Appendix

This appendix supplements the paper titled **Towards Open Federated Learning Platforms: Survey and Vision from Technical and Legal Perspectives**. Due to the page limitations of the journal publication, we have condensed the original manuscript and relocated some content to here. This appendix is organized as follows:

- In ‡I, we present the related concepts of this survey and point out their differences.
- ‡II lists our recommended filter conditions for query-based FL, along with the reasons behind each recommendation.
- In ‡III, we provide our preferences for choosing licenses in the context of model reusing, and we discuss copyright concerns related to reused results.
- In ‡IV, we present our findings and discuss the relationship between model reusing and model licenses.
- ‡V further discusses the model protection challenge, which plays an important role in advocating open FL platforms.
- ‡VI, we briefly discuss some parallel works from Web3-based field and explore the opportunities for decentralization and monetization in contract-based FL.

I. Related Topics

As summarized in TABLE I, most surveys extensively discuss the challenges of efficiency, heterogeneity, privacy in FL systems design, while the surveys from blockchain fields offer 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 delegates it to data holders to complete. This *server-dominated* cooperation framework is a narrow implementation of the FL systems. Therefore, this survey aims 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.

A. 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 [31] and query suggestion [32] for Gboard applications. Subsequently, many novel FL systems have emerged to adapt to diverse federated training scenarios, such as Horizontal FL (e.g., TFF [33], FedLab [34], Felicitas [35], IBM FL [36], OpenFL [37]), Vertical FL [38] or both (e.g., FATE [39], FedML [40], PaddleFL [41], Flower [42], FedTree [43], NVFLARE [44]). 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 which provides federated training services to the public.

Unlike the FL systems mentioned above, PySyft [45] 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 [46]. 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 [47]. In such FL platforms, every user is free to collaborate on the training of machine learning models while privacy is protected.

B. As-a-Service Business Model

In the current context of Software-as-a-Service (SaaS) [48], 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) [49]–[53] and Machine-Learning-as-a-Service (MLaaS) [54]–[58] encapsulate model execution and model development as services. The original concept of MaaS [49], [50] 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 [51] and deep learning based systems [52], [53]. 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 [59]. 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

[1] Announcing Proof-of-concept Support for TFF in Pysyft 0.7

TABLE I
SUMMARY OF EXISTING FL SURVEYS. SYS DENOTES FL SYSTEMS DESIGN, APP DENOTES FL APPLICATIONS, SDC DENOTES SERVER-DOMINATED COOPERATION FRAMEWORKS. ✓: ELABORATED, ✗: NOT ELABORATED.

		Challenges					Contents		
Scenarios/Tasks	FL Surveys	Efficiency	Heterogeneity	Privacy	Incentive	Decentralize	SYS	APP	SDC
General	Yang *et al.* [1]	✓	✓	✓	✓	✓	✓	✓	✓
	Li *et al.* 2020 [2]	✓	✓	✓	✗	✓	✓	✓	✓
	Zhang *et al.* 2021 [3]	✓	✓	✓	✗	✗	✓	✓	✓
	Gupta *et al.* [4]	✓	✓	✓	✗	✓	✓	✓	✓
	Xu *et al.* [5]	✓	✓	✓	✗	✓	✓	✓	✓
	Li *et al.* 2021 [6]	✓	✓	✓	✓	✓	✓	✓	✓
	El *et al.* [7]	✗	✗	✓	✗	✓	✓	✗	✓
	Kulkarni *et al.* [8]	✓	✓	✗	✗	✗	✓	✗	✓
	Liu *et al.* [9]	✓	✗	✓	✗	✓	✓	✗	✓
	Tan *et al.* [10]	✗	✓	✗	✗	✗	✓	✗	✓
	Zhu *et al.* 2021 [11]	✗	✓	✗	✗	✗	✓	✗	✓
	Ma *et al.* [12]	✓	✓	✓	✗	✗	✓	✗	✓
	Aledhari *et al.* [13]	✓	✓	✗	✗	✗	✓	✓	✓
	Kairouz *et al.* [14]	✓	✓	✓	✓	✓	✓	✓	✓
	AbdulRahman *et al.* [15]	✓	✓	✓	✓	✗	✓	✓	✓
	Lim *et al.* [16]	✓	✓	✓	✓	✗	✓	✓	✓
Healthcare	Xu *et al.* [17]	✓	✓	✓	✗	✗	✓	✓	✓
	Pfitzner *et al.* [18]	✓	✓	✓	✗	✗	✓	✓	✓
	Antunes *et al.* [19]	✗	✓	✓	✗	✗	✗	✓	✓
	Rieke *et al.* [20]	✗	✓	✓	✗	✓	✓	✓	✓
IoT	Zhang *et al.* 2022 [21]	✓	✓	✗	✗	✗	✓	✓	✓
	Boopalan *et al.* [22]	✓	✓	✓	✓	✓	✓	✓	✓
	Ramu *et al.* [23]	✓	✓	✓	✗	✓	✓	✓	✓
	Du *et al.* [24]	✓	✓	✓	✓	✓	✓	✓	✓
Cybersecurity	Agrawal *et al.* [25]	✓	✓	✓	✗	✓	✓	✓	✓
	Alazab *et al.* [26]	✗	✗	✓	✗	✗	✓	✓	✓
	Ghimire *et al.* [27]	✓	✗	✓	✗	✗	✓	✓	✓
Blockchain	Nguyen *et al.* [28]	✓	✓	✓	✓	✓	✓	✓	✓
	Qu *et al.* [29]	✓	✓	✓	✓	✓	✓	✓	✓
	Zhu *et al.* 2022 [30]	✓	✓	✓	✓	✓	✓	✓	✓

reduce the required computational resources. MLaaS enables users to upload their data for training [54], [56], [60] or inference [55], 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 training and inference 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) [55], [61] and Homomorphic Encryption [56], [62], it is worth noting that our focus is not solely on privacy.

