基于联邦学习的分布式医疗影像诊断系统 - 需求分析报告

1. 引言

1.1 项目背景

> 当前，医疗影像诊断领域面临着数据、隐私、效率与资源均衡等多重挑战。为应对这些挑战，本项目旨在构建一个基于联邦学习（Federated Learning, FL）的分布式**肺结节检测系统**。该系统利用LUNA16数据集进行肺结节的智能识别，在不暴露原始数据的前提下，通过多方数据协同训练精准的AI模型，从而提升诊断智能化水平、保障数据安全、促进医疗资源共享。

本项目旨在解决以下四大核心挑战：

1. **诊断智能化升级需求**：传统人工阅片耗时、费力，且存在主观误差与漏诊风险。AI可提升效率与精度，但其模型训练高度依赖大规模、多样化的数据集。
2. **数据孤岛与协同合作难题**：海量医疗数据分散于各机构，因隐私、法规和技术壁垒无法自由共享，严重阻碍了高性能AI模型的研发进程。
3. **数据隐私与安全问题**：医疗数据作为高度敏感的个人信息，在传统的集中式处理模式下面临巨大的泄露风险，亟需更安全的数据合作范式。
4. **医疗资源均衡化挑战**：优质医疗资源（特别是专家经验和先进设备）分布不均，导致基层和偏远地区的诊断水平受限，技术赋能成为必然趋势。

1.2 项目目标

本项目的核心目标是开发一个**安全、高效、可扩展的肺结节检测联邦学习平台**。具体目标分解如下：

- **构建安全的联邦学习框架**：实现一个允许各参与方（如医院、诊所）在本地进行模型训练，仅通过交换加密后的模型参数来聚合全局模型的系统，确保原始患者数据不出本地。
- **实现高精度的肺结节检测模型**：利用联邦学习汇集多方数据知识，基于先进的3D UNet架构，训练出在肺结节检测任务上具有高准确率和强泛化能力的深度学习模型。
- **开发可视化的Web交互平台**：基于Flask和WebSocket技术，为服务端（管理员）和客户端（医疗机构）提供直观、实时的交互界面，用于用户管理、数据上传、训练监控和结果展示。
- **保障系统的安全与可追溯性**：集成用户认证、会话管理、实时日志等关键机制，并支持云端数据库（Supabase）与本地存储的双重备份，确保系统稳定可靠。
- **提供智能推理服务**：基于训练成熟的联邦模型，为医疗机构提供CT影像的肺结节智能检测服务，并生成图文并茂的可视化诊断报告。

2. 功能需求 (Functional Requirements)

2.1 用户管理

功能模块	功能描述	优先级
角色划分	系统支持两种核心角色：**服务端 (Server)** 和 **客户端 (Client)**。 服务端负责管理整个联邦学习流程和模型聚合，客户端作为参与方贡献数据和算力。	高
用户认证	基于 `bcrypt` 加密的安全登录机制， 支持本地JSON存储和Supabase云数据库双重认证。 用户需通过用户名和密码进行身份验证。	高
用户注册	提供用户注册功能，新用户默认角色为客户端，支持邮箱验证和密码强度检查。	高
会话管理	基于 `Flask Session` 的会话管理，系统能跟踪在线用户状态、角色权限， 并通过WebSocket实时更新连接状态。	高
数据库集成	优先使用 `Supabase` 云数据库存储用户信息，本地JSON文件作为备份方案， 确保系统稳定性。	中

2.2 数据管理

功能模块	功能描述	优先级
医疗影像数据上传	客户端用户能够通过Web界面上传LUNA16格式的医疗影像数据 （ `.mhd` 和 `.raw` 文件对），支持批量上传和文件完整性验证。	高
数据本地化存储	遵循联邦学习原则，客户端上传的数据存储在服务器的独立目录中 （ `uploads/{username}_data` ），训练时直接从本地读取，不进行跨客户端共享。	高
数据格式支持	专门支持LUNA16数据集的 `.mhd` （元数据文件）和 `.raw` （原始影像数据） 格式，确保文件配对完整性。	高
数据状态监控	服务端实时监控各客户端的数据准备情况，包括上传状态、文件数量、 文件配对情况等，并通过WebSocket实时更新。	高
标注数据管理	系统集成LUNA16数据集的标注信息（ `annotations.csv` ）， 包含肺结节位置坐标和尺寸信息，用于训练和验证。	高

