切换到备用资源，保障业务连续性。

7.5 可重现性定性指标要求

可重现性定性指标包括：

a) 依赖明确性：系统的所有外部依赖必须进行严格的版本管理，通过依赖清单或容器化等技术精确锁定并记录其具体版本与配置，确保整个软件环境在任何时候均具备一致性，从而实现构建结果的完全可重现。

7.6 可诊断性定性指标要求

可诊断性定性指标包括：

a) 诊断接口完备性：系统必须为其核心服务与组件提供标准化、统一化的诊断接口与工具集，确保运维与开发人员能够通过这些接口，全面、高效地获取系统内部状态，快速定位与诊断故障根因。

7.7 可覆盖性定性指标要求

可覆盖性定性指标包括：

a) 群体覆盖公平性：系统的测试策略与用例设计必须系统性地识别并覆盖所有关键用户群体、子群体及代表性不足的边缘场景，确保测试能有效揭示并验证系统在不同群体上的性能表现与行为差异，从而保障输出结果的公平性，消除因数据或模型偏差导致的歧视风险。

7.8 可自动化性定性指标要求

可自动化性定性指标包括：

a) 输入/输出可结构化程度：指完成某任务所需的全部输入信息与任务产出的全部输出信息，在多大程度上可以被机器稳定地表示、解析、校验与交换。

b) 规则可编码性：指为达成某任务目标而依赖的业务逻辑，能在多大程度上被形式化表述并以程序实现，使其在给定输入与边界条件下可计算、可验证、可复现。

7.9 数据可测性定性指标要求

数据可测性定性指标包括：

a) 数据合法性与隐私合规性：系统所有任务的输入与输出信息必须具备严格定义、机器可读的数据契约。该契约应明确定义数据的结构、类型、取值范围与语义，确保信息能够被稳定地解析、进行有效性校验、并在系统间实现无损交换，为任务的高效、可靠自动化奠定基础；

b) 数据漂移监控性：为实现任务目标的自动化，其核心业务逻辑必须能够被形式化为明确的、可执行的规则或算法。这些规则应在给定输入和边界条件下，具备确定性的计算过程、可验证的执行结果与完全的可复现性，最大限度减少对不可编码的主观判断或模糊经验的依赖。

8 复杂智能系统测试性定量指标

8.1 可观测性定量指标

可观测性定量指标包括：

a) 指标完整率：指标完整性是指在系统测试性设计中，规划出的关键监测或测试指标真正落地部署的数量。指标完整率的计算公式见式（1）。

$$X_1 = \frac{A_1}{B_1} \quad \text{.....（1）}$$

式中：

X_1——指标完整率；

A_1——已实现关键指标数；

B_1——规划指标数。

其中，已实现关键指标数是指在实际系统中已经实现、可被采集和监控的关键指标数量。这些指标必须与规划时定义的指标相对应，并且能在运行环境中实际获取到数据。规划指标数是指在测试性设计阶段列出的全部关键指标数量，它是一个事先就确定好的清单。

b) 漂移监控覆盖率：漂移监控覆盖率是指在系统可观测性设计中，有多少重要特征被实时监控其分布变化。漂移监控覆盖率的计算公式见式（2）。

$$X_2 = \frac{A_2}{B_2} \quad \text{.....（2）}$$

式中：

X_2——漂移监控覆盖率；

A_2——已布设漂移监控特征数；

B_2——关键特征数。

其中，关键特征数是指在业务需求、风险分析和模型重要性分析阶段，被识别为必须重点监控的输入特征集合。已布设漂移监控特征数是指已经在生产环境中配置并运行漂移检测机制的特征数量。

c) 解释可重现率：解释可重现率是指在相同的输入数据和相同的配置条件下，多次调用模型的可解释性方法时，解释结果保持一致的概率。解释可重现率的计算公式见式（3）。

$$X_3 = \frac{A_3}{B_3} \quad \text{.....（3）}$$

式中：

X_3——解释可重现率；

A_3——重复同输入生成解释一致次数；

B_3——总测试次数。

其中，重复同输入生成解释一致次数是指在测试中，给定一组完全相同的输入，重复执行解释过程，并得到解释内容一致的次数。总测试次数是指为这组输入样本重复生成解释的总次数。

d) 关键决策点日志覆盖率：关键决策点日志覆盖率是指系统在推理、预测、判断、业务规则执行等过程中，会显著影响最终结果或用户体验的逻辑节点。关键决策点日志覆盖率的计算公式见式（4）。

