

ICS 35.240.01

CCS L70

SZSD

数字山东工程标准

SZSD13 0002—2024

城市智能中枢 算法协同接入要求

Urban intelligent center requirements of algorithm collaborative access

2024-04-15 发布

2024-06-01 实施

临沂市大数据局

发布

目 次

前 言 0

1 范围 1

2 规范性引用文件 1

3 术语和定义 1

4 缩略语 1

5 总体架构 2

6 算法管理 3

 6.1 算法展示与查询 3

 6.2 算法封装 4

 6.3 算法发布 6

 6.4 外部算法注册/注销 6

 6.5 算法导入 7

7 算法库 7

 7.1 算法类别 7

 7.2 算法存储 8

 7.3 算法元数据 8

 7.4 算法生命周期 9

8 算法服务 9

 8.1 算法调度 9

 8.2 算法配置 9

9 配置管理 10

 9.1 权限管理 10

 9.2 算法统计分析 10

 9.3 算法审核 10

10 算法开放接口 10

 10.1 算法管理接口 10

 10.2 算法服务接口 11

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。

本文件由临沂市大数据局提出并归口。

本文件起草单位：临沂市大数据局、山东浪潮新基建科技有限公司、临沂市大数据中心。

本文件主要起草人：熊云峰、英杰、庄绪路、解桂林、李龙、周波、郝建和、冯兵池、吕厚雷、孙凤巧。

城市智能中枢 算法协同接入要求

1 范围

本文件确立了临沂市多种类型算法统一管理、统一服务和配置的算法协同接入要求。
本文件适用于临沂市多种智能算法协同应用的规划、设计、建设与运行维护。

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。

DB37/T 4613.1-2023 智慧城市 基础设施 第1部分：城市智能中枢总体要求

3 术语和定义

下列术语和定义适用于本文件。

3.1

算法协同接入 `algorithm collaboration access`

将多种来源的算法按照统一格式进行封装、发布、注册、导入、管理与配置等，向用户提供一致的算法服务方式。

3.2

算法包 `algorithm package`

将算法代码及相关依赖项、资源配置、帮助说明等附属信息，按照统一格式封装为可复用、可部署的一种文件形式。

3.3

城市智能中枢 `urban intelligent hub`

运用云计算、大数据、人工智能等技术构建的为城市运行提供科学决策和指挥调度能力的平台型人工智能中枢，是数字政府、数字社会和数字经济的载体，汇集政府、企业和社会等数据，进行融合计算，感知城市生命特征，城市的全域精准分析、整体研判、协同指挥和科学治理提供支撑。

[来源：DB37/T 4613.1-2013, 3.1]

4 缩略语

下列缩略语适用于本文件。

AI：人工智能（Artificial Intelligence）

JSON：JavaScript 对象标记语言（JavaScript Object Notation）

URI: 统一资源标识符(universal resource identifier)

5 总体架构

算法协同接入总体架构如图1:

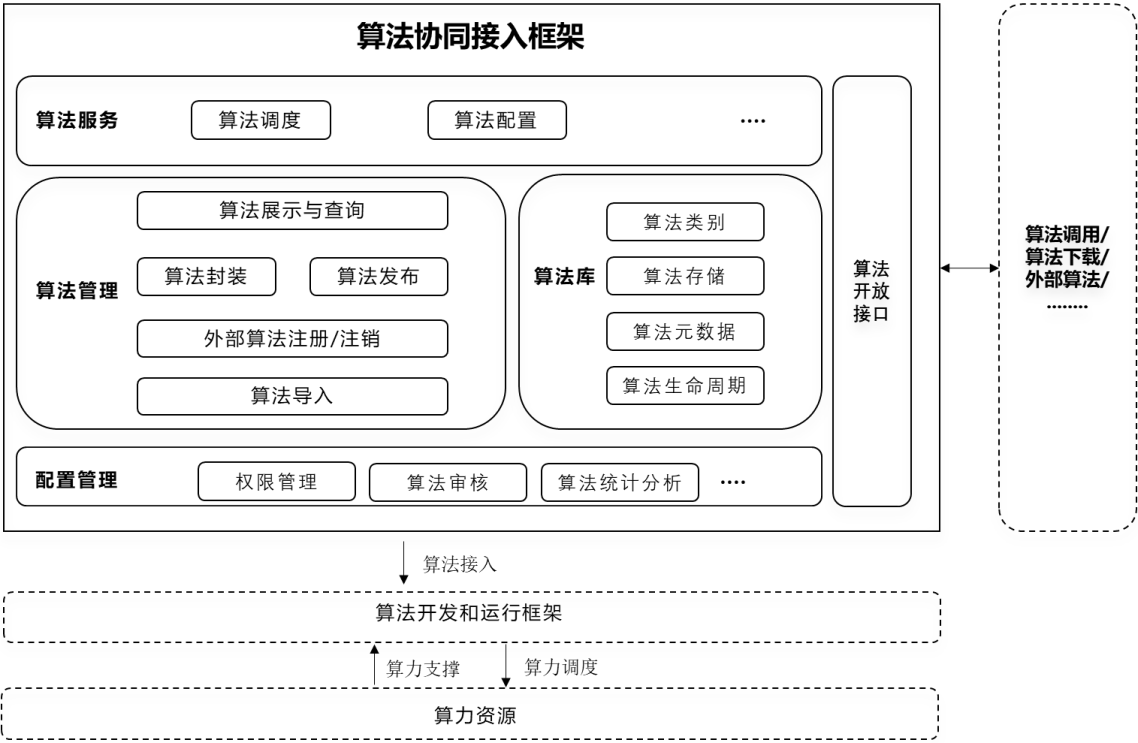


图1 算法协同接入框架

算法协同接入框架包括算法管理、算法库、算法服务、配置管理及算法开放接口组成，具体描述如下：

- a) 算法管理：算法协同接入的主要功能组件，包括算法展示与查询、算法封装、算法发布、外部算法注册/注销、算法导入等；
 - 1) 算法展示与查询：展示所管理各类算法及算法相关属性信息，提供多种查询算法的方式；
 - 2) 算法封装：将不同类型算法按照统一的格式或方式封装为标准算法包，算法协同接入框架可以识别并安装（加载）算法包，算法封装简化算法的使用和管理，并提供一致的接口和功能；
 - 3) 算法发布：将算法按照可公开访问的方式发布到算法协同接入框架，便于算法使用方查询、下载、安装配置；
 - 4) 外部算法注册/注销：外部算法可以注册（加载）到算法协同接入框架，算法使用方可以在框架下管理和运行调试外部算法。对不再使用的外部算法框架可以注销（删除/下架）。
 - 5) 算法导入：导入算法代码版本，并按算法版本管理；
- b) 算法库：描述算法库分类管理算法的要求，以及算法相关存储、配置等方面的要求，包括算法类别、算法存储、算法元数据、算法生命周期；
- c) 算法服务：为算法使用发提供相关算法服务，包括但不限于算法调度、算法配置等；
 - 1) 算法调度：对外部提供可视化的任务编排和调度能力，实现不同算法协同工作；
 - 2) 算法配置：为保障算法正常工作，对算法场景、算法使用条件、规则等方面的配置要求；

- d) 配置管理：对各算法进行统一配置管理，包括权限管理、算法审核、算法统计分析：
 - 1) 权限管理：是对管理算法协同接入的人员权限进行配置和管理；
 - 2) 算法审核：针对人工智能算法涉及个人隐私，安全风险、知识产权风险等，提出算法审核要求；
 - 3) 算法统计分析：描述算法使用和运行等方面的统计和分析；
 - e) 算法开放接口：描述算法协同接入对外接口功能，包括算法管理接口和算服务接口等；
 - f) 算法开发和运行框架：支撑算法开发和实际运行算法的应用框架，算法协同接入框架为算法开发和运行框架提供多类型、多算法接入，算法开发和运行框架对算力资源进行算力调度，算力资源为算法开发和运行框架提供算力支撑。
- 注：本文件描述了总体架构中实线部分的技术要求，不涉及虚线部分的技术要求。

