Towards Open Federated Learning Platforms: Survey and Vision from Technical and Legal Perspectives

MOMING DUAN, National University of Singapore, Singapore

Traditional Federated Learning (FL) follows a server-domincated cooperation paradigm which narrows the application scenarios of FL and decreases the enthusiasm of data holders to participate. To fully unleash the potential of FL, we advocate rethinking the design of current FL frameworks and extending it to a more generalized concept: Open Federated Learning Platforms. We propose two reciprocal cooperation frameworks for FL to achieve this: query-based FL and contract-based FL. In this survey, we conduct a comprehensive review of the feasibility of constructing an open FL platform from both technical and legal perspectives. We begin by reviewing the definition of FL and summarizing its inherent limitations, including server-client coupling, low model reusability, and non-public. In the query-based FL platform, which is an open model sharing and reusing platform empowered by the community for model mining, we explore a wide range of valuable topics, including the availability of up-to-date model repositories for model querying, legal compliance analysis between different model licenses, and copyright issues and intellectual property protection in model reusing. In particular, we introduce a novel taxonomy to streamline the analysis of model license compatibility in FL studies that involve batch model reusing methods, including combination, amalgamation, distillation, and generation. This taxonomy provides a systematic framework for identifying the corresponding clauses of licenses and facilitates the identification of potential legal implications and restrictions when reusing models. Through this survey, we uncover the the current dilemmas faced by FL and advocate for the development of sustainable open FL platforms. We aim to provide guidance for establishing such platforms in the future, while identifying potential problems and challenges that need to be addressed.

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 [234] and software [110]. This has led to the world entering the third wave of AI development: Deep Learning [130]. The success of current data-driven AI relies on massive amounts of training data and follows a gather-and-analyze paradigm [259], which confronts with challenges of complying with rigorous data protection regulations such as OECD Privacy Guidelines [240] and General and Data Protection Regulation (GDPR) [248]. So although data-centric AI is currently mainstream paradigm, Federated Learning [146], a novel model-centric distributed collaborative training framework, is gaining popularity in both academia and industry for its advantages in complying with privacy regulations [243].

According to the definitions of IEEE Standard for Federated Machine Learning (FML, aka FL) [225], *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 [19]. Based on this workflow pattern, many FL frameworks have been derived with specialized improvements in communication [123, 176, 268], optimizaiton [119, 142, 147], robustness [53, 137, 218] and privacy [20, 39, 73]. 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: Moming Duan, National University of Singapore, Singapore, Singapore, moming@nus.edu.sg.

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.

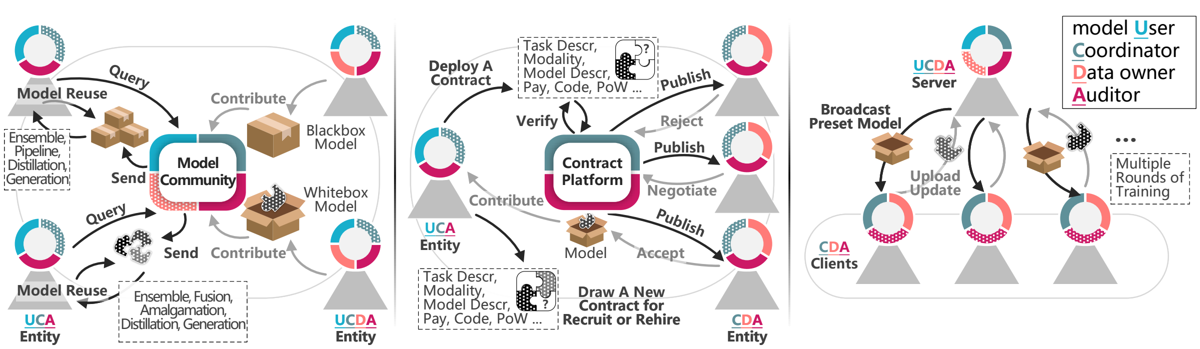
In this paper, we try to answer the question: **Can we establish a sustainable open FL platform based on a novel reciprocal cooperation framework?** Obviously, to answer this quesion, it is insufficient simply study the basic concepts of FL and investigate existing FL techniques. We also need to conduct 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 an open repository named Model Community. There are many valuable challenges that can be explored, such as how to query for models, how to reuse the retrieved models or how to transfer knowledge from these models, how to ensure the legal compliance between different model licenses, how to protect the intelligent properties of released models (ref. Section 4).
- **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 setting include model pricing, model contribution evaluation and so on.

It's worth noting that the definitions of the four roles illustrated in Fig. 1 (i.e. model user, coordinator, data owner, auditor, ref. Section 3.1) are adopted for compatibility with the IEEE standard [225], and our proposals are also within the standard definitions of FML. 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 and model averaging until the global model converges. In contrast, the entities in query-based FL and contract-based FL are proactive in their participate. We believe that these reciprocal cooperation frameworks have the potential to expand the prevalence of open FL and establish FL ecosystems.

1.1 Our Contribution

In contrast to previous surveys that primarily focused on the server-dominated cooperation framework in FL, our new survey explores the feasibility of reciprocal cooperation frameworks in FL. To the best of our knowledge, our work represents the first systematic survey in this area. The major contributions of this survey are as follows:



(a) Query-based FML (b) Contract-based FML (c) Traditional FML

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 [225], and colors with grid lines indicate non-essential roles.

- We introduce the concept of open FL platforms by presenting two reciprocal cooperation frameworks, namely query-based FL and contract-based FL, along with an overview of their key features and properties.
- We explore the query functionalities of online model repositories, such as Huggingface and OpenVINO, to investigate their feasibility for model query in query-based FL settings.
- We summarize the rights, restrictions, and enforcements of in-service model licenses and highlight the legal compliance and copyrightability issues in collaborative modeling. Additionally, we provide guidelines for selecting licenses to minimize conflicts and prevent license proliferation.
- We propose a new taxonomy to streamline the legal compliance analysis in FL studies, which is also useful for quickly identifying suitable model reusing methods for open FL platforms. A comprehensive comparison of current FL studies based on this taxonomy is surveyed.
- We analyze the requirements for model protection in the context of query-based FL and identify applicable solutions from deep intellectual property protection technologies.

The rest of this paper is organized as follows. We compare this survey to other related surveys and show our distinction in Section 2. In Section 3, we presente the overview and point the limitations of traditional FL. We comprehensively explore the feasibility and challenges of query-based FL in Section 4, which includes model query (Section 4.2), model license comparison (Section 4.3.1) and selection (Section 4.3.2), copyright issues (Section 4.3.3), and analysis of license conflicts in model reusing (Section 4.3.6). In Section 4.3.4, we present our taxonomy from a model reusing perspective, and in Section 4.3.5, we summarize FL studies based on this new taxonomy. The discussion on model intellectual property protection is presented in Section 4.4.

2 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 applications 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.* [272]. They categorized FL into horizontal FL, vertical FL and federated transfer learning based on the distribution characteristics of data, which are written in IEEE Standard 3652.1-2020 [225, 271]. Following this, an increasing number of surveys have emerged focusing on enhancing FL system design [8, 117, 143, 146, 287]. From the algorithmic perspective, personlized FL [127, 235] aims to learn personlized models for each client to address the challenge of statistical heterogeneity [168]. Besides, the privacy-perserving computing platforms and model aggregation protocols for FL systems also been widely studied and sumaried by [57, 162, 167, 275]. Furthermore, many advanced FL architectures had been proposed, such as asynchronous [268], decentralized and blockchain-based FL frameworks [180, 198, 303]. 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 [9, 194, 208, 269]. The massive real-time data generated by IoT devices in smart cities [203, 299], industries [21], vehicles [52] 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, 7, 74].

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 listed surveys just present the same story from slightly different angles and 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,

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.* [272]	✓	✓	✓	✓	✓	✓	✓	✓
	Li *et al.* 2020 [146]	✓	✓	✓		✓	✓	✓	✓
	Zhang 2021*et al.* [287]	✓	✓	✓			✓	✓	✓
	Gupta *et al.* [85]	✓	✓	✓		✓	✓	✓	✓
	Xu *et al.* [268]	✓	✓	✓		✓	✓	✓	✓
	Li *et al.* 2021 [143]	✓	✓	✓	✓	✓	✓	✓	✓
	El *et al.* [57]			✓		✓	✓		✓
	Kulkarni *et al.* [127]	✓	✓				✓		✓
	Liu *et al.*[162]	✓		✓		✓	✓		✓
	Tan *et al.* [235]		✓				✓		✓
	Zhu *et al.* 2021 [302]		✓				✓		✓
	Ma *et al.* [168]	✓	✓	✓			✓		✓
	Aledhari *et al.* [8]	✓	✓				✓	✓	✓
	Kairouz *et al.* [117]	✓	✓	✓	✓	✓	✓	✓	✓
	AbdulRahman *et al.* [3]	✓	✓	✓	✓		✓	✓	✓
	Lim *et al.* [156]	✓	✓	✓	✓		✓	✓	✓
Healthcare	Xu *et al.* [269]	✓	✓	✓			✓	✓	✓
	Pfitzner *et al.*[194]	✓	✓	✓			✓	✓	✓
	Antunes *et al.* [9]		✓	✓				✓	✓
	Rieke *et al.* [208]		✓	✓		✓	✓	✓	✓
IoT	Zhang 2022*et al.* [299]	✓	✓				✓	✓	✓
	Boopalan *et al.* [21]	✓	✓	✓	✓	✓	✓	✓	✓
	Ramu *et al.* [203]	✓	✓	✓		✓	✓	✓	✓
	Du *et al.* [52]	✓	✓	✓	✓	✓	✓	✓	✓
Cybersecurity	Agrawal *et al.* [5]	✓	✓	✓		✓	✓	✓	✓
	Alazab *et al.* [7]			✓			✓	✓	✓
	Ghimire *et al.* [74]	✓		✓			✓	✓	✓
Blockchain	Nguyen *et al.* [180]	✓	✓	✓	✓	✓	✓	✓	✓
	Qu *et al.* [198]	✓	✓	✓	✓	✓	✓	✓	✓
	Zhu *et al.* 2022 [303]	✓	✓	✓	✓	✓	✓	✓	✓

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.

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 the following sections, we will distinguish our survey by highlingting the similarities and differences between these related concepts.

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 [89] and query suggestion [273] 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 [285], Felicitas [297], IBM FL [165], OpenFL [64]), Vertical FL [266] or both (e.g., FATE [160], FedML [91], PaddleFL [169], Flower [16], FedTree [139], NVFLARE [213]). 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 federated training services to the public.

Unlike the FL systems mentioned above, PySyft [306] 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 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 [68]. To preserve the privacy advantages of FL, in this survey, we investigate open and data-free FL platforms under the scope of model-centric ML [164]. In such FL platforms, every user is free to collaborate on the training of machine learning models while privacy is protected.

2.2 As-a-Service Business Model

In the current context of Software-as-a-Service (SaaS) [24], 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) [70, 158, 209, 231, 307] and Machine-Learning-as-a-Service (MLaaS) [87, 96, 125, 138, 207] encapsulate model execution and model development as services. The original concept of MaaS [70, 209] 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 systems. Subsequently, this concept has been extended to the field of recommendation systems [307] and deep learning based systems [158, 231]. 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 [86]. 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 [96, 207, 300] or inference [87], 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 Trusted Execution Environment (TEE) [87, 175] and Homomorphic Encryption [71, 96], it is worth noting that our focus is not solely on privacy.

Recently, Kourtellis *et al.* [125] 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.* [112] propose an open FL ecosystem for mobile devices, which shares a similar concept to

[1] https://blog.openmined.org/announcing-proof-of-concept-support-for-tff-in-pysyft-0-7/ [2] https://cloud.google.com/vertex-ai
[3] https://azure.microsoft.com/products/machine-learning/ [4] https://chat.openai.com/chat

FLaaS. However, those approach also follow the traditional server-dominated cooperation framework, which falls under the scope of previous FL surveys[117, 146, 272].

2.3 Dcentralized FL

Decentralized FL [98, 118, 128, 172, 230], a novel server-less paradigm of FL, emphasizes the advantages of employing a peer-to-peer model delivery and aggregation network that is free from the dependencies of a central trusted server. Instead of solely communicating with a central server, participants can fully leverage the network bandwidth by utilizing the network connections between them. For example, Lalitha *et al.* [128] proposed exchange and merge of posterior distribution among neighboring users to collaboratively estimate the global optimal parameter. Similar to the local training of FedAvg, DFedAvgM [230] also suggests that each client communicates with its neighbors after multiple training iterations to improve the convergence rate of training. On the other hand, ProxyFL [118] improves the privacy of neighbor-wide model sharing by sharing a proxy model through knowledge distillation [97]. However, the major bottleneck lies in the high communication cost of sharing the model with all neighbors in a fully connected network.

To address this issue, Marfoq *et al.* [172] proposed improving the efficiency of model sharing by selecting a connected subgraph. Another approach is the use of a gossip-based approach, where model parameters or segments of model parameters are randomly shared with peer neighbors [94, 98]. Despite the advantages brought by the decentralized design, the training procedure of these frameworks also follows a preset learning task and lacks sustainable cooperation, resulting in a non-public and low reusability FL platform similar to centralized FL. In fact, our vision for open FL platforms is to extend the FAIR principles [261] for scientific data to the context of machine learning. We believe that all dedicated models in these platforms should adhere to the principles of being **Findable, Accessible, Interoperable, and Reusable**.

3 BASIC CONCEPTS OF FEDERATED LEARNING

3.1 Definition

Federated Learning [176, 225] 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, FL systems consist of two main entities in terms of task organization: the server and the participants. 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 [117]:

- 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 setting typically falls within the scope of horizontal FL. The cross-device FL applications include: Gboard input suggestion [89, 202, 273], e-commerce recommendation [182] and advertising [249].
- 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 [160] and NVFLARE [213]. 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 [148, 149, 220], epidemiology [46] and EHR [25, 99].