$$X_4 = \frac{A_4}{B_4} \quad \text{.....（4）}$$

式中：

X_4——关键决策点日志覆盖率；

A_4——已记录关键决策点数；

B_4——规划关键决策点数。

其中，已记录关键决策点数是指在实际运行系统中，日志模块已经覆盖并记录的关键决策点数量。日志需包含可追溯的关键信息，以满足调试、审计、合规需求。规划关键决策点数是指在设计阶段明确规划需要记录的关键决策点数量。

e) 故障发现时间；

故障发现时间即故障发生至有效告警触发的时间间隔，是指系统中某个组件、服务或模型出现性能异常、逻辑错误、安全事件等问题的实际起始时刻到监控系统检测到故障并发出符合预设条件的告警信号的实际时刻之间的间隔。

f) 预警有效性。

预警有效性是指监控告警系统在识别真实故障时的准确性和可靠性，其计算公式见式（5）。

$$X_5 = 1 - \frac{A_5}{B_5} \quad \text{.....（5）}$$

式中：

X_5——预警有效性；

A_5——误报漏报事件数；

B_5——总事件数。

8.2 可控性定量指标

可控性定量指标如下：

a) 随机种子统一率：随机种子统一率是指系统涉及随机数生成的组件统一管理的比率，其计算公式见式（6）。

$$X_6 = \frac{A_6}{B_6} \quad \text{.....（6）}$$

式中：

X_6——随机种子统一率；

A_6——受控组件数；

B_6——总需控组件数。

其中受控组件数是指已经实现随机种子固定的组件数量。总需控组件数是指设计阶段识别出的、需要统一随机种子的全部组件数量。

b) 故障注入脚本覆盖率：故障注入脚本覆盖率是用于衡量在故障测试阶段，系统在设计阶段覆盖的故障类型。其计算公式见式（7）。

$$X_7 = \frac{A_7}{B_7} \quad \text{.....（7）}$$

式中：

X_7——故障注入脚本覆盖率；

A_7——已实现故障类型数；

B_7——计划故障类型数。

其中计划故障类型数是指通过风险分析、历史故障统计、混沌工程设计等手段列出的需要模拟的全部故障类型。已实现故障类型数是指已经编写并在测试环境中可直接运行的故障注入脚本所支持的故障类型数量。

c) 故障注入类型覆盖率。

故障覆盖率是指系统在故障测试阶段，已经通过故障注入技术验证的故障类型占全部已知故障类型的比例。其计算公式见式（8）。

$$X_8 = \frac{A_8}{B_8} \quad \text{.....（8）}$$

式中：

X_8——故障注入类型覆盖率；

A_8——被注入的故障类型数；

B_8——已知故障类型总数。

其中已知故障类型总数是指通过历史故障记录、风险分析、混沌工程设计或行业经验总结出的所有已知可能影响系统稳定性或可用性的故障类型集合。被注入的故障类型数是指在测试阶段，已实际执行过故障注入验证的故障类型数量。

8.3 可分解性定量指标

可分解性定量指标包括：

a) 阶段独立测试比率：阶段独立测试比率是指在系统各生命周期阶段中，能够在不依赖整个系统运行的情况下进行独立测试的比例。

其计算公式见式（9）。

$$X_9 = \frac{A_9}{B_9} \quad \text{.....（9）}$$

式中：

X_9——阶段独立测试比率；

A_9——具独立测试脚本的阶段数；

B_9——总阶段数。

总阶段数是指在系统设计或测试规划中，按处理逻辑或生命周期划分的全部阶段数量。具独立测试脚本的阶段数是指每个阶段都拥有可单独运行的，不依赖其他阶段的输入输出链路即可进行功能验证和性能测试的测试脚本的数量。

b) 模块边界可测性比率：模块边界可测性比率是指系统中具有明确且可独立验证的输入输出接口约定的模块比例，其计算公式见式（10）。

$$X_{10} = \frac{A_{10}}{B_{10}} \quad \text{.....（10）}$$

式中：

X_{10}——模块边界可测性比率；

A_{10}——具备独立输入/输出契约的模块数；

B_{10}——模块总数。

其中，模块总数是指在系统架构中划分出的功能单元数量。具备独立输入/输出契约的模块数是指已定义了清晰的输入数据格式、输出数据格式、调用方式及约束条件，并能在脱离全链路的情况下，通过模拟输入、检查输出进行独立测试的模块数量。

