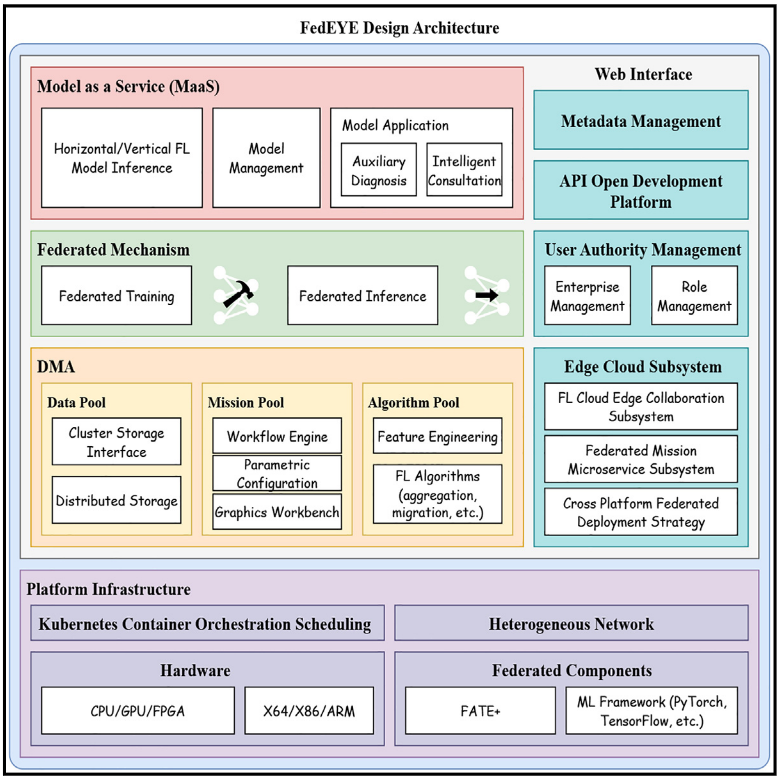
Descriptor

Patterns

FedEYE: A scalable and flexible end-to-end federated learning platform for ophthalmology

Graphical abstract



Highlights

- FedEYE is a scalable and flexible federated learning platform for ophthalmology
- FedEYE empowers ophthalmologists to apply AI techniques to private data
- Guided by design principles, FedEYE offers a full solution for federated learning

Authors

Bingjie Yan, Danmin Cao, Xinlong Jiang, ..., Qian Chen, Zhen Yan, Zhirui Wang

Correspondence

yqchen@ict.ac.cn (Y.C.), daiweiwei@aierchina.com (W.D.)

In brief

FedEYE is an end-to-end federated learning platform tailored for ophthalmologists with limited programming expertise. Its user-friendly interface, modular architecture, and flexible deployment empower ophthalmologists to easily create collaborative projects for tasks like fundus image classification across data silos. FedEYE strives to make federated learning more accessible for ophthalmic research.

Yan et al., 2024, Patterns *5*, 100928
February 9, 2024 © 2024 The Authors.
https://doi.org/10.1016/j.patter.2024.100928

CellPress

Patterns

CellPress
OPEN ACCESS

Descriptor

FedEYE: A scalable and flexible end-to-end federated learning platform for ophthalmology

Bingjie Yan,[1,2,3,8] Danmin Cao,[6,8] Xinlong Jiang,[1,2,3,8] Yiqiang Chen,[1,2,3,7,9,*] Weiwei Dai,[4,5,*] Fan Dong,[1,2,3] Wuliang Huang,[1,2,3] Teng Zhang,[1,2,3] Chenlong Gao,[1,2,3] Qian Chen,[1,2,3] Zhen Yan,[1,2,3] and Zhirui Wang[1,2,3]
[1]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2]Beijing Key Laboratory of Mobile Computing and Pervasive Device, Beijing, China
[3]University of Chinese Academy of Sciences, Beijing, China
[4]Institute of Digital Ophthalmology and Visual Science, Changsha Aier Eye Hospital, Hunan, China
[5]AnHui Aier Eye Hospital, Anhui Medical University, Anhui, China
[6]Aier Eye Hospital of Wuhan University, Wuhan, China
[7]Peng Cheng Laboratory, Shenzhen, Guangdong, China
[8]These authors contributed equally
[9]Lead contact
*Correspondence: yqchen@ict.ac.cn (Y.C.), daiweiwei@aierchina.com (W.D.)
https://doi.org/10.1016/j.patter.2024.100928

THE BIGGER PICTURE Federated learning (FL) enables training machine learning models on decentralized medical data while preserving privacy. Despite growing research on FL algorithms and systems, building real-world FL applications requires extensive expertise, posing barriers for medical researchers. FedEYE, an end-to-end FL platform tailored for ophthalmologists without programming skills, is developed here to easily create federated projects on tasks like image classification. The platform provides rich capabilities, scalability, flexible deployment, and separation of concerns. With user-friendly interfaces and comprehension of underlying mechanisms, FedEYE strives to democratize FL for ophthalmology.

SUMMARY

Data-driven machine learning, as a promising approach, possesses the capability to build high-quality, exact, and robust models from ophthalmic medical data. Ophthalmic medical data, however, presently exist across disparate data silos with privacy limitations, making centralized training challenging. While ophthalmologists may not specialize in machine learning and artificial intelligence (AI), considerable impediments arise in the associated realm of research. To address these issues, we design and develop FedEYE, a scalable and flexible end-to-end ophthalmic federated learning platform. During FedEYE design, we adhere to four fundamental design principles, ensuring that ophthalmologists can effortlessly create independent and federated AI research tasks. Benefiting from the design principles and architecture of FedEYE, it encloses numerous key features, including rich and customizable capabilities, separation of concerns, scalability, and flexible deployment. We also validated the applicability of FedEYE by employing several prevalent neural networks on ophthalmic disease image classification tasks.

INTRODUCTION

Recent advancements in artificial intelligence (AI) and machine learning (ML) have revolutionized various industries, including medical and ophthalmology. The ability to process vast amounts of data has led to the development of powerful algorithms and large deep learning (DL) models that can assist in diagnosing diseases accurately and efficiently, identify potential drug candidates, and personalize patient treatment plans. However, the success of ML models primarily depends on the quality and quantity of the training data. In ophthalmology, data are generally stored in data silos with privacy and policy limitations, such as the General Data Protection Regulation,[1] which was published by the European Parliament and Council of the European Union in 2016 on data protection and privacy, making it hard to train models via centralized training across multiple institutions.