Recently, Kourtellis *et al.* [58] 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.* [63] propose an open FL ecosystem for mobile devices, which shares a similar concept to FLaaS. However, those approach also follow the traditional server-dominated cooperation framework, which falls under the scope of previous FL surveys [1], [2], [14].

C. Dcentralized FL

Decentralized FL [64]–[69], 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.* [64] 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 [68] 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 [65] improves the privacy of neighbor-wide model sharing by sharing a proxy model through knowledge distillation [70]. 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.* [66] proposed improving the efficiency of model sharing by selecting a connected sub-graph. Another approach is the use of a gossip-based approach, where model parameters or segments of model parameters are randomly shared with peer neighbors [67], [69], [71]. Despite the advantages brought by the decentralized design, the training procedure of these frameworks also follows a pre-set learning task and lacks sustainable cooperation, resulting in a non-public and low reusability FL platform similar to

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

centralized FL. In fact, our vision for open FL platforms is to extend the FAIR principles [72] 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.

II. Filter Conditions for Query-based FL

A. Data Description

Similar with the data heterogeous challenges in FL [73], 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 [74]. Besides, label errors are pervasive even in open datasets [75]. 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 used to profile the data distribution [76] and complementary provenance information.

B. 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 [77], such as MLflow[5] and Neptune[6], 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[7] can be leveraged to generate a report that displays the FLOPS and bandwidth requirements.

C. 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 [78]. In some cases, contributed models may rely on other models as dependencies. For example, Fast R-CNN [79] uses VGG16 [80] as its backbone. It is crucial to release this information for further model license compatibility analysis.

D. Fairness and Robustness

To enhance trust in AI services, IBM has introduced FactSheets [81], which promote the use of supplier-provided documents detailing potential bias, proof of robustness, optimal conditions, etc., associated with AI services. This information is useful for understanding the strengths and weaknesses of models, enabling further improvement through collaborative learning. We highly recommend readers visit the FactSheets website and explore their examples for additional reading[8].

III. License Choosing Preferences

A. 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 [82], which was trained on data from the Case Law Access Project[9] (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.

B. 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

[5] https://mlflow.org

[6] https://neptune.ai

[7] https://docs.nvidia.com/deeplearning/frameworks/dlprof-user-guide

[8] https://aifs360.res.ibm.com/

[9] https://case.law

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).

C. Preferences for Models

Apache-2.0, AFL-3.0, Artistic-2.0, ECL-2.0 > OpenRAILs

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 §IV-C TABLE II, 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 [83], [84]. 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, traditional permissive licenses, which follow worse-is-better design philosophy [85], 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 following.

D. 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* [86]. 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 §IV-C TABLE II 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 [87].

The determination of copyrightability and authorship of computer-generated content is an open issue that needs to be addressed through corresponding legislation [86]–[88]. European Parliament regarded that *consideration must first be given to assessing patent law in the light of the development of AI*[10]. 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 [87]. Licensors can also make efforts to clarify this issue by including relevant claims in their licenses. For example, the license of Stable Diffusion [89] explicitly states that *Licensor claims no rights in the Output You generate using the Model*. Similarly, ChatGPT[11], even though it is a proprietary software of OpenAI company, its sharing & publication policy[12] 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 [90] and SEER [91] 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.

IV. Advanced Model Reusing

A. Hybrid Model Reusing in FL

Following the taxonomy we introduced, 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. 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 [92]–[94], using solely amalgamation often lead to weight divergence [10], [95], 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 §V-E TABLE V). 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 [96], which uses labeled validation set, ref. §V-C TABLE III). Getting rid of the limitation of amalgamation, these distillation studies (e.g., FedAD [97], FedKD$_1$ [98]) 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 [99] and FRD [100]) as distillation by directly generating synthetic data for fusion. This one-shot capacity can also be extended to the hybrid case, such as FedCAV [101], FedBE [102], FedAUX [103], DENSE [104]. 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 [100], 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 §V-E TABLE V.

> *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,

[10] REPORT on Intellectual Property Rights for the Development of Artificial Intelligence Technologies

[11] https://openai.com/blog/chatgpt

[12] https://openai.com/policies/sharing-publication-policy

we can refer to §V-E TABLE V and consider the following factors:

- **Data Dependency**. Recalling that query-based FL is contactless, 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 [105], DynaFed [106], etc.), distillation (FedFusion [107], MOON [108], etc.), generation (SemiFL [109], NeighGen [110], etc.). So, refer to §V-E TABLE V, 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 [99], FRD [100]). 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 §V-E TABLE V in red and orange, respectively. Even though the KD procedure of DENSE [104] 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 [111] (only combination) and Models Fusion [94], [112], [113] (only amalgamation), can be directly applied in a query-based FL setting. However, to avoid redundancy, these studies have been omitted from §V-E TABLE V. 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.