Furthermore, FL entities can also serve multiple functional roles to support advanced features such as privacy enhancement [20, 73, 182], participant scheduling [2, 137], model verification [222, 239] and incentive mechanisms [279]. Recall that there are four roles defined in the FL standard [225]:

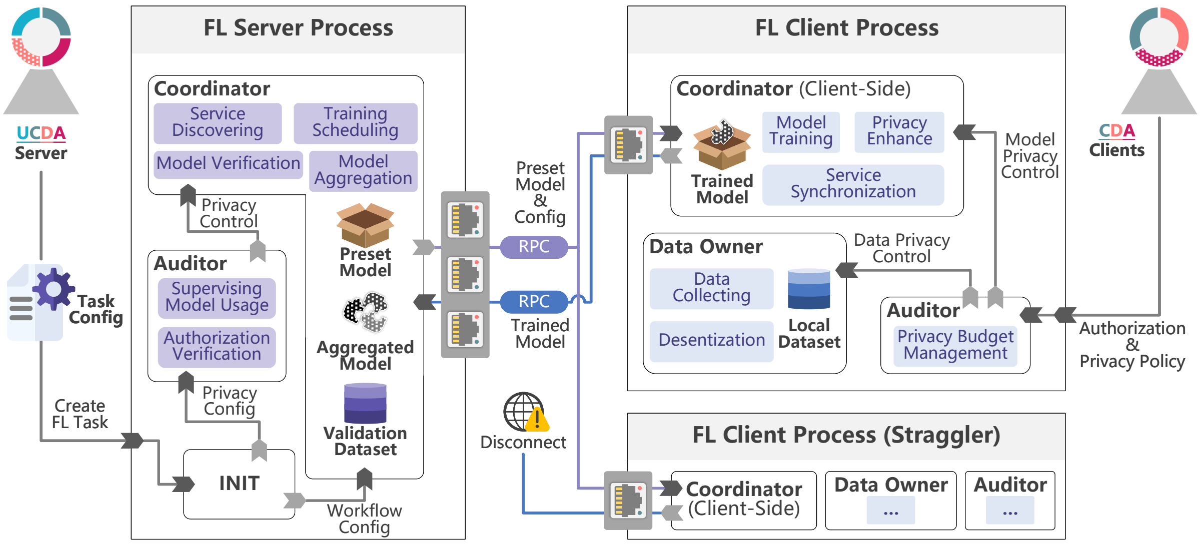


Fig. 2. An overview of traditional FL systems. (U: model User, C: Coordinator, D: Data owner, A: Auditor)

- **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 [87], scheduling the training and aggregation workflows for improve system efficiency, such as by alleviating the straggler effect [29, 135], optimizing data heterogeneity [2, 54] and compressing model transfer [123, 218]. Additionaly, the FL coordinator provides privacy control mechanisms [20, 57, 96] 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 [217].
- **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 [56] 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 [248]) 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 [115, 257, 304], making it crucial for auditors to scrutinize the model transmission [153, 258] and verify the ownership of models [222, 239].

Fig. 2 illustrates the typical architecture of FL systems, which as a distributed modeling toolkits consists of server part and client part. In a 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 on client devices. The server and clients are connected via Internet and typically with the help of Remote Procedure Call (RPC) interface for coordinating [1, 16, 64, 91, 160, 297]. 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) [19, 176] 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 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 [160, 213, 306].

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 [26, 145, 176]. 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.

3.2 Limitations of Traditional FL

Previous surveys [7, 117, 146, 180, 235, 272, 299, 303] 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 a same FL prototype [176], 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**.

3.2.1 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, energy constrains and data licenses, is a challenging task that significantly increases the complexity of the FL systems.

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 [26, 176, 285], or cause resource depletion on client-side [19, 38, 182], or even piggyback malicious executable codes [136]. 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 [23, 61, 188, 217] or insert backdoors [12, 250]. In addition, the unstable network environment can drive clients to drop out from traning (i.e. straggler effect), thereby reducing system efficiency [188, 206]. Therefore, the server-client coupling design of traditional FL systems make them susceptible to unpredictable runtime environments, leading to system vulnerability and low reliability.

3.2.2 Low Model Reusability. The traditional FL scheduling follows a task-centric manner and terminates once the training reaches a preset number of rounds or meets traget metrics on global model set by FL server [19]. 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. Meanwhile, 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.

3.2.3 Non-public. As we mention in Section 2.1, except PySyft [306], the application scenarios of mainstream FL frameworks [1, 16, 26, 91, 160, 165, 213, 285] 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 [19] and billions [182], 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 network even if it is public accessible. Therefore, the cornerstone of buliding a sustainable open FL platform is to create a reciprocal FL cooperation framework, followed by corresponding multi-source knowledge aggregation strategies, which we discuss in the following sections.

4 QUERY-BASED FEDERATED LEARNING

4.1 Overview

Let us continue by establishing 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 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 [49], BLOOM [219] 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, aka model mining [277]. The derivatives of model mining can learn representations from multiple domains, resulting in more promising performance that can be evaluated by platform users. Furthermore, the contributors can release 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 communication channels need not be maintained; (3) **Community-powered**, whereby sharing models enrichs the entire community.

Actually, we aim to advocate a novel SaaS [24] ML platform with automatic batch model reuse integrated, which has potential to leverage the transportability of models to address previously unexplored ML problems. Due

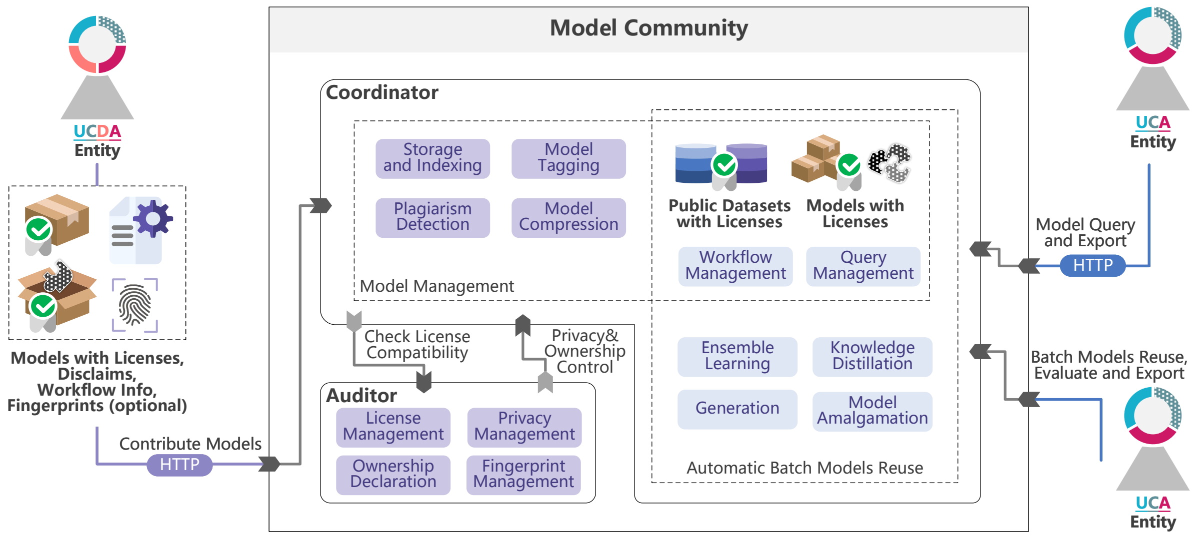


Fig. 3. An overview of query-based FL systems. (U: model User, C: Coordinator, D: Data owner, A: Auditor)

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 (ref. Section 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 [104] and knowledge distillation [97], we will explain the reasons in the following section.

4.2 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 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, and the associated tasks. To illustrate, one might search for the model name GPT [200], models trained on the MNIST dataset [131], 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.

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/output poses a challenge for query-based FL. Additionally, 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

Table 2. Filter conditions and characteristics of DNNs repositories. ✓: Supported, ✗ : Unsupported, ! : Information provided but unsearchable, listed in descending order by number of released models. (Accessed on January 17, 2024)

	DS Name	Model Architecture	Modality/Task	Tag	License	Input-Output	Batch Export	# of Models
Hugging Face[5]	✓	✓	✓	✓	✓	!	✗	470,263
Model Zoo[6]	✓	✓	✓	✓	✗	✗	✗	3,245
Tensorflow Hub[7]	✓	✓	✓	✓	!	!	✗	2,186
NVIDIA NGC[8]	!	✓	✓	✓	!	!	✗	680
OpenVINO[9]	!	✓	✓	✗	!	!	✓	277
Pytorch Hub[10]	!	✓	✗	✗	✗	!	✗	52

of DNNs repositories to enable quick identification of readily reusable models for knowledge aggregation. We further suggest following filter conditions for query-based FL.

4.2.1 Data Description. Similar with the data heterogeous challenges in FL [140]. 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 [46]. Besides, label errors pervasive even in open datasets [183]. 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 [149] and complementary provenance information.

4.2.2 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 [245], such as MLflow [11] and Neptune [12], to automatically track and store the ML workflow during model building process. This information can also assist in identifying potential model plagiarism within the model community. In addition, to ensure that the computational consumption of models is within budget, Deep Learning Profiler [13] can be leveraged to generate a report that displays the FLOPS and bandwidth requirements.

4.2.3 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 [45]. In some cases, contributed models may rely on other models as dependencies. For example, Fast R-CNN [75] uses VGG16 [224] 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 [176] aggregates the local models weights element-wise, which requires full access to the models. In contrast, MoE with a gating network [104] 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

[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

executable modules (e.g. dynamic 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.

4.3 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 [187]. 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 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 [187], Ensemble Leanring [301], Domain Adaptation [254], Knowledge Distillation [253], Deep Generative Models [27] and Model Fusion [107]. Meanwhile, 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 [179]) and protect the IP of creators, models are typically published under a license agreed upon by the licensor. Here, we summary the granted rights, restrictions and enforcements of licenses for ML models posted on Hugging Face in Table 3. The following sections will provide a detailed survey of these licenses.

4.3.1 Model Licensing Forms. As shown in Table 3, 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 ML code, model, and data. Thinking in terms of software, ML models can be released with reproducable code and be considered as a component of software. So many Free and Open Source Software (FOSS) licenses [212] are naturally deferred for licensing of models. The most popular license is Apache-2.0, which is a permissive FOSS license that allows the freedom to make derivative works. However, the model building process also relies on a massive amount of data [130] that may be licensed under different licenses, which can lead to license conflicts. A real-world example is BERT [49], 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.

Thinking in terms of content and database, some word embedding models, such as GloVe [191], 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 [161] (licensed under MIT) with SQuAD2 [201] (licensed under 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 [11, 116, 284], which is usually not restricted by traditional licenses designed for software and content. We can infer the concerns of unethical use of GPT-2 [200] from its modified MIT license granted by its inventors, which states, *We don't claim ownership of the*

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

Table 3. Licenses for ML models available on Hugging Face with a focus on their rights, restrictions and enforcements, grouped by FOSS 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, † : Public Domain License. Only the source code of the original work under AFL-3.0 or Artistic-2.0 is required to be disclosed. You may not distribute the modified materials licensed under CC-BY-NC-ND or CC-BY-ND. (Accessed on January 17, 2024)

Licenses	Modify / Merge	Redistribution	Sublicensing	Commercial Use	Patent Use	Trademark Use	State Changes	Disclose Source	Responsible-use Restrictions	License/Attribution Preservation	# of Models	Licensed Materials / Remarks
Apache-2.0	✓	✓	✓	✓	✓	✗	✓	✗	✗	✓	65,985	BERT [49]
MIT	✓	✓	✓	✓	!	!	✗	✗	✗	✓	30,344	GPT-2 [200]
AFL-3.0	✓	✓	✓	✓	✓	✗	✓	(✓)	✗	✓	2,208	Italian-Legal-BERT [155]
*GPL-3.0	✓	✓	✗	✓	✓	✗	✓	✓	✗	✓	1,242	PersonaGPT [237]
Artistic-2.0	✓	✓	✓	✓	✓	✗	✓	(✓)	✗	✓	675	Include original source
BSD-3-Clause&-Clear	✓	✓	✓	✓	!	✗	✗	✗	✗	✓	636	CodeGen [181]/ A MIT-style license
†WTFPL-2.0	✓	✓	!	✓	!	!	✗	✗	✗	✗	409	A MIT-style permissive license
*AGPL-3.0	✓	✓	✗	✓	✓	✗	✓	✓	✗	✓	265	Extended GPL covers SaaS
†Unlicense	✓	✓	!	✓	!	!	✗	✗	✗	✗	254	A MIT-style permissive license
*GPL-2.0	✓	✓	✗	✓	!	!	✓	✓	✗	✓	91	Not compatible with GPL-3.0
*LGPL-3.0&2.1	✓	✓	✗	✓	✓	!	✓	✓	✗	✓	84	For software libraries
BSD-2-Clause	✓	✓	✓	✓	!	!	✗	✗	✗	✓	82	A MIT-style permissive license
BSL-1.0	✓	✓	✓	✓	!	!	✗	✗	✗	✓	77	A MIT-style permissive license
*OSL-3.0	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓	55	Linking is not derivative work
*Ms-PL	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	43	Weak copyleft license
ECL-2.0	✓	✓	✓	✓	✓	✗	✓	✗	✗	✓	38	For education communities
Zlib	✓	✓	!	✓	!	!	✗	✗	✗	✓	30	Rename if modified
*MPL-2.0	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓	23	State changes under MPL only
*EPL-2.0&1.0	✓	✓	✓	✓	✓	!	✗	✓	✗	✓	19	Can link proprietary license code
ISC	✓	✓	!	✓	!	!	✗	✗	✗	✓	15	MIT-style license w/o sublicense
*EUPL-1.1	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓	15	License of EU covers SaaS
NCSA	✓	✓	✓	✓	!	✗	✗	✗	✗	✓	10	Include full text of license
PostgreSQL	✓	✓	!	✓	!	!	✗	✗	✗	✓	7	A MIT-style license
OpenRAIL	>Responsible AI License template, w/o full text										22,947	ControlNet [294]
CreativeML-OpenRAIL-M	✓	✓	✓	✓	✓	✗	✓	✗	✓	✓	15,591	Stable Diffusions v1 [210]
Llama2	✓	✓	✗	(✓)	✓	✗	✗	✗	✓	✓	3,538	Llama 2 [242]
OpenRAIL++	>Same as CreativeML-OpenRAIL-M										1,433	Stable Diffusion v2 [210]
BigScience-OpenRAIL-M	>Same as BigScience-BLOOM-RAIL-1.0										659	A general version of 1.0
BigScience-BLOOM-RAIL-1.0	✓	✓	✓	✓	✓	✗	✓	✗	✓	✓	527	BLOOM [219]
BigCode-OpenRAIL-M	>Same as BigScience-BLOOM-RAIL-1.0										320	StarCoder [144]
OPT-175B	✓	✗	✗	✗	✗	✗	✗	✗	✓	✓	≈ 94	OPT LLM [298]
SEER	>Same as OPT-175B, ban on reverse-engineer										≈ 23	SEER Vision Model [80]
CC-BY-NC-4.0&3.0&2.0	✓	✓	✗	✗	✗	✗	✓	✗	✗	✓	4,747	GALACTICA [238]
CC-BY-4.0&3.0&2.5&2.0	✓	✓	✗	✓	✗	✗	✓	✗	✗	✓	3,429	RoBERTa-SQuAD2.0 [201]
*CC-BY-NC-SA-4.0&3.0&2.0	✓	✓	✗	✗	✗	✗	✓	✓	✗	✓	1,783	LayoutLMv3 [101]
*CC-BY-SA-4.0&3.0	✓	✓	✗	✓	✗	✗	✓	✓	✗	✓	1,510	LEGAL-BERT [30]
CC-BY-NC-ND-4.0&3.0	(✓)	✗	✗	✗	✗	✗	✗	✗	✗	✓	406	NonCommercial, NoDerivatives
†CC0-1.0	✓	✓	!	✓	✗	✗	✗	✗	✗	✗	330	BlueBERT [190]
C-UDA	✓	✓	✓	✗	!	!	✗	✗	✓	✓	72	Data for computational use only
†PDDL	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	50	Database-specific license
CC-BY-ND-4.0	(✓)	✗	✗	✓	✗	✗	✓	✗	✗	✓	47	Disallow making derivatives
*GFDL	>Same as GPL, a free document license										30	txtai-wikipedia
*ODbL	✓	✓	✗	✓	✗	✗	✓	✓	✗	✓	20	Automatic relicensing
*LGPL-LR	✓	✓	✗	✗	!	!	✓	✓	✗	✓	19	LGPL for linguistic resources
ODC-By	✓	✓	✗	✓	✗	✗	✗	✗	✗	✓	15	Automatic relicensing

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. On the other hand, the original licensing frameworks for software and content (e.g. MIT, CC BY) 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. CreativeML OpenRAIL-M license, proposed by Responsible AI [43], 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 work to reuse them in bulk. It is therefore imperative to establish guidelines for selecting the licenses for models and other related components that are ready for query-based FL.

