YOUR **COMPANY NAME**

**2024年全国大学生信息安全竞赛**

**概要及详细设计说明书**

**作品名称：**  Fed Privacy Shield

**电子邮箱：** 2948953101@qq.com

**提交日期：** 2024年6月5号

**目录**

**[1.1总体设计](#_Toc14410)** [1](#_Toc14410)

[1.1.1架构设计 1](#_Toc11094)

[1.1.2 总体流程设计 4](#_Toc19306)

[1.1.3 LOGO设计 5](#_Toc15793)

**[1.2 功能设计](#_Toc29894)** [6](#_Toc29894)

[1.2.1 用户端功能设计 6](#_Toc21885)

[1.2.2 管理端功能设计 18](#_Toc21091)

**[1.3 关键算法设计](#_Toc11202)** [29](#_Toc11202)

[1.3.1 聚合方案 29](#_Toc26588)

[1.3.2 通信传输 45](#_Toc29086)

[1.3.3 安全加密 48](#_Toc5660)

**[1.4数据库设计](#_Toc12161)** [58](#_Toc12161)

**[1.5接口设计](#_Toc10788)** [65](#_Toc10788)

**[1.6本章小结](#_Toc4988)**[70](#_Toc4988)

**概要及详细说明书**

本章主要介绍Fed Privacy Shield平台的总体设计、功能设计、核心技术设计、数据库设计以及接口设计。

1.1总体设计

本节概述Fed Privacy Shield平台的总体设计，从平台架构设计、平台功能模块设计、关键算法设计、系统数据库设计及接口设计对系统结构进行介绍。

1.1.1架构设计

1.1.1.1软硬件架构

Fed Privacy Shield平台软件架构设计分为五个层次，分别为用户层、展现层、云数据中心层、边缘服务器层和训练设备层。软件硬件架构图如图1-1所示：

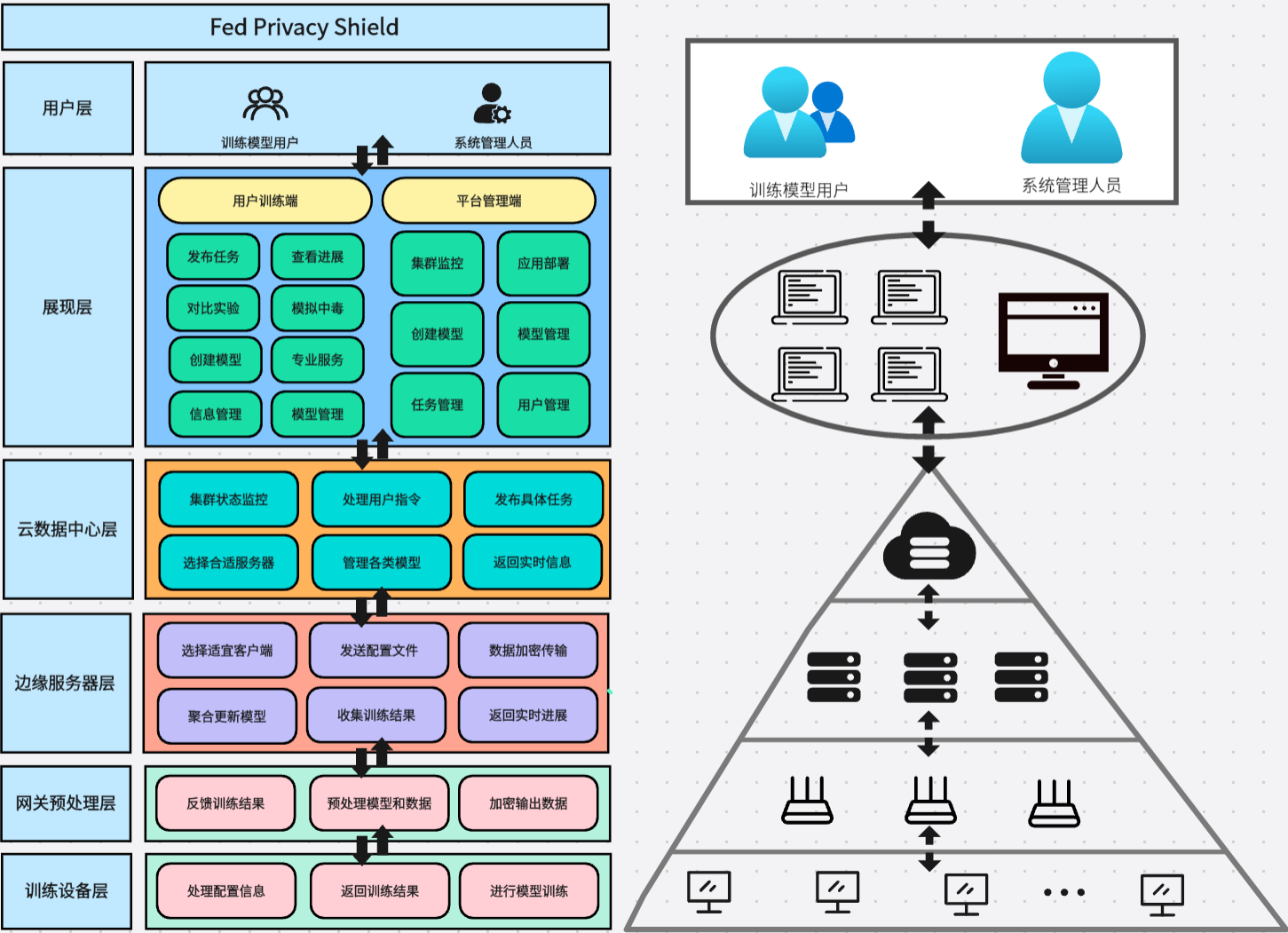


图1-1 平台软硬件架构设计图

**1) 用户层**

Fed Privacy Shield平台共设计两类用户：训练模型的用户和平台管理人员。本平台适合于有隐私保护和联邦学习计算需求的用户，兼顾医疗、金融和智能驾驶等专业性服务。用户接入平台后，发起进展查询、提交计算任务等请求。平台管理人员则通过专用的管理端监控平台状态，进行任务和资源的管理。

**2) 展现层**

Fed Privacy Shield平台针对两类用户分别设计用户端和管理员端，以Web平台为基础，根据不同用户角色的业务功能需求，设计不同的功能模块。管理员通过这一层监控平台状态、部署应用程序，同时还可执行平台维护和更新操作。用户在展现层进行任务提交、进展监控和查看展示。本层的设计旨在满足不同用户角色的需求，通过精心设计的用户界面(UI)和用户体验(UX)，提高平台的整体操作效率。

1. **云数据中心层**

云数据中心层是平台的数据处理和计算中心。本层能够进行大规模的数据处理和复杂的计算任务。它接受来自展现层的用户请求，协调各计算节点执行任务，并对结果进行汇总和分析。同时，它还负责平台中数据的存储管理和数据的安全保护，确保数据的私密性和完整性。

1. **边缘服务层**

边缘服务层位于云数据中心与用户之间，发挥分配训练任务，聚合更新模型，返回实时进展等功能。本层包含一系列分布式的边缘服务器，处理用户的局部数据和计算请求，聚合和更新模型，并将结果及时反馈给用户。

1. **网关预处理层**

网关预处理层作为平台的前置处理层，负责初步处理来自用户层的原始数据。通过数据清洗、转换和加密等预处理操作，确保数据在传输到边缘服务层或云数据中心层时的安全性和高效性。本层的设计强调了对数据的初步把控和优化，提高了后续处理的效率和质量。

**6) 训练设备层**

训练设备层是平台的基础设施层，由大量的训练设备组成，包括但不限于GPU服务器、专用计算机和其他硬件资源。本层的主要功能是提供稳定而强大的计算能力支持联邦学习模型的训练。设备之间协作训练，模型在本地迭代更新。

#### **1.1.1.2 技术架构**

Fed Privacy Shield平台采用分布式微服务架构，整体架构支撑着从客户端到服务端的完整请求流程，提供了高可用性和可伸缩性。系统技术架构图如图1-2所示：

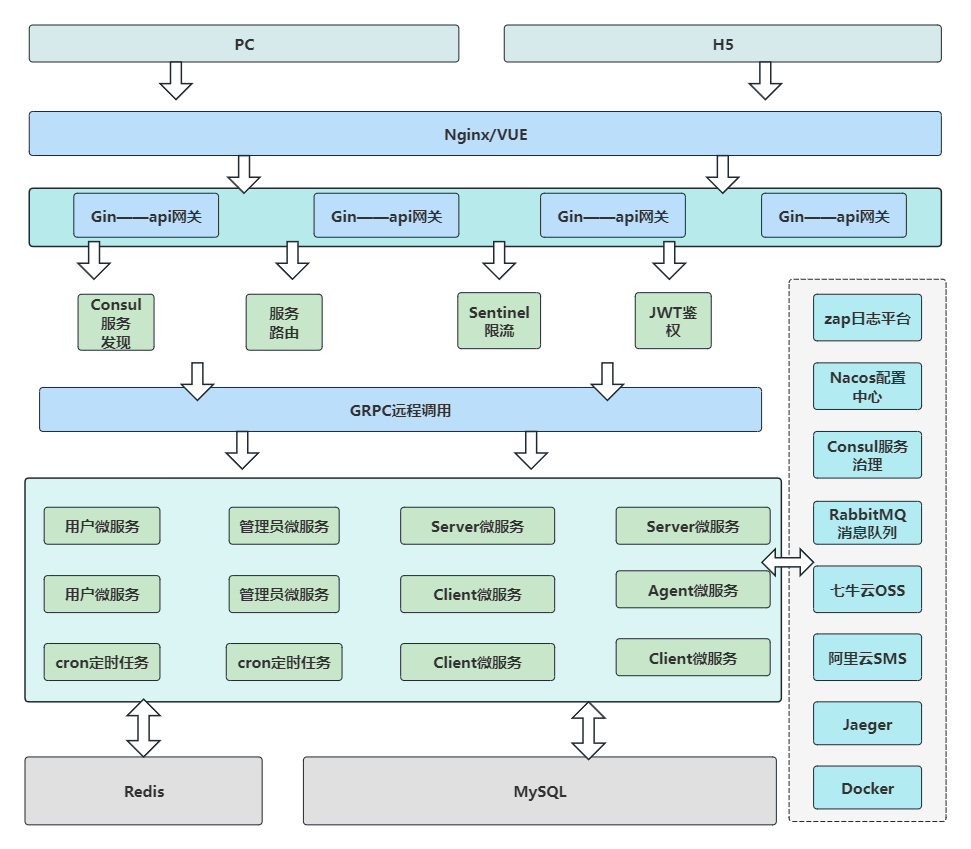


图1-2 平台技术框架图

Fed Privacy Shield平台作为一个高度解耦的微服务平台，通过 Nginx 和 Vue.js 框架实现的前端用户界面与后端服务进行交互。后端服务使用 Go 语言开发的 Gin 框架提供 RESTful API，利用 Consul 进行服务发现与配置，同时集成 Sentinel 实现流量控制和服务熔断以保障平台稳定性。平台鉴权通过 JWT 实现，而服务间调用则通过 gRPC 实现高效通信。

此外，微服务架构中包含了专门的用户服务、管理员服务、用于聚合的Server服务、训练模型的Client服务和充当“证人”的Agent服务，并通过如 zap 日志库、Nacos 配置中心、RabbitMQ 消息队列等周边支持服务来增强日志管理、配置管理和异步通信能力。

平台监控采用 Jaeger 进行分布式追踪，同时结合 Docker 容器化技术以优化服务的部署和扩展。数据存储方面，使用 Redis 进行缓存管理以提升响应速度，MySQL 处理关系数据持久化。整个架构充分考虑了现代化微服务所需的可伸缩性、弹性和容错性，适用于复杂的隐私计算平台。

### 1.1.2 总体流程设计

结合系统功能需求与非功能需求，设计系统总体流程如图1-3所示

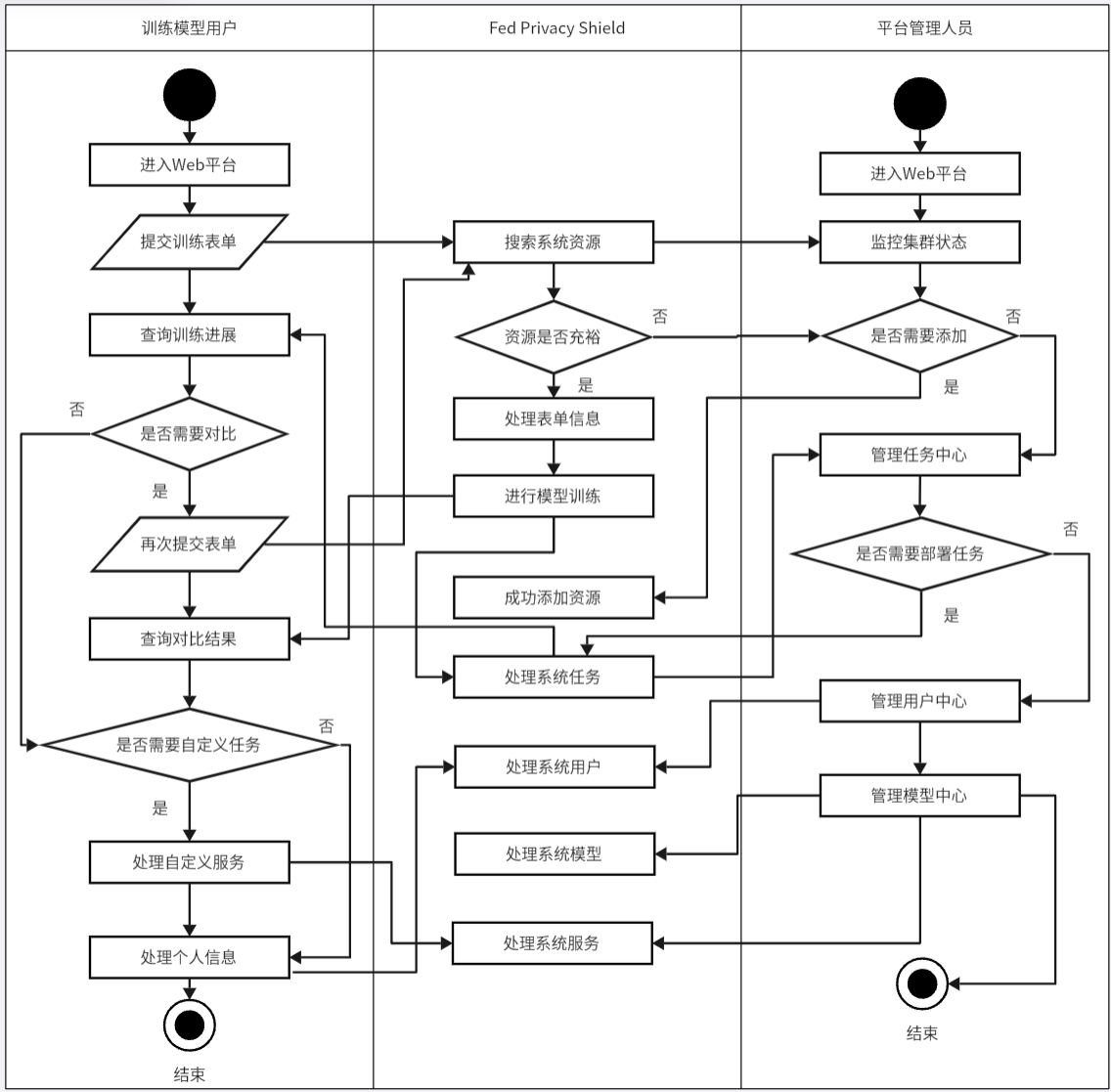


图1-3 平台执行流程泳道图

### **1.1.3 LOGO设计**

Fed Privacy Shield平台LOGO如图1-4所示



图1-4 Fed Privacy Shield平台LOGO图

Fed Privacy Shield平台LOGO以深蓝色为主色调，暗含技术和专业的形象。蓝色在心理学上常与智慧、稳定和安全性关联，这与平台致力于提供安全可靠的联邦学习环境的核心价值观相吻合。蓝色的选择也体现了技术的冷静和客观，暗示平台能在处理大规模数据和复杂计算时保持高效和精确。

LOGO图形由两个关键要素构成，一是盾，二是抽象的云计算图形。盾的图案强调了平台对于隐私保护的承诺，暗喻通过联邦学习模式，用户的数据将得到良好的保护，符合用户对隐私保护的需求。而云计算图形则隐喻了云数据中心层的功能，代表着平台能够通过云端处理大数据，支持强大的计算能力。

## 1.2 功能设计

Fed Privacy Shield平台立足于web端，分为训练模型用户和后台管理人员两大部分，共计18个模块。平台功能模块如图1-5所示

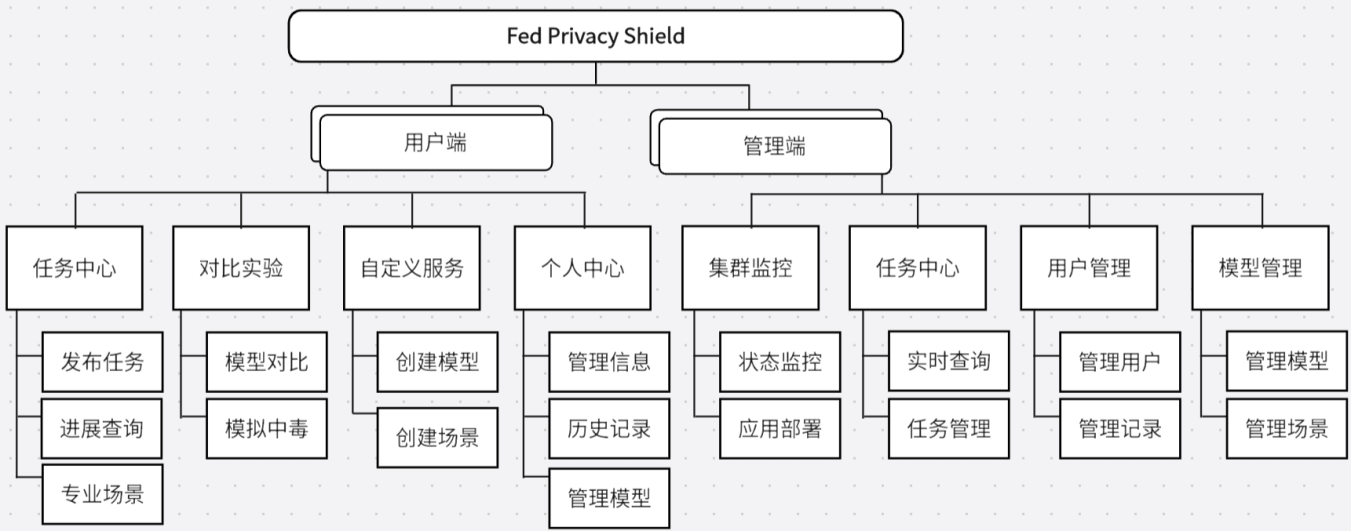


图1-5 平台功能模块图

1.2.1 用户端功能设计

用户端发布于Web平台，功能主要分为任务中心、对比实验、自定义服务和个人中心四大模块。在任务中心模块中，分为发布任务、进展查询和专业场景共三个功能点；对比实验模块分为模型对比和模型中毒共两个功能点；自定义服务模块分为创建模型和创建场景共两个功能点；个人中心分为管理信息、历史记录和管理模型共三个功能点。

详细功能用例规约如下。

**1.2.1.1发布任务**

表1-1 发布任务用例规约

|  |  |
| --- | --- |
| **属性** | **内容** |
| **功能模块** | 训练模型客户Web端——任务中心 |
| **功能描述** | 用户设置任务相关的各种参数，如优化算法、迭代次数、隐私加密方案及设备数量，发布新的联邦学习任务；并可利用平台提供的资源评估功能预估任务的可行性、所需资源和预计完成时间 |
| **执行者** | 注册用户 |
| **前置条件** | 1、用户必须已注册并成功登录平台。  2、用户具备发布任务所需的权限和足够的平台资源。 |
| **后置条件** | 1、新的训练任务在平台中创建并开始执行。  2、用户的资源使用情况更新 |
| **主过程描述** | 1、用户进入任务发布页面，开始创建新任务。  2、用户输入任务描述并设置训练任务的相关参数（如优化算法、迭代次数等）。  3、用户选择隐私加密方案，并指定所需的设备数量。  4、平台提供实时资源评估，提示用户任务的可行性。  5、用户确认任务设置后，提交任务至平台。  6、平台验证任务设置的正确性，创建新任务并开始执行。 |
| **分支过程描述** | 1.1、用户点击“创建新任务”按钮，导航至任务创建页面。  2.1、用户在表单中填写任务描述，并选择或输入相应的训练参数。  3.1、用户根据需求选择隐私保护方案。  3.2、用户指定训练所需的设备数量。  5.1、用户提交任务前预览所有设置，并可返回进行编辑。  5.2、用户点击“提交任务”，平台处理并开始任务调度。 |
| **异常过程描述** | 2.1.1、如果用户设置的参数不符合要求，平台提示错误并要求修正。  4.1.1、如果用户的资源不足以执行任务，平台提供资源不足的提示。  6.1.1、如果平台无法创建任务，则通知用户并记录错误。 |
| **业务规则** | 1、所有任务参数必须满足平台规定的格式和范围要求。  2、平台资源评估必须在用户提交任务之前完成。  3、用户提交的任务必须通过平台验证，包括参数校验和资源评估。 |
| **涉及实体** | 训练任务表、部署信息表 |

