blockchains

MDPI

Article

Decision Tree-Based Federated Learning: A Survey

Zijun Wang and Keke Gai *

School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China; 3120221290@bit.edu.cn

* Correspondence: gaikeke@bit.edu.cn

Abstract: Federated learning (FL) has garnered significant attention as a novel machine learning technique that enables collaborative training among multiple parties without exposing raw local data. In comparison to traditional neural networks or linear models, decision tree models offer higher simplicity and interpretability. The integration of FL technology with decision tree models holds immense potential for performance enhancement and privacy improvement. One current challenge is to identify methods for training and prediction of decision tree models in the FL environment. This survey addresses this issue and examines recent efforts to integrate federated learning and decision tree technologies. We review research outcomes achieved in federated decision trees and emphasize that data security and communication efficiency are crucial focal points for FL. The survey discusses key findings related to data privacy and security issues, as well as communication efficiency problems in federated decision tree models. The primary research outcomes of this paper aim to provide theoretical support for the engineering of federated learning with decision trees as the underlying training model.

Keywords: federated learning; machine learning; decision tree; privacy protection; communication efficiency

Citation: Wang, Z.; Gai, K. Decision Tree-Based Federated Learning: A Survey. *Blockchains* **2024**, *2*, 40–60. https://doi.org/10.3390/blockchains2010003

Academic Editor: Sujit Biswas

Received: 29 December 2023
Revised: 17 January 2024
Accepted: 29 February 2024
Published: 7 March 2024

1. Introduction

Machine learning (ML) has experienced rapid development in the field of artificial intelligence, particularly in areas such as computer vision, natural language processing, speech recognition, etc. [1–6]. However, machine learning, especially use of deep learning models, often requires large datasets to ensure model performance and that training results are accurate. Currently, data are mostly stored in a decentralized manner, individually maintained by data owners. Directly sharing this data poses privacy and security risks. For instance, in the financial sector, user deposit and lending information are stored in different financial institutions (such as banks, insurance companies, etc.), and user data between these institutions are independent. In an ideal scenario, if institutions could collaborate and use federated data to train user credit models collectively, each institution could benefit. However, due to concerns about data privacy and the establishment of relevant laws and regulations [7,8], owners of highly sensitive data are unwilling to share them and prefer to keep such data within their own control [9]. Simultaneously, the high cost makes aggregating scattered data between different institutions challenging. Addressing the fragmentation and isolation of data while adhering to privacy protection regulations is a crucial challenge in the field of artificial intelligence.

Federated learning (FL) [10–13], as a distributed machine learning technique popular in recent years, trains a central model by collecting local updates or model parameters from users instead of raw data, thereby protecting users' sensitive local data and addressing privacy and security concerns in multi-party model training [13]. Based on different data partitions, FL can be categorized into horizontal federated learning (HFL), vertical federated learning (VFL), and federated transfer learning (FTL), as shown in Figure 1. In HFL, data from different users differ in the sample space but are the same in the feature space. Each party trains a local model, and the model parameters or gradients are sent to a central

server. The server collects and aggregates the results, then returns the updated results to each user. In VFL, the data samples between participants are aligned, but there are differences in the data features. Each participant keeps the data and models locally and exchanges intermediate computation results with the server. FTL is suitable for scenarios where there is little overlap in both data samples and data features among participants.

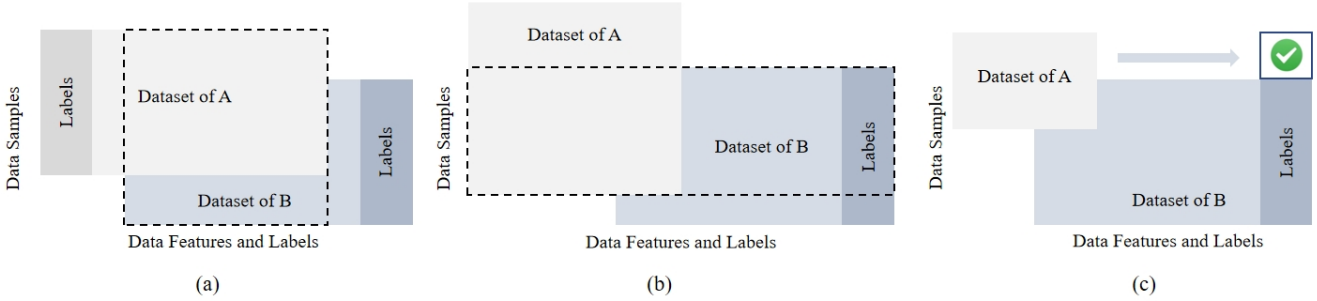

Figure 1. Classification of federated learning in [10]. (**a**) Horizontal federated learning (HFL), partitioned by samples. (**b**) Vertical federated learning (VFL), partitioned by features. (**c**) Federated transfer learning (FTL).