8.4 可隔离性定量指标

可隔离性定量指标包括：

a) 测试数据污染概率：测试数据污染概率是用于衡量在测试过程中，生产环境数据被意外写入测试路径或测试环境的风险大小，其计算公式见式（11）。

$$X_{11} = \frac{A_{11}}{B_{11}} \quad \text{.....（11）}$$

式中：

X_{11}——测试数据污染概率；

A_{11}——生产数据不被写入测试路径事件数；

B_{11}——总测试执行。

其中，生产数据不被写入测试路径事件数是指在一次测试执行中，系统成功避免将生产数据写入测试数据库、测试存储或测试消息通道的事件次数。总测试执行是指在统计周期内的所有测试运行次数。

b) 隔离成功率：隔离成功率是用于衡量系统在面对故障时，能够准确定位并隔离问题源头的能力。

其计算公式见式（12）。

$$X_{12} = \frac{A_{12}}{B_{12}} \quad \text{.....（12）}$$

式中：

X_{12}——隔离成功率；

A_{12}——成功隔离的故障数量；

B_{12}——故障总数量。

成功隔离的故障数量是指在测试或运行过程中，系统识别出故障并采取隔离措施后，避免了故障向其他模块或系统蔓延的次数。故障总数量是指在统计周期内观测到的所有故障事件总数。