6 算法管理

6.1 算法展示与查询

6.1.1 基本要求

算法展示与查询展示各类算法相关属性信息，包括但不限于以下基本要求：

- a) 应支持展示和查看算法的算法名称、版本号信息、算法类别、场景类型、激活状态、使用说明等信息，具体见表1；
- b) 应支持通过算法列表、算法缩略图等多种方式查询算法具体信息及运行详情；
- c) 应支持展示算法模型的描述内容、语言结构、实例文档等信息；
- d) 应支持按照算法类别、场景类型、算法类型、关键字、算法分析所需的数据源类型、算法分析输出目标类型等一种或多种条件进行筛选或组合查询，并支持记录导出。

表 1 算法基本属性信息

属性内容	描述
算法名称	展示算法名称，如打架斗殴行为发现、违规停车发现等。
版本信息	每个算法对应的版本号信息。
算法类别	展示算法归属分类，如行为分析类算法、统计类算法等。
场景类型	算法场景用于管理算法具体应用，一个算法可有多种算法场景标签。
激活状态	展示算法激活状态信息，如已注册激活、未激活、已过期等状态信息。
使用说明	展示算法简介信息、详细功能说明等。

6.1.2 算法展示

算法展示对所管理的各类算法及其属性信息，能以清晰和直观的方式进行展示，便于用户快速浏览和了解可用的算法，从而做出合适的选择，应满足以下要求：

- a) 算法分类展示：算法按照不同的分类标准进行分组展示，如类型（分类、回归、聚类等）、领域（自然语言处理、计算机视觉等）或其他自定义的分类方式；
- b) 算法列表展示：在每个算法分类下，以列表的形式展示所管理的算法，列表中包含算法的基本信息，如名称、作者、版本等；

- c) 算法详情展示：在算法详情页中展示更全面的算法信息，包括算法描述、适用场景、主要功能、优势和限制等；
- d) 算法属性信息：展示算法的属性信息，如算法参数、输入输出格式、模型大小、训练数据要求等；
- e) 示例和演示：为算法提供示例和演示，以帮助用户更好地理解算法的工作原理和应用场景。示例可以是简单的代码片段或者实际的数据案例，演示可以是算法在真实环境中的运行效果展示；
- f) 算法标签和关键词：为算法添加标签和关键词，以使用户通过标签或关键词搜索特定类型或特定功能的算法；
- g) 算法评价和评分：允许用户对每个算法进行评价和打分，以反映其使用体验和性能。这些评价和评分可以帮助其他用户更准确地选择合适的算法，并提供参考和决策依据。

6.1.3 算法查询

算法查询提供多种查询算法的方式，使用户能够根据特定需求和条件快速找到符合要求的算法，应满足以下要求：

- a) 关键字搜索：允许用户通过输入关键字进行算法搜索。用户可以输入算法名称、关键词、领域或其他相关描述，系统将根据输入的关键字匹配算法名称、描述、标签等相关信息，并返回与关键字匹配的算法列表；
- b) 高级搜索：提供高级搜索选项，允许用户按照更详细的条件进行算法查询。用户可以选择算法类型（分类、回归、聚类等）、领域（自然语言处理、计算机视觉等）、开发者、版本号或其他自定义的筛选条件来查询；
- c) 筛选和排序：在查询结果页面，用户可以使用筛选和排序功能进一步细化结果。用户可以根据算法属性（如模型大小、训练数据要求）或用户评分等条件进行筛选，并根据名称、作者、更新时间等属性进行排序；
- d) 相似算法推荐：基于用户的查询历史和算法评价数据，系统可以提供相似算法推荐功能。根据用户查询的算法或相关条件，系统会推荐其他相似类型或功能的算法，帮助用户发现可能符合需求的其他算法选择；
- e) 智能建议和补全：在用户输入查询关键字时，系统可以提供智能建议和补全功能。根据已有的算法库和用户的输入，系统会自动提示相关的算法名称、关键词或领域，帮助用户更快速地完成查询；
- f) 用户评价参考：为每个查询结果展示用户评价和评分，以提供参考和决策依据。用户可以查看其他用户对该算法的评价和使用体验，从而更好地了解算法的优势和限制。