4.3.2 License Choosing Preferences. In query-based FL, the model community aims to promote the reuse of models contributed by users, which raises unique concerns about model licensing:

- 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 [37]; and
- Some licenses are **copyleft** (marked with * in Table. 3), which means the derivatives must be licensed under the same license or a compatible license, leading to potential license conflicts and proliferation [76]; last
- All granted rights are preferably **irrevocable** by the licensors [204].

Furthermore, 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, based on the comparisons between different licenses provided in Table 3, we present several strategies for selecting licenses in query-based FL to minimize conflicts.

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

Our recommended licenses for training datasets and databases for query-based FL are CC0-1.0, PDDL, ODC-By and CC BY (the preferred version of CC BY is 4.0 due to the grant of *Sui Generis Database Rights in Art.1c*). CC0-1.0, PDDL 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. Additionally, CC0-1.0 and PDDL are public domain licenses and do not require the declaration of the original license. Although some of these licenses do not explicitly grant sublicensing rights, they provide an **automatic licensing** policy for downstream recipients.

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 of data is not allowed. Nonetheless, C-UDA explicitly exempts reused results from any restrictions, which is highly favorable for our scenario of model mining. 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 [30], 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. Note that FOSS 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, on the other side, these licenses also 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 ML models. Although some of those copyleft licenses can be made compliant with others, we recommend isolating the software licenses from the resulting models to preserve the freedom to use the models further (e.g. close-source, relicense).

Preferences for Models: Apache-2.0, AFL-3.0, Artistic-2.0, ECL-2.0 > OpenRAIL series.

There are two recommended choices for model licenses for query-based FL. The first is permissive FOSS 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 [81, 192]. The user behavioral restrictions stated 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. Therefore, including such statements in licenses may ultimately lead to the licensed materials becoming closed source. Additionally, to enable remote control for the responsible use of AI, CreativeML OpenRAIL-M includes the clause *You shall undertake reasonable efforts to 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,

[16] https://case.law

traditional permissive licenses, which follow worse-is-better design philosophy [67], 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.

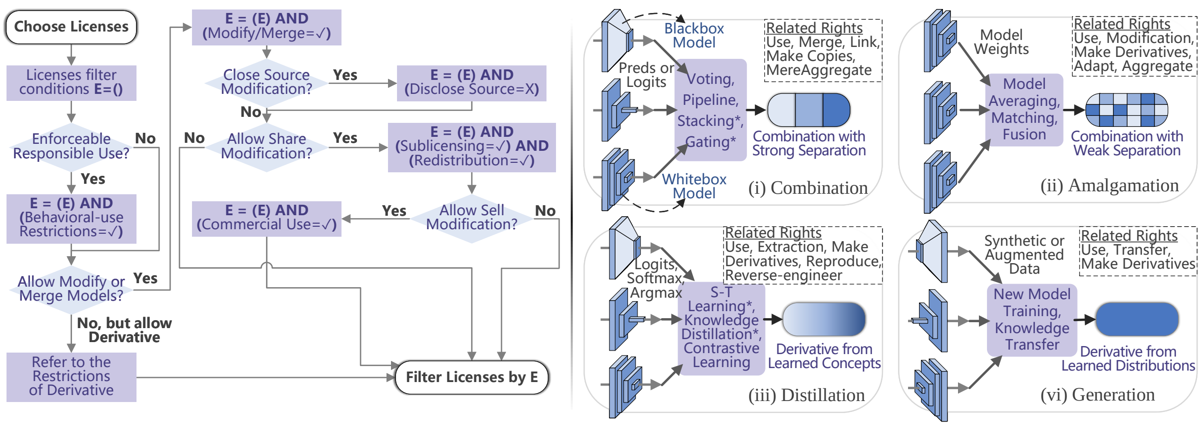
It is worth noting that the above discussion only deals with the licenses of inputs for open FL platforms, which aim to provide legal compliance and freedom of outputs as much as possible, but does not involve the copyright issue for the outputs. In fact, except for some public domain dedication licenses 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.

4.3.3 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* [184]. 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 [93].

The determination of copyrightability and authorship of computer-generated content is an open issue that needs to be addressed through corresponding legislation [93, 173, 184]. European Parliament regarded that *consideration must first be given to assessing patent law in the light of the development of AI* [17]. 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 [93]. Licensors can also make efforts to clarify this issue by including relevant claims in their licenses. For example, the license of Stable Diffusion [210] explicitly states that *Licensor claims no rights in the Output You generate using the Model*. Similarly, ChatGPT [18], even though it is a proprietary software of OpenAI company, its sharing & publication policy [19] 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 [298] and SEER [80] 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 further provide a flowchart in Fig. 4(a) to guide the license selection in the context of model query and model reusing.

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. The terms definition for model reusing in different licenses is a novel and interesting issue that is rarely discussed. For example, interpreting model reusing as creating

[17] https://www.europarl.europa.eu/doceo/document/A-9-2020-0176_EN.html
[18] https://openai.com/blog/chatgpt
[19] https://openai.com/policies/sharing-publication-policy



(a) Model Licenses Selection Flowchart (b) ML Batch Model Reuse Mechanisms

Fig. 4. (a) Flowchart for model licenses selection in the context of model query and model reusing. (b) Proposed taxonomy categorizing batch model reuse mechanisms based on the reused results.

derivatives or combinations would involve different clauses in the licenses. To provide a better understanding of these implications, let's first provide an overview of typical model reuse mechanisms.

4.3.4 Batch Model Reuse Mechanisms. Instead of summarizing the batch model reuse mechanisms from a technological and algorithmic aspect, we propose grouping 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 and may correspond to different regulations in licenses.

Combination [301] 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 [104, 263]. 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 [193], which can be determined by stacking estimators [263], Bayes approach [42] or backpropagation of gating networks [104]. Voting is a workaround strategy 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 [266] 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 raw 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.* [69] proposed a zero-shot dense retrieval system named HyDE by pipelining a natural language generation (NLG) model [186] and a natural language understanding (NLU) model [103]. 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. Similarly, through query-based FL, we can query a vicarious NLG model for a novel scenario, such as ProGen [171] 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. Meanwhile, 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 [168], the statistical heterogeneity and model heterogeneity nature of these crowdsourced models can actually enrich population diversity, which is crucial for creating a good ensemble [170, 185]. **However**, the storage consumption of combined models increases linearly with the number of base learners, which can strain the communication resources of a collaborative model training network. As a result, alternative approaches involve amalgamating or merging multiple models to create a new consensus model. In the following, we provide a summary of these methods.

Amalgamation involves combining models through model parameters granularity operations, such as median [17, 196] and coordinate-wise averaging with consideration of heterogeneity [147, 176], security [233], scalability [205], matching [251, 278], specificality [82], generalizability [199], 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. However, in order to avoid confusion with the term "combination," which is frequently used interchangeably with "aggregation" in software licenses (e.g., Artistic, GPL), we opt to use the term "amalgamation" instead.

FedAvg [176] is the most popular model averaging method in FL with many follow-up works. For instance, Sun *et al.* [233] proposed applying norm thresholding of local model updates to defend against backdoor attacks. Similarly, Blanchard *et al.* [17] proposed using more robust median-based amalgamation for resilience against Byzantine behavior. Consider the ordering of parameters, Wang *et al.* [251] match and average the NNs parameters layer-wise across clients, based on their similarities. Yu *et al.* [278] further attribute the misalignment in FL models to the non-IID training data, propose allocating an independent structure for each class and updating models through a feature paired averaging strategy. **Even though** amalgamation mechanisms can achieve a balance between model performance and resource efficiency by maintaining only one global model, they often rely on multiple rounds of communication to converge, which is not applicable in a query-based FL scenario. Therefore, instead of trying to directly concatenate multiple sources of models while dealing with intricate parameter mismatch, transferring the latent knowledge learned by local models to a new model is a good alternative (i.e. Federated Distillation [105, 113]).

Another direction of model amalgamation is leveraging Bayesian nonparametrics to learn the shared global latent structures among local models [129, 282, 283]. 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 regarded 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. Most recently, Su *et al.* [227], inspired by null-space in continual learning [124, 256], propose MA-Echo which leverages layer-wise projection matrices to preserve the original loss of local models after amalgamation. Their evaluation results present a moderate improvement in one-shot setting compared to FedAvg and ensemble. **Unfortunately**, this improvement is not consistently observed in multiple-round experiments. Meanwhile, to tackle the issue of catastrophic forgetting, FedPR [63] follows similar ideas to facilitate the server's learning of visual prompts from clients for MRI reconstruction applications, **but** the improvement compared to FedAvg is limited even in multi-round setting.

Table 4. Summary of privacy-preserving **Distillation** works in the field of FL. Some works are listed multiple times because they contain multiple KD procedures with different strategies. Works with naming conflicts are distinguished by subscript.

Strategies		FL Studies
KD@Server	w/ Validation Set	FedED [228]
	w/ Unlabeled Data	FedDF [157], One-shot FL [83], FedBE [35], PerAda [267], FedET [41]
KD@Client w/ Local Data	from Global Model	FedFusion [274], FedKD_2 [265], MOON [142], FedNTD [132], FedMLB [120], FedCAD [92], FedAlign_1 [177], FedAlign_2 [292]
	from Other Clients' Model	FedMatch [106], CCL [6], FedProto_1 [178], FedProto_2 [236]
	from Self Model	FedFusion [274], FedDistill [111], FedKD_2 [265], MOON [142], FCCL [100], pFedSD [114], RSCFed [154], CCL [6]
KD w/ Public Unlabeled Datasets		FedKT [141], FedMD [134], FedAD [77], FedMD-NFDP [229], FCCL [100], FedAUX [216], RHFL [62], FedKD_1 [78], Cronus [31], KT-pFL [291], DS-FL [102]
KD w/ Generated Data (DFKD)		DENSE [289], FedCAVE-KD [95], FedGen [305], FedFTG [295]
KD w/ Differential Privacy		FedKC [252], FedSSL [58]

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) [42] 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 example, the coordinate-wise operations or the 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.* [97] 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[47]. Each specialist only learns fine-grained distinctions of a subset of classes, which is very similar to the non-IID setting in FL [140]. By using Knowledge Distillation (KD), we can also compress wide and deep teacher networks into lightweight student networks [211], which is promising for addressing system heterogeneity in cross-device FL [156]. Therefore, it is natural to extend the KD technologies to FL filed [35, 77, 111, 134, 141, 157, 229, 265]. Actually, many recent FL works [6, 31, 62, 77, 78, 90, 100, 102, 134, 166, 178, 228, 229, 236, 291] solely leverage KD without following the model averaging paradigm of FedAvg, we leave this discussion for later.

Despite directly retraining a Generalist model through knowledge distillation (KD), an alternative approach is to construct an ensemble of knowledge. For example, Furlanello *et al.* [66] consecutively generate student models with the guidance of knowledge distilled from earlier generations and find that the ensemble of multiple generations of internal models achieves state-of-the-art performance. Dvornik *et al.* [55] leverage the distilled knowledge from each learner to encourage cooperation and prediction diversity within the population, which leads to better ensemble results. In the context of FL, the main **advantage** of distillation is the decoupling between knowledge distillation and knowledge learning. This allows us to split the model architecture for the purpose of system heterogeneity and efficiency [241, 247]. Moreover, the well-learned knowledge from clients only needs to be communicated once [77, 78], while the server can perform multiple epochs of local training to complete the transfer. The **drawback** of KD is that it is data-dependent and the shared knowledge may be extracted from local sensitive data, which exposes a new attack surface for potential model inversion attacks [65, 121]. To mitigate this issue, some efforts [58, 252] have been made to add differential privacy noise [56] to the shared content.

In general, there are three mechanisms for avoiding the sharing of sensitive knowledge. First, push the KD procedure to the server-side, where the knowledge of local models is transferred to the global model through a validation set [228] or unlabeled dataset [35, 41, 83, 157, 267] held by the server. Second, we can keep the KD

Table 5. Classification of **Generation** works in the field of FL. Some of the works are also listed in Table 4 because they utilize hybrid model reuse mechanisms.

	Enriching Training Set	Improving Generalization	Enabling Semi-supervised Learning
For Training	FedSage+ [293], GFL [40], FRD [28]	Fed-ZDA [88], FOSTER [280], FedDG [159], DynaFed [195], SDA-FL [152], FedMix [276], FedCAVE-Ens [95]	FedDISC [270], SemiFL [50]
For KD	DENSE [289], FedBE [35], FedDyn [113], FedZKT [296]	FedGen [305], FedFTG [295], FD+FAug [105], FedCAVE-KD [95]	FedSSL [58]

procedure at the client-side, allowing the knowledge of the global model [92, 120, 132, 142, 177, 265, 274, 292], other clients' models [6, 106, 178, 236] or self-model [100, 111, 114, 142, 154, 265, 274] to be transferred based on the local training data. In the above two strategies, only model parameters are exchanged in the training network, which means they can provide the same level of privacy protection as traditional FL.

The last mechanism is to assume that a public unlabeled dataset, which does not contain sensitive information, can be accessed by both the server and clients for KD [31, 62, 77, 78, 100, 102, 134, 141, 216, 229, 291]. Sharing these extracted contents will not raise any privacy concerns, and only minimal communication is generated during KD for the purpose of aligning sample IDs. In cases where such public datasets are not available on the server, a recent approach called Data-Free Knowledge Distillation (DFKD) has been proposed by Lopes *et al.* [163]. This approach regenerates batches of data based on layer activation statistics or spectrum coefficients collected during training phase, and this synthetic data is then used for distillation. DENSE [289] is the first attempt to extend DFKD to FL. It leverages the ensemble of local models to guide the training of a data generator on the server, and the generated data is then used to distill the knowledge from local models to the global model. FedCAVE-KD [95] leverages locally trained conditional autoencoders (CVAEs) [122] to generate samples based on the data distribution of clients. These CVAEs are sent to server used to construct a global generator via KD, which will later provide synthetic training data for the global discriminator.