In recent years, many researchers have dedicated their efforts to FL algorithms to support effective machine learning models. Gascón et al. [14] proposed a linear regression model on vertically partitioned datasets and combined multi-party computation protocols to achieve scalable secure training. Cellamare et al. [15] introduced federated generalized linear models for the privacy-preserving analysis of horizontally partitioned data in real-world scenarios. Zhu et al. [16] used a multi-objective evolution algorithm to optimize neural network structures and proposed a scalable method for encoding network connectivity in a federated setting, thereby improving model training efficiency. Compared to traditional linear models [14,17] and neural network models [16,18,19], decision trees, as a significant machine learning algorithm, exhibit better accuracy and interpretability. According to our survey, decision tree models are widely used in classification and regression problems. Liu et al. [20] introduced the concept of revocable federated learning and implemented a new framework for federated random forests. Hou et al. [21] proposed a vertically federated random forest scheme based on dynamic changes in user data, which, while preserving privacy, can verify the integrity of data. Another highly representative approach is gradient boosting decision trees (GBDTs). A GBDT consists of multiple decision trees, and it sequentially builds weak learners through gradient boosting to minimize the loss function. Specific implementations of GBDT include XGBoost [22], SecureBoost [23], and FederBoost [24]. Due to the simple operation, high efficiency and interpretability, tree models have been widely applied in FL in recent years, demonstrating robust performance in practical applications, such as finance [25] and healthcare [26].

Many researches have explored the differences between tree-based models and other deep learning models in terms of accuracy and efficiency. Memon et al. [27] demonstrated that XGBoost exhibits superior performance in image classification tasks compared to artificial neural networks. On the other hand, tree-based models outperform other models on tabular datasets [28]. Due to the uneven distribution of features, small sample sizes, and large outliers in tabular data, using neural networks for prediction and training becomes challenging. In contrast, the inductive biases of decision trees make them perform better on tabular data [29–31]. Despite the various advantages of FL, our research indicates two challenges in using decision trees as the underlying model for FL.

The first types of challenge are security and privacy issues. In HFL, different participants can leverage local data for training, making privacy during gradient transmission and backpropagation a focal point. In VFL, security concerns arise primarily from the labels being on one side and the non-overlapping features. Different participants must ensure that local feature information is not disclosed. Meanwhile, participants without labels need to transmit gradient information for model training and enhancement. Current research

often employs techniques such as homomorphic encryption [32,33] or secure multi-party computation [34] to achieve federated decision tree training. Introducing lightweight secure aggregation helps reduce the computational overhead for encryption and decryption without exposing individual user data [24,35,36]. With the popularity of differential privacy techniques, training federated decision trees under differential privacy conditions can protect user privacy while minimizing the impact on model accuracy [35,37,38].

Another type of challenge is model convergence and communication efficiency issues. While tree models offer better interpretability compared to other machine learning models, the decision-making at intermediate nodes also introduces significant communication overhead. The characteristic of distributed storage in FL, coupled with the need to transmit substantial gradient information during the training process, poses a disadvantage for building decision tree models. In the context of federated decision tree implementation, the lack of an efficient communication mechanism among multiple participants becomes a critical limiting factor, especially when dealing with large-scale data communication. Our research suggests that establishing asynchronous/parallel training mechanisms [39], resource allocation optimization [36], and effective transformation of transmitted information [40] can all be viable approaches to enhance the efficiency of federated decision tree training.

We note that there are already many research schemes that combine federated learning with decision trees. However, there is a scarcity of survey work that systematically consolidates these efforts [41–43]. Therefore, the focus of this survey is the technical integration of federated learning and decision trees. This work combines research in relevant fields in recent years, covering various aspects, such as algorithm implementation, privacy protection, and the efficiency of tree models in different federated settings. Our contributions are as follows:

- This survey categorizes and summarizes federated decision tree models, explains the innovative points of each scheme, and compares the differences and connections between different schemes in multiple aspects.
- We elaborate on the main issues currently facing privacy in federated learning and summarize the performance differences brought about by using different cryptographic techniques and privacy protection schemes in training federated decision tree models.
- We consider using decision trees as the underlying model for federated learning, which involves a large amount of computation and communication between multiple parties during the training process. Therefore, we discuss iterative and aggregation strategies in federated learning to improve the convergence and communication efficiency of the model.
- Finally, we provide prospects for future research directions in this field.

The rest of the paper is organized as follows: Section 2 summarizes the implementation of decision tree algorithms in various federated settings and introduces the performance advantages of existing algorithms. Section 3 discusses the security and privacy protection methods in federated decision trees in detail. Section 4 discusses schemes for improving efficiency. The final section provides a conclusion to the paper.

2. Federated Decision Tree

A decision tree is a common machine learning algorithm. It judges data attributes in a tree structure and outputs the judgment results layer-by-layer. The final leaf node represents a classification or prediction result. The traditional ID3, C4.5, and Cart algorithms can obtain good training results in large databases.

