Model-Centric Federated Machine Learning

AUTHORS, Institute xx, Country

Traditional Federated Machine Learning follows a server-domincated cooperation paradigm which narrows the application scenarios of federated learning and decreases the enthusiasm of data holders to participate.

CCS Concepts: • **Computer systems organization → Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: datasets, neural networks, gaze detection, text tagging

ACM Reference Format:
Authors. 2018. Model-Centric Federated Machine Learning. *Proc. ACM Meas. Anal. Comput. Syst.* 37, 4, Article 111 (August 2018), 28 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

In recent years, the barriers to the development of Artificial Intelligence (AI) have been broken down with the rapid progress of ABC technologies in computing: AI, Big Data, and Cloud Computing, as well as the emergence of cost-effective specialized hardware [164] and software [73]. This has led to the world entering the third wave of AI development: Deep Learning [85]. The success of current data-driven AI relies on massive amounts of training data and follows a gather-and-analyze paradigm [179], which confronts with challenges of complying with rigorous data protection regulations such as OECD Privacy Guidelines [168] and General and Data Protection Regulation (GDPR) [171]. So although data-centric AI is now the mainstream, a novel model-centric distributed collaborative training framework called Federated Learning is gaining popularity in both academia and industry due to its advantages in complying with privacy regulations. So although data-centric AI is currently mainstream, Federated Learning (FL) [99], a novel model-centric distributed collaborative training framework, is gaining popularity in both academia and industry for its advantages in complying with privacy regulations [169].

According to the definitions of IEEE Standard for Federated Machine Learning (FML, aka FL) [160], *FL is a framework or system that enables multiple participants to collaboratively build and use machine learning models without disclosing the raw and private data owned by the participants while achieving good performance.* For example, a typical workflow of FL systems is that the entity with modeling demand (aka FL server) first deploys the FL services and initializes the model training task, and then distributing this task to participants with training data (aka FL clients) for modeling [13]. Based on this workflow pattern, many FL frameworks have been derived with specialized improvements in communication [81, 120, 184], optimizaiton [79, 96, 100], robustness [35, 91, 154] and privacy [14, 26, 46]. While these fascinating improvements greatly enhance the utility of FL, they all follow a task-based interaction paradigm, in which an FL server dominates the cooperation between FL participants. In this narrow interpretation of FL, the data owner is treated more like a worker than a collaborator and performs training primarily for the benefit of the server's goals. Due to the above defects, clients have little enthusiasm to participate, and the potential for redundant training also leads to low model reusability, further diminishing

Author's address: Authors, Institute xx, City, Country, @mail.com.

2476-1249/2018/8-ART111 $15.00
https://doi.org/XXXXXXX.XXXXXXX

the efficiency of the FL systems. This explains why current FL frameworks are more akin to private distributed modeling services rather than sustainable and privacy-preserving modeling platforms for everyone as expected.

In this paper, we try to answer the question: **Can we establish a sustainable open FL platform based on a novel reciprocal cooperation framework?** Obviously, to answer this quesion, it is insufficient simply study the basic concepts of FL and investigate existing FL techniques. We also need to conduct a wide survey of potential techniques that can facilitate the construction of open FL platforms. To aid understanding, Fig. 1 provides a first glimpse of two novel FL cooperation frameworks we advocated:

- **Query-based FL.** It follows a loosely-coupled cooperation framework between entities (we use "entities" instead of "participants" to emphasizes equality), where any entity can freely upload their local models or retrieve models from the open repository named Model Community. There are many valuable challenges that can be explored, such as how to query for models, how to "assemble" the retrieved models, or how to transfer knowledge from these models (see Section 3).
- **Contract-based FL.** It follows a mutual choice cooperation framework, where each entity can deploy model training contracts with specialized requirements such as task modality, execution environment, model architecture and license. Meanwhile, entities holding data can choose whether to accept the contract. Research topics in this area include model pricing, model ownership verification and .... (see Section **??**)

It's worth noting that the definitions of the four roles (i.e., model user, coordinator, data owner, auditor) are adopted for compatibility with the IEEE standard [160], and our proposals are also within the scope of FML definitions. The diagram in Fig. 1(c) illustrates the workflow of traditional FL, where all FL clients are required to accept the training schedule from the FL server and perform multiple rounds of local training until the model converges. In contrast, the entities in query-based FL and contract-based are proactive in their participate. We believe that these reciprocal cooperation frameworks have the potential to expand the prevalence of FL and establish FL ecosystems.

1.1 Related Surveys

Federated learning has become a buzzword in various fields, leading to the emergence of numerous FL studies. These works can be classified into three primary categories: FL systems design, FL appllications and FL toolkits. Extensive surveys are available to summarized the advancement of federated learning, as shown in Table 1. The initial architectures and concepts for FL systems were summaried by Yang *et al.* [187]. They categorized FL into

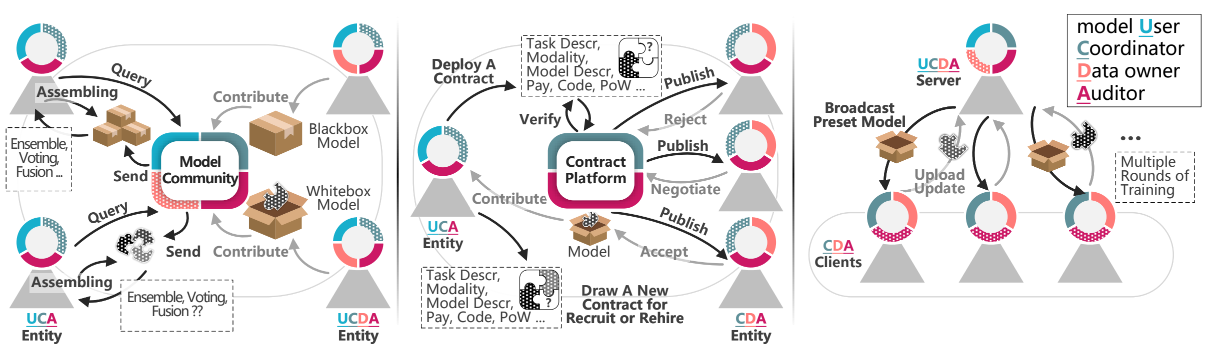


Fig. 1. A schematic diagram of three cooperation frameworks of FL. (a) (b) are the proposed open FL platforms, (c) is the traditional FL platform. Four colors correspond to four roles in [160], and colors with grid lines indicate non-essential roles.

horizontal FL, vertical FL and federated transfer learning based on the distribution characteristics of data, which are written in IEEE Standard 3652.1-2020 [160, 186]. Following this, an increasing number of surveys have emerged focusing on enhancing FL system design [7, 78, 97, 99, 198]. From the algorithmic perspective, personlized FL [83, 165] aims to learn personlized models for each client to address the challenge of statistical heterogeneity [115]. Besides, the privacy-perserving computing platforms and model aggregation protocols for FL systems also been widely studied and sumaried by [38, 110, 114, 190]. Furthermore, many advanced FL architectures had been proposed, such as asynchronous [184], decentralized and blockchain-based FL frameworks [123, 138, 208]. Given that federated learning technologies enable collaboration among distributed participants in model training and decision-making, this capability holds great promise in a wide range of application scenarios. For instance, multiple geogrphically distributed medical insitutions can enhace medication recommendation, drug-drug interaction prediction and medical image analysis in a collaborative manner without exchanging any sensitive data [8, 136, 148, 185]. The massive real-time data generated by IoT devices in smart cities [143, 204], industries [15], vehicles [34] has also sparked interest in exploring how FL technology can be used to deliver more advanced services such as intrusion detection, anomaly detection, fraud detection and network load prediction [5, 6, 47].

As summarized in Table 1, most surveys extensively discuss the challenges of efficiency, heterogeneity, privacy in FL systems design, with the surveys from blockchain fileds offering the most comprehensive review. However, except for a few blockchain-based FL studies, most of the above surveys just present the same story from slightly different angles or backgrounds, i.e a server sets the model training task and delegate it to data holders to complete. This *server-dominated* cooperation framework is a narrow implementation of the FL systems. Therefore, this survey aim to fill the gap by investigating and surveying the associated tenchnologies that support more open and inclusive cooperation frameworks in FL systems, where all entities, whether they own the data or not, can benefit from it. The challenges investigated in this survey are not listed in the Table 1, to the best of our knowledge, this is the first survey that focuses on the **cooperation frameworks** of FL. In the following section, we will differentiate this survey from other related concepts in the field of FL.

1.2 Distinction of Our Survey

This survey focuses on exploring the innovative cooperation frameworks in FL, which will involve some FL concepts such as decentralized FL, blockchain-based FL, few-shot FL, ML related platforms and services but goes beyond them. In this section, we will distinguish our survey by highlingting the similarities and differences between these related concepts.

1.2.1 FL Systems. Federated learning, with its nature advantages in privacy-preserving decision sharing, has garnered significant attention in both industry and academia, leading to the rapid development of federated learning systems. The earliest attempt at the large-scale FL system was by Google, where FL was used to improve next-word prediction [60] and query suggestion [188] for Gboard applications. Subsequently, many novel FL systems have emerged to adapt to diverse federated training scenarios, such as Horizontal FL (e.g. TFF [1], FedLab [197], Felicitas [202], IBM FL [113]), Vertical FL [182] or both (e.g. FATE [108], FedML [61], PaddleFL [116], Flower [11], FedTree [93], NVFLARE [151]). Despite these frameworks covering a wide range of application scenarios, they all follow the server-dominated cooperation mechanism. This business model restricts FL to function as a collaborative modeling software, rather than an open platform that provides FL services to the public.

Unlike the FL systems mentioned above, PySyft [211] developed by OpenMined depicts a novel FL cooperation frameworks which is closely realted to our focus. PySyft encourages data owners to share their data on a private domain server, which provides data management and privacy controls, as well as limited machine learning analysis APIs for third-party data scientists. Besides, a public network server will provide connections between data owners and data scientist, enabling datasets search and discovery for platform users. Recently, a new FL

Table 1. Summary of existing FL surveys, SYS denotes FL Systems Design, APP denotes FL Applications, SDC denotes Server-Dominated Cooperation frameworks.

		Challenges					Contents		
Scenarios/Tasks	FL Surveys	Efficiency	Heterogeneity	Privacy	Incentive	Decentralized	SYS	APP	SDC
General	Yang *et al.* [187]	✓	✓	✓	✓	✓	✓	✓	✓
	Li *et al.* 2020 [99]	✓	✓	✓		✓	✓	✓	✓
	Zhang 2021*et al.* [198]	✓	✓	✓			✓	✓	✓
	Gupta *et al.* [56]	✓	✓	✓		✓	✓	✓	✓
	Xu *et al.* [184]	✓	✓	✓		✓	✓	✓	✓
	Li *et al.* 2021 [97]	✓	✓	✓	✓	✓	✓	✓	✓
	El *et al.* [38]			✓		✓	✓		✓
	Kulkarni *et al.* [83]	✓	✓				✓		✓
	Liu *et al.*[110]	✓		✓		✓	✓		✓
	Tan *et al.* [165]		✓				✓		✓
	Zhu *et al.* 2021 [207]		✓				✓		✓
	Ma *et al.* [115]	✓	✓	✓			✓		✓
	Aledhari *et al.* [7]	✓	✓				✓	✓	✓
	Kairouz *et al.* [78]	✓	✓	✓	✓	✓	✓	✓	✓
	AbdulRahman *et al.* [3]	✓	✓	✓	✓		✓	✓	✓
	Lim *et al.* [105]	✓	✓	✓	✓		✓	✓	✓
Healthcare	Xu *et al.* [185]	✓	✓	✓			✓	✓	✓
	Pfitzner *et al.*[136]	✓	✓	✓			✓	✓	✓
	Antunes *et al.* [8]		✓	✓				✓	✓
	Rieke *et al.* [148]		✓	✓		✓	✓	✓	✓
IoT	Zhang 2022*et al.* [204]	✓	✓				✓	✓	✓
	Boopalan *et al.* [15]	✓	✓	✓	✓	✓	✓	✓	✓
	Ramu *et al.* [143]	✓	✓	✓		✓	✓	✓	✓
	Du *et al.* [34]	✓	✓	✓	✓	✓	✓	✓	✓
Cybersecurity	Agrawal *et al.* [5]	✓	✓	✓		✓	✓	✓	✓
	Alazab *et al.* [6]			✓			✓	✓	✓
	Ghimire *et al.* [47]	✓		✓			✓	✓	✓
Blockchain	Nguyen *et al.* [123]	✓	✓	✓	✓	✓	✓	✓	✓
	Qu *et al.* [138]	✓	✓	✓	✓	✓	✓	✓	✓
	Zhu *et al.* 2022 [208]	✓	✓	✓	✓	✓	✓	✓	✓

platform named PySyTFF[1] was announced. It integrates TFF and PySyft, allowing data scientists to train models under the coordination of TFF and the datasets provided by PySyft domain servers. However, even with inference controls of datasets, there is still a high security risk associated with exposing access to sensitive data on the Internet [42]. To preserve the privacy advantages of FL, in this survey, we aim to discuss an open and data-free FL platform under the scope of model-centric ML [112]. In such FL platform, every user is free to collaborate on the training of machine learning models while privacy is protected.

1.2.2 As-a-Service Business Model. In the current context of Software-as-a-Service (SaaS) [17], there are several as-a-service cloud computing frameworks that encapsulate ML tasks as services and provides unified APIs for upper layer applications. For example, Model-as-a-Service (MaaS) [44, 107, 149, 162, 212] and Machine-Learning-as-a-Service (MLaaS) [58, 64, 82, 92, 147] encapsulate model execution and model development as services. The original concept of MaaS [44, 149] was to provide re-usable and fine-grained user interfaces and visualization tools of domain-specific models (e.g wealther model, oil spill detection model) for environmental decision support

[1] https://blog.openmined.org/announcing-proof-of-concept-support-for-tff-in-pysyft-0-7/

systems. Subsequently, this concept has been extended to the field of recommendation systems [212] and deep learning based systems [107, 162]. However, in contrast to the focus of this survey, the aforementioned MaaS framework does not involve any user collaboration but solely provides model inference APIs to users.