c) 故障扩散率：故障扩散率用于衡量在一次故障事件中，系统中受影响的关键组件占全部系统组件的比例。其计算公式见式（13）。

$$X_{13} = \frac{A_{13}}{B_{13}} \quad \text{.....（13）}$$

式中：

X_{13}——故障扩散率；

A_{13}——受影响的关键组件数量；

B_{13}——系统组件数量。

其中，受影响的关键组件数量是指在故障发生后，因直接或间接原因导致性能下降、功能异常或完全失效的关键组件数。系统组件数量是指系统结构中所有关键组件的总数。

8.5 可重现性定量指标

可重现性定量指标包括：

a) 依赖锁定率：依赖锁定率用于衡量系统在软件依赖管理中，对外部依赖版本的可控程度。其计算公式见式（14）。

$$X_{14} = \frac{A_{14}}{B_{14}} \quad \text{.....（14）}$$

式中：

X_{14}——依赖锁定率；

A_{14}——有版本锁定的依赖数；

B_{14}——依赖总数。

其中，有版本锁定的依赖是指能明确指定版本号、版本范围上限下限或通过锁文件固定版本的依赖项数量。依赖总数是指系统所使用的全部外部依赖数量。

b) 环境一致性指数：环境一致性指数是指分布式或多节点系统中，各节点在运行环境配置下与预设基准环境的一致程度。其计算公式见式（15）。

$$X_{15}=\frac{A_{15}}{B_{15}} \quad \text{.....（15）}$$

式中：

X_{15}——环境一致性指数；

A_{15}——与基准镜像/驱动/固件一致的节点数；

B_{15}——节点总数。

与基准一致的节点数是指在操作系统镜像、关键驱动、固件版本等方面，与基准环境完全匹配的节点数量。节点总数是指参与系统运行或测试的全部节点数量。

8.6 可诊断性定量指标

可诊断性定量指标包括：

a) 定位准确率：定位准确率是指系统在故障诊断环节中，对故障原因或位置的判断的准确性。公式见式（16）。

$$X_{16}=\frac{A_{16}}{B_{16}} \quad \text{.....（16）}$$

式中：

X_{16}——定位准确率；

A_{16}——正确诊断的故障数；

B_{16}——总故障数。

正确诊断的故障数是指在诊断结果中与实际故障原因、位置一致的案例数。总故障数是指在评估周期内发生的全部故障数量。

8.7 可覆盖性定量指标

可覆盖性定量指标包括：

a) 最差群体测试样本占比：最差群体测试样本占是指在测试集中，样本数量最少的群体所占的比例。

其计算公式见式（17）。

$$X_{17}=\frac{A_{17}}{B_{17}} \quad \text{.....（17）}$$

式中：

X_{17}——最差群体测试样本占比；

A_{17}——最小群体测试样本数；

B_{17}——总测试样本数。

其中，最小群体测试样本数是指在所有分组中测试样本数量最少的那个群体的样本数。总测试样本数是指整个测试集中的样本总数量。

b) 激发神经元覆盖率：激发神经元覆盖率是指在一次或多次测试输入中，模型内部被成功激活的神经元占总神经元的比例。其计算公式见式（18）。

$$X_{18}=\frac{A_{18}}{B_{18}} \quad \text{.....（18）}$$

式中：

X_{18}——激发神经元覆盖率；

A_{18}——触发激活阈值的神经元数；

B_{18}——总神经元数。

其中，触发激活阈值的神经元数是指在测试过程中，输出值超过预设激活阈值的神经元数量。总神经元数是指模型所有层中神经元的总数。

c) 攻击策略覆盖率：攻击策略覆盖率是指在安全或鲁棒性测试中，实际执行的攻击策略数量与原先规划的攻击策略总数之间的比例。其计算公式见式（19）。

$$X_{19} = \frac{A_{19}}{B_{19}} \quad \text{.....（19）}$$

式中：

X_{19}——攻击策略覆盖率；

A_{19}——已执行攻击策略数；

B_{19}——规划策略数。

其中，已执行攻击策略数是指在测试过程中已被实施的攻击方法数量。规划策略数是指测试计划中预先列出的所有攻击策略数量。

d) 关键场景覆盖率：关键场景覆盖率用于衡量在系统测试过程中，预先规划的关键运行场景被实际测试覆盖的程度。

其计算公式见式（20）。

$$X_{20} = \frac{A_{20}}{B_{20}} \quad \text{.....（20）}$$

式中：

X_{20}——关键场景覆盖率；

A_{20}——已覆盖的关键场景数；

B_{20}——规划关键场景数。

其中，已覆盖的关键场景数是指在测试中成功执行并验证的关键场景数量。规划关键场景数是指测试计划中定义的全部关键场景数量。

8.8 可自动化性定量指标

可自动化性定量指标包括：

a) 自动化测试执行率：自动化测试执行率是指在整个测试用例集里，用例能够通过自动化方式执行的比例。计算公式见式（21）。

$$X_{21} = \frac{A_{21}}{B_{21}} \quad \text{.....（21）}$$

式中：

X_{21}——自动化测试执行率；

A_{21}——自动执行测试用例数；

B_{21}——总测试用例数。

自动执行测试用例数是指能够由测试框架、脚本、CI/CD 流程等自动运行的测试用例数量。总测试用例数是指测试计划中全部的测试用例数量。

b) 监控任务自动化覆盖率：监控任务自动化覆盖率是指系统中已实现自动化持续监控的关键指标占全部规划关键指标的比例。其计算公式见式（22）。

$$X_{22} = \frac{A_{22}}{B_{22}} \quad \text{.....（22）}$$

式中：

X_{22}——监控任务自动化覆盖率；

A_{22}——持续监控模块覆盖的关键指标数；

B_{22}——规划关键指标数。

持续监控模块覆盖的关键指标数是指已被监控模块实时采集、记录并分析的关键指标数量。规划关键指标数是指测试或运维设计阶段明确需要监控的关键指标总数。

8.9 数据可测性定量指标

数据可测性定量指标包括：

a) 标签冲突率 = 冲突标签数 / 总标签实例。

标签冲突率是指数据集中出现标签定义不一致、重复标注或语义冲突的比例。

其计算公式见式（23）

$$X_{23} = \frac{A_{23}}{B_{23}} \quad \text{.....（23）}$$

式中：

X_{23}——标签冲突率；

A_{23}——冲突标签数；

B_{23}——总标签实例。

冲突标签数是指在同一数据样本上出现标注结果不一致、或标签间存在语义冲突的标签数量。总标签实例是指数据集中所有标签标注记录的总数。

9 测试性指标指数综合评估

9.1 子指标计算方法

对于每个维度D_i下的子指标x_{ij}，计算方法如下：

对于如故障发现时间、平均定位时间、恢复时间等越小越优的指标，计算公式见式（24）。

$$n_{ij} = 1 - \frac{x_{ij} - min(x_{ij})}{max(x_{ij}) - min(x_{ij})} \quad \text{.....（24）}$$

对如覆盖率、通过率等越大越优的指标，计算公式见式（25）。

$$n_{ij} = \frac{x_{ij} - min(x_{ij})}{max(x_{ij}) - min(x_{ij})} \quad \text{.....（25）}$$

9.2 维度加权聚合

每个维度D_i的得分S_i为该维度内各指标加权求和，计算公式见式（26）。

$$S_i = \sum_{j=1}^{m_i} w_{ij} \cdot n_{ij} \quad \text{.....（26）}$$

其中w_{ij}满足下式（27）：

$$\sum_j w_{ij} = 1 \quad \text{.....（27）}$$

式中：

w_{ij}——第 i 个维度中第 j 项指标的权重，视不同智能系统由专家打分给出。

9.3 综合指标计算

整体测试性指数 TI 计算公式如下，计算公式见式（28）。

$$TI = \sum_{i=1}^{N} W_i \cdot S_i \quad \text{.....（28）}$$

W_i满足式（29）：

$$\sum_i W_i = 1 \quad \text{.....（29）}$$

式中：

W_i——第 i 个维度的权重系数，且视不同智能系统由专家打分给出。

9.4 测试性等级计算模型

根据 TI 值，将系统测试性划分为 5 个等级，具体见表 2。

表 2 测试性等级划分表格

等级	TI 范围	定性描述
A	0.90～1.00	测试性高，所有关键场景和故障类型均覆盖，具备全链路可观测、可控、可隔离能力。
B	0.80～0.90	测试性较高，绝大多数场景和故障类型均有覆盖，测试活动自动化程度高。
C	0.70～0.80	测试性达到基本要求，能满足主要测试需求，但部分维度存在不足。
D	0.60～0.70	测试性偏低，部分关键风险无法有效验证。
E	＜0.60	测试性严重不足，大量关键风险未覆盖，无法满足标准要求。