Using the decision tree as the underlying model for federated learning, each participant can train a decision tree model locally and share the learned model parameters or gradients, so as to build an overall model that uses decentralized data for training. One of the main differences between the tree model and the neural network model is that when the federated average of the neural network is used, the model structure is predetermined and

sent to all participants. However, when building a decision tree, the tree structure is learned according to the local data of the different participants. This means that when using neural networks, all local models have the same structure, but when using decision trees, local models may have different structures [44]. Table 1 summarizes tree-based training models in different types of federated learning proposed in recent years.

Table 1. Summary of decision tree-based models under federated learning.

Proposed Model	HFL/VFL	Tree Algorithm	Security Measure	Performance Improvement		
				Security	Accuracy	Efficiency
Tree-based FL [35]	HFL	GBDT	DP + SecAgg	✓		✓
FEDXGB [33]	HFL	XGBoost	HE + SS	✓		✓
F-XGBoost [45]	HFL	XGBoost	K-Anon	✓		✓
Federated Forest [37]	HFL	Extra trees	LDP	✓		✓
DFedForest [46]	HFL	RF	Blockchain	✓	✓	
FL-XGBoost [47]	HFL	XGBoost	Encryption	✓		
FedXGB [48]	HFL	XGBoost	SS		✓	✓
DPBoost [49]	HFL	GBDT	DP		✓	
SimFL [50]	HFL	XGBoost	LSH		✓	
eFL-Boost [36]	HFL	GBDT	SecAgg			✓
Pri Fed GBDT [38]	HFL	GBDT	RDP	✓	✓	✓
SecureBoost [23]	VFL	XGBoost	HE	✓		
SecureBoost+ [51]	VFL	XGBoost	HE	✓		✓
Pivot [34]	VFL	RF & GBDT	HE + MPC	✓		✓
Secure XGBoost [52]	VFL	XGBoost	SecEnclave	✓		
FLSectree [53]	VFL	XGBoost	Encryption		✓	✓
FedXGBoost [54]	VFL	XGBoost	DP	✓		
VF2Boost [39]	VFL	GBDT	SecAgg			✓
FEVERLESS [55]	VFL	XGBoost	SecAgg + CDP	✓		✓
Fed-EINI [32]	VFL	RF & GBDT	HE		✓	✓
FedGBF [56]	VFL	RF & GBDT	Encryption			✓
FedRF [57]	VFL	RF	Encryption			✓
OpBoost [58]	VFL	XGBoost	LDP	✓	✓	
VPRF [21]	VFL	RF	HE	✓		✓
SGBoost [59]	VFL	XGBoost	SS + FE + SHE	✓	✓	✓
VF-CART [40]	VFL	CART	HE			✓
PriVDT [60]	VFL	GBDT	FSS			✓
FederBoost [24]	HFL/VFL	GBDT	SecAgg + DP		✓	✓

* DP denotes Differential Privacy. SecAgg denotes Secure Aggregation. HE denotes Homomorphic Encryption. K-Anon denotes K-Anonymity. LDP denotes Local Differential Privacy. LSH denotes Locality Sensitive Hashing. RDP denotes Rényi Differential Privacy. MPC denotes Multi-Party Computation. CDP denotes Center Differential Privacy. FSS denotes Function Secret Sharing. SS denotes Secret Sharing. FE denotes Functional Encryption. SHE denotes Symmetric Homomorphic Encryption.

In a horizontal federated setting, participants' samples share the same feature dimensions, as shown in Figure 2. At the beginning of training, participants initialize their local models independently by selecting suitable decision tree structures and hyperparameters. Participants autonomously construct decision trees on their local data using feature se-

lection and node-splitting strategies for model training. Once the local model training is completed, participants need to integrate their models into a global model. A common integration method is to use aggregation algorithms, such as FedAvg. In FedAvg, participants send the weights of their local models to the server, and the coordinator averages the model weights based on each participant's contribution. This process results in a global decision tree model. After one training round, participants update their local models and repeat the training for the next round until the model converges.

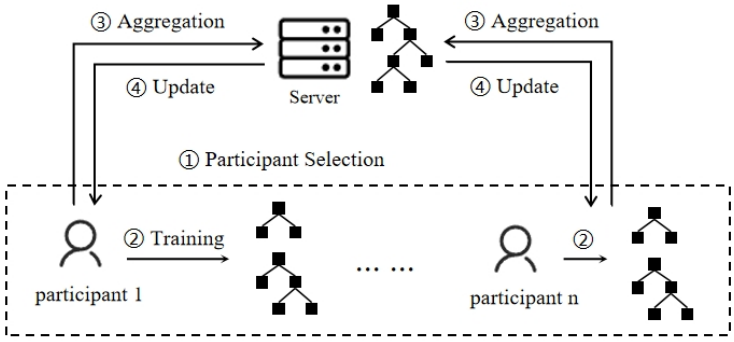

Figure 2. Training process of federated decision tree in horizontal federated learning.