**1.2.1.2进展查询**

表1-2 进展查询用例规约

|  |  |
| --- | --- |
| **属性** | **内容** |
| **功能模块** | 训练模型客户Web端——任务中心 |
| **功能描述** | 用户通过交互式仪表板实时监控其发布的训练任务状态，包括任务进度、资源使用情况和模型性能等多维度数据。 |
| **执行者** | 注册用户 |
| **前置条件** | 1、用户必须已注册并成功登录平台。  2、用户已发布至少一个训练任务。 |
| **后置条件** | 1、用户能够获得最新的任务进展信息。  2、平台记录用户的查询活动。 |
| **主过程描述** | 1、用户打开任务中心。  2、用户从任务列表中选择一个任务。  3、平台显示选定任务的当前状态和性能指标。  4、用户查看任务的详细进度和资源消耗数据。  5、用户在需要时刷新数据，以获取最新信息。 |
| **分支过程描述** | 2.1、用户点击任务列表中的某个任务。  3.1、平台根据用户的选择显示相关的实时数据和图表。  4.1、用户点击“刷新”按钮。  4.2、平台更新并显示最新的任务数据。 |
| **异常过程描述** | 3.1.1、如果所选任务无法显示状态，平台提示“任务状态暂时不可用”。  4.1.1、如果刷新过程中发生错误，平台提示“无法更新数据，请稍后再试” |
| **业务规则** | 1、仪表板上显示的数据必须是实时的或具有极短的延迟。  2、用户每次进入仪表板时平台都应自动更新数据 |
| **涉及实体** | 监测任务表、部署信息表 |

**1.2.1.3专业场景**

表1-3 专业场景配置用例规约

|  |  |
| --- | --- |
| **属性** | **内容** |
| **功能模块** | 训练模型客户Web端——任务中心 |
| **功能描述** | 为用户提供专门针对高度专业化领域的服务，包括但不限于医疗影像分析、金融市场分析和车联网数据处理。用户可以访问高级模型和数据集，使用专业的分析工具，并获得定制的模型调优和数据处理服务 |
| **执行者** | 注册用户 |
| **前置条件** | 1、用户必须已注册并成功登录平台。  2、用户已识别并选择特定的应用场景。 |
| **后置条件** | 1、用户获取了所需的模型和数据集访问权限。  2、平台记录用户对专业场景的使用情况 |
| **主过程描述** | 1、用户进入专业场景模块。  2、用户选择一个特定的应用场景。  3、平台展示可用的高级模型和数据集。  4、用户选择所需的模型或数据集。  5、用户使用专门的分析工具开始数据分析或模型训练。  6、用户请求定制化服务，如模型调优或数据处理。 |
| **分支过程描述** | 1.1、用户在界面中选择“专业场景”选项。  2.1、用户在列出的领域中选择特定的应用场景。  3.1、用户浏览不同的模型和数据集选项。  3.2、用户根据项目需求选择适当的模型或数据集。  4.1、用户启动分析工具，进行数据处理或模型训练。  5.1、用户通过填写表单来明确定制服务的需求。  5.2、平台收到定制请求后，为用户提供相应的专业支持。 |
| **异常过程描述** | 3.1.1、如果所选应用场景下没有可用的模型或数据集，平台提示“当前领域没有可用资源”。  5.1.1、如果用户请求的定制服务超出平台能力范围，平台提示“所请求服务不可行，请联系管理员” |
| **业务规则** | 1、用户选择的应用场景必须对应有可用的高级模型和数据集。  2、平台必须能够提供用户选择领域内的分析工具和定制服务。  3、所有定制服务请求应经过平台验证，确认可行性后方可执行。 |
| **涉及实体** | 监测任务表 |

**1.2.1.4模型对比**

表1-4 模型对比用例规约

|  |  |
| --- | --- |
| **属性** | **内容** |
| **功能模块** | 训练模型客户Web端——对比试验 |
| **功能描述** | 用户可以在不同数据集上进行多个模型的性能对比，包括响应时间、准确率、资源消耗等关键性能指标。平台会自动生成详细的对比报告，包括性能图表和关键指标的统计分析 |
| **执行者** | 注册用户 |
| **前置条件** | 1、用户必须已注册并成功登录平台。  2、用户已访问对比实验模块，并且拥有至少两个训练模型 |
| **后置条件** | 1、平台生成模型性能的对比报告。  2、用户可以访问和下载报告。 |
| **主过程描述** | 1、用户进入对比实验模块。  2、用户选择要进行对比的模型。  3、用户设定或确认对比的数据集。  4、平台执行模型对比，处理性能数据。  5、平台生成对比报告，展示性能图表和关键指标统计。  6、用户查看和分析报告。 |
| **分支过程描述** | 1.1、用户在界面中选择“对比实验”选项。  2.1、用户在可用模型列表中勾选需要对比的模型。  3.1、用户选择特定数据集或使用默认设置。  4.1、平台开始计算模型在数据集上的表现并收集性能数据。  5.1、用户选择查看详细报告或下载报告以进行进一步分析。。 |
| **异常过程描述** | 2.1.1、如果用户尚未选择足够的模型进行对比，平台提示“请选择至少两个模型进行对比”。  4.1.1、如果对比过程中发生错误，平台提示“模型对比失败，请重试或联系管理员”。 |
| **业务规则** | 1、至少需要两个模型才能执行对比实验。  2、所选择的数据集必须适用于所有被比较的模型。  3、平台必须记录每次对比实验的结果，以便生成报告和进一步的分析。 |
| **涉及实体** | 模型表、数据集表、监测任务表 |

**1.2.1.5模型中毒**

表1-5 模型中毒用例规约

|  |  |
| --- | --- |
| **属性** | **内容** |
| **功能模块** | 训练模型客户Web端——对比试验 |
| **功能描述** | 提供模型中毒的检测和防御机制。用户可以上传模型进行安全性检测，平台将自动识别潜在的中毒攻击，并提供修复建议或安全增强策略。用户也可以进行模型中毒攻击的模拟，定义攻击类型和强度，进行实验并分析结果 |
| **执行者** | 注册用户 |
| **前置条件** | 1、用户必须已注册并成功登录平台。  2、用户已访问对比实验模块，并准备进行模型中毒检测或模拟。 |
| **后置条件** | 1、平台完成模型的安全性检测或中毒攻击模拟。  2、用户接收到检测报告或模拟结果分析。 |
| **主过程描述** | 1、用户选择进行模型中毒检测或模拟攻击。  2、用户上传模型或选择平台中的模型进行检测。  3、平台分析模型数据，识别潜在中毒攻击。  4、平台提供可能的中毒攻击的检测报告。  5、如进行攻击模拟，用户设定攻击参数并启动模拟。  6、平台执行模拟，追踪并分析学习过程。  7、平台生成模拟的详细报告供用户分析。 |
| **分支过程描述** | 1.1、用户在界面中选择“模型中毒检测”或“模型中毒模拟”选项。  2.1、用户通过文件上传功能上传本地模型。  3.1、平台自动运行安全性检测算法。  4.1、如果是模拟，用户定义攻击类型和强度，选择恶意数据。  5.1、平台根据用户设定的参数执行中毒攻击模拟。  6.1、平台完成模拟后，展示模拟过程和效果对比。 |
| **异常过程描述** | 2.1.1、如果上传的模型格式不符合要求，平台提示“不支持的模型格式”。  3.1.1、如果平台无法处理模型，提示“模型处理失败，请检查模型有效性或联系管理员”。  5.1.1、如果用户设定的攻击参数不合理，平台提示“攻击参数设置错误，请重新设定”。 |
| **业务规则** | 1、所有上传的模型必须符合平台规定的格式和标准。  2、平台必须能够准确地识别并报告模型中毒的潜在风险。  3、模型中毒模拟必须在安全的环境中执行，不影响实际的模型和数据。 |
| **涉及实体** | 检测任务表 |

**1.2.1.6创建模型**

表1-6 创建模型用例规约

|  |  |
| --- | --- |
| **属性** | **内容** |
| **功能模块** | 训练模型客户Web端——自定义服务 |
| **功能描述** | 用户可以通过平台提供的内置模型构建器创建个性化的模型，选择不同的算法组件，调整模型的层次结构，以及定制模型的超参数。此外，平台还支持上传现有模型以进行微调。 |
| **执行者** | 注册用户 |
| **前置条件** | 1、用户必须已注册并成功登录平台。  2、用户已进入自定义服务模块 |
| **后置条件** | 1、用户创建的新模型在平台中被保存。  2、用户可以开始使用新模型进行训练任务。 |
| **主过程描述** | 1、用户选择创建新模型的选项。  2、用户通过模型构建器选择算法组件和调整层次结构。  3、用户定制模型的超参数设置。  4、用户提交模型构建请求。  5、平台根据用户配置生成新模型。  6、平台提供模型的摘要信息，包括参数和结构，供用户确认。  7、用户保存新模型以进行后续的训练任务。 |
| **分支过程描述** | 1.1、用户在自定义服务模块选择“创建模型”功能。  2.1、用户在模型构建器中选择算法组件，如神经网络层、激活函数等。  3.1、用户输入超参数的值，如学习率、批大小等。  4.1、用户审查所有选择和设置，确认无误后提交。  5.1、平台处理用户请求，使用选择的组件和超参数配置构建新模型。  6.1、平台展示新模型的详细信息，包括层次结构图和参数列表。 |
| **异常过程描述** | 3.1.1、如果用户输入的超参数不符合规定的范围或格式，平台提示“超参数设置错误，请重新输入”。  5.1.1、如果平台在构建模型时发生内部错误，平台提示“模型创建失败，请稍后重试或联系管理员”。 |
| **业务规则** | 1、用户必须按照平台支持的格式和规则设置算法组件和超参数。  2、用户提交的模型构建请求必须包含所有必需的信息才能被处理。  3、平台必须在用户提交后验证模型配置的可行性，并提供反馈。 |
| **涉及实体** | 模型表 |

**1.2.1.7创建场景**

表1-7 创建场景用例规约

|  |  |
| --- | --- |
| **属性** | **内容** |
| **功能模块** | 训练模型客户Web端——自定义服务 |
| **功能描述** | 用户能够根据自己的业务需求开发专属算法和服务流程。该功能允许用户在特定应用场景下创建场景，整合多源数据，配置数据流，并部署到自定义的硬件环境中。 |
| **执行者** | 注册用户 |
| **前置条件** | 1、用户必须已注册并成功登录平台。  2、用户已进入自定义服务模块，并已经或正在创建相关模型。 |
| **后置条件** | 1、用户定义的服务在平台中创建并可以配置。  2、平台记录服务的创建和配置详情。 |
| **主过程描述** | 1、用户选择创建新服务的选项。  2、用户定义服务的业务逻辑和数据处理流程。  3、用户整合不同来源的数据输入。  4、用户配置服务的数据流和处理步骤。  5、用户选择目标硬件环境，并配置部署选项。  6、平台验证服务配置的正确性和可执行性。  7、平台在用户确定的硬件环境中部署服务。 |
| **分支过程描述** | 1.1、用户在自定义服务模块选择“创建场景”功能。  2.1、用户通过可视化界面或代码编辑器定义服务逻辑。  3.1、用户配置数据输入点，选择或上传数据源。  4.1、用户拖放或编码以设置数据处理的各个步骤。  5.1、用户在平台支持的硬件选项中选择并设置部署参数。  6.1、平台执行配置检查，确保所有组件和流程无误。  7.1、用户审查服务配置摘要，并执行部署命令。 |
| **异常过程描述** | 4.1.1、如果数据流配置错误，平台提示“数据流配置不正确，请根据指导修正”。  6.1.1、如果服务配置存在不兼容的硬件需求，平台提示“所选硬件环境与服务配置不兼容”。 |
| **业务规则** | 1、所有服务配置必须满足平台规定的格式和逻辑规范。  2、用户必须确保数据源的合法性和兼容性。  3、服务部署必须在平台验证无误后才能进行。 |
| **涉及实体** | 部署信息表 |

**1.2.1.8管理信息**

表1-8 管理信息用例规约

|  |  |
| --- | --- |
| **属性** | **内容** |
| **功能模块** | 训练客户Web端——个人中心 |
| **功能描述** | 用户管理个人基础信息，如密码、电话、头像等 |
| **执行者** | 注册用户 |
| **前置条件** | 1、用户注册并成功登录平台 |
| **后置条件** | 1、被修改的用户个人基础信息在平台中更新 |
| **主过程描述** | 1、用户进入密码修改页面，根据提示修改账户密码；  2、用户进入电话修改页面，根据提示修改电话号码；  3、用户进入头像修改页面，根据提示修改头像 |
| **分支过程描述** | 1.1 、用户点击“修改密码”按钮，进入密码修改页面；  1.2 、用户输入现密码、新密码、重复新密码，点击“确认修改”按钮；  2.1 、用户点击“修改电话”按钮，进入电话修改页面；  2.2 、用户输入新电话号码，点击“发送验证码”按钮；  2.3 、用户输入短信验证码，点击“确认修改”按钮；  3.1 、用户点击“修改头像”按钮，进入头像修改页面；  3.2 、用户在手机相册中选择新头像，点击“确认修改”按钮 |
| **异常过程描述** | 1.2.1、新密码与现密码相同，提示“新密码不能与现密码相同！”；  1.2.2、新密码不符合密码强度规范，提示“不符合密码强度，请重现输入！”；  1.2.3、重复密码与新密码输入不一致，提示“两次密码输入不一致！”；  2.2.1、用户输入的电话号码格式不合法，提示“请输入有效电话号码！”；  2.3.1、短信验证码验证失败，提示“请输入正确验证码” |
| **业务规则** | 1、密码强度要求：8～14 位，包含数字、中英文及特殊字符；  2、电话号码每月仅能修改 1 次 |
| **涉及实体** | 用户表 |

**1.2.1.9历史记录**

表1-9 历史记录管理用例规约

|  |  |
| --- | --- |
| **属性** | **内容** |
| **功能模块** | 训练客户Web端——个人中心 |
| **功能描述** | 用户查看和管理个人历史记录，其中包括任务提交、查询、下载结果等活动。平台根据这些活动提供操作建议和改进提示。 |
| **执行者** | 注册用户 |
| **前置条件** | 1、用户必须已注册并成功登录平台。  2、用户已进行过至少一项活动。 |
| **后置条件** | 1、用户可以查阅和了解历史活动详情。  2、平台可能根据用户的历史活动提供个性化建议。 |
| **主过程描述** | 1、用户进入个人中心的历史记录模块。  2、平台显示用户所有的历史活动列表。  3、用户选择查看特定活动的详细信息。  4、平台提供选定活动的详细信息，包括时间、类型、结果等。  5、用户查看平台给出的操作建议和改进提示。  6、用户根据需要对历史活动进行管理，如删除记录或标记重要活动。 |
| **分支过程描述** | 1.1、用户点击“历史记录”标签。  2.1、平台按时间顺序展示用户历史活动。  3.1、用户点击某一历史记录条目。  4.1、平台展示该历史活动的完整详情。  5.1、用户利用提供的信息和提示进行自我评估和计划未来的活动。 |
| **异常过程描述** | 3.1.1、如果平台无法检索特定活动的详细信息，平台提示“信息无法加载，请重试”。  6.1.1、如果用户尝试删除或修改关键历史记录时，平台提示“此项操作不被允许”。 |
| **业务规则** | 1、用户的所有活动必须被准确地记录和存储。  2、用户有权查看自己的历史活动记录。  3、平台根据用户活动的历史模式提供个性化建议。 |
| **涉及实体** | 训练任务表、部署信息表 |

**1.2.1.10管理模型**

表1-10 管理模型用例规约

|  |  |
| --- | --- |
| **属性** | **内容** |
| **功能模块** | 训练客户Web端——个人中心 |
| **功能描述** | 用户查看和管理自己创建或上传的所有模型。功能包括进行版本控制、模型共享、性能追踪和维护操作。用户还可以对模型进行分类和标记，分享给其他用户或小组，并设置访问权限。 |
| **执行者** | 注册用户 |
| **前置条件** | 1、用户必须已注册并成功登录平台。  2、用户已创建或上传至少一个模型。 |
| **后置条件** | 1、用户完成对模型的管理操作，如更新、共享或删除模型。  2、平台更新模型信息和权限设置，反映用户的操作。 |
| **主过程描述** | 1、用户进入个人中心的“管理模型”模块。  2、平台显示用户所有模型的列表。  3、用户选择一个模型进行更详细的查看或执行管理操作。  4、用户可以执行诸如版本更新、共享、分类标记等操作。  5、对于需要共享的模型，用户设置访问权限和分享对象。  6、平台根据用户操作更新模型的状态和权限。 |
| **分支过程描述** | 1.1、用户选择“管理模型”选项。  2.1、平台根据用户标识展示其所有模型的概览。  3.1、用户点击特定模型条目以访问其详细页面。  4.1、用户在模型详情页面选择进行版本控制或更新模型描述。  4.2、用户选择共享模型，设置目标用户或小组和访问级别。  5.1、用户对模型进行分类和添加标签，以便更好地组织和检索。 |
| **异常过程描述** | 4.1.1、如果用户尝试访问未授权的模型，平台提示“您没有权限执行此操作”。  5.1.1、如果设置的访问权限超出平台允许的范围，平台提示“无效的权限设置”。 |
| **业务规则** | 1、用户必须拥有模型的所有权或足够权限才能进行管理操作。  2、所有模型共享和权限更改操作必须遵守平台安全策略。  3、模型的任何更改或更新都应记录并可追踪，以支持版本控制。 |
| **涉及实体** | 模型表 |

**1.2.2 管理端功能设计**

管理端发布于Web平台，主要分为集群监控、任务中心、用户管理和模型管理四大模块。集群监控模块分为状态监控和应用部署共两大功能点；任务中心模块分为实时查询和任务管理共两大功能点；用户管理模块分为管理用户和管理记录共两大功能点；模型管理分为管理模型和管理场景两大功能点。

**1.2.2.1状态监控**

表1-11 状态监控用例规约

|  |  |
| --- | --- |
| **属性** | **内容** |
| **功能模块** | 平台管理人员Web端——集群监控 |
| **功能描述** | 监控平台的整体状态和管理所有的服务。管理员可以在本模块上查看云数据中心、边缘服务器以及端设备的状态，如在线状态、IP地址、资源使用情况等。管理员可以迅速识别潜在问题，并进行相关的管理操作，如设备上下线、资源调度等 |
| **执行者** | 平台管理员 |
| **前置条件** | 1、管理员必须已登录平台。  2、管理员具有集群监控的访问权限。 |
| **后置条件** | 1、平台状态和服务信息得到实时更新。  2、管理员完成对集群的监控或管理操作。 |
| **主过程描述** | 1、管理员进入集群监控模块。  2、平台展示当前的平台状态概览，包括所有服务和设备的状态。  3、管理员选择特定的服务或设备以查看详细信息。  4、平台展示所选服务或设备的详细状态信息。  5、管理员根据需要执行管理操作，如调整资源分配、设备上下线。  6、平台根据管理员的操作更新状态信息。 |
| **分支过程描述** | 1.1、管理员点击“集群监控”选项进入该模块。  2.1、平台按类别展示服务状态和设备列表。  3.1、管理员选择查看特定服务的状态。  3.2、管理员选择查看特定设备的详细信息。  4.1、平台提供选定服务或设备的运行状态、性能指标和配置信息。  5.1、面对资源使用率高的情况，管理员增加资源配额或调度任务。  5.2、对于故障设备，管理员执行重启或下线操作。 |
| **异常过程描述** | 4.1.1、如果平台无法获取服务或设备的状态信息，平台提示“状态信息获取失败，请稍后重试”。  5.1.1、如果管理员尝试的操作超出权限范围，平台提示“您没有执行此操作的权限”。 |
| **业务规则** | 1、只有具有相应权限的管理员才能查看和管理平台状态和设备信息。  2、平台状态信息必须实时更新，确保管理员能够基于最新数据进行决策。  3、所有管理操作都应记录日志，以便审计和故障排查。 |
| **涉及实体** | 部署信息表、监测任务表 |

**1.2.2.2应用部署**

表1-12 应用部署用例规约

|  |  |
| --- | --- |
| **属性** | **内容** |
| **功能模块** | 平台管理人员Web端——集群监控 |
| **功能描述** | 管理员管理和监控集群中所有部署的服务。包括查看应用的当前状态、配置和日志，管理容器生命周期，实现应用的快速迭代和高可用部署。管理员可以进行应用的部署、更新、回滚和停止操作，以确保服务按预期运行 |
| **执行者** | 平台管理员 |
| **前置条件** | 1、管理员必须已登录平台。  2、管理员具有对应用部署和管理的权限。 |
| **后置条件** | 1、所管理的应用状态更新，反映出管理员的操作。  2、平台记录操作日志，用于审计和故障排查。 |
| **主过程描述** | 1、管理员访问应用部署模块。  2、平台展示所有部署的应用列表及其状态。  3、管理员选择一个应用以查看更多详情。  4、平台提供所选应用的状态、配置和日志信息。  5、管理员根据需要对应用执行部署、更新、回滚或停止等操作。  6、平台执行管理员的操作，并更新应用状态。 |
| **分支过程描述** | 1.1、管理员在集群监控界面选择“应用部署”功能。  2.1、平台按照服务分类展示当前部署的所有应用。  3.1、管理员点击某个应用条目，进入详情页面。  4.1、管理员查看应用的运行状态、配置信息和最近的操作日志。  5.1、管理员执行更新操作，选择新的镜像版本或配置文件。  5.2、管理员执行回滚操作，选择之前的某个稳定版本。  5.3、管理员决定停止某个不再需要的服务。  6.1、平台验证管理员操作的合法性，执行操作并反馈执行结果。 |
| **异常过程描述** | 5.1.1、如果管理员选择的应用配置不符合要求，平台提示“配置文件错误，请检查后再试”。  6.1.1、如果操作执行失败，平台提示“操作失败，请查看日志了解详情”。 |
| **业务规则** | 1、管理员在执行应用部署和管理操作前，必须确保所选操作与应用的当前状态兼容。  2、所有操作必须记录在平台操作日志中，包括操作类型、时间和执行结果。  3、平台应提供实时反馈，让管理员知道操作是否成功，如不成功应提供错误信息。 |
| **涉及实体** | 部署信息表、训练任务表 |