6.2 算法封装

算法封装应满足以下要求：

- a) 应支持不同类型算法遵循统一格式封装为标准算法包，算法包格式可以是一种约定的目录结构或打包格式，以确保算法包的一致性和可用性；
- b) 算法协同接入框架可以识别并安装（加载）算法包，算法包应支持统一命名规范、目录结构规范、配置规范，包含算法代码、依赖库等基本部分；
- c) 应支持将算法工程（包括算法的代码和相关资源）通过工具或脚本构建为算法包，通过编译、依赖项管理、资源复制等过程生成算法包文件，算法工程可支持Node.js、Python、C/C++、Java等开发语言；
- d) 应支持将算法工程打成压缩文件包（比如zip文件格式），上传到算法协同接入平台；

- e) 应支持算法包依赖管理，如果算法工程引入第三方依赖，可根据开发语言类型，将第三方依赖和算法工程文件封包成压缩文件包（比如zip文件格式），或将第三方依赖和算法工程文件一起封包并编译成可执行文件（比如jar文件格式）；
- f) 应支持对一个或多个原子算法进行封装，可包括但不限于算法模型、算法形式化描述文件、算法程序等，一个算法包应只包含一种类型的计算设备的算法模型和算法程序；
- g) 算法包应包含说明文档和示例，包括算法的介绍、参数说明、输入输出格式、使用示例等；
- h) 算法包应支持版本管理，每个算法包应有唯一的版本号，支持管理和跟踪不同版本的算法包，可选择和切换不同版本的算法；
- i) 应支持算法包命名要求和目录结构要求，算法包名称包含算法标识、厂商代号和版本号等信息，目录结构要求见表2，算法属性信息要求见表3；
- j) 宜支持在云原生环境进行算法封包，将算法工程（包含依赖项）及其运行环境打包为便携式容器镜像，提供运行时环境,可在支持容器的平台上部署和运行。

表 2 算法包目录结构要求

目录名称	选择状态	文件/子目录说明		内容说明
bin	必选	算法程序文件		存放算法程序、算法模型等文件
script	必选	算法管理		算法程序自动、停止和状态查询脚本
data	可选	以算法标识为目录		存放用于演示算法功能效果的样本数据
META-INF	必选	以算法标识为目录	cover	封面附件资料目录，存放展示算法能力封面的视图文件
			condtion	应用场景附件资料目录，存用展示算法能力应用场景的说明文件
			managementinfo.json	算法属性信息描述文件
			echnicalinfo.json	算法技术信息描述文件

表 3 算法属性信息要求

属性内容	描述
算法标识	系统自动生成
算法应用版本	算法版本号信息，遵循语义版本管理编码规则
算法应用展示名称	中文，不超过 128B
算法场景类型标签	需填标签标识（id）
算法应用类型	视频流解析、图片流解析等
算法应用描述信息	算法功能简介，不超过 2KB
算法应用图标	PNG 格式图片，64 KB 以内
算法作者	算法作者信息或所有者信息
算法应用授权	算法授权项信息
算法授权状态	系统返回
资源配置参数	指定算法运行时所需要的硬件资源，包括 CPU 核数/内存/GPU 卡类型及数量等
性能配置参数	指定算法运行时能达到的计算能力，如视频以路数为单位（区分不同分辨率）
算法许可证	算法所使用的许可证方式及许可证文件
输入说明	下发任务时用户输入配置参数的模式描述
输出说明	用户自定义输出部分的 JSON 模式描述