2.3 联邦学习训练

功能模块	功能描述	优先级
训练任务发起	服务端可通过Web界面发起联邦学习训练任务，支持自定义全局训练轮数（1-20轮）和本地训练轮数（1-10轮）等超参数。	高

功能模块	功能描述	优先级
FedAvg算法实现	采用经典的 `FedAvg` （Federated Averaging）算法进行模型聚合， 确保各客户端模型参数的有效融合。	高
3D UNet模型架构	基于 `3D UNet` 网络架构进行肺结节检测，支持三维医疗影像的语义分割任务。	高
分布式训练协调	实现联邦服务器（ `FederatedServer` ）和联邦客户端（ `FederatedClient` ） 的协调机制，支持异步训练和参数聚合。	高
实时训练监控	通过WebSocket实时广播训练进度，包括当前轮次、损失值、 参与客户端数量等关键指标。	高
训练历史记录	系统自动保存训练会话信息到 `training_history.json` ，包括损失变化、 参与客户端、模型路径等详细记录。	高
模型版本管理	每个训练会话生成独立的模型文件 （ `models/session_{id}_federated_model.pth` ），支持模型版本控制和回滚。	中

2.4 模型推理与诊断

功能模块	功能描述	优先级
在线推理服务	服务端和客户端均可上传新的CT影像数据， 调用训练好的联邦模型进行肺结节智能检测。	高
联邦模型推理	系统集成 `FederatedLungNodulePredictor` 类， 支持加载联邦训练模型并进行高精度预测。	高
快速推理模式	提供快速推理选项，在保证准确性的前提下提升推理速度，满足临床实时诊断需求。	中
结果可视化	推理完成后，系统自动生成结节检测结果的可视化图像， 在原始CT切片上标记检测到的结节位置和置信度。	高
推理状态监控	实时显示推理进度，包括模型加载、图像处理、结果生成等各个阶段的状态信息。	高
多文件推理	支持批量上传多个 `.mhd/.raw` 文件对进行批量推理，提高诊断效率。	中

2.5 系统监控与可视化

功能模块	功能描述	优先级
服务端仪表盘	基于Flask模板的中心化仪表盘，展示系统总体状态、在线用户列表、 客户端数据状态、训练进度等关键信息。	高
客户端仪表盘	为客户端提供简洁的Web界面，显示连接状态、数据上传信息、训练参与状态等。	高

功能模块	功能描述	优先级
实时通信系统	基于 Socket.IO 的WebSocket实时通信，支持状态广播、日志推送、进度更新等实时交互。	高
双重日志系统	分别记录服务器运行日志和训练过程日志，通过队列管理日志缓存，支持实时日志查看。	高
训练可视化	集成 Chart.js 图表库，实时显示训练损失变化、客户端参与情况等可视化图表。	高
历史记录查看	提供专门的历史记录页面，用户可查看过往训练会话的详细信息和性能指标。	中

3. 技术架构与实现

3.1 系统架构

本项目采用分层架构设计，确保模块化、可扩展性和可维护性。主要技术组件如下：

3.1.1 前端技术栈

- **UI 框架**：`HTML` / `CSS` / `JavaScript` (构建现代Web标准的用户界面)
- **实时通信**：`Socket.IO Client` (实现与后端的实时双向通信)
- **数据可视化**：`Chart.js` (用于训练过程和结果的数据可视化图表)
- **图标库**：`Font Awesome` (提供丰富的矢量图标，美化界面)

3.1.2 后端技术栈

- **Web 框架**：`Flask 3.0.3` (轻量级、灵活的Web应用框架)
- **实时通信**：`Flask-SocketIO 5.5.1` (为Flask应用提供WebSocket支持)
- **深度学习**：`PyTorch 2.6.0` (用于深度学习模型的训练、推理和管理)
- **医学影像处理**：`SimpleITK 2.5.0` (专业的医学影像读取、处理和分析库)
- **数据处理**：`NumPy` / `Pandas` (科学计算与数据分析的核心库)
- **安全**：`bcrypt` (用于密码的哈希加密与安全验证)