As the architectures of deep neural networks (DNNs) become increasingly complex, training and maintaining DNNs become more and more challenging [57]. To address this issue, cloud service providers have introduced MLaaS, which offers an integrated development environment as a service for constructing and operationalizing ML workflows, aiming to reduce the computational resources required. MLaaS enables users to upload their data for training [64, 147, 205] or inference [58], freeing them from the responsibility of managing hardware resources and implementation. Most MLaaS providers adopt a pay-by-query business model, such as Google Vertex AI[2], Microsoft Azure Machine Learning[3] and ChatGPT[4]. However, privacy protection can be compromised when users upload data to perform inference and training in the cloud. Moverover, under this model, users are not given the ability to contribute their own models to the repository or collaborate with others to enhance the diversity of available models. While there are some ongoing efforts to offer privacy-preserving MLaaS services using techniques such as Isolated Execution Environment [58, 119] and Homomorphic Encryption [45, 64], it is worth noting that our focus is not solely on privacy. Rather, the FL framework we focus on emphasizes a collaborative framework where all entities involved have equal access to services and mutual benefits.

Recently, Kourtellis *et al.* [82] propose Federated Learning as a Service (FLaaS) that provides high-level and extensible APIs aim to enabling third-party applications to build collaborative, decentralized, privacy-preserving ML models. Jiang *te al.* [75] propose an open FL ecosystem for mobile devices, which shares a similar concept to FLaaS. However, those approach also follow the traditional server-dominated cooperation framework, which falls under the scope of previous FL surveys[78, 99, 187].

1.2.3 Dcentralized FL. TODO: given the high scalability of modern edge computing networks, a single MEC server cannot manage to aggregate all updates offloaded from millions of devices. Therefore, there is an urgent need to develop a more decentralized FL approach without using a central server so as to solve security and scalability issues for enabling the next generation intelligent edge networks.

1.2.4 Blockchain-based FL. TODO:

1.2.5 Few-shot FL. TODO:

1.3 FAIR in FL

TODO: FAIR Data Principles: Findable, Accessible, Interoperable, Reusable.

2 BASIC CONCEPTS OF FEDERATED LEARNING

2.1 Definition

Federated Learning [120, 160] is a collaborative machine learning modeling paradigm that enables sharing and aggregation of knowledge from multiple sources while maintaining the confidentiality of source data. Generally, in terms of task organization, there are two kinds of entities in FL systems: the server and participant. The FL server can launche a federated training task and invites participants with sufficient training data and hardware resources to contribute their local modeling results for multi-source knowledge aggregation. In practice, FL systems can be divided into two categories based on application scenarios [78]:

- Cross-device FL. In this setting, the participants are numberous end devices with relatively small dataset size, such as mobiles, IoT sensors and wearable devices, the server is hosted in the cloud. Since there is low context correlation between the data of distributed end devices and less overlapping sample ids, this

[2] https://cloud.google.com/vertex-ai [3] https://azure.microsoft.com/products/machine-learning/ [4] https://chat.openai.com/chat

setting typically falls within the scope of horizontal FL. The cross-device FL applications include: Gboard input suggestion [60, 142, 188], e-commerce recommendation [125].

- Cross-silo FL. In this setting, the participants are orginizations or institutions with large amounts of well-maintained structured data, and the server is hosted by a trusted FL service providers such as FATE [108] and NVFLARE [151]. As participants can be different departments within an organization, the data silo owned by these departments can have a large overlap in sample space and less overlap in feature space, which falls within vertical FL. The applications of cross-silo FL include federated data analysis for radiomics [101, 102, 156], epidemiology [31] and EHR [18, 66].

The allocation to the server and participants in FL is dependent on the particular application context. Furthermore, FL entities can also serve multiple functional roles to support advanced features such as privacy enhancement [14, 46, 125], participant scheduling [2, 91], model verification [158, 167] and incentive mechanisms [193]. Recall that there are four roles defined in the FL standard [160]:

- Model User. The FL model users can request for FL modeling services and preset the targeted task, and then establish cooperation with participants who provide training data. This role can leverage the benefits of collaborative training to improve the preformance of its objective models.
- Coordinator. The FL coordinators are responsible for providing FL services to all FL entities. This role involves setting up communication channels with entities, initializing the execution environment of participants [58], scheduling the training and aggregation workflows for improve system efficiency, such as by alleviating the straggler effect [21, 89], optimizing data heterogeneity [2, 36] and compressing model transfer [81, 154]. Additionaly, the FL coordinator provides privacy control mechanisms [14, 38, 64] for model users and authorization verification for participants to maintain the security of FL systems. Furthermore, the coordinator can hold a validation dataset for evaluate the models contributed by participants or detect potential disturbances from Byzantine attacks [153].
- Data Owner. The FL data owners are knowledge contributors of FL systems, they collect and desentize raw data to maintain a local dataset for federated training. Although they have full authority of data processing and modeling, they cannot share the raw data due to privacy concerns. To address these concerns, de-identification [4] and differential privacy [37] techniques can be applied to meet privacy budgets as required by privacy policies.
- Auditor. The FL auditors are responsible for formulating privacy control policies and establishing supervisory mechanisms that ensure the training process is compliant with data protection regulations (e.g. HIPAA [4], GDPR([171])) and preventing potential privacy breaches for both model users and data owners. Especially in FL, the latent knowledge in models can potentially reveal the sensitive information of training data [76, 177, 209], making it crucial for auditors to scrutinize the model transmission [103, 178] and verify the ownership of models [158, 167].

Fig. 2 illustrates the typical architecture of FL systems, which as a distributed modeling toolkits consists of server part and client part. In general FL setting, the server part is the central aggregator installed in a trusted cloud environment, while the client part of software can operate in different operating environments of client devices. The server and clients are connected via Internet and typically with the help of Remote Procedure Call (RPC) interface for coordinating [1, 11, 61, 108, 202]. We use four colors to represent the four FL roles and the colors with grid lines indicate non-essential roles. For example, in Fig. 2, the UCDA server takes on the roles of model user, coordinator and audior in traditional FL. However, it is no necessary to hold training data or validation data, so the role of data owner is non-essential. To illustrate the workflow of traditional FL, we leverage the vanilla FL framework Federated Averaging (FedAvg) [13, 120] as an example.

First, the FL server pre-defines the objective modeling task and initializes the server process. Secondly, the coordinator in server-side specifies a preset global model and the operational parameters. Thirdly, the coordinator

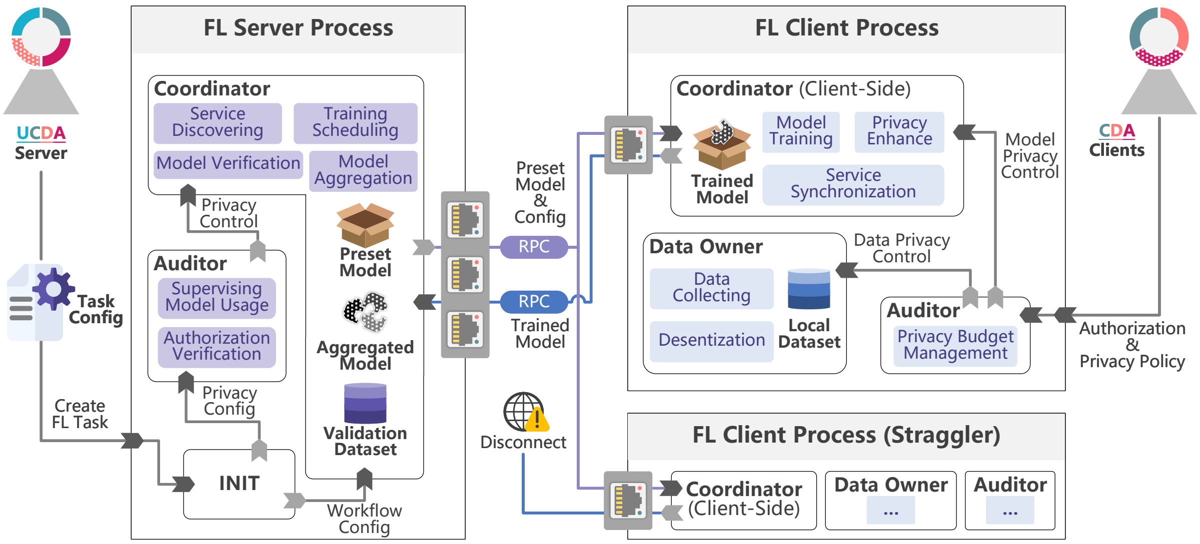


Fig. 2. An overview of traditional FL systems. (U: model User, C: Coordinator, D: Data owner, A: Auditor)

discovers the availability of clients' FL services, boardcasts the global model and training config to them. The training config contains bath size, local epoch round, optimizer parameters and so on. Then, the coordinator will wait for the trained results contributed by the coordinator in clients-side and drop those clients with network problems. Finally, the server aggregates the trained resultes received from various clients into the global model and begins a new round based on this aggregated global model. The aggregation strategy adopted in FedAvg is the weighted model parameters based on the size of local dataset, which means the global objective of FL can be regarded as a joint objective function of clients. By this way, the FL server can learn a generalized global model by jointly optimizing all local optimization objectives and incorporating the latent knowledge from the local models. Although the auditor component was not included in earlier FedAvg, it play an important role in the later business-ready FL frameworks [108, 151, 211].

However, in comparing FedAvg workflow described above with Fig. 2, it is easy to notice that the client part has been excluded. This is because we are elaborating from a server-side perspective, which is usual way FL is presented [19, 98, 120]. Actually, the underlying reason is that in traditional FL, the client-side process is tightly coupled with server-side process, and there is no alternative for clients other than to either accept or reject the training scheduling from the server wholesale. So the clients are not considered as an autonomous entities but rather work as subordinates to server. In this server-domianted cooperation framework, the benefits and autonomy of clients are compromised, which hinders their enthusiasm to participate in FL network and subsequently limits the applicability of FL. From this perspective, we summarize the limitations of traditional FL in the next section, which motivates us to explore more innovative sustainable FL cooperation frameworks.

2.2 Limitations of Traditional FL

Previous surveys [6, 78, 99, 123, 165, 187, 204, 208] has extensively discussed the challenges in FL systems from various aspects However, the cooperation mechanism of FL systems has been overlook because almost all mainsteam FL frameworks follow the FL prototype [120], which shape the form of current FL frameworks:

a modeling software. We summarize three inherent limitations of traditional FL cooperation mechanism: (1) **Server-client Coupling**, (2) **Low Model Reusability**, (3) **Non-public**.

2.2.1 Server-client Coupling. The tightly-coupled server-client design is a major limitation of FL systems. From the perspective of FL service providers, adapting the programs to heterogeneous client hardware and software components, such as various operating and database systems, processor and storage architectures, communication protocols, eneragy constrains and data licenses, is a challenging task that significantly increases the complexity of the FL system.

On the other hand, the invasive software deploy mode compromises the integrity of client environments and expose them to new privacy risks. Specifically, the coordinator components (client-side) pushed by the server may not offer demanded privacy control mechanisms [19, 120, 197], or cause resource depletion on client-side [13, 25, 125], or even piggyback malicious executable codes [90]. So the auditor role of client is non-essential as depicted in Fig. 2, not only because the client maybe lacks a corresponding policy for FL training, but also because its privacy is not completely under its control. Likewise, the malicious clients can also exploit the vulnerability in the aggreagation strategy to currupt the FL tranining process [16, 39, 131, 153] or insert backdoors [10, 172]. In addition, the unstable network environment can drive clients to drop out from traning (i.e. straggler effect), thereby reducing system efficiency [131, 146]. Therefore, the server-client coupling design of traditional FL systems make them susceptible to unpredictable runtime environments, leading to system vulnerability and low reliability.

2.2.2 Low Model Reusability. The traditional FL scheduling follows a task-centric manner and erminates once the training reaches a preset number of rounds or meets traget metrics on global model set by FL server [13]. As a result, only FL server can guarantee having the latest global model after the task is terminated. This ad-hoc modeling paradigm results in low model reusability and transportability. For example, if a client who participated in the previous training turn wants to continue training, they can only start the task from scratch unless they have the up-to-date global model. Since only FL server is able to maintain the complete modeling trajectory, it is difficult for the client to roll back the training itself to eliminate the potential privacy risk. Furthermore, the non-deliverable scheduling mechanism of FL tasks also hinders inter-task model reuse, which leads to unnecessary wasted energy and time on participants that have been involved in similar tasks.

2.2.3 Non-public. As we mention in Sec. 1.2.1, except PySyft [211], the application scenarios of mainstream FL frameworks [1, 11, 19, 61, 108, 113, 151, 197] aim to provide private collaborative ML traning service, and there is no any accessible FL platform for the public. Although there have been real-world deployment practices of FL for the public with scales of millions [13] and billions [125], these have been carried out only by tech giants with a massive base of active users. For an individual user, there is no practical way to organize such a large-scale FL training network.

But in fact, due to the limitations in the cooperation mechanism mentioned above, data owners are not sufficiently motivated to participate in this server-take-all FL training even if it is public accessible. Therefore, the cornerstone of buliding a sustainable open FL platform is to establish a reciprocal FL cooperation framework, followed by corresponding mulit-source knowledge aggregation strategies, which we discuss in the following sections.