**1.2.2.3实时查询**

表1-13 实时查询用例规约

|  |  |
| --- | --- |
| **属性** | **内容** |
| **功能模块** | 平台管理人员Web端——实时查询 |
| **功能描述** | 管理员实时查询平台中的活跃用户及他们当前的任务状态。通过一个集成的控制面板，管理员可以快速响应用户请求和问题，优化用户体验和平台性能。 |
| **执行者** | 平台管理员 |
| **前置条件** | 1、管理员必须已登录平台。  2、管理员具有查询平台用户状态的权限。 |
| **后置条件** | 1、管理员获得所查询用户的实时活动和状态信息。  2、平台记录查询操作，用于审计和性能优化。 |
| **主过程描述** | 1、管理员访问用户管理模块的实时查询功能。  2、平台展示一个动态更新的用户活动和任务状态的实时视图。  3、管理员输入或选择特定用户进行查询。  4、平台立即显示所选用户的当前活动详情，包括进行中的任务、资源使用情况等。  5、管理员根据需要进行进一步的操作或调整，如调整用户权限、资源分配等。  6、平台根据管理员的操作更新用户的状态或资源配置。 |
| **分支过程描述** | 1.1、管理员在用户管理界面选择“实时查询”选项。  2.1、平台显示所有用户的活动概览，包括在线状态和当前任务。  3.1、管理员通过搜索框输入用户名称或从列表中选择用户。  4.1、平台展示选定用户的详细活动记录和资源使用情况。  5.1、面对资源使用高的用户，管理员可能决定增加其资源配额。  5.2、若发现异常用户行为，管理员可能暂停其任务或限制其权限。 |
| **异常过程描述** | 3.1.1、如果没有找到指定的用户信息，平台提示“未找到用户”。  4.1.1、如果无法加载用户的活动详情，平台提示“加载用户数据失败，请稍后重试” |
| **业务规则** | 1、实时查询必须显示最新的数据，确保管理员可以基于当前情况做出决策。  2、管理员对用户的任何操作都应记录在平台日志中，包括时间、操作内容和操作者信息。  3、平台应保障用户数据的隐私和安全，确保管理员仅访问其权限范围内的信息。 |
| **涉及实体** | 用户表、部署信息表 |

**1.2.2.4任务管理**

表1-14 任务管理用例规约

|  |  |
| --- | --- |
| **属性** | **内容** |
| **功能模块** | 平台管理人员Web端——任务管理 |
| **功能描述** | 在控制面板上分配任务给用户，调整任务优先级和资源分配，并监控任务的执行情况。通过优化任务分配和资源利用，管理员确保按时完成任务，提高平台的整体效率和用户满意度。 |
| **执行者** | 平台管理员 |
| **前置条件** | 1、管理员必须已登录平台。  2、管理员具有管理任务和分配资源的权限。 |
| **后置条件** | 1、选定的用户任务更新，反映新的分配或调整。  2、平台记录管理员的操作，用于审计和性能评估。 |
| **主过程描述** | 1、管理员访问用户管理模块下的任务管理功能。  2、平台展示当前所有用户的任务列表和状态。  3、管理员选择特定用户或任务进行管理。  4、管理员可以执行如下操作：分配新任务、调整现有任务的优先级、重新分配资源。  5、管理员提交操作后，平台更新任务状态和用户资源配置。  6、平台提供操作结果反馈给管理员。 |
| **分支过程描述** | 1.1、管理员在用户管理界面选择“任务管理”选项。  2.1、平台按用户或任务状态显示任务列表。  3.1、管理员通过搜索或筛选找到特定用户或任务。  4.1、管理员为选定用户分配新的任务，设置任务参数和截止日期。  4.2、管理员调整选定任务的优先级，以优化资源使用和任务完成顺序。  4.3、管理员根据任务需求和用户性能重新分配资源。  5.1、平台处理管理员的请求，并实时更新任务和资源配置。  6.1、平台显示操作成功提示或详细的错误信息。 |
| **异常过程描述** | 4.1.1、如果任务参数不符合要求，平台提示“任务参数错误，请检查后再试”。  5.1.1、如果资源重新分配导致资源冲突或超出限制，平台提示“资源分配冲突，请重新分配”。 |
| **业务规则** | 1、所有任务分配和调整必须考虑用户的能力和平台的资源限制。  2、管理员的每一项操作都需记录日志，包括操作类型、时间和受影响的用户或任务。  3、平台应确保任务管理操作不违反任何用户隐私政策或平台安全策略。 |
| **涉及实体** | 部署信息表 |

**1.2.2.5管理用户**

表1-15 管理用户用例规约

|  |  |
| --- | --- |
| **属性** | **内容** |
| **功能模块** | 平台管理人员Web端——管理用户 |
| **功能描述** | 管理所有用户信息和分配任务。管理员可以实时查询平台中的活跃用户和他们当前的任务状态，通过集成的控制面板快速响应用户请求和问题，调整用户权限，设置资源配额，并监控用户的活跃度和平台使用情况 |
| **执行者** | 平台管理员 |
| **前置条件** | 1、管理员必须已登录平台。  2、管理员具有用户管理的权限。 |
| **后置条件** | 1、用户信息或状态更新，反映管理员的操作。  2、平台记录管理员操作，用于审计和用户行为分析。 |
| **主过程描述** | 1、管理员访问管理用户模块。  2、平台展示一个用户列表，包括用户信息和当前状态。  3、管理员选择特定用户进行详细操作。  4、管理员可以执行多种操作，包括更新用户信息、调整权限设置、分配或调整资源配额。  5、对于特定的用户请求或问题，管理员提供即时的帮助或反馈。  6、平台根据管理员的操作更新用户的信息和权限设置。 |
| **分支过程描述** | 1.1、管理员在管理界面选择“管理用户”功能。  2.1、平台按注册时间、活跃度等标准展示用户列表。  3.1、管理员通过搜索或筛选找到特定用户。  4.1、管理员更新用户的个人信息或联系方式。  4.2、管理员增加或减少用户的平台权限，如访问特定模块的能力。  4.3、管理员根据用户的使用情况调整其资源配额。  5.1、管理员回应用户提交的请求或解决用户报告的问题。  6.1、平台实时记录管理员的每项操作，并更新相关用户的信息和状态。 |
| **异常过程描述** | 4.1.1、如果管理员输入的信息格式不正确，平台提示“输入格式错误，请检查后再试”。  4.2.1、如果管理员尝试赋予用户超出其角色允许范围的权限，平台提示“权限设置超出范围”。 |
| **业务规则** | 1、管理员在更改用户信息或权限前必须验证用户的身份和操作的合法性。  2、平台必须记录所有用户信息的更改和管理员操作的详细日志。  3、任何对用户权限和资源配额的调整都必须在不违反平台安全策略的前提下进行。 |
| **涉及实体** | 用户表 |

**1.2.2.6管理记录**

表1-16 管理记录用例规约

|  |  |
| --- | --- |
| **属性** | **内容** |
| **功能模块** | 平台管理人员Web端——管理记录 |
| **功能描述** | 管理平台所有的模型训练任务。管理员可以查看所有任务的详细记录，包括提交时间、执行时长、所需资源和结果。这使得管理员能够进行有效的任务监控、审计和资源优化，支持平台的决策和优化。 |
| **执行者** | 1、管理员必须已登录平台。  2、管理员具有任务记录查看和管理的权限。 |
| **前置条件** | 1、用户必须已注册并成功登录平台。  2、用户已创建或上传至少一个模型。 |
| **后置条件** | 1、任务记录被查询、分析或更新，以反映管理员的监控和管理操作。  2、平台优化任务分配和资源利用，提高执行效率。 |
| **主过程描述** | 1、管理员访问任务中心的“管理记录”模块。  2、平台展示所有任务的记录列表，包括每个任务的基本信息和状态。  3、管理员选择特定任务记录进行深入查看。  4、平台提供所选任务的详细信息，如提交者、执行时间、资源消耗和完成状态。  5、管理员根据任务执行情况进行分析，识别优化点。  6、管理员可以基于分析结果调整任务优先级或重新分配资源。  7、平台根据管理员的操作更新任务记录和资源分配。 |
| **分支过程描述** | 1.1、管理员在任务中心选择“管理记录”功能。  2.1、平台按提交时间、状态或其他标准排序任务记录。  3.1、管理员通过搜索功能定位特定的任务记录。  4.1、管理员查看选定任务的所有相关细节和性能指标。  5.1、对于执行时间长或资源消耗高的任务，管理员评估可能的优化措施。  6.1、管理员执行任务调度优化，如更改执行顺序或增加资源。  7.1、平台记录管理员的操作，并实施相应的任务和资源调整。 |
| **异常过程描述** | 3.1.1、如果所选任务记录不存在或无法访问，平台提示“任务记录不存在或访问受限”。  6.1.1、如果资源调整与现有平台策略冲突，平台提示“资源调整失败，与现有策略冲突”。 |
| **业务规则** | 1、所有任务记录必须真实反映任务的执行情况和资源使用。  2、管理员进行的任何操作都应遵循平台安全和资源分配的策略。  3、平台应记录管理员的每一项操作，包括查看、分析和调整任务记录的活动。 |
| **涉及实体** | 监测任务表 |

**1.2.2.7管理模型**

表1-17 管理模型用例规约

|  |  |
| --- | --- |
| **属性** | **内容** |
| **功能模块** | 平台管理人员Web端——管理模型 |
| **功能描述** | 管理员对所有平台模型进行性能监控、版本控制、比较不同版本性能以及进行及时的模型更新和优化。管理员可以管理模型的生命周期，包括发布新模型、更新现有模型、回滚至之前的版本，以及删除过时的模型。此外，管理员还能设置模型的访问权限，确保只有授权用户可以访问特定模型。 |
| **执行者** | 平台管理员 |
| **前置条件** | 1、管理员必须已登录平台。  2、管理员具有模型管理的权限。 |
| **后置条件** | 1、模型的状态和信息更新，反映管理员的管理操作。  2、平台记录模型管理操作的日志，用于审计和追踪。 |
| **主过程描述** | 1、管理员访问管理模型模块。  2、平台展示所有模型的列表，包括每个模型的状态和版本信息。  3、管理员选择一个模型进行详细管理。  4、管理员可以执行以下操作：更新模型、回滚模型版本、删除模型或设置模型的访问权限。  5、管理员提交操作后，平台更新模型状态并反馈操作结果。  6、平台记录管理员的每一项操作，确保可追踪性和审计性。 |
| **分支过程描述** | 1.1、管理员在管理界面选择“模型管理”功能。  2.1、平台按模型的创建时间、使用频率或其他标准展示模型列表。  3.1、管理员通过搜索或筛选找到特定模型。  4.1、管理员更新模型至新的版本，上传新的模型文件。  4.2、管理员选择回滚操作，将模型恢复到以前的稳定版本。  4.3、对于不再需要的模型，管理员执行删除操作。  4.4、管理员设置或修改模型的访问权限，限定特定用户或用户组的访问。  5.1、平台处理管理员的请求，执行模型更新、回滚、删除或权限设置。  6.1、平台更新操作日志，记录管理员的操作细节和操作时间。 |
| **异常过程描述** | 4.1.1、如果模型文件格式或大小不符合要求，平台提示“模型文件不符合上传标准”。  4.4.1、如果管理员尝试设置的权限超出其操作范围，平台提示“权限设置超出管理员权限”。 |
| **业务规则** | 1、管理员在执行模型更新或回滚前，必须验证模型的兼容性和稳定性。  2、模型的任何更改都应先在测试环境中验证，以确保不会影响平台的正常运行。  3、所有模型管理操作都必须记录在平台日志中，包括操作类型、操作者、时间和操作结果。 |
| **涉及实体** | 模型表 |

**1.2.2.8管理场景**

表1-18 管理场景用例规约

|  |  |
| --- | --- |
| **属性** | **内容** |
| **功能模块** | 平台管理人员Web端——管理场景 |
| **功能描述** | 管理员对所有平台中的服务进行监控和管理。管理员可以查看服务的当前状态、配置信息、性能指标，以及日志信息。此外，管理员有权进行服务的部署、更新、停止和重新启动操作。管理场景功能确保服务能够高效运行，并在出现问题时快速恢复。 |
| **执行者** | 平台管理员 |
| **前置条件** | 1、管理员必须已登录平台。  2、管理员具有对服务进行管理的权限。 |
| **后置条件** | 1、服务的状态或配置更新，以反映管理员的操作。  2、平台记录了服务管理活动的日志，用于后续的审计和故障排查。 |
| **主过程描述** | 1、管理员访问模型管理模块下的“管理场景”功能。  2、平台展示所有服务的列表，包含服务的名称、状态、以及关键性能指标。  3、管理员选择某个服务以查看其详细信息和管理选项。  4、管理员根据服务的运行情况执行相应的管理操作，如更新配置、重新启动服务或停止服务。  5、平台执行管理员的命令，并更新服务的状态或配置。  6、管理员查看操作结果，确保服务按预期运行。  7、平台记录管理员的每项操作，包括操作类型、执行时间和结果。 |
| **分支过程描述** | 1.1、管理员在模型管理界面选择“管理场景”选项。  2.1、平台按照服务的活跃度、资源消耗等指标排序服务列表。  3.1、管理员点击服务名称进入服务详情页面。  4.1、如果服务需要更新，管理员上传新的配置文件或修改现有配置。  4.2、对于不正常运行的服务，管理员执行停止或重新启动操作以恢复服务。  5.1、平台根据管理员的操作执行更新、停止或重启服务，并实时反馈操作状态。  6.1、管理员通过平台提供的实时日志和性能监控确保服务正常运行。  7.1、平台自动记录所有管理操作，包括成功或失败的详细信息。 |
| **异常过程描述** | 4.1.1、如果新配置与现有平台不兼容，平台提示“配置更新失败，存在不兼容项”。  5.1.1、如果服务停止或重启操作失败，平台提示“操作失败，请检查服务状态和日志信息”。 |
| **业务规则** | 1、所有服务管理操作必须遵循平台的安全和兼容性要求。  2、管理员在执行任何操作前应评估操作的影响，避免对平台稳定性和用户体验产生负面影响。  3、操作日志必须详细记录，包括操作者、操作时间、操作类型和操作结果。 |
| **涉及实体** | 部署任务表 |

## 1.3 关键算法设计

本节概述Fed Privacy Shield平台中关键算法设计的核心理念和实现方法。详细介绍了聚合方案、通信传输和安全加密三个主要部分，这些算法的设计对于维护平台的效率、可靠性和安全性至关重要

1.3.1 聚合方案

在Fed Privacy Shield平台中，聚合方案的设计关键在于如何有效地综合来自各个节点的更新，以优化和提升全局模型的性能。本部分将讨论我们采用的多种聚合算法，包括但不限于加权平均、模型修正和梯度剪切等方法。这些算法考虑了不同节点间数据分布的异质性，以及全局模型在不同节点间的一致性，旨在实现高效和公平的聚合

**1.3.1.1聚合方案算法简介**

联邦学习作为一种新兴的分布式机器学习框架，通过在用户私有数据不出域的情况下进行联合建模训练，有效地解决了传统机器学习中的数据孤岛和隐私泄露问题。

在分布式的联邦学习平台中，一系列客户端独立地根据各自的数据集 计算模型参数的更新。这些更新需要被合理地聚合，以形成一个全局模型。

考虑到客户端的数据分布可能不均，聚合方案通常采用加权的方法以反映每个客户端数据的重要性。这里，我们引入权重​,通常与客户端的数据量或质量成正比。

全局模型的更新通过以下方式聚合：

（1-1）

其中 为所有权重的总和。

在实际应用中，每个客户端的模型更新可以通过多种方式计算，例如梯度下降法。一旦计算完成，这些更新就会被送往中央服务器。服务器根据预定的聚合策略更新全局模型。考虑到平台的通信成本，通常还会采取压缩技术来减少每次通信的数据量。

此外，为了进一步优化全局模型的性能，我们可能会加入自适应调整机制，比如：

（1-2）

其中，是一个超参数，用于调整权重以减小噪声较大或偏离全局模型较远的客户端更新的影响。

最终，通过这种方式，联邦学习框架可以协调多个客户端的模型更新，形成一个整体性能优化的全局模型。

**1.3.1.2创新型聚合方案算法设计**

**1)****Fast-PDMM Fed理论的安全聚合方案算法**

首先进行数据的初始化:每个客户端i生成一个虚拟客户端i',并随机地分出一半同分布的数据集给虚拟客户端i'。确定全局迭代次数R,和局部迭代次数K,为设置合适的值,。客户端,虚拟客户端和服务端上的原变量可任意赋值,客户端i与虚拟客户端i'之间的对偶变量的值。客户端i上的对偶变量,虚拟客户端i'上的对偶变量以及服务端s上的对偶变量可以任意初始化。

客户端i将发送给服务端s,服务端s将,发送给客户端i,客户端i和虚拟客户端i'共享所有数据信息(包括变量信息)。

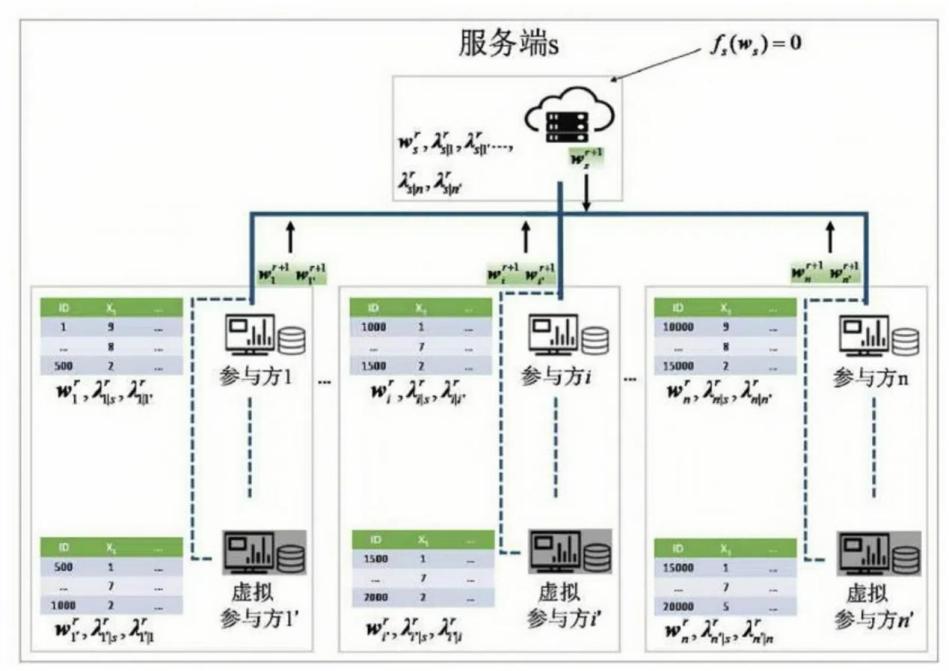


图1-6 Fast-PDMM Fed理论的安全聚合方案算法梯度更新流程

For r=0,...R do

服务端s：原变量的更新

（1-3）

对偶变量的更新：

（1-4）

（1-5）

并将发给客户端。

客户端i和虚拟客户端i':原变量的更新:

（1-6）

（1-7）

对偶变量的更新:

（1-8）

（1-9）

If k == K:

（1-10）

客户端i将发给服务端s。

当客户端接收到服务端s的时，客户端i更新服务端s的对偶变量：

（1-11）

（1-12）

当服务端接收到服务端，时，客户端i更新服务端s的对偶变量：

（1-13）

（1-14）

满足临界条件后结束循环，全局模型由此确定。

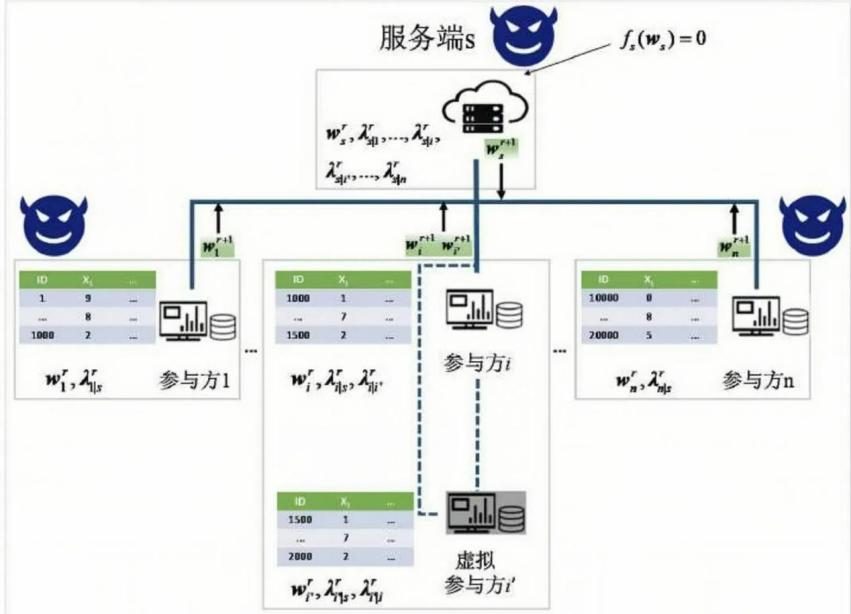
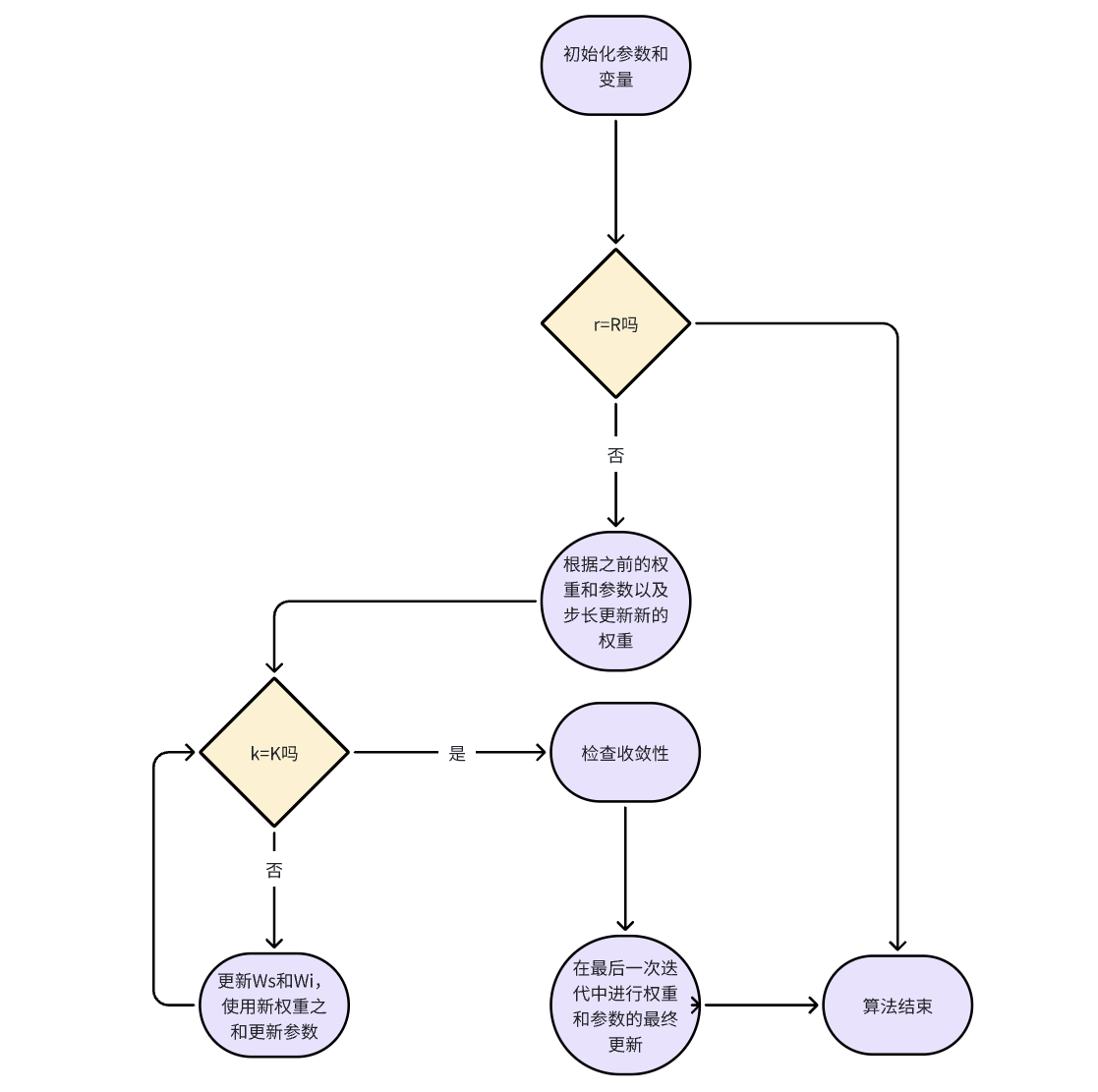


图1-7 面对攻击时Fast-PDMM Fed理论的安全聚合方案算法流程图

为了解决联邦学习在异构数据上模型精度低和通信效率低的问题,本算法利用了在中心化的联邦学习上使用具有次线性收敛率的分布式PDMM优化理论，并且借鉴了分布式PDMM理论的子空间扰动方法,提出了基于PDMM的联邦学习参数安全聚合方案。

本算法使用了增加客户端本地迭代次数的方法，提高了基于PDMM的联邦学习参数安全聚合方案的收敛速率,同时提出了对客户端本地损失函数进行二次近似的方法，大大降低了计算开销。我们把这种提高收敛速率和降低计算开销的方法称为Fast PDMM-Fed方案,并证明了Fast PDMM-Fed方案的收敛性和收敛率。

我们最后使用MNIST和Fashion MNIST数据集训练softmax回归模型,并得到了实际工作中的效果。从该方案在实际工作的效果中可以发现,Fast PDMM-Fed方案能在保护数据隐私的前提下依然有着较快的模型收敛速度和较高的模型精度。

图1-8 Fast-PDMM Fed理论的安全聚合方案算法流程图

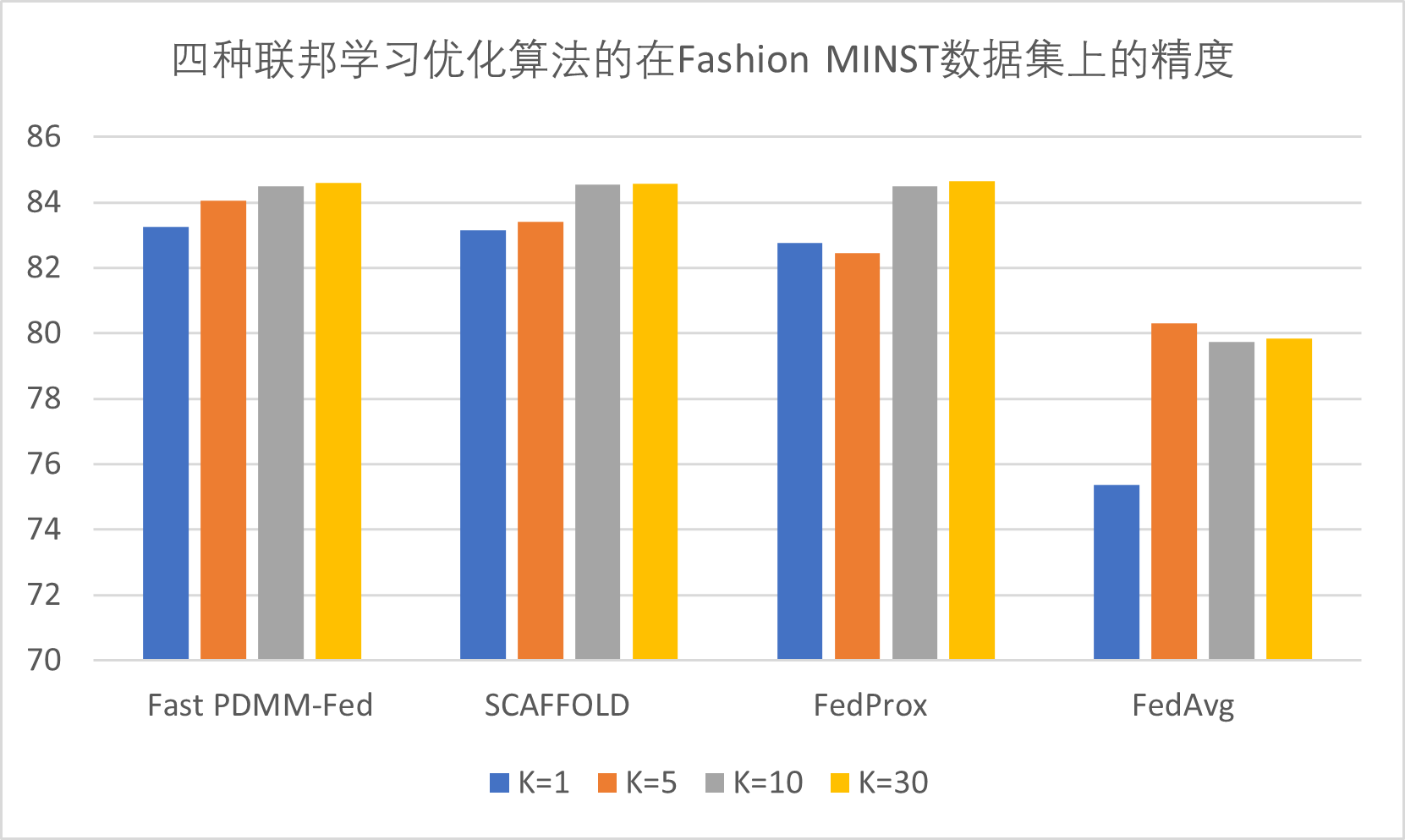


图1-9 四种联邦学习优化算法的在Fashion MINST数据集上的精度

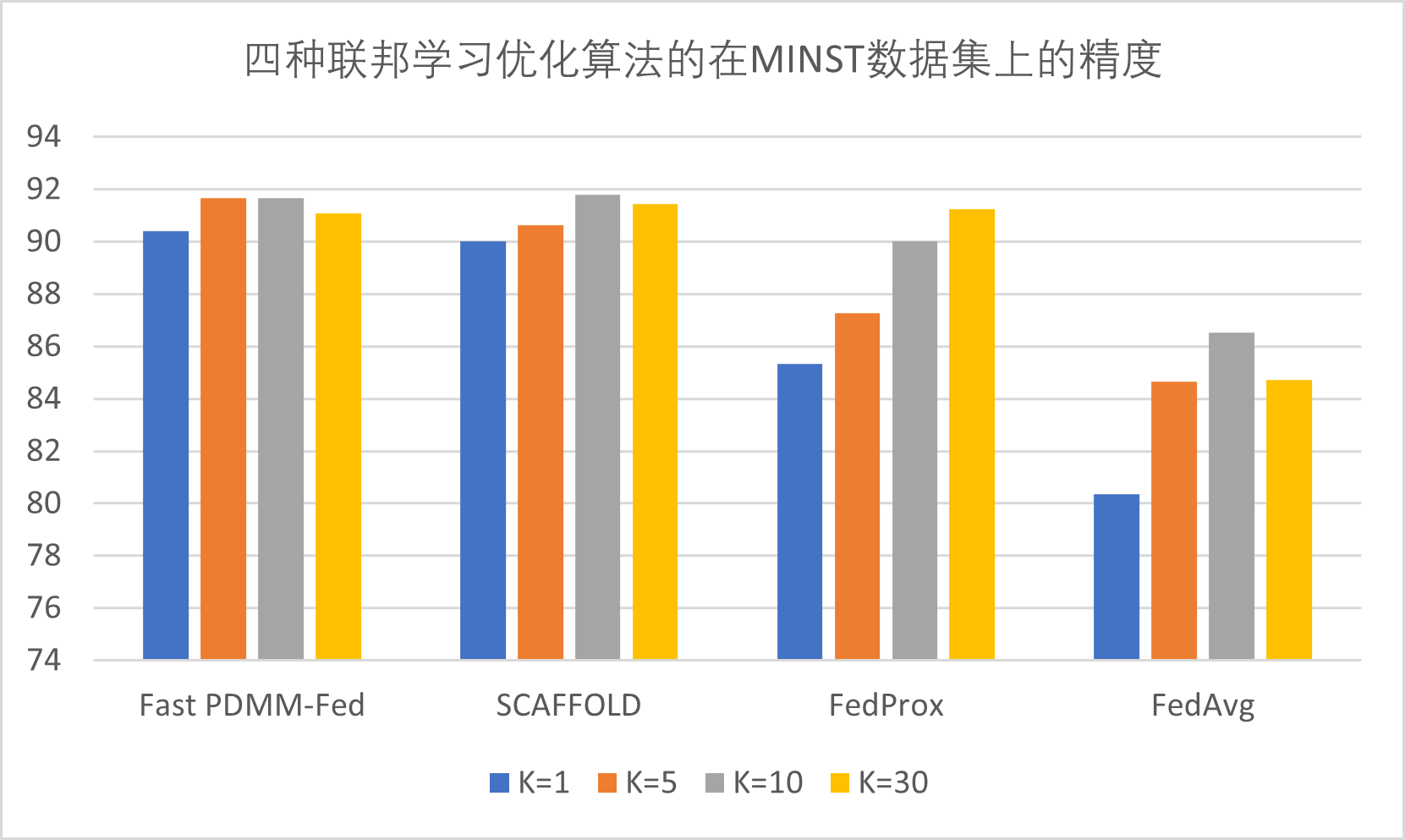


图1-10 四种联邦学习优化算法的在MINST数据集上的精度

**2)Re-Sim鲁棒聚合算法**

Re-Sim主要包含数据重采样和鲁棒聚合算法两部分。

数据重采样算法描述如下：假设有个设备，每个设备拥有的本地数据表示为其中第个设备采集到标签的样本数表示为。每个设备在统计本地数据后不再随机采样样本进行训练，而是依据公式来估计有效样本数，0,1)是超参数，并通过有效样本数的倒数设置采样权重。令设备本地数据中第个数据样本的标签为，根据数据中标签的数量来设置的采样权重

（1-15）

则选中第i个样本的概率表示为：

（1-16）

上述过程只在训练开始前在每个设备上计算每 个标签的采样概率，设备不需要将任何数据的类别分布信息发送给边缘服务器，充分保护数据隐私，同时数据重采样带来的计算开销可以忽略不计。

参考梯度选择：为了提高收敛速度，需要在每次迭代中找到最优的本地梯度，即参考梯度

（1-17）

梯度相似度评估：选取参考梯度后，需要选择特征信息来评估设备的本地梯度和参考梯度之间的相似度。由于模型更新的本质是由多维分量组成的梯度向量，考虑两方面特征信息：符号相似度梯度的符号决定了模型参数每个维度更新的方向。利用梯度符号相似度防御拜占庭攻击，符号相似度低的梯度有较高概率是拜占庭梯度，同时符号相似度可以减轻大方差噪声的负面影响，提升模型收敛速率。

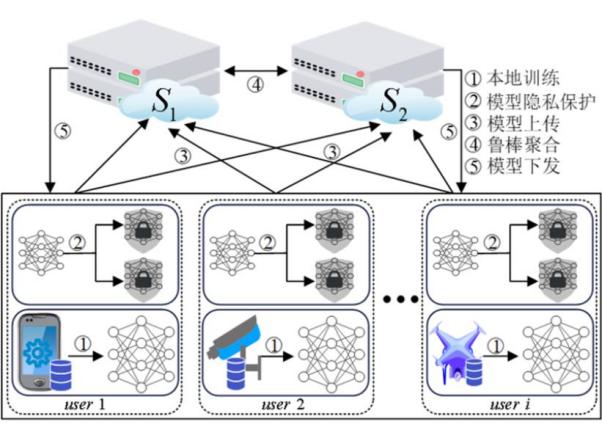


图1-11 基于Re-Sim算法的全局模型流程图

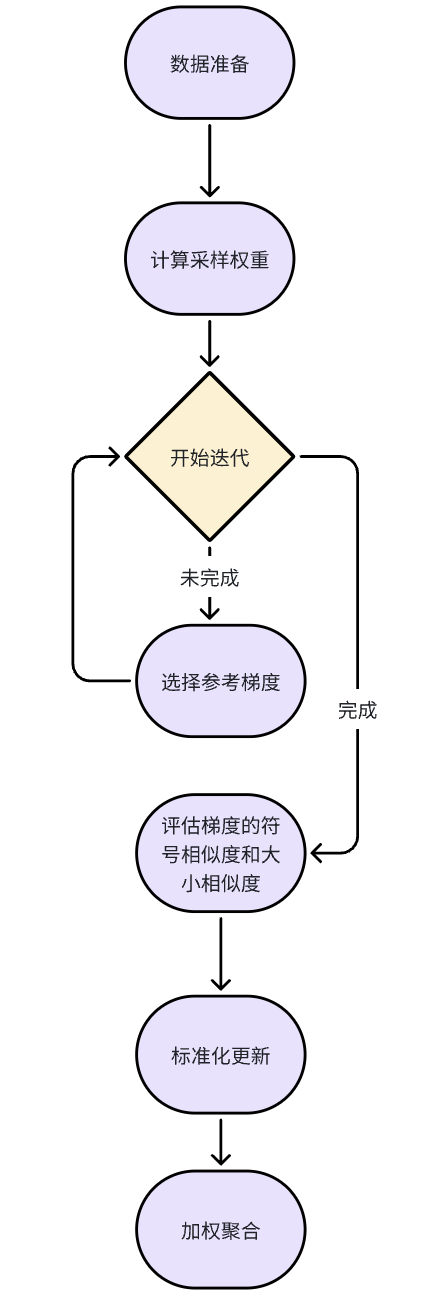


图1-12 Re-Sim流程图

具体来说，通过计算本地梯度和参考梯度在各个维度上对应符号相同分量的数量，并根据分量总数对结果进行标准化，得到相同符号分量的百分比。设本地更新的梯度向量为，计算符号相似度的公式为

（1-18）

其中，是符号函数；大小相似度：拜占庭攻击者为控制全局模型，其上传梯度与参考梯度的大小可能存在较大差异。使用大小相似度鉴别拜占庭梯度的过程如下， 首先计算本地梯度与参考梯度的大小相似度，其后依据大小相似度将每个本地模型更新进行标准化，均衡每个本地模型更新对全局模型更新的影响。具体公式为