On the basis of traditional solutions, Zhao et al. [35] proposed a federated learning scheme based on GBDT in data mining. Each participant uses local data to train a decision tree, which is added to the global model in turn. The privacy protection method is provided when sharing the model to reduce the risk of privacy data disclosure of the leaf node weight. At the same time, each update only needs to send the global model to the next owner, making the communication cost low. On this basis, Yamamoto et al. [47] used the greedy idea to improve the prediction performance. Through the central server, this actively selects the tree with the greatest loss to add to the global model, so that each model update only requires one round of communication.

In building a tree-based model, we need to determine the splitting points of samples and the weights of the leaf nodes. Usually, each participant selects the basis for node partitioning based on the feature distribution of their local data, for example, selecting features with the maximum information gain as the basis for node partitioning [61], or selecting features with the minimum Gini index for node partitioning [62,63]. In order to maintain the consistency and mergeability of the decision tree, participants need to coordinate the basis for node partitioning by sharing local data feature statistical information or making collaborative decisions for node partitioning in a secure computing environment. This can ensure that participants have a consistent basis for node partitioning, thereby obtaining a decision tree model that can be merged.

Different from horizontal federation settings, in vertical federated learning, data labels are usually owned by one party, and other participants need to cooperate with the party owning the labels to determine the tree partition structure. SecureBoost [23] defines the participant with a label as the active party, and the participant without a label as the passive party, as shown in Figure 3. After sample alignment, the active party sends the gradient and Hessian value of the samples to the passive party. The passive party divides the samples into buckets based on local features and returns the aggregated values of each bucket to the active party in the form of a histogram, as shown in Figure 4. The active party calculates the optimal splitting of nodes locally and synchronizes updates directly among the different participating parties. SecureBoost provides a fundamental framework for future research.

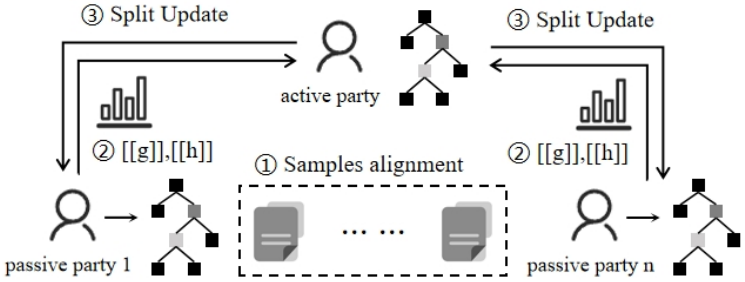

Figure 3. Training process of federated decision tree in vertical federated learning.

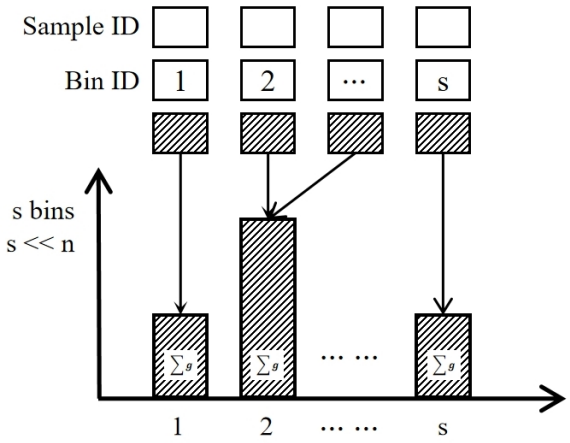

Figure 4. Passive parties build a histogram and aggregate the samples into bins.

The key step in building a decision tree is to find the optimal splitting of samples based on a feature and calculate the weight values of leaf nodes, that is, to calculate Equations (1) and (2).

$$L_{split} = \frac{1}{2}[\frac{(\sum_{i\in I_L} g_i)^2}{\sum_{i\in I_l} h_L + \lambda} + \frac{(\sum_{i\in I_R} g_i)^2}{\sum_{i\in I_l} h_R + \lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_L + \lambda}]. \tag{1}$$

$$w_l^{(t)} = -\frac{\sum_{i\in I_l} g_i}{\sum_{i\in I_l} h_i + \lambda}. \tag{2}$$

As can be seen, the calculation of this split score is only related to the first-order and second-order gradients, as well as the order of the samples. Therefore, in the scheme proposed by Tian et al. [24], each participant was asked to sort their samples based on their eigenvalues and put them in different buckets, and the sorting was coneyed to the active party for training. For any participant, they only know the order of the buckets and do not know the order of the samples inside the buckets, so they do not require any encryption or decryption operations. Xu et al. [40] proposed to hash the sample ID using the same hash function for each participant and then sorted the hashed IDs according to the characteristics. Participants divided the samples into histograms, where bins stored the hash ID of samples. In this scheme, the active party can obtain the sample distribution of the passive party in one communication, so the passive party only needs to communicate with the active party once during the training phase.