3 QUERY-BASED FEDERATED LEARNING

3.1 Overview

Let us continue by establish a sustainable open FL platform based on a query-based cooperation framework. An overview of this platform is presented in Fig. 3, the desin philosophy behind this framework is to break the coupling between FL server and clients. In the query-based FL systems, all traditional FL roles and components

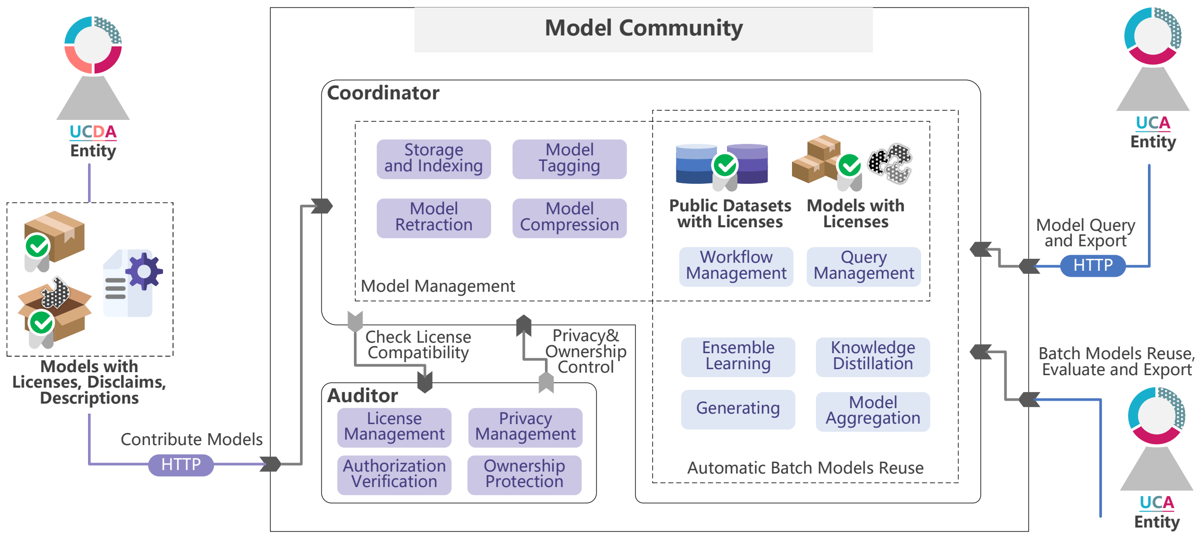


Fig. 3. An overview of query-based FL systems. (U: model User, C: Coordinator, D: Data owner, A: Auditor)

are maintained on an open model repository called Model Community. The Model Community privdes a one-stop ML models redistribution and reuse service, including model indexing, automatic batch model reuse, license management, privacy control and so on. In addition to large-scale pretrained models like BERT [33], BLOOM [155] with great generalization abilities, we also encourage individuals to upload their task-specific models trained on limited domain data to boost the knowledge mining within models [191]. The derivatives of knowledge mining can learn representations from multiple domains, resulting in more promising performance that can be evaluated by platform users. Furthermore, the contributors can open models under applicable licenses, granting them distribution control and legal protection of their intellectual property (IP). In summary, the properties of query-based FL are: (1) **Model Agnostic**, as there are no restrictions on the types and architectures of the models uploaded by users; (2) **Contactless**, as coomunication channels need not be maintained; (3) **Community-powered**, whereby sharing models enrichs the entire community.

Actually, we aim to advocate a novel SaaS [17] ML platform with automatic model reuse integrated, which has potential to leverage the transportability of models to address previously unexplored ML problems. Due to the high computational demands of deep learning, current ML platforms primarily concentrate on computing, for example, MaaS, MLaaS, FLaaS provide ML models deployment and development services to handle user-specified tasks. (Section 1.2.2). On the other hand, there are several ML platforms provide open model search and download services. So, can we leverage leverage off-the-shelf open model platforms to build a query-based FL system? Unfortunately, these platforms are designed solely for sharing and are no suitable for more advanced functionalities such as model ensemble [70] and knowledge distillation [65], we will explain the reasons in the following section.

3.2 How to Query for Models

To establish a query-based FL platform, the first thing that comes to mind is how to query for models. Unlike traditional ML model sharing repositories that mainly query for a specific model by name, it requires an efficiency approach to export a batch of target models that ready for ensemble or distillation. We summaried the filter

Table 2. Filter conditions and characteristics of DNNs repositories. ✓: Supported, ✗ : Unsupported, ! : Information provided but unsearchable, listed in descending order by number of models.

	DS Name	Model Architecture	Modality/Task	Tag	License	Input-Output	Batch Export	# of Models
Hugging Face[5]	✓	✓	✓	✓	✓	!	✗	133,641
Model Zoo[6]	✓	✓	✓	✓	✗	✗	✗	3,426
Tensorflow Hub[7]	✓	✓	✓	✓	!	!	✗	1,356
NVIDIA NGC[8]	!	✓	✓	✓	!	!	✗	527
OpenVINO[9]	!	✓	✓	✗	!	!	✓	278
Pytorch Hub[10]	!	✓	✗	✗	✗	!	✗	49

conditions of existing DNNs sharing repositories in Table. 2. The prevailing method for querying models involves searching for the desired model by its name, datasets used, associated tasks. To illustrate, one might search for the model name GPT [140], models trained on the MNIST dataset [86], or models capable of performing image segmentation tasks. However, this model retrieval method requires the users have a strong priori knowledge in data science, thus raising the barrier for knowledge mining within models. For example, there is no effective way to acquire a batch of image classfication models that contains the knowledge of *lesser panda* for further distillation. A compromise solution is to manually search the schema of each dataset one-by-one and subsequently search for models trained on those datasets.

Additionally, as shown in Table. 2, most DNNs repositories are simply list the description of input/output (e.g., NVIDIA NGC, OpenVINO) or even just present the source codes (e.g., Tensorflow Hub, Pytorch Hub), This lack of unified convention for model input/ouput poses a challenge for query-based FL. Besides, most of DNNs repositories do not enable querying models by licenses, resulting in the cumbersome task of individually handling model licenses and ensuring compatibility among different licenses. Hence, it is imperative to reconsider the design of DNN repositories to enable quick identification of readily reusable models for model knowledge mining. We further suggest following filter conditions for query-based FL.

3.2.1 Data Description. Similar with the data heterogeous challenges in FL [94]. The local datasets of contributors have varing quality and contain intractable biases, imbalances and noisies that can be attributed to the natural characteristics of demographic or improper data collection mechanisms [31]. Besides, label errors pervasive even in open datasets [126]. So, in addition to searching for domain-specific datasets based on their data descriptions, we are also seeking such descriptions for the purpose of future traceability and debugging. The data description can consist of statistical analysis results or the visualization diagrams that used to profile the data distribution [102] and complementary provenance information.

3.2.2 Workflow and History. The process of building an ML model is iterative, involving repeated hyperparameter tuning and architecture exploration, resulting in abundant workflow and historical trajectory data. This information includes pipelines, model structures, hyperparameter values for pre-training and fine-tuning, test metrics, and results. These data can be useful in filtering models that meet specific requirements, such as those with data standardization in preprocessing or evaluated using mean average precision (mAP). Instead of manually saving and uploading the logs and configuration files, a more efficient method is to leverage ML workflow management tools [170], such as MLflow [11] and Neptune [12], to automatically track and store the ML workflow during model building process. Additionally, to ensure that the computational consumption of models is within budget, the Deep Learning Profiler [13] can be leveraged to generate a report that shows the FLOPS and bandwidth requirements.

[5] https://huggingface.co [6] https://modelzoo.co/ [7] https://tfhub.dev/ [8] https://catalog.ngc.nvidia.com/models
[9] https://docs.openvino.ai/latest/model_zoo.html [10] https://pytorch.org/hub/ [11] https://mlflow.org [12] https://neptune.ai
[13] https://docs.nvidia.com/deeplearning/frameworks/dlprof-user-guide/index.html

3.2.3 Software Dependency. ML models are software that depend on underlying ML libraries, so it is important to declare the dependencies of the model to analyze software compatibility between batches of models. For instance, resource-constrained devices may need to trim down the list of software-dependent libraries to meet limited storage space requirements [30]. In some cases, contributed models may rely on other models as dependencies. For example, Fast R-CNN [48] uses VGG16 [159] as its backbone. It is crucial to release this information for further model license compatibility analysis.

The aforementioned filter conditions provide comprehensive coverage of the ML modeling process. However, there are additional requirements depending on the reuse mechanisms of the model retrieval side. For example, FedAvg [120] aggregates the local models weights element-wise, which requires full access to the models. In contrast, MoE with a gating network [70] only ensembles a batch of model outputs, so the individual models can remain blackboxes in this scenario. So, in the context of software licenses or model licenses, the batch models reused by FedAvg should be released as source code, while those reused by MoE can be released as binary object code (static linking). The above distinction is crucial for ensuring that model reuse results meet the legal framework, and this has been overlooked in traditional FL. We will expand on this topic in the following section.

3.3 How to Reuse Batch of Models

Once we have acquired a certain number of models that can contribute to the new target task, the next step is to reuse the knowledge of these pre-trained models, i.e., transfer their knowledge from source domain to the target domain [130]. However, before deciding on how to reuse the model, it is important to ensure that the necessary legal rights and permissions have been obtained. This may involve reviewing the terms and conditions of the licenses under which the models were originally released or obtaining permission from the original creators or copyright holders. Therefore, in this section, we will not focus on the technical details of how to reuse models, which is already covered by many related surveys, such as Transfer Learning [130], Ensemble Leanring [206], Domain Adaptation [176], Knowledge Distillation [175], Deep Generative Models [20] and Model Fusion [72]. Furthermore, the specific model reuse technique or techniques used is at the user's discretion, and the query-based FL platform we advocate is not bound or restricted to any particular model reuse algorithm. Innovatively, we study how to reuse batch of models, from the perspective of **legal compliance**.

The machine learning community benefits from the openness of ideas and code, and many high-impact ML conferences and journals encourage authors to publish their source code and dataset to research platforms like Papers With Code [14] and Code Ocean [15] to increase exposure and facilitate reproducibility. To restrict the use of ML techniques for unethical purposes (i.e. Deepfakes [122]) and protect the IP of creators, models are typically published under a license agreed upon by the licensor. Here, we summary the licenses, granted rights, restrictions and enforcements for ML models posted on Hugging Face in Table. 3.

3.3.1 Model Licensing Forms. ML models are licensed in three main forms: as software (e.g. Apache, MIT, GPL), as a model (e.g. OpenRAIL), and as content/database (e.g. CC-BY, PDDL). The reason for the mixed use of licenses is the ambiguity in the dependency relationship between the code, model, and data. ML models can be released with reproducable code and be considered as a component of software. So many open software licenses are naturally deferred for licensing of models. The most popular license is Apache-2.0, which is a permissive open software license that allows the freedom to make derivative works. However, the model building process also relies on a massive amount of data [85] that may be licensed under different licenses, which can lead to license conflicts. A practical example is BERT [33], which was published under the Apache-2.0 license but pre-trained on English Wikipedia documents that are licensed under CC BY-SA 3.0. This changing of license violates the requirement of the CC BY-SA 3.0, which states that any contribution must be distributed under the **same license** as the original work.

[14] https://paperswithcode.com [15] https://codeocean.com

From the perspective of content and database licensing, some word embedding models, such as GloVe [133], compute vector representations of words based on licensed open linguistic resources. These representations can be regarded as a translation of corpus and fall under the license of the original linguistic resources. A more complex scenario arises when the model is fine-tuned with other data that has a different license, for example, fine-tune RoBERTa [109] (MIT license) with SQuAD2 [141] (CC BY 4.0). The resulting model can be interpreted as both derived works and combined works.

Not only limited to protecting the IP and controlling the diffusion of ideas, but AI companies and researchers are also concerned about licensees using their models for unethical purposes [9, 77, 196], which is not restricted by traditional licenses based on the context of software and content. We can infer the concerns of the inventors of GPT-2 [140] about the unethical use of the model from its modified MIT license, which states, *We don't claim ownership of the content you create with GPT-2, so it is yours to do with as you please. We only ask that you use GPT-2 responsibly and clearly indicate your content was created using GPT-2.* However, such a statement lacks legal enforcement, and users may avoid accountability by convincing themselves that despite their efforts to minimize harm, they could not predict the AI artifact they generated would be used for harmful purposes. Besides, the original licensing frameworks (e.g. MIT, CC BY) for software and content are not well suited to the data-driven ML. Many ML operations, such as training, fine-tuning, inference, and distillation, are not explicitly defined in traditional software and content licenses, leaving a potential legal loophole for licensees.

To address the unique challenges and considerations surrounding the use and distribution of ML models, several specific licenses for ML models have been proposed. The CreativeML OpenRAIL-M license, proposed by Responsible AI [29], is the most popular model-specific license on Hugging Face and enables legally enforceable responsible use. By accepting this license, licensees must adhere to the use-based restrictions stated by the licensor, and these restrictions must also apply to derivative works. With a multitude of different model licenses available, it becomes a challenging and tedious task to reuse them in bulk. It is therefore imperative to establish guidelines for selecting a license for models that are ready for query-based FL.

3.3.2 License Choosing Preferences. In query-based FL, the model community automatically reuses models contributed by users, which raises unique concerns about licensing of models:

- A model license ready for open FL paltforms should allow the **modification, combination and redistribution** of original works and any derived works; and
- **Sublicensing** right should be granted to lubricate the republication of derived works resulting from knowledge mining; note
- Some licenses enforce the source of the derived works to be **disclosed** and prohibit their **commercial use**, which hinders model selling [24]; and
- Some licenses are **copyleft** (marked with * in Table. 3), which means the derivatives must be licensed under the same license or a compatible one, leading to potential license conflicts and proliferation [49]; last
- All granted rights are preferably **irrevocable** by the licensors [144].

Besides, it is important to consider the licensing of two other components when building and reusing models: data and algorithms, which may have entirely different license terms. Here, we present several strategies for selecting licenses in query-based FL to minimize conflicts.

Preferences for Datasets or Databases: CC0-1.0, ODC-By > CC BY > C-UDA > LGPL-LR.

Our recommended licenses for training datasets and databases for query-based FL are CC0-1.0, ODC-By, and CC BY. CC BY-4.0 is preferred due to the grant of *Sui Generis Database Rights (Art.1c)*. CC0-1.0 and ODC-By are more permissive than CC BY since they do not require licensees to disclose any modifications made to the dataset or database, and CC0-1.0 does not require the declaration of the original license. Although these licenses do not explicitly grant sublicensing rights, they provide an automatic licensing policy for downstream recipients.