B. 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 II. The analysis covers the most popular model licenses (used by over 100 released models) on Hugging Face, which includes 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 that is not governed by the original license. Additionally, we can refer to TABLE II 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 II for the corresponding delineations of each model reuse mechanism instead of relying solely on the technical name.
- **Restrictions of combination and amalgamation**. As shown in TABLE II, 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. §IV-C Fig. 5(b) and §IV-C TABLE II) for batch model combination and amalgamation.
- **User behavioral restrictions of responsible AI licenses**. OpenRAIL licenses [114] 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 [84] (ref. ‡III). To prevent license proliferation [115], 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 [104], 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 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 permis-

TABLE II
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	✗	✗

sive licenses can make things easier and facilitate the model reusability of open FL platforms.

V. How to Protect Models

In general, there are three kinds of mechanisms that can be employed to protect DNN models: authorization [55], watermarking [116], and fingerprinting [117]. 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 [118], [119]. 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 [120] or trigger data [121] acting as a backdoor [122]. On the other hand, fingerprinting strategies aim to generate an identifier for each model based on snapshots of training history [123] or inference results that approximate decision boundaries [124]. 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 [125], [126]. 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 [127]. 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 [124]. Meanwhile, these backdoors can be exploited by attackers to manipulate the model's predictions [128]. 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 [119] 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. 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 [86]. 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 [117], required bit-length, and deployment cost [120], [121] of model protection solutions such

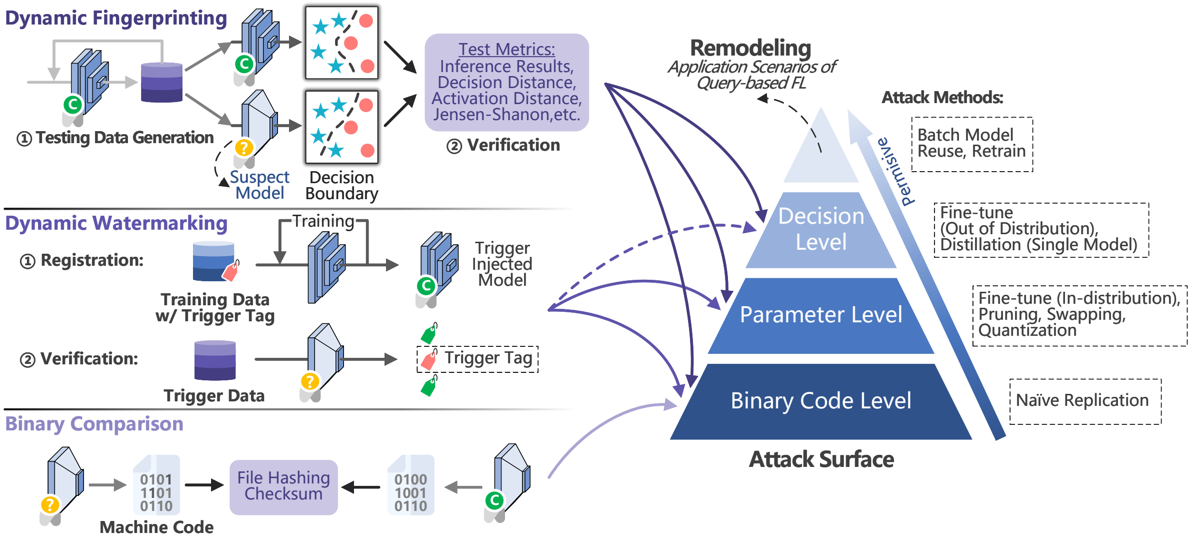

Fig. 1. Overview of DNNs IP protection methods.

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.* [116] 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 [117] 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 [129] 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 [125], [126], **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 [124], 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 [130], 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. 1.

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.

VI. Decentralization and Monetization in Contract-based FL

Parallel to contract-based FL, several efforts are underway to build decentralized AI platforms through Web3-based techniques [131]. For instance, Blythman *et al.* proposed decentralized AI Hubs [132], empowered by DAOs (e.g., IPFS [133], Ocean [134], OpenMined), to run ML activities in a trustless setting. Wickström *et al.* [135] propose the concept of AI market which builds a ML analysis network for IoT devices based on Ethereum. Similar to DMoE, Bittensor [136] 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 [137], 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. ‡V). 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.

References

[1] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[2] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine (SPM)*, vol. 37, no. 3, pp. 50–60, 2020.

[3] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems (KBS)*, vol. 216, p. 106775, 2021.

[4] R. Gupta and T. Alam, "Survey on federated-learning approaches in distributed environment," *Wireless Personal Communications*, vol. 125, no. 2, pp. 1631–1652, 2022.

[5] C. Xu, Y. Qu, Y. Xiang, and L. Gao, "Asynchronous federated learning on heterogeneous devices: A survey," *Computer Science Review*, vol. 50, p. 100595, 2023.

[6] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: vision, hype and reality for data privacy and protection," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2021.

[7] A. El Ouadrhiri and A. Abdelhadi, "Differential privacy for deep and federated learning: A survey," *IEEE Access*, vol. 10, pp. 22 359–22 380, 2022.

