面向算力物联网的联邦学习系统及设计研究进展

鲁剑锋[1]，祁盼[1]，潘林雨[2]，李冰[1]，曹书琴[1]，靳延安[3]
（1. 武汉科技大学计算机科学与技术学院，湖北 武汉 430081；2. 中国人民解放军91999部队，山东 青岛 266001；
3. 湖北经济学院信息管理学院，湖北 武汉 430205）

摘　要：算力物联网（CPIoT, computing power Internet of things）通过整合物联网（IoT, Internet of things）设备与强大的计算资源，为数据密集型任务提供了强大的支持，实现了智能决策。在 CPIoT 的隐私保护需求背景下，联邦学习（FL, federated learning）作为一种旨在保护数据隐私、进行分布式学习的技术，为解决数据“孤岛”问题、执行复杂训练任务及大模型训练提供了新途径。虽然研发人员一直致力于开发更加成熟的 FL 系统以适应 CPIoT 环境，但目前的研究在深入探讨 FL 系统设计技术的优势与短板、技术特点与差异、支持与适用情况等方面仍然存在不足。因此，首先深入研究了当前业内有影响力的 FL 系统，包括开源框架和基准测试平台，并在 CPIoT 的不同技术维度上深入对比分析系统设计差异，建立了 CPIoT 环境下详细的 FL 开源框架与基准测试平台的选择标准及建议，使开发人员可以更加高效地选择合适的框架及平台。然后，列举了多种 CPIoT 场景下 FL 系统的选择与完整系统搭建的实验，使开发人员可以更好地借助上述技术实现 FL 应用。最后，总结了 FL 系统设计领域的标准化现状和发展挑战，并对未来发展进行了展望。旨在全面概述 FL 系统及其设计研究进展，为推动 CPIoT 与 FL 网络的深度融合提供参考，也为未来研究提供思路。

关键词：算力物联网；联邦学习；开源框架；基准测试平台；计算范例
中图分类号：TP311
文献标志码：A
doi：10.11959/j.issn.2096−3750.2024.00438

Recent advances on federated learning systems and the design for computing power Internet of things

LU Jianfeng[1], QI Pan[1], PAN Linyu[2], LI Bing[1], CAO Shuqin[1], JIN Yan’an[3]
1. School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430081, China
2. 91999 Unit of the Chinese People’s Liberation Army, Qingdao 266001, China
3. School of Information Management, Hubei University of Economics, Wuhan 430205, China

Abstract: Computing power Internet of things (CPIoT) integrates Internet of things (IoT) devices with substantial computational resources to support data-intensive tasks, facilitating intelligent decision-making. Within the context of privacy protection requirements for CPIoT, federated learning (FL) that is a distributed learning technique upholds data privacy, and offers a novel approach to addressing data silos for executing complex training tasks, and training large models. Although researchers have been committed to develop more mature federated learning systems to adapt to the CPIoT environment, current research lacks in-depth exploration of the strengths and limitations, technical features and differences, and support and applicability of federated learning system design techniques. Firstly, the most influential federated

收稿日期：2024−10−08；**修回日期：**2024−11−19
通信作者：潘林雨，ply92323@sina.com
基金项目：国家自然科学基金资助项目（No. 62072411，No. 62372343，No. 62402352）；浙江省自然科学基金资助项目（No. LR21F020001）；湖北省重点研发项目（No. 2023BEB024）；武汉东湖新技术开发区“揭榜挂帅”项目（No. 2024KJB301）
Foundation Items: The National Natural Science Foundation of China (No. 62072411, No. 62372343, No. 62402352), The Zhejiang Provincial Natural Science Foundation (No. LR21F020001), The Key Research and Development Program of Hubei Province (No. 2023BEB024), The “Revealing the Leader” Project in Wuhan Donghu New Technology Development Zone (No. 2024KJB301)

learning systems in the industry were studied, including open-source frameworks and benchmarking platforms. The system design differences in various technical dimensions of CPIoT in an in-depth comparison were analyzed. Detailed criteria and recommendations for selecting open-source frameworks and benchmarking platforms in the CPIoT environment were established, so that developers could efficiently choose the most suitable frameworks and platforms. Seeondly, various experiments for selecting federated learning systems and building complete systems were presented in multiple CPIoT scenarios, to assist developers in better realizing federated learning applications by utilizing the aforementioned technologies. Finally, the current state of standardization and development challenges in the field of federated learning system design were summarized, and future development prospects were discussed. The purpose is to provide a comprehensive overview of FL systems and the design research progress, serving as a reference for the deep integration of CPIoT and FL networks and offering insights for future research.

Key words: CPIoT, FL, open-source framework, benchmarking platform, computing paradigm

0 引言

人工智能（AI, artificial intelligence）作为当前信息技术领域的重要发展方向，已经广泛应用于金融、医疗、物流等领域[1-4]。算力物联网（CPIoT, computing power Internet of things）作为AI与物联网（IoT, Internet of things）的结合体，成为信息技术领域的重要发展方向。CPIoT将算力嵌入物联网设备，使设备可以执行复杂的计算任务并进行智能决策，在智慧城市、智能家居和工业自动化等领域展现出了巨大潜力。此外，AI的训练通常需要海量数据支撑以保证其性能，但由于大部分领域的数据存在分布式、私有化的特点，数据使用和共享方式依然面临着诸多挑战，特别是涉及个人隐私和数据安全方面。这种数据的隔离状态形成了数据“孤岛”问题[5]，意味着数据资源无法跨平台或跨组织流动，这不仅限制了数据的潜在价值，也阻碍了AI的发展和应用，因为它限制了模型训练和分析的数据量与多样性。据国际数据公司（IDC, International Data Corporation）估计，至2025年，将有超过557亿台入网设备，其中75%的入网设备将连接到物联网平台[6]。预估这将导致这些设备生成的数据流增至73.1 ZB[7]。Gartner相关报告显示，2025年之前，约60%的大型企业将应用至少一种隐私保护计算技术。联邦学习（FL, federated learning）在迫切需要打破“孤岛”壁垒的需求下应运而生，它允许参与方在不暴露原始数据的前提下，实现数据的共享和模型的协同训练[8]。得益于CPIoT技术的发展，特别是在FL框架和基准测试平台方面的研究，为解决这些问题提供了新的解决方案，为用户数据共享提供了新的解决方案，增加了可用数据的总量，打破了数据“孤岛”壁垒，充分挖掘了隐私数据中的潜在价值。

随着IoT技术和FL技术的迅猛发展，这一领域已获得学术界与产业界的广泛关注，相关研究成果被收录于各类国际期刊与会议。在此背景下，众多FL框架与基准测试平台相继涌现，这些系统设计技术不仅面向CPIoT提供了高效可靠的FL系统快速搭建方案，而且适应多种现实应用场景，同时提供了一系列隐私保护策略。此外，它们还为FL新兴算法的诞生提供了基准测试条件。例如，面向学术研究领域的Flower[9]、面向工业生产领域的Mindspore Federated[10]、面向自然语言处理的FedNLP[11]、业内首个同时支持3种FL架构的FATE[12]、集成了联邦攻击策略的FederatedScope[13]和FedML[14]、面向基准测试的FedScale[15]等，这些技术在不同维度上存在着显著差异。由于尚不存在各技术维度上全面占优的特定FL系统设计技术，且各项目通常无法在项目设计文档中细致全面地描述其各个维度的特点，这导致FL系统开发人员需要花费大量精力对比各框架（或基准测试平台）以选择最大程度符合开发、生产、维护等各方面需求的系统。因此，全面详细且多维度的系统结构设计调查和完整深入的对比分析对FL系统设计领域是一项必要且非常有意义的工作。

本文的内容框架如图1所示。首先，面向CPIoT详细地横向对比各FL开源框架、基准测试平台的系统设计技术以剖析它们的设计优势与短板，基于详细到各技术维度的对比结果建立了框架与平台选择标准。然后，列举了多种CPIoT应用场景下的完整FL系统搭建实验，基于不同系统开发需求选取适宜的系统设计技术并搭建系统进行实验，为开发

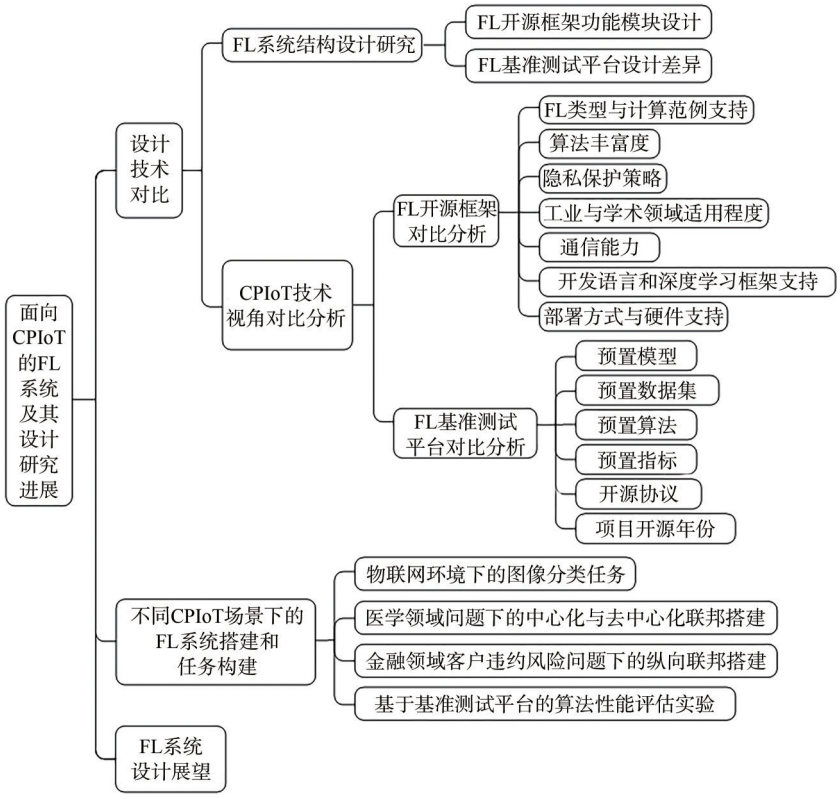

图1　本文的内容框架

人员实现FL的应用提供参考。最后，进一步在FL系统设计领域的宏观层面给出对该领域发展的展望，为未来的研究提供参考和思路。

本文的主要贡献包括以下4个方面。

1) 剖析了框架的优势与短板，面向CPIoT为读者建立FL框架选择标准。针对20个主流FL开源框架进行分析和横向对比，通过对系统结构设计的横向对比揭示了架构设计差异，进一步面向技术实现视角深入对比了7个主要视角下各框架的支持程度和性能。