Table 3. Licenses for ML models available on Hugging Face with a focus on their rights, restrictions and enforcements, grouped by free software licenses, AI model licenses, free content or database licenses in descending order of number of models (GPL, BSD, LGPL, CC licenses with unspecified versions are excluded, the similar revisions are merged). ✓: Permited or Required, ✗ : Not Permited or Not Required, ! : Not Explicitly Permited, * : Copyleft License

Licenses	Modify / Merge	Redistribution	Sublicensing	Commercial Use	Patent Use	Trademark Use	State Changes	Disclose Source	Responsible-use Restrictions	License/Disclaim Preservation	# of Models	Licensed Materials / Remarks
Apache-2.0	✓	✓	✓	✓	✓	✗	✓	✗	✗	✓	23,519	BERT [33]
MIT	✓	✓	✓	✓	!	!	✗	✗	✗	✓	9,605	GPT-2 [140]
AFL-3.0	✓	✓	✓	✓	✓	✗	✓	✗	✗	✓	1,561	Italian-Legal-BERT [104]
*GPL-3.0	✓	✓	✗	✓	✓	✗	✓	✓	✗	✓	404	CKIP BERT Chinese
Artistic-2.0	✓	✓	✓	✓	✓	✗	✓	✗	✗	✓	331	Include original source
BSD-3-Clause&-Clear	✓	✓	✓	✓	!	!	✗	✗	✗	✓	209	CodeGen [124] / A MIT-style license
WTFPL-2.0	✓	✓	!	✓	!	!	✗	✗	✗	✗	131	A MIT-style permissive license
*AGPL-3.0	✓	✓	✗	✓	✓	✗	✓	✓	✗	✓	96	Distributed under AGPL only
Unlicense	✓	✓	!	✓	!	!	✗	✗	✗	✗	90	A MIT-style permissive license
BSL-1.0	✓	✓	✓	✓	!	!	✗	✗	✗	✓	60	A MIT-style permissive license
*GPL-2.0	✓	✓	✗	✓	!	!	✓	✓	✗	✓	34	Not compatible with GPL-3.0
BSD-2-Clause	✓	✓	✓	✓	!	!	✗	✗	✗	✓	34	A MIT-style permissive license
*LGPL-2.1&3.0	✓	✓	✗	✓	!	!	✓	✓	✗	✓	25	For software libraries
*OSL-3.0	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓	22	Linking is not derivative work
ECL-2.0	✓	✓	✓	✓	✓	✗	✓	✗	✗	✓	12	For education communities
*MPL-2.0	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓	9	State changes under MPL only
ISC	✓	✓	!	✓	!	!	✗	✗	✗	✓	8	MIT-style license w/o sublicense
Zlib	✓	✓	!	✓	!	!	✗	✗	✗	✓	8	Rename if modified
*Ms-PL	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	7	Weak copyleft license
*EPL-1.0&2.0	✓	✓	✓	✓	✓	!	✗	✓	✗	✓	6	Can link proprietary license code
NCSA	✓	✓	✓	✓	!	✗	✗	✗	✗	✓	4	Include full text of license
PostgreSQL	✓	✓	!	✓	!	!	✗	✗	✗	✓	2	A MIT-style license
OFL-1.1	✓	✓	✗	✓	!	!	✗	✗	✗	✓	2	For font software
*EUPL-1.1	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓	1	License of EU covers SaaS
LPPL-1.3c	✓	✓	✓	✓	!	✗	✓	✓	✗	✓	1	Covering stewardship transfer
CreativeML-OpenRAIL-M	✓	✓	✓	✓	✓	✗	✓	✗	✓	✓	3,590	Stable Diffusions v1 [150]
OpenRAIL	>Responsible AI License template, w/o full text										2,393	ControlNet [201]
BigScience-BLOOM-RAIL-1.0	✓	✓	✓	✓	✓	✗	✓	✗	✓	✓	196	BLOOM [155]
BigScience-OpenRAIL-M	>Same as BigScience-BLOOM-RAIL-1.0										155	A general version of 1.0
OpenRAIL++	>Same as CreativeML-OpenRAIL-M										72	Stable Diffusion v2 [150]
OPT-175B	✓	✗	✗	✗	✗	✗	✗	✗	✓	✓	≈ 66	OPT LLM [203]
SEER	>Same as OPT-175B, ban on reverse-engineer										/	SEER Vision Model [52]
CC-BY-4.0&3.0&2.5&2.0	✓	✓	!	✓	✗	✗	✓	✗	✗	✓	1,740	RoBERTa-SQuAD2.0 [141]
*CC-BY-SA-4.0&3.0	✓	✓	✗	✓	✗	✗	✓	✓	✗	✓	590	LEGAL-BERT [22]
*CC-BY-NC-SA-4.0&3.0	✓	✓	✗	✗	✗	✗	✓	✓	✗	✓	556	LayoutLMv3 [68]
CC-BY-NC-4.0&3.0&2.0	✓	✓	✗	✗	✗	✗	✓	✗	✗	✓	499	GALACTICA [166]
CC0-1.0	✓	✓	!	✓	✗	✗	✗	✗	✗	✗	165	BlueBERT [132]
CC-BY-NC-ND-4.0&3.0	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	21	NonCommercial, NoDerivatives
PDDL	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	16	Database-specific license
C-UDA	✓	✓	✓	✗	!	!	✗	✗	✓	✓	13	Data for computational use only
*LGPL-LR	✓	✓	✗	✓	!	!	✓	✓	✗	✓	12	LGPL for linguistic resources
*GFDL	>Same as GPL, a free document license										12	txtai-wikipedia
CC-BY-ND-4.0	✓	✗	✗	✓	✗	✗	✓	✗	✗	✓	11	Disallow making derivatives
ODC-By	✓	✓	✗	✓	✗	✗	✗	✗	✗	✓	7	Automatic relicensing
*ODbL	✓	✓	✗	✓	✗	✗	✓	✓	✗	✓	6	Automatic relicensing

C-UDA is an alternative license that grants sublicensing rights, but it includes additional usage restrictions that limit its application to computational use only, which indicates commercial use is not allowed. To avoid license proliferation, it is not recommended to use any data under copyleft licenses for building models, as the resulted models could be seen as remixing and making derivatives of the original datasets, leading to potential conflicts between licenses. Among them, LGPL-LR is an exception because it contains an exemption clause for *work that uses the Linguistic Resource (Art.3)*, which is suitable for end-to-end training, fine-tuning, and embedding. But it is worth noting that the embedded representations may be considered *translated straightforwardly into another language (Art.0)*, which falls within the scope of LGPL-LR license.

An example of license proliferation is LEGAL-BERT [22], which was trained on data from the Case Law Access Project [16] (licensed under CC BY-SA 4.0). This restricts LEGAL-BERT to the same license and prevents further model reusing on datasets or models licensed under incompatible copyleft licenses, such as LGPL-LR and GPL.

Preferences for Software: Apache-2.0, AFL-3.0, Artistic-2.0, ECL-2.0 > MIT, BSD-3-Clause&-Clear, BSL-1.0, BSD-2-Clause, NCSA ≈ Ms-PL > WTFPL-2.0, Unlicense, ISC, Zlib, PostgreSQL.

Our top recommended software licenses for training and reusing models are Apache, AFL, Artistic, and ECL. These permissive licenses allow modification and sublicensing, explicitly grant the use of patents and permit commercial use, and do not require the disclosure of the source code but only the stating of any changes made to the original work.

The next set of recommendations are MIT, BSD, BSL, and NCSA. These licenses do not explicitly grant patent rights but instead, do not require the stating of modifications made to the original work, thus avoiding the tedious task of tracking model reusing or incremental training procedures. Ms-PL offers two advantages simultaneously, but it is a weak copyleft license that requires the modified source code to also be licensed under Ms-PL, and the derivative object code to be compliant with a license compatible with Ms-PL. However, open software licenses do not provide a clear definition for software-generated outputs such as models. It is unclear whether models are considered a portion of the software, and whether they are in source code form or object code form. This ambiguity makes it difficult to determine the applicable clauses for models.

Our latest recommended licenses include WTFPL, Unlicense, ISC, Zlib, and PostgreSQL. These licenses are very permissive and allow almost anything without restrictions. However, it's important to note that these licenses do not explicitly grant sublicensing rights and patent, which can lead to ambiguity in interpreting the license clauses. For the avoidance of doubt, copyleft licenses such as GPL, AGPL, LGPL, OSL, MPL, EPL, and EUPL are not recommended, despite the loophole that they do not have a specific definition for models. Furthermore, although some of those copyleft licenses can be made compliant with others, we recommend isolating the software license from the resulting models to preserve the freedom to use the models further (e.g., close-source or relicense).

Preferences for Models: Apache-2.0, AFL-3.0, Artistic-2.0, ECL-2.0 > OpenRAIL and its derivatives

There are two recommended choices for model licenses for query-based FL. The first is permissive open software licenses like Apache, AFL, Artistic, and ECL. The second is open model-specific licenses like OpenRAIL and its derivatives. As shown in Table 3, the main difference between the two choices is that OpenRAIL offers additional user behavioral restriction clauses and enforces these restrictions via a copyleft-style agreement. For example, CreativeML OpenRAIL-M license claims *Therefore You cannot use the Model and the Derivatives of the Model for the specified restricted uses ... You shall require all of Your users who use the Model or a Derivative of the Model to comply with the terms of this paragraph.* The restricted uses include actions that could cause harm, provide medical advice, generate or disseminate verifiably false information, and more. So, the model owners may adopt these licenses for the purpose of responsible model use.

However, in practice, such discrimination of user behavior cannot completely guarantee that the models will not be misused, and may potentially compromise the openness of the models [53, 134]. The user behavioral restrictions in licenses can be compared to manufacturers prohibiting the use of their laptops for hacking, and furthermore, the vendors can be held jointly and severally liable for any future violations, which is unreasonable and may lead to the licensed materials becoming closed source. Furthermore, to enable remote control for the responsible use of AI, CreativeML OpenRAIL-M includes the clause *You shall undertake reasonable efforts to*

[16] https://case.law

use the latest version of the Model, which requires licensees to keep up with the updates of the original work and may render their prior development efforts useless. Therefore, traditional permissive licenses, which follow worse-is-better design philosophy [41], are good choices for model licensing in query-based FL, as they promote openness and facilitate the sharing of publicly contributed models.

The remaining model licenses, OPT-175B and SEER, are proprietary licenses that allow licensees to use and reproduce the licensed models subject to certain restrictions. Given that their granted rights are revocable, we do not recommend using any content of works and derivatives under these licenses in query-based FL.

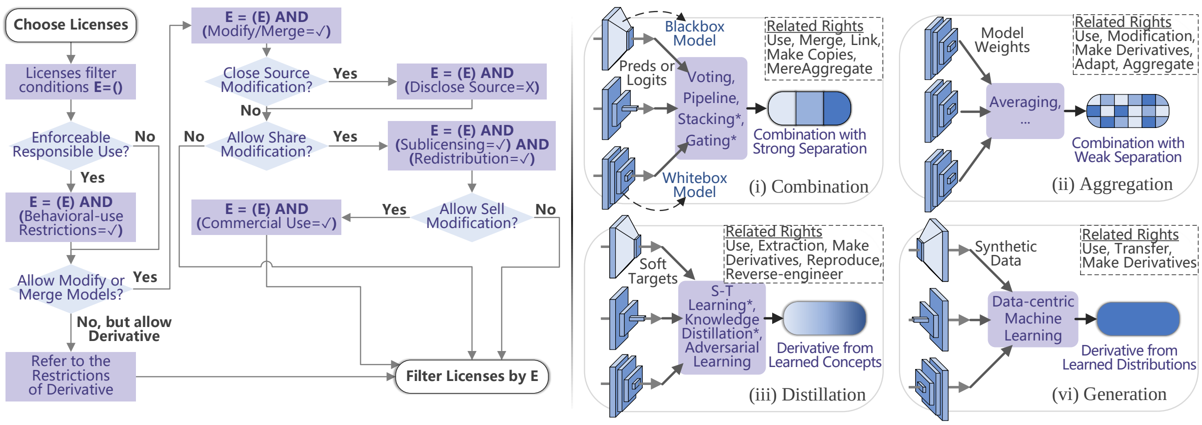
It is worth noting that the above discussion only deals with the licenses of inputs for open FL platforms, which aim to provide legal compliance and freedom of outputs as much as possible, but does not involve the copyright issue for the outputs. In fact, except for some public domain dedication licenses like CC0-1.0, PDDL, Unlicense, and WTFPL, most licenses only grant non-exclusive rights for use and distribution, and the original copyright and attribution are retained by the licensors. Whether the reused models are copyrightable is crucial for incentivizing model sharing and mining, so we will elaborate on this topic in the next section.

3.3.3 Copyright of Reused Models. Software and computer code are indisputably copyrightable, but what about computer-generated content such as distillation and ensembles of models? The copyrightable of a computer-generated work is controversial, which may depend on such as the level of creativity and originality and *presence of at least minimal human creative effort at the time the work is produced* [127]. According to this definition, programmers who engage in model design and training meet the threshold requirements of copyrightability and own the copyright of the model. That is why all the licenses listed in Table 3 contain claims of copyright. But the debating point is whether the reused models also copyrightable? Unfortunately, there is no universal answer to this question as it can depend on the specific case and fact pattern. The crux is whether the efforts involved in reusing the model meet the minimum creative requirements for copyrightability. For example, if we simply stack two models end-to-end, it may not meet the threshold for copyrightability. However, if we improve a basis model using distilled knowledge from other domains, that would be more likely to meet the requirements for copyrightability. Except for copyrightability, the authorship of a reused model is also open to controversy, as it depends on whose *original intellectual conceptions* the work embodies, and joint authorship is also possible [63].