[8] V. Kulkarni, M. Kulkarni, and A. Pant, "Survey of personalization techniques for federated learning," in *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*. IEEE, 2020, pp. 794–797.

[9] Z. Liu, J. Guo, W. Yang, J. Fan, K.-Y. Lam, and J. Zhao, "Privacy-preserving aggregation in federated learning: A survey," *IEEE Transactions on Big Data (TBD)*, pp. 1–20, 2022.

[10] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, pp. 1–17, 2022.

[11] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-iid data: A survey," *Neurocomputing*, vol. 465, pp. 371–390, 2021.

[12] X. Ma, J. Zhu, Z. Lin, S. Chen, and Y. Qin, "A state-of-the-art survey on solving non-iid data in federated learning," *Future Generation Computer Systems (FGCS)*, vol. 135, pp. 244–258, 2022.

[13] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications," *IEEE Access*, vol. 8, pp. 140 699–140 725, 2020.

[14] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[15] S. AbdulRahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond," *IEEE Internet of Things Journal (IoT-J)*, vol. 8, no. 7, pp. 5476–5497, 2020.

[16] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials (COMST)*, vol. 22, no. 3, pp. 2031–2063, 2020.

[17] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning for healthcare informatics," *Journal of Healthcare Informatics Research*, vol. 5, pp. 1–19, 2021.

[18] B. Pfitzner, N. Steckhan, and B. Arnrich, "Federated learning in a medical context: A systematic literature review," *ACM Transactions on Internet Technology (TOIT)*, vol. 21, no. 2, pp. 1–31, 2021.

[19] R. S. Antunes, C. André da Costa, A. Küderle, I. A. Yari, and B. Eskofier, "Federated learning for healthcare: Systematic review and architecture proposal," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 4, pp. 1–23, 2022.

[20] N. Rieke, J. Hancox, W. Li, F. Milletari, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein *et al.*, "The future of digital health with federated learning," *NPJ digital medicine*, vol. 3, no. 1, p. 119, 2020.

[21] T. Zhang, L. Gao, C. He, M. Zhang, B. Krishnamachari, and A. S. Avestimehr, "Federated learning for the internet of things: applications, challenges, and opportunities," *IEEE Internet of Things Magazine (IoTM)*, vol. 5, no. 1, pp. 24–29, 2022.

[22] P. Boopalan, S. P. Ramu, Q.-V. Pham, K. Dev, P. K. R. Maddikunta, T. R. Gadekallu, T. Huynh-The *et al.*, "Fusion of federated learning and industrial internet of things: A survey," *Computer Networks*, p. 109048, 2022.

[23] S. P. Ramu, P. Boopalan, Q.-V. Pham, P. K. R. Maddikunta, T. Huynh-The, M. Alazab, T. T. Nguyen, and T. R. Gadekallu, "Federated learning enabled digital twins for smart cities: Concepts, recent advances, and future directions," *Sustainable Cities and Society*, vol. 79, p. 103663, 2022.

[24] Z. Du, C. Wu, T. Yoshinaga, K.-L. A. Yau, Y. Ji, and J. Li, "Federated learning for vehicular internet of things: Recent advances and open issues," *IEEE Open Journal of the Computer Society (OJ-CS)*, vol. 1, pp. 45–61, 2020.

[25] S. Agrawal, S. Sarkar, O. Aouedi, G. Yenduri, K. Piamrat, M. Alazab, S. Bhattacharya, P. K. R. Maddikunta, and T. R. Gadekallu, "Federated learning for intrusion detection system: Concepts, challenges and future directions," *Computer Communications*, 2022.

[26] M. Alazab, S. P. RM, M. Parimala, P. K. R. Maddikunta, T. R. Gadekallu, and Q.-V. Pham, "Federated learning for cybersecurity: concepts, challenges, and future directions," *IEEE Transactions on Industrial Informatics (TII)*, vol. 18, no. 5, pp. 3501–3509, 2021.

[27] B. Ghimire and D. B. Rawat, "Recent advances on federated learning for cybersecurity and cybersecurity for federated learning for internet of things," *IEEE Internet of Things Journal (IoT-J)*, 2022.

[28] D. C. Nguyen, M. Ding, Q.-V. Pham, P. N. Pathirana, L. B. Le, A. Seneviratne, J. Li, D. Niyato, and H. V. Poor, "Federated learning

meets blockchain in edge computing: Opportunities and challenges," *IEEE Internet of Things Journal (IoT-J)*, vol. 8, no. 16, pp. 12 806–12 825, 2021.

[29] Y. Qu, M. P. Uddin, C. Gan, Y. Xiang, L. Gao, and J. Yearwood, "Blockchain-enabled federated learning: A survey," *ACM Computing Surveys (CSUR)*, vol. 55, no. 4, pp. 1–35, 2022.

[30] J. Zhu, J. Cao, D. Saxena, S. Jiang, and H. Ferradi, "Blockchain-empowered federated learning: Challenges, solutions, and future directions," *ACM Computing Surveys (CSUR)*, 2022.

[31] A. Hard, K. Rao, R. Mathews, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[32] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, and F. Beaufays, "Applied federated learning: Improving google keyboard query suggestions," *arXiv preprint arXiv:1812.02903*, 2018.