2) 分析了基准测试平台之间的差异。针对3个主流FL基准测试平台进行结构设计分析，对平台内置数据集信息、平台技术特点和算法评估指标进行横向对比，分析平台之间的特点和差异，并进一步提供了平台选择建议。

3) 提供了多种不同CPIoT场景下的FL系统完整搭建和任务构建的具体思路。针对各场景，根据任务的需求和特征选择合适的系统设计技术，并设计了完整的FL系统搭建实验，最后基于实验结果对系统的有效性进行了详细分析。

4) 在FL系统设计领域的宏观层面，对领域的标准化现状进行了综述，并探讨了当前面临的挑战。此外，对该领域未来的发展提出了展望，为学术界和工业界的研发人员提供参考和启示。

1　FL系统结构设计研究

FL的显著优点是可实现训练过程中的隐私保护，减少了本地数据的泄露风险，并且FL训练出的模型通常由于数据多样性而具备较好的泛化能力。然而FL系统的可用性仍然受到以下因素的影响：1) 训练过程依赖于频繁通信，因此对通信条件的要求较高[16]；2) 现实中非独立同分布的数据通常不利于FL模型的训练[17]，因此需要高性能的算法解决该问题；3) 存在安全性和信任问题，训练过程中恶意第三方或参与方的攻击可能导致训练失败或隐私泄露[18]，因此构建安全的FL系统是一个重要的研究方向；4) 基于分布式网络构建的训练程序通常难以调试和监控，相关算法性能需要构建模拟环境来进行实验和评估。为了保障FL系统在现实世界的落地和成熟，各系统都采用不同的结构设计，本文针对上述问题提出了解决方案。此外，这些设计也使上述系统在各维度体现出不同的特点和能力。

FL基准测试平台是评估和比较FL算法和系统

性能的重要组成部分，属于FL系统设计技术的一种。与FL框架的主要区别在于：1) 基准测试平台通常用于与基线算法对比并评估算法性能，而FL框架通常用于完整的FL系统搭建；2) 基准测试平台通常具备更高的评估指标丰富度；3) 基准测试平台通常具备较多且高质量的数据集、领域公认且常用的基线算法；4) 部分基准测试平台还具备一些攻击策略脚本以及与算法抵御攻击能力相关的评估指标。

由于框架（或基准测试平台）之间存在显著的设计差异，为了降低开发人员编写代码的难度，提高系统的可靠性和可扩展性，系统会进行模块化设计，因此系统之间的结构设计差异主要体现在：是否具备与某功能相关的代码逻辑、该功能是否为独立的功能模块、模块设计的区别。因此，各功能模块的设计差异可以有效地反映系统结构设计的异同。

本文涉及的系统设计技术见表1，依据框架在业界的受欢迎程度和影响力，选了20款开源框架和3款基准测试平台，从系统结构设计层面对比分析各设计技术的差异，并重点关注涉及的关键功能模块。模块的设计对系统的架构和工作原理具有重要影响，因此，下文将对功能模块进行详细对比。

1.1　FL开源框架功能模块设计

FL开源框架各功能模块设计见表2，对比了上述20个开源框架的系统内相关功能模块设计。其中，FedNLP作为一个子框架，已整合进了FedML项目的整体架构中。这种整合意味着FedNLP的能力和特性已经与FedML项目的核心功能相融合，共同构成了一个结构协同的综合框架。因此，在模块设计的对比中，本文不再单独区分FedNLP和FedML，而是作为一个整体项目进行分析。对于FL框架而言，采用低耦合的模块设计可以增加系统的可靠性，降低系统学习门槛。需要注意的是，某些框架的部分功能模块在表2中附有备注，这些内容是该项评价的影响因素。〇表示难以界定的评价项，这是由于相关功能尽管已经模块化实现，但仍不够完整或依赖于其他项目的能力。

模型库、数据集库、联邦算法库、隐私算法库和通信协议库的设计直接影响开发人员使用框架的难易程度。完善的设计可以便捷地添加新模型、数据集或算法等到系统中，或供调用。安全组件的良好设计有利于框架的稳健性。网络和集群相关模块以及工业相关功能模块使框架在工业生产环境下的开发、搭建和维护更加便捷。任务调度及生命周期管理器的设计影响开发人员定制FL流程的方式和程度。用户界面模块和联邦任务监控组件的独立设计为框架的接入形式提供了更多可能性，良好的设计使框架可以与各类软件或系统进行无缝整合。框架设计方向还包括云端支持，云端支持的存在与否也是评价框架可扩展性的重要因素。基于上述内容和表2中的结果，可对比框架在这些方面设计成熟度的显著差异。

在表2中，FATE和FedML框架具有最高水平的模块化程度，这是因为它们具备最全面、独立且丰富的系统功能模块。FEDn通过其独特的网络编排机制，拥有多个独立的通信服务，尽管没有独立的网络结构定义组件，但仍然能够构建复杂的网络结构。FedLearner面向广告和推荐行业，因此联邦任务的调度和生命周期管理任务由平台承担，要定制其他场景下的联邦训练任务会比较困难。9NFL的系统设计耦合度较高，功能模块数量较少，同样

表1　　本文涉及的系统设计技术

系统设计技术类型	名称	开发团队
FL开源框架	FATE[12]	微众银行
	FederatedScope[13]	阿里巴巴达摩院
	PaddleFL[19]	百度
	Mindspore Federated[10]	华为
	PrimiHub[20]	原语科技
	SecrectFlow[21]	蚂蚁隐语
	FedLearner[22]	字节跳动
	FedLab[23]	SMILELab
	Rosetta[24]	矩阵元
	Fedlearn-Algo[25]	京东金融
	9NFL[26]	京东九数
	FedML[14]	FedML-AI
	Flower[9]	Adap
	TFF[27]	Tensorflow
	PySyft[28]	OpenMined
	OpenFL[29]	Secure Federated AI
	FEDn[30]	Scaleout Systems
	APPFL[31]	APPFL
	FedNLP[11]	FedML-AI
	FLSim[32]	Facebook（Meta）
FL基准测试平台	LEAF[33]	TalwalkarLab
	FedScale[15]	SymbioticLab
	FedEval[34]	Di Chai

表2 FL开源框架各功能模块设计

框架	模型库	数据集库	联邦算法库	隐私算法库	通信协议库	安全组件	网络通信路由管理组件	网络结构定义组件	集群管理组件	与工业相关功能模块	任务调度及生命周期管理器设计	用户界面	联邦任务监控组件	云端支持
FATE	√	√	√	√	√	√	√	×	√集群节点、流量、模型管理	服务治理，流量控制，异步事件，异常处理，持久化组件	多参与方任务调度器	√	√可视化用户界面	FATE-Cloud可构建和管理工业级FL云服务
Federated Scope	√	√	√	√	×	×	√	×	×	服务治理，流量控制，异步事件	消息驱动	×	√	×
PaddleFL	√	√	√	√	×	×	√	×	√	弹性调度	联邦任务生成器（FL-job-generator）	×	√	支持云端联邦服务器（FL-Server）
Mindspore Federated	√依赖于MindSpore计算框架		√	√	×	×	√云侧	×	√	弹性伸缩负载均衡，容灾能力	云侧FL调度器	×	√	云侧在FL任务中承担联邦服务器角色
PrimiHub	×	×	√	√	×	√	√	×	√	异步事件，原语科技，还提供了企业版以满足更丰富功能需求	调度器	√	√通过PrimiHub-webconsole组件提供了联邦任务可视化监控功能	×
Secrect Flow	×	×	√	√	×	×	○基于Ray框架管理	×	○基于Ray框架管理	×	工作流层，负责管理与协调Secret-Flow系统的整个工作流程	×	√	×
Fed Learner	×	×	√	√	×	×仅闭源版本支持	√	×	√	×仅闭源版本支持	平台侧控制	×	√Web站点进行训练任务管理和指标监控	云原生部署方案
FedLab	√	√	√	×	×	×	√	×	√	异步事件	训练器（trainer）	×	√	×
Rosetta	×	×	×	√	×	×	√	×	×	×	调度器	×	√	×