In addition, most existing work assumes that participants are honest; however, malicious clients may launch attacks on intermediate results or the final model. Blockchain [64–66] provides a reliable method to ensure mutual trust in a distributed environment. The introduction of blockchain, leveraging features such as consensus mechanisms, smart contracts, and incentive mechanisms, provides a more secure model training process for FL [67,68]. Souza et al. [46] proposed an FL system for creating random forests in a distributed manner, utilizing blockchain technology to ensure mutual trust. The system employs a distributed

registration of references to local model addresses, preventing malicious participants from compromising the model's accuracy. This marked the first attempt to integrate blockchain into federated decision tree models.

3. Security Scheme

Although in FL, participants do not send training data away from the local model, the local model can complete the training of the global model only by uploading parameters or gradients. However, attackers can still obtain the privacy information of users' local data through gradient or parameter information [69,70], and can steal the local model, which greatly threatens data privacy and brings security risks. On the other hand, due to the low level of mutual trust among participants in federated learning, the model is required to be more robust. We have summarized several attack issues that federated decision trees may face in practical applications. In real-world scenarios, federated decision trees often face the following types of attacks:

- **Model Inference Attack:** Attackers can infer sensitive information about training data by observing the output of the federated decision tree model [70–72]. Attackers can establish training data or infer specific eigenvalues by utilizing the probability distribution or decision path of the output, thereby infringing on the privacy of participants.
- **Model/Data Poisoning Attack:** Attackers may attempt to manipulate model updates or gradients of participants during federated learning, or use malicious data for model training, causing malicious interference to the aggregated decision tree model. This may lead to a decrease in the performance of the final model or produce misleading results during the inference stage [73–75].
- **Aggregation Information Leakage Attack:** Attackers can infer the data information of participants by observing the aggregation process of the model parameters or gradients [76,77]. By analyzing the changes in aggregation results, attackers may obtain sensitive information about data distribution or features.
- **Malicious Behavior:** Participants in federated learning may engage in malicious behavior, such as providing false model updates [78,79], tampering with data labels [80], or manipulating the aggregation process [81–83]. This malicious behavior may undermine the accuracy and reliability of the federated decision tree model and may also threaten data privacy.

Many solutions have been proposed for different attacks in the process of federated learning training and prediction. In the federated decision tree, privacy protection schemes based on cryptography, differential privacy, or security aggregation are often used for privacy protection. Table 2 provides a comparison of different privacy protection mechanisms and technologies used in the federated decision tree. In the next section, we explain these in detail.

Table 2. Comparison of privacy protection mechanisms and technologies.

Mechanisms	Technology	Principle	Advantages	Disadvantages
Data ambiguity	DP	Central/local noise provided to perturb data or gradient values.	High computational efficiency; Low communication overhead; Post-processing, protecting published data.	Decreased accuracy and availability of training models.
Process encryption	HE	Gradient encryption, operating on ciphertext.	Strict privacy protection.	Unable to handle complex operations; Low computational efficiency; High storage overhead.
	SMC	Intermediate data are private and cannot be learned by other parties.	Prevent man-in-the-middle attack and data leakage.	Low computational efficiency; High communication overhead.

3.1. Cryptographic Technology

Homomorphic encryption (HE), as an important part of cryptography, has been used in many federated learning algorithms [34,84,85] due to its good performance arithmetically. From Equations (1) and (2), we can see that in the decision tree model, the decision of split points and the calculation of leaf weights involve a large number of gradient summation calculations, and some algorithms also need addition operations to build histograms. Therefore, additive homomorphic encryption can be used as a good privacy protection scheme.

HE can provide privacy protection for all parties in the training process of a federated decision tree model [23,32–34]. An HE system allows direct operation of ciphertext without decryption operation, which is proved to ensure high security [86]. In the typical model, the active party homomorphic encrypts the calculated gradient value and the Hessian value of the samples and sends them to other passive parties. The passive party accumulates the encrypted data according to the feature, and finally returns them to the active party for decryption. It can be seen for participants with data labels that HE can protect the gradient values from being stolen by malicious parties during transmission to infer the original data, and at the same time ensure that the data are used correctly, thus ensuring the security of the label. On the other hand, for data providers without data labels in vertical federated learning, HE can reduce the disclosure of feature dimension information by use of the histogram.

Liu et al. [33] proposed solutions to two unresolved issues in the current HE-based FL system. One was to perform forced aggregation on the server before decryption, otherwise the server may be able to learn user model updates. Second, since most existing HE-based federated learning solutions cannot solve the problem of accidental user dropout. They first proposed a hybrid scheme combining homomorphic encryption and secret sharing, which can force the central server to perform aggregation operations and is robust against user exits.