[33] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2016, pp. 265–283.

[34] D. Zeng, S. Liang, X. Hu, H. Wang, and Z. Xu, "FedLab: A flexible federated learning framework," *Journal of Machine Learning Research*, vol. 24, no. 100, pp. 1–7, 2023.

[35] Q. Zhang, T. Wu, P. Zhou, S. Zhou, Y. Yang, and X. Jin, "Felicitas: Federated learning in distributed cross device collaborative frameworks," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 4502–4509.

[36] H. Ludwig, N. Baracaldo, G. Thomas, Y. Zhou, A. Anwar, S. Rajamoni, Y. Ong, J. Radhakrishnan, A. Verma, M. Sinn *et al.*, "Ibm federated learning: an enterprise framework white paper v0. 1," *arXiv preprint arXiv:2007.10987*, 2020.

[37] P. Foley, M. J. Sheller, B. Edwards, S. Pati, W. Riviera, M. Sharma, P. N. Moorthy, S.-h. Wang, J. Martin, P. Mirhaji *et al.*, "OpenFL: the open federated learning library," *Physics in Medicine & Biology*, vol. 67, no. 21, p. 214001, 2022.

[38] Z. Wu, Q. Li, and B. He, "Practical vertical federated learning with unsupervised representation learning," *IEEE Transactions on Big Data (TBD)*, 2022.

[39] Y. Liu, T. Fan, T. Chen, Q. Xu, and Q. Yang, "FATE: An industrial grade platform for collaborative learning with data protection," *The Journal of Machine Learning Research (JMLR)*, vol. 22, no. 1, pp. 10 320–10 325, 2021.

[40] C. He, S. Li, J. So, X. Zeng, M. Zhang, H. Wang, X. Wang, P. Vepakomma, A. Singh, H. Qiu *et al.*, "FedML: A research library and benchmark for federated machine learning," in *NeurIPS 2020 Workshop on Scalability, Privacy, and Security in Federated Learning*, 2020.

[41] Y. Ma, D. Yu, T. Wu, and H. Wang, "Paddlepaddle: An open-source deep learning platform from industrial practice," *Frontiers of Data and Domputing*, vol. 1, no. 1, pp. 105–115, 2019.

[42] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, P. P. de Gusmão, and N. D. Lane, "Flower: A friendly federated learning research framework," *arXiv preprint arXiv:2007.14390*, 2020.

[43] Q. Li, Z. Wu, Y. Cai, C. M. Yung, T. Fu, B. He *et al.*, "FedTree: A federated learning system for trees," in *Proceedings of Machine Learning and Systems (MLSys)*, 2023.

[44] H. R. Roth, Y. Cheng, Y. Wen, I. Yang, Z. Xu, Y.-T. Hsieh, K. Kersten, A. Harouni, C. Zhao, K. Lu *et al.*, "Nvidia flare: Federated learning from simulation to real-world," in *NeurIPS 2022 Workshop on Federated Learning: Recent Advances and New Challenges*, 2022.

[45] A. Ziller, A. Trask, A. Lopardo, B. Szymkow, B. Wagner, E. Bluemke, J.-M. Nounahon, J. Passerat-Palmbach, K. Prakash, N. Rose *et al.*, "PySyft: A library for easy federated learning," *Federated Learning Systems: Towards Next-Generation AI*, pp. 111–139, 2021.

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

[47] Y. Lou, L.-Y. Duan, Y. Luo, Z. Chen, T. Liu, S. Wang, and W. Gao, "Towards efficient front-end visual sensing for digital retina: A model-centric paradigm," *IEEE Transactions on Multimedia*, vol. 22, no. 11, pp. 3002–3013, 2020.

[48] P. Brereton, D. Budgen, K. Bennnett, M. Munro, P. Layzell, L. MacCaulay, D. Griffiths, and C. Stannett, "The future of software," *Communications of the ACM*, vol. 42, no. 12, pp. 78–84, 1999.

[49] G. N. Geller and W. Turner, "The model web: a concept for ecological forecasting," in *2007 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2007, pp. 2469–2472.

[50] D. Roman, S. Schade, A.-J. Berre, N. R. Bodsberg, and J. Langlois, "Model as a service (maas)," in *AGILE Workshop-Grid Technologies for Geospatial Applications*, 01 2009.

[51] G. Zou, B. Zhang, J. Zheng, Y. Li, and J. Ma, "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, 2012, pp. 1125–1130.

[52] H. Liu, Q. Gao, J. Li, X. Liao, H. Xiong, G. Chen, W. Wang, G. Yang, Z. Zha, D. Dong *et al.*, "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*. New York, NY, USA: Association for Computing Machinery, 2021, pp. 3289–3298.

[53] T. Sun, Y. Shao, H. Qian, X. Huang, and X. Qiu, "Black-box tuning for language-model-as-a-service," in *Proceedings of the 39th International Conference on Machine Learning (ICML)*. PMLR, 2022, pp. 20 841–20 855.

[54] M. Ribeiro, K. Grolinger, and M. A. Capretz, "MLaaS: Machine learning as a service," in *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*. IEEE, 2015, pp. 896–902.