Federated learning (FL) or federated ML (FML) is a promising solution to this issue. It allows multiple institutions to train an ML model collaboratively without sharing data. Instead of relying on a centralized model, institutions can train the models using their data and local machines. The parameters of all locally trained models can be aggregated to create a federated global model. This approach ensures greater data privacy and security, benefiting institutions from more accurate and robust global ML models. One of the earliest works in FL is the federated averaging algorithm proposed by Mcmahan et al.[2] in 2017. In addition to these algorithmic advances, there has been much work on the theoretical analysis of FL, including convergence analysis[3,4] and communication complexity analysis.[5,6] Many researchers have also studied the practical challenges of deploying FL systems in real-world settings, such as the heterogeneity of devices,[7,8] the reliability of wireless networks,[9,10] and the variability of user behavior.[11–13]

As research on FL progresses, more researchers focus on studying FL system architecture. Due to the heterogeneity of FL, unique features of different platforms require the resolution of heterogeneity issues. Therefore, several FL frameworks are designed to address these issues, such as PySyft,[14] OpenFL,[15] FLOWER,[16] Fedlearn-Algo,[17] FedJAX,[18] FedScale,[19] and FederatedScope,[20] among others. Although most FL frameworks mentioned above are still in the simulation stage, the first enterprise-level framework, FATE,[21] and the developing FedML[22] are presently designed for practical applications. FATE[21] is an FL framework based on multi-party collaborative and secure computing that addresses privacy issues and data sensitivity in data sharing. The framework protects data privacy, accelerates training speed, reduces dataset maintenance costs, and enhances model prediction capabilities. FedML[22] is an open FL research library and benchmark that aims to promote the development and fair comparison of FL algorithms.

FL's unique ability to train ML models on decentralized data sources while preserving user privacy makes it a promising approach for a wide range of healthcare, finance, smart cities, and Internet of Things applications. In the medical field, by allowing multiple medical institutions to collaborate and train ML models on their respective datasets without sharing sensitive patient information, FL can help overcome the limitations of data silos and promote knowledge sharing across regions. Many studies have investigated the use of FL in medical subfields,[23–25] including lung cancer,[26–28] pneumonia,[29–31] kidney diseases,[32,33] and even intensive care unit mortality rate prediction.[34] Also, there are many researches in ophthalmology.[35–38]

The utilization of FL has been acknowledged as a promising approach in medical research. However, building an FL application requires significant expertise in computer science and ML, and most of the existing FL systems, which are designed for FL researchers, are not friendly to medical researchers or ophthalmologists and are challenging to use, which can be a barrier for ophthalmologists who may not have extensive training in this research.

Hence, we developed FedEYE, a scalable and flexible end-to-end FL system tailored specifically for medical researchers, particularly ophthalmologists. This platform tackles these challenges by providing a user-friendly and parameterized interface. While ophthalmologists and ophthalmic researchers may not be versed in ML and AI, these technologies enable them to effortlessly launch independent or federated projects with their research collaborators, encompassing tasks such as fundus image classification, segmentation, and more. FedEYE strives to democratize the FL process for ophthalmologists by minimizing the requirement for intricate programming, facilitating comprehension of the underlying mechanism, and providing convenient access to diverse medical systems. The platform adheres to the MLDevOps paradigm,[39] effectively segregating the algorithmic layer from the application layer, simplifying management and maintenance tasks. Additionally, FedEYE supports software and hardware components, enabling its deployment on various devices with flexible configurations.

RESULTS

FedEYE represents a federated edge computing platform meticulously designed for FL, adhering to a set of guiding principles. It harnesses the power of edge computing capabilities, including privacy preservation, prompt processing, and offline computing, to facilitate distributed computing of multi-modal data while upholding robust data security measures. Moreover, FedEYE provides a user-friendly and easily comprehensible web interface, empowering medical professors and researchers to embark on their own federated training research collaborations. Furthermore, in order to cater to the specific research requirements of ophthalmologists, the current platform has already incorporated support for a diverse array of ophthalmic research tasks, encompassing but not limited to tabular data categorization, image classification, text classification, image segmentation, and object detection. This comprehensive coverage ensures that a wide range of ophthalmic research pursuits can be accommodated.

In Table 1, we have conducted a comparative analysis of our FL platform against existing platforms, evaluating them across 13 dimensions and 46 indicators. This evaluation encompasses aspects such as deployment support, privacy protection, data security, availability of built-in ML models, scalability, and more. The privacy protection section encompasses support for methods such as differential privacy (DP), multi-party computing (MPC), and homomorphic encryption (HE). In ML models and benchmarks sections, it includes the support for tree-based models, neural network-based models including regression, convolutional neural network (CNN), recurrent neural network (RNN), and the quantity of common neural network (NN) for computer vision (CV).

In this section, we will introduce FedEYE, including its guiding design principles, detailed platform architecture, key features, and benchmarks.

Design principles of FedEYE

Designing an FL platform for ophthalmology that can accommodate the heterogeneity of ophthalmology data and enable effective collaboration between medical professionals and data scientists presents a significant challenge.

To address this challenge, we propose an FL system FedEYE that is designed with the following principles.

(1) Low-code programming: the aim is to minimize the need for complex programming, making it easier for medical professionals to participate in the FL process and to understand it better.
(2) Accessibility: the platform needs to have OpenAPI, which makes it easy to integrate with traditional medical systems.

Table 1. Comparison with existing platforms/frameworks/libraries

Metrics		Platform/framework/library						
Information	name	FedEYE (ours)	FATE	FedML	FederatedScope	PySyft	TensorFlow Federated	PaddleFL
	version	1.0	1.10.0	0.7.502	0.2.0	0.7.0	0.50.0	1.2.0
FL algorithm	horizontal FL	✓	✓	✓	✓	✓	✓	✓
	vertical FL	✓	✓	✓	✓	✓	✗	✓
	split learning	✓	✓	✓	✗	✓	✓	✓
Benchmarks	regression	✓	✓	✓	✓	✓	✓	✓
	NN	✓	✓	✓	✓	✓	✓	✓
	tree-based model	✓	✓	✗	✓	✗	✗	✗
Documentation	tutorial	✓	✓	✓	✓	✓	✓	✓
	code example	✓	✓	✓	✓	✓	✓	✓
	API detailed document	✓	✓	✗	✓	✓	✓	✓
Deployment support	standalone	✓	✓	✓	✓	✓	✓	✓
	GPU support	✓	✓	✓	✓	✓	✓	✓
	cluster support	✓	✓	✗	✗	✗	✗	✗
	container deployment	✓	✓	✓	✓	✓	✓	✓
	heterogeneous hardware	✓	◗[a]	◗	◗	◗	◗	◗
Privacy and data security	DP	✓	✓	✓	✓	✓	✓	✓
	MPC	✓	✓	✓	✓	✓	✓	✓
	HE	✓	✓	✗	✓	✓	✓	✓
	distributed storage	✓	✗	✗	✗	✗	✗	✗
Built-in DL models	CNN	✓	✓	✓	✓	✓	✓	✓
	RNN	✓	✓	✓	✓	✓	✓	✓
	custom optimizer	✓	✓	✓	✓	✓	✓	✓
	# NN (CV)	24	<10	<15	<10	<10	<10	<10
DL backend	PyTorch	✓	✓	✓	✓	✓	✗	✗
	TensorFlow	✓	✓	✓	✓	✗	✓	✗
	PaddlePaddle	✗	✗	✗	✗	✗	✗	✓
	JAX	✗	✗	✓	✗	✗	✗	✗
	Caffe	✗	✗	✗	✗	✗	✗	✗
Scheduling	task scheduling	✓	✓	✗	✗	✗	✗	✗
	resource scheduling	✓	✓	✗	✗	✗	✗	✗
	task parallelism	✓	✓	✗	✗	✗	✗	✗
Scalability	model/data/algorithm	✓	✓	✓	✓	✓	✓	✓
	service	✓	✗	✗	✗	✗	✗	✗
Logging	task logging	✓	✓	✓	✓	✓	✓	✓
	cluster status	✓	✓	✗	✗	✗	✗	✗
	device status	✓	✓	✓	✓	✓	✓	✓
Maintain	user management	✓	✗	✓	✗	✗	✗	✗
	role management	✓	✗	✗	✗	✗	✗	✗
Completeness	full-stack implement	✓	✗	✓	✗	✗	✗	✗
	low code	✓	✗	✗	✗	✗	✗	✗
	parameters	✓	✗	✓	✗	✗	✗	✗
	algorithms pool	✓	✗	✗	✗	✗	✗	✗
	model pool	✓	✗	✗	✗	✗	✗	✗
	mission pool	✓	✗	✓	✗	✗	✗	✗
	data pool	✓	✗	✗	✗	✗	✗	✗