10 复杂智能系统测试性支撑技术与方法

10.1 复杂智能系统测试性支撑技术

针对不同测试内容，其对应技术指导见表 3。

表 3 测试内容对应技术指导表格

测试内容	范围定义	典型技术
功能与行为	验证系统功能、边界条件、异常处理和输入约束	功能正确性测试、边界值分析、错误处理验证、输入模式/Schema 检查
性能与效率	评估吞吐、延迟、可扩展性和资源效率	性能基准测试、压力测试、批量大小敏感性分析、资源使用监控
稳健性与鲁棒性	检查系统在噪声、缺失值和极端场景下的稳定性	随机噪声注入、轻量对抗扰动、缺失值处理测试、长尾场景模拟
公平性与偏见	检测群体间性能差异	群体分布差异、统计平等差、机会平等、TPR 差、交叉群体检测、残差分析
可解释性与透明性	验证模型可解释性及稳定性	局部解释、全局重要性、反事实、解释稳定性测试
数据质量	保证数据可用性与完整性	空值、重复、异常值检测、标签冲突、时间完整性、数据漂移
隐私	验证数据保护与隐私防护能力	成员推断测试、差分隐私预算验证、访问控制测试
对抗测试	评估系统应对恶意输入与攻击的能力	FGSM、PGD、CW、DeepFool、AutoAttack 等白盒对抗测试、NES、Boundary Attack、Transfer-based 等黑盒对抗测试、后门检测、模型抽取与反演
模糊测试	发现输入处理链路缺陷	输入变异、语义保持变异、API 序列模糊、覆盖引导模糊
差分测试	比较系统在不同实现、平台或环境下输出差异	功能差分、性能差分、跨平台一致性检测
数据扰动测试	真实退化与噪声条件下的稳健性	ImageNet-C/COCO-C 风格扰动族；加性噪声/模糊/天气合成/JPEG 压缩；亮度/对比度/伽马；旋转/缩放/平移/透视；音频加噪/变速/变调/混响；文本拼写扰动/同义改写/字符缺失；时序抖动/缺样
遮挡与模态缺失测试	局部遮挡/关键区域遮挡/子模态缺失	Cutout、Random Erasing、随机矩形/条纹遮挡；关键部位/字段掩蔽；模态 Dropout；分辨率降采样
校准与不确定性测试	概率输出与真实频率一致性	温度缩放（Temperature Scaling）、等值回归（Isotonic）、Platt Scaling、MC Dropout、Deep Ensembles、Conformal Prediction、SelectiveNet/选择性预测
事实性、归因测试	答案可溯源与避免捏造	证据检索与段落对齐（RAG）；句/段级引用比对；NLI/事实蕴含校验；声明抽取+检索+验证流水线；约束解码/引用强制
提示注入与越权测试	抵御越狱/间接注入/数据越权	红队语料与越狱模板；间接注入链构造；系统/工具提示模板审计；最小权限与 Tool allowlist；沙箱执行；canary 诱捕；输出/调用过滤器

表 3 测试内容对应技术指导表格（续）

测试内容	范围定义	典型技术
检索与排序测试	检索质量与对下游影响	稀疏检索（BM25）、稠密检索（DPR/ANCE/Contriever）、混合检索（BM25+ANN）；交叉编码器重排（MonoT5/ColBERTv2）；难例采样；查询扩展；段落/句级切分
长上下文与引用一致性测试	多文档/长序列稳健性	分块+滑窗；层级检索/摘要；位置扰动与顺序打乱；链路跳转任务；长上下文微调/RoPE 扩展；KV-Cache 压测脚本
合成代码测试	代码正确性/可运行性/安全性	自动评测 Harness（pytest/ctest）；容器化编译-运行；模糊测试（fuzzing）；静态/安全扫描（Bandit/Semgrep/CodeQL/ESLint/Flake8）；沙箱与资源隔离（nsjail/Docker）
多模态一致性与视觉幻觉	图文/视频-文本一致	CLIP/ALIP 一致性检查；GroundingDINO/GLIP 区域对齐；VQA/Referring 表达任务；检测-描述一致性校验；区域级 caption/问答
OCR/文档/表格/图表理解	视觉文档理解与抽取	OCR（Tesseract/PaddleOCR/TrOCR/Donut）；版面检测（LayoutParser/Detectron2）；表格结构化（PubTabNet/CascadeTabNet）；Chart 解析（ChartQA 管线）；PDF 解析流水线
生成模型质量与多样性测试	生成质量/覆盖/对齐	标准提示集评测（DrawBench/GenEval/COCO Captions）；种子/步数扫描；对齐评测与约束采样；蒸馏前后回归对照；双盲人评流程
视频时序与跨帧一致性测试	时间一致性与语义稳定	光流/关键点跟踪（RAFT+KP）；遮挡/运动模糊合成；时间抖动扰动；跨帧对象一致性任务；长视频分片对比
强化学习与安全约束测试	随机性/域移/约束下策略可靠性	多种子评测脚本；域随机化与 OOD 场景；安全约束环境（Safety-Gym/Minigrid-Constraints）；离线策略评估（IPS/DR）；风险评估；对抗扰动/环境攻击
记忆测试	训练数据记忆/泄露风险	会员推断攻击（Shadow Models/MIA）；canary 字符串注入；近邻搜索对照；提示泄漏探针；温度/解码策略扫描