[55] L. Hanzlik, Y. Zhang, K. Grosse, A. Salem, M. Augustin, M. Backes, and M. Fritz, "MLCapsule: Guarded offline deployment of machine learning as a service," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2021, pp. 3300–3309.

[56] E. Hesamifard, H. Takabi, M. Ghasemi, and R. N. Wright, "Privacy-preserving machine learning as a service," *Proceedings on Privacy Enhancing Technologies*, vol. 2018, no. 3, pp. 123–142, 2018.

[57] L. E. Li, E. Chen, J. Hermann, P. Zhang, and L. Wang, "Scaling machine learning as a service," in *International Conference on Predictive Applications and APIs*. PMLR, 2017, pp. 14–29.

[58] N. Kourtellis, K. Katevas, and D. Perino, "FLaaS: Federated learning as a service," in *Proceedings of the 1st workshop on distributed machine learning*, 2020, pp. 7–13.

[59] X. Han, Z. Zhang, N. Ding, Y. Gu, X. Liu, Y. Huo, J. Qiu, Y. Yao, A. Zhang, L. Zhang *et al.*, "Pre-trained models: Past, present and future," *AI Open*, vol. 2, pp. 225–250, 2021.

[60] L. Zhao, Q. Wang, C. Wang, Q. Li, C. Shen, and B. Feng, "VeriML: Enabling integrity assurances and fair payments for machine learning as a service," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 32, no. 10, pp. 2524–2540, 2021.

[61] F. McKeen, I. Alexandrovich, I. Anati, D. Caspi, S. Johnson, R. Leslie-Hurd, and C. Rozas, "Intel® software guard extensions (Intel® SGX) support for dynamic memory management inside an enclave," in *Proceedings of the Hardware and Architectural Support for Security and Privacy*, 2016, pp. 1–9.

[62] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing (STOC)*, 2009, pp. 169–178.

[63] X. Jiang, H. Hu, T. On, P. Lai, V. D. Mayyuri, A. Chen, D. M. Shila, A. Larmuseau, R. Jin, C. Borcea *et al.*, "Flsys: Toward an open ecosystem for federated learning mobile apps," *IEEE Transactions on Mobile Computing (TMC)*, 2022.

[64] A. Lalitha, S. Shekhar, T. Javidi, and F. Koushanfar, "Fully decentralized federated learning," in *NeurIPS 2018 Workshop on Bayesian Deep Learning*, 2018.

[65] S. Kalra, J. Wen, J. C. Cresswell, M. Volkovs, and H. Tizhoosh, "Decentralized federated learning through proxy model sharing," *Nature Communications*, vol. 14, no. 1, p. 2899, 2023.

[66] O. Marfoq, C. Xu, G. Neglia, and R. Vidal, "Throughput-optimal topology design for cross-silo federated learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 19 478–19 487.

[67] C. Hu, J. Jiang, and Z. Wang, "Decentralized federated learning: A segmented gossip approach," in *IJCAI 2019 Workshop on Federated Learning for User Privacy and Data Confidentiality*, 2019.

[68] T. Sun, D. Li, and B. Wang, "Decentralized federated averaging," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2022.

[69] Y. Shi, L. Shen, K. Wei, Y. Sun, B. Yuan, X. Wang, and D. Tao, "Improving the model consistency of decentralized federated learning," in *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023, pp. 31 269–31 291.

[70] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2014.

[71] I. Hegedűs, G. Danner, and M. Jelasity, "Decentralized learning works: An empirical comparison of gossip learning and federated learning," *Journal of Parallel and Distributed Computing (JPDC)*, vol. 148, pp. 109–124, 2021.

[72] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne *et al.*, "The fair guiding principles for scientific data management and stewardship," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[73] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-iid data silos: An experimental study," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2022, pp. 965–978.

[74] I. Dayan, H. R. Roth, A. Zhong, A. Harouni, A. Gentili, A. Z. Abidin, A. Liu, A. B. Costa, B. J. Wood, C.-S. Tsai *et al.*, "Federated learning for predicting clinical outcomes in patients with covid-19," *Nature medicine*, vol. 27, no. 10, pp. 1735–1743, 2021.

[75] C. G. Northcutt, A. Athalye, and J. Mueller, "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)*, 2021.

[76] X. Li, Y. Gu, N. Dvornek, L. H. Staib, P. Ventola, and J. S. Duncan, "Multi-site fmri analysis using privacy-preserving federated learning and domain adaptation: Abide results," *Medical Image Analysis*, vol. 65, p. 101765, 2020.

[77] M. Vartak, H. Subramanyam, W.-E. Lee, S. Viswanathan, S. Husnoo, S. Madden, and M. Zaharia, "Modeldb: a system for machine learning model management," in *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, 2016, pp. 1–3.

[78] R. David, J. Duke, A. Jain, V. Janapa Reddi, N. Jeffries, J. Li, N. Kreeger, I. Nappier, M. Natraj, T. Wang *et al.*, "Tensorflow lite micro: Embedded machine learning for tinyml systems," *Proceedings of Machine Learning and Systems (MLSys)*, vol. 3, pp. 800–811, 2021.