It is worth noting that in the query-based FL setting, direct access to the original data is not available, thus the second mechanism mentioned earlier cannot be directly applied. A circumvention method is to train a generator following the inspiration of DFKD. Fortunately, this is praticable if the workflow and history information of modeling are tracked and queryable, as we advocated in Section 4.2.2. Recalling that, as shown in Fig. 4(b), Generation is the last category in our taxonomy. Actually, such a hybrid model reuse strategy is quite common in FL. For example, the previously mentioned DENSE [289] incorporates three model reuse mechanisms: Combination (creating an ensemble), Generation (generating synthetic data), and Distillation. Therefore, our taxonomy can cover traditional FL works, such as FedAvg and MOON [142], as well as the broad sense FL, including Federated Distillation [105, 113] and Ensemble Learning [223, 255]. We provide a comparison of these hybrid works in Section 4.3.5. The summarize of above privacy-perserving KD works is given in Table. 4.

Generation is designed to generate synthetic samples that resemble the original data distribution by building a probabilistic model [72] or deep learning model [27, 79, 122] that can capture the underlying distribution pattern and latent structure of original data. Generally speaking, generation techniques can be classified into three categories: data-level, probabilistic, and representation-based approaches. Data-level approaches involve sample granularity operations such as interpolation [33, 288] and augmentation [264] to generate synthetic features based on the original feature space of data and share. Even though these methods are training-free and easy to implement, they cannot be directly applied to the FL setting due to privacy concerns. Recent proposed FedMix [276] aims to alleviate the negative effect of non-IID data by using *mixup* [288], where the average of local data is linearly interpolated with the training data to generate augmented samples. However, the potential risk of data leakage when sharing the mixup data is not comprehensively evaluated in the original work.

Probabilistic approachs aim to estimate the real data distribution using probabilistic method. For example, Markov Chain Monte Carlo (MCMC) [72] methods construct a Markov Chain that converges to the desired target distribution by iteratively proposing new states based on the current state of the chain and acceptance probabilities. Gaussian Mixture Models (GMM) assumes the data is generated from a mixture of Gaussian distributions and can generate new samples by sampling from the learned distributions. To generate the high-dimensional structured data, representation-based approaches try to reconstruct the data from laten feature space. For instance, Variational Autoencoders (VAEs) [122] learn the distribution of the latent representation space given the observed data and then use a decoder network to reconstruct data based on sampled latent representations. Generative Adversarial Networks (GANs) [79] train a generator network to produce samples that resemble realistic data by optimizing an adversarial objective against a discriminator network.

Compare to the other model reuse mechanisms, generation has three unique **advantages**. The first advantage is visualization and verification. Unlike the extracted knowledge in distillation, the quality of generated content can be visualized and validated by humans. This capability aids in assessing the contributions made by participants in terms of generating valuable content. Second, the flexibility of generation methods allow us to generate data with any desired amount or class, which enables more effective handling of imbalanced [33] and non-IID [295] data. The third advantage is multi-format sharing. Participants have the freedom to choose the form of their contributions. For example, they can upload the learned generative models in source code (e.g., Stable Diffusion [210], GPT-2 [200]) or binary form, upload synthetic data, or provide model inference APIs like ChatGPT. This sharing policy can greatly empower the model community in open FL platforms, fostering collaboration and knowledge sharing.

Given the aforementioned advantages, generation methods have been extensively studied in the field of FL. As summaried in Table. 5, these works can be classified into two main categories: generation for training [28, 40, 50, 88, 95, 152, 159, 195, 270, 280, 293], generation for KD [35, 58, 95, 105, 113, 289, 295, 296, 305], and serving three purposes: enriching the training set [28, 35, 40, 113, 289, 293, 296], improving generalization ability [88, 95, 105, 152, 159, 195, 280, 295, 305], and enabling semi-supervised learning [50, 58, 270]. As an example of generation for training, FedSage+ [293] trains a missing neighbors generator to mend the links between cross-subgraph nodes, thereby increasing the connectivity of local data and benefiting from this collaboration across clients. Previous mentioned FedCAVE-KD [95] is an example of generation for KD, where locally trained CVAEs and local label distributions are uploaded to the server for DFKD, ensuring privacy while also enhancing generalization of global model. Another example is FedGen [305], where the generator is maintained by the server and sent to clients in each round. The generator has knowledge about the global view of the data distribution, which is used to KD into the local models, thereby enhancing their generalizability. The last application of generation is in semi-supervised learning, which is a common real-world scenario where the client data is unlabeled. For example, FedDISC [270] leverages the average and cluster centroids of hidden representations across pseudo-labels as input to a pre-trained diffusion model, aiming to generate high-quality samples for training.

In fact, the three purposes of generation correspond to three types of data heterogeneity in FL [140]: Quantity Skew, Label Distribution Skew (non-IID), and Missing Labels, which are challenging to address with traditional model amalgamation methods. The hybrid model reusing strategies, which leverage each other's strengths, have become a common paradigm in recent FL studies [40, 95, 113, 267, 270, 280, 292]. Therefore, in the next section, we will summarize the popular hybrid model reusing studies in FL and then filter the studies that are suitable for query-based FL platforms.

4.3.5 Hybrid Model Reusing in FL. Following the taxonomy we introduced in Section 4.3.4, it can be observed that almost all FL studies can be regarded as a permutation of four model reuse mechanisms: Combination, Amalgamation, Distillation, and Generation. To enhance our understanding of current model reusing studies

Table 6. Comparative analysis of FL studies categorized by our taxonomy for batch model reuse mechanisms. Studies applicable and conditional applicable to query-based FL are marked with different colors; Purple denotes operations completed on Server, or knowledge distilled from Global or Consensus Model; Blue denotes operations completed on Clients, or knowledge distilled from Local, Personlized, or Generative Models; **Knowledge** or **Generated Content** based on **Public, Proxy or Generated** data, and *Knowledge* or *Generated Content* based on *Local, Private or Sensitive* data; [ ]*1: one round of communication (aka one-shot), [ ]*N: multiple rounds of communication, processes ahead ...[ ] are performed only once (i.e. preprocessing), processes inside [...] are main functional part; Slash "/": model training based on *private* data, Comma ",": model training based on **non-sensitive** data; Goals of works: **E**fficiency **H**eterogeneity **P**rivacy.

FL Studies	Combination	Amalgamation	Distillation	Generation	Process	Goals
FedAvg [176]	n/a	Model Avg	n/a	n/a	[/A]*N	EH
FedAD [77]	n/a	n/a	KD **Attention, Logits**	n/a	/[D]*1	HP
FedKD$_1$ [78]	n/a	n/a	KD **Weighted Logits**	n/a	/[D]*1	EHP
FedED [228]	n/a	n/a	KD **Logits Avg on Global Validation Data**	n/a	[,D]*N	EHP
FedIris [166]	n/a	n/a	KD *Hidden*	n/a	[D]*N	H
FedProto [236]	n/a	n/a	KD *Per-Class Hidden Avg*	n/a	[/D]*N	EH
FedMD [134]	n/a	n/a	KD **Logits Avg**	n/a	[D/]*N	H
FedMD-NFDP [229]	n/a	n/a	KD **Logits/Softmax/Argmax Avg**	n/a	/[D/]*N	HP
DS-FL [102]	n/a	n/a	KD **Entropy Reduced Logits Avg**	n/a	[/D]*N	EH
RHFL [62]	n/a	n/a	KD **Weighted Logits**	n/a	/[D]*N	EH
KT-pFL [291]	n/a	n/a	KD **Learned Weighted Softmax**	n/a	[/D]*N	EH
Cronus [31]	n/a	n/a	KD **Robust Mean Estimation of Softmax**	n/a	/[/D]*N	EHP
FedDISC [270]	n/a	n/a	n/a	*Synthetic Data*	[G]*1,	EH
FRD [28]	n/a	n/a	n/a	*Mixup Data*	[G]*1,	E
One-Shot FL [83]	Output Avg	n/a	KD **Softmax**	n/a	/[CD]*1	EP
FedDF [157]	n/a	Model Avg	KD **Logits Avg**	n/a	[/AD]*N	HP
PerAda [267]	n/a	Adapter Avg	KD **Logits Avg**	n/a	[//AD]*N	EHP
FedFiMa [34]	n/a	Rep. Layer Avg	KD **Hidden Avg**	n/a	[/AD]*N	EH
FedFusion [274]	n/a	Model Avg	KD *Hidden*, *Hidden*	n/a	[DA]*N	E
FedNTD [132]	n/a	Model Avg	KD *Not-True Classes Softmax*	n/a	[DA]*N	H
FedKC [252]	n/a	Model Avg	KD *Clustered Hidden Avg*	n/a	[DA]*N	HP
FedDistill [111]	n/a	Model Avg	KD *Softmax of Latest Local Model*	n/a	[/DA]*N	H
pFedSD [114]	n/a	Model Avg	KD *Softmax of Pervious Local Model*	n/a	[/DA]*N	H
FedMLB [120]	n/a	Model Avg	KD *Softmax, Scaled Softmax*	n/a	[/DA]*N	EH
FedAlign$_1$ [177]	n/a	Model Avg	KD *Lipschitz Constants* [221]	n/a	[/DA]*N	EH
MOON [142]	n/a	Model Avg	Contrastive Learning *Hidden*, *Hidden*	n/a	[/DA]*N	EH
FedCAD [92]	n/a	Model Avg	KD *Class-Wise Softmax*	n/a	[/DA]*N	H
FedGKT [90]	n/a	n/a	KD *Logits* KD *Logits, Hidden, Argmax*	n/a	[/D/D]*N	EH
FCCL [100]	n/a	n/a	Contrastive Learning **Logits Avg** Continual Learning *Logits*	n/a	[D/D]*N	H
GFL [40]	n/a	Model Avg	n/a	**Synthetic Data**	/G[,A,]*N	HP
FOSTER [280]	n/a	Model Avg	n/a	**Synthetic Outliers**	,[G/A]*N	EH
FedDG [159]	n/a	Model Avg	n/a	**Interpolated Data**	[G/A]*N	H
SemiFL [50]	n/a	Model Avg	n/a	*Augmented and Mixup Data*	[,G/A]*N	H
NeighGen [293] FedSage [293]		Gradient Avg Model Avg	n/a	*Synthetic Node*	[G/A]*N [/A]*N	H
Fed-ZDAC [88] Fed-ZDAS [88]	n/a n/a	Model Avg n/a	n/a n/a	**Zero-shot Synthetic Data** **Zero-shot Synthetic Data**	[G/A]*N [/G,A]*N	HP
FedMix [276]	n/a	Model Avg	n/a	*Mixup Data Mixup Data*	[GA,]*NG,	HP
FedDyn [113]	n/a	n/a	KD *Hidden, Logits Avg*	**Synthetic Data**	[/GD]*N	E
FD+FAug [105]	n/a	n/a	KD *Per-Class Logits Avg*	**Synthetic Data**	[/G/D]*N	EP
FedCAVE-Ens [95] FedCAVE-KD [95]	Collection n/a	n/a n/a	n/a KD **Softmax**	**Synthetic Data**	/C[G,]*1 /C[GDG,]*1	H H
Fed-ET [41]	n/a	Rep. Layer Avg Model Avg	KD **Argmax, Weighted Logits**	n/a	[/ADA]*N	EH
FedBE [35]	n/a	Model Avg	KD **Softmax Avg**	**Synthetic Model**	[/AGD]*1	H
DynaFed [195]	n/a	Model Avg	n/a	**Synthetic Data**	[/A]*NG[A,]*N	EH
FedAUX [216]	n/a	Model Avg	Contrastive Learning **Hidden**, *Hidden* KD **Weighted Logits**	n/a	,D[A/D]*N ,D[A/D]*1	HP
FedKD$_2$ [265]	n/a	Gradient Avg	KD *Hidden, Attention, Logits* KD *Hidden, Attention, Logits*	n/a	[/DAD]*N	EH
FedGen [305]	n/a	Model Avg	KD **Softmax** KD **Logits Avg**	**Augmented Data**	D[/G,AD]*N	EHP
SDA-FL [152]	n/a	Model Avg	KD **Argmax**	**Synthetic Data** *Augmented Data*	/G[G/AD]*N	HP
FedKT [141]	Voting Voting	n/a	KD **Argmax** KD **Argmax**	n/a	[/CDCD]*1	HP
FedMatch [106]	Voting	Model Avg	KD *Argmax*	n/a	[C/DAA]*N [,CDA]*N	EH
FedFTG [295]	n/a	Model Avg	KD **Softmax**, **Softmax** KD **Softmax**	**Synthetic Data**	[/AGDD]*N	EH
RSCFed [154] (Unlabeled Case)	n/a	Model EMA Model Avg	KD *Softmax*	n/a	[DAAA]*N	H
FedAlign$_2$ [292]	n/a	Model Avg	Contrastive Learning *Argmax, Argmin*	n/a	[DDAA]*N	EH
FedSSL [58]	n/a	n/a	KD **Softmax** KD **Softmax**, *Interpolated Softmax*	**Synthetic Data**	[/GD/GD]*N	HP
DENSE [289]	Collection	n/a	KD **Logits Avg**, *Batch-Wise Statistics*, **Softmax** KD **Softmax**, **Softmax**	**Synthetic Data**	/C[G,DGD]*1	HP
FedZKT [296]	n/a	n/a	KD **Softmax**, **Softmax Avg** KD **Softmax Avg** KD **Softmax**	**Synthetic Data**	[/GDGDGD]*N	H

and identify methods applicable for constructing an open FL platform, we provide a comprehensive summary in Table 6. We employ different colors and fonts in Table 6 to emphasize the distinctions among studies, while certain processes have been omitted without ambiguity. In addition, we have listed the main goals of each study, and only those with explicit designs, experiments, or proof are counted. Please refer to the table caption for the explanation of our denotations. As an example, the process of RSCFed [154] involves the following steps: ① KD, the knowledge is extracted as the softmax values from the self model based on local private data (ref. Table. 4) and performed at the client-side; ② Model exponential moving averaging performed at the client-side; ③ Simple Model Averaging performed two times at the server-side. These processes are repeated across multiple communication rounds until completion. In this way, we can categorize these works and make intuitive comparisons between them. We demonstrate the usage of our taxonomy by presenting three novel findings based on it.

Finding 1. Amalgamation and Distillation ([AD] or [DA]) are the most popular combo in FL studies, followed by Distillation as a standalone ([D]) strategy and Generation before Amalgamation ([GA]).

Due to the presence of parameter mismatch [251, 278, 282], using solely amalgamation often lead to weight divergence [150, 235], particularly in scenarios with high data skew. As a result, distillation, which provides a global view of the data distribution, serves as a complementary solution for addressing data heterogeneity (indicated as "H" in Table 6). However, to achieve acceptable performance, amalgamation requires multiple accesses to local data for training and multiple communications for model averaging ([AD]*N or [DA]*N), which makes it challenging to apply in one-shot FL settings.