续表2

框架	模型库	数据集库	联邦算法库	隐私算法库	通信协议库	安全组件	网络通信路由管理组件	网络结构定义组件	集群管理组件	与工业相关功能模块	任务调度及生命周期管理器设计	用户界面	联邦任务监控组件	云端支持
Fedlearn-Algo	√依赖于开源项目Datasets[35]	√	√	×	×	×	√	×	×	×	集成在Server中	×	√	×
9NFL	×	×	×	×	×	×仅闭源版本支持	√	×	√	异步事件，集群资源管理与调度模块	整体调度与转发模块	×	√	×
FedML（FedNLP）	√	√	√	√	√	√	√	√	√	异步事件，FedML开放平台	FedML-core包含的训练引擎	√	√可视化用户界面	公有云聚合服务器，私有云Docker部署支持
Flower	√	√	√	×	×	×	√	×	√	网络扰动，跨平台工作负载	集成在Server中	×	√能耗监控	支持云端部署，模拟云中带宽限制
TFF	√	√	√	×	×	×	√	×	×	异步事件	TFF执行层	×	√	×
PySyft	×	×	√	√	×	×	√	√	√	×	调度器	×	√FALCON监控平台	×
OpenFL	×	×	√	×	×	×	√	×	×	×	联邦任务分配器	×	√	×
FEDn	√	√	√	×	√	○ Reducer-Combiner网络的端口安全设计	√多个独立的gRPC服务器	×	√	异步事件	集成在降维器（reducer）中	√	√可视化用户界面	支持云端部署
APPFL	×	×	√	×	√	×	√	×	×	×	训练引擎（trainer）	×	√	×
FLSim	×	×	√	×	×	×	√	×	×	×	训练器（trainer）	×	√	×

面向电商推荐和广告推荐行业这一特定领域，并且针对性地设计了部分与工业生产相关的功能，如集群资源管理与调度模块。另外，FedLearn-Algo 和 SecretFlow 的部分功能模块依赖于其他项目实现。这是因为搭建 FL 系统需要涉及通信、存储、计算等多个方面的技术支持，部分框架将工作重点放在了特定技术维度上。因此，这些框架通过整合其他框架来提供完整的解决方案。各框架之间的差异可以通过表 2 中的信息进行直观对比。

1.2　FL 基准测试平台设计差异

基准测试平台需要满足实验人员快速搭建 FL 系统的需求，并可以便捷地调用模型、数据集和基线算法。LEAF、FedScale 和 FedEval 都设计了模型库、数据集库和联邦算法库，都具备 FL 过程中任务指标监控机制，但预置内容的丰富程度存在差异。另外，FedScale 额外设计了一个攻击策略脚本库，旨在提供可供调用的功能，用于评估 FL 算法和系统的安全性能。将在后续章节中对基准测试平台中各模块预置内容丰富程度进行详细对比。

2　CPIoT 技术视角对比分析

除了宏观上的结构差异，为了更直观地剖析对比各系统面向 CPIoT 具体技术实现的设计差异，本节对选取的主流开源框架和基准测试平台在多技术维度上进行对比和总结，直观地从对比结论中获得使用、学习和开发这些框架或平台的有价值信息。

2.1　FL 开源框架对比分析

面向选取的 20 个开源框架，分别从 FL 类型与计算范例支持、算法丰富度、隐私保护策略、工业与学术领域适用程度、通信能力、开发语言和深度学习框架支持、部署方式与硬件支持 7 个主要 CPIoT 系统基本技术视角展开对比，为领域学者和研究人员提供系统、全面的横向技术对比结果。需要注意的是，前 11 个框架由国内团队开发，余下的 9 个框架由国外团队开发。FedNLP 已作为子框架整合至 FedML 项目架构之中，因此在 FL 类型与计算范例支持、部署方式与硬件支持等部分视角下依然将二者视为同一项目。

2.1.1　FL 类型与计算范例支持

本节针对上述开源框架进行了调查、统计，并对比了它们在 FL 类型、计算范例和分布式网络架构支持方面的情况。这些技术特点可以衡量这些框架对不同 CPIoT 环境的支持程度。框架 FL 类型与计算范例支持情况对比见表 3，其中，大多数国内 FL 框架都同时支持横向联邦和纵向联邦，而大多数国外 FL 框架仅支持横向联邦。此外，多数框架提供了独立仿真训练和分布式训练的计算范例。同时，一些框架还支持移动设备训练，这对于测试移动设备训练场景下的 FL 算法性能非常有帮助，包括车联网场景[36-37]。在架构方面，大多数 FL 框架都支持中心架构，少数框架提供了去中心架构的支持。另外，一些框架支持分层联邦[38]、混合联邦[39]等场景。需要注意的是，目前只有极少数的框架对联邦迁移学习提供了相应的支持。

2.1.2　算法丰富度

FL 框架内置的 FL 算法通常能够方便开发者快速搭建 FL 系统，并且在学术研究领域，更多内置算法可作为基线算法，用于对比新算法的性能。因此，本节进一步对比了上述框架中预置的 FL 算法和支持的机器学习类型。框架算法丰富度对比见表 4，可以看出，各框架内置的 FL 算法种类和机器学习类型的支持情况差异较大。在 FL 算法方面，FedAvg、FedProx、FedOpt 等常见算法得到了广泛支持。结果显示，一些框架仅内置了 FedAvg 算法，例如，PrimiHub、FedLearner、PySyft、FEDn、APPFL；而 Rosetta 和 9NFL 没有提供预置的 FL 算法，需要开发者自行实现；其余框架都预置了两个或更多不同的 FL 算法供开发者调用，其中，FATE、FederatedScope、FedLab、FedML 和 Flower 提供的 FL 算法丰富度相对较高。另外，大多数框架支持逻辑回归、线性回归、神经网络等常见的机器学习类型，值得注意的是，PySyft 框架目前仅支持神经网络，不兼容逻辑回归、梯度提升决策树等传统方法。

在选择 FL 框架时，考虑框架内置的算法支持是至关重要的。综合考虑框架所支持的 FL 算法种类、适用的机器学习类型，以及算法性能和扩展性等因素，可以选择适用于特定应用场景的最佳框架。

2.1.3　隐私保护策略

隐私保护机制是 FL 算法在实现工业生产环境落地之前必须具备的安全保障，许多学者都在探索具备更高性能和更高安全性的隐私保护策略。为了满足基于框架搭建的 FL 系统对基本或工业级隐私保护能力的需求，并支持新兴隐私保护策略与传统

表3　　框架FL类型与计算范例支持情况对比

框架名称	联邦类型				计算范例			架构	
	横向联邦	纵向联邦	联邦迁移学习	其他	独立仿真训练	分布式训练	移动设备训练	中心	去中心
FATE	√	√	√	×	√	√	×	√	√
Federated Scope	√	√	×	×	√	√	×	√	×
PaddleFL	√	√	√ 由PFM方案间接支持	×	√	√	× 但可模拟实现	√	×
Mindspore Federated	√	√	×	×	×	√	√	√	×
PrimiHub	√	√	×	×	√	√	×	√	×
SecrectFlow	√	√	×	混合联邦	√	√	×	√	×
FedLearner	√	√	×	×	√	√	×	√	×
FedLab	√	√	×	×	√	√	×	√	×
Rosetta	√	√	×	×	√	×	×	√	×
Fedlearn-Algo	√	√	×	×	√	√	×	√	×
9NFL	√	√	×	×	√	√	×	√	×
FedML（FedNLP）	√	√	×	分层联邦	√	√	√	√	√
Flower	√	×	×	×	√	√	√	√	×
TFF	√	×	×	×	√	√	×	√	×
PySyft	√	×	×	×	√	√	×	√	×
OpenFL	√	×	×	×	√	√	×	√	×
FEDn	√	√	×	分层联邦	×	√	×	√	×
APPFL	√	×	×	×	√	√	×	√	×
FLSim	√	√	×	×	√	√	×	√	×

表4　　框架算法丰富度对比

框架名称	FL算法	机器学习算法
FATE	FedAvg，FedProx，D2C，P2P，SecureBoost，SecureSum，PFedMe等	逻辑回归、线性回归、泊松回归、神经网络等
FederatedScope	FedAvg，FedProx，FedOpt，FedBN，pFedMe，Ditto，FedEM等	逻辑回归、线性回归、神经网络等
PaddleFL	FedAvg，DPSGD，SecAgg	逻辑回归、线性回归、神经网络等
Mindspore Federated	FedAvg，FedProx，FedCM	逻辑回归、线性回归、神经网络等
PrimiHub	FedAvg	逻辑回归、分类任务、神经网络等
SecrectFlow	FedAvg，FedProx，FedSCR，FedSTC	逻辑回归、神经网络等
FedLearner	FedAvg	逻辑回归、线性回归、神经网络等
FedLab	FedAvg，FedProx，FedDyn，q-FFL，FedNova，Ditto等	逻辑回归、线性回归、神经网络等
Rosetta	无	逻辑回归、神经网络等
Fedlearn-Algo	FedAvg	逻辑回归、神经网络等
9NFL	无	逻辑回归、线性回归、神经网络等
FedML（FedNLP）	FedNova，FedGKT，FedAvg，FedNAS，FedSEG，FedOpt等	逻辑回归、线性回归、神经网络等
Flower	FedAvg，FedRrox，QFedAvg，FedOpt等	逻辑回归、线性回归、神经网络等
TFF	FedAvg，FedSDG等	逻辑回归、神经网络等
PySyft	FedAvg	目前仅支持神经网络
OpenFL	FedAvg，FedProx，FedOpt，FedCurv	逻辑回归、线性回归、神经网络等
FEDn	FedAvg	逻辑回归、线性回归、神经网络等
APPFL	FedAvg	逻辑回归、线性回归、神经网络等
FLSim	FedAvg，FedSGD，FedAdam	逻辑回归、线性回归、神经网络等