The determination of copyrightability and authorship of computer-generated content is an open issue that needs to be addressed through corresponding legislation [63, 127]. The possible answers to the question of authorship of computer-generated models are model authors, model users, data owners, any combination of them, or no one [63]. Licensors can also make efforts to clarify this issue by including relevant claims in their licenses. For example, the license of Stable Diffusion [150] explicitly states that *Licensor claims no rights in the Output You generate using the Model.* Similarly, ChatGPT [17], even though it is a proprietary software of OpenAI company, its sharing & publication policy [18] states *The published content is attributed to your name or company.* Therefore, we are free to use their generated content for model reusing and can claim the copyright of reused models. On the contrary, the licenses of OPT [203] and SEER [52] do not grant any copyright for the data produced by the licensed software. Given that, we should avoid using their derivatives and generated content in query-based FL to prevent copyright infringement. Once we obtain the right to relicense the modification models, the choice of a new license depends on the application scenario of models. We provide a flowchart in Fig. 4(a) to guide license selection. For now, we have provided a comprehensive perspective and suggestions regarding the regulations and legal issues related to batch model reusing with only one piece missing: the definition of terms and corresponding clauses for different reusing mechanisms in different licenses. We leave this content in the following section.

3.3.4 Batch Model Reuse Mechanisms. The definition of terms for model reusing in different licenses is a novel and interesting issue that is rarely discussed. For example, interpreting model reusing as creating derivatives

[17] https://openai.com/blog/chatgpt [18] https://openai.com/policies/sharing-publication-policy



(a) Reused Model Licenses Choosing Flowchart (b) ML Batch Model Reuse Mechanisms

Fig. 4. flowchart

or combinations would involve different clauses in the licenses. Therefore, instead of summarizing the batch model reuse mechanisms from a technological and algorithmic aspect, we group these mechanisms based on the classification of their resulting outputs for ease of justifying license clauses. As shown in Fig. 4(b), there are four categories of batch model reuse mechanisms: **Combination, Amalgamation, Distillation, and Generation**, each resulting in different forms of outputs.

Combination [206] is a straightforward way to reuse batch of models (aka. base learners), in which multiple models jointly contribute to the output by combination strategies such as averaging, voting, learning [70, 180]. For regression estimates, averaging can improve the generalization by taking the mean of the outputs of all weak learners in a population. Additionally, the outputs of each learner can be weighted by extra parameters [135], which can be determined by stacking estimators [180], Bayes approach [28] or backpropagation of gating networks [70]. Voting is a workaround for classification tasks and also applicable for stacking and gating. Both stacking and gating rely on an additional holdout or validation dataset for calculating extra parameters, marked as * in Fig. 4(b). The difference is that gating can adapt the weights of each model's estimation based on the inputs, providing better generalizability performance of the combined model.

There are many advantages of combination mechanisms from the perspective of FL. First, the input spaces of base models can be unaligned, which is ideal for the scenario of vertical FL [182] where each client may have inconsistent features in their data. Secondly, especially for query-based FL, it can simultaneously support multiple types and heterogeneous models, which means that it does not rely on any prior assumptions of the models, such as whether they are DNNs or decision trees, released with weights (whitebox) or binary forms (blackbox). Thirdly, the tasks of models can be different if we pipeline the base models end-to-end, which is usually overlooked as a combination mechanism of models. Pipelining can fully leverage the transferability of models to solve previously unexplored ML problems. For instance, Gao *et al.* [43] proposed a zero-shot dense retrieval system named HyDE by pipelining a natural language generation (NLG) model [129] and a natural language understanding (NLU) model [69]. The generated content, which may lack factual grounding, from the NLG model is used as query embeddings to facilitate real document retrieval by the NLU model. Through query-based FL, we can query a vicarious NLG model for a novel scenario, such as ProGen [118] for protein sequences generation, and quickly adapt this system to proteomics. Not limited to that, we can query a batch of NLG models by a well-chosen filter condition and then combine models through averaging or gating to significantly expand the exploration space for knowledge discovery. Lastly, the combinated models have strong separation from each other, meaning that we can add or remove a batch of models without significant changes to the remaining ones. Furthermore, combination mechanisms do not rely on the transparency of models and support blackbox sharing. Thus, the base model can establish loose connections with other models only through run scripts, providing revocability of such combination and circumvention of the restrictions of licenses.

On the other hand, instead of being treated as a challenge for FL [115], the statistical heterogeneity and model heterogeneity nature of these crowdsourced models can actually enrich population diversity, which is crucial for creating a good ensemble [117, 128].

Amalgamation involves combining models through model parameters granularity operations, such as median [12, 137] and coordinate-wise averaging with consideration of heterogeneity [100, 120], security [163], scalability [145], matching [173, 192], specificality [54], generalizability [139], resulting in a combination with weak separation. This reusing approach is widely used in FL works and is often referred to as "aggregation" procedures for local models. Here, we avoid using the term "aggregation" to distinguish it from "combination". The latter is often used interchangeably with "combination" in software licenses (e.g. Artistic, GPL).

FedAvg [120] is the most popular model averaging method in FL with many follow-up works. For instance, Sun *et al.* [163] proposed applying norm thresholding of local model updates to defend against backdoor attacks. Similarly, Blanchard *et al.* [12] proposed using more robust median-based amalgamation strategies for resilience against Byzantine behavior. Consider the ordering of parameters, Wang *et al.* [173] match and average the neural parameters layer-wise across local models, based on their similarities. However, these methods require multiple rounds of communication to converge, which is not applicable in a query-based FL setting.

Another group of studies leverages Bayesian nonparametrics to learn the shared global latent structures among local models [84, 194, 195]. These methods, known as Model Fusion, can identify distributions of neural components across local models and only fuse the components with the same distribution, which can be explained as a model compression between FedAvg (coordinate-wise averaging) and combination (w/o averaging). However, the model fusion strategies rely on multiple communication rounds to boost the fusion efficiency, and the model performance of one-shot fusion is even worse than that of Ensemble.

It is worth noting that our taxonomy is based on the form of the resulting model, which may not be entirely consistent with the terminology used in the technical perspective. For example, Bayes Model Averaging (BMA) [28] estimates posterior probabilities of each model given the observed data, which results in a separable weighted model. Therefore, it should be classified as Combination instead of Amalgamation like FedAvg. This novel taxonomy method is useful for analyzing compatibility with licenses. For instance, coordinate-wise operations and fusion of model parameters generate fine-grained combinations of models that are almost irreversible, which corresponds to clauses such as adapt, modify, dynamic link, etc., in software licenses.

Distillation was initially proposed by Hinton *et al.* [65] to transfer knowledge from a batch of independently trained neural network models (Specialists) to create a new Generalist model. Their motivation was to explore the parallelization of training of specialists and improve the efficiency of distributed NNs modeling[32]. Each specialist only learns fine-grained distinctions of a subset of classes, which is very similar to the non-IID setting in FL [94]. Therefore, it is natural to extend the knowledge distillation to FL filed [23, 50, 74, 88, 95, 106, 161, 181].

Data-Free Knowledge Distillation(DFKD) [111]

Generation

CreativeML Open RAIL-M: we added use-based restrictions not permitting the use of the Model in very specific scenarios, in order for the licensor to be able to enforce the license in case potential misuses of the Model may occur.

Table 4. FL model reuse; Server Client Knowledge Distillation (KD) Challenges: **E**fficiency **H**eterogeneity **P**rivacy

FL Studies	**C**ombination	**A**malgamation	**D**istillation	**G**eneration	Process	Goals
FedAvg [120]		Model Avg			[A]*N	EH
FedAD [50]			KD **Attention, Logits**		[D]*1	HP
$FedKD_1$ [51]			KD **Weighted Logits**		[D]*1	EHP
FedMD [88]			KD **Logits Avg**		[D]*N	H
FedMD-NFDP [161]			KD **Logits/Softmax/Argmax Avg**		[D]*N	HP
RHFL [40]			KD **Weighted Logits**		[D]*N	EH
One-Shot FL [55]	Output Avg		KD **Softmax**		[CD]*1	EP
FedDF [106]		Model Avg	KD **Logits Avg**		[AD]*N	HP
PerAda [183]		Adapter Avg	KD **Logits Avg**		[AD]*N	EHP
FedDistill [74]		Model Avg	KD *Softmax*		[DA]*N	H
FedFusion [189]		Model Avg	KD **Hidden**		[DA]*N	E
FedMLB [80]		Model Avg	KD *Softmax, Scaled Softmax*		[DA]*N	EH
FedAlign [121]		Model Avg	KD *Lipschitz Constants* [157]		[DA]*N	EH
FedNTD [87]		Model Avg	KD *Not-True Classes Softmax*		[DA]*N	H
FedKC [174]		Model Avg	KD *Clustered Hidden Avg*		[DA]*N	HP
MOON [96]		Model Avg	Contrastive Learning *Hidden*, *Hidden*		[DA]*N	EH
FedCAD [62]		Model Avg	KD **Class-Wise Softmax**		[DA]*N	H
FCCL [67]			Contrastive Learning **Logits Avg** Continual Learning *Logits*		[DD]*N	H
GFL [27]		Model Avg		Synthetic Data	G[A]*N	HP
FedSage+ [200]		Gradient Avg Model Avg		Synthetic Node	[GA]*N [A]*N	H
Fed-ZDAC [59]		Model Avg		Synthetic Data of Global Model	[GA]*N	HP
Fed-ZDAS [59]		Model Avg		Synthetic Data of Local Models	[GA]*N	HP
DENSE [199]	Collection		KD **Logits**	Synthetic Data	[GCD]*N	HP
FedBE [23]		Model Avg	KD **Softmax Avg**	Synthetic Model	[AGD]*1	H
FedAUX [152]		Model Avg	Contrastive Learning **Hidden** KD **Weighted Logits**		D[AD]*N D[AD]*1	HP
$FedKD_2$ [181]		Gradient Avg	KD *Hidden, Attention, Logits* KD *Hidden, Attention, Logits*		[DAD]*N	EH
FedKT [95]	Voting Voting		KD **Argmax** KD **Argmax**		[CDCD]*1	HP
FedMatch [71]	Voting	Model Avg	KD **Argmax**		[CDAA]*N [CDA]*N	EH
FedGen [210]		Model Avg	KD **Softmax** KD **Logits Avg**	Augmented Data	D[GAD]*N	EHP

3.4 How to Protect Models

ACKNOWLEDGMENTS

ACK.

Table 5. model reuse

	Combinated Work with Strong Separation	Combinated Work with Weak Separation	Derivative Work from Concepts	Derivative Work from Distributions
Apache-2.0	Separable -> Independent Work	Modify -> Derivative	✗	✗
MIT	✗	✗	✗	✗
AFL-3.0	✗	Modify -> Derivative	✗	✗
GPL-3.0	Blackbox: Aggregate -> Independent Work Other: Link -> Modified Version	Modify -> Covered Work	Output no consititutes a covered work -> Independent Work	Output no consititutes a covered work -> Independent Work
Artistic-2.0	Blackbox: MereAggregation -> Independent Work Other: Link -> Modified Version	Aggregate -> Modified Version	✗	✗
BSD-3-Clause	Blackbox: Rredsitribution in binary forms -> ✗ Other: Redistribution of source code -> ✗	✗	✗	✗
WTFPL-2.0	✗	✗	✗	✗
OpenRAIL Licenses	Transfer of patterns of output -> Derivative	Transfer of patterns of weights -> Derivative	Transfer of patterns of activations -> Derivative	Transfer of patterns of output -> Derivative
Creative Commons Licenses	Reproduce -> Adapted Material	Adapt -> Adapated Material	✗	✗
CC0-1.0	Reproduce -> Independent Work	Adapt -> Independent Work	✗	✗

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*. 265–283.

[2] Sawsan AbdulRahman, Hanine Tout, Azzam Mourad, and Chamseddine Talhi. 2020. FedMCCS: multicriteria client selection model for optimal IoT federated learning. *IEEE Internet of Things Journal* 8, 6 (2020), 4723–4735.

[3] Sawsan AbdulRahman, Hanine Tout, Hakima Ould-Slimane, Azzam Mourad, Chamseddine Talhi, and Mohsen Guizani. 2020. A survey on federated learning: The journey from centralized to distributed on-site learning and beyond. *IEEE Internet of Things Journal* 8, 7 (2020), 5476–5497. https://doi.org/10.1109/JIOT.2020.3030072

[4] Accountability Act. 1996. Health insurance portability and accountability act of 1996. *Public law* 104 (1996), 191.

[5] Shaashwat Agrawal, Sagnik Sarkar, Ons Aouedi, Gokul Yenduri, Kandaraj Piamrat, Mamoun Alazab, Sweta Bhattacharya, Praveen Kumar Reddy Maddikunta, and Thippa Reddy Gadekallu. 2022. Federated learning for intrusion detection system: Concepts, challenges and future directions. *Computer Communications* (2022). https://doi.org/10.1016/j.comcom.2022.09.012

[6] Mamoun Alazab, Swarna Priya RM, M Parimala, Praveen Kumar Reddy Maddikunta, Thippa Reddy Gadekallu, and Quoc-Viet Pham. 2021. Federated learning for cybersecurity: concepts, challenges, and future directions. *IEEE Transactions on Industrial Informatics* 18, 5 (2021), 3501–3509. https://doi.org/10.1109/TII.2021.3119038

[7] Mohammed Aledhari, Rehma Razzak, Reza M Parizi, and Fahad Saeed. 2020. Federated learning: A survey on enabling technologies, protocols, and applications. *IEEE Access* 8 (2020), 140699–140725. https://doi.org/10.1109/ACCESS.2020.3013541

[8] Rodolfo Stoffel Antunes, Cristiano André da Costa, Arne Küderle, Imrana Abdullahi Yari, and Björn Eskofier. 2022. Federated learning for healthcare: Systematic review and architecture proposal. *ACM Transactions on Intelligent Systems and Technology (TIST)* 13, 4 (2022), 1–23. https://doi.org/10.1145/3501813

[9] Edmond Awad, Sohan Dsouza, Richard Kim, Jonathan Schulz, Joseph Henrich, Azim Shariff, Jean-François Bonnefon, and Iyad Rahwan. 2018. The moral machine experiment. *Nature* 563, 7729 (2018), 59–64. https://doi.org/10.1038/s41586-018-0637-6

[10] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2020. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2938–2948.

[11] Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Titouan Parcollet, Pedro PB de Gusmão, and Nicholas D Lane. 2020. Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390* (2020).

[12] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. 2017. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in Neural Information Processing Systems (NIPS)* 30 (2017).