On the other hand, we can fuse the distilled knowledge for training without any model amalgamation ([D]*N or [D]*1). However, additional precautions should be taken to prevent sensitive knowledge leakage. One common approach is to introduce an auxiliary public unlabeled dataset for knowledge extraction (except for FedED [228], which uses labeled validation set, ref. Table 4). Getting rid of the limitation of amalgamation, these distillation studies (e.g., FedAD [77], FedKD_1 [78]) enable the feasibility of one-shot learning, making them good candidates for constructing a query-based FL platform. Meanwhile, the generation approachs also offer the same one-shot capacity (i.e., FedDISC [270] and FRD [28]) as distillation by directly generating synthetic data for fusion. This one-shot capacity can also be extended to the hybrid case, such as FedCAV [95], FedBE [35], FedAUX [216], DENSE [289]. That is why the final step in all one-shot solutions is typically distillation or generation on the server-side (excluding local training), rather than amalgamation. But it is worth noting that the one-shot feasibility is not sufficient and necessary in the context of query-based FL. For example, in FRD [28], participants generate mixup data for server training rather than contributing their models. Currently, it is evident that the once strong connection between model averaging and FL has become blurred, and distillation has emerged as a popular method in the field of FL.

Finding 2. Combination is the least common model reusing method in FL studies.

One drawback of using combination in the FL context is the high communication consumption required for broadcasting the combined model to multiple participants, which can be expensive in terms of bandwidth and latency. Hence, a practical approach is to compress the contributed models through methods like amalgamation or distillation, which may already implicitly include combination. For example, in FedAvg, the FL server collects local models from clients and averages them immediately. However, to avoid redundancy, we did not include these temporary combinations in Table 6.

Finding 3. Amalgamation after Distillation ([DA]) primarily distills knowledge from private data, while Distillation after Amalgamation ([AD]) and solely Distillation ([D]) primarily distill knowledge from non-sensitive data.

This difference arises from two different FL agreement mechanisms. One strategy is to leverage model amalgamation to achieve agreement ([DA]). For example, the knowledge from multiple data sources is generalized by applying appropriate weighting before being transferred to local models, which are then averaged to achieve the target model. Another strategy is to leverage a public dataset to achieve agreement ([D] or [AD]). In this approach, the knowledge distilled from multiple local models can be fused to a consensus agreement and then applied to the target model, eliminating the need for model amalgamation.

Question: How to select the suitable model reusing method for query-based FL?

From the previous discussion, we have demonstrated how to summarize recent FL studies from a model reusing perspective. Similarly, to select a suitable method for query-based FL, we can refer to Table 6 and consider the following factors:

- **Data Dependency**. Recalling that query-based FL is contactless (Section 4.1), which means we cannot access the training data again once the model has been uploaded by the entity. Therefore, any method that requires multiple communication rounds of local or sensitive data access is not applicable to query-based FL. Such as, local data dependency due to local training (FedMD [134], DynaFed [195], etc.), distillation (FedFusion [274], MOON [142], etc.), generation (SemiFL [50], NeighGen [293], etc.). So, refer to Table 6, the filtering criteria for unqualified methods can be expressed as [.../...]*N, [...D...]*N with *knowledge in italic* and [...G...]*N with *generated content in italic*.
- **User Privacy**. Due to the openness of the system, directly sharing embeddings from a pre-trained encoder or mixup of local data poses a risk of leakage (FedDISC [270], FRD [28]). It is strongly discouraged to include such methods, even if they have one-shot feasibility.
- **Compatibility**. Additional, since query-based FL is model agnostic, it is not always possible to guarantee the compatibility of coordinate-wise operations required by model amalgamation. Therefore, the applicability of methods that involve model amalgamation may be limited in this context. We call these methods are conditional applicable to query-based FL.

In summary, we have marked the applicable and conditionally applicable methods to query-based FL in Table 6 in red and orange, respectively. Even though the KD procedure of DENSE [289] relies on batch-wise statistics collected during local training, we consider this information to be non-sensitive. Therefore, we include it in the applicable methods. Please note that some single combination or amalgamation approaches, such as Fed-ensemble [223] (only combination) and Models Fusion [129, 282, 283] (only amalgamation), can be directly applied in a query-based FL setting. However, to avoid redundancy, these studies have been omitted from Table 6. By now, we have witnessed the success of using the new taxonomy, and in the next section, we will discuss how to combine our taxonomy with the context of model licenses.

4.3.6 Model Reusing vs. Model Licenses. We have demonstrated that almost all FL studies can be regarded as a permutation of model reusing methods. Therefore, we can further analyze the corresponding clauses in model licenses perspective for FL studies by decomposing them into model reuse methods. Recalling that there are four kinds of model reusing mechanisms: Combination, Amalgamation, Distillation, and Generation, which result in four forms of outputs: Combination with strong separation, Combination with weak separation, Derivative from learned concepts, and Derivative from learned distributions, respectively. By interpreting "work" as "model" rather than as "software", we can easily identify the applicable clauses for different model reusing mechanisms.

We present the analysis of model licenses for model reuse in Table 8. The analysis covers the most popular model licenses (used by over 100 released models) on Hugging Face, which include free software licenses, free content licenses, and AI model licenses. For simplicity, we have merged some similar licenses without loss of clarity. To avoid license conflicts when engaging in batch model reuse, it is preferable for the reused model to be considered an **independent** work or fall under an **undefined** category, not covered by the original license. Additionally, we can refer to Table 8 and consider the following factors:

- **Differences in terminology**. In the definitions of licenses, connecting multiple works into a separable union is typically referred to as *aggregation, redistribution, and reproduce*, which is completely different from the concept of *model aggregation* in the ML field. Therefore, it is necessary to refer to Table 8 for the

Table 8. Analysis of the model licensing clauses corresponding to different batch model reuse mechanisms, denoted by "keywords in licenses" -> "delineation of reused results", ✗: undefined. Note: GPL-3.0, CC-BY-SA, CC-BY-NC-SA are copyleft.

	Combination	**Amalgamation**	**Distillation**	**Generation**
	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 constitutes a covered work -> Independent Work	Output no constitutes a covered work -> Independent Work
Artistic-2.0	Blackbox: Merely Extend -> Own Work Other: Link -> Own Work	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	✗	✗

corresponding delineations of each model reuse mechanism instead of relying solely on the technical name.

- **Restrictions of combination and amalgamation**. As shown in Table 8, the majority of model licenses have specific terms and definitions regarding these two methods, except for blackbox combination. This implies that there may be potential restrictions that need to be taken into account and complied with. These restrictions may also proliferate to the reused results if original licenses of models are copyleft, such as GPL-3.0, CC-BY-SA, CC-BY-NC-SA, etc. Therefore, we recommend avoiding the use of models and datasets under such copyleft licenses (ref. Fig. 4(b) and Tabel 3) for batch model combination and amalgamation.
- **User behavioral restrictions of responsible AI licenses**. OpenRAIL licenses [43] clearly define all four model reuse mechanisms, which means the effect of copyleft-style behavioral-use clauses will spread to the reused results, which can potentially result in the licensed artifacts becoming closed source [81] (ref. Section 4.3.2). To prevent license proliferation [76], it is not recommended to reuse models under these OpenRAIL licenses.

Lastly, we can further analyze the potential license conflicts in hybrid model reuse methods. Let's consider the example of DENSE [289], where the process is /C[G,DGD]*1 and assume that the local models are licensed under GPL-3.0. In this case, the collection of local models should be treated as a whitebox in order to distill the logits value from it. As a result, this collection should be considered a *modified version* of the original works according to the terms of GPL-3.0. However, the generation and distillation processes of DENSE are based on non-sensitive data and result in the creation of an independent work. Considering that GPL-3.0 is a copyleft license, the reused results should be licensed under GPL-3.0 if the new release includes the collection in its

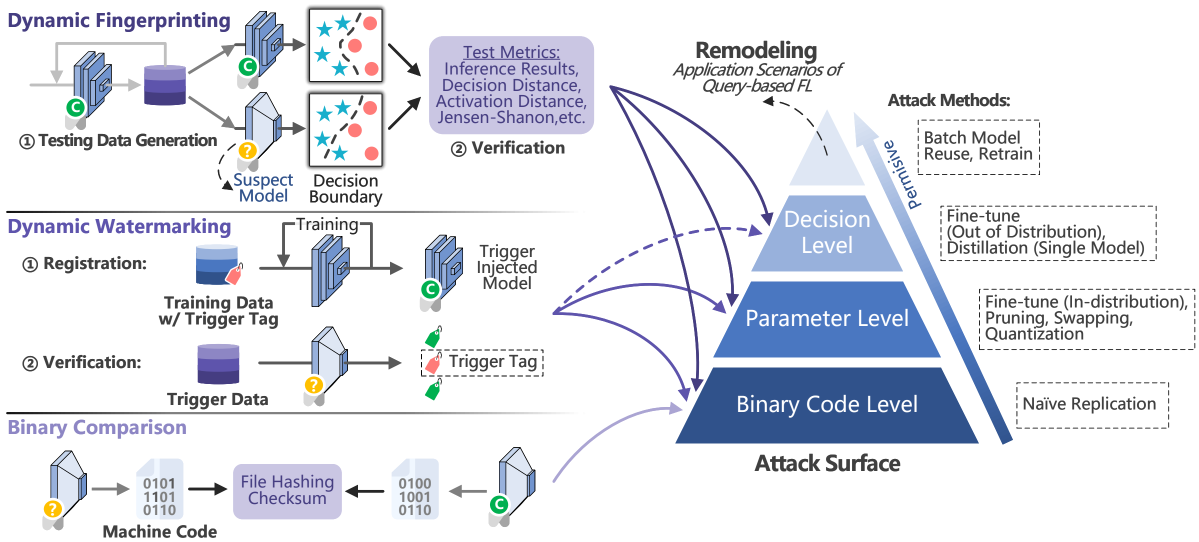


Fig. 5. Overview of DNNs IP protection methods.

whitebox form. Otherwise, if the GPLed local models executed as a separable and replaceable subprogram, the reused results can be considered an independent work without any specific license restrictions.

The example above demonstrates that analyzing hybrid methods under a copyleft license can be complicated. Therefore, selecting models, algorithms, datasets with more permissive licenses, as advocated in Section 4.3.2, can make things easier and facilitate the model reusability of open FL platforms.

4.4 How to Protect Models

In the previous sections, we have extensively discussed the construction of a query-based FL platform from both technical and legal perspectives. Nonetheless, the matter of model management and protection continues to be a significant concern, raising questions such as *How can I protect my models from plagiarism after they are released?* In general, there are three kinds of mechanisms that can be employed to protect DNN models: authorization [87], watermarking [239], and fingerprinting [133]. Authorization mechanisms, including hardware-based memory encryption techniques like TEE, can effectively prevent unauthorized access to the model copy. Additionally, researchers have explored embedding authorization mechanisms within the DNN architecture itself, such as with passport layers [59, 290]. However, the need for specific hardware and modifications to model architectures can significantly increase the barriers for participants and compromise the openness of query-based FL platforms.

As passive protection methods, watermarking and fingerprinting can detect potential infringement while preserving the openness of the model. A watermark can be embedded into DNNs through training or fine-tuning with a modified loss function [244] or trigger data [44] acting as a backdoor [12]. On the other hand, fingerprinting strategies aim to generate an identifier for each model based on snapshots of training history [109] or inference results that approximate decision boundaries [36]. Note that the technical details of IP protection methods are beyond the scope of this survey. We recommend referring to up-to-date surveys such as [189, 232]. Additionally, it is important to recognize that the goal of query-based FL platforms is not to provide perfect protection against model plagiarism and unauthorized use. Interestingly, certain model reusing strategies like KD and ensemble methods are even regarded as model extraction attacks within the realm of DNNs IP protection [32].

This highlights the inherent contradiction between promoting model reusing and preventing all forms of model plagiarism. As a result, it is necessary to determine the scope of protection and strike a balance between model protection and platform openness.

First, it is helpful to consider the following requirements for model protection in the context of query-based FL:

- **Non-invasive**. Any attempt to invasively embed backdoors into the weights and architectures of models is likely to result in changes to their functionalities and could lead to unexpected failures after deployment [36]. Meanwhile, these backdoors can be exploited by attackers to manipulate the model's predictions [151]. Accordingly, it is more recommend leaving the decision of adding invasive protection to the users instead of forcing it through platforms.
- **Compatible**. Due to the model agnostic nature of query-based FL, relying on protection mechanisms that are specific to certain model structures and formats will limit the applicability of the platform. For instance, the passport layer [290] is built upon the normalization layer and requires joint training with the target model for ownership verification. However, in practice, both the normalization layer and whitebox access to the model may be unavailable. Therefore, considering the compatibility of model protection mechanisms is essential to ensure wide support for model sharing in query-based FL.
- **Permissive**. As mentioned earlier, it is not essential to identify and address every instance of plagiarism in query-based FL. On the contrary, we encourage platform users to engage in model mining and reusing, as outlined in Section 4.3.4. Therefore, model protection methods should be permissive enough to allow for model reusing while still identifying the less creative effort operations that result in minimal or no change to the model's functionality, such as naive replication, quantization, pruning, and invariant neural swapping. Note that the presence of sufficient human creative effort is an important criterion for determining the copyrightability of a computer-generated work [184]. The consideration of copyrightability can serve as a guiding principle in determining the scope of model protection in query-based FL.
- **Large-scale cost-effective**. Unlike traditional FL, where the size of training networks in each round is fixed, the size of a query-based FL platform is continuously expandable. Therefore, it is undesirable if the potential conflicts [133], required bit-length, and deployment cost [44, 244] of model protection solutions such as watermarking and fingerprinting increase with the platform size. This suggests that solutions that require modifying optimization goals and training or fine-tuning with trigger data to embed identity information into the target model should be excluded.

Recently, several model protection algorithms have been proposed to address the challenge of model protection in FL. Tekgul *et al.* [239] propose WAFFLE that uses augmented Gaussian noisy images as a watermark set to determine the ownership of the global model. However, this method is invasive and incurs high costs due to multiple rounds of training. To ensure reliable watermarking for FL participants, FedIPR [133] embeds different watermarks into models of different participants through training with additional trigger samples and a modified optimization goal. But this method is designed for model verification on the client-side, so it is independent to the model management in FL platforms. FedTracker [222] embeds personalized local fingerprints into the Batch Normalization (BN) layer to enable traceability of model leakers. One limitation of FedTracker is that its fingerprinting strategies are not compatible with other DNNs that do not use BN or do not provide whitebox access. These examples highlight the incompatibility between the IP protection methods designed for traditional FL and the requirements of the query-based FL scenario, which have different protection goals and targets.