机制的性能对比，一些框架内置了一些隐私保护机制供开发者调用以快速实现。框架隐私保护策略对比见表5，可以看出，不同FL框架的隐私保护策略方案各不相同。其中，差分隐私是主要隐私保护策略方法中框架支持最广泛的一类机制，可以在一定程度上保护用户的隐私。除此之外，一些框架还支持特定的差分隐私算法，如FederatedScope框架支持特征向量机制、分段机制、直方图机制等。在多方安全计算机制方面，一些框架如FATE、PaddleFL、PrimiHub等支持多种多方安全计算协议，尤其是FATE内置的多方安全计算算法非常丰富。在这些框架中，同态加密算法也得到了广泛的支持，且其支持程度近似于多方安全机制，其中，Paillier算法是最受支持的同态加密方法。相比之下，可信执行环境（TEE, trusted execution environment）方法是被支持最少的隐私保护策略，只有PrimiHub框架提供了相关支持。

在横向对比结果中，可以看出，FATE、FederatedScope、PaddleFL、PrimiHub、Rosetta、PySyft等框架在隐私保护策略方面为开发者提供了更丰富的选择。相比于国外框架，国内的FL框架普遍为除差分隐私之外的隐私保护策略提供了更好的支持。需要注意的是，一些框架仍然没有内置特定的隐私保护策略，如FEDn框架。因此，在使用这些框架构建FL系统时，开发者需要自行实现隐私保护策略。

表5　　框架隐私保护策略对比

框架名称	差分隐私	多方安全计算	同态加密	TEE	其他
FATE	支持Laplace机制、指数机制、特征向量机制、分段机制、直方图机制	支持SPDZ（secretShare MPC protocol）、OT（oblivious transfer）、DH（diffne hellman key exchange）、Feldman Vss（feldman verifiable secret sharing）协议	支持Paillier和RSA同态加密方案	×	–
FederatedScope	灵活的DP API	提供附加秘密共享的基本功能	支持Paillier等加密方案	×	预置了攻击策略
PaddleFL	DP-SGD算法	支持ABY3[40]三方安全计算协议和PrivC[41]两方计算协议	×	×	–
Mindspore Federated	噪声方案和SignDS方案	支持基于多方安全计算的安全聚合方案	×	×	–
PrimiHub	DP-SGD算法	支持ABY3、crypTFlow2、cheetah等多方安全计算算法	支持Paillier加密方案	√	支持PIR、PSI和联邦特征工程
SecrectFlow	噪声机制方案	支持SPDZ协议	支持Paillier、BFV、CKKS、TFHE、OU方案	×	–
FedLearner	×	支持基于秘密共享的多方安全计算	支持Paillier加密方案	×	–
FedLab	×	×	×	×	–
Rosetta	×	支持SecureNN三方安全协议，集成高效的零知识证明协议Mystique	×	×	–
Fedlearn-Algo	×	×	×	×	–
9NFL	×	×	×	×	–
FedML（FedNLP）	DP-SGD算法	基于秘密共享的多方安全计算	支持RSA加密方案	×	–
Flower	DP-SGD算法	×	×	×	–
TFF	DP-SGD算法	×	×	×	–
PySyft	DP-SGD和PATE差分隐私	支持SPDZ	支持CKKS加密方案	×	–
OpenFL	支持来自Opacus的多种差分隐私策略	×	×	×	–
FEDn	×	×	×	×	–
APPFL	支持Laplace机制	×	×	×	–
FLSim	噪声机制方案	×	×	×	–

2.1.4　工业与学术领域适用程度

针对上述框架现阶段在工业生产和学术研究领域中的支持和适用情况，对上述框架进行了调研和评估。框架工业与学术领域适用程度对比见表6，在表6的备注项中标注了影响其支持情况判断的一些特点。在11个国内框架中，有8个框架支持学术研究，同时也有8个框架支持工业生产。而在9个国外框架中，所有框架都支持学术研究，但仅有4个框架支持工业生产。通过对比可以看出，在学术领域的支持程度上，国内和国外的FL框架相当。然而，在工业领域的支持程度上，国内框架稍占优势。国内的FATE、PaddleFL、Mindspore Federated等框架在工业生产领域得到了广泛的应用和支持。而国外的一些框架，如FedML，正逐渐向工业生产方向发展。Flower、TFF、OpenFL、FEDn等框架主要用于学术研究。

表6　框架工业与学术领域适用程度对比

框架名称	工业/学术		
	工业	学术	备注
FATE	√	√	支持异步聚合，侧重工业领域
Federated-Scope	√	√	–
PaddleFL	√	√	可绘制消息传递有向图
Mindspore Federated	√	×	负载均衡和容灾能力
PrimiHub	√	√	PrimiHub开源版、PrimiHub企业版
SecrectFlow	√	√	工业领域处于起步阶段
FedLearner	√	×	字节跳动面向广告和推荐行业开发
FedLab	×	√	–
Rosetta	×	√	Rosetta框架在FL领域的发展仍在初期
Fedlearn-Algo	×	√	–
9NFL	√	×	容灾、高吞吐、高可用性和高性能特性
FedML	√	√	逐渐适用于工业生产领域，支持从本地开发环境到生产环境的快速迁移
Flower	×	√	–
TFF	×	√	–
PySyft	√	√	–
OpenFL	×	√	–
FEDn	√	√	Reducer-Combiner网络设计结合了企业安全功能，但主要被用于学术研究
APPFL	×	√	–
FedNLP	√	√	作为FedML的NLP子框架投入学术研究和工业生产应用
FLSim	×	√	–

在表6的备注项中，一些框架具有影响其是否适用于工业或学术领域的特点。例如，FATE框架支持异步聚合，Mindspore Federated和9NFL等框架具有面向工业生产的容灾和高性能特点，而FedNLP专注于自然语言处理领域，并且可以依托FedML框架实现在工业级程序中的部署。这些特点为开发者提供了关键的信息，帮助他们快速了解不同FL框架在各领域的适用情况，并根据任务需求选择合适的框架来构建FL系统。因此，开发者可以根据自身需求更好地选择适用于特定领域的框架来搭建FL系统。

2.1.5　通信能力

针对上述框架，进一步对它们的通信能力进行了比较，包括是否支持自定义网络拓扑、支持的通信后端以及是否支持FL网络拥塞模拟3个方面，以帮助开发者直观地了解这些框架之间的通信能力差异。框架通信能力对比见表7，通过比较结果可以看出，在上述框架中，除了FedML、PySyft和FedNLP，其他FL框架都不支持自定义网络拓扑。

表7　框架通信能力对比

框架名称	网络拓扑自定义	通信后端	FL网络拥塞模拟
FATE	×	gRPC，HTTP/HTTPS	×
FederatedScope	×	gRPC	×
PaddleFL	×	gRPC，Gloo	×
Mindspore Federated	×	TCP，HTTP	×
PrimiHub	×	gRPC	×
SecrectFlow	×	gRPC	×
FedLearner	×	gRPC	×
FedLab	×	Gloo	×
Rosetta	×	需要开发者自行实现	×
Fedlearn-Algo	×	gRPC	×
9NFL	×	gRPC	×
FedML	√	MQTT+S3，MPI，NCCL，MQTT，gRPC，PyTorch RPC等	×
Flower	×	gRPC	√
TFF	×	gRPC	×
PySyft	√	gRPC	×
OpenFL	×	gRPC	×
FEDn	×	gRPC，HTTP/HTTPS	×
APPFL	×	gRPC，MPI	×
FedNLP	√	MPI，RPC，MQTT	×
FLSim	×	Gloo	×

此外，在网络拥塞模拟方面，只有Flower提供了网络拥塞模拟的支持。在通信后端支持方面，gRPC通信协议是被支持最广泛的，部分框架还支持多种通信后端供用户选择。其中，FedML对通信后端的支持种类最广泛，该框架支持MQTT+S3、MPI、NCCL、MQTT、gRPC、PyTorch RPC等多种通信协议。

2.1.6 开发语言和深度学习框架支持

框架作为便捷搭建和开发FL系统的工具，研发人员在开发和搭建系统时需要了解框架的编程语言类型以及框架底层支持的深度学习框架类型。这可以帮助开发者更好地理解框架的搭建逻辑，以便掌握框架的兼容性，并更好地使用和调试框架。框架编程语言和深度学习框架支持情况对比见表8，可以看出，上述框架都支持使用Python编程语言进行开发，其中，FATE、Flower和FEDn框架还提供了多种开发者语言的支持。在底层深度学习框架支持方面，PyTorch和TensorFlow是目前主流的深度学习框架，在FL框架中得到了广泛的支持。而一些框架（如PaddleFL、Mindspore Federated、SecrectFlow）是基于其他计算平台进行搭建的。需要注意的是，FedNLP原生支持PyTorch，但也兼容TensorFlow，因此开发者可以轻松地实现TensorFlow的接入。经过对比，FATE、Flower和OpenFL框架在底层深度学习框架支持方面为开发者提供了更为丰富的选择。综合各框架对编程语言的支持和底层深度学习框架的支持程度，FATE和Flower框架在此维度上具备更广泛的兼容性。

2.1.7 部署方式与硬件支持

本节汇总了上述框架的部署方式与硬件支持，并进行了横向对比。通过对比，可以根据使用需求选择具备特定特点的框架，为FL系统的搭建提供便利。框架部署方式与硬件支持对比见表9，可以看出，上述框架都支持在Linux操作系统上部署，而少数框架还支持在Mac和Windows操作系统上部