[a]◗ means not specifically processed or not mentioned.

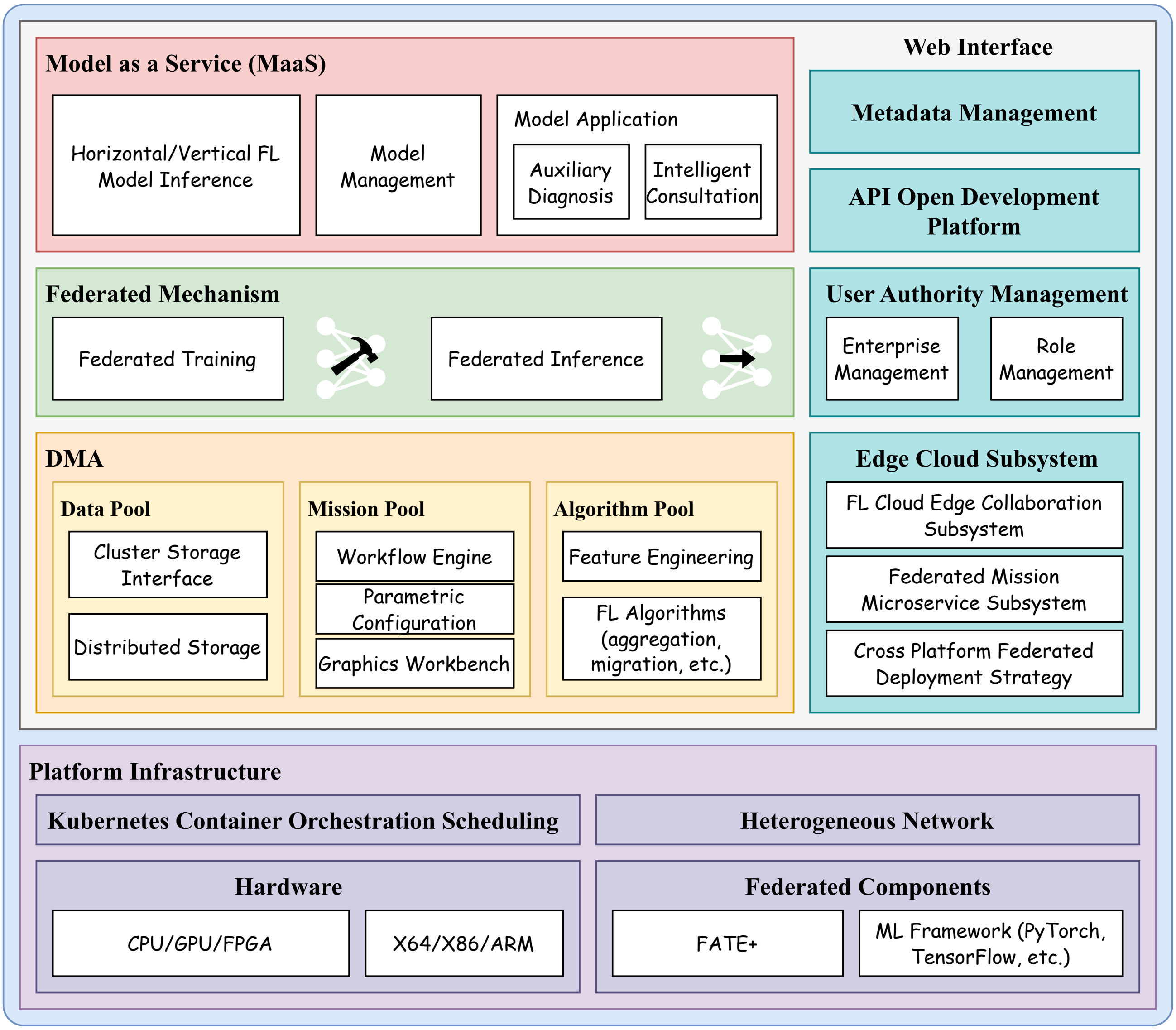


Figure 1. Architecture design overview of FedEYE

(3) MLDevOps[39]: the aim is to separate the work of the algorithm layer from the application layer, which can improve the system's manageability and maintainability.
(4) Heterogeneous hardware support: the architecture should support both software and hardware components, making it possible to run on various devices with heterogeneous hardware configurations.

Detailed architecture design of FedEYE

As shown in Figure 1, FedEYE can be divided into four layers from bottom to top: a platform infrastructure layer; a data, mission, and algorithm (DMA) layer; a federated mechanism layer; and a model as a service (MaaS) layer. The DMA layer, federated mechanism layer, and MaaS layer shall collectively be referred to as the DMA-MaaS framework.

Platform infrastructure

The platform infrastructure layer provides fundamental computing and storage capabilities to the upper layers. This layer is comprised of hardware facilities, containers, networks, and federated components.

FedEYE's cluster management capabilities rely on Kubernetes[40] to provide real-time reporting of the running status of federated devices and services, making it easier for the cloud control console to monitor them. This platform containerizes model services, allowing for hosted models to be pushed to the edge and for integrated, visualized management and flexible scheduling of edge devices and services.