10.1.1 功能与行为

功能与行为测试聚焦于验证系统功能实现是否符合设计规范，包含正常功能路径、边界条件处理、错误处理机制以及输入约束检查。此类测试通常涵盖输入输出的正确性、模式一致性、接口契约符合性等内容，并确保异常输入能触发合理的错误响应或保护机制。

10.1.2 性能与效率

性能与效率测试评估系统在不同工作负载下的响应速度、吞吐能力、资源利用率和可扩展性。重点包括延迟分布分析、峰值性能评估、批量大小变化对性能的影响，以及 CPU、GPU、内存和网络带宽的占用监控，以判断系统在高压条件下的稳定性与扩展能力。

10.1.3 稳健性与鲁棒性

稳健与鲁棒性测试旨在验证系统在非理想条件下的可靠性，包括在输入存在随机噪声、缺失值、极端值或罕见场景时的表现。测试需要评估输出的稳定性、结果的合理性，以及在长尾样本或罕见场景下性能保持度，防止在边缘条件下出现异常行为。

10.1.4 公平性与偏见

公平性与偏见测试关注模型在不同群体间的性能一致性，防止决策存在系统性不公。测试方法包括分析数据层的群体分布差异和代表性不足，结果指标的统计平等差、机会平等、均衡机会，以及交叉群体检测和拟合残差分析，用于识别错误率与特征或群体间的相关性。

10.1.5 可解释性与透明性

可解释性与透明测试旨在评估模型输出解释的准确性和稳定性。采用局部解释方法、全局特征重要性分析、反事实生成，并通过解释稳定性测试验证相邻输入扰动下的解释一致性，确保模型在可审计性和可理解性方面符合要求。

10.1.6 数据质量

数据质量测试涵盖数据完整性、一致性和可用性验证，包括空值检测、重复值检测、异常值分析、标签冲突检查、时间序列完整性，以及数据漂移检测。测试需建立质量基线，设置监控与报警机制。

10.1.7 隐私

隐私测试验证系统对用户数据的保护能力，包括训练数据成员推断测试、差分隐私预算验证及访问控制验证。需模拟可能的隐私攻击场景，评估防护机制的有效性，并确保隐私预算使用记录透明可审计。

10.1.8 对抗测试

对抗测试用于发现和验证系统抵御恶意输入与攻击的能力，包括 FGSM、PGD、CW、DeepFool、AutoAttack 等白盒对抗测试方法和 NES、Boundary Attack、Transfer-based 等黑盒对抗测试方法，以及后门触发器检测、模型抽取和反演攻击模拟。测试过程需分级记录攻击强度、成功率，并保留复现实验的环境配置与样本。

10.1.9 模糊测试

模糊测试旨在通过自动生成多样化的变异输入来发现系统潜在缺陷。测试范围包括文本的字符级/词级变异、图像的失真、压缩、遮挡，语义保持变异、API 序列模糊，以及针对深度模型的覆盖引导模糊等。

10.1.10 差分测试

差分测试专注于比较系统在不同实现、平台、编译器、硬件或运行环境下的表现差异，旨在发现因实现细节不同导致的功能偏差或性能异常。

10.1.11 蜕变测试

蜕变测试适用于缺乏明确“正确输出”的任务，通过定义输入的变换规则及预期的输出关系来验证系统。

10.1.12 数据扰动测试

用于评估模型在真实退化条件下的稳定性；需包括噪声、模糊、天气、压缩、亮度、几何变换、音频噪声/变速、文本拼写扰动与同义改写、时序缺样与抖动等扰动；测试过程需按严重度分级汇报性能降幅，并输出鲁棒曲线与最差场景表现。