For the passive party, the construction of a histogram is essentially a homomorphic addition operation. If the addends differ in the exponential terms, homomorphic addition will perform a scaling operation, as shown in Equation (3).

$$[[u]] \oplus [[v]] = \begin{cases} \langle e_u, (B^{e_v - e_u} \otimes [[U]]) \oplus [[V]] \rangle, & if\ e_v < e_u \\ \langle e_u, [[U]] \oplus [[V]] \rangle, & if\ e_v = e_u \\ \langle e_u, [[U]] \oplus (B^{e_v - e_u} \otimes [[V]]) \rangle, & if\ e_v > e_u \end{cases} \tag{3}$$

When constructing a histogram, the encrypted gradient statistical data are accumulated one-by-one into the bin of the histogram. The exponential term of each bin is determined by the maximum exponential term, and the total number of scaling operations is affected by the data order. Fu et al. [39] proposed a re-ordering encryption scheme. They sorted the encrypted statistics according to the exponential term from the smallest to the largest before performing the operation, so that only $(E-1)$ scaling was required, where E is the number of unique values in the exponential term. This was also the first operation proposed using the federal decision tree model to optimize homomorphic addition.

Another common solution is secure multi-party computing (SMC) [87,88]. It ensures that all participants, except for the output, cannot learn any other information, which can be used to securely aggregate the transmitted gradient. However, SMC does not provide privacy guarantees for the final model and is still vulnerable to inference attacks and model reversal attacks. These vulnerabilities have been highlighted in previous studies [89,90].

Mohassel et al. [91] applied SMC in their proposed solution. However, they assume that each participant's data can be outsourced to many non-collusive servers. However, in practical application scenarios, this assumption is difficult to implement. Wu et al. [34] proposed Pivot, which does not rely on any trusted third party to provide protection in a vertically federated environment. Pivot makes use of two complementary encryption technologies: threshold homomorphic encryption (TPHE) and MPC. When TPHE can only

support a set of restrictive calculations, SMC is called to complete the training. This scheme ensures that each client only learns the final tree model and does not learn anything else.

3.2. Different Privacy

The initial goal of federated learning is to share the weight of the training model rather than the original data so as to protect local sensitive information. However, some studies show that the weight will also leak privacy, thus enabling inference about the original data [92].

Differential privacy (DP), as a robust standard for measuring privacy, establishes security through rigorous mathematical proofs, enabling data analysis without disclosing individual private information [93,94]. In the federated setting, compared to cryptography-based techniques, DP reduces the computation and communication rounds, thereby mitigating significant time and space costs. The core idea of DP involves introducing noise into individual data, making it challenging for attackers to distinguish whether a specific individual's data participated in the computation. Therefore, based on our research, in the training process of federated decision tree models, DP techniques can be applied from three perspectives: the data input, the learning process, and the learned model.

(1) *Inputs.* Participants can perturb their local training data by adding noise to individual feature values or model parameters. For instance, Laplace noise [95] or Gaussian noise [96] can be applied to real-valued data, while the exponential mechanism can be used for adding noise to discrete data [97].

(2) *Learning process.* Participants can add noise to local model parameters during model training. By adding noise to gradients or perturbing parameters, the privacy of the model parameters can be protected. Participants need to control the privacy budget for each training round to ensure a level of privacy protection is maintained during the training process. Studies on the allocation of privacy budgets among different trees, on the one hand, use sequential composition to evenly allocate budgets to each tree [98,99]. However, when the number of trees is large, the privacy budget allocated to each tree is very small; if the noise scale is set to be proportional to the number of trees, it can cause a significant loss of accuracy. On the other hand, consideration can be given to disjoint inputs to different trees [35], with each tree satisfying differential privacy through parallel composition. However, in this approach, when the number of trees is large, the instances assigned to each tree may be very small because the inputs cannot overlap. As a result, the tree can be too weak to achieve meaningful learnt models.

Li et al. [49] pointed out that invalid privacy budget allocations and overly loose sensitivity bounds among different trees in the GBDT model may cause serious accuracy losses to the final model. They proposed to adaptively control the gradients of the training data and the clipping of leaf nodes in each iteration based on the property of the gradient and the contribution of each tree in GBDTs. In their framework, a two-level boosting structure named EoE was proposed, which allocates privacy budgets between trees through a combination of sequential and parallel composition. For a single tree, half of the privacy budget is allocated to the leaf nodes, and the remaining half is allocated equally to the depth of each layer of the internal nodes. For multiple tree sets, parallel composition is applied within the set by sampling disjoint subsets, and multiple such sets are continuously trained using the same training set through sequential composition. This design allows for increased effectiveness while leveraging privacy budgets.