（1-19）

是标准化后的本地模型更新。加权聚合：通过对标准化的本地梯度重新分配权重， 作为全局梯度

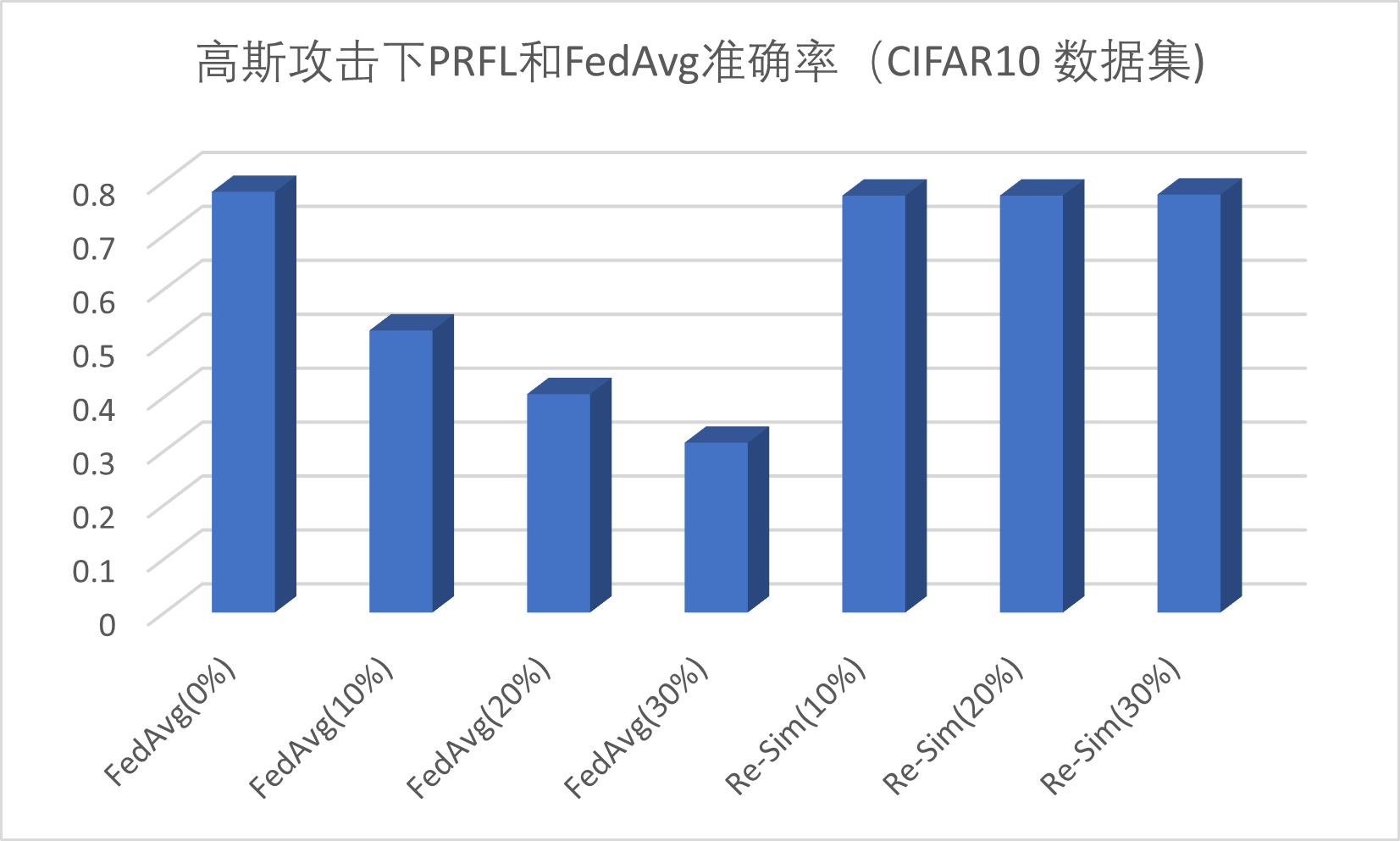
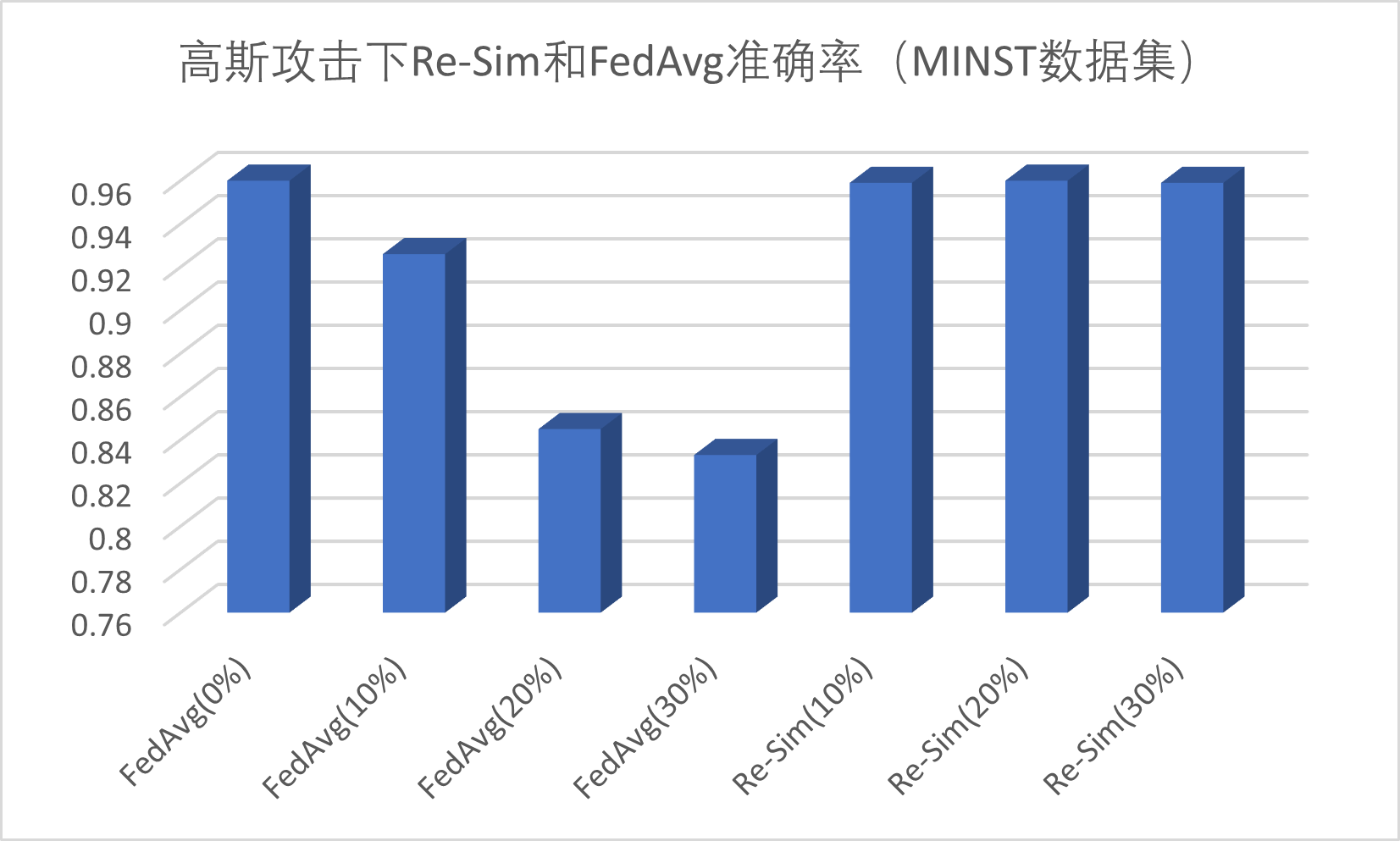
（1-20）

最后全局模型参数更新为：

（1-21）

鲁棒聚合方法是抵御拜占庭攻击的常用方法，但传统鲁棒聚合方法对于节点数据独立同分布( IID)的假设并不适用于本项目的场景。 受地理位置、用户偏好等因素影响，设备采集的数据往往是非独立同分布(Non-IID)的，传统鲁棒聚合方法难以区分独异于全局的梯度是拜占庭梯度，还是数据异构产生的次优化梯度。

该算法提出面向联邦学习的带有数据重采样的鲁棒聚合方法Re-Sim，通过测量方向相似性和标准化更新幅度实现模型的鲁棒聚合，并采用数据重采样技术增强数据异构环境下模型的鲁棒性。



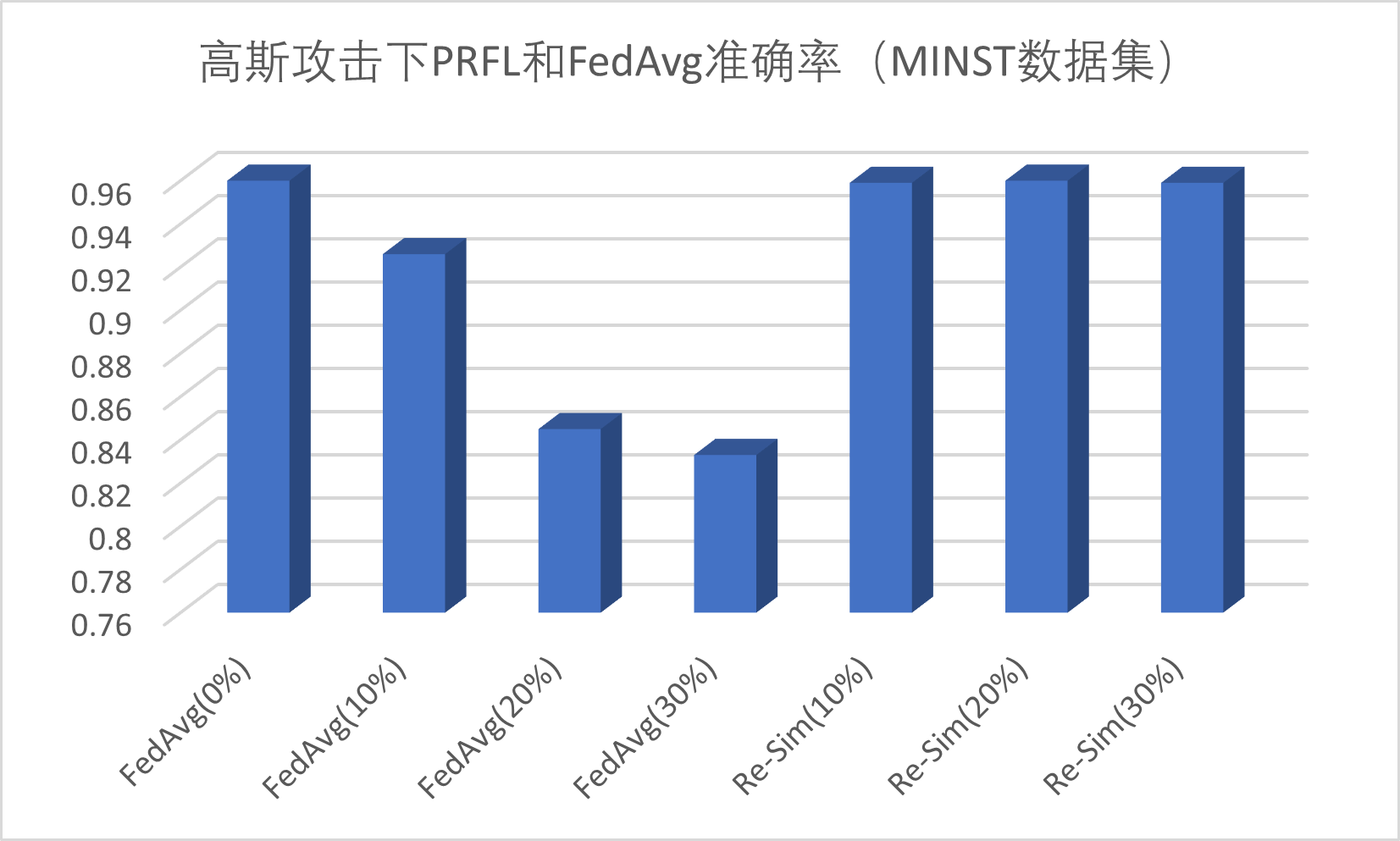
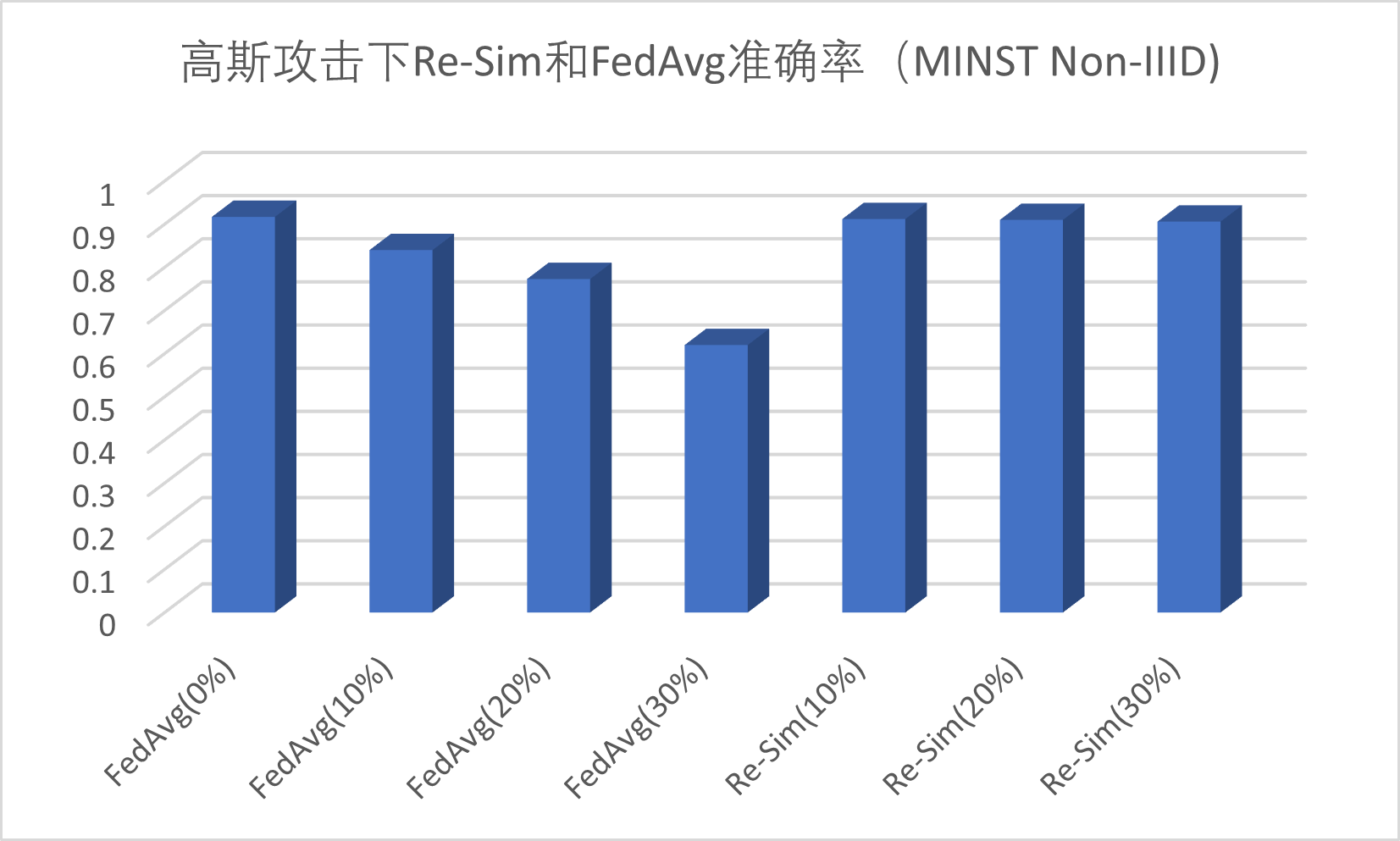


图1-13 FedAvg和Re-Sim的在面对攻击时的准确率

**3)基于改进的权重分配的联邦学习聚合方案**

针对全局模型参数，定义全局模型参数差异集，其中代表第个客户端的模型参数差异。通过算法核心步骤，进行模型参数的累积和更新。

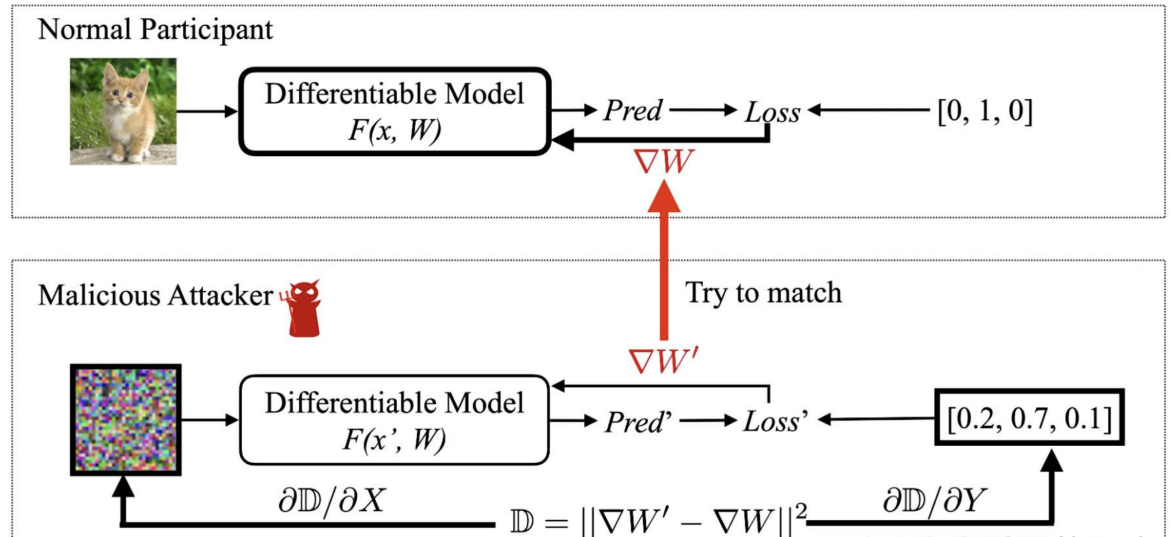


图1-14 面对攻击时联邦学习聚合算法的流程示意图

首先，定义加权因子，动量因子，学习率，权重衰减以及当前迭代次数。全局模型参数通过以下步骤更新：

实现数据初始化：动量和学习率初始化：初始化动量缓存和学习率缓存，对于每个参数，初始化为零向量和学习率。本地训练的过程：对于每个客户端，执行本地训练local\_train，获得模型差异和性能评分。先进行权重更新：更新权重累加器

（1-22）

梯度裁剪：计算累加后的平均梯度并应用梯度裁剪：

（1-23）

根据上述得到的结果进行动量更新：更新动量缓存M:

（1-24）

同时加入自适应学习率：

（1-25）

最后，学习率缓存（针对每个模型参数）按照以下方式更新：

（1-26）

更新参数：更新全局模型参数

直至满足收敛条件或完成预定的迭代次数。

通过上述步骤，此算法不仅关注性能评分高的客户端模型，也加入了动量和自适应学习率机制，以增强全局模型的稳定性和准确性。在不断迭代的过程中，模型能够逐渐适应各个客户端的数据特征，减少模型在特定数据集上的过拟合风险，并提高对整体数据分布的泛化能力。

此算法通过引入性能得分(performance\_score)作为权重，对每个客户端的模型更新进行加权。这种方法优先考虑那些对模型贡献更大的客户端，使得全局模型更偏向于学习这些客户端的数据分布，从而可能提高模型在多样化数据上的表现。同时为防止梯度爆炸，引入梯度裁剪机制来控制梯度的大小。梯度裁剪有助于稳定训练过程，特别是在处理非独立同分布（Non-IID）数据时，这些数据分布的差异可能导致梯度的大幅波动。最后学习率和动量的更新考虑了当前迭代的梯度信息，使得每一步的更新都是基于最新的模型状态，能够更灵敏地调整学习策略。

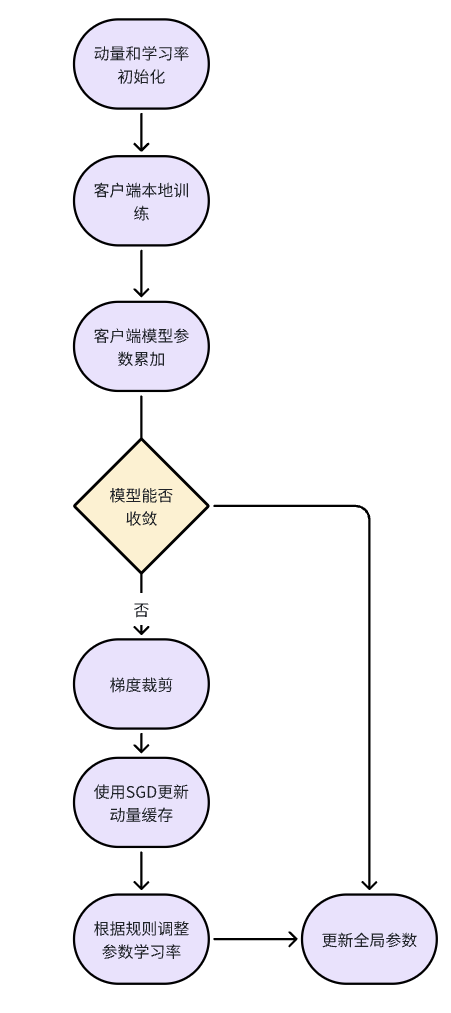


图1-15 Alterfed 算法的具体流程图

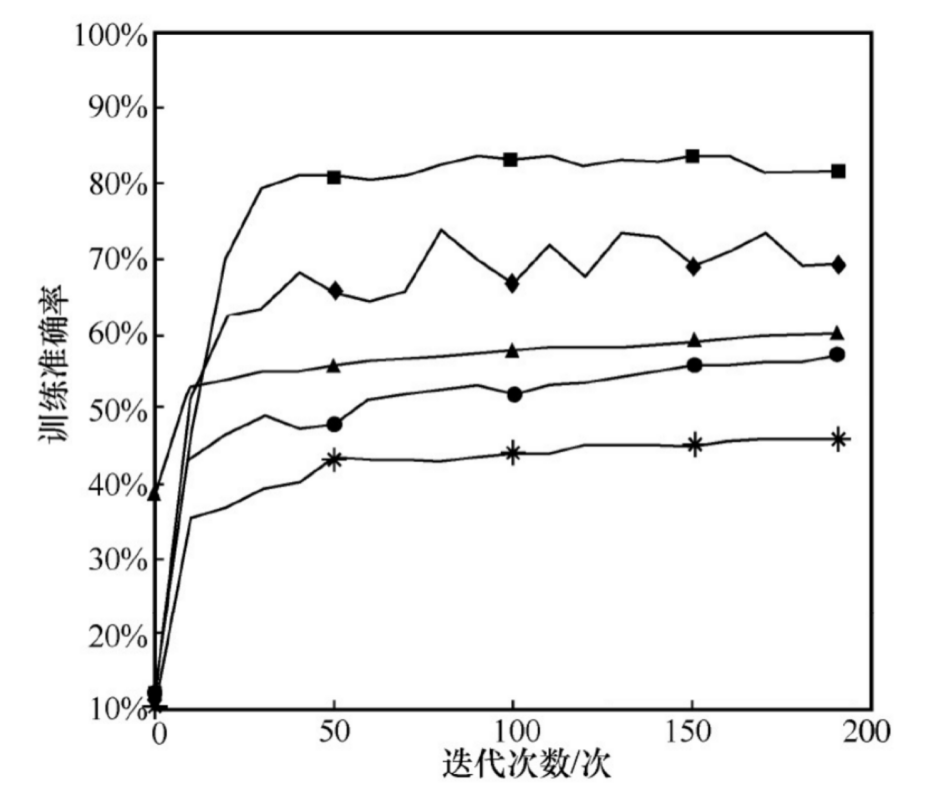


图1-16 Alterfed、Krum、RSA、ByrdSAGA、Zeno算法在模型训练时的准确率

**4)基于中值归一化余弦相似性的联邦学习聚合算法**

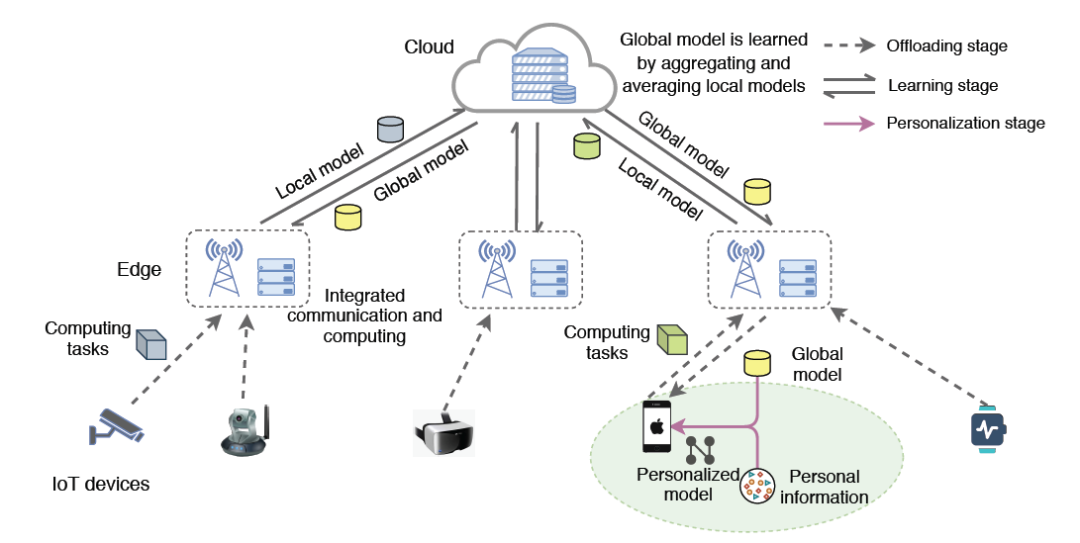


图1-17 联邦学习全局聚合算法示意图

首先进行数据的初始化：定义余弦相似度函数，其中a和b 是向量，·表示点乘，是向量的范数

为每个客户端c初始化梯度存储和梯度量存储。接下来进行本地训练与梯度聚合：对每个客户端 c，计算模型更新差异 ，将 中的所有梯度重新塑形为一维向量并串联，得到，如果则和，计算中值向量：对于所有，对每一维度i计算中值，构成中值向量M，其中 的计算如下：