[79] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE international Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[80] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[81] M. Arnold, R. K. Bellamy, M. Hind, S. Houde, S. Mehta, A. Mojsilović, R. Nair, K. N. Ramamurthy, A. Olteanu, D. Piorkowski *et al.*, "FactSheets: Increasing trust in ai services through supplier's declarations of conformity," *IBM Journal of Research and Development*, vol. 63, no. 4/5, pp. 6–1, 2019.

[82] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The muppets straight out of law school," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 2898–2904.

[83] B. Perens, "The open source definition," *Open sources: voices from the open source revolution*, vol. 1, pp. 171–188, 1999.

[84] E. Greenbaum, "The non-discrimination principle in open source licensing," *Cardozo Law Review*, vol. 37, no. 4, pp. 1297–1344, 2016.

[85] R. Gabriel, "The rise of "worse is better"," *Lisp: Good News, Bad News, How to Win Big*, vol. 2, no. 5, 1991.

[86] N. C. on New Technological Uses of Copyrighted Works (US), "Final report of the national commission on new technological uses of copyrighted works, july 31, 1978." Library of Congress, 1979.

[87] S. F. Hedrick, "I think, therefore i create: Claiming copyright in the outputs of algorithms," *New York University Journal of Intellectual Property & Entertainment Law (JIPEL)*, vol. 8, no. 2, pp. 324–375, 2019.

[88] T. Margoni, "Artificial intelligence, machine learning and eu copyright law: Who owns ai?" *Machine Learning and EU Copyright Law: Who Owns AI*, 2018.

[89] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10 684–10 695.

[90] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin *et al.*, "OPT: Open pre-trained transformer language models," *arXiv preprint arXiv:2205.01068*, 2022.

[91] P. Goyal, Q. Duval, I. Seessel, M. Caron, M. Singh, I. Misra, L. Sagun, A. Joulin, and P. Bojanowski, "Vision models are more robust and fair when pretrained on uncurated images without supervision," *arXiv preprint arXiv:2202.08360*, 2022.

[92] F. Yu, W. Zhang, Z. Qin, Z. Xu, D. Wang, C. Liu, Z. Tian, and X. Chen, "Fed2: Feature-aligned federated learning," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2066–2074.

[93] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," in *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020.

[94] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, and N. Hoang, "Statistical model aggregation via parameter matching," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[95] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020.

[96] D. Sui, Y. Chen, J. Zhao, Y. Jia, Y. Xie, and W. Sun, "FedED: Federated learning via ensemble distillation for medical relation extraction," in *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, 2020, pp. 2118–2128.

[97] X. Gong, A. Sharma, S. Karanam, Z. Wu, T. Chen, D. Doermann, and A. Innanje, "Ensemble attention distillation for privacy-preserving federated learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 15 076–15 086.

[98] ——, "Preserving privacy in federated learning with ensemble cross-domain knowledge distillation," in *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, vol. 36, no. 11, 2022, pp. 11 891–11 899.

[99] M. Yang, S. Su, B. Li, and X. Xue, "Exploring one-shot semi-supervised federated learning with a pre-trained diffusion model," *arXiv preprint arXiv:2305.04063*, 2023.

[100] H. Cha, J. Park, H. Kim, S.-L. Kim, and M. Bennis, "Federated reinforcement distillation with proxy experience memory," in *IJCAI 2019 Workshop on Federated Learning for User Privacy and Data Confidentiality*, 2019.

[101] C. E. Heinbaugh, E. Luz-Ricca, and H. Shao, "Data-free one-shot federated learning under very high statistical heterogeneity," in *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023.

[102] H. Chen and W. Chao, "FedBE: Making bayesian model ensemble applicable to federated learning," in *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.

[103] F. Sattler, T. Korjakow, R. Rischke, and W. Samek, "FedAUX: Leveraging unlabeled auxiliary data in federated learning," *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2021.

[104] J. Zhang, C. Chen, B. Li, L. Lyu, S. Wu, S. Ding, C. Shen, and C. Wu, "DENSE: Data-free one-shot federated learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[105] D. Li and J. Wang, "FedMD: Heterogenous federated learning via model distillation," in *NeurIPS 2019 Workshop on Federated Learning for User Privacy and Data Confidentiality*, 2019.

[106] R. Pi, W. Zhang, Y. Xie, J. Gao, X. Wang, S. Kim, and Q. Chen, "DynaFed: Tackling client data heterogeneity with global dynamics," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 12 177–12 186.

[107] X. Yao, T. Huang, C. Wu, R. Zhang, and L. Sun, "Towards faster and better federated learning: A feature fusion approach," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 175–179.

[108] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 10 713–10 722.

[109] E. Diao, J. Ding, and V. Tarokh, "SemiFL: Semi-supervised federated learning for unlabeled clients with alternate training," vol. 35, 2022, pp. 17 871–17 884.

[110] K. Zhang, C. Yang, X. Li, L. Sun, and S. M. Yiu, "Subgraph federated learning with missing neighbor generation," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021, pp. 6671–6682.

[111] N. Shi, F. Lai, R. A. Kontar, and M. Chowdhury, "Fed-ensemble: Ensemble models in federated learning for improved generalization and uncertainty quantification," *IEEE Transactions on Automation Science and Engineering (T-ASE)*, 2023.

[112] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni, "Bayesian nonparametric federated learning of neural networks," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*. PMLR, 2019, pp. 7252–7261.