(3) *The learnt model.* The split decision can reveal sensitive information about the training set to a certain extent. In the class estimation process of leaf nodes, noise can be added to blur the final prediction results to protect individual privacy. However, the addition of noise will directly affect the accuracy of the final model, so the trade-off between noise and error has been investigated. Currently, there are many studies on training decision tree models under centralized differential privacy (CDP) [39,100,101]. Most of them are based on the decision forest, and due to the strict constraints of CDP, there will be precision loss when it is extended to federal settings. Wang et al. [55] proposed to

randomly select a noise leader according to the score in each iteration process to aggregate the local noise provided by different participants, which not only eliminates the requirement for a trusted third party, but also avoids the superposition of a large number of noises.

Considering that the tree splitting process will leak privacy, FederBoost [24] satifies the $\epsilon-$ LDP, making any two samples in a bucket indistinguishable when dividing the samples. FedXGBoost [54] uses LDP to add noise to perturb the first-order approximation, and calculates the split score through the perturbed results so as to accelerate the training process on the premise of small accuracy loss. OpBoost [58] desensitizes the training data using distance-based LDP (dLDP), and combines an effective sampling distribution to find the trade-off between desensitization values and privacy, thus improving the accuracy and efficiency of the original LDP model.

In addition to the above plan, Rényi differential privacy (RDP) [102] is a new variant of differential privacy, which is used to calculate the privacy loss of a composition mechanism. Previous research [103,104] has added RDP as noise to the federated learning local model training process. Recent research [38] has applied RDP to the calculation of split candidates, the selection of split points, and the calculation of weights. The balance between importance and added noise through privacy accounting was used to constrain the cumulative privacy loss caused by the DP algorithm. Through the combination of privacy, the model performance was close to the non-privacy setting, and the training accuracy rate reached over 90%.

3.3. Data Security Aggregation

In FL, different participants train private models through local data. When the central server aggregates, the high-dimensional model or data will lead to the problem of privacy budget explosion. Due to the requirements of the high dimensionality of the data, uncertainty of users, and robustness of training models in FL, the demand for security aggregation protocols has been stimulated [105].

A secure aggregation scheme based on secure shuffling [106–108] provides privacy assurances through local encoders and a third-party shuffler. Erlingsson et al. [109] initially demonstrated the amplification effect of shuffling models on local differential privacy. Combining theoretical aspects of subsampling, Kasiviswanathan et al. proved that secure shuffling can transform local differential privacy into centralized differential privacy, resulting in a significant reduction in the privacy budget. The FLAME model proposed by Liu et al. [110] builds upon secure shuffling by implementing high-dimensional gradient second sampling, further reducing the privacy budget while significantly enhancing the model's usability.

In a distributed system, Google proposed a scheme to aggregate local training models into a federated global model through iteration, and split the model training process into multiple participants [111]. Different users can obtain local decision tree models through parallel training, and for the resulting decision tree set, the aggregation algorithm usually selects a representative tree or merges trees through an aggregation protocol. The tree selection method usually employs the similarity to select the most similar and representative decision tree with other models [112]. According to previous research [113], we classify the aggregation of decision trees, as shown in Table 3.

The goal of tree merging is to minimize the entropy of the split nodes when building each decision tree, which is consistent with the goal of reducing the prediction error of each participant in federated learning. Zhao et al. [35] used only local data to promote each tree in the training process, and did not use data information from other parties. This method has limitations. Li et al. [50] proposed SimFL, which uses the idea of "weights" to collect gradients from other parties through sample similarity and to aggregate them locally. Their scheme resulted in a decrease in the error rate of the training model on different datasets. The FEVERLESS mechanism proposed by Wang et al. [55] safely aggregates gradient information into a private histogram by using an aggregation scheme, and uses a histogram for subsequent calculations. At the same time, in order to avoid the problem of multiple

noise accumulation leading to a decline in model accuracy, a verifiable random function (VRF) was used to select the noise leader in their scheme, aggregate the differential privacy noise from other participants, and adding them to the histogram for privacy protection.

Table 3. Classification of decision tree aggregation.

Aggregating decision trees	**Structure-based**: according to the hierarchical structure of the tree, then aggregating different layers. Classifying the samples in the sub-nodes in the hierarchy.
	Weight-based: considering the division of the tree as a set, and aggregating the weight values of the samples in the set.
	Logic-based: considering the setting up of the decision tree as a set of logical rules, and then aggregating the logical expressions.
	Dataset-based: fitting the results of multiple decision trees onto a complete dataset.
Selecting decision trees	In one iteration, selecting the single tree that best represents the information of all datasets as the global model.

From another perspective, the decision tree can be conceptualized as hierarchical structures representing the dataset. Each record in the dataset is classified by the tree and assigned to a specific leaf. By focusing solely on the hierarchical structure rather than the queries associated with each node, the aggregation of decision trees can be understood as the aggregation of these hierarchical structures [113]. Kargupta et al. [114] proposed a method for tree aggregation based on datasets. They proposed transforming decision trees into Fourier spectra, and combining them into an entire dataset through vector addition in dual space.