Based on the above observation and the taxonomy of deep IP protection [189, 232], **dynamic fingerprinting strategies with blackbox verification support** are considered suitable model protection methods for query-based FL. These methods aim to approximate the similarities between the decision boundaries of different models by evaluating the value of a well-tailored testing metric on a self-constructed testing set. A typical example is DeepJudge [36], which employs an ensemble of multi-level (Property, Neuron, Layer) testing metrics to achieve confident and robust plagiarism identification. The verification process of DeepJudge can be evaluated in a blackbox setting by calculating distances based on model predictions. However, the generation of testing set in DeepJudge requires whitebox access to the suspect models in order to calculate adversarial samples, which may be unreachable if the suspect models are in binary format.

Another example is Zest [108], which randomly samples several training images and then applies super-pixel and segmentation techniques to construct the testing set. However, this testing samples generation method is limited to images and may expose private training data. A compromise approach is to seek non-sensitive or desensitized testing set from users when they upload their models. Inspired by DeepJudge, we can develop an ensemble of multi-level testing strategies for query-based FL. In the first step, we can compare the hash codes (e.g. MD5, SHA-256) of the raw model weights or its binary execution to rapidly filter out cases of naive replication plagiarism. Then, in the second step, we can filter out cases of quantization or invariant neural swapping plagiarism by comparing the distance in predictions obtained from an out-of-distribution dataset. In addition, we can further evaluate the similarities between models using testing-based approaches such as DeepJudge and Zest, which can detect cases of pruning and direct KD plagiarism. Furthermore, if the workflow information is supports it, we can swiftly eliminate batch model reusing from being suspected of plagiarism. To enhance understanding, we present an overview of the typical methods used for protecting IP of DNNs in Fig. 5.

In summary, finding a balance between model protection and platform openness is indeed a challenge. It is crucial to carefully consider the trade-offs and explore alternative approaches that can provide a reasonable level of model protection without excessively compromising the openness and usability of query-based FL platforms.

4.5 Limitations of Query-based FL

In query-based FL systems, the processes of model production and reuse are decoupled to maximize the autonomy of each participant. However, this losse cooperation paradigm is no longer compatiable with online collaboration ML frameworks, which means that it cannot fully harness participants' communication and computational resource to enhace the training performance. Therefore, even though the server-dominated cooperation frameworks like FedAvg have limitations, as described in Section 3.2, it is still worthwhile to provide support for its underlying distributed ML training methods to enhance the flexibility and compatibility of our open FL platforms. In the next section, we illustrate another cooperation framework named contract-based FL, which follows a mutual choice design philosophy similar to crowdsourcing platforms [246]. It can serve as an extension of traditional FL and query-based FL.

5 CONTRACT-BASED FEDERATED LEARNING

Let us consider another open FL platform based on a contract-based cooperation framework, where cooperation can be built by publishing ML tasks and accepting ML collaboration requests. An overview of this platform is presented in Fig. 6, the major difference between traditional FL (ref Fig. 2) and contract-based FL is that we involve a trustworthy third-party platform called the Contract Platform to host and coordinate ML tasks for platform users. Rather than directly pushing ML tasks from servers to clients, employers need to publish the tasks to the contract platform and wait for acceptance from workers, which involves a mutual choice procedure. The conditions and payment for participation are also appliable, and the final model will be audited and evaluated by the platform for fairness. In addition, privacy-enhanced technologies [96] and fingerprint management [36] can also be implemented by the platform to protect the IP of platform users. The properties of contract-based FL are: 1) **Opt-in**, as workers reserve the right to join or quit from training networks; 2) **Contractual**, enabling employers to define payment, organization mode, model quality criteria, rehire rules, etc. through contracts; 3) **Market-based**, where contracts are open and task pricing is influenced and determined by the market.

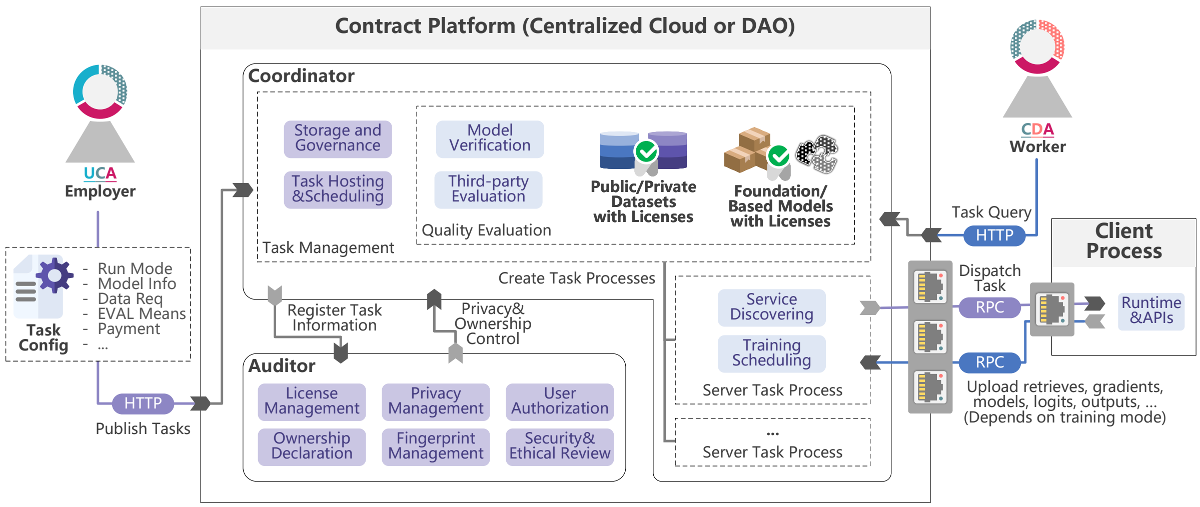


Fig. 6. An overview of contract-based FL systems. (U: model User, C: Coordinator, D: Data owner, A: Auditor)

In short, contract-based FL retains all functionalities related to model training at the platform and only provides task publishing services to employers, who play the role of model training in traditional FL. Furthermore, similar to model-reusing mechanisms (refer to Section 4.3.4) that can be applied to query-based FL, our contract-based FL is not limited to an amalgamation-based collaborative training paradigm like FedAvg. In fact, the contract platform is designed as a crowdsourcing platform among data miners for ML cooperation tasks, including distributed training, fine-tuning, and ensembling based on scratch or Foundation Models (FMs) [281]. However, most current crowdsourcing platforms like Amazon Mechanical Turk[20] and Appen[21] primarily deal with human intelligence tasks such as data collection and annotation (aka. microtasks) rather than ML modeling tasks. It remains unexplored how to design and publish an ML task to seek crowd labor. On the other hand, many studies [13, 18, 48, 51, 84] and products [174, 226, 306] emphasize monetizing ML activities through blockchain-based techniques and Decentralized Autonomous Organizations (DAOs). However, both monetization and decentralization are non-essential functions for our contract platforms, and we will summarize and compare these works later.

5.1 How to Design ML Microtasks

When we come to the scenario of contract-based FL, we first need to define how to design ML microtasks for crowd workers. Unfortunately, previous collaborative ML studies [143, 180] rarely discuss this question because they default to assuming that all employers or workers have the same resource type to fit into their modeling frameworks. On the contrary, in contract-based FL, we leave the freedom of design microtasks and choosing modeling method to the employers.

The answer to this question depends on what resources the employers have and what tasks the workers can perform. For example, in horizontal FL, the server should provide an initial model, and clients should have training data under the same feature space and provide computational power. However, in vertical FL, the clients should have training data under the same sample space (i.e. ID space). This difference determines which modeling framework employers can adopt. Following this idea, we provide comparisons of typical FL, distributed ML, and

[20] https://www.mturk.com/ [21] https://appen.com/crowd-2/

Table 9. Comparisons of typical model training frameworks with the intervention of a trustworthy platform. *, *, * indicate resources from Employer, Platform, and Worker, respectively. Resources in different color columns (e.g., Data in column Platform) represent transmission. ★: Location of model training. *Resources* received from other workers.

	Employer	Platform	Worker
Centralized ML	Model	Model Data ★	Data
FedAvg [176]	Model	Model Gradient	Model Data ★
DMoE [215]	Gating NN Data	Gating NN Data Output ★	Model Data ★
DeDES [255]	Model Subset	Model	Model Data ★
Borzunov *et al.* [22]	Model	Model Data Gradient	Model Data-Stream ★
Moshpit SGD [214]	Model Data	Model Data Gradient	Model Data-Loader ★
VC-ASGD [10]	Model Data	Model Data Gradient	Sub-model H-Data ★
SplitNN [247]	Model ID-Label	Model ID-Label Hidden ★	Sub-model V-Data Gradient ★
DT-FM [281]	FM	FM Data	Sub-FM Micro-Batch *Gradient* ★
FS-LLM [126]	FM	FM Adapter	FM Data ★
FedKSeed [197]	FM	FM Seed Scaler Gradient	FM Seed Data ★
Berdoz *et al.* [15]	Class	Class NN Structure Hidden	NN Structure Data Hidden AVG ★
CCL [6]	Class	Class	Data *Model* *Hidden AVG* ★
FedProto [178, 236]	Class	Class Hidden	Data Model Hidden AVG ★
Ocean [174]	Model Output	Metadata	Metadata Model Data ★

blockchain-based modeling frameworks in Table 9. For simplicity, we introduce the **Platform** between Employer and Worker in some frameworks to make it adaptable for contract-based FL. You can quickly restore the original structure by merging Employer and Platform columns. To represent the transmission of resources, we mark resources provided by each group of entities with different colors. The computational resource used for training is marked with ★. By this way, we can conveniently find the feasible and privacy-preserving modeling frameworks for collaborative learning. Here, we briefly introduce these frameworks group by scenarios[22].

5.1.1 Train from scratch with workers' private data. There are four frameworks that support this scenario. The first is **Centralized ML**, which requires workers to upload their data to a cloud platform for modeling. However, this approach is not feasible for FL scenarios due to privacy concerns. The second is **FedAvg** [176], which requires workers to upload computed gradients upon the scratch model and their private data (horizontally split). Computing resources need to be provided by workers as well. The third is **SplitNN** [247], where each worker trains the cut layers of the model using their vertically split data. The hidden representations from these layers are uploaded to the platform for the training of the remaining layers. The last and distinctive is a Web3 system named **Ocean** [174]. It enables workers to upload only the metadata of their local dataset to a blockchain-based platform to attract buyers. Interested buyers then need to purchase datatokens to access the local dataset and deploy their modeling algorithm. To ensure the privacy and IP security of workers, only trusted algorithms are allowed, and only model predictions will be sent to buyers.

5.1.2 Ensemble workers' private model. As we presented in Section 4.3.4, ensembling is an effective method to integrate knowledge, and the distributed version is proposed by **DMoE** [215]. It employs a Distributed Hash Table (DHT) to store metadata and worker node statuses, constructing a decentralized expert network. To make inferences using this network, each model user must train a local gating network to select a subset of experts

[22] From the viewpoint of Employers.

tailored to their input. No additional training required, Wang *et al.* purposed **DeDES** [255], which select a diverse subsect of weak models from population and make inference by voting. The unique advantage of ensembling lies in its inherent support for model heterogeneity, and its integration is extensible. This capability enables the establishment of a cooperative network while maintaining flexibility and availability.

5.1.3 Train from fundation model. Foundation models, trained on larger-scale data with robust generalization abilities across various downstream tasks, serve as a solid foundation for collaborative training. Yuan *et al.* proposed **DT-FM** [281] to establish an effective geo-distributed learning system across 8 regions for training the language model GPT-3. Similarly, Borzunov *et al.*[22] recruited volunteer nodes to train a transformer model over the Internet. To enhance system robustness, **Moshpit SGD** [214] divides worker nodes into small, independent groups to ensure that all-reduce is not affected by the failure of a single participant. Additionally, **VC-ASGD**[10] divides the deep learning training job into asynchronous parameter update subtasks to improve scalability. In these cases, employers are required to supply the initial model and training data needed to launch training subtasks. This training paradigm is particularly suitable for worker nodes that possess computational and communication resources but lack their own training data.

Instead of initiating training from scratch and consuming substantial computational resources, we can also collaboratively adapt these foundation models to specific local tasks based on relatively restricted hardware and data size through fine-tuning. Recently, Woisetschläger *et al.* [262] have explored the opportunity to fine-tune large language models, such as FLAN-T5, by edge devices. Simultaneously, **FS-LLM** [126] employs Parameter-Efficient Fine-Tuning (PEFT) methods for federated fine-tuning of LLaMA-7B, minimizing communication and computation costs. **FATE-LLM** [60] offers a solution called FedHeteroLLM, leveraging KD to train a mentee model from its local pre-trained LLM for federated aggregation. To alleviate the computational burden associated with backpropagation-based optimization methods, **FedKSeed** [197] employs zeroth-order optimization (ZOO). Only seed and scalar gradients need to be transmitted for federated fine-tuning.

5.1.4 Representations Sharing. Another popular scheme for organizing ML microtasks involves sharing the learned representations of workers. For instance, the use of class-conditional average of last hidden layer activations (aka. Prototypes) can enhance class discrimination across different clients [6, 15, 178, 236]. To configure the microtasks, employers only need to set the target class, after which the platform can collect and distribute per-class prototypes among workers. With no central server required, **CCL** [6] presents a decentralized learning approach based on cross-workers prototypes sharing.

5.2 Decentralization and Monetization

Parallel to contract-based FL, several efforts are underway to build decentralized AI platforms through Web3-based techniques [84]. For instance, Blythman *et al.* proposed decentralized AI Hubs [18], empowered by DAOs (e.g., IPFS [14], Ocean [174], OpenMined), to run ML activities in a trustless setting. Wickström *et al.* [260] propose the concept of AI market which builds a ML analysis network for IoT devices based on Ethereum. Similar to DMoE, Bittensor [226] encapsulates neural network models as services and rewards peers contributing information-theoretic value to the system with TAO coins. Remote peers can be reused, for example, in the form of a MoE, or they can be pipelined to a new service.

The advantages of these blockchain-based systems are that transactions are transparent and workers can easily be incentivized. Therefore, it is a solution when it is challenging to establish a trust relationship between platform users. However, if available, we can replace these DAOs' infrastructural components with a trusted crowdsourcing platform. Like Amazon Mechanical Turk, the incentivized mechanisms is no nessary for contract-based FL platforms as the rewards are set by employers in contract. While many studies consider Shapley value to evaluate the contribution of clients in FL systems [286], we emphasize that a third-party evaluation can also be conducted in a crowdsourcing manner. Similarly, as the model community in query-based FL, contract platforms can further provide IP protection services to users (ref. Section 4.4). Dynamic fingerprints of models can be recorded and used for plagiarism detection. Three kinds of interests should be considered: 1) Interest of employers, to prevent the information leakage of task configurations; 2) Interest of workers, requiring fair evaluation of labor; 3) Interest of the public, to curb the infringement of privacy. However, this is not only a challenge from a technical perspective but also necessitates corresponding legislative oversight to provide guidelines to standardize this kind of AI marketplace commercialization. We believe that a comprehensive regulatory framework is essential to ensure data privacy, and establish fair practices in the evolving landscape of open FL platforms.