表8　　框架编程语言和深度学习框架支持情况对比

框架名称	编程语言		底层深度学习框架支持		
	系统主要开发语言	开发者语言支持	PyTorch	Tensorflow	其他
FATE	Python，Java，C++	Python，Java，Go，Scala，R等	√	√	兼容Spark、EggRoll分布式计算引擎
FederatedScope	Python	Python	√	√	–
PaddleFL	Python，C++	Python	×	×	基于PaddlePaddle（飞浆）计算框架
Mindspore Federated	Python，C++	Python	×	×	基于MindSpore计算框架
PrimiHub	Python，C++	Python	√	×	–
SecrectFlow	Python	Python	×	×	基于高性能分布式框架Ray
FedLearner	Python	Python	×	√	–
FedLab	Python	Python	√	×	–
Rosetta	Python，C++	Python	×	√	–
Fedlearn-Algo	Python	Python	√	×	兼容OneFlow深度学习计算框架
9NFL	Python，C++	Python	×	√	–
FedML	Python	Python	√	×	–
Flower	Python	Python，C++，Java	√	√	支持在TensorFlow、PyTorch、Hugging Face、PyTorch Lightning、MXNet、Pandas、fastai、JAX、scikit-learn等主流深度学习框架和应用场景部署
TFF	Python，C++	Python	×	√	–
PySyft	Python	Python	√	×	–
OpenFL	Python	Python	√	√	兼容Flax深度学习框架
FEDn	Python	Python，Java，C++等	√	×	兼容Keras深度学习框架
APPFL	Python	Python	√	×	–
FedNLP	Python	Python	√	×	可兼容Tensorflow框架
FLSim	Python	Python	√	×	–

表9　　框架部署方式与硬件支持对比

框架名称	部署方式			部署硬件支持						
	本地编译安装	预编译安装	Docker镜像部署	Mac	Linux	Windows	Android	IOS	其他硬件支持说明	大规模部署
FATE	√	√	√支持Docker-compose和Kubernetes	√	√	×	×	×	–	√
FederatedScope	√	√	√	√	√	×	×	×	–	√
PaddleFL	√	√	√支持Kubernetes	√	√	√	×	×	–	√
Mindspore Federated	√	√	×	×	√	×	√	×	–	√
PrimiHub	√	√	√支持Docker-compose	×	√	×	×	×	–	×
SecrectFlow	√	√	√	×	√	×	×	×	–	×
FedLearner	√		√支持Kubernetes	√	√	√	×	×	–	×
FedLab	√	√	√	√	√	√	×	×	–	√
Rosetta	√	√	×	×	√	×	×	×	–	×
Fedlearn-Algo	√	√	×	×	√	×	×	×	–	×
9NFL	√		√支持Kubernetes	×	√	×	×	×	–	√
FedML（FedNLP）	√	√	√	√	√	√	√	√	支持Raspberry PI或NVIDIA Jetson等物联网设备	√
Flower	√	√	×	×	√	√	√	√	支持Rapsberry PI或NVIDIA Jetson等物联网设备	√
TFF	√	√	√	√	√	√	×	×	–	×
PySyft	√	√	√支持Docker-compose和Kubernetes	√	√	√	×	×	–	×
OpenFL	√	√	√	×	√	×计划支持	×	×	–	×
FEDn			√支持Docker-compose	×	√	×	×	×	–	√
APPFL	√	√	√	×	√	×	×	×	–	×
FLSim	√	√	×	×	√	×	×	×	–	√

署。在部署方式方面，本地编译安装是大多数框架都支持的方式，部分框架还支持预编译安装和Docker容器化部署方式。值得注意的是，FEDn框架并不支持本地编译安装，其部署相对复杂，部署时需要在3台主机或虚拟机上分别完成基础支持服务、Combiner和Reducer的部署。在硬件支持方面，大多数框架都支持在普通的服务器或工作站上部署，少数框架还支持在物联网设备上部署。此外，一些框架提供了大规模集群部署支持。

根据具体的使用需求，可以选择具备特定特点的FL框架。例如，在物联网设备上部署的场景中，可以选择支持物联网设备的框架，如FedML和Flower。这些框架提供了适用于物联网设备的轻量级通信和计算方案，可以有效地在资源受限的设备上进行FL。另外，在大规模集群上部署的场景中，可以选择支持集群部署的框架，如FEDn和FATE等框架。这些框架具备高性能分布式计算和通信的能力，可以在大规模集群上高效地执行FL任务。根据具体的部署需求选择合适的框架，可以最大限度地满足系统的要求，并确保FL系统的高效性和可靠性。

2.2 FL 基准测试平台对比分析

针对 3 个主流 FL 基准测试平台，对比了预置模型、预置数据集、预置算法、预置指标、开源协议和项目开源年份 6 个维度，以便于研发人员更直观地了解不同基准测试平台之间的差异，为 CPIoT 应用搭建选择适宜的基准测试平台提供参考。

FL 基准测试平台对比见表 10，可以看出 3 个平台都提供了预置模型，值得关注的是 FedScale 提供了最丰富的模型库，包括 70 多个常用的机器学习模型，其中还包括计算机视觉领域的常用模型。在预置数据集方面，LEAF 提供了 5 个真实数据集和 1 个合成数据集，FedScale 提供了 20 个真实数据集，而 FedEval 提供了 6 个真实数据集。这些数据集涉及图像识别、自然语言处理等多个领域，可以满足不同应用场景的需求。在预置算法方面，FedScale 提供了 FedAvg 和 FedYogi，LEAF 提供了 FedAvg 和 mini batch SGD，而 FedEval 提供了 FedProx、FedAvg、FedOpt、FedSTC、FedSGD、SecAgg、HEAgg 等算法。这些算法适用于 FL 中的模型聚合、参数优化等任务。在预置指标方面，FedEval 提供了相对最全面的评估指标，涵盖了 4 个 FL 基本评估方面中的多种指标和实验中的联邦配置。

综上，FedScale 提供了更加丰富的预置模型和预置数据集，而 FedEval 则具备更全面的预置评估指标。LEAF 作为最早开源的平台，具有更广泛的行业认同，但其预置的模型、数据集和算法相对较少。针对不同的应用场景和需求，开发者可以选择符合需求的 FL 基准测试平台。

3 不同 CPIoT 场景下的 FL 系统搭建和任务构建

为了帮助系统开发人员选择最大程度符合 CPIoT 应用的开发、生产、维护等各方面需求的系统，本节依据现实中常见的 FL 系统应用场景，选取了 4 个代表性的 CPIoT 场景下的 FL 系统搭建任务，根据不同的需求选择合适的系统设计技术并进行完整的系统搭建实验。

3.1 物联网环境下的图像分类任务

随着 IoT 设备中嵌入式摄像机的普及，设备能够收集大量图像数据。图像分类是计算机视觉领域的一个关键任务，它涉及将图像分为不同的类别，例如，识别物体、场景和人脸等。将图像分类任务移至物联网设备之间构建的 FL 系统中进行处理，可以使目标模型更好地服务于该 CPIoT 场景下的样本分类任务。

选择 Flower 框架作为基础，因为其可以轻松地部署在 Raspberry PI 4 等物联网设备上，且可以兼容多种底层深度学习框架，这为不同种类硬件共同构成的 FL 系统的搭建工作提供了极大的便捷。此外，该框架还支持根据需要配置网络扰动进行带宽限制来量化网络环境对学习过程产生的影响，但由

表 10　　FL 基准测试平台对比

	LEAF	FedScale	FedEval
预置模型	CNN，LR，Stacked LSTM	包括 BobileNet，ResNet，RNN，LeNet，VGG，LR 在内的常用机器学习模型，以及大量计算机视觉领域的常用模型，共计 70 多个	LeNet，MLP，Stacked LSTM，LR
预置数据集	FMNIST，Celeba，Shakespeare，Synthetic Dataset	iNature，FMNIST，OpenImage，Google Landmark，Charades，VLOG，Waymo Motion，Europarl，Blog Corpus，Stackoverflow，Amazon Review，CoQA，LibriTTS，Google Speech，Common Voice，Taxi Trajectory，Puffer，Taobao，Go dataset	Celeba，FEMNIST，FedImage，FedMatrix，Sentiment140，Shakespeare
预置算法	FedAvg，mini batch SGD	FedAvg，FedYogi	FedProx，FedAvg，FedOpt，FedSTC，FedSGD，SecAgg，HEAgg 等
预置指标	包括模型评估精度、客户端读写字节数、客户端浮点运算次数在内的性能观测指标	模型精度、损失	涵盖了 FL 的 4 个基本评估方面：隐私、稳健性、有效性和效率，以及各种明确定义的指标和实验中的联邦配置
开源协议	BSD-2	Apache-2.0	无协议开源
项目开源年份	2018 年	2021 年	2020 年

于本实验采用真实物联网设备，因此无须调用网络扰动模块。

实验使用4台Raspberry PI 4和2台NVIDIA Jetson Nano物联网设备，其中一台Raspberry PI 4作为中心服务器，用于处理模型聚合，使用CIFAR-10和SVHN两个数据集分别进行图像分类任务。CIFAR-10数据集是一个广泛用于图像分类任务的彩色图像数据集，包含10个标签类别的RGB彩色图片。SVHN数据集是一个基于实景图片的数字识别数据集，每张图像包含一个裁剪后的数字，其图像来自Google Earth街景图中的门牌号，因此像素信息中包含了自然场景的复杂性，这增加了数字识别的难度。上述两个数据集具有不同的特点，可以帮助评估物联网环境下FL的效果。实验中两个数据集上均采用卷积神经网络（CNN, convolutional neural network）模型和Adam优化器，学习率为0.01，动量为0.9，权重衰减为0，采用FedAvg算法进行模型的聚合。客户端本地数据样本间呈非独立同分布，这样可以更好地模拟物联网环境中设备之间的差异。