[13] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konecny, Stefano Mazzocchi, H Brendan McMahan, et al. 2019. Towards Federated Learning at Scale: System Design. In *Proceedings of the 2nd SysML Conference*.

[14] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 1175–1191.

[15] Parimala Boopalan, Swarna Priya Ramu, Quoc-Viet Pham, Kapal Dev, Praveen Kumar Reddy Maddikunta, Thippa Reddy Gadekallu, Thien Huynh-The, et al. 2022. Fusion of federated learning and industrial Internet of Things: A survey. *Computer Networks* (2022), 109048. https://doi.org/10.1016/j.comnet.2022.109048

[16] Nader Bouacida and Prasant Mohapatra. 2021. Vulnerabilities in federated learning. *IEEE Access* 9 (2021), 63229–63249. https://doi.org/10.1109/ACCESS.2021.3075203

[17] Pearl Brereton, David Budgen, Keith Bennnett, Malcolm Munro, Paul Layzell, Linda MaCaulay, David Griffiths, and Charles Stannett. 1999. The future of software. *Commun. ACM* 42, 12 (1999), 78–84. https://doi.org/10.1145/322796.322813

[18] Theodora S Brisimi, Ruidi Chen, Theofanie Mela, Alex Olshevsky, Ioannis Ch Paschalidis, and Wei Shi. 2018. Federated learning of predictive models from federated electronic health records. *International journal of medical informatics* 112 (2018), 59–67. https://doi.org/10.1016/j.ijmedinf.2018.01.007

[19] Sebastian Caldas, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. 2018. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097* (2018).

[20] Hanqun Cao, Cheng Tan, Zhangyang Gao, Guangyong Chen, Pheng-Ann Heng, and Stan Z Li. 2022. A survey on generative diffusion model. *arXiv preprint arXiv:2209.02646* (2022).

[21] Zheng Chai, Ahsan Ali, Syed Zawad, Stacey Truex, Ali Anwar, Nathalie Baracaldo, Yi Zhou, Heiko Ludwig, Feng Yan, and Yue Cheng. 2020. Tifl: A tier-based federated learning system. In *Proceedings of the 29th international symposium on high-performance parallel and distributed computing*. 125–136. https://doi.org/10.1145/3369583.3392686

[22] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The Muppets straight out of Law School. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. 2898–2904. https://doi.org/10.18653/v1/2020.findings-emnlp.261

[23] Hong-You Chen and Wei-Lun Chao. 2021. FedBE: Making Bayesian Model Ensemble Applicable to Federated Learning. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*.

[24] Lingjiao Chen, Paraschos Koutris, and Arun Kumar. 2019. Towards model-based pricing for machine learning in a data marketplace. In *Proceedings of the 2019 International Conference on Management of Data*. 1535–1552. https://doi.org/10.1145/3299869.3300078

[25] Yanjiao Chen, Baolin Zheng, Zihan Zhang, Qian Wang, Chao Shen, and Qian Zhang. 2020. Deep learning on mobile and embedded devices: State-of-the-art, challenges, and future directions. *ACM Computing Surveys (CSUR)* 53, 4 (2020), 1–37. https://doi.org/10.1145/3398209

[26] Kewei Cheng, Tao Fan, Yilun Jin, Yang Liu, Tianjian Chen, Dimitrios Papadopoulos, and Qiang Yang. 2021. Secureboost: A lossless federated learning framework. *IEEE Intelligent Systems* 36, 6 (2021), 87–98. https://doi.org/10.1109/MIS.2021.3082561

[27] Yihang Cheng, Lan Zhang, and Anran Li. 2023. GFL: Federated Learning on Non-IID data via Privacy-preserving Synthetic data. In *2023 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 61–70. https://doi.org/10.1109/PERCOM56429.2023.10099110

[28] Bertrand Clarke. 2003. Comparing Bayes model averaging and stacking when model approximation error cannot be ignored. *Journal of Machine Learning Research* 4, Oct (2003), 683–712.

[29] Danish Contractor, Daniel McDuff, Julia Katherine Haines, Jenny Lee, Christopher Hines, Brent Hecht, Nicholas Vincent, and Hanlin Li. 2022. Behavioral use licensing for responsible AI. In *2022 ACM Conference on Fairness, Accountability, and Transparency*. 778–788. https://doi.org/10.1145/3531146.3533143

[30] Robert David, Jared Duke, Advait Jain, Vijay Janapa Reddi, Nat Jeffries, Jian Li, Nick Kreeger, Ian Nappier, Meghna Natraj, Tiezhen Wang, et al. 2021. Tensorflow lite micro: Embedded machine learning for tinyml systems. *Proceedings of Machine Learning and Systems* 3 (2021), 800–811.

[31] Ittai Dayan, Holger R Roth, Aoxiao Zhong, Ahmed Harouni, Amilcare Gentili, Anas Z Abidin, Andrew Liu, Anthony Beardsworth Costa, Bradford J Wood, Chien-Sung Tsai, et al. 2021. Federated learning for predicting clinical outcomes in patients with COVID-19. *Nature medicine* 27, 10 (2021), 1735–1743. https://doi.org/10.1038/s41591-021-01506-3

[32] Jeffrey Dean, Greg S Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V Le, Mark Z Mao, Marc'Aurelio Ranzato, Andrew Senior, Paul Tucker, et al. 2012. Large scale distributed deep networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems (NeurIPS)*. 1223–1231.

[33] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[34] Zhaoyang Du, Celimuge Wu, Tsutomu Yoshinaga, Kok-Lim Alvin Yau, Yusheng Ji, and Jie Li. 2020. Federated learning for vehicular internet of things: Recent advances and open issues. *IEEE Open Journal of the Computer Society* 1 (2020), 45–61. https://doi.org/10.1109/OJCS.2020.2992630

[35] Moming Duan, Duo Liu, Xianzhang Chen, Renping Liu, Yujuan Tan, and Liang Liang. 2020. Self-balancing federated learning with global imbalanced data in mobile systems. *IEEE Transactions on Parallel and Distributed Systems (TPDS)* 32, 1 (2020), 59–71.

[36] Moming Duan, Duo Liu, Xianzhang Chen, Yujuan Tan, Jinting Ren, Lei Qiao, and Liang Liang. 2019. Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications. In *Proceedings of the IEEE 37th International Conference on Computer Design (ICCD)*. IEEE, 246–254.

[37] Cynthia Dwork. 2006. Differential privacy. In *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II 33*. Springer, 1–12. https://doi.org/10.1007/11787006_1

[38] Ahmed El Ouadrhiri and Ahmed Abdelhadi. 2022. Differential privacy for deep and federated learning: A survey. *IEEE Access* 10 (2022), 22359–22380. https://doi.org/10.1109/ACCESS.2022.3151670

[39] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. 2020. Local model poisoning attacks to byzantine-robust federated learning. In *Proceedings of the 29th USENIX Conference on Security Symposium*. 1623–1640.

[40] Xiuwen Fang and Mang Ye. 2022. Robust federated learning with noisy and heterogeneous clients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10072–10081.

[41] Richard Gabriel. 1991. The rise of "worse is better". *Lisp: Good News, Bad News, How to Win Big* 2, 5 (1991).

[42] Attlee M Gamundani and Lucas M Nekare. 2018. A review of new trends in cyber attacks: A zoom into distributed database systems. In *2018 IST-Africa Week Conference (IST-Africa)*. IEEE, Page–1.

[43] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise Zero-Shot Dense Retrieval without Relevance Labels. *arXiv preprint arXiv:2212.10496* (2022).

[44] Gary N Geller and Woody Turner. 2007. The model web: a concept for ecological forecasting. In *2007 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2469–2472. https://doi.org/10.1109/IGARSS.2007.4423343

[45] Craig Gentry. 2009. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing (STOC)*. 169–178. https://doi.org/10.1145/1536414.1536440

[46] Robin C Geyer, Tassilo Klein, and Moin Nabi. 2017. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557* (2017).

[47] Bimal Ghimire and Danda B Rawat. 2022. Recent advances on federated learning for cybersecurity and cybersecurity for federated learning for internet of things. *IEEE Internet of Things Journal* (2022). https://doi.org/10.1109/JIOT.2022.3150363

[48] Ross Girshick. 2015. Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision*. 1440–1448.

[49] Robert W Gomulkiewicz. 2009. Open Source License Proliferation: Helpful Diversity or Hopeless Confusion? *Washington University Journal of Law & Policy* 30, 1 (2009).

[50] Xuan Gong, Abhishek Sharma, Srikrishna Karanam, Ziyan Wu, Terrence Chen, David Doermann, and Arun Innanje. 2021. Ensemble attention distillation for privacy-preserving federated learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 15076–15086.

[51] Xuan Gong, Abhishek Sharma, Srikrishna Karanam, Ziyan Wu, Terrence Chen, David Doermann, and Arun Innanje. 2022. Preserving privacy in federated learning with ensemble cross-domain knowledge distillation. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, Vol. 36. 11891–11899. https://doi.org/10.1609/aaai.v36i11.21446

[52] Priya Goyal, Quentin Duval, Isaac Seessel, Mathilde Caron, Mannat Singh, Ishan Misra, Levent Sagun, Armand Joulin, and Piotr Bojanowski. 2022. Vision models are more robust and fair when pretrained on uncurated images without supervision. *arXiv preprint arXiv:2202.08360* (2022).

[53] Eli Greenbaum. 2016. The Non-Discrimination Principle in Open Source Licensing. *Cardozo Law Review* 37, 4 (2016), 1297–1344.

[54] Gautham Krishna Gudur, Bala Shyamala Balaji, and Satheesh K Perepu. 2020. Resource-constrained federated learning with heterogeneous labels and models. In *KDD 2020 Workshop on Artificial Intelligence of Things*.

[55] Neel Guha, Ameet Talwalkar, and Virginia Smith. 2018. One-shot federated learning. In *NeurIPS 2018 Workshop on Machine Learning on the Phone and other Consumer Devices*.

[56] Ruchi Gupta and Tanweer Alam. 2022. Survey on federated-learning approaches in distributed environment. *Wireless Personal Communications* 125, 2 (2022), 1631–1652. https://doi.org/10.1007/s11277-022-09624-y

[57] Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, et al. 2021. Pre-trained models: Past, present and future. *AI Open* 2 (2021), 225–250. https://doi.org/10.1016/j.aiopen.2021.08.002

[58] Lucjan Hanzlik, Yang Zhang, Kathrin Grosse, Ahmed Salem, Maximilian Augustin, Michael Backes, and Mario Fritz. 2021. Mlcapsule: Guarded offline deployment of machine learning as a service. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3300–3309.

[59] Weituo Hao, Mostafa El-Khamy, Jungwon Lee, Jianyi Zhang, Kevin J Liang, Changyou Chen, and Lawrence Carin Duke. 2021. Towards fair federated learning with zero-shot data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 3310–3319.

[60] Andrew Hard, Kanishka Rao, Rajiv Mathews, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2018. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604* (2018).

[61] Chaoyang He, Songze Li, Jinhyun So, Xiao Zeng, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, et al. 2020. FedML: A research library and benchmark for federated machine learning. In *NeurIPS 2020 Workshop on Scalability, Privacy, and Security in Federated Learning*.

[62] Yuting He, Yiqiang Chen, Xiaodong Yang, Yingwei Zhang, and Bixiao Zeng. 2022. Class-wise adaptive self distillation for heterogeneous federated learning. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, Vol. 22. 12967–12968. https://doi.org/10.1609/aaai.v36i11.21620

[63] Samantha Fink Hedrick. 2019. I Think, Therefore I Create: Claiming Copyright in the Outputs of Algorithms. *New York University Journal of Intellectual Property & Entertainment Law (JIPEL)* 8, 2 (2019), 324–375.

[64] Ehsan Hesamifard, Hassan Takabi, Mehdi Ghasemi, and Rebecca N Wright. 2018. Privacy-preserving machine learning as a service. *Proc. Priv. Enhancing Technol.* 2018, 3 (2018), 123–142. https://doi.org/10.1515/popets-2018-0024

[65] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2014. Distilling the Knowledge in a Neural Network. In *NIPS Deep Learning and Representation Learning Workshop*.

[66] Li Huang, Andrew L Shea, Huining Qian, Aditya Masurkar, Hao Deng, and Dianbo Liu. 2019. Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records. *Journal of biomedical informatics* 99 (2019), 103291. https://doi.org/10.1016/j.jbi.2019.103291

[67] Wenke Huang, Mang Ye, and Bo Du. 2022. Learn from others and be yourself in heterogeneous federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10143–10153.

[68] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*. 4083–4091. https://doi.org/10.1145/3503161.3548112

[69] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Towards unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118* (2021).

[70] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation* 3, 1 (1991), 79–87. https://doi.org/10.1162/neco.1991.3.1.79

[71] Wonyong Jeong, Jaehong Yoon, Eunho Yang, and Sung Ju Hwang. 2021. Federated semi-supervised learning with inter-client consistency & disjoint learning. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*.

[72] Shaoxiong Ji, Teemu Saravirta, Shirui Pan, Guodong Long, and Anwar Walid. 2021. Emerging trends in federated learning: From model fusion to federated x learning. *arXiv preprint arXiv:2102.12920* (2021).

[73] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*. 675–678. https://doi.org/10.1145/2647868.2654889

[74] Donglin Jiang, Chen Shan, and Zhihui Zhang. 2020. Federated learning algorithm based on knowledge distillation. In *2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*. IEEE, 163–167. https://doi.org/10.1109/ICAICE51518.2020.00038

[75] Xiaopeng Jiang, Han Hu, Thinh On, Phung Lai, Vijaya Datta Mayyuri, An Chen, Devu M Shila, Adriaan Larmuseau, Ruoming Jin, Cristian Borcea, et al. 2022. FLSys: Toward an Open Ecosystem for Federated Learning Mobile Apps. *IEEE Transactions on Mobile Computing* (2022). https://doi.org/10.1109/TMC.2022.3223578

[76] Xiao Jin, Pin-Yu Chen, Chia-Yi Hsu, Chia-Mu Yu, and Tianyi Chen. 2021. CAFE: Catastrophic data leakage in vertical federated learning. *Advances in Neural Information Processing Systems (NeurIPS)* 34 (2021), 994–1006.