6 CONCLUSION

Traditional federated learning systems with a server-dominated cooperation framework suppress the enthusiasm of participants and limit the further extension of such collaboration. To explore the opportunity to establish a more open and reciprocal cooperation platform, we investigate current progress in federated learning, decentralized machine learning, blockchain-based systems, and model reusing systems. In this way, we depict two rough sketches of open federated learning platforms: query-based federated learning and contract-based federated learning. Based on these two proposed platforms, we survey their possible supported techniques and their related legal issues, including ML licensing and copyrightability. We believe this survey can encourage a rethinking of current collaborative ML systems design and lead to the pervasive availability of AI for everyone.

ACKNOWLEDGMENTS

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 (IoT-J)* 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 (IoT-J)* 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] Sai Aparna Aketi and Kaushik Roy. 2024. Cross-feature Contrastive Loss for Decentralized Deep Learning on Heterogeneous Data. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 12–21.

[7] 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 (TII)* 18, 5 (2021), 3501–3509. https://doi.org/10.1109/TII.2021.3119038

[8] 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

[9] 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

[10] Medha Atre, Birendra Jha, and Ashwini Rao. 2021. Distributed deep learning using volunteer computing-like paradigm. In *2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 933–942. https://doi.org/10.1109/IPDPSW52791.2021.00144

[11] 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

[12] 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 (AISTATS)*. PMLR, 2938–2948.

[13] Zahra Batool, Kaiwen Zhang, and Matthew Toews. 2022. FL-MAB: client selection and monetization for blockchain-based federated learning. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing (SAC)*. 299–307. https://doi.org/10.1145/3477314.3507050
[14] Juan Benet. 2014. IPFS - content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561* (2014).
[15] Frédéric Berdoz, Abhishek Singh, Martin Jaggi, and Ramesh Raskar. 2022. Scalable Collaborative Learning via Representation Sharing. In *NeurIPS 2022 Workshop on Decentralization and Trustworthy Machine Learning in Web3: Methodologies, Platforms, and Applications*.
[16] 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).
[17] 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).
[18] Richard Blythman, Mohamed Arshath, Salvatore Vivona, Jakub Smékal, and Hithesh Shaji. 2022. Opportunities for Decentralized Technologies within AI Hubs. In *NeurIPS 2022 Workshop on Decentralization and Trustworthy Machine Learning in Web3: Methodologies, Platforms, and Applications*.
[19] 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*.
[20] 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.
[21] 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
[22] Alexander Borzunov, Max Ryabinin, Tim Dettmers, Quentin Lhoest, Lucile Saulnier, Michael Diskin, Yacine Jernite, and Thomas Wolf. 2022. Training transformers together. In *Proceedings of the NeurIPS 2021 Competitions and Demonstrations Track*. PMLR, 335–342.
[23] Nader Bouacida and Prasant Mohapatra. 2021. Vulnerabilities in federated learning. *IEEE Access* 9 (2021), 63229–63249. https://doi.org/10.1109/ACCESS.2021.3075203
[24] 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
[25] 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
[26] 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).
[27] 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).
[28] Han Cha, Jihong Park, Hyesung Kim, Seong-Lyun Kim, and Mehdi Bennis. 2019. Federated reinforcement distillation with proxy experience memory. In *IJCAI 2019 Workshop on Federated Learning for User Privacy and Data Confidentiality*.
[29] 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 (HPDC)*. 125–136. https://doi.org/10.1145/3369583.3392686
[30] 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
[31] Hongyan Chang, Virat Shejwalkar, Reza Shokri, and Amir Houmansadr. 2021. Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer. In *NeurIPS 2021 Workshop on New Frontiers in Federated Learning: Privacy, Fairness, Robustness, Personalization and Data Ownership*.
[32] Laurent Charette, Lingyang Chu, Yizhou Chen, Jian Pei, Lanjun Wang, and Yong Zhang. 2022. Cosine Model Watermarking Against Ensemble Distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 9512–9520. https://doi.org/10.1609/aaai.v36i9.21184
[33] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research (JAIR)* 16, 1 (2002), 321–357. https://doi.org/10.1613/jair.953
[34] Tianshi Che, Zijie Zhang, Yang Zhou, Xin Zhao, Ji Liu, Zhe Jiang, Da Yan, Ruoming Jin, and Dejing Dou. 2022. Federated Fingerprint Learning with Heterogeneous Architectures. In *2022 IEEE International Conference on Data Mining (ICDM)*. IEEE, 31–40. https://doi.org/10.1109/ICDM54844.2022.00013

[35] 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)*.

[36] Jialuo Chen, Jingyi Wang, Tinglan Peng, Youcheng Sun, Peng Cheng, Shouling Ji, Xingjun Ma, Bo Li, and Dawn Song. 2022. Copy, Right? A testing framework for copyright protection of deep learning models. In *IEEE Symposium on Security and Privacy (SP)*. IEEE, 824–841. https://doi.org/10.1109/SP46214.2022.9833747

[37] 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 (COMAD)*. 1535–1552. https://doi.org/10.1145/3299869.3300078

[38] 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

[39] 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

[40] 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

[41] Yae Jee Cho, Andre Manoel, Gauri Joshi, Robert Sim, and Dimitrios Dimitriadis. 2022. Heterogeneous Ensemble Knowledge Transfer for Training Large Models in Federated Learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI)*. 2881–2887. https://doi.org/10.24963/ijcai.2022/399

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

[43] 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 (FAccT)*. 778–788. https://doi.org/10.1145/3531146.3533143

[44] Bita Darvish Rouhani, Huili Chen, and Farinaz Koushanfar. 2019. DeepSigns: An end-to-end watermarking framework for ownership protection of deep neural networks. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*. 485–497. https://doi.org/10.1145/3297858.3304051

[45] 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 (MLSys)* 3 (2021), 800–811.

[46] 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

[47] 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.

[48] Yang Deng, Tao Han, and Ning Zhang. 2021. FLeX: Trading edge computing resources for federated learning via blockchain. In *IEEE Conference on Computer Communications Workshops*. IEEE, 1–2. https://doi.org/10.1109/INFOCOMWKSHPS51825.2021.9484628

[49] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 17th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. 4171–4186. https://doi.org/10.18653/v1/n19-1423

[50] Enmao Diao, Jie Ding, and Vahid Tarokh. 2022. SemiFL: Semi-supervised federated learning for unlabeled clients with alternate training. *Advances in Neural Information Processing Systems (NeurIPS)* 35, 17871–17884.

[51] Henrique Dias and Nirvana Meratnia. 2022. BlockLearning: A Modular Framework for Blockchain-Based Vertical Federated Learning. In *Inernational Conference on Ubiquitous Security (UbiSec)*. Springer, 319–333. https://doi.org/10.1007/978-981-99-0272-9_22

[52] 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 (OJ-CS)* 1 (2020), 45–61. https://doi.org/10.1109/OJCS.2020.2992630

[53] 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.

[54] 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.

[55] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. 2019. Diversity with cooperation: Ensemble methods for few-shot classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 3723–3731.

[56] 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

[57] 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

[58] Chenyou Fan, Junjie Hu, and Jianwei Huang. 2022. Private Semi-Supervised Federated Learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI)*. 2009–2015. https://doi.org/10.24963/ijcai.2022/279

[59] Lixin Fan, Kam Woh Ng, and Chee Seng Chan. 2019. Rethinking deep neural network ownership verification: Embedding passports to defeat ambiguity attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 32.

[60] Tao Fan, Yan Kang, Guoqiang Ma, Weijing Chen, Wenbin Wei, Lixin Fan, and Qiang Yang. 2023. FATE-LLM: A industrial grade federated learning framework for large language models. *arXiv preprint arXiv:2310.10049* (2023).

[61] 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.

[62] 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.

[63] Chun-Mei Feng, Bangjun Li, Xinxing Xu, Yong Liu, Huazhu Fu, and Wangmeng Zuo. 2023. Learning Federated Visual Prompt in Null Space for MRI Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[64] Patrick Foley, Micah J Sheller, Brandon Edwards, Sarthak Pati, Walter Riviera, Mansi Sharma, Prakash Narayana Moorthy, Shih-han Wang, Jason Martin, Parsa Mirhaji, et al. 2022. OpenFL: the open federated learning library. *Physics in Medicine & Biology* 67, 21 (2022), 214001. https://doi.org/10.1088/1361-6560/ac97d9

[65] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 1322–1333. https://doi.org/10.1145/2810103.2813677

[66] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born again neural networks. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*. PMLR, 1607–1616.

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

[68] 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.

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

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

[71] 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

[72] Charles J Geyer. 1992. Practical markov chain monte carlo. *Statist. Sci.* 7, 4 (1992), 473–483.

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

[74] 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 (IoT-J)* (2022). https://doi.org/10.1109/JIOT.2022.3150363

[75] Ross Girshick. 2015. Fast R-CNN. In *Proceedings of the IEEE international Conference on Computer Vision (ICCV)*. 1440–1448.

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

[77] 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 (ICCV)*. 15076–15086.

[78] 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

[79] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144. https://doi.org/10.1145/3422622

[80] 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).

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

[82] 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*.

[83] 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*.

[84] Shaoyong Guo, Fan Zhang, Song Guo, Siya Xu, and Feng Qi. 2023. Blockchain-Assisted Privacy-Preserving Data Computing Architecture for Web3. *IEEE Communications Magazine* 61, 8 (2023), 28–34. https://doi.org/10.1109/MCOM.001.2200408

[85] 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

[86] 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

[87] 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 (CVPR)*. 3300–3309.

[88] 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.

[89] 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).

[90] Chaoyang He, Murali Annavaram, and Salman Avestimehr. 2020. Group knowledge transfer: Federated learning of large cnns at the edge. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33. Curran Associates, Inc., 14068–14080.

[91] 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*.

[92] 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

[93] 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.

[94] István Hegedűs, Gábor Danner, and Márk Jelasity. 2021. Decentralized learning works: An empirical comparison of gossip learning and federated learning. *Journal of Parallel and Distributed Computing (JPDC)* 148 (2021), 109–124. https://doi.org/10.1016/j.jpdc.2020.10.006

[95] Clare Elizabeth Heinbaugh, Emilio Luz-Ricca, and Huajie Shao. 2023. Data-Free One-Shot Federated Learning Under Very High Statistical Heterogeneity. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*.

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

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

[98] Chenghao Hu, Jingyan Jiang, and Zhi Wang. 2019. Decentralized federated learning: A segmented gossip approach. In *IJCAI 2019 Workshop on Federated Learning for User Privacy and Data Confidentiality*.

[99] 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 (JBI)* 99 (2019), 103291. https://doi.org/10.1016/j.jbi.2019.103291

[100] 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.

[101] 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 (MM)*. 4083–4091. https://doi.org/10.1145/3503161.3548112

[102] Sohei Itahara, Takayuki Nishio, Yusuke Koda, Masahiro Morikura, and Koji Yamamoto. 2021. Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data. *IEEE Transactions on Mobile Computing (TMC)* 22, 1 (2021), 191–205. https://doi.org/10.1109/TMC.2021.3070013

[103] 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).

[104] 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

[105] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. 2018. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. In *NeurIPS 2018 Workshop on Machine Learning on the Phone and other Consumer Devices*.

[106] 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)*.

[107] 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).

[108] Hengrui Jia, Hongyu Chen, Jonas Guan, Ali Shahin Shamsabadi, and Nicolas Papernot. 2022. A Zest of LIME: Towards Architecture-Independent Model Distances. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*.

[109] Hengrui Jia, Mohammad Yaghini, Christopher A Choquette-Choo, Natalie Dullerud, Anvith Thudi, Varun Chandrasekaran, and Nicolas Papernot. 2021. Proof-of-Learning: Definitions and Practice. In *IEEE Symposium on Security and Privacy (SP)*. IEEE, 1039–1056. https://doi.org/10.1109/SP40001.2021.00106

[110] 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 (MM)*. 675–678. https://doi.org/10.1145/2647868.2654889

[111] 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

[112] 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 (TMC)* (2022). https://doi.org/10.1109/TMC.2022.3223578

[113] Cheng Jin, Xuandong Chen, Yi Gu, and Qun Li. 2023. FedDyn: A dynamic and efficient federated distillation approach on Recommender System. In *2022 IEEE 28th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 786–793. https://doi.org/10.1109/ICPADS56603.2022.00107

[114] Hai Jin, Dongshan Bai, Dezhong Yao, Yutong Dai, Lin Gu, Chen Yu, and Lichao Sun. 2022. Personalized edge intelligence via federated self-knowledge distillation. *IEEE Transactions on Parallel and Distributed Systems (TPDS)* 34, 2 (2022), 567–580. https://doi.org/10.1109/TPDS.2022.3225185

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

[116] 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

[117] 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

[118] Shivam Kalra, Junfeng Wen, Jesse C Cresswell, Maksims Volkovs, and HR Tizhoosh. 2023. Decentralized federated learning through proxy model sharing. *Nature Communications* 14, 1 (2023), 2899. https://doi.org/10.1038/s41467-023-38569-4

[119] 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 (ICML)*. PMLR, 5132–5143.

[120] 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.

[121] Jongwon Kim, Sungho Shin, Yeonguk Yu, Junseok Lee, and Kyoobin Lee. 2020. Multiple Classification with Split Learning. In *The 9th International Conference on Smart Media and Applications*. 358–363. https://doi.org/10.1145/3426020.3426131

[122] Diederik P Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*.

[123] 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).

[124] Yajing Kong, Liu Liu, Zhen Wang, and Dacheng Tao. 2022. Balancing Stability and Plasticity Through Advanced Null Space in Continual Learning. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXVI*. Springer, 219–236. https://doi.org/10.1007/978-3-031-19809-0_13

[125] 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

[126] Weirui Kuang, Bingchen Qian, Zitao Li, Daoyuan Chen, Dawei Gao, Xuchen Pan, Yuexiang Xie, Yaliang Li, Bolin Ding, and Jingren Zhou. 2023. FederatedScope-LLM: A comprehensive package for fine-tuning large language models in federated learning. *arXiv preprint arXiv:2309.00363* (2023).

[127] 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

[128] Anusha Lalitha, Shubhanshu Shekhar, Tara Javidi, and Farinaz Koushanfar. 2018. Fully decentralized federated learning. In *NeurIPS 2018 Workshop on Bayesian Deep Learning*.

[129] 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 (ICML)*. PMLR, 5948–5958.

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

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

[132] 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)*.

[133] Bowen Li, Lixin Fan, Hanlin Gu, Jie Li, and Qiang Yang. 2023. FedIPR: Ownership verification for federated deep neural network models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 45, 4 (2023), 4521–4536. https://doi.org/10.1109/TPAMI.2022.3195956

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

[135] 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).