在使用Flower搭建系统的过程中，基于PyTorch实现训练管道，为搭载Linux系统的6台物联网设备基于Python语言编写FL应用程序，其中，服务端程序描述了FL任务的生命周期。通过Flower数据集库可以便捷地将上述两个数据集引入系统，然后，根据数据样本的非独立同分布特征，为参与方设备分配本地样本集合。借助Flower策略模块直接调用FedAvg策略算子。系统由作为服务器的设备提供RPC通信服务，其余所有物联网设备在本地局域网中直接使用RPC通信协议与服务器实现通信。在训练过程中，测试精度(Acc)、损失(Loss)等指标将由Flower框架联邦服务器的底层自动完成并输出。

重复实验10次并对每条实验数据取平均值，CIFAR-10和SVHN数据集在Flower框架下的收敛曲线如图2所示。实验结果表明，在CIFAR-10数据集上模型经过300轮次的训练后测试精度达到70%，而SVHN数据集上模型逼近同样的测试精度需要至少380轮次的训练。SVHN数据集包含真实世界中的房屋门牌号码图像，这些图像来自Google街景图像，具有多样性，例如，不同的光照条件、背景和尺度，这些特性可能增加了SVHN数据集的难度，因为模型需要具备对不同条件下的数字字符进行准确分类的能力。上述实验验证了该系统的有效性。

3.2　医学领域问题下的中心化与去中心化联邦搭建

器官定位是许多医学图像分析任务的重要预处理步骤，如图像配准、器官分割和病变检测。本文实验中采用Organ-S/A/CMNIST数据集[42]，其中包含腹部CT图像中11个身体器官的器官定位数据，这些数据集包含大量的2D图像，总计107 731张32 px×32 px尺寸的图像。同时，为了模拟CPIoT下中心化和去中心化FL的效果和性能差异，使用了6台虚拟机作为参与方，实验中所有虚拟机均采用GPU硬件加速计算以提高模型训练和推理的效率。由于FedML框架可以便捷地搭建中心化与去中心化FL系统，内置了不同通信拓扑中的代表性算法，并允许借助Docker容器技术快速地部署到各端，因此该框架较符合本任务需求。FedML MLOps平台可

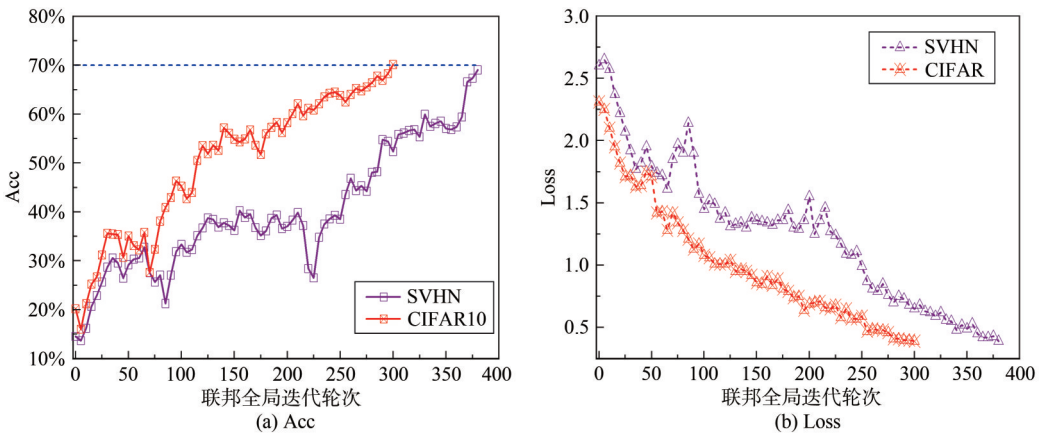

图2　CIFAR-10和SVHN数据集在Flower框架下的收敛曲线

为整个训练流程提供Docker容器自动部署和训练指标可视化追踪，可以在去中心化FL训练中追踪并可视化管理各参与方节点，这有利于实验结果的观察与分析。

在搭建系统的过程中调用FedML Beehive系列API，分布式通信基于gRPC通信协议实现，训练中所有数据包通过底层的FedML-core组件实现传递。首先，在各虚拟机上安装FedML监听服务，然后，通过局域网将各设备绑定到FedML MLOps平台，平台会自动为各虚拟机执行框架部署。对于中心化联邦架构，调用FedAvg算法，并选择其中一台虚拟机作为中心服务器。对于去中心化联邦架构，调用Decentralized FL算法，并为各虚拟机分配计算资源，其中为节点3分配的虚拟机计算缓存资源占比明显低于其他4个节点，但足以维持FL本地训练，目的是观察劣势节点参与去中心化架构训练时与优势节点的性能差异。训练中FL任务生命周期由协调器管理。在Organ-SMNIST、Organ-AMNIST和Organ-CMNIST数据集的预处理阶段，采用Dirichlet方法[43]将样本非独立同分布地划分给各虚拟机。系统采用Adam优化器，学习率设置为0.01。

在进行了200轮次模型训练后，Organ-S/A/CMNIST在FedML的中心化和去中心化架构下的模型精度见表11。可以看出，在中心化架构下模型的训练具有显著效率优势，而去中心化架构中的节点本地模型精度总是低于中心化架构下对应轮次的精度，这是由于去中心化架构的特点导致参与方在一轮训练中通常无法获得所有其他参与方对模型性能的贡献。在去中心化架构中，性能更强的参与方通常可以获得更多来自网络相邻节点对自身本地模型的提升，因此计算资源不足的节点3的模型性能相较其他节点存在明显劣势。

表11　Organ-S/A/CMNIST在FedML的中心化和去中心化架构下的模型精度

轮次	去中心化架构						中心化架构
	节点1	节点2	节点3	节点4	节点5	节点6	
0	22.91	28.91	26.54	28.91	29.68	27.43	36.05
20	29.91	33.54	33.43	33.54	34.21	29.02	46.23
40	30.01	41.56	38.71	51.56	49.83	39.23	53.68
60	35.12	58.29	45.76	57.96	56.28	36.74	58.93
80	31.92	58.94	48.72	59.22	58.47	34.38	63.23
100	41.22	60.47	50.87	61.76	60.84	50.33	69.95
120	50.12	62.32	49.93	63.31	62.15	55.92	73.23
140	54.12	62.19	52.08	65.93	64.31	65.72	79.64
160	55.23	66.45	52.01	73.04	66.22	64.62	83.33
180	60.14	67.34	52.20	79.48	61.02	70.22	85.47
200	64.63	67.73	54.10	78.71	69.12	70.15	86.83

3.3　金融领域客户违约风险问题下的纵向联邦搭建

银行在市场经济中起着至关重要的作用，信用卡是银行的核心业务，然而粗放式管理会导致信用卡客户违约率较高，因此如何有效针对信用卡业务进行风险管理已经成为银行业的热点关注问题之一。Default of Credit Card Clients数据集[44]包含2005年内4—9月期间台湾信用卡客户的还款、客户基本信息、付款历史和账单报表等信息，共计3万条数据，每条数据具体包含24个特征和表示用户10月份是否支付还款的样本标签。

为了模拟CPIoT应用纵向联邦场景下拥有不同特征维度的两个数据参与方A和B，编写脚本程序进一步划分了模拟数据集，在划分过程中，参与方A和B分别获得4、6、8月和5、7、9月的客户数据，以此模拟两家银行各自的财务信息。采用两台物理机分别作为参与方，均搭载Linux操作系统。

由于FATE框架包含多种可调用的纵向FL算法，并能够高效执行工业级FL任务，同时还提供了FATE-Board组件，实现了联邦建模过程中的可视化用户操作面板，因此基于该框架搭建系统。

在搭建纵向FL系统时，分别调用FATE内置的Hetero Factorization Machine和Hetero SVD++作为纵向FL算法。针对该数据集上的二分类任务，采用逻辑回归（LR, logistic regression）模型实现预测。首先，FATE在两台主机上使用Docker容器快速部署，然后，通过可视化用户界面将它们绑定，主机间的通信由框架底层自动实现，而FATE-Flow模块在训练中进行FL任务调度和生命周期管理。最后，在完成模拟数据集划分后，进一步进行数据集预处理，主要包括数据清洗和特征编码[45]。在系统训练过程中，FATE-Board组件将全程提供建模追踪和指标监控服务。学习率设置为0.05，采用Adam优化器。