6.3 算法发布

算法发布应满足以下要求：

- a) 应支持在算法发布前进行验证和审核；
- b) 验证过程可包括但不限于对算法代码进行静态分析、检查算法的合规性和安全性，以确保算法的质量和可用性；
- c) 审核过程应由相关管理人员或专业团队进行，确保算法符合规定的发布标准和要求；
- d) 应支持将算法按照6.2算法封装方式发布到算法协同接入框架，供算法使用方查询、下载、安装配置；
- e) 应支持将算法版本发布，可支持基于版本的更新；
- f) 应支持算法按照统一流程规范进行发布，提供对应发布说明和测试说明；
- g) 发布说明应包含算法名称、版本号、作者、发布时间、功能点描述、场景边界、支持硬件等信息；
- h) 应支持发布算法的权限管理，可以设定公开访问的算法，供所有用户使用，或者设定私有算法，只允许特定用户或组织访问和使用；
- i) 测试说明应包含算法性能测试、处理性能测试、精度等级等信息；
- j) 应支持算法版本的查询、转让、下载、注销（下架）等功能；
- k) 应支持算法使用方在算法发布信息基础上进行算法可用性管理、安装配置等；
- l) 应支持对已发布的算法进行状态监控，包括算法的运行状态、调用次数、性能指标等。

6.4 外部算法注册/注销

6.4.1 外部算法注册

外部算法注册应满足以下要求：

- a) 应支持外部算法主动注册到统一调度平台，注册过程应有平台分配的密钥，密钥可作为准入、加密签名使用；
- b) 在注册过程中应提供外部算法的相关信息，包括但不限于算法名称、路径、依赖项、输入输出格式等；
- c) 应支持心跳，主动上报外部算法状态；
- d) 应支持外部算法对自身资源使用情况进行上报，用于支持多个外部算法服务节点间的资源调度；
- e) 应支持外部算法服务上报算法结果、算法运行状况、算法异常现象等；
- f) 外部算法通过调用接口可以上传算法应用包及授权相关信息完成算法加载；
- g) 算法协同接入框架应支持提供调试环境，进行外部算法应用包管理、算法任务管理等相关功能的调试验证；
- h) 外部算法注册和注销时应进行认证，认证方式应支持数字认证方式；

6.4.2 外部算法注销

外部算法注销是将不再使用的外部算法从算法平台中注销或删除，应满足以下要求：

- a) 应支持对外部算法服务进行注销（下架），外部算法提供方应支持功能调试，下架后的外部算法应不在框架内展示；
- b) 对于已启动运行的算法，应先停止运行后再进行下架操作；

- c) 应支持从算法平台中移除外部算法的相关信息和配置，包括但不限于算法的注册信息、路径、依赖项等；
- d) 在注销过程中应支持由用户选择是否删除与该外部算法相关的数据和文件；
- e) 如果其他算法或应用程序依赖于该外部算法，在注销过程中应支持提供警告或提示信息，确保注销操作不会影响到其他功能的正常运行，支持用户根据需要决定是否解决依赖项问题或调整相关配置；
- f) 外部算法注销后，应支持执行相应的清理操作，包括但不限于释放占用的资源、删除相关的文件或数据等，确保注销的外部算法不再占用资源。

6.5 算法导入

算法导入应满足以下要求：

- a) 应支持算法通过导入的方式导入到服务器端；
- b) 导入成功后应支持任务配置下发；
- c) 应支持导入已注册算法在线更新版本；
- d) 应支持按算法版本或分支管理，或覆盖已有算法版本或分支；
- e) 应支持算法版本之间的切换，包括同一算法多次更新升级版本的管理、多版本共存管理等；
- f) 应支持算法分析服务机群感知算法版本的变化，并更新本地算法；
- g) 宜支持在线编辑代码、文件引入、上传压缩包等多种代码导入方式，支持的语言可包括Node.js、Python、C/C++、Java等；
- h) 应支持对导入的已注册算法版本进行推理服务资源、动态调度算法服务部署等进行统一管理。
- i) 根据用户的需求和算法的更新情况选择导入方式；
- j) 应支持用户通过算法的唯一标识、名称、路径等信息进行识别和选择，在管理界面中查看已注册的算法，并选择要导入的算法；
- k) 导入算法后应支持验证和审核验证算法的合法性，并根据需要更新算法的相关信息和配置；
- l) 应支持按算法分支管理，若选择按照新的算法分支导入算法，将创建新的算法分支，并将导入的算法作为该分支的一部分，方便用户在不同分支上进行算法的开发、测试和调试；
- m) 应支持覆盖已有算法，若选择覆盖已有的算法版本或分支，将更新现有算法的相关信息和配置，并用导入的算法替换原有算法，确保算法的更新得到及时应用；
- n) 应支持对导入算法的启用和禁用、参数配置、调试选项设置等，可通过统一管理界面，对外部算法进行管理与配置；
- o) 宜支持将更新后的算法逐步引入并发布到生产环境，以降低潜在的风险和影响，根据灰度发布计划，逐步扩大更新算法的发布范围和受影响的用户，保证稳定性的同时，逐步替换旧算法。