[113] T. C. Lam, N. Hoang, B. K. H. Low, and P. Jaillet, "Model fusion for personalized learning," in *Proceedings of the 38th International Conference on Machine Learning (ICML)*. PMLR, 2021, pp. 5948–5958.

[114] D. Contractor, D. McDuff, J. K. Haines, J. Lee, C. Hines, B. Hecht, N. Vincent, and H. Li, "Behavioral use licensing for responsible ai," in

2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT), 2022, pp. 778–788.

[115] R. W. Gomulkiewicz, "Open source license proliferation: Helpful diversity or hopeless confusion?" *Washington University Journal of Law & Policy*, vol. 30, no. 1, 2009.

[116] B. G. Tekgul, Y. Xia, S. Marchal, and N. Asokan, "WAFFLE: Watermarking in federated learning," in *Proceedings of the 40th International Symposium on Reliable Distributed Systems (SRDS)*. IEEE, 2021, pp. 310–320.

[117] B. Li, L. Fan, H. Gu, J. Li, and Q. Yang, "FedIPR: Ownership verification for federated deep neural network models," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 45, no. 4, pp. 4521–4536, 2023.

[118] L. Fan, K. W. Ng, and C. S. Chan, "Rethinking deep neural network ownership verification: Embedding passports to defeat ambiguity attacks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[119] J. Zhang, D. Chen, J. Liao, W. Zhang, G. Hua, and N. Yu, "Passport-aware normalization for deep model protection," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 22 619–22 628.

[120] Y. Uchida, Y. Nagai, S. Sakazawa, and S. Satoh, "Embedding watermarks into deep neural networks," in *Proceedings of the 2017 ACM on international conference on multimedia retrieval (ICMR)*, 2017, pp. 269–277.

[121] B. Darvish Rouhani, H. Chen, and F. Koushanfar, "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)*, 2019, pp. 485–497.

[122] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2020, pp. 2938–2948.

[123] H. Jia, M. Yaghini, C. A. Choquette-Choo, N. Dullerud, A. Thudi, V. Chandrasekaran, and N. Papernot, "Proof-of-learning: Definitions and practice," in *IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021, pp. 1039–1056.

[124] J. Chen, J. Wang, T. Peng, Y. Sun, P. Cheng, S. Ji, X. Ma, B. Li, and D. Song, "Copy, right? a testing framework for copyright protection of deep learning models," in *IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 824–841.

[125] S. Peng, Y. Chen, J. Xu, Z. Chen, C. Wang, and X. Jia, "Intellectual property protection of dnn models," *World Wide Web*, pp. 1–35, 2022.

[126] Y. Sun, T. Liu, P. Hu, Q. Liao, S. Ji, N. Yu, D. Guo, and L. Liu, "Deep intellectual property: A survey," *arXiv preprint arXiv:2304.14613*, 2023.

[127] L. Charette, L. Chu, Y. Chen, J. Pei, L. Wang, and Y. Zhang, "Cosine model watermarking against ensemble distillation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 9, 2022, pp. 9512–9520.

[128] Y. Li, Y. Bai, Y. Jiang, Y. Yang, S.-T. Xia, and B. Li, "Untargeted backdoor watermark: Towards harmless and stealthy dataset copyright protection," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[129] S. Shao, W. Yang, H. Gu, J. Lou, Z. Qin, L. Fan, Q. Yang, and K. Ren, "Fedtracker: Furnishing ownership verification and traceability for federated learning model," *arXiv preprint arXiv:2211.07160*, 2022.

[130] H. Jia, H. Chen, J. Guan, A. S. Shamsabadi, and N. Papernot, "A zest of LIME: Towards architecture-independent model distances," in *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, 2022.

[131] S. Guo, F. Zhang, S. Guo, S. Xu, and F. Qi, "Blockchain-assisted privacy-preserving data computing architecture for web3," *IEEE Communications Magazine*, vol. 61, no. 8, pp. 28–34, 2023.

[132] R. Blythman, M. Arshath, S. Vivona, J. Smékal, and H. Shaji, "Opportunities for decentralized technologies within ai hubs," in *NeurIPS 2022 Workshop on Decentralization and Trustworthy Machine Learning in Web3: Methodologies, Platforms, and Applications*, 2022.

[133] J. Benet, "IPFS - content addressed, versioned, p2p file system," *arXiv preprint arXiv:1407.3561*, 2014.

[134] T. McConaghy, "Ocean protocol: Tools for the web3 data economy," in *Handbook on Blockchain*. Springer, 2022, pp. 505–539.

[135] J. Wickström, M. Westerlund, and E. Raj, "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, 2022, pp. 238–245.

[136] J. Steeves, A. Shaabana, Y. Hu, F. Luus, S. T. Liu, and J. D. Tasker-Steeves, "Incentivizing intelligence: The bittensor approach," in *NeurIPS 2022 Workshop on Decentralization and Trustworthy Machine Learning in Web3: Methodologies, Platforms, and Applications*, 2022.

[137] Y. Zhan, J. Zhang, Z. Hong, L. Wu, P. Li, and S. Guo, "A survey of incentive mechanism design for federated learning," *IEEE Transactions on Emerging Topics in Computing (TETC)*, vol. 10, no. 2, pp. 1035–1044, 2021.