用户信用卡违约风险预测LR模型在FATE框架下的收敛曲线如图3所示，可以看出，随着200轮次FL全局迭代，在使用两种纵向FL算法执行训练

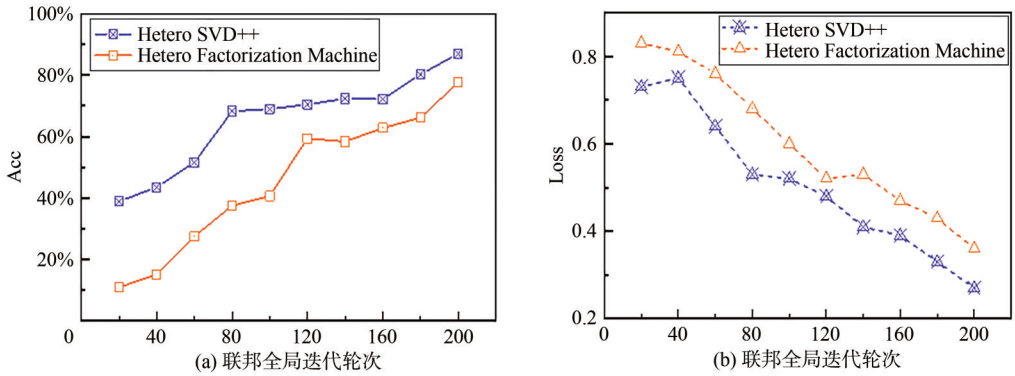

图3　用户信用卡违约风险预测LR模型在FATE框架下的收敛曲线

任务时，训练速度均有稳定合理的表现。在Hetero SVD++算法下，模型预测信用卡客户违约最终的测试精度可达86.94%，这证明该系统在纵向FL环境下执行训练任务可行且有效。

3.4　基于基准测试平台的算法性能评估实验

FL基准测试平台可以快速模拟CPIoT并搭建FL系统以比较和评估不同的FL算法、模型或其他成果的性能。在FL领域，新兴算法的诞生通常需要与基线算法进行对比，并在多种模型上进行性能验证。为了模拟这一场景，引用文献[46]中的FedProx作为待验证的算法。

依据第2.2节的FL基准测试平台选择建议，由于需要在基线算法和多种模型上进行对比，所以选择FedScale基准测试平台。调用预置的FedAvg和FedYogi作为基线算法，并在OpenImage数据集[47]上分别使用ShuffleNet-V2、MobileNet-V2和ResNet18模型进行训练。训练过程中平台会自动输出模型测试精度等指标。实验中训练采用SGD优化器。

基于FedScale平台的算法性能评估实验结果如图4所示，实验结果展示了FedAvg、FedYogi和FedProx这3种FL算法分别采用ShuffleNet-V2、ResNet18和MobileNet-V2模型在OpenImage数据集上进行1 000轮次训练后得到的全局模型测试精度曲线。其中，FedProx的惩罚项proximal（即mu）值增大会加剧本地模型参数更新的约束，实验中，该参数取0.05时，相对于取0.01时，精度显著下降，这是由于过大的约束会导致本地模型无法充分学习局部数据集的特征，从而导致模型的性能下降。最终实验结果表明，MobileNet-V2模型在该任务中可以带来更好的性能提升，而ResNet18相对最慢。在使用相同模型条件下，FedYogi算法的收敛速度最佳，而FedProx在配置了合适的惩罚项参数时性能表现优于FedAvg，反之则可能会带来负面效果。通过上述基准测试，FedProx算法的性能可以得到充分验证和评估。

4　FL系统设计展望

未来FL系统设计领域与CPIoT深度融合的发展前景是备受瞩目的。为了全面把握这一领域的宏观态势，本文通过深入调查和对比分析基于FL框架和基准测试平台的研究成果，细致梳理了该领域面临的挑战以及未来的发展方向。本文期待并鼓励

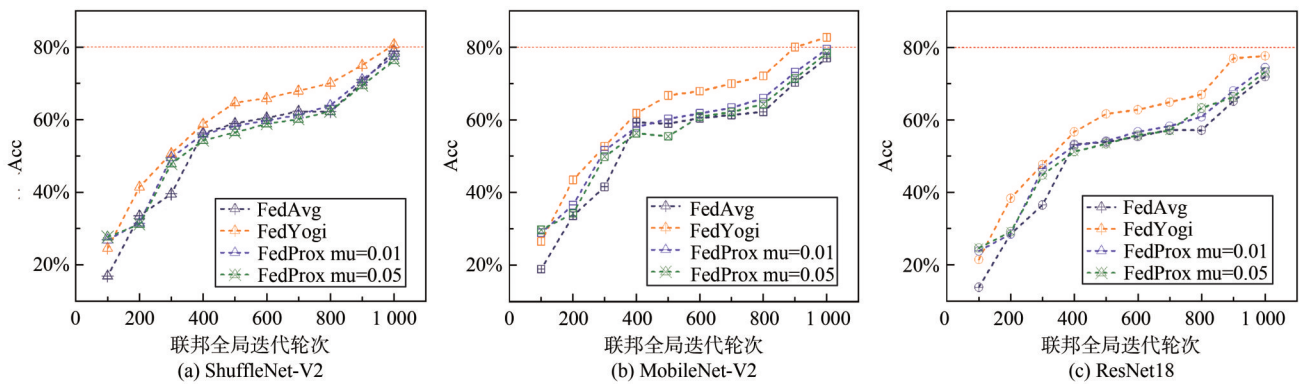

图4　基于FedScale平台的算法性能评估实验结果

更多的研发人员积极应对这些挑战，为推动融合发展做出贡献。

尽管 FL 系统在实践中面临着一系列挑战[48]，然而随着技术的不断进步，在 CPIoT 中实现全面成熟的 FL 系统依然具备潜力。因此，对 FL 的发展进行以下展望。

1) 更好的隐私保护技术。隐私保护一直是 FL 技术的重要研究方向，现有的一些框架或基准测试平台已经内置了一些联邦攻击或防御策略，如 FederatedScope、FedML 和 FedEval，这些防御策略使得 FL 系统具备更出色的隐私保护能力，而攻击策略可以衍生出更多新兴的隐私保护算法。期待未来更加先进的隐私保护技术的出现。

2) 更加统一的 FL 系统开发标准。期待继 IEEE P3652.1 标准化文件[49]之后，FL 系统能够采用更加统一成熟的开发标准，以提高系统的互操作性和可扩展性。这将使不同的 FL 系统之间更加协同和兼容，宏观上也可以使这一领域持续和健康地发展。

3) 更强大的异质性设备支持。期待 FL 系统设计领域在未来开发出具备强大异质性设备兼容能力的框架，可以更好地支持 CPIoT 下异质性设备的部署和计算，实现无须人工配置的异质性设备自适应。这将使 FL 在更广泛的 CPIoT 场景下得到应用。现有的 FedML 和 Flower 框架已经在这一方面做出了一些成果。

4) 更高的模型、算法、数据集、隐私保护策略等算子或组件丰富程度。在 FL 标准化的发展中，期待可以诞生算子或组件定义标准规范，从而使得现有算法、模型等可以兼容任何 FL 框架。进一步，也期待构建出 FL 领域算法和组件通用库，各框架均可以调用基于统一的标准进行开发并完善的成熟组件，消除各开发者针对同一组件的重复开发调试成本。

5) 跨学术领域和工业领域的统一技术桥梁。为了促进学术领域和工业领域之间的需求统一，开发人员希望系统在满足工业生产的高性能、高安全性等要求的同时，也可以满足学术研究的高扩展、高灵活、丰富算子等特点。以 FATE 和 PrimiHub 为代表的部分现有框架在工业生产领域和学术研究领域都已具有很好的兼容性，期待未来这两种场景可以基于该技术桥梁实现需求统一，并且在系统中的实现无须额外配置即可同时兼容。

6) 便捷的新任务定制。未来可以考虑采用更加统一的模块化可扩展设计，并进一步实现算子或组件库和程序接口的统一规范。例如，各框架可以使用统一规范的插件式模型和数据加载器，以便更灵活地添加新的模型和数据集。同时，也可以探索新的 FL 算法和协议，以适应更广泛的 CPIoT 业务中的联邦行为。

7) 简单易用、对开发者友好。应探索更加友好的系统开发模式，提供完善易读的开发者文档或教程，以及具备更加灵活简单的模块化框架设计，使得开发者可以快速上手并进行定制开发，降低上手门槛。未来 FL 系统设计技术的发展需着重提升用户体验和开发者便捷性，消除仿真和部署接口的割裂，子服务之间解耦且紧密配合工作。另外，应不断探索创新算法与技术，以应对需求的不断演进。在此基础之上，应主动推动并严格遵循本领域逐步成熟的标准与规范，确保框架设计的科学性和前瞻性。

5　结束语

算力物联网结合了 IoT 技术和强大的计算能力，为智能决策和数据处理提供了新的可能性。FL 作为挖掘隐私数据中潜在价值、解决隐私保护前提下数据隔离问题的新方向，面向 CPIoT 逐渐体现出强大的研究潜力。CPIoT 场景下 FL 系统设计领域的研究工作旨在促进 FL 框架和基准测试平台的发展与成熟，并最终帮助研发人员搭建健全、稳健、规范的 FL 系统以实现 CPIoT 环境下的 FL 应用。

现有 FL 框架和平台在技术特点层面存在显著差异，且由于尚不存在各技术维度上全面占优的特定 FL 系统设计技术，导致 FL 系统开发人员需要花费大量精力选择系统，因此，本文选择了目前主流的 FL 框架和基准测试平台，并详细地横向对比了各 FL 开源框架、基准测试平台的特点，以剖析系统设计优势与短板。全面探讨各设计技术的优点与不足，并基于 CPIoT 技术维度的对比结果建立了框架与平台选择标准，使 FL 系统开发人员无须花费大量精力对比就可以选择最符合开发、生产、维护等各方面需求的系统。

总体来说，全面详细且多维度的系统结构设计调查和完整深入的对比分析是一项必要且非常有意义的研究工作，既可以推动 FL 系统设计与 CPIoT 深度融合发展，也可以为该领域未来的研究提供参考和思路。

参考文献：

[1] HE S J, YANG X Y. Federated learning-based financial risk early warning model for Baijiu enterprises[J]. Academic Journal of Management and Social Sciences, 2023, 3(3): 133-138.