（1-27）

计算余弦相似度：将全局模型参数通过M转换为中值梯度。对于每个和，计算余弦相似度:，之后进行相似度调整：

（1-28）

最后进行权重调整与全局模型更新：为每个客户端计算权重

（1-29）

之后再应用加权更新到全局模型：

（1-30）

更新全局模型:

（1-31）

这样的处理方式确保了算法在考虑每个客户端的模型更新时不仅仅关注更新的大小和方向，还通过中值向量和余弦相似度的引入来提高聚合过程的稳定性和效率。

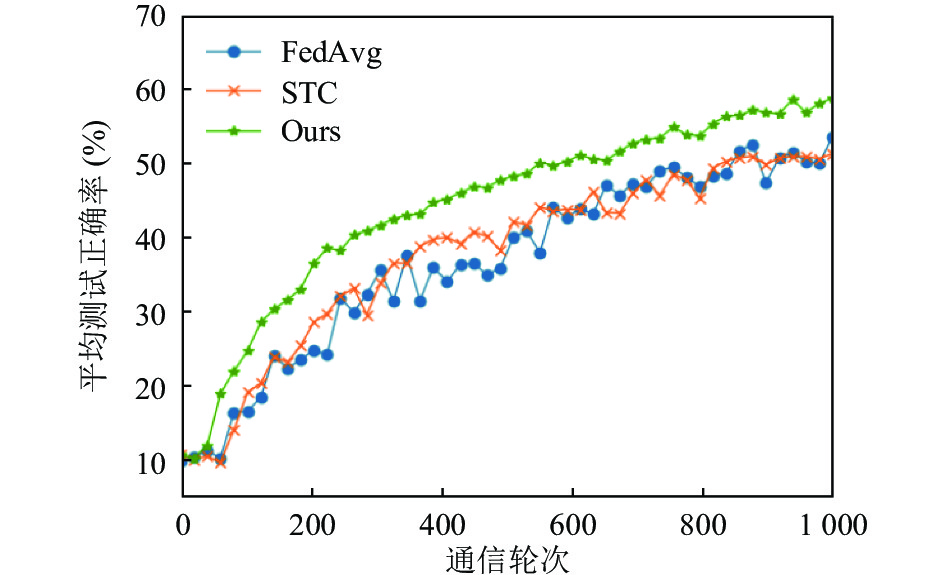


图1-18 FedAvg、STC、Median-Cos算法在测试集上的正确率对比示意图

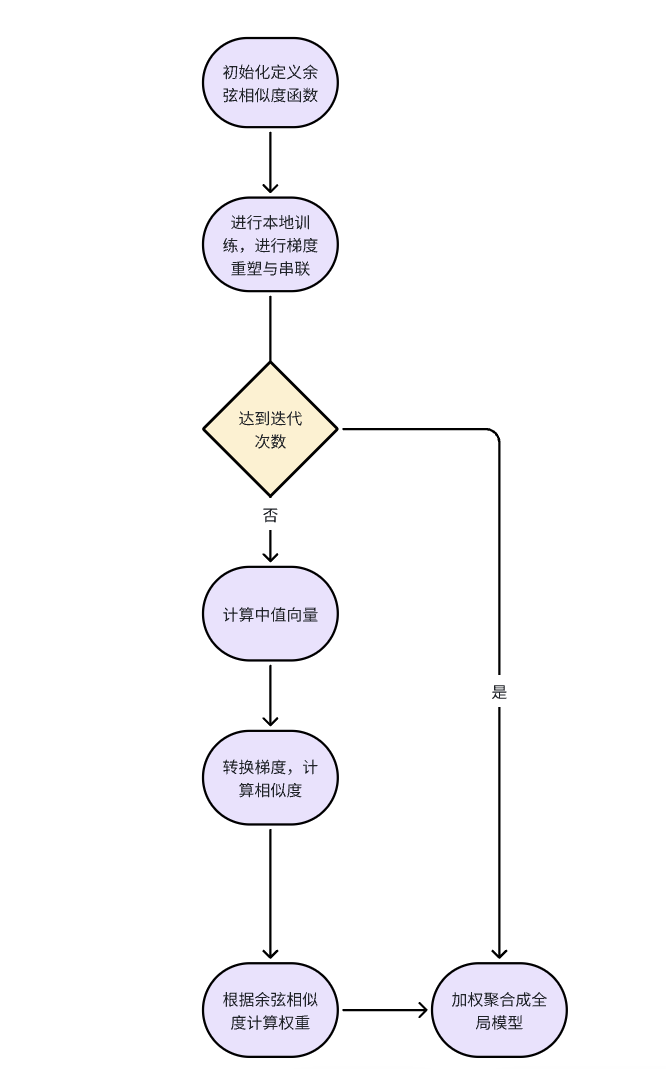


图1-19 Median-Cos算法流程示意图

**1.3.1.3 通用联邦学习算法实现**

**Fedavg算法**

聚合的方式通常是对所有模型参数取平均值，这就是“FedAvg”名称的由来。即使参与方的数据分布不同，平均化的过程也有助于获得一个更广泛适用的模型。

（1-32）

**Krum算法**

异常模型检测（Outlier Model Detection）：在中央服务器上，首先计算每个模型参数的差异度，然后根据这些差异度来确定哪些模型被认为是异常的。通过计算每个模型的余弦相似度来完成。排除异常模型后，剩余模型的参数将被聚合以形成全局模型。

**Trimmed-mean算法**

在中央服务器上，对接收到的模型参数进行修剪。这可以通过移除每个参与方发送的模型参数中的一定比例的最高值和最低值来实现。

**Median算法**

对合并的模型参数集合进行排序，并找到其中的中间值。如果模型参数个数为偶数，则取中间两个值的平均值作为中值。中值作为全局模型参数。

**Perfl算法**

对于每个参与方，将梯度转换为向量，并将这些向量存储在 grad\_vector 中。然后，通过对所有参与方的梯度向量的中位数进行计算，得到一个全局的中位数向median\_vector。计算每个参与方的梯度向量与全局中位数向量的皮尔逊相关系数（Pearson correlation coefficient），以衡量它们之间的相似度。从而确定参与方的贡献程度。根据皮尔逊相关系数，计算每个参与方的权重 mu，用以调整其梯度的贡献,根据每个参与方的权重，将其调整后的梯度加权累加到 weight\_accumulator 中，并将结果用于更新全局模型。

**Sheildfl算法**

对于每个参与方（candidates），首先通过本地训练数据与全局模型之间的差异来获取梯度。为了消除向量长度对余弦相似度计算的影响，将每个参与方的梯度向量进行归一化处理，以确保它们具有单位范数（即向量长度为1）。计算每个参与方梯度向量与全局梯度向量之间的余弦相似度。找到余弦相似度最小的梯度向量作为基准梯度（baseline\_grad）。根据每个参与方的梯度向量与基准梯度向量的余弦相似度，计算权重系数 mu。据此聚合全局模型。

1.3.2 通信传输

通信传输的算法设计涉及数据在节点之间传输的效率和稳定性。在本节中，我们将介绍用于优化数据传输过程的算法，如压缩技术、编码策略和调度协议。这些算法解决了Fed Privacy Shield平台中数据传输量大、带宽限制和延迟敏感等挑战，确保了数据在各计算节点和中心服务器之间可靠且高效地传输。

#### 1.3.2.1安全与认证

在当今数据驱动的时代，数据安全和隐私保护是构建任何技术解决方案的核心要素。特别是在面向联邦学习数据流交互任务的隐私计算平台中，这一点尤为重要。我们的方案着眼于解决现存的安全挑战，采纳先进的技术标准，以确保在数据交互过程中的安全性和隐私性，支撑联邦学习的广泛应用。

**1)SSL/TLS**

为确保平台访问的安全性与数据传输的加密，本平台采纳了基于SSL的HTTPS协议作为通信标准。通过在Nginx配置文件中部署腾讯云申请的SSL证书和密钥，强制用户通过HTTPS访问，从而加密客户端与服务器之间的通信。这种做法有效防止了敏感信息如个人身份数据等在互联网传输过程中被第三方窃听或截获。gRPC框架默认使用HTTP/2 作为传输协议，为了保证数据传输的安全性，我们使用证书颁发机构（CA）颁发签名的证书，实现了gRPC的所有客户端与服务端基于SSL/TLS进行安全通信。

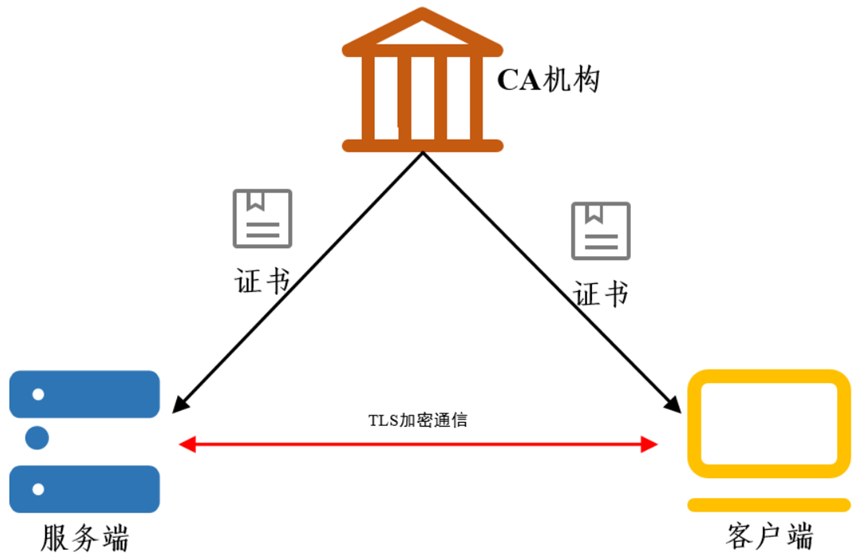


图1-20 TLS/SSL安全通信

**2)双向认证与JWT鉴权**

为gRPC的客户端和服务端分别生成证书和密钥，并在配置过程中启用双向认证，这构建了一个可信的通信环境，并保障了数据的保密性和完整性。同时我们的平台设置了全链路的JWT鉴权，在HTTP框架Gin中加入了鉴权中间件，在gRPC的客户端与服务端分别设置拦截器，从前端访问后端接口开始，再到后端调用联邦学习相关微服务，都需要携带合法有效的Token。此外，用于联邦学习的Server服务、Client服务以及Agent服务在互相通信前也需要JWT鉴权，最大程度防止了数据被窃听、篡改或伪造。链路中每有一层服务的调用，就会有一次Token的刷新，每次HTTPS请求结束后，也会返回给前端刷新过后的Token,保证了Token的可用性。

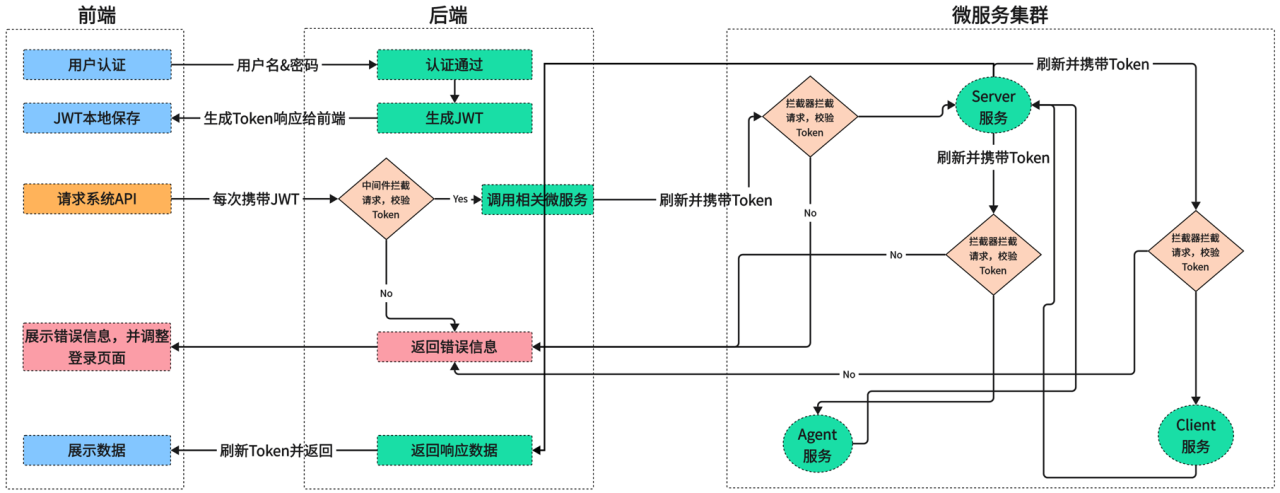


图1-21 全链路JWT鉴权

**3)敏感信息加密**

为确保用户敏感信息的最高级别安全，所有相关数据在存储前都通过对称密钥加密算法AES加密处理。当需要访问并返回敏感数据时，只有通过唯一密钥才能解密，这保证了数据的安全性和隐私性。

#### 1.3.2.2流式传输机制

流式传输机制通过将文件分割成多个小块，并利用流式RPC传输至服务端或客户端，有效减少了内存资源的消耗，同时提高了数据传输的效率。无论是后端与联邦学习微服务之间，还是联邦学习内部各服务（如Server、Client）之间的通信，均采用双向流式文件传输机制，实现了数据处理的高效性。

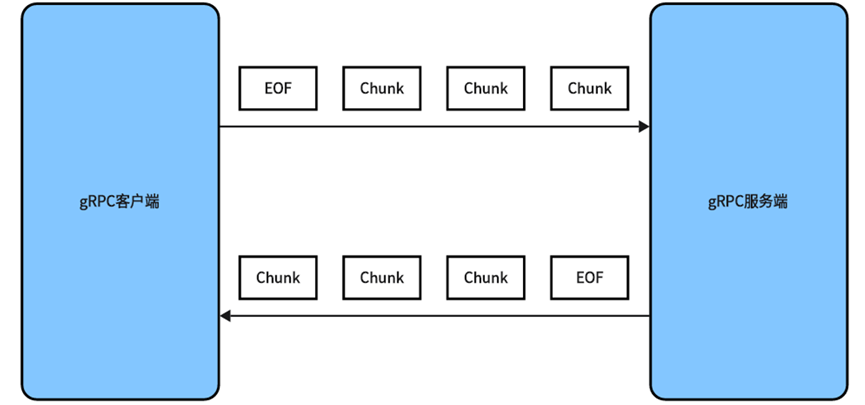


图1-22 流式传输

#### 1.3.2.3多层负载均衡

前端访问后端应用时，会先通过Nginx，这里设置了第一层负载均衡，调用具体的微服务（比如Server）前会通过Consul进行服务发现，所有服务发现的过程均使用了轮询的负载均衡算法， 后端调用Server微服务时设置了第二层负载均衡，Sever微服务再调用Agent微服务，这里有第三层负载均衡，随后Server微服务再调用Client微服务，出现第四层负载均衡，通过多层负载均衡的设计，大大提高了隐私计算平台的稳定性和容错性，使得整个平台能够抗住高并发。

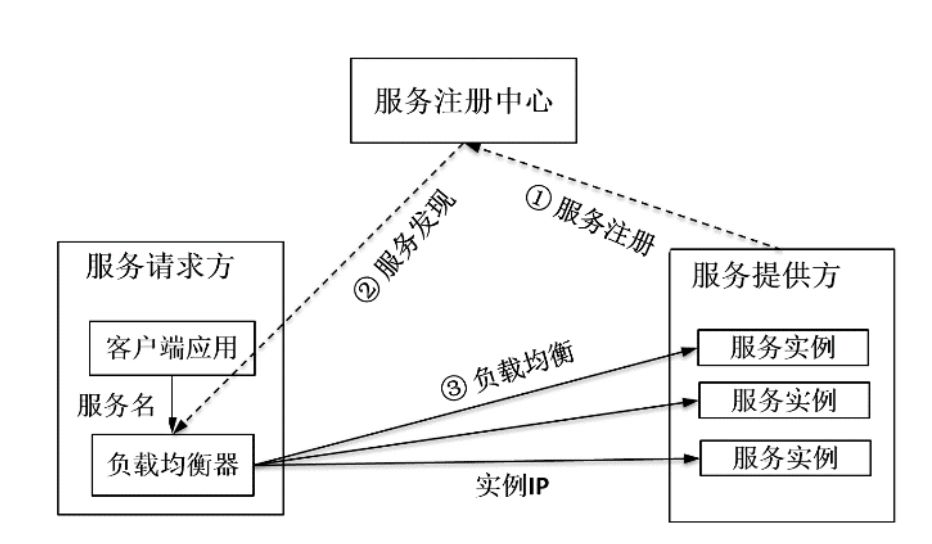


图1-23 服务发现与负载均衡

### 1.3.3 安全加密

安全加密部分关注的是确保在整个联邦学习过程中的数据安全和用户隐私。我们将探讨在平台中实现的多种安全机制，这些算法和技术的设计目标是在不牺牲计算效率的前提下，最大化地保护用户数据的隐私和模型的安全性，防止信息泄露和恶意攻击。

**1.3.3.1 差分隐私**

差分隐私的思想是当敌手试图从数据库中查询个体信息时，尽量少地降低数据查询的准确性，同时最大限度减少识别个体信息记录的机会。

**1.差分隐私简介**

**差分隐私原理**

记数据集为D，由所有D的子集组成的训练数据库记为，参数空间为。我们称一种训练机制满足（ϵ,δ)−DP，如果对于任意两个毗邻的训练集d,d'∈D, 以及任何参数范围S⊂T，其输出的参数分布满足：

 （1-33）

ϵ是隐私预算，ϵ越小，隐私保护等级越高，噪声越多，数据可用度越低，δ是松弛因子。在该定义下，数据集中的单个数据对于模型的影响被控制在一定的范围内，实现了在模型层面对于差分攻击的“不可分辨性”。

为了控制个体数据的影响，利用高斯机制产生噪声对梯度施加差分隐私。假设存在一个确定函数f:D→T，敏感度为

Δ2(f)=max||f(d)−f(d')||2，d,d'∈D （1-34）

那么对于任意的δ∈(0,1)，给定随机噪声服从正态分布N(0,σ2)，那么随机算法服从(ϵ,δ)−DP，其中

 （1-35）

利用高斯机制，我们对梯度分三个步骤增加差分噪声：首先，对每一个样本对应的梯度裁剪到一个固定范围[−C,C]，以控制个体数据的影响，此时梯度的敏感度

 （1-36）

然后，对裁剪后的梯度增加高斯噪声N(0,σ2)，以得到满足差分隐私的梯度数据。最后，用这些梯度更新模型，并计算模型的隐私损失。记噪声乘子为z=σ/C，那么该训练平台服从(ϵ,δ)−DP的条件为

​ （1-37）

**差分隐私效果**

**防范梯度泄露**联邦学习通过交换梯度信息来利用分布式数据，训练数据完全驻留在客户本地。研究表明，训练数据可以从交换的梯度信息中重建。这种攻击方式被称为梯度泄露攻击。使用差分隐私对梯度进行保护，攻击者即使获得一个人的梯度信息，依旧无法重建训练数据信息。

**防范成员推理** 使用差分隐私能够使攻击者即使有成员数据也得不到相同的训练结果，无法分辨出某些数据样本是否被用于学习模型的训练过程，降低了攻击者进行成员推理攻击的成功率。

**防范数据投毒** 使用差分隐私进行保护时，可以设置梯度范式上限，通过计算梯度范式，对梯度进行缩放，以限制梯度的大小不超过指定的阈值，防止数据投毒等原因造成的模型不收敛。

**2.差分隐私算法设计与实现**

1)本地化进行差分隐私

在联邦学习过程中，端节点需要将训练结果发送给云服务器，当端节点不信任云服务器时，通过采取本地差分隐私会解决中心化差分隐私的缺陷并确保隐私。在这个过程中，在与云服务器共享数据之前，对每个客户端的数据应用了差分私有更改这允许在没有中央服务器参与的情况下实现DP。由于每一方都独立修改其数据，使隐私得到了保护。