The key design abstracts platform differences through a unified edge runtime. The edges use CPUs, GPUs, and FPGAs for training and inference via scheduling and containers. It allows FL model computation on devices, continuous inference during

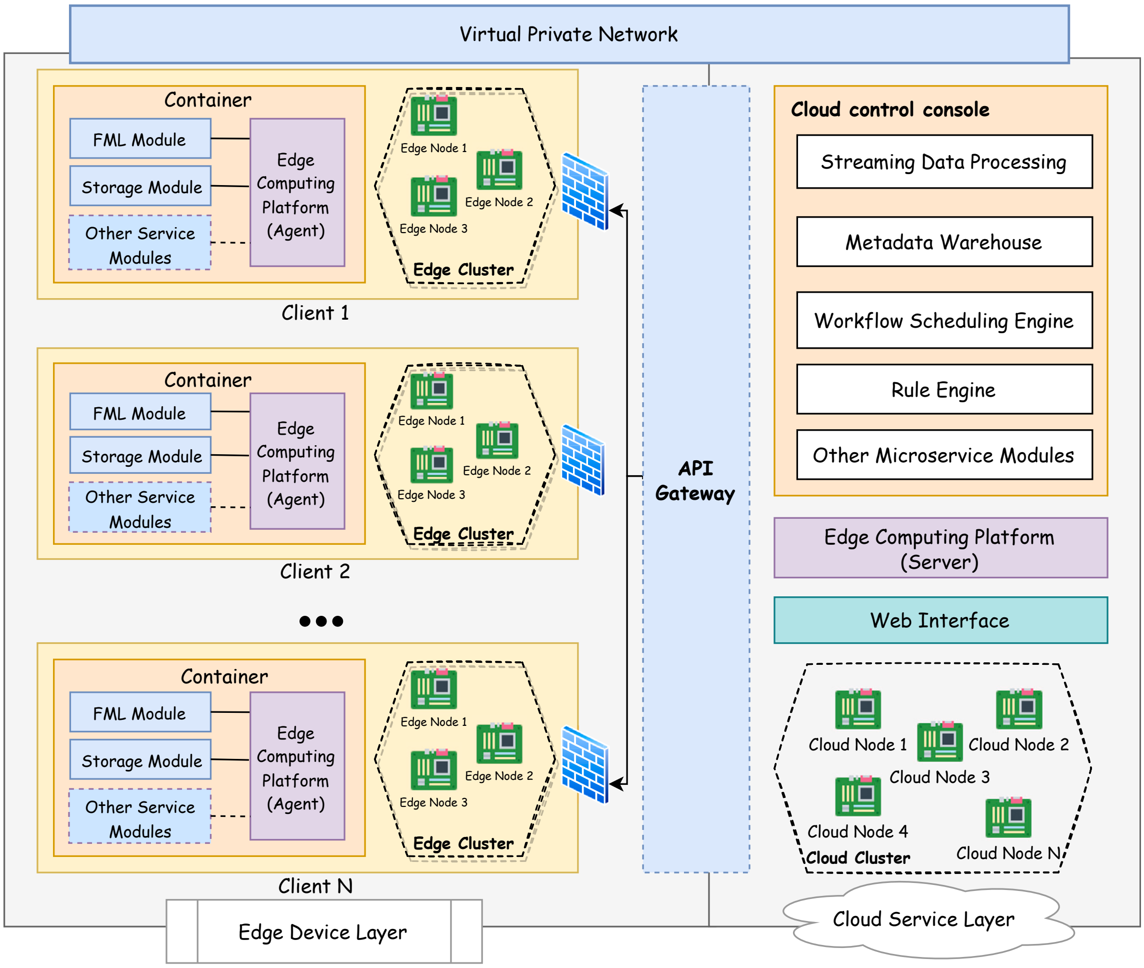


Figure 2. FedEYE deployment with heterogeneous networks

training, and seamless model upgrades. Models and applications are containerized plug-and-play modules. Integrating extended FATE frameworks' containers enables cloud-edge collaboration for FL.

In order to address the challenges posed by heterogeneous networks in real-world deployment scenarios, FedEYE incorporates a cloud service layer and an edge device layer, as depicted in Figure 2. The edge device layer is connected to the cloud server through an application programming interface (API) gateway after passing through a firewall. The cloud service layer includes micro-services such as the cloud control console, the server side of the edge computing platform, and a cloud-based model management platform. The edge device layer comprises client devices participating in FL modeling. Each client device contains an agent side of the edge computing platform and federated components and is deployed in a containerized virtual environment. The network environment of all devices is composed of virtual private networks (VPNs), which can ensure the privacy and security of data transmission and prevent hijacking and attacks. For example, Wire-Guard, which implements encrypted VPNs and is open source, can be used.

DMA-MaaS framework

The DMA layer includes a data pool, mission pool, and algorithm pool, while the MaaS layer mainly provides model services for the model market. To ensure data security and privacy, we store the necessary data, algorithm, and model in a distributed manner across the edge nodes while providing external interfaces for data input and management. The algorithms used include feature engineering and federated algorithms such as aggregation optimization and federated transfer. Upon initiating a task, a parameterized configuration and a graphical interface are provided for display and participation. Once the task is completed, the resulting model can be set to private or added to the model pool for others to use for research and inference. Our platform offers essential model management services, which include model inference for vertical and horizontal FL, model upload and download, and model version management.

We use the federated mechanism layer as the middle layer to connect the DMA layer and the MaaS layer, mainly by combining and orchestrating algorithms, data, and models to build two different types of federated tasks.

Table 2. FedEYE federated learning benchmarks

Dataset	Partition	Model	Accuracy	Recall	Data size
DR	iid	Swin-T[46]	0.9711	0.9653	7901
DR	iid	ViT[47]	0.9714[a]	0.9690[a]	7901
DR	iid	ResNet[45]	0.9099	0.9144	7901
RVO	iid	Swin-T[46]	0.9859	0.9466	5547
RVO	iid	ViT[47]	0.9496	0.9870[a]	5547
RVO	iid	ResNet[45]	0.9910[a]	0.9686	5547
AMD	iid	Swin-T[46]	0.9422[a]	0.9344	8261
AMD	iid	ViT[47]	0.9268	0.9232	8261
AMD	iid	ResNet[45]	0.9072	0.9103	8261
HMCN	iid	Swin-T[46]	0.9989[a]	0.9977[a]	5369
HMCN	iid	ViT[47]	0.9978	0.9971	5369
HMCN	iid	ResNet[45]	0.9985	0.9975	5369
PM	iid	Swin-T[46]	0.9511	0.8721	5852
PM	iid	ViT[47]	0.9552[a]	0.8767[a]	5852
PM	iid	ResNet[45]	0.9532	0.8794	5852
EDDL	iid	Swin-T[46]	0.9607[a]	0.9293[a]	712
EDDL	iid	ViT[47]	0.9438	0.8905	712
EDDL	iid	ResNet[45]	0.9002	0.8783	712
EDDL	non-iid	Swin-T[46]	0.9542	0.7784	712
EDDL	non-iid	ViT[47]	0.9663[a]	0.9459[a]	712
EDDL	non-iid	ResNet[45]	0.9052	0.6327	712
OCT17	iid	Swin-T[46]	0.9897[a]	0.9850[a]	83484
OCT17	iid	ViT[47]	0.9852	0.9791	83484
OCT17	iid	ResNet[45]	0.9706	0.9734	83484
OCT17	non-iid	Swin-T[46]	0.9949[a]	0.9850[a]	83484
OCT17	non-iid	ViT[47]	0.9775	0.9316	83484
OCT17	non-iid	ResNet[45]	0.9865	0.9759	83484

[a]Best results

DR, diabetic retinopathy classification dataset; AMD, age-related macular degeneration dataset; RVO, retinal vein occlusion dataset; HMCN, highly myopic choroidal neovascularization dataset; PM, pathological myopia dataset.