7 算法库

7.1 算法类别

7.1.1 存储管理

存储管理应满足以下要求：

- a) 应支持按不同类型的对象设置不同的存储策略，允许存储空间进行管理；
- b) 应支持设置存储对接、加密方式；

- c) 应支持对存储情况进行监控，包括读写性能、存储空间等，发生异常时进行告警；
- d) 应支持对存储空间、读写性能等进行统计。

7.1.2 任务管理

任务管理应满足以下要求：

- a) 应支持为不同应用场景匹配不同算法任务，以应用场景形式分类管理算法；
- b) 应支持接受算法协同框架下发的任务指令，包括但不限于任务启动、停止、更新、查询、删除、配置指令等；
- c) 应支持为上层应用提供算法任务管理服务，可以手动创建算法任务，也支持通过上层应用调用自动创建任务。

7.2 算法存储

每个算法源代码文件应具有资源路径标识，资源路径由统一资源标识符URI标识。

7.3 算法元数据

7.3.1 算法分类

算法分类应满足以下要求：

- a) 算法分类应保持稳定，对于不稳定的分类，应以标签形式体现；
- b) 分类的结构为：

标识号 id

名称 name

父分类标识号 parent_id

7.3.2 算法标签

算法标签满足以下要求：

- a) 算法标签包括预定义的内置标签、自定义标签等；
- b) 标签的结构为：

标识号 id

名称 name

分类 category

- c) 内置标签可包括以下结构：

编程语言和版本，示例： <language_version> <language>

框架名称和版本，示例： <framework_version> <framework>

适用的行业，示例： <industry_id> <industry>

7.3.3 算法基本信息

算法基本信息结构包括但不限于：名称、版本、参数、存储位置、依赖、描述等。

7.3.4 指标信息

指标信息结构包括但不限于：存续时长、下载总次数、下载总人数、最近一周下载次数、最近一月下载次数、最多下载月份。

7.3.5 评估矩阵

应选取可灵活订制的指标组成的矩阵，设定不同指标的权重，根据计算公式，算出综合分数，综合反映该算法的状况。

7.4 算法生命周期

算法从使用到归档的生命周期内应遵循规范流程，应满足以下要求：

- a) 应按照算法存在区域划分为在线区、近线区、离线区，近线区和离线区在于归档要求；
- b) 算法不同区域间移动应有严格的权限管理，记录操作日志；
- c) 使用期间应提供服务，归档期间仅提供查询服务；
- d) 在线区算法没有被使用满足一定时间阈值时，应停止服务，并将算法代码和元数据一起转移至近线区；
- e) 算法在近线区存在达到一定时间阈值时，应将算法代码和元数据一起转移至离线区。

8 算法服务

8.1 算法调度

算法调度应满足以下要求：

- a) 应支持按任务情况进行算法调度，将相关算法在统一资源池内进行调度；
- b) 基于资源动态调度机制，应支持按优先级调度分配，优先调度优先级高的任务；
- c) 应根据上层业务的实际需求，动态、按需加载算法模型；
- d) 应支持根据任务计划或指令进行多种算法的调度；
- e) 应支持按照算法库规定的接口标准，通过界面交互方式实现可视化的任务编排、多算法编排，灵活按需调度、按需分配资源池中的算力资源；
- f) 应支持同类算法模型的资源复用，合理利用计算资源；
- g) 应支持加载、释放、调度、运行、监控相应的算法，满足算法任务、算力资源的动态分配需求；
- h) 应支持将多种算法统一、按需分配计算资源，提升资源的整体利用率；
- i) 应支持对算力资源使用情况、运行状态进行监控。