3.1.3 数据存储

- **云数据库**：`Supabase` (基于PostgreSQL的云数据库，作为主要的用户信息存储)
- **本地备份**：`本地JSON文件` (作为用户数据的本地备份方案，增强系统鲁棒性)
- **文件系统**：`本地文件系统` (用于存储医疗影像数据、模型文件等非结构化数据)

3.2 核心模块

模块名称	文件路径	功能描述
Web应用主程序	`app.py`	Flask应用的核心入口，负责路由管理、WebSocket事件处理和整体流程协调。
联邦学习训练	`src/federated_training.py`	实现了 `FedAvg` （Federated Averaging）算法，负责模型参数的聚合与更新。
联邦推理引擎	`src/federated_inference_utils.py`	封装了模型推理和结果可视化的功能，支持加载联邦模型进行预测。
3D UNet模型	`src/train_simple_model.py`	定义了用于肺结节检测的 `3D UNet` 深度学习网络结构。
数据标注	`src/annotations.csv`	包含了LUNA16数据集的肺结节标注信息，是模型训练的“答案”。

3.3 数据流程

1. **用户认证**: 客户端用户通过Web界面输入凭证，后端使用 `bcrypt` 进行验证后登录。
2. **数据上传**: 登录后的客户端可将本地的 `.mhd/.raw` 影像文件上传至服务器，文件存储在与用户绑定的专属目录中。
3. **训练发起**: 服务端管理员在仪表盘上配置训练参数（如全局轮次），启动联邦学习任务。
4. **模型聚合**: 各客户端在本地进行模型训练，并将加密后的模型参数发送至服务端，服务端使用 `FedAvg` 算法进行聚合，生成新的全局模型。
5. **推理服务**: 用户上传新的CT影像，系统调用训练好的联邦模型进行肺结节检测。
6. **结果展示**: 系统将检测结果（如结节位置）在原始CT切片上进行可视化标记，并生成报告。

4. 非功能性需求 (Non-Functional Requirements)

4.1 性能需求

需求类别	描述
响应时间	Web界面核心操作响应时间应小于 **2秒**；WebSocket实时通信延迟应低于 **100毫秒**；模型推理时间根据影像大小应在 **1-5分钟** 内完成。
并发性	系统需支持多个客户端同时在线（理论上无限制），支持并发数据上传，单个训练会话支持多客户端同时参与。
训练效率	单轮联邦学习训练（参数分发-本地训练-聚合）时间应控制在合理范围内，并支持可配置的训练轮数和本地轮数以平衡效率与精度。

需求类别	描述
内存使用	`3D UNet` 模型在训练时，GPU显存占用应小于 **8GB**， 并支持在无GPU环境下自动降级至CPU模式运行。

4.2 可用性需求

需求类别	描述
易用性	采用响应式Web界面设计，提供直观的仪表盘布局。用户无需复杂培训即可完成数据上传、 训练发起、结果查看等核心操作。
可视化效果	集成 `Chart.js` 实时图表，确保训练损失曲线、客户端状态、推理结果等关键信息能够清晰、 直观地展示。
容错性	文件上传支持格式验证和文件配对检查；错误操作提供明确、友好的提示信息； 支持训练中断后的状态恢复。
多语言支持	界面文案采用中文，代码注释采用中英文混合， 以适应中文用户的使用习惯和开发者的维护需求。

4.3 可靠性需求

需求类别	描述
稳定性	系统应支持7x24小时长时间稳定运行。训练过程中， 单个客户端的网络波动或掉线不应影响整体训练进程，并支持断点续训。
数据一致性	通过WebSocket实时同步机制，确保服务端和所有客户端的状态保持一致。 训练历史等关键数据需持久化存储，防止意外丢失。
错误处理	具备完善的异常捕获和处理机制，当系统发生崩溃或意外中断时， 能够自动恢复到最近的稳定状态。
备份机制	采用**双重存储**（`Supabase` 云数据库 + 本地JSON文件）策略，确保用户数据的安全与高可用性。 模型文件按会话ID自动进行版本管理。