Key features of FedEYE

Leveraging the design principles and architecture of FedEYE, the platform boasts a multitude of key features. These encompass a diverse range of rich and customizable capabilities, clear separation of concerns, scalability, and flexible deployment.

Rich and customizable capabilities

FedEYE is a ML platform that enables both federated and standalone ML tasks across structured data like tabular data as well as unstructured data such as images and text. The platform facilitates secure and privacy-preserving FL by allowing federated training of ML models on sensitive data that remains on users' local devices. Beyond FL, FedEYE also supports independent research tasks on private datasets. The platform can handle a wide variety of tasks, including binary and multi-class classification for tabular data, image and text data, and object detection for image data. FedEYE seamlessly integrates popular model libraries such as timm,[41] torchvision,[42] transformers,[43] and more. Ophthalmic researchers can leverage the capabilities of the platform to engage in numerous federated tasks, such as the classification and diagnosis of fundus images, lesion identification, and analysis of patient morbidity rates, among other scientific collaborators.

We develop a web-based interactive interface that schedules tasks through the edge-cloud subsystem. When creating tasks, we offer a parameterized task configuration method and a graphical interface to construct tasks modularly. This makes it easy for medical staff without solid programming skills.

Clear separation of concerns

Aligned with the design principle of MLDevOps,[39] FedEYE empowers individuals with distinct responsibilities to independently manage various aspects of the platform. For instance, doctors and patients can create inference tasks and complete model applications through the MaaS in FedEYE's web-based interactive interface. Additionally, doctors can utilize the DMA-MaaS framework of the platform to initiate tasks and involve other medical institutions. The maintenance staff of the hospital system can integrate the system and fulfill specific personalized requirements using the FedEYE API open platform. ML algorithm engineers can focus on the algorithm pool and related inference task development, while hardware engineers can concentrate on the infrastructure layer. The platform's operations personnel focus on user, role, metadata management, and platform development.

CellPress
OPEN ACCESS

Table 3. FedEYE independent research benchmarks

Dataset	Client 0	Client 1	Client 2	Client 3	FedEYE FL task
EDDL	0.7569	0.7222	0.7361	0.7222	0.9607[a]
OCT17	0.9566	0.9566	0.9591	0.9542	0.9897[a]

[a]Best results

Scalability

The scalability of the FedEYE platform has two key aspects: the scalability of the cluster nodes and the scalability of the functionalities.

The cluster nodes' scalability of FedEYE is due to using K3s[40] as the underlying tool for managing computing nodes. K3s is a lightweight Kubernetes distribution designed for the Internet of Things and edge computing. K3s minimizes external dependencies, requiring only kernel and cgroup mounting, making it easy to scale computing nodes up or down. When a computing node needs to join the cluster, simply install the K3s agent with one command. When a node becomes unavailable due to a crash or another reason, the platform automatically migrates the applications on the node to other healthy nodes. At the same time, the platform dynamically binds namespaces with computing nodes to ensure that application components of each federated party can be migrated to suitable computing nodes during node scaling.

The functional scalability of FedEYE is due to the micro-service architecture. Micro-service architecture involves breaking down different functional modules of the system into multiple separate services, each independently developed and deployed. Each service runs in its own process, so updates to one service do not affect the running of other services. Because each service deploys independently, it is possible to monitor the resource consumption of each service more accurately and to quickly identify performance bottlenecks between services. Each service is independently developed and can use different programming languages, reducing code conflicts and duplication. This makes the logical processing flow clearer and makes maintenance and expansion easier in the future.

Flexible deployment

FedEYE offers readily available edge computing machines seamlessly connected to the federated platform once the network configuration is complete. Moreover, FedEYE provides a containerized deployment approach facilitated through Docker.[44] Users have the convenience of installing and upgrading the platform components by simply downloading the Docker image.

FedEYE benchmarks

FedEYE provides FL benchmarks on the public and private datasets, and the detailed benchmarks are shown in Table 2. We conducted image classification experiments based on FedAvg using ResNet,[45] Swin-Transformer,[46] and ViT[47] models on each dataset. The experimental results display only the top three highest accuracy rates for each model.

To further illustrate the significance of the platform, we also conduct experiments on the Eye Disease Deep Learning Dataset (EDDL) and the Retinal OCT Images Dataset (OCT17), conducting independent ML experiments on different clients. It can be seen from the results shown in Table 3 that performing federated tasks through the FedEYE platform can achieve better model performance while protecting data privacy.

The detailed description is in the experimental procedures section.