2)本地化差分隐私作用于梯度

深度学习的训练与推断流程通常由四步组成：构建深度模型并初始化模型参数；输入训练数据并计算梯度；用梯度下降法更新模型直到收敛；将收敛的模型用于数据预测。

最直接的隐私保护方法是对模型预测直接添加噪声，由于深度学习的模型参数非常容易获取，因此仅仅在预测阶段添加噪声往往无法达到隐私保护的目的。在数据集上直接增加噪声往往会使得数据集本身变得脏而不可用。另一个想法是在训练好的模型参数上直接添加噪声，但是，深度神经网络对参数非常敏感，直接对训练好的参数添加高斯噪声会使得整个算法完全失效，因此差分隐私机制需要在端节点训练过程中加入。

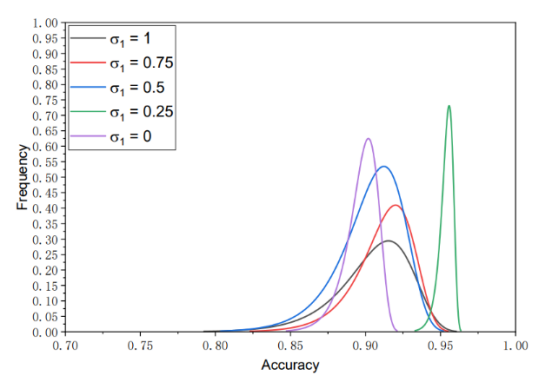


图1-24 本地化差分隐私大小对准确度影响

3)中心化差分隐私

当云服务器聚合来自不同端节点的模型更新时，我们加入噪声来削弱不同端节点对模型的贡献。来自不同端节点的模型更新为，对于每个来自不同端节点的更新，计算其范式，得到总范式，Agent计算灵敏度为的平均数，最后更新全局模型

（1-38）

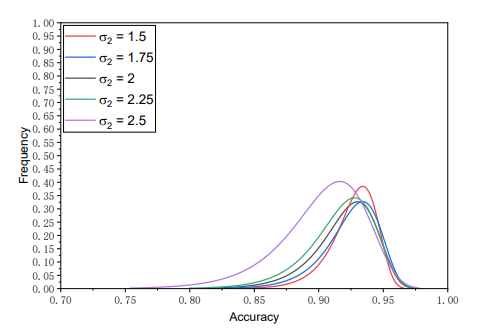


图1-25 中心化差分隐私大小对准确度影响

**3.差分隐私自适应性**

由于采用本地化差分隐私，差分隐私噪声聚合后强度可能过高，需要采取适当方法调整差分隐私噪声强度，在较少地降低隐私保护强度的同时提高模型的精度。

该自适应性由两方面提供：

1)调整差分隐私噪声强度：使用差分隐私优化模型训练时，原本所有用户采取了固定的差分隐私噪声强度。通过比较全局模型更新与本地梯度差异，当全局模型与本地梯度差异较小时，本地梯度需要更加精确从而为全局模型提供更准确的更新，故选取相对较小的差分隐私噪声强度。并且也为自适应提供了限度，既能维持小噪声防止模型过拟合和隐私保护水平低，也能避免噪声过大影响模型精度。

2)调整梯度范式大小：差分隐私优化的随机梯度下降将梯度裁剪阈值C作为输入，C的选择对优化器的性能影响较大。如果C太小，会导致梯度的过度截断，造成聚合缓慢；如果C太大，不必要的噪声将被添加到梯度更新，会导致生成梯度质量不好。自适应选择阈值，C令其约等于输入的样例x的梯度范式，这样保证了输入x对平均梯度的贡献具有最小相关误差的最大保护。

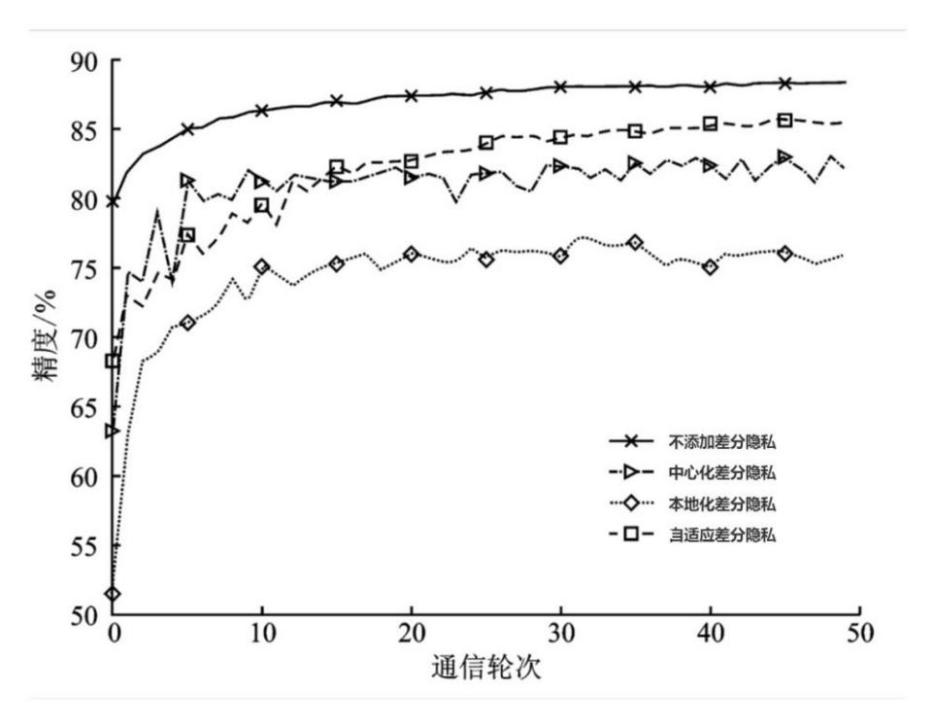


图1-26 添加自适应差分隐私对精度影响图

**1.3.3.2同态加密**

一般的加密方案关注的都是数据存储安全。加密后的数据没有密钥的用户，不可能从加密结果中得到有关原始数据的任何信息。只有拥有密钥的用户才能够正确解密，得到原始的内容。此过程中用户除了解密、存储和传输无法对加密结果做任何操作的。

同态加密提供了对加密数据进行处理的功能。不具有私钥的用户可以对加密数据进行处理，且处理过程不会泄露任何原始内容。同时，拥有私钥的用户对处理过的数据进行解密后，得到正确的处理后的结果。

**1.同态加密简介**

**同态加密原理**

**由于进行聚合时需要计算梯度范式等数据，同态加密采用ckks全同态加密方案。**

编码与加密：在CKKS方案中，首先将实数或复数数据编码成多项式，然后对这些多项式进行加密。

同态操作：加密后的数据可以进行同态加法和同态乘法操作。在这个过程中，对加密数据的算术操作等价于对原始数据的相应算术操作。

解密与解码：计算完成后，可以使用私钥对加密的结果进行解密，得到包含计算结果的多项式。然后，这个多项式被解码回原始的浮点数或复数结果。

近似算法：CKKS方案是基于近似算法的，这意味着它允许在加密数据上进行近似算术操作，便于处理实数与复数运算。

**同态加密效果**

**1）.防止云服务器获取训练端梯度结果：**

通过使用同态加密技术，在联邦学习中可以确保不可信的云服务器无法获取每个训练端的训练梯度结果。模型参数在上传到云服务器之前可以进行加密处理，使得云服务器只能在加密状态下进行计算和聚合，无法获得原始梯度信息。

**2.同态加密算法设计与实现**

同态加密整体流程：

1. 模型发布：CS生成ckks公私钥对，发送模型与公钥给Cs。

2.本地培训：每Cs使用本地数据集在自己的设备上本地训练模型。然后Cs将更新梯度加密上传到CS。

3.模型聚合：CS选择聚合方案，聚合有效的模型参数，得到全局加密模型。

4.模型分析：CS对模型参数进行解密并决定是否进行再差分或直接更新模型或结束任务。

5.更新：CS发布更新后的全局模型给Cs，Cs并重复上述步骤，直到任务完成。

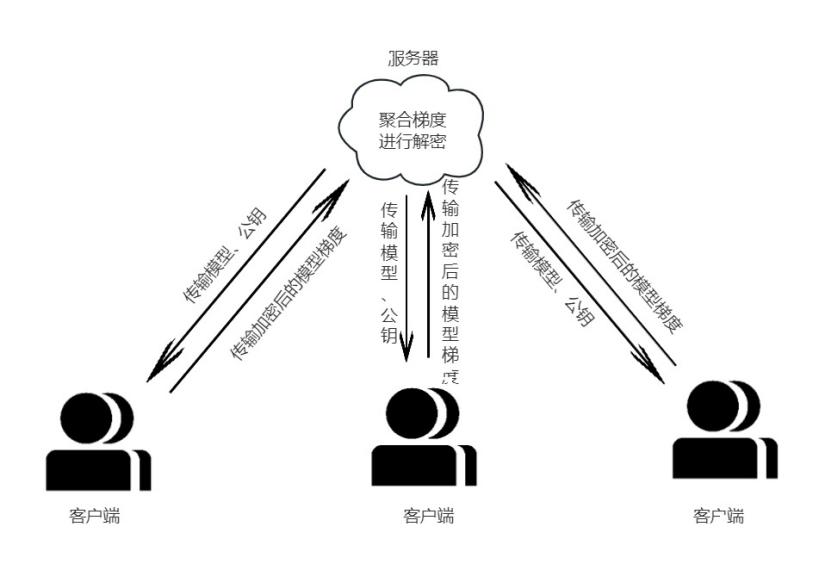


图1-27 同态加密基本数据传输

**3.选择性提交梯度更新**

同态加密将密文替代模型梯度参数进行传输，会大幅增加模型传输与聚合时的通信与计算开销，可以通过对梯度更新进行裁剪，选择性提交梯度更新来实现模型的压缩和精简, 减少通信和计算开销。

在差分隐私时，通过差分隐私产生的影响计算得到梯度数据的隐私度，对于隐私度较低的数据进行梯度压缩。

**4.引入agent半可信端**

将同态加密整体流程修改为：

1)初始化：Agent一个生成同态加密密钥对并将公钥发布到有效的Cs,CS。

2)本地培训：每Cs使用本地数据集在自己的设备上本地训练模型。然后Cs将更新梯度加密上传到CS.

3)模型聚合：CS聚合有效的模型参数，得到全局加密模型。然后，CS将模型发送到Agent.

4)模型分析：这Agent解密此全局模型以获得当前的全局模型e。然后Agent分析e并决定是否将相应阶段的全局梯度更新进行再差分或直接发布给CS或结束任务。

5)更新：CS发布更新后的全局模型给Cs，Cs根据CS发布的全局模型执行下一轮局部训练，然后将更新后的模型参数上传到CS，并重复上述步骤，直到任务完成。

**由于CS可能无法直接对同态加密后的密文进行加权聚合等，为了适配不同的聚合方案，根据聚合方案的不同，CS需要额外计算梯度的范式等相关数据，一并交给Agent，Agent完成加权聚合以及决定是否进行再差分等操作。**

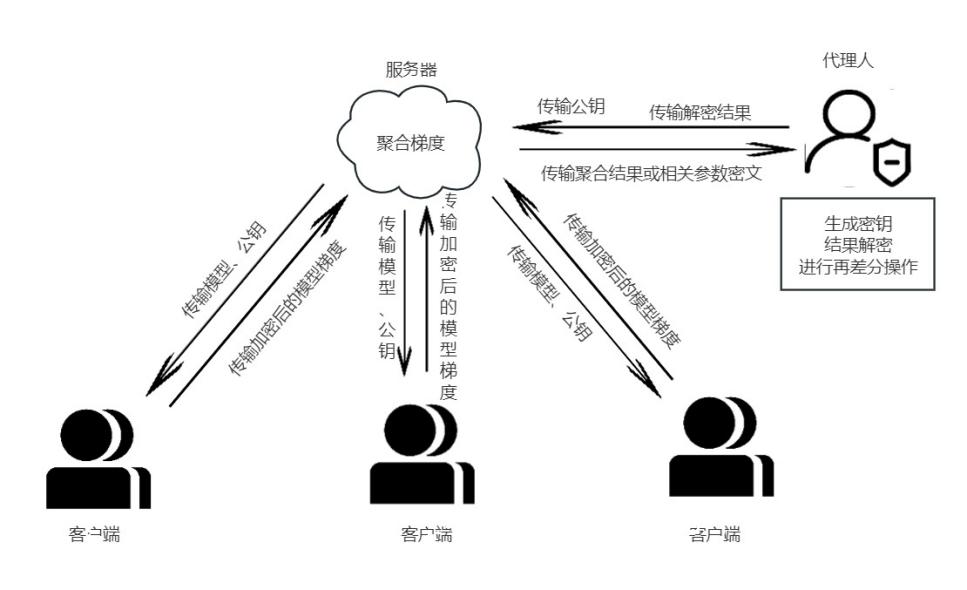


图1-28 添加证人Agent后的数据传输

选择将密钥移交给Agent半可信端，可以：

1. **隐私保护**：1.避免云服务器依次解密每位训练端传递的梯度，从而泄露敏感信息的风险。2.避免云服务器与训练端合谋。以及避免训练端的功能不对等。3.避免了扩大通信开销，减轻训练端计算压力。

**2）决策与控制**：Agent作为半可信端能对全局模型更新进行决策，可以在判断是否需要进行再差分等操作，为平台提供进一步保护。

**1.3.3.3 使用掩码方式安全聚合**

同态加密将梯度中数字进行加密会导致原本较短的数字长度暴增，为计算与通信带来较大压力。为防止计算与通信压力过大，给客户端带来影响，我们还设计了基于多方安全计算的聚合方案。

但由于使用掩码方式进行安全聚合需要提前商议聚合权重进行聚合，对聚合方案限制较大，现在只能支持平均聚合以及固定权重聚合。

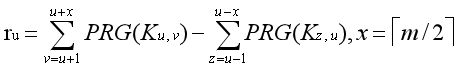
**1.基于多方安全计算的安全聚合简介**

客户端训练结束后，服务端将进行安全聚合。在此过程中，安全保护的目标是保护每个用户的本地梯度，服务器只能拿到用户梯度总和，无法破解单个用户模型梯度，且不影响模型聚合准确性。

基于多方安全计算的安全聚合，需要客户端进行协商，每个用户使用一个随机梯度对自己的梯度进行掩盖，并且需要保证所有随机梯度加权之和为零。再将掩盖后的梯度发给服务端进行聚合，既能保护单个客户端模型梯度隐私，也能得到准确聚合结果。

**2.算法设计与实现**

采用一种链式零共享算法，客户端按ID大小构建首尾相连的链式结构。假设用户总数为m，用户u生成m/2个随机种子Kuv，并将Kuv 分别发给链式结构中位于自己后方的m/2个用户v。同时，接收链式结构中位于自己前方的m/2个用户z分别发送的随机种子Kzu。此后，用户u根据式生成随机梯度ru。

 （1-39）

PRG表示伪随机构造器，输入相同的随机种子将得到同样的随机结果。

客户端u计算出ru后，模型梯度可以被隐蔽为：IMG_257。服务端得到所有用户上传的掩码模型梯度，则根据式子

IMG_258 （1-40）

可计算出用户梯度之和。即边缘服务器在不知道用户真实模型梯度的情况下实现了模型聚合，进而保护了用户隐私。

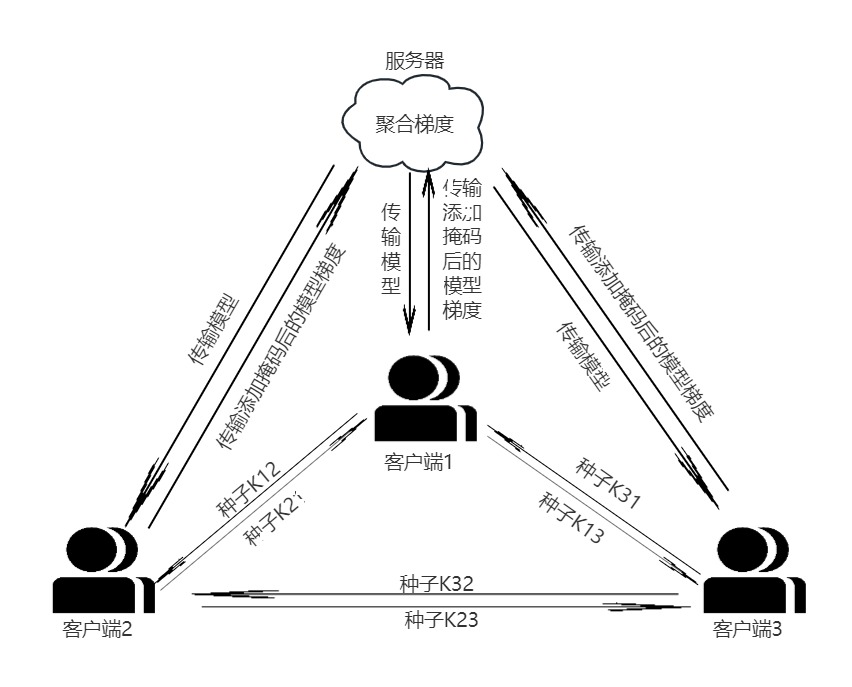


图1-29 使用安全聚合数据传输

如图所示，客户端1的模型梯度可以隐蔽为：

 （1-41）

**3.秘密共享算法应对用户掉线**

基于多方安全计算的安全聚合方案，如果出现用户掉线的情况，会使随机梯度和不为零，使聚合结果错误。为避免这种情况，可以使用秘密共享算法将生成的随机种子的种子或随机梯度共享给其他用户，当用户掉线时，服务器可以向其他用户询问秘密切片，从而重构出掉线用户的种子或随机梯度，从而消除聚合结果误差。

拉格朗日秘密共享算法:

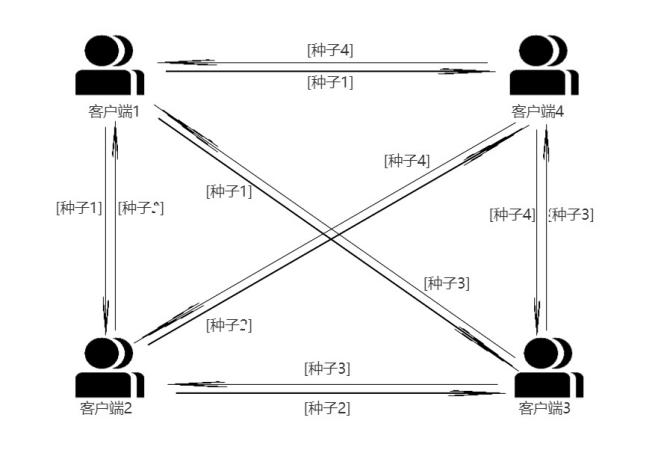


图1-30 秘密共享数据传输

如图，[种子1]代表被分割的秘密。

使用拉格朗日插值法进行秘密分享，已知给定一个多项式：

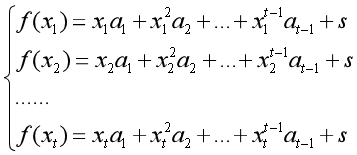
IMG_259 （1-42）

已知任意k个或更多的点(x,f(x))可以解出f(x)的表达式

已知小于k个点(x,f(x))将无法得到f(x)的表达式

因此，对于阈值t，用户可以使用密文s来构造一个t-1阶多项式f(x)，然后选取n对(x, f(x))，并分发出去，将t个点组合起来可以解出多项式的系数，常数项即是密文。

所以当用户掉线时，服务器会向其他用户请求秘密切片，求解方程组

 （1-43）

解出密文s，服务器便能消除误差。

## 1.4数据库设计

本节概述Fed Privacy Shield平台的数据库设计。

平台采用MySQL关系型数据库存储数据。共包含十张表：用户表(user)、边缘服务器表(edge)、证人表(agent)、设备表(device)、模型表(model)、数据集表(dataset)、部署信息表(deployment)、容器表(container)、训练任务表(train)、监测任务表(monitor)。

**通用字段**

为了方便对业务数据的状态进行跟踪，在每个数据库表中都包含了三个相同的字段：创建时间、更新时间和删除时间。通过在数据库设计中统一包含这四个字段，可以提高数据管理的效率和一致性，同时也为应用程序的数据处理和状态跟踪提供了便利。设置deleted\_at字段的目的是方便实现软删除，可以帮助用户找回丢失的训练记录、模型信息等其他重要数据。三个公共字段的详细设计如下所示：

表1-19 数据库通用字段表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **列名** | **类型** | **是否必须** | **备注** |
| 创建时间 | created\_at | datetime | 是 |  |
| 更新时间 | updated\_at | datetime | 是 |  |
| 删除时间 | deleted\_at | datetime | 是 | 索引 |

**用户表(user)**

用户表是设计用来存储平台中所有用户相关信息的数据库表。它包括用户的基本信息，如用户ID、用户名、密码、电话号码和头像，以及用户的角色。每个用户都有一个唯一的用户ID作为主键，手机号设置索引以便快速查找，用户的头像存储为一个链接地址。用户的角色定义了用户的权限级别，比如普通用户或管理员，这对于平台安全和功能访问控制至关重要。