10.1.13 遮挡与模态缺失测试

用于检验输入部分缺失时的稳健性；需包括 Cutout/Random Erasing、随机遮挡、关键信息掩蔽、多模态缺失与分辨率退化；测试过程需统计关键目标/字段被遮挡时的性能下界与故障模式。

10.1.14 校准与不确定性测试

用于衡量概率输出与真实频率的一致性并暴露过度自信；测试过程需产出 ECE/NLL/Brier 与分段校准误差，并检查选择性预测。

10.1.15 事实性、归因测试

用于验证智能系统回答是否有据可依、是否捏造；测试过程需逐条核对引用是否覆盖关键信息，统计幻觉率、引用覆盖率与不一致样例。

10.1.16 提示注入与越权测试

用于检验模型抵御 Prompt Injection、间接注入与数据越权的能力；测试过程需记录越权成功率、敏感信息泄露率、工具调用误触发率，并保留系统/工具提示模板与沙箱日志。

10.1.17 检索与排序测试

用于衡量检索子系统质量及其对下游答案的影响；测试过程需对召回-准确的权衡做灵敏度分析，并给出端到端答案质量与引用一致性的关联。

10.1.18 长上下文与引用一致性测试

用于评估长序列/多文档场景下的稳健性；测试过程需记录命中率、位置偏差容忍度、引用跳转正确率与超长上下文下的延迟/内存。

10.1.19 合成代码测试

用于验证代码生成的正确性、可运行性与安全性；测试过程需全自动“编译-运行-断言”，输出通过率、运行时错误率与安全缺陷数。

10.1.20 多模态一致性与视觉幻觉

用于检测图文/视频-文本不一致与编造细节；测试过程需分别统计跨模态一致性得分、幻觉率与定位 IoU/命中率等。

10.1.21 OCR/文档/表格/图表理解测试

用于验证文档视觉理解与文本抽取质量；需包括 TextVQA/DocVQA/ChartQA、编辑距离、字段级精确-召回、版面理解；测试过程需保存 PDF/图像与解析结果对齐标注，输出字段级 F1 与端到端 EM。

10.1.22 生成模型质量与多样性测试

用于衡量生成质量、覆盖与对齐；测试过程需分开报告“质量-多样性”与“文本-图像对齐”，并做种子/步数-质量曲线与蒸馏前后回归。

10.1.23 视频时序与跨帧一致性测试

用于评估视频生成/理解的时间一致性与语义稳定；需包括 FVD、tLPIPS、跨帧关键点漂移、遮挡/运动模糊耐受等；测试过程需统计片段/长视频两种粒度指标与时间一致性错误类型。

10.1.24 强化学习与安全约束测试

用于评估策略在随机性、域移与约束下的智能系统可靠性与安全性；测试过程需多种子重复，报告 IQM/Bootstrap CI、样本效率、Regret、成功率与约束违规率等。

10.1.25 记忆测试

用于检测训练数据被记忆/泄露的风险；测试过程需在不同温度/采样下测暴露率，给出阈值-ROC 曲线与防护前后对比。

10.1.26 数据治理与污染检查测试

用于确保评测数据独立、标签可靠与许可证合规；测试过程需产出去重率、污染样例清单与修订后的对比结果。

10.1.27 压缩、量化和蒸馏回归测试

用于验证模型压缩后的功能不回退；测试过程需设定关键指标跌幅阈值，输出 Before/After 对照与鲁棒性差异。

10.1.28 跨域、跨设备和多语泛化测试

用于衡量在新域/新设备/新语言下的迁移能力；测试过程需报告平均与最差域表现、相对降幅与微调样本/时长等适配成本指标。

10.2 复杂智能系统测试性执行流程

智能系统测试性执行流程旨在确保在系统全生命周期内，从需求定义到运行退役，各阶段均具备可验证、可度量、可复现的测试能力。流程包括以下主要步骤：

a) 测试分析：明确业务目标与系统功能范围，识别关键风险源。制定包含测试范围、优先级、测试性目标等内容的初步测试策略；

b) 测试设计与环境准备：确定测试类型、对应测试指标以及技术手段。准备测试数据集、模拟器/仿真器、硬件配置及监控工具，构建可控的测试环境。此外，需确保环境具备可重复性与可隔离性，以便在不同时间、不同团队间一致复现结果；

c) 执行与监控：按测试计划实施测试用例，实时采集运行日志、性能指标、测试覆盖率数据。记录所有输入、输出、配置和版本信息，为可追溯性提供依据。

d) 分析与反馈：对测试结果进行统计分析，发现测试性不足的环节。形成缺陷报告与改进建议，闭环提升系统测试性。

e) 持续评估与维护：在系统更新、模型再训练或环境变化后，重复测试性验证，确保改动未降低系统的测试性水平。同时建立长期的测试性监控机制，与持续集成/持续部署流程结合，实现自动化测试性检测。