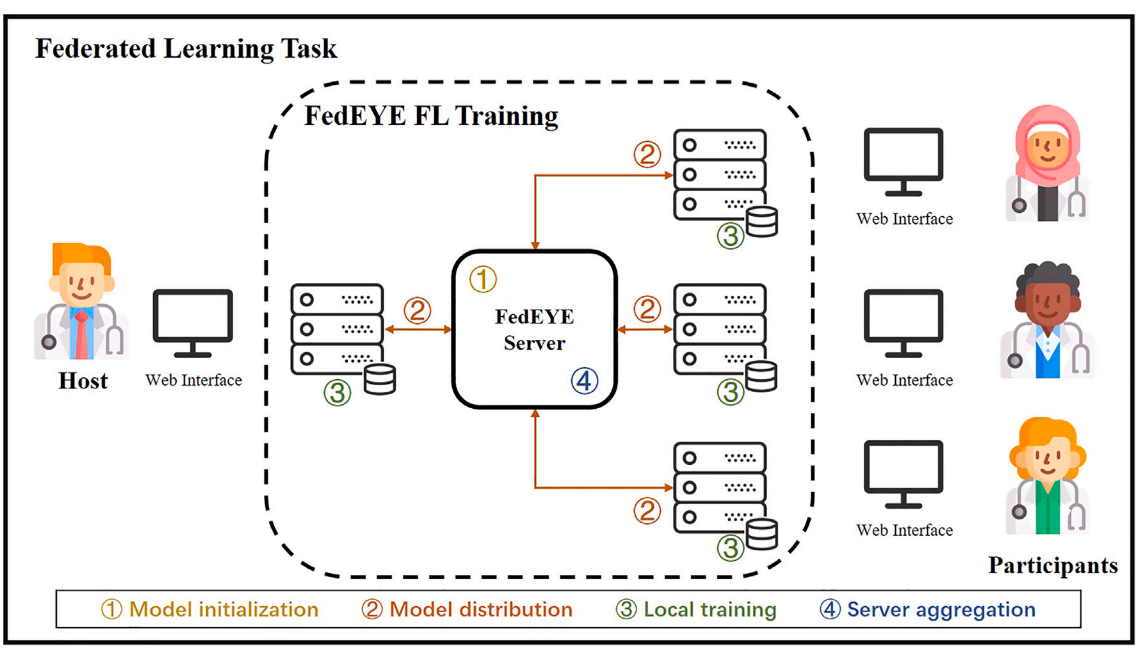


Figure 3. Federated learning process on FedEYE

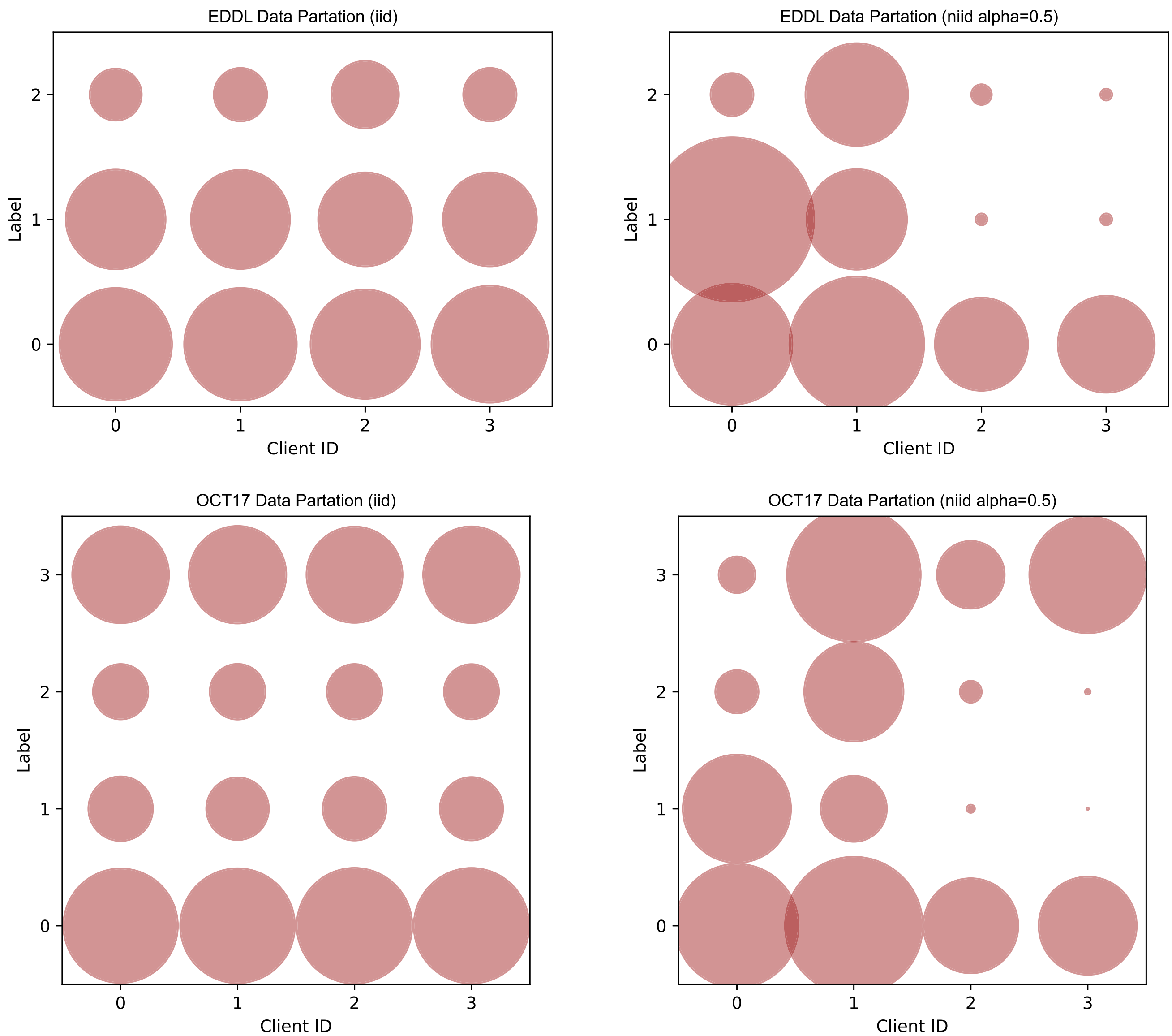


Figure 4. The data partition of EDDL and OCT17
The size of the circles represents the volume of data.

DISCUSSION

The FedEYE platform is designed to address the unique challenges of training ML models in ophthalmology. By providing a user-friendly and scalable system that separates the concerns of data, algorithm, model, and hardware management, we believe that our platform can enable more effective and efficient ophthalmology research while protecting patient privacy. However, FedEYE still has some limitations. We have solely contemplated the training and inference process of the model, whereas the configuration of model parameters necessitates further guidance and expertise. Moreover, data collection and annotation automation remain focal points for future advancements and for refinement of the platform. With the development of platforms such as FedEYE, FL collaborations for ophthalmology research and data can become more accessible to medical professionals and institutions, further advancing the field of medical research.

EXPERIMENTAL PROCEDURES

Resource availability

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Yiqiang Chen (yqchen@ict.ac.cn).

Materials availability

This study did not generate new materials.

Data and code availability

The EDDL[48] dataset is available at the repositories,[48] and the OCT17 can be found at Mendeley Data.[49] All other data reported in this descriptor will be provided by the lead contact upon request. However, we highly recommend contacting the lead contact to access the FedEYE platform and participate in the federated training to use the complete data.

The FedEYE platform can be accessed at https://fedeye.aierchina.com/. All original code for platform and data pre-processing is available at GitHub (https://github.com/beiyuouo/FedEYE) and has been deposited at Zenodo.[50]

Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

Experiment setup

To demonstrate the capabilities of the platform, we conducted an FL experiment on ophthalmic medical images using FedEYE deployed in the real world. We first introduced the definition of the FL problem, experimental settings, platform deployment, and benchmark experimental results.

Problem definition

FL is a distributed ML approach that enables training ML models on decentralized data located on devices like mobile phones or hospital servers without exchanging local private data samples. This allows collaborative model training while keeping data localized and private.

The key steps in FL are

(1) Model initialization: a global model is created on a central server with randomly initialized weights.
(2) Model distribution: the central server distributes the global model back to all clients.
(3) Local training: each client trains the model on local private data and updates the model locally.
(4) Server aggregation: the locally updated models are sent back to the central server. In FedAvg, the server aggregates these models by averaging their weights and creates an improved global model.
(5) Iteration: steps 2–4 are repeated until the global model converges.

The schematic diagram of the FL process is illustrated in Figure 3. The goal is to collaboratively train an accurate model while keeping data decentralized on devices. FL preserves privacy and reduces communication costs. It is well-suited for medical applications where sensitive patient data need to remain localized. The iterative process allows building a shared global model from local insights without sharing the data.