4.4 可维护性与可扩展性需求

需求类别	描述
模块化设计	项目遵循高内聚、低耦合的原则，采用清晰的目录结构进行功能划分（ `app.py` 主程序， `src/` 核心算法，`templates/` 前端模板，`static/` 静态资源）。
代码规范	后端Python代码遵循 **PEP 8** 编码规范，包含详细的函数注释和模块说明， 便于团队协作和后期维护。

需求类别	描述
配置管理	支持通过环境变量（ `.env` 文件）进行配置管理，数据库连接、模型路径等关键参数可灵活配置，无需硬编码。
可扩展架构	架构设计支持未来平滑地新增客户端节点、集成新的深度学习模型或尝试不同的联邦学习算法。
版本控制	训练生成的模型按会话ID自动进行版本管理，训练历史被完整记录，支持模型回滚、效果对比和问题追溯。

4.5 安全需求

需求类别	描述
数据隐私	严格遵守联邦学习核心原则，客户端的原始医疗影像数据**绝对不离开**本地服务器，仅在各方之间交换加密后的模型参数。
身份认证	基于 `bcrypt` 实现强密码加密存储，提供安全的用户注册和登录机制，有效防止未授权的非法访问。
会话安全	为 `Flask Session` 配置安全密钥（Secret Key），确保WebSocket连接在建立前需要经过身份验证，防止恶意连接和会话劫持。
访问控制	实施严格的角色权限控制（RBAC），客户端只能访问自身相关的数据和功能，服务端拥有全局管理权限。
数据传输	模型参数在传输过程中包含数据完整性校验机制，并为未来集成TLS/SSL加密传输预留接口，保障链路安全。

5. 约束条件与假设

5.1 技术约束

- **硬件要求**: 推荐使用NVIDIA GPU（显存 > 8GB）以加速模型训练，最低支持CPU模式运行。
- **操作系统**: 跨平台支持，可在Windows、Linux、macOS等主流操作系统上部署。
- **Python版本**: 项目依赖Python 3.8+环境，推荐使用Python 3.9+。
- **浏览器兼容**: 客户端界面兼容Chrome、Firefox、Safari等现代主流浏览器。

5.2 数据约束

- **数据格式**: 系统目前专门针对LUNA16数据集的 `.mhd/.raw` 格式进行处理。
- **数据规模**: 单个CT扫描文件大小通常在100MB至1GB之间，系统需能处理此规模的数据。
- **标注数据**: 模型训练依赖LUNA16官方提供的 `annotations.csv` 文件进行监督学习。

5.3 业务假设

- **网络连接**: 假设所有参与联邦学习的客户端与中心服务器之间具备稳定、可靠的网络连接。
- **数据合规**: 假设所有参与方均已获得其所使用数据的必要法律与伦理授权。
- **技术能力**: 假设系统使用者具备基本的Web浏览器操作和文件管理能力。

6. 总结

本需求分析报告详细阐述了“基于联邦学习的分布式肺结节检测系统”的功能需求、非功能性需求、技术架构及约束条件。该系统通过创新的联邦学习架构，在有效保护患者隐私的前提下，实现了多方数据的协同训练，为医疗影像智能诊断领域提供了一个安全、高效且可扩展的解决方案。

系统的核心价值在于：

1. **隐私保护为本**：原始医疗数据不出本地，从根本上解决了数据共享中的隐私安全痛点。
2. **打破数据孤岛**：使得多家医疗机构可以安全、高效地合作，共同训练出性能更强大的AI模型。
3. **部署轻便快捷**：基于Web的B/S架构支持快速部署和使用，降低了技术门槛。
4. **管理直观可视**：提供直观的Web界面和实时监控图表，极大地提升了用户体验。
5. **融合前沿技术**：是联邦学习、深度学习与现代Web技术在医疗领域深度融合的前沿应用。

该系统的成功实施，不仅能产出一个高质量的肺结节检测模型，更将为医疗影像AI的发展和推广提供重要的技术支撑和应用示范。