[2] YANG W S, ZHANG Y H, YE K J, et al. FFD: a federated learning based method for credit card fraud detection[M]. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019: 18-32.

[3] PFITZNER B, STECKHAN N, ARNRICH B. Federated learning in a medical context: a systematic literature review[J]. ACM Transactions on Internet Technology, 2021, 21(2): 1-31.

[4] WANG Z H, FU D Q, ZHANG J W. Logistics data sharing method based on federated learning[C]//Proceedings of the Fifth International Conference on Computer Information Science and Artificial Intelligence (CISAI 2022). SPIE, 2023: 380-385.

[5] ABHISHEK V A, BINNY S, JOHAN T R, et al. Federated learning: collaborative machine learning without centralized training data[J]. Google Research Blog, 2017, 3.

[6] 丁波涛 . 国家数据局成立将有力赋能数字经济高质量发展[J]. 信息资源管理学报, 2023, 13(4): 4-5, 34.
DING B T. The upcoming national data bureau will promote high-quality development of digital economy[J]. Journal of Information Resources Management, 2023, 13(4): 4-5, 34.

[7] IDC. 未来算力推动企业迈向数字化2.0[R]. 2021.
IDC. Future computing power drives enterprises towards digitalization[R]. 2021.

[8] MCMAHAN H B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data[J]. arXiv preprint, 2016, arXiv: 1602.05629.

[9] BEUTEL D J, TOPAL T, MATHUR A, et al. Flower: a friendly federated learning research framework[J]. arXiv preprint, 2020, arXiv: 2007.14390.

[10] CHEN L. Deep learning and practice with MindSpore[M]. London: Springer Nature, 2021.

[11] LIN B Y, HE C Y, ZENG Z H, et al. FedNLP: benchmarking federated learning methods for natural language processing tasks[J]. arXiv preprint, 2021, arXiv: 2104.08815.

[12] LIU Y, FAN T, CHEN T, et al. Fate: an industrial grade platform for collaborative learning with data protection[J]. Journal of Machine Learning Research, 2021, 22(226): 1-6.

[13] XIE Y X, WANG Z, GAO D W, et al. FederatedScope: a flexible federated learning platform for heterogeneity[J]. arXiv preprint, 2022, arXiv: 2204.05011.

[14] HE C Y, LI S Z, SO J, et al. FedML: a research library and benchmark for federated machine learning[J]. arXiv preprint, 2020, arXiv: 2007.13518.

[15] LAI F, DAI Y W, SINGAPURAM S S, et al. FedScale: benchmarking model and system performance of federated learning at scale[J]. arXiv preprint , 2021, arXiv: 2105.11367.

[16] SHAHID O, POURIYEH S, PARIZI R M, et al. Communication efficiency in federated learning: achievements and challenges[J]. arXiv preprint , 2021, arXiv: 2107.10996.

[17] ZHU H Y, XU J J, LIU S Q, et al. Federated learning on non-IID data: a survey[J]. Neurocomputing, 2021, 465: 371-390.

[18] LI Q B, WEN Z Y, WU Z M, et al. A survey on federated learning systems: vision, hype and reality for data privacy and protection[J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(4): 3347-3366.

[19] PaddlePaddle. PaddleFL[EB]. 2023.

[20] CHENG S H. The extended technology stack of Web 3.0[M]. Web 3.0: Concept, Content and Context. Singapore: Springer Nature Singapore, 2024: 57-81.

[21] MA J, ZHENG Y, FENG J, et al. SecretFlow-SPU: a performant and user-friendly framework for privacy-preserving machine learning[C]//2023 USENIX Annual Technical Conference (USENIX ATC 23). 2023: 17-33.

[22] Bytedance. FedLearner[EB]. 2023.

[23] ZENG D, LIANG S, HU X, et al. FedLab: a flexible federated learning framework[J]. Journal of Machine Learning Research, 2023, 24(100): 1-7.

[24] CHEN Y, HUANG G, SHI J, et al. Rosetta: a privacy-preserving framework based on tensorFlow[EB]. 2023.

[25] LIU B, TAN C W, WANG J Z, et al. Fedlearn-Algo: a flexible open-source privacy-preserving machine learning platform[J]. arXiv preprint , 2021, arXiv: 2107.04129.

[26] JD Open Source. 9n-mpc[EB]. 2024.

[27] SUN Z T, KAIROUZ P, SURESH A T, et al. Can you really backdoor federated learning? [J]. arXiv preprint, 2019, arXiv: 1911.07963.

[28] ZILLER A, TRASK A, LOPARDO A, et al. PySyft: a library for easy federated learning[M]. Studies in Computational Intelligence. Cham: Springer International Publishing, 2021: 111-139.

[29] REINA G A, GRUZDEV A, FOLEY P, et al. OpenFL: an open-source framework for federated learning[J]. arXiv preprint , 2021, arXiv: 2105.06413.

[30] Voltaire Edoh I. Federated learning with FEDn for financial market surveillance[M]. Master thesis, Department Faculty Science and Technology, 2022: 58.

[31] RYU M, KIM Y, KIM K, et al. APPFL: open-source software framework for privacy-preserving federated learning[C]//Proceedings of the 2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). Piscataway: IEEE Press, 2022: 1074-1083.

[32] LI L, WANG J, XU C Z. FLSim: an extensible and reusable simulation framework for federated learning[M]. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Cham: Springer International Publishing, 2021: 350-369.

[33] CALDAS S, DUDDU S M K, WU P, et al. LEAF: a benchmark for federated settings[J]. arXiv preprint , 2018, arXiv: 1812.01097.

[34] CHAI D, WANG L Y, YANG L, et al. FedEval: a holistic evaluation framework for federated learning[J]. arXiv preprint, 2020, arXiv:

2011.09655.
[35] LHOEST Q, DEL MORAL A V, JERNITE Y, et al. Datasets: a community library for natural language processing[J]. arXiv preprint, 2021, arXiv: 2109.02846.
[36] WANG X B, WU Q, FAN P Y, et al. Vehicle selection for C-V2X mode 4-based federated edge learning systems[J]. IEEE Systems Journal, 2024, 18(16): 1927-1938.
[37] ZHANG C, ZHANG W J, WU Q, et al. Distributed deep reinforcement learning based gradient quantization for federated learning enabled vehicle edge computing[J]. IEEE Internet of Things Journal, 2024: 1-15.
[38] LIM W Y B, NG J S, XIONG Z H, et al. Decentralized edge intelligence: a dynamic resource allocation framework for hierarchical federated learning[J]. IEEE Transactions on Parallel and Distributed Systems, 2022, 33(3): 536-550.
[39] ZHANG X W, YIN W T, HONG M Y, et al. Hybrid federated learning: algorithms and implementation[J]. arXiv preprint, 2020, arXiv: 2012.12420.
[40] BONAWITZ K, IVANOV V, KREUTER B, et al. Practical secure aggregation for privacy-preserving machine learning[C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2017: 1175-1191.
[41] HE K, YANG L, HONG J, et al. PrivC—a framework for efficient secure two-party computation[M]. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Cham: Springer International Publishing, 2019: 394-407.
[42] XU X A, ZHOU F G, LIU B, et al. Efficient multiple organ localization in CT image using 3D region proposal network[J]. IEEE Transactions on Medical Imaging, 2019, 38(8): 1885-1898.
[43] NG K W, TIAN G L, TANG M L. Dirichlet and related distributions: theory, methods and applications[M]. Chichester: Wiley, 2011.
[44] KAGGLE. Default of Credit Card Clients[EB]. 2017.
[45] 单华玮. 基于机器学习的银行信用卡违约预测研究[J]. 数据挖掘, 2019, 9: 145.
SHAN H W. Research on bank credit card default prediction based on machine learning[J]. Hans Journal of Data Mining, 2019, 9: 145.
[46] LI T, SAHU A K, ZAHEER M, et al. Federated optimization in heterogeneous networks[J]. Proceedings of Machine Learning and Systems, 2020, 2: 429-450.
[47] KUZNETSOVA A, ROM H, ALLDRIN N, et al. The open images dataset V4[J]. International Journal of Computer Vision, 2020, 128(7): 1956-1981.
[48] LI T, SAHU A K, TALWALKAR A, et al. Federated learning: challenges, methods, and future directions[J]. IEEE Signal Processing Magazine, 2020, 37(3): 50-60.
[49] IEEE Guide for Architectural Framework and Application of Federated Machine Learning: IEEE 3652.1-2020[S]. Institute of Electrical and Electronics Engineers, 2020.

[作者简介]

鲁剑锋（1982– ），男，博士，武汉科技大学计算机科学与技术学院教授、博士生导师，主要研究方向为边缘智能、联邦学习和群智计算。

祁盼（2001– ），男，武汉科技大学计算机科学与技术学院硕士生，主要研究方向为联邦学习资源优化和激励机制。

潘林雨（1980– ），男，中国人民解放军91999部队海军专业技术上校，主要研究方向为作战数据应用和智能计算。

李冰（1995– ），女，武汉科技大学计算机科学与技术学院博士生，主要研究方向为联邦学习和群智计算。

曹书琴（1992– ），女，博士，武汉科技大学计算机科学与技术学院讲师，主要研究方向为联邦学习、车联网和交通数据挖掘。

靳延安（1975– ），男，博士，湖北经济学院信息管理学院副教授，主要研究方向为信息智能处理、数据挖掘和智慧养老。