[77] Anna Jobin, Marcello Ienca, and Effy Vayena. 2019. The global landscape of AI ethics guidelines. *Nature Machine Intelligence* 1, 9 (2019), 389–399. https://doi.org/10.1038/s42256-019-0088-2

[78] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning* 14, 1–2 (2021), 1–210. https://doi.org/10.1561/2200000083

[79] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 5132–5143.

[80] Jinkyu Kim, Geeho Kim, and Bohyung Han. 2022. Multi-level branched regularization for federated learning. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*. PMLR, 11058–11073.

[81] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492* (2016).

[82] Nicolas Kourtellis, Kleomenis Katevas, and Diego Perino. 2020. FLaaS: Federated learning as a service. In *Proceedings of the 1st workshop on distributed machine learning*. 7–13. https://doi.org/10.1145/3426745.3431337

[83] Viraj Kulkarni, Milind Kulkarni, and Aniruddha Pant. 2020. Survey of personalization techniques for federated learning. In *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*. IEEE, 794–797. https://doi.org/10.1109/WorldS450073.2020.9210355

[84] Thanh Chi Lam, Nghia Hoang, Bryan Kian Hsiang Low, and Patrick Jaillet. 2021. Model Fusion for Personalized Learning. In *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 5948–5958.

[85] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436.

[86] Yann LeCun, Corinna Cortes, and CJ Burges. 2010. MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann. lecun. com/exdb/mnist* 2 (2010).

[87] Gihun Lee, Minchan Jeong, Yongjin Shin, Sangmin Bae, and Se-Young Yun. 2022. Preservation of the global knowledge by not-true distillation in federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[88] Daliang Li and Junpu Wang. 2019. FedMD: Heterogenous federated learning via model distillation. *NeurIPS 2019 Workshop on Federated Learning for User Privacy and Data Confidentiality* (2019).

[89] Li Li, Moming Duan, Duo Liu, Yu Zhang, Ao Ren, Xianzhang Chen, Yujuan Tan, and Chengliang Wang. 2021. FedSAE: A Novel Self-Adaptive Federated Learning Framework in Heterogeneous Systems. *arXiv preprint arXiv:2104.07515* (2021).

[90] Li Li, Daoyuan Li, Tegawendé F Bissyandé, Jacques Klein, Yves Le Traon, David Lo, and Lorenzo Cavallaro. 2017. Understanding android app piggybacking: A systematic study of malicious code grafting. *IEEE Transactions on Information Forensics and Security* 12, 6 (2017), 1269–1284. https://doi.org/10.1109/TIFS.2017.2656460

[91] Li Li, Duo Liu, Moming Duan, Yu Zhang, Ao Ren, Xianzhang Chen, Yujuan Tan, and Chengliang Wang. 2022. Federated learning with workload-aware client scheduling in heterogeneous systems. *Neural Networks* 154 (2022), 560–573. https://doi.org/10.1016/j.neunet.2022.07.030

[92] Li Erran Li, Eric Chen, Jeremy Hermann, Pusheng Zhang, and Luming Wang. 2017. Scaling machine learning as a service. In *International Conference on Predictive Applications and APIs*. PMLR, 14–29.

[93] Qinbin Li, Yanzheng Cai, Yuxuan Han, Ching Man Yung, Tianyuan Fu, and Bingsheng He. 2022. FedTree: A Fast, Effective, and Secure Tree-based Federated Learning System. https://github.com/Xtra-Computing/FedTree/blob/main/FedTree_draft_paper.pdf.

[94] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. 2021. Federated learning on non-iid data silos: An experimental study. *arXiv preprint arXiv:2102.02079* (2021).

[95] Qinbin Li, Bingsheng He, and Dawn Song. 2020. Practical one-shot federated learning for cross-silo setting. *arXiv preprint arXiv:2010.01017* (2020).

[96] Qinbin Li, Bingsheng He, and Dawn Song. 2021. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10713–10722.

[97] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. 2021. A survey on federated learning systems: vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* (2021). https://doi.org/10.1109/TKDE.2021.3124599

[98] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. 2020. Ditto: Fair and robust federated learning through personalization. *arXiv preprint arXiv:2012.04221* (2020).

[99] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine* 37, 3 (2020), 50–60.

[100] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. In *Proceedings of the 3rd SysML Conference*.

[101] Wenqi Li, Fausto Milletarì, Daguang Xu, Nicola Rieke, Jonny Hancox, Wentao Zhu, Maximilian Baust, Yan Cheng, Sébastien Ourselin, M Jorge Cardoso, et al. 2019. Privacy-preserving federated brain tumour segmentation. In *Machine Learning in Medical Imaging: 10th International Workshop, MLMI 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Proceedings 10*. Springer, 133–141. https://doi.org/10.1007/978-3-030-32692-0_16

[102] Xiaoxiao Li, Yufeng Gu, Nicha Dvornek, Lawrence H Staib, Pamela Ventola, and James S Duncan. 2020. Multi-site fMRI analysis using privacy-preserving federated learning and domain adaptation: ABIDE results. *Medical Image Analysis* 65 (2020), 101765. https://doi.org/10.1016/j.media.2020.101765

[103] Zhuohang Li, Jiaxin Zhang, Luyang Liu, and Jian Liu. 2022. Auditing privacy defenses in federated learning via generative gradient leakage. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10132–10142.

[104] Daniele Licari and Giovanni Comandè. 2022. ITALIAN-LEGAL-BERT: A Pre-trained Transformer Language Model for Italian Law. In *Companion Proceedings of the 23rd International Conference on Knowledge Engineering and Knowledge Management (CEUR Workshop Proceedings, Vol. 3256)*. CEUR, Bozen-Bolzano, Italy.

[105] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. 2020. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials* 22, 3

(2020), 2031–2063. https://doi.org/10.1109/COMST.2020.2986024
[106] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. 2020. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems* 33 (2020), 2351–2363.
[107] Hao Liu, Qian Gao, Jiang Li, Xiaochao Liao, Hao Xiong, Guangxing Chen, Wenlin Wang, Guobao Yang, Zhiwei Zha, Daxiang Dong, et al. 2021. Jizhi: A fast and cost-effective model-as-a-service system for web-scale online inference at baidu. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. Association for Computing Machinery, New York, NY, USA, 3289–3298. https://doi.org/10.1145/3447548.3467146
[108] Yang Liu, Tao Fan, Tianjian Chen, Qian Xu, and Qiang Yang. 2021. FATE: An industrial grade platform for collaborative learning with data protection. *The Journal of Machine Learning Research* 22, 1 (2021), 10320–10325.
[109] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).
[110] Ziyao Liu, Jiale Guo, Wenzhuo Yang, Jiani Fan, Kwok-Yan Lam, and Jun Zhao. 2022. Privacy-preserving aggregation in federated learning: A survey. *IEEE Transactions on Big Data (TBD)* (2022), 1–20. https://doi.org/10.1109/TBDATA.2022.3190835
[111] Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. 2017. Data-free knowledge distillation for deep neural networks. *NIPS 2017 Workshop on Learning with Limited Labeled Data: Weak Supervision and Beyond* (2017).
[112] Yihang Lou, Ling-Yu Duan, Yong Luo, Ziqian Chen, Tongliang Liu, Shiqi Wang, and Wen Gao. 2020. Towards efficient front-end visual sensing for digital retina: A model-centric paradigm. *IEEE Transactions on Multimedia* 22, 11 (2020), 3002–3013. https://doi.org/10.1109/TMM.2020.2966885
[113] Heiko Ludwig, Nathalie Baracaldo, Gegi Thomas, Yi Zhou, Ali Anwar, Shashank Rajamoni, Yuya Ong, Jayaram Radhakrishnan, Ashish Verma, Mathieu Sinn, et al. 2020. IBM Federated Learning: an Enterprise Framework White Paper V0. 1. *arXiv preprint arXiv:2007.10987* (2020).
[114] Lingjuan Lyu, Han Yu, and Qiang Yang. 2020. Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133* (2020).
[115] Xiaodong Ma, Jia Zhu, Zhihao Lin, Shanxuan Chen, and Yangjie Qin. 2022. A state-of-the-art survey on solving non-IID data in Federated Learning. *Future Generation Computer Systems* 135 (2022), 244–258. https://doi.org/10.1016/j.future.2022.05.003
[116] Yanjun Ma, Dianhai Yu, Tian Wu, and Haifeng Wang. 2019. PaddlePaddle: An open-source deep learning platform from industrial practice. *Frontiers of Data and Domputing* 1, 1 (2019), 105–115. https://doi.org/10.11871/jfdc.issn.2096.742X.2019.01.011
[117] Richard Maclin, Jude W Shavlik, et al. 1995. Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks. In *IJCAI*, Vol. 95. 524–531.
[118] Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos Jr, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. 2023. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology* (2023), 1–8. https://doi.org/10.1038/s41587-022-01618-2
[119] Frank McKeen, Ilya Alexandrovich, Ittai Anati, Dror Caspi, Simon Johnson, Rebekah Leslie-Hurd, and Carlos Rozas. 2016. Intel® software guard extensions (Intel® SGX) support for dynamic memory management inside an enclave. In *Proceedings of the Hardware and Architectural Support for Security and Privacy 2016*. 1–9. https://doi.org/10.1145/2948618.2954331
[120] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*. 1273–1282.
[121] Matias Mendieta, Taojiannan Yang, Pu Wang, Minwoo Lee, Zhengming Ding, and Chen Chen. 2022. Local Learning Matters: Rethinking Data Heterogeneity in Federated Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8397–8406.
[122] Yisroel Mirsky and Wenke Lee. 2021. The creation and detection of deepfakes: A survey. *ACM Computing Surveys (CSUR)* 54, 1 (2021), 1–41. https://doi.org/10.1145/3425780
[123] Dinh C Nguyen, Ming Ding, Quoc-Viet Pham, Pubudu N Pathirana, Long Bao Le, Aruna Seneviratne, Jun Li, Dusit Niyato, and H Vincent Poor. 2021. Federated learning meets blockchain in edge computing: Opportunities and challenges. *IEEE Internet of Things Journal* 8, 16 (2021), 12806–12825. https://doi.org/10.1109/JIOT.2021.3072611
[124] Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. 2022. A conversational paradigm for program synthesis. *arXiv e-prints* (2022), arXiv–2203.
[125] Chaoyue Niu, Fan Wu, Shaojie Tang, Lifeng Hua, Rongfei Jia, Chengfei Lv, Zhihua Wu, and Guihai Chen. 2020. Billion-scale federated learning on mobile clients: A submodel design with tunable privacy. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 1–14.
[126] Curtis G Northcutt, Anish Athalye, and Jonas Mueller. 2021. Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks. In *Proceedings of the 35th International Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.
[127] National Commission on New Technological Uses of Copyrighted Works (US). 1979. Final Report of the National Commission on New Technological Uses of Copyrighted Works, July 31, 1978. Library of Congress.

[128] David Opitz and Jude Shavlik. 1995. Generating accurate and diverse members of a neural-network ensemble. *Advances in Neural Information Processing Systems (NIPS)* 8 (1995).

[129] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems (NeurIPS)* 35 (2022), 27730–27744.

[130] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering (TKDE)* 22, 10 (2009), 1345–1359.

[131] Jungwuk Park, Dong-Jun Han, Minseok Choi, and Jaekyun Moon. 2021. Sageflow: Robust federated learning against both stragglers and adversaries. *Advances in neural information processing systems (NeurIPS)* 34 (2021), 840–851.

[132] Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*. 58–65. https://doi.org/10.18653/v1/W19-5006

[133] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543. https://doi.org/10.3115/v1/D14-1162

[134] Bruce Perens. 1999. The open source definition. *Open sources: voices from the open source revolution* 1 (1999), 171–188.

[135] Michael P Perrone and Leon N Cooper. 1995. When networks disagree: Ensemble methods for hybrid neural networks. In *How We Learn; How We Remember: Toward An Understanding Of Brain And Neural Systems: Selected Papers of Leon N Cooper*. World Scientific, 342–358.

[136] Bjarne Pfitzner, Nico Steckhan, and Bert Arnrich. 2021. Federated learning in a medical context: A systematic literature review. *ACM Transactions on Internet Technology (TOIT)* 21, 2 (2021), 1–31. https://doi.org/10.1145/3412357

[137] Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. 2022. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing* 70 (2022), 1142–1154. https://doi.org/10.1109/TSP.2022.3153135

[138] Youyang Qu, Md Palash Uddin, Chenquan Gan, Yong Xiang, Longxiang Gao, and John Yearwood. 2022. Blockchain-enabled federated learning: A survey. *ACM Computing Surveys (CSUR)* 55, 4 (2022), 1–35. https://doi.org/10.1145/3524104

[139] Zhe Qu, Xingyu Li, Rui Duan, Yao Liu, Bo Tang, and Zhuo Lu. 2022. Generalized federated learning via sharpness aware minimization. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*. PMLR, 18250–18280.

[140] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[141] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2383–2392. https://doi.org/10.18653/v1/D16-1264

[142] Swaroop Ramaswamy, Rajiv Mathews, Kanishka Rao, and Françoise Beaufays. 2019. Federated learning for emoji prediction in a mobile keyboard. *arXiv preprint arXiv:1906.04329* (2019).

[143] Swarna Priya Ramu, Parimala Boopalan, Quoc-Viet Pham, Praveen Kumar Reddy Maddikunta, Thien Huynh-The, Mamoun Alazab, Thanh Thi Nguyen, and Thippa Reddy Gadekallu. 2022. Federated learning enabled digital twins for smart cities: Concepts, recent advances, and future directions. *Sustainable Cities and Society* 79 (2022), 103663. https://doi.org/10.1016/j.scs.2021.103663

[144] Hersh R. Reddy. 2009. Jacobsen v. Katzer: The Federal Circuit Weighs in on the Enforceability of Free and Open Source Software Licenses II. Copyright - Note. *Berkeley Technology Law Journal* 24, 1 (2009), 299–320.

[145] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. 2020. FedQAP: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2021–2031.