Datasets

We used seven datasets, including two public datasets: the EDDL[48] and the OCT17.[49] We performed both independently and identically distributed (iid) data partition and non-independently and identically (non-iid) data partition based on the Dirichlet distribution (alpha = 0.5) for each dataset. The relationship between the number of data instances and label quantities for each client is depicted in Figure 4. In addition, there are five private datasets belonging to Aier Eye Hospital: the diabetic retinopathy classification dataset, the age-related macular degeneration dataset, the retinal vein occlusion dataset, the highly myopic choroidal neovascularization dataset, and the pathological myopia dataset (note: all of Aier Eye's data were collected from several natural data centers).

Experiments detailed

For the CNN-based model, ResNet[45] employs a learning rate of 1e−4 and conducts training with the Adam optimizer. For the transformer-based model, Swin-Transformer[46] and ViT[47] apply a learning rate of 3e−5 and training using the AdamW optimizer. All models are pre-trained on ImageNet. We utilize a batch size of 8 for the OCT17 dataset, iterating for 20 rounds, while employing a batch size of 2 for other datasets, iterating for ten rounds. Before the training procedure, the platform shall autonomously partition the dataset in an 8:2 ratio, distinguishing between the training and validation datasets. This division serves the purpose of evaluating the model's generalization capability and safeguarding against overfitting.

Deployment detailed

The FedEYE platform is deployed in clusters across five data centers of Aier Eye Hospital, which include Aier Eye Hospital headquarters, Changsha, Wuhan, Chongqing, and Shenzhen.

ACKNOWLEDGMENTS

This work is supported by the Science and Innovation Leadership Plan of Hunan Province of China (no. 2021GK4015), the Youth Innovation Promotion Association CAS, the Natural Science Foundation of China (no. 61972383), and the Hunan Provincial Natural Science Foundation of China (no. 2023JJ70008).

AUTHOR CONTRIBUTIONS

Conceptualization, Y.C., X.J., and W.D.; experiments, B.Y.; resources and data curation, B.Y., W.H., and D.C.; writing – original draft, B.Y.; writing – review & editing, X.J., W.D., Y.C., B.Y., and D.C.; platform model and algorithm support, B.Y., F.D., D.C., T.Z., W.H., Q.C., Z.Y., C.G., and Z.W.; medical and ophthalmology consulting, W.D. and D.C.; supervision and project administration, Y.C., W.D., X.J., and F.D.; funding acquisition, X.J., Y.C., and W.D.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: July 31, 2023
Revised: October 3, 2023
Accepted: January 11, 2024
Published: February 2, 2024

REFERENCES

1. Regulation, General Data Protection (2018). General Data Protection Regulation (GDPR). Intersoft Consulting. Accessed in October *24*.
2. McMahan, B., Moore, E., Ramage, D., Hampson, S., and Arcas, B.A.y. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. Proceedings of the 20th International Conference on Artificial Intelligence and Statistics *54*, 1273–1282.
3. Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. (2019). On the Convergence of FedAvg on Non-IID Data. In International Conference on Learning Representations (ICLR).
4. Zhong, Z., Zhou, Y., Wu, D., Chen, X., Chen, M., Li, C., and Sheng, Q.Z. (2021). P-FedAvg: Parallelizing federated learning with theoretical guarantees. In IEEE INFOCOM 2021-IEEE Conference on Computer Communications.
5. Khanduri, P., Sharma, P., Yang, H., Hong, M., Liu, J., Rajawat, K., and Varshney, P. (2021). STEM: A Stochastic Two-Sided Momentum Algorithm Achieving Near-Optimal Sample and Communication Complexities for Federated Learning. Adv. Neural Inf. Process. Syst. *34*, 6050–6061.
6. Tran Dinh, Q., Pham, N.H., Phan, D., and Nguyen, L. (2021). FedDR–Randomized Douglas-Rachford splitting algorithms for nonconvex federated composite optimization. Adv. Neural Inf. Process. Syst. *34*, 30326–30338.
7. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. (2020). Federated optimization in heterogeneous networks. Proceedings of Machine Learning and Systems *2*, 429–450.
8. Laguel, Y., Pillutla, K., Malick, J., and Harchaoui, Z. (2021). A Superquantile Approach to Federated Learning with Heterogeneous Devices. 2021 55th Annual Conference on Information Sciences and Systems (CISS).
9. Chen, Y., Ning, Y., Slawski, M., and Rangwala, H. (2020). Asynchronous Online Federated Learning for Edge Devices with Non-IID Data. In 2020 IEEE International Conference on Big Data (Big Data).
10. Lima Pilla, L. (2021). Optimal Task Assignment for Heterogeneous Federated Learning Devices. 2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS).
11. Li, T., Hu, S., Beirami, A., and Smith, V. (2021). Ditto: Fair and Robust Federated Learning Through Personalization. Proceedings of the 38th International Conference on Machine Learning *139*, 6357–6368.
12. Li, D., and Wang, J. (2019). FedMD: Heterogenous Federated Learning via Model Distillation. Preprint at arXiv. https://doi.org/10.48550/arXiv.1910.03581.
13. Arivazhagan, M.G., Aggarwal, V., Singh, A.K., and Choudhary, S. (2019). Federated Learning with Personalization Layers. Preprint at arXiv. https://doi.org/10.48550/arXiv.1912.00818.

14. Ryffel, T., Trask, A., Dahl, M., Wagner, B., Mancuso, J., Rueckert, D., and Passerat-Palmbach, J. (2018). A generic framework for privacy preserving deep learning. Preprint at arXiv. https://doi.org/10.48550/arXiv.1811.04017.

15. Reina, G.A., Gruzdev, A., Foley, P., Perepelkina, O., Sharma, M., Davidyuk, I., Trushkin, I., Radionov, M., Mokrov, A., Agapov, D., et al. (2021). OpenFL: An open-source framework for Federated Learning. Preprint at arXiv. https://doi.org/10.48550/arXiv.2105.06413.

16. Mathur, A., Beutel, D.J., de Gusmão, P.P.B., Fernandez-Marques, J., Topal, T., Qiu, X., Parcollet, T., Gao, Y., and Lane, N.D. (2021). On-device Federated Learning with Flower. On-Device Intelligence Workshop at the Fourth Conference on Machine Learning and Systems (MLSys).

17. Liu, B., Tan, C., Wang, J., Zeng, T., Shan, H., Yao, H., Huang, H., Dai, P., Bo, L., and Chen, Y. (2021). Fedlearn-Algo: A flexible open-source privacy-preserving machine learning platform. Preprint at arXiv. https://doi.org/10.48550/arXiv.2107.04129.

18. Ro, J.H., Suresh, A.T., and Wu, K. (2021). FedJAX: Federated learning simulation with JAX. Preprint at arXiv. https://doi.org/10.48550/arXiv.2108.02117.

19. Lai, F., Dai, Y., Singapuram, S., Liu, J., Zhu, X., Madhyastha, H., and Chowdhury, M. (2022). FedScale: Benchmarking model and system performance of federated learning at scale. Proceedings of the 39th International Conference on Machine Learning *162*, 6357–6368.