表1-20 数据库用户表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **列名** | **类型** | **是否必须** | **备注** |
| 用户id | id | bigint | 是 | 主键 |
| 用户名 | username | varchar(20) | 是 |  |
| 密码 | password | varchar(256) | 是 |  |
| 手机号 | mobile | char(11) | 是 | 索引 |
| 头像 | avatar | varchar(256) | 是 |  |
| 角色 | role | int | 是 | 1-用户，2-管理员 |

**边缘服务器表(edge)**

边缘服务器表用于记录平台中所有边缘服务器的详细信息。每条记录都包含了服务器的唯一编号、所属管理者id、服务器名称、ip地址以及服务器当前的状态。该表中的每个服务器都有一个唯一的编号，作为表的主键，而管理者id作为外键，关联到用户表中的用户id，以确定服务器的负责人。服务器状态字段则用于标识服务器是处于在线还是离线状态，为平台的资源调度和任务分配提供了依据。

表1-21 数据库边缘服务器表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **列名** | **类型** | **是否必须** | **备注** |
| 服务器编号 | id | bigint | 是 | 主键 |
| 管理者id | user\_id | bigint | 是 | 外键 |
| 服务器名称 | name | varchar(20) | 是 |  |
| ip地址 | address | varchar(256) | 是 |  |
| 服务器状态 | status | int | 是 | 1-离线，2-在线 |

**证人表(agent)**

证人表记录了网络中所有证人节点的基本信息，包括证人编号、管理者id、证人名称、ip地址和当前的在线状态。证人编号为每个证人节点提供了唯一标识，保证了网络中证人的唯一性和可追踪性。管理者id连接到用户表，标明了每个证人节点的管理者，而证人状态字段则明确了节点是否可用，对于维护网络的完整性和安全性非常重要。

表1-22 数据库证人表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **列名** | **类型** | **是否必须** | **备注** |
| 证人编号 | id | bigint | 是 | 主键 |
| 管理者id | user\_id | bigint | 是 | 外键 |
| 证人名称 | name | varchar(20) | 是 |  |
| ip地址 | address | varchar(256) | 是 |  |
| 证人状态 | status | int | 是 | 1-离线，2-在线 |

**设备表(device)**

设备表是为了管理和跟踪平台中使用的所有设备而设计的。它包括了设备编号、管理者id、设备名称、ip地址、设备类型以及当前的设备状态。每个设备都通过唯一的编号进行标识，而设备类型字段则区分了设备是手机、平板还是电脑等，为设备管理和任务调度提供了详尽的分类信息。设备状态字段则指出设备是在线还是离线，便于平台实时监控设备的可用性。

表1-23 数据库设备表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **列名** | **类型** | **是否必须** | **备注** |
| 设备编号 | id | bigint | 是 | 主键，雪花算法生成 |
| 管理者id | user\_id | bigint | 是 | 外键 |
| 设备名称 | name | varchar(20) | 是 |  |
| ip地址 | address | varchar(256) | 是 |  |
| 设备类型 | type | int | 是 | 1-手机，2-平板，3-电脑 |
| 设备状态 | status | int | 是 | 1-离线，2-在线 |

**模型表(model)**

模型表用于存储与联邦学习模型相关的所有信息，包括模型的唯一编号、上传者id、模型名称、版本、存储地址、类型和参数等。每个模型都由一个唯一的编号标识，这个编号通常通过雪花算法生成以保证全局唯一性。模型的上传者id作为外键，关联到用户表中的用户id，以跟踪模型的上传者。模型的状态字段标识了模型目前的状态，如正在准备训练、正在训练或已经投入使用。

表1-24 数据库模型表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **列名** | **类型** | **是否必须** | **备注** |
| 模型编号 | id | bigint | 是 | 主键，雪花算法生成 |
| 上传者id | user\_id | bigint | 是 | 外键 |
| 模型名称 | name | varchar(20) | 是 |  |
| 模型版本 | version | varchar(20) | 是 |  |
| 模型地址 | address | varchar(256) | 是 | 线上地址 |
| 模型类型 | type | int | 是 |  |
| 模型参数 | pt | varchar(256) | 是 | 线上地址 |
| 模型状态 | status | int | 是 | 1-准备训练，2-正在训练，3-正在使用 |

**数据集表(dataset)**

数据集表记录了所有联邦学习项目中使用的数据集的详细信息。每条记录包含数据集的唯一编号、上传者id、数据集名称、在线存储地址和数据集类型。就像模型表一样，数据集编号通常使用雪花算法生成，保证了其唯一性。上传者id关联到用户表，提供了数据集上传者的信息。数据集地址字段存储了数据集在线访问的url，而数据集类型字段则说明了数据集的种类，例如图像、文本或其他类型的数据集。

表1-25 数据库数据集表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **列名** | **类型** | **是否必须** | **备注** |
| 数据集编号 | id | bigint | 是 | 主键，雪花算法生成 |
| 上传者id | user\_id | bigint | 是 | 外键 |
| 数据集名称 | name | varchar(20) | 是 |  |
| 数据集地址 | address | varchar(256) | 是 | 线上地址 |
| 数据集类型 | type | int | 是 |  |

**部署信息表(deployment)**

部署信息表用于记录模型部署任务的详细情况，包括部署任务编号、部署者id、模型id、数据集id、部署状态和部署环境等信息。每个部署任务都有一个唯一编号，而部署状态字段则描述了任务是正在进行、已完成还是出现问题等状态，为模型的部署过程提供了全面的跟踪和管理。部署环境字段则指出了模型被部署的具体环境，如测试或生产环境，确保了部署任务的适应性和灵活性。

表1-26 数据库部署信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **列名** | **类型** | **是否必须** | **备注** |
| 部署任务编号 | id | bigint | 是 | 主键 |
| 部署者id | user\_id | bigint | 是 | 外键 |
| 模型id | model\_id | bigint | 是 | 外键 |
| 数据集id | dataset\_id | bigint | 是 | 外键 |
| 部署状态 | status | int | 是 |  |
| 部署环境 | environment | int | 是 |  |

**容器表(container)**

容器表记录了所有用于模型部署的容器的信息，如容器编号、部署任务id、容器名称和容器状态等。每个容器都由一个唯一的编号标识，并且关联到具体的部署任务，这样就能够清晰地追踪到每个容器是为哪个部署任务服务的。容器状态字段则提供了容器是处于运行中、已停止或发生错误等状态的信息，为容器的管理和监控提供了关键数据。

表1-27 数据库容器表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **列名** | **类型** | **是否必须** | **备注** |
| 容器编号 | id | bigint | 是 | 主键 |
| 部署任务id | deployment\_id | bigint | 是 | 外键 |
| 容器名称 | name | varchar(20) | 是 |  |
| 容器状态 | status | int | 是 |  |

**训练任务表(train)**

训练任务表是用来管理和跟踪联邦学习模型训练过程的数据库表。它包含了训练任务的唯一编号、发布者id、模型id、数据集id、训练状态、配置参数和训练结果。每个训练任务都有一个唯一的编号。发布者id、模型id和数据集id作为外键，分别关联到用户表、模型表和数据集表，这样就能清楚地知道谁在使用哪个模型和数据集进行训练。训练状态字段跟踪了任务的当前状态，而配置参数和训练结果以json字符串的形式存储，记录了训练的具体配置和最终结果。

表1-28 数据库训练任务表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **列名** | **类型** | **是否必须** | **备注** |
| 训练编号 | id | bigint | 是 | 主键 |
| 发布者id | user\_id | bigint | 是 | 外键 |
| 模型id | model\_id | bigint | 是 | 外键 |
| 数据集id | dataset\_id | bigint | 是 | 外键 |
| 训练状态 | status | int | 是 | 1-正在进行，2-任务暂停，3-任务结束 |
| 配置参数 | conf | text | 是 | json字符串 |
| 训练结果 | result | text | 是 | json字符串 |

**监测任务表(monitor)**

监测任务表用于记录和管理模型部署后的性能监测任务。它包含监测任务的唯一编号、发布者id、模型id、数据集id、监测状态和监测结果。类似于训练任务表，监测任务的每一个环节都通过外键与用户、模型和数据集建立了联系。监测状态字段说明了任务当前的执行状态，监测结果以json字符串格式保存，详细记录了模型在实际运行中的表现和可能出现的问题，以便进行后续的优化和调整。

表1-29 数据库监测任务表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **列名** | **类型** | **是否必须** | **备注** |
| 监测编号 | id | bigint | 是 | 主键 |
| 发布者id | user\_id | bigint | 是 | 外键 |
| 模型id | model\_id | bigint | 是 | 外键 |
| 数据集id | dataset\_id | bigint | 是 | 外键 |
| 监测状态 | status | int | 是 | 1-正在进行，2-任务暂停，3-任务结束 |
| 监测结果 | result | text | 是 | json字符串 |

Fed Privacy Shield平台概念数据模型图如下所下所示。

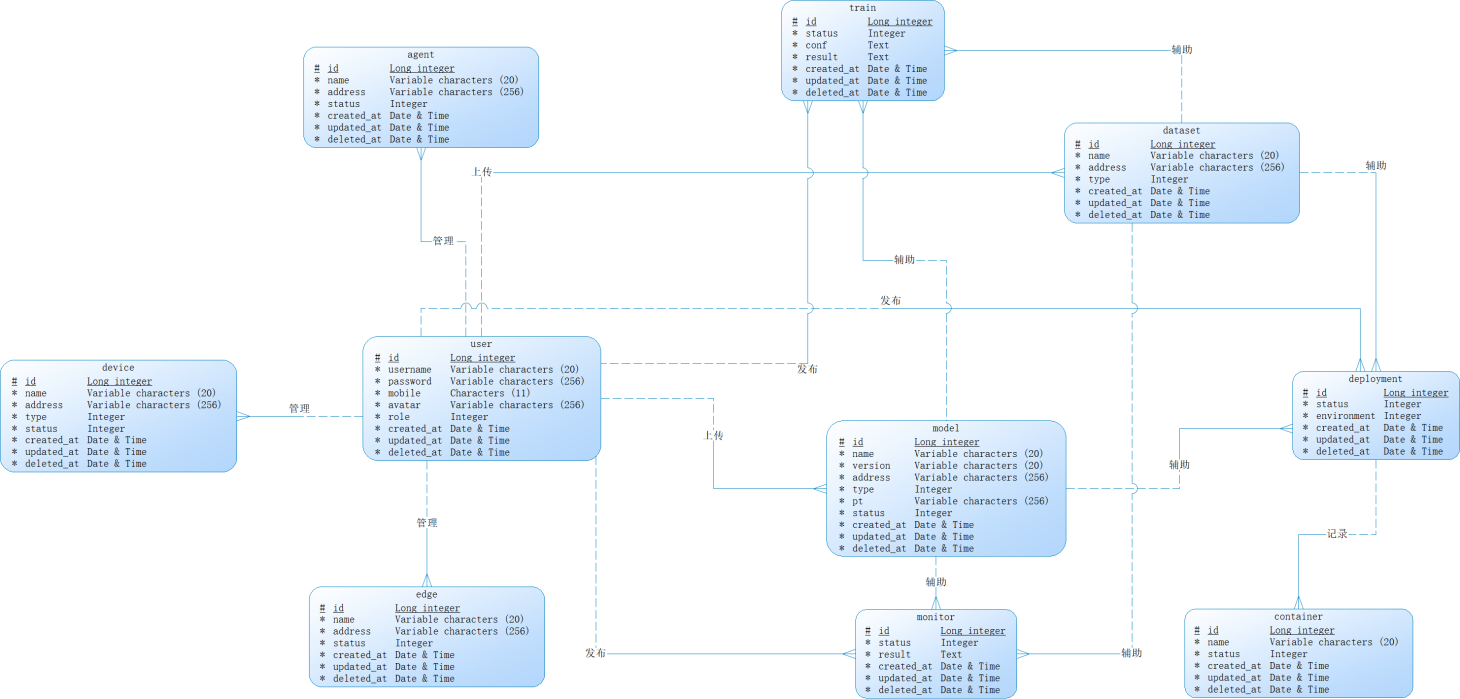


图1-31 系统概念数据模型图

## 1.5接口设计

本节概述Fed Privacy Shield平台的接口设计，分为了13个部分：用户相关、管理员相关、第三方服务相关、云服务器相关、边缘服务器相关、证人相关、端设备相关、部署相关、容器相关、数据集相关、模型相关、监测任务相关、训练相关。

为了使前后端的交互更加方便和规范，我们规定了统一的结果返回实体，如下所示：

{

"code": "响应码",

"msg": "返回信息",

"token": "返回token",

"data": "返回数据"

}

其中token字段和data字段可省。

**用户相关接口**

表1-30 用户相关接口表

|  |  |  |  |
| --- | --- | --- | --- |
| **功能描述** | **请求方法** | **URL** | **返回值** |
| 注册 | POST | /api/user/register | 返回token |
| 登录 | POST | /api/user/login | 返回token |
| 修改密码 | PUT | /api/user/reset\_password | 返回token |
| 更新用户信息 | PUT | /api/user/ update\_user | 返回token |
| 获取用户信息 | GET | /api/user/get\_user | 用户信息实体 |

**管理员相关接口**

表1-31 管理员相关接口表

|  |  |  |  |
| --- | --- | --- | --- |
| **功能描述** | **请求方法** | **URL** | **返回值** |
| 登录 | POST | /api/admin/login | 返回token |
| 修改密码 | PUT | /api/admin/reset\_password | 返回token |
| 删除用户 | DELETE | /api/admin/delete\_user | 返回token |
| 更新用户信息 | PUT | /api/admin/ update\_user | 返回token |
| 获取用户信息 | GET | /api/admin/get\_user | 用户信息实体 |
| 获取用户列表 | GET | /api/admin/get\_user\_list | 用户信息实体列表 |
| 获取管理员信息 | GET | /api/admin/get\_admin | 管理员信息实体 |
| 更新管理员信息 | PUT | /api/admin/update\_admin | 返回token |

**第三方服务相关接口**

表1-32 第三方服务相关接口表

|  |  |  |  |
| --- | --- | --- | --- |
| **功能描述** | **请求方法** | **URL** | **返回值** |
| 上传文件 | POST | /api/third\_party/upload\_file | 返回云端地址实体 |
| 删除文件 | DELETE | /api/third\_party/delete\_file | 返回token |
| 获取图片验证码 | GET | /api/third\_party/captcha | 图片验证码实体 |
| 获取短信验证码 | POST | /api/third\_party/send\_sms | 短信验证码实体 |

**云服务器相关接口**

表1-33 云服务器相关接口表

|  |  |  |  |
| --- | --- | --- | --- |
| **功能描述** | **请求方法** | **URL** | **返回值** |
| 获取云服务器信息列表 | GET | /api/cloud/list | 云服务器信息实体列表 |
| 获取特定云服务器信息 | GET | /api/cloud/{id} | 云服务器信息实体 |

**边缘服务器相关接口**

表1-34 边缘服务器相关接口表

|  |  |  |  |
| --- | --- | --- | --- |
| **功能描述** | **请求方法** | **URL** | **返回值** |
| 启动边缘服务器 | POST | /api/edge/run | 返回token |
| 注销边缘服务器 | DELETE | /api/edge/{id} | 返回token |
| 获取边缘服务器信息 | GET | /api/edge/{id} | 边缘服务器信息实体 |
| 更新边缘服务器信息 | PUT | /api/edge/{id} | 返回token |
| 获取边缘服务器信息列表 | GET | /api/edge/list | 边缘服务器信息实体列表 |

**证人相关接口**

表1-35 证人相关接口表

|  |  |  |  |
| --- | --- | --- | --- |
| **功能描述** | **请求方法** | **URL** | **返回值** |
| 启动证人 | POST | /api/agent/run | 返回token |
| 注销证人 | DELETE | /api/agent/{id} | 返回token |
| 获取证人信息 | GET | /api/agent/{id} | 证人信息实体 |
| 更新证人信息 | PUT | /api/agent/{id} | 返回token |
| 获取证人信息列表 | GET | /api/agent/list | 证人信息实体列表 |

**端设备相关接口**

表1-36 端设备相关接口表

|  |  |  |  |
| --- | --- | --- | --- |
| **功能描述** | **请求方法** | **URL** | **返回值** |
| 启动设备 | POST | /api/device/run | 返回token |
| 注销设备 | DELETE | /api/device/{id} | 返回token |
| 获取设备信息 | GET | /api/device/{id} | 设备信息实体 |
| 更新设备信息 | PUT | /api/device/{id} | 返回token |
| 获取设备信息列表 | GET | /api/device/list | 设备信息实体列表 |

**部署相关接口**

表1-37 部署相关接口表

|  |  |  |  |
| --- | --- | --- | --- |
| **功能描述** | **请求方法** | **URL** | **返回值** |
| 获取部署信息 | GET | /api/image/{id} | 部署信息实体 |
| 删除特定镜像 | DELETE | /api/image/{id} | 返回token |
| 获取部署信息列表 | GET | /api/image/list | 部署信息实体列表 |
| 更新镜像配置文件 | PUT | /api/image/{id} | 返回token |

**容器相关接口**

表1-38 容器相关接口表

|  |  |  |  |
| --- | --- | --- | --- |
| **功能描述** | **请求方法** | **URL** | **返回值** |
| 创建容器 | POST | /api/container/run | 返回token |
| 获取容器信息 | GET | /api/container/{id} | 容器信息实体 |
| 删除特定容器 | DELETE | /api/container/{id} | 返回token |
| 获取容器信息列表 | GET | /api/container/list | 容器信息实体列表 |

**数据集相关接口**

表1-39 数据集相关接口表

|  |  |  |  |
| --- | --- | --- | --- |
| **功能描述** | **请求方法** | **URL** | **返回值** |
| 新增数据集 | POST | /api/dataset/add | 返回token |
| 更新数据集 | PUT | /api/dataset/{id} | 返回token |
| 获取数据集信息 | GET | /api/dataset/{id} | 数据集信息实体 |
| 删除特定数据集 | DELETE | /api/dataset/{id} | 返回token |
| 获取数据集信息列表 | GET | /api/dataset/list | 数据集信息实体列表 |

**模型相关接口**

表1-40 模型相关接口表

|  |  |  |  |
| --- | --- | --- | --- |
| **功能描述** | **请求方法** | **URL** | **返回值** |
| 新增模型 | POST | /api/model/add | 返回token |
| 更新模型信息 | PUT | /api/model/{id} | 返回token |
| 获取模型信息 | GET | /api/model/{id} | 模型信息实体 |
| 删除特定模型 | DELETE | /api/model/{id} | 返回token |
| 获取模型信息列表 | GET | /api/model/list | 模型信息实体列表 |

**监测任务相关接口**

表1-41 监测任务相关接口表

|  |  |  |  |
| --- | --- | --- | --- |
| **功能描述** | **请求方法** | **URL** | **返回值** |
| 新增监测任务 | POST | /api/monitor/run | 返回token |
| 暂停监测任务 | GET | /api/monitor/pause/{id} | 返回token |
| 重启监测任务 | GET | /api/monitor/restart/{id} | 返回token |
| 终止监测任务 | GET | /api/monitor/terminate/{id} | 返回token |
| 删除监测任务 | DELETE | /api/monitor/{id} | 返回token |
| 获取监测信息 | GET | /api/monitor/{id} | 监测信息实体 |
| 获取监测信息列表 | GET | /api/monitor/list | 监测信息实体列表 |

**训练相关接口**

表1-42 训练相关接口表

|  |  |  |  |
| --- | --- | --- | --- |
| **功能描述** | **请求方法** | **URL** | **返回值** |
| 新增训练 | POST | /api/train/run | 返回token |
| 暂停训练 | GET | /api/train/pause/{id} | 返回token |
| 重启训练 | GET | /api/train/restart/{id} | 返回token |
| 终止训练 | GET | /api/train/terminate/{id} | 返回token |
| 上传模型参数 | POST | /api/train/upload\_pt | 返回token |
| 删除训练记录 | DELETE | /api/train/{id} | 返回token |
| 训练任务对比 | POST | /api/train/compare | 训练对比信息实体 |
| 获取训练信息 | GET | /api/train/{id} | 训练信息实体 |
| 获取训练信息列表 | GET | /api/train/list | 训练信息实体列表 |

## 1.6本章小结

在本章中，我们全面展开了Fed Privacy Shield平台的平台设计。章节内容涵盖了从总体架构到接口实现的多个关键设计领域，确保读者对平台的构建和运作有一个清晰的理解。

**第1.1节**，我们概述了平台的总体设计。这部分内容详尽介绍了平台架构的层次和组成、平台的总体流程和不同模块的功能等等。

**第1.2节**详细描述了平台的功能设计。我们依据不同角色——训练模型用户和平台管理人员——细分并具体描述了平台的功能模块。

**第1.3节**，我们针对平台的核心技术进行了细致阐述，描述了聚合方案、通信传输和安全机密三个方面的多种算法。

**第1.4节**描述了平台的数据库设计。详细描述了平台的 MySQL关系数据库存储数据，共包含十 张表。分别是用户表、边缘服务器表、证人表、设备表、模型表、数据集表、部署信息表、容器表、训练任务表、监测任务表。

**第1.5节**中，我们描述了平台的接口设计，主要分为了十三个部分：用户相关、管理员相关、第三方服务相关、云服务器相关、边缘服务器相关、证人相关、端设备相关、部署相关、容器相关、数据集相关、模型相关、监测任务相关、训练相关。