8.2 算法配置

算法配置应满足以下要求：

- a) 应支持算法分类，根据应用场景选择算法库中对应的算法种类，包括第三方算法；
- b) 应支持算法包信息的基本操作，包括但不限于增加、删除、修改、查询等；
- c) 应支持配置规则类型，选择可下发任务的具体算法仓库事件；
- d) 应支持配置任务ID；
- e) 应支持配置算法参数，下发任务指定的算法参数，根据算法配置格式配置算法参数；
- f) 应支持配置算法包的初始化信息，包括算法包路径、处理结果回调函数、初始化参数、运行环境信息等；

- g) 应支持配置任务优先级,创建任务时标定的任务优先级,任务调度时会根据任务优先级优先调度;
- h) 对需要配置触发事件的算法,应支持配置触发时间,配置触发事件的时间及持续发生时间;
- i) 对需要配置触发事件的算法,应支持配置算法触发事件的间隔时间。

9 配置管理

9.1 权限管理

权限管理应满足以下要求:

- a) 权限管理应满足DB37/T 4613.1-2023中6.6的要求;
- b) 应支持多种访问控制方式,包括但不限于管理用户使用的登录端IP地址、管理用户多点登陆等;
- c) 应支持权限控制的基本操作,包括但不限于添加、回收、查看、修改、删除等权限;
- d) 应支持用户组织结构管理及权限控制。

9.2 算法统计分析

算法统计分析应满足以下要求:

- a) 应支持监测和统计分析所有分析节点的算力使用情况、当前任务数、运行算法类型和详情数据等;
- b) 应支持监测和统计分析算法事件情况,包括但不限于事件数量、事件分布等;
- c) 应支持多种数据展现形式,包括但不限于饼图、折线图、柱状图等。

9.3 算法审核

算法审核应满足以下要求:

- a) 应支持人工审核算法运行结果,用于算法训练、调优;
- b) 应支持以算法为维度的审核开关,按需截流参与审核;
- c) 应支持自定义审核结果,包括但不限于正检、误检,还能包括重复正检、重复误检等自定义标签;
- d) 应支持审核结果的分类统计。

10 算法开放接口

10.1 算法管理接口

算法管理接口应满足以下功能要求:

- a) 算法封装管理接口应支持调用算法封装相关接口进行管理,包括但不限于算法包封装格式配置、算法包上传、算法包删除、算法包查询等功能;
- b) 算法应用实例管理接口应支持调用算法应用实例管理接口进行管理,包括但不限于算法应用实例创建、删除指定算法应用实例、获取指定算法应用实例信息详情、更新指定算法应用实例等功能;
- c) 算法任务接口应支持调用算法任务相关接口实现任务管理,包括但不限于算法任务创建、算法任务删除、任务对象配置、查询任务列表、查询任务详情、任务状态回调等功能;
- d) 算法应用信息接口应支持调用算法应用信息相关接口进行管理,包括但不限于列举展示算法应用信息、删除指定算法应用、获取算法应用详细信息、算法应用信息更新等功能;

- e) 算法应用授权接口应支持调用算法应用授权接口进行授权信息管理，包括但不限于获取算法授权信息、算法授权状态等功能。

10.2 算法服务接口

算法服务接口应满足以下功能要求：

- a) 算法调度任务接口应支持调用调度相关接口实现算法调度服务，包括但不限于算法调度任务相关配置、任务编排、获取算法解析资源、资源配置修改等。
 - b) 算法开发接口应支持调用算法开发相关接口支撑开发工作，支持包括但不限于算法输入输出格式、算法运行实例、算法应用包调试等。
 - c) 算法配置接口应支持调用算法配置相关接口进行算法配置，支持包括但不限于算法配置格式查看、修改算法配置参数、删除指定算法配置信息等。
-