20. Xie, Y., Wang, Z., Gao, D., Chen, D., Yao, L., Kuang, W., Li, Y., Ding, B., and Zhou, J. (2023). FederatedScope: A Flexible Federated Learning Platform for Heterogeneity. Proceedings VLDB Endowment *16*, 1059–1072.

21. Liu, Y., Fan, T., Chen, T., Xu, Q., and Yang, Q. (2021). FATE: An Industrial Grade Platform for Collaborative Learning With Data Protection. J. Mach. Learn. Res. *22*, 1–6.

22. He, C., Li, S., So, J., Zeng, X., Zhang, M., Wang, H., Wang, X., Vepakomma, P., Singh, A., Qiu, H., et al. (2020). FedML: A Research Library and Benchmark for Federated Machine Learning. Preprint at arXiv. https://doi.org/10.48550/arXiv.2005.05083.

23. Yuan, B., Ge, S., and Xing, W. (2020). A Federated Learning Framework for Healthcare IoT devices. Preprint at arXiv. https://doi.org/10.48550/arXiv.2005.05083.

24. Chen, Y., Qin, X., Wang, J., Yu, C., and Gao, W. (2020b). FedHealth: A Federated Transfer Learning Framework for Wearable Healthcare. IEEE Intell. Syst. *35*, 83–93.

25. Xu, J., Glicksberg, B.S., Su, C., Walker, P., Bian, J., and Wang, F. (2021). Federated Learning for Healthcare Informatics. J. Healthc. Inform. Res. *5*, 1–19.

26. Adnan, M., Kalra, S., Cresswell, J.C., Taylor, G.W., and Tizhoosh, H.R. (2022). Federated learning and differential privacy for medical image analysis. Sci. Rep. *12*, 1953.

27. Pradhan, K., and Chawla, P. (2020). Medical Internet of things using machine learning algorithms for lung cancer detection. Journal of Management Analytics *7*, 591–623.

28. Deist, T.M., Dankers, F.J.W.M., Ojha, P., Scott Marshall, M., Janssen, T., Faivre-Finn, C., Masciocchi, C., Valentini, V., Wang, J., Chen, J., et al. (2020). Distributed learning on 20 000+ lung cancer patients–The Personal Health Train. Radiother. Oncol. *144*, 189–200.

29. Yan, B., Wang, J., Cheng, J., Zhou, Y., Zhang, Y., Yang, Y., Liu, L., Zhao, H., Wang, C., and Liu, B. (2021). Experiments of Federated Learning for COVID-19 Chest X-ray Images. Communications in Computer and Information Science *1423*, 41–53.

30. Cetinkaya, A.E., Akin, M., and Sagiroglu, S. (2021). A communication efficient federated learning approach to multi chest diseases classification. In 2021 6th International Conference on Computer Science and Engineering (UBMK).

31. Feki, I., Ammar, S., Kessentini, Y., and Muhammad, K. (2021). Federated learning for COVID-19 screening from Chest X-ray images. Appl. Soft Comput. *106*, 107330.

32. Ogier du Terrail, J., Ayed, S.-S., Cyffers, E., Grimberg, F., He, C., Loeb, R., Mangold, P., Marchand, T., Marfoq, O., Mushtaq, E., et al. (2022). FLamby: Datasets and Benchmarks for Cross-Silo Federated Learning in Realistic Healthcare Settings. Adv. Neural Inf. Process. Syst. *35*, 5315–5334.

33. Parekh, V.S., Lai, S., Braverman, V., Leal, J., Rowe, S., Pillai, J.J., and Jacobs, M.A. (2021). Cross-Domain Federated Learning in Medical Imaging. Preprint at arXiv. https://doi.org/10.48550/arXiv.2112.10001.

34. Mondrejevski, L., Miliou, I., Montanino, A., Pitts, D., Hollmén, J., and Papapetrou, P. (2022). FLICU: A Federated Learning Workflow for Intensive Care Unit Mortality Prediction. In 2022 IEEE 35th International Symposium on Computer-Based Medical Systems (CBMS).

35. Hanif, A., Lu, C., Chang, K., Singh, P., Coyner, A.S., Brown, J.M., Ostmo, S., Chan, R.V.P., Rubin, D., Chiang, M.F., et al. (2022). Federated Learning for Multicenter Collaboration in Ophthalmology: Implications for Clinical Diagnosis and Disease Epidemiology. Ophthalmol. Retina *6*, 650–656.

36. Teo, Z.L., Lee, A.Y., Campbell, P., Chan, R.P., and Ting, D.S. (2022). Developments in Artificial Intelligence for Ophthalmology: Federated Learning. Asia Pac. J. Ophthal. *11*, 500–502.

37. Lo, J., Yu, T.T., Ma, D., Zang, P., Owen, J.P., Zhang, Q., Wang, R.K., Beg, M.F., Lee, A.Y., Jia, Y., and Sarunic, M.V. (2021). Federated Learning for Microvasculature Segmentation and Diabetic Retinopathy Classification of OCT Data. Ophthalmol. Sci. *1*, 100069.

38. Nguyen, D.C., Pham, Q.-V., Pathirana, P.N., Ding, M., Seneviratne, A., Lin, Z., Dobre, O., and Hwang, W.-J. (2022). Federated Learning for Smart Healthcare: A Survey. ACM Comput. Surv. *55*, 1–37.

39. Kreuzberger, D., Kühl, N., and Hirschl, S. (2023). Machine Learning Operations (MLOps): Overview, Definition, and Architecture. IEEE Access *11*, 31866–31879.

40. The Kubernetes Authors (2014). Kubernetes. GitHub. https://github.com/kubernetes/kubernetes.

41. Wightman, R., Raw, N., Soare, A., Arora, A., Ha, C., Reich, C., Guan, F., and Kaczmarzyk, J. (2019). mrT23; Mike. PyTorch Image Models. Zenodo. https://doi.org/10.5281/zenodo.7618837.

42. TorchVision Maintainers and Contributors (2016). TorchVision: PyTorch's Computer Vision Library. GitHub. https://github.com/pytorch/vision

43. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2020). Transformers: State-of-the-Art Natural Language Processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations.

44. Merkel, D. (2014). Docker: Lightweight Linux Containers for Consistent Development and Deployment. Linux J. *2*.

45. He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

46. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021c). Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).

47. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In International Conference on Learning Representations.

48. Deng, L., Lyu, J., Huang, H., Deng, Y., Yuan, J., and Tang, X. (2020). The SUSTech-SYSU dataset for automatically segmenting and classifying corneal ulcers. Sci. Data *7*, 23.

49. Kermany, D., Zhang, K., and Goldbaum, M. (2018). Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification. Mendeley Data. https://doi.org/10.17632/rscbjbr9sj.2.

50. Yan, B., Cao, D., Jiang, X., Chen, Y., Dai, W., Dong, F., Huang, W., Zhang, T., Gao, C., Chen, Q., FedEYE, et al. (2023). A Scalable and Flexible End-To-End Federated Learning Platform for Ophthalmology. Zenodo. https://doi.org/10.5281/zenodo.10432014.