[146] Amirhossein Reisizadeh, Hossein Taheri, Aryan Mokhtari, Hamed Hassani, and Ramtin Pedarsani. 2019. Robust and communication-efficient collaborative learning. *Advances in Neural Information Processing Systems (NeurIPS)* 32 (2019).

[147] Mauro Ribeiro, Katarina Grolinger, and Miriam AM Capretz. 2015. MLaaS: Machine learning as a service. In *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*. IEEE, 896–902. https://doi.org/10.1109/ICMLA.2015.152

[148] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. 2020. The future of digital health with federated learning. *NPJ digital medicine* 3, 1 (2020), 119. https://doi.org/10.1038/s41746-020-00323-1

[149] Dumitru Roman, Sven Schade, Arne-Jørgen Berre, Nils Rune Bodsberg, and J Langlois. 2009. Model as a Service (MaaS). In *AGILE Workshop-Grid Technologies for Geospatial Applications*.

[150] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10684–10695.

[151] Holger R Roth, Yan Cheng, Yuhong Wen, Isaac Yang, Ziyue Xu, Yuan-Ting Hsieh, Kristopher Kersten, Ahmed Harouni, Can Zhao, Kevin Lu, et al. 2022. NVIDIA FLARE: Federated Learning from Simulation to Real-World. (2022).

[152] Felix Sattler, Tim Korjakow, Roman Rischke, and Wojciech Samek. 2021. FedAUX: Leveraging unlabeled auxiliary data in federated learning. *IEEE Transactions on Neural Networks and Learning Systems* (2021). https://doi.org/10.1109/TNNLS.2021.3129371

[153] Felix Sattler, Klaus-Robert Müller, Thomas Wiegand, and Wojciech Samek. 2020. On the Byzantine Robustness of Clustered Federated Learning. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 8861–8865.

[154] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. 2019. Robust and communication-efficient federated learning from non-iid data. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)* 31, 9 (2019), 3400–3413. https://doi.org/10.1109/TNNLS.2019.2944481

[155] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. BLOOM: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100* (2022).

[156] Jonas Scherer, Marco Nolden, Jens Kleesiek, Jasmin Metzger, Klaus Kades, Verena Schneider, Michael Bach, Oliver Sedlaczek, Andreas M Bucher, Thomas J Vogl, et al. 2020. Joint imaging platform for federated clinical data analytics. *JCO clinical cancer informatics* 4 (2020), 1027–1038. https://doi.org/10.1200/CCI.20.00045

[157] Yuzhang Shang, Bin Duan, Ziliang Zong, Liqiang Nie, and Yan Yan. 2021. Lipschitz Continuity Guided Knowledge Distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 10675–10684.

[158] Shuo Shao, Wenyuan Yang, Hanlin Gu, Jian Lou, Zhan Qin, Lixin Fan, Qiang Yang, and Kui Ren. 2022. FedTracker: Furnishing Ownership Verification and Traceability for Federated Learning Model. *arXiv preprint arXiv:2211.07160* (2022).

[159] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[160] IEEE Computer Society. 2021. IEEE Guide for Architectural Framework and Application of Federated Machine Learning. *IEEE Std 3652.1-2020* (2021), 1–69. https://doi.org/10.1109/IEEESTD.2021.9382202

[161] Lichao Sun and Lingjuan Lyu. 2021. Federated Model Distillation with Noise-Free Differential Privacy. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*. International Joint Conferences on Artificial Intelligence Organization, 1563–1570. https://doi.org/10.24963/ijcai.2021/216

[162] Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. 2022. Black-box tuning for language-model-as-a-service. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*. PMLR, 20841–20855.

[163] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. 2019. Can you really backdoor federated learning? *NeurIPS 2019 Workshop on Federated Learning for User Privacy and Data Confidentiality* (2019).

[164] Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S Emer. 2017. Efficient processing of deep neural networks: A tutorial and survey. *Proc. IEEE* 105, 12 (2017), 2295–2329. https://doi.org/10.1109/JPROC.2017.2761740

[165] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. 2022. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)* (2022), 1–17. https://doi.org/10.1109/TNNLS.2022.3160699

[166] Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. GALACTICA: A large language model for science. *arXiv preprint arXiv:2211.09085* (2022).

[167] Buse GA Tekgul, Yuxi Xia, Samuel Marchal, and N Asokan. 2021. WAFFLE: Watermarking in federated learning. In *2021 40th International Symposium on Reliable Distributed Systems (SRDS)*. IEEE, 310–320. https://doi.org/10.1109/SRDS53918.2021.00038

[168] Omer Tene. 2011. Privacy: The new generations. *International data privacy law* 1, 1 (2011), 15–27. https://doi.org/10.1093/idpl/ipq003

[169] Nguyen Truong, Kai Sun, Siyao Wang, Florian Guitton, and YiKe Guo. 2021. Privacy preservation in federated learning: An insightful survey from the GDPR perspective. *Computers & Security* 110 (2021), 102402. https://doi.org/10.1016/j.cose.2021.10240

[170] Manasi Vartak, Harihar Subramanyam, Wei-En Lee, Srinidhi Viswanathan, Saadiyah Husnoo, Samuel Madden, and Matei Zaharia. 2016. ModelDB: a system for machine learning model management. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*. 1–3. https://doi.org/10.1145/2939502.2939516

[171] Paul Voigt and Axel Von dem Bussche. 2017. The EU general data protection regulation (GDPR): A Practical Guide. *Springer International Publishing* (2017). https://doi.org/10.1007/978-3-319-57959-7

[172] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. 2020. Attack of the tails: Yes, you really can backdoor federated learning. *Advances in Neural Information Processing Systems (NeurIPS)* 33 (2020), 16070–16084.

[173] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. 2020. Federated Learning with Matched Averaging. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*.

[174] Haoyu Wang, Handong Zhao, Yaqing Wang, Tong Yu, Jiuxiang Gu, and Jing Gao. 2022. FedKC: Federated Knowledge Composition for Multilingual Natural Language Understanding. In *Proceedings of the ACM Web Conference 2022*. 1839–1850. https://doi.org/10.1145/3485447.3511988

[175] Lin Wang and Kuk-Jin Yoon. 2021. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 6 (2021), 3048–3068. https://doi.org/10.1109/TPAMI.2021.3055564

[176] Mei Wang and Weihong Deng. 2018. Deep visual domain adaptation: A survey. *Neurocomputing* 312 (2018), 135–153. https://doi.org/10.1016/j.neucom.2018.05.083

[177] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. 2019. Beyond Inferring Class Representatives: User-Level Privacy Leakage From Federated Learning. In *Proceedings of the 2019 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2512–2520.

[178] Wenqi Wei, Ling Liu, Yanzhao Wut, Gong Su, and Arun Iyengar. 2021. Gradient-leakage resilient federated learning. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 797–807. https://doi.org/10.1109/ICDCS51616.2021.00081

[179] Steven Euijong Whang, Yuji Roh, Hwanjun Song, and Jae-Gil Lee. 2023. Data collection and quality challenges in deep learning: A data-centric ai perspective. *The VLDB Journal* (2023), 1–23. https://doi.org/10.1007/s00778-022-00775-9

[180] David H Wolpert. 1992. Stacked generalization. *Neural networks* 5, 2 (1992), 241–259.

[181] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. 2022. Communication-efficient federated learning via knowledge distillation. *Nature communications* 13, 1 (2022), 2032. https://doi.org/10.1038/s41467-022-29763-x

[182] Zhaomin Wu, Qinbin Li, and Bingsheng He. 2022. Practical vertical federated learning with unsupervised representation learning. *IEEE Transactions on Big Data* (2022). https://doi.org/10.1109/TBDATA.2022.3180117

[183] Chulin Xie, De-An Huang, Wenda Chu, Daguang Xu, Chaowei Xiao, Bo Li, and Anima Anandkumar. 2023. PerAda: Parameter-Efficient and Generalizable Federated Learning Personalization with Guarantees. *arXiv preprint arXiv:2302.06637* (2023).

[184] Chenhao Xu, Youyang Qu, Yong Xiang, and Longxiang Gao. 2021. Asynchronous federated learning on heterogeneous devices: A survey. *arXiv preprint arXiv:2109.04269* (2021).

[185] Jie Xu, Benjamin S Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. 2021. Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research* 5 (2021), 1–19. https://doi.org/10.1007/s41666-020-00082-4

[186] Qiang Yang, Lixin Fan, Richard Tong, and Angelica Lv. 2021. IEEE Federated Machine Learning. *IEEE Federated Machine Learning - White Paper* (2021), 1–18.

[187] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19. https://doi.org/10.1145/3298981

[188] Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. 2018. Applied federated learning: Improving google keyboard query suggestions. *arXiv preprint arXiv:1812.02903* (2018).

[189] Xin Yao, Tianchi Huang, Chenglei Wu, Ruixiao Zhang, and Lifeng Sun. 2019. Towards faster and better federated learning: A feature fusion approach. In *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 175–179. https://doi.org/10.1109/ICIP.2019.8803001

[190] Xuefei Yin, Yanming Zhu, and Jiankun Hu. 2021. A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. *ACM Computing Surveys (CSUR)* 54, 6 (2021), 1–36. https://doi.org/10.1145/3460427

[191] Shan You, Chang Xu, Fei Wang, and Changshui Zhang. 2021. Workshop on Model Mining. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 4177–4178. https://doi.org/10.1145/3447548.3469471

[192] Fuxun Yu, Weishan Zhang, Zhuwei Qin, Zirui Xu, Di Wang, Chenchen Liu, Zhi Tian, and Xiang Chen. 2021. Fed2: Feature-aligned federated learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2066–2074. https://doi.org/10.1145/3447548.3467309

[193] Han Yu, Zelei Liu, Yang Liu, Tianjian Chen, Mingshu Cong, Xi Weng, Dusit Niyato, and Qiang Yang. 2020. A fairness-aware incentive scheme for federated learning. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 393–399. https://doi.org/10.1145/3375627.3375840

[194] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, and Nghia Hoang. 2019. Statistical model aggregation via parameter matching. *Advances in Neural Information Processing Systems 32 (NeurIPS)* 32 (2019).

[195] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. 2019. Bayesian nonparametric federated learning of neural networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*. PMLR, 7252–7261.

[196] Rafael Yuste, Sara Goering, Blaise Agüera y Arcas, Guoqiang Bi, Jose M Carmena, Adrian Carter, Joseph J Fins, Phoebe Friesen, Jack Gallant, Jane E Huggins, et al. 2017. Four ethical priorities for neurotechnologies and AI. *Nature* 551, 7679 (2017), 159–163. https://doi.org/10.1038/551159a

[197] Dun Zeng, Siqi Liang, Xiangjing Hu, Hui Wang, and Zenglin Xu. 2021. Fedlab: A flexible federated learning framework. *arXiv preprint arXiv:2107.11621* (2021).

[198] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. 2021. A survey on federated learning. *Knowledge-Based Systems (KBS)* 216 (2021), 106775. https://doi.org/10.1016/j.knosys.2021.106775

[199] Jie Zhang, Chen Chen, Bo Li, Lingjuan Lyu, Shuang Wu, Shouhong Ding, Chunhua Shen, and Chao Wu. 2022. DENSE: Data-Free One-Shot Federated Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[200] Ke Zhang, Carl Yang, Xiaoxiao Li, Lichao Sun, and Siu Ming Yiu. 2021. Subgraph federated learning with missing neighbor generation. *Advances in Neural Information Processing Systems (NeurIPS)* 34 (2021), 6671–6682.

[201] Lvmin Zhang and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543* (2023).

[202] Qi Zhang, Tiancheng Wu, Peichen Zhou, Shan Zhou, Yuan Yang, and Xiulang Jin. 2022. Felicitas: Federated Learning in Distributed Cross Device Collaborative Frameworks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4502–4509. https://doi.org/10.1145/3534678.3539039

[203] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068* (2022).

[204] Tuo Zhang, Lei Gao, Chaoyang He, Mi Zhang, Bhaskar Krishnamachari, and A Salman Avestimehr. 2022. Federated learning for the internet of things: applications, challenges, and opportunities. *IEEE Internet of Things Magazine* 5, 1 (2022), 24–29. https://doi.org/10.1109/IOTM.004.2100182

[205] Lingchen Zhao, Qian Wang, Cong Wang, Qi Li, Chao Shen, and Bo Feng. 2021. Veriml: Enabling integrity assurances and fair payments for machine learning as a service. *IEEE Transactions on Parallel and Distributed Systems* 32, 10 (2021), 2524–2540. https://doi.org/10.1109/TPDS.2021.3068195

[206] Zhi-Hua Zhou. 2012. *Ensemble methods: foundations and algorithms*. CRC press.

[207] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. 2021. Federated learning on non-IID data: A survey. *Neurocomputing* 465 (2021), 371–390. https://doi.org/10.1016/j.neucom.2021.07.098

[208] Juncen Zhu, Jiannong Cao, Divya Saxena, Shan Jiang, and Houda Ferradi. 2022. Blockchain-empowered federated learning: Challenges, solutions, and future directions. *ACM Computing Surveys (CSUR)* (2022). https://doi.org/10.1145/3570953

[209] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. *Advances in Neural Information Processing Systems (NeurIPS)* 32 (2019).

[210] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. 2021. Data-free knowledge distillation for heterogeneous federated learning. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*. PMLR, 12878–12889.

[211] Alexander Ziller, Andrew Trask, Antonio Lopardo, Benjamin Szymkow, Bobby Wagner, Emma Bluemke, Jean-Mickael Nounahon, Jonathan Passerat-Palmbach, Kritika Prakash, Nick Rose, et al. 2021. PySyft: A library for easy federated learning. *Federated Learning Systems: Towards Next-Generation AI* (2021), 111–139. https://doi.org/10.1007/978-3-030-70604-3_5

[212] Guobing Zou, Bofeng Zhang, Jianxing Zheng, Yinsheng Li, and Jianhua Ma. 2012. MaaS: Model as a service in cloud computing and Cyber-I space. In *2012 IEEE 12th International Conference on Computer and Information Technology*. IEEE, 1125–1130. https://doi.org/10.1109/CIT.2012.228

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009