[136] 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 (TIFS)* 12, 6 (2017), 1269–1284. https://doi.org/10.1109/TIFS.2017.2656460

[137] 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

[138] 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.

[139] 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.

[140] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. 2022. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 965–978. https://doi.org/10.1109/ICDE53745.2022.00077

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

[142] 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 (CVPR)*. 10713–10722.

[143] 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

[144] Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, et al. 2023. StarCoder: may the source be with you! *arXiv preprint arXiv:2305.06161* (2023).

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

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

[147] 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*.

[148] 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

[149] 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

[150] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2020. On the Convergence of FedAvg on Non-IID Data. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*.

[151] Yiming Li, Yang Bai, Yong Jiang, Yong Yang, Shu-Tao Xia, and Bo Li. 2022. Untargeted Backdoor Watermark: Towards Harmless and Stealthy Dataset Copyright Protection. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[152] Zijian Li, Jiawei Shao, Yuyi Mao, Jessie Hui Wang, and Jun Zhang. 2022. Federated learning with GAN-based data synthesis for non-IID clients. In *IJCAI 2022 Workshop on Trustworthy Federated Learning*. Springer, 17–32. https://doi.org/10.1007/978-3-031-28996-5_2

[153] 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 (CVPR)*. 10132–10142.

[154] Xiaoxiao Liang, Yiqun Lin, Huazhu Fu, Lei Zhu, and Xiaomeng Li. 2022. RSCFed: random sampling consensus federated semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10154–10163.

[155] 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.

[156] 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 (COMST)* 22, 3 (2020), 2031–2063. https://doi.org/10.1109/COMST.2020.2986024

[157] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. 2020. Ensemble distillation for robust model fusion in federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33. 2351–2363.

[158] 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

[159] Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. 2021. FedDG: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1013–1023.

[160] 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 (JMLR)* 22, 1 (2021), 10320–10325.

[161] 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).

[162] 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

[163] 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).

[164] 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

[165] 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).

[166] Zhengquan Luo, Yunlong Wang, Zilei Wang, Zhenan Sun, and Tieniu Tan. 2022. FedIris: Towards More Accurate and Privacy-preserving Iris Recognition via Federated Template Communication. In *CVPR 2022 Workshop on Federated Learning for Computer Vision*. 3357–3366.

[167] Lingjuan Lyu, Han Yu, and Qiang Yang. 2020. Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133* (2020).

[168] 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 (FGCS)* 135 (2022), 244–258. https://doi.org/10.1016/j.future.2022.05.003

[169] 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

[170] Richard Maclin, Jude W Shavlik, et al. 1995. Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Vol. 95. 524–531.

[171] 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

[172] Othmane Marfoq, Chuan Xu, Giovanni Neglia, and Richard Vidal. 2020. Throughput-optimal topology design for cross-silo federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33. 19478–19487.

[173] Thomas Margoni. 2018. Artificial Intelligence, Machine learning and EU copyright law: Who owns AI? *Machine Learning and EU Copyright Law: Who Owns AI* (2018). https://doi.org/10.2139/ssrn.3299523

[174] Trent McConaghy. 2022. Ocean Protocol: Tools for the Web3 Data Economy. In *Handbook on Blockchain*. Springer, 505–539.

[175] 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*. 1–9. https://doi.org/10.1145/2948618.2954331

[176] 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.

[177] 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.

[178] Umberto Michieli and Mete Ozay. 2021. Prototype guided federated learning of visual feature representations. *arXiv preprint arXiv:2105.08982* (2021).

[179] 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

[180] 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 (IoT-J)* 8, 16 (2021), 12806–12825. https://doi.org/10.1109/JIOT.2021.3072611

[181] Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. 2023. CodeGen: An open large language model for code with multi-turn program synthesis. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*.

[182] 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 (MobiCom)*. 1–14.

[183] 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)*.

[184] 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.

[185] 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).

[186] 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 35. 27730–27744.

[187] 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.

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

[189] Sen Peng, Yufei Chen, Jie Xu, Zizhuo Chen, Cong Wang, and Xiaohua Jia. 2022. Intellectual property protection of DNN models. *World Wide Web* (2022), 1–35. https://doi.org/10.1007/s11280-022-01113-3

[190] 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

[191] 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

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

[193] 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.

[194] 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

[195] Renjie Pi, Weizhong Zhang, Yueqi Xie, Jiahui Gao, Xiaoyu Wang, Sunghun Kim, and Qifeng Chen. 2022. DYNAFED: Tackling Client Data Heterogeneity with Global Dynamics. *arXiv preprint arXiv:2211.10878* (2022).

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

[197] Zhen Qin, Daoyuan Chen, Bingchen Qian, Bolin Ding, Yaliang Li, and Shuiguang Deng. 2023. Federated Full-Parameter Tuning of Billion-Sized Language Models with Communication Cost under 18 Kilobytes. *arXiv preprint arXiv:2312.06353* (2023).

[198] 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

[199] 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.

[200] 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.

[201] 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 (EMNLP)*. 2383–2392. https://doi.org/10.18653/v1/D16-1264

[202] 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).

[203] 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

[204] 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.

[205] 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.

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

[207] 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

[208] 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

[209] 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*.

[210] 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 (CVPR)*. 10684–10695. https://doi.org/10.1109/CVPR52688.2022.01042

[211] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2015. FitNets: Hints for thin deep nets. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

[212] Lawrence Rosen. 2005. *Open Source Licensing: Software Freedom and Intellectual Property Law*. Prentice Hall Professional Technical Reference, New Jersey.

[213] 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. In *NeurIPS 2022 Workshop on Federated Learning: Recent Advances and New Challenges*.

[214] Max Ryabinin, Eduard Gorbunov, Vsevolod Plokhotnyuk, and Gennady Pekhimenko. 2021. Moshpit SGD: Communication-efficient decentralized training on heterogeneous unreliable devices. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 34. 18195–18211.

[215] Max Ryabinin and Anton Gusev. 2020. Towards crowdsourced training of large neural networks using decentralized mixture-of-experts. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33. 3659–3672.

[216] 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 (TNNLS)* (2021). https://doi.org/10.1109/TNNLS.2021.3129371

[217] 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.

[218] 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

[219] 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).

[220] 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

[221] 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.

[222] 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).

[223] Naichen Shi, Fan Lai, Raed Al Kontar, and Mosharaf Chowdhury. 2023. Fed-ensemble: Ensemble Models in Federated Learning for Improved Generalization and Uncertainty Quantification. *IEEE Transactions on Automation Science and Engineering (T-ASE)* (2023). https://doi.org/10.1109/TASE.2023.3269639

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

[225] 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

[226] Jacob Steeves, Ala Shaabana, Yuqian Hu, Francois Luus, Sin Tai Liu, and Jacqueline Dawn Tasker-Steeves. 2022. Incentivizing Intelligence: The Bittensor Approach. In *NeurIPS 2022 Workshop on Decentralization and Trustworthy Machine Learning in Web3: Methodologies, Platforms, and Applications*.

[227] Shangchao Su, Bin Li, and Xiangyang Xue. 2023. One-shot Federated Learning without server-side training. *Neural Networks* 164 (2023), 203–215. https://doi.org/10.1016/j.neunet.2023.04.035

[228] Dianbo Sui, Yubo Chen, Jun Zhao, Yantao Jia, Yuantao Xie, and Weijian Sun. 2020. FedED: Federated learning via ensemble distillation for medical relation extraction. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*. 2118–2128. https://doi.org/10.18653/v1/2020.emnlp-main.165

[229] 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

[230] Tao Sun, Dongsheng Li, and Bao Wang. 2022. Decentralized federated averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2022). https://doi.org/10.1109/TPAMI.2022.3196503

[231] 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.

[232] Yuchen Sun, Tianpeng Liu, Panhe Hu, Qing Liao, Shouling Ji, Nenghai Yu, Deke Guo, and Li Liu. 2023. Deep Intellectual Property: A Survey. *arXiv preprint arXiv:2304.14613* (2023).

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

[234] 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

[235] 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

[236] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. 2022. FedProto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 8432–8440. https://doi.org/10.1609/aaai.v36i8.20819

[237] Fengyi Tang, Lifan Zeng, Fei Wang, and Jiayu Zhou. 2021. Persona authentication through generative dialogue. *arXiv preprint arXiv:2110.12949* (2021).

[238] 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).

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

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

[241] Chandra Thapa, Pathum Chamikara Mahawaga Arachchige, Seyit Camtepe, and Lichao Sun. 2022. Splitfed: When federated learning meets split learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 36. 8485–8493. https://doi.org/10.1609/aaai.v36i8.20825

[242] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[243] 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

[244] Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. 2017. Embedding watermarks into deep neural networks. In *Proceedings of the 2017 ACM on international conference on multimedia retrieval (ICMR)*. 269–277. https://doi.org/10.1145/3078971.3078974

[245] 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

[246] Jennifer Wortman Vaughan. 2018. Making better use of the crowd: How crowdsourcing can advance machine learning research. *The Journal of Machine Learning Research (JMLR)* 18, 1 (2018), 7026–7071.

[247] Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. 2019. Split learning for health: Distributed deep learning without sharing raw patient data. In *ICLR 2019 Workshop on AI for Social Good*.

[248] 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

[249] Ewen Wang, Boyi Chen, Mosharaf Chowdhury, Ajay Kannan, and Franco Liang. 2023. FLINT: A Platform for Federated Learning Integration. *Proceedings of Machine Learning and Systems (MLSys)* 5 (2023).

[250] 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33. 16070–16084.

[251] 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)*.

[252] 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

[253] 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 (TPAMI)* 44, 6 (2021), 3048–3068. https://doi.org/10.1109/TPAMI.2021.3055564

[254] 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

[255] Naibo Wang, Wenjie Feng, Moming Duan, Fusheng Liu, See-Kiong Ng, et al. 2023. Data-Free Diversity-Based Ensemble Selection for One-Shot Federated Learning. *Transactions on Machine Learning Research* (2023).

[256] Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. 2021. Training networks in null space of feature covariance for continual learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)*. 184–193.

[257] 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.

[258] 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

[259] 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

[260] John Wickström, Magnus Westerlund, and Emmanuel Raj. 2022. Decentralizing Machine Learning Operations using Web3 for IoT Platforms. In *Proceedings of the 13th IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*. IEEE, 238–245. https://doi.org/10.1109/CloudCom55334.2022.00041

[261] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data* 3, 1 (2016), 1–9. https://doi.org/10.1038/sdata.2016.18

[262] Herbert Woisetschläger, Alexander Isenko, Shiqiang Wang, Ruben Mayer, and Hans-Arno Jacobsen. 2023. Federated Fine-Tuning of LLMs on the Very Edge: The Good, the Bad, the Ugly. *arXiv preprint arXiv:2310.03150* (2023).

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

[264] Sebastien C Wong, Adam Gatt, Victor Stamatescu, and Mark D McDonnell. 2016. Understanding data augmentation for classification: when to warp?. In *Proceedings of the 2016 international conference on digital image computing: techniques and applications (DICTA)*. IEEE, 1–6.

[265] 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

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

[267] 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).

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

[269] 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

[270] Mingzhao Yang, Shangchao Su, Bin Li, and Xiangyang Xue. 2023. Exploring One-shot Semi-supervised Federated Learning with A Pre-trained Diffusion Model. *arXiv preprint arXiv:2305.04063* (2023).
[271] Qiang Yang, Lixin Fan, Richard Tong, and Angelica Lv. 2021. IEEE Federated Machine Learning. *IEEE Federated Machine Learning - White Paper* (2021), 1–18.
[272] 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
[273] 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).
[274] 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
[275] 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
[276] Tehrim Yoon, Sumin Shin, Sung Ju Hwang, and Eunho Yang. 2021. FedMix: Approximation of Mixup under Mean Augmented Federated Learning. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*.
[277] 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
[278] 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
[279] 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 (AIES)*. 393–399. https://doi.org/10.1145/3375627.3375840
[280] Shuyang Yu, Junyuan Hong, Haotao Wang, Zhangyang Wang, and Jiayu Zhou. 2023. Turning the curse of heterogeneity in federated learning into a blessing for out-of-distribution detection. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*.
[281] Binhang Yuan, Yongjun He, Jared Davis, Tianyi Zhang, Tri Dao, Beidi Chen, Percy S Liang, Christopher Re, and Ce Zhang. 2022. Decentralized training of foundation models in heterogeneous environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 35. 25464–25477.
[282] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, and Nghia Hoang. 2019. Statistical model aggregation via parameter matching. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 32.
[283] 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.
[284] 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
[285] Dun Zeng, Siqi Liang, Xiangjing Hu, Hui Wang, and Zenglin Xu. 2021. Fedlab: A flexible federated learning framework. *arXiv preprint arXiv:2107.11621* (2021).
[286] Yufeng Zhan, Jie Zhang, Zicong Hong, Leijie Wu, Peng Li, and Song Guo. 2021. A survey of incentive mechanism design for federated learning. *IEEE Transactions on Emerging Topics in Computing (TETC)* 10, 2 (2021), 1035–1044. https://doi.org/10.1109/TETC.2021.3063517
[287] 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
[288] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2018. mixup: Beyond Empirical Risk Minimization. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.
[289] 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)*.
[290] Jie Zhang, Dongdong Chen, Jing Liao, Weiming Zhang, Gang Hua, and Nenghai Yu. 2020. Passport-aware normalization for deep model protection. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33. 22619–22628.
[291] Jie Zhang, Song Guo, Xiaosong Ma, Haozhao Wang, Wenchao Xu, and Feijie Wu. 2021. Parameterized knowledge transfer for personalized federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 34. 10092–10104.
[292] Jiayun Zhang, Xiyuan Zhang, Xinyang Zhang, Dezhi Hong, Rajesh K Gupta, and Jingbo Shang. 2023. Federated Learning with Client-Exclusive Classes. *arXiv preprint arXiv:2301.00489* (2023).
[293] Ke Zhang, Carl Yang, Xiaoxiao Li, Lichao Sun, and Siu Ming Yiu. 2021. Subgraph federated learning with missing neighbor generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 34. 6671–6682.

[294] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 3836–3847.

[295] Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. 2022. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10174–10183.

[296] Lan Zhang, Dapeng Wu, and Xiaoyong Yuan. 2022. FedZKT: Zero-shot knowledge transfer towards resource-constrained federated learning with heterogeneous on-device models. In *Proceedings of the IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 928–938. https://doi.org/10.1109/ICDCS54860.2022.00094

[297] 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

[298] 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).

[299] 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 (IoTM)* 5, 1 (2022), 24–29. https://doi.org/10.1109/IOTM.004.2100182

[300] 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 (TPDS)* 32, 10 (2021), 2524–2540. https://doi.org/10.1109/TPDS.2021.3068195

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

[302] 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

[303] 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

[304] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 32. Curran Associates, Inc.

[305] 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.

[306] 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

[307] 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 (CIT)*. IEEE, 1125–1130. https://doi.org/10.1109/CIT.2012.228