11 智能系统生命周期阶段与测试性活动

对于智能系统测试性活动，分别从需求与风险分析、数据获取与准备、模型开发与训练、集成与系统测试、部署与验证、运行与持续评估、退役与存档七个生命周期阶段对智能系统进行分析。其生命周期阶段与测试性活动映射表格见表 4。

表 4 智能系统生命周期阶段与测试性活动映射表格

阶段	测试目标	主要活动	产出物
需求与风险分析阶段	明确业务目标与 AI 系统预期功能范围；建立风险识别与分级体系	a) 业务与功能需求确认 b) AI用例分类 c) 风险识别与分类 d) 初步测试策略制定 e) 偏见敏感属性清单建立	风险登记册、测试策略草案、偏见敏感属性清单
数据获取与设计阶段	确保数据合法、合规、可用；建立质量与偏见基线	a) 数据审计与质量测试 b) 场景与分布标注 d) 偏见基线分析	数据审计报告、数据质量评估报告、 数据分布与覆盖率基线、数据偏见基线报告
模型开发与训练阶段	验证模型结构合理性；确保核心模块功能正确性与鲁棒性；建立训练可追溯性	a) 白盒结构分析 b) 单元级测试 c) 对抗鲁棒性评估 d) 神经元/通路覆盖探索	模型结构分析报告、 单元测试报告、训练过程日志与分析结果、初步鲁棒性评估报告
集成与系统测试阶段	验证系统接口正确性；评估性能与安全性	a) API 测试 b) 性能测试 c) 模糊测试 d) 对抗测试 e) 公平性回归测试 f) 安全/隐私测试	系统接口验证报告、性能测试报告、对抗测试与公平性回归报告、安全与隐私测试报告
部署与运行验证阶段	在生产或准生产环境中验证性能与合规性；建立运行基线	a) 流量回放测试 b) 建立可信性基线 c) 合规核对	验收测试报告、性能与可信性基线文档、合规核查记录
维护与持续评估更新阶段	实时监控性能、漂移与公平性；确保及时触发更新与再验证	a) 在线监测 b) 模型健康评分 c) 再训练触发策略 d) 漂移后再验证与差分测试	运行监控报告、模型健康评分记录、再训练触发与验证报告
退役与存档阶段	确保退役过程可追溯、合规；形成经验闭环	a) 模型与数据封存 b) 审计日志留存 c) 偏见与性能改进记录闭环	退役与存档记录、审计日志、改进闭环报告

参 考 文 献

[1] GB/T 25000 系统与软件工程 系统与软件质量要求和评价
[2] GB/T 35273 信息技术安全 个人信息安全规范
[3] GB/T 36344-2018 信息技术 数据质量评价指标
[4] GB/T 42018 信息技术 人工智能 平台计算资源规范
[5] GB/T 42018-2022 智能制造 远程运维系统通用要求
[6] GB/T 44662-2024 健康管理 终端设备数据采集与传输协议
[7] GB/T 45087-2024 人工智能 服务器系统性能测试方法
[8] GB/T 45225-2025 人工智能 深度学习算法评估
[9] GB/T 45288.2-2025 人工智能 大模型 第 1 部分：通用要求
[10] GB/T 45288.2-2025 人工智能 大模型 第 2 部分：评测指标与方法
[11] ISO/IEC 5259 信息技术 人工智能 数据分析与机器学习
[12] ISO/IEC 5338 信息技术 人工智能 人工系统生命周期过程
[13] ISO/IEC 23894 信息技术 人工智能 风险管理指南
[14] ISO/IEC TR 24028 信息技术 人工智能 可信性概述
[15] ISO/IEC 25010 系统与软件工程 系统与软件质量要求和评价
[16] ISO/IEC TS 25058 系统和软件工程 人工智能系统质量评估指南
[17] ISO 26262 道路车辆功能安全
[18] ISO/IEC/IEEE 29119 系统与软件工程 软件测试
[19] ISO/IEC 42001 信息技术 人工智能 管理体系
[20] NIST AI RMF 1.0 AI 风险管理框架
[21] NIST SP 800-53 Rev5 信息系统和组织的安全和隐私控制
[22] NIST SP 800-218 人工智能和两用基础模型的安全软件开发实践
[23] NIST AI 100 人工智能风险管理框架
[24] EN 50129 轨道交通功能安全
[25] EU AI Act 欧盟人工智能法案
[26] IEEE 7001 自治系统透明度
[27] IEEE 7003 算法偏差考量
[28] IEEE 1232 人工智能在测试环境中的交换与服务连接标准
[29] IEEE3110 深度学习框架中算法接口应用程序接口的技术要求