Another solution for secure aggregation is direct selection. The purpose of tree selection is to select a single tree that can represent the overall data features. Compared with integrated methods, it reduces the workload of the additional calculations. On the other hand, only representative models are retained without the need to provide additional privacy levels for other data information. Miglio et al. [115] proposed a framework for comparing decision trees through semantic similarity and dataset similarity. Semantic similarity measures the consistency of class predictions across decision trees in the attribute space, while dataset similarity considers attribute space probability distribution, class joint probability distribution, and conditional class probability distribution. However, based on these two methods, the selection phase of the tree requires additional runtime to apply each decision tree to the validation instance. The decision tree with the highest accuracy can simply be chosen from multiple decision trees [116,117]. The advantage of this method is that it is simple and intuitive, easy to implement and explain. However, it ignores the confidence differences between decision trees. If there are significant differences in the accuracy of the decision trees, selecting only the decision tree with the highest accuracy as the aggregation result may overlook the important contributions of the other decision trees.

Summary. We summarize the above studies as follows:

- Current research on decision trees in federated learning is mostly focused on VFL. In a horizontal setting, it is difficult to aggregate directly through model parameters like neural networks, as different participants use different features to split the intermediate nodes. In VFL, it is more feasible to establish synchronization layer-by-layer among the parties.
- We summarized the existing security schemes applied in the federal decision tree model, including, but not limited to, HE, MPC, DP and secure aggregation. We explained the application of various technologies in federated decision trees, as well as the privacy protection capabilities and performance advantages and disadvantages they provide.

- The introduction of security technology has had an impact on the cost of computation and communication, as well as on the accuracy of models. Following review, we believe that balancing privacy protection and model accuracy in designing security technology solutions is a direction for further research.

4. Efficiency Scheme

The performance evaluation of models is mostly considered from three aspects: accuracy, security, and communication efficiency. Various indicators can be used to evaluate the accuracy of the decision tree model, such as accuracy, precision, recall rate, F1-score, etc. These indicators can be calculated by predicting the model on local data and comparing it with real labels. The evaluation of the privacy protection level of the model needs to consider the risk of privacy leakage. Both of these aspects are mentioned in our earlier introduction. In this section, we focus on evaluating the efficiency of the federated decision tree.

In FL, model training is usually carried out through federated iteration. The model gradually improves and converges to a global optimal solution in each iteration. Due to the distributed storage of data, it is very important to pay attention to the training efficiency of the federated decision tree for the practical application of federated learning. Firstly, federated learning involves model training and parameter exchange between multiple participants. If the training efficiency is low, it will consume a lot of computing resources and time. Secondly, if the training efficiency is low, model updates may become slow, resulting in the model not being able to adapt to new data and situations in a timely manner. Finally, in practical application scenarios, inefficient training processes can delay the deployment and application of the model, as well as adversely affect the experience and satisfaction of participants.

In terms of neural networks and linear regression models, the research on improving the communication efficiency of federated learning focuses on the number of communication rounds and bandwidth. The main programs include selecting representative clients, reducing the number of model updates, and compressing the model. The client can be selected to reduce costs by limiting the number of participants.The framework proposed in Chen et al. [118] used a probabilistic device selection scheme to select only those clients with high probability for model transmission, thus enhancing convergence speed and reducing training loss. Another approach involves reducing model updates, where techniques like Bayesian neural networks [18,119,120] enable synchronous and asynchronous model updates across multiple machines. Kasturi et al. [121] proposed a simpler federated fusion learning scheme, which allows the distribution parameters of the local data to be sent to the central server instead of the model parameters. These parameters are used to generate synthetic data on the central server to train a global machine learning model, reducing communication to a single round. In vertical federated learning, each party is considered to make multiple local updates before each communication, which can reduce the number of communication rounds between clients. Li et al. [120] proposed an asynchronous vertical federated learning framework with gradient prediction and two-terminal sparse compression, in which compression occurs on local models to reduce the training time and costs. While existing compression techniques focus on gradient compression to reduce the training time and transmission cost, considering compressing local client data before aggregation for final model training can further minimize communication rounds [122]. This approach protected privacy by not exposing local data and limited the process to a single communication round.

In decision tree models, the improvement of efficiency can be considered in terms of the two stages of training and prediction, as shown in Table 4.

In the training stage, the goal is to reduce the number of iteration rounds and accelerate the speed of each iteration. Inspired by previous work, dimensionality reduction is a strategy to improve the communication efficiency, which we define in two categories: client dimensionality reduction and feature dimensionality reduction. Unlike previous random

schemes [123,124], the FedMint scheme proposed by Wehbi et al. [125] uses a bootstrap method to obtain initial accuracy values for new IoT devices, and utilizes a matching game method to create selection lists for clients and federated servers based on specific criteria. Unlike random selection, their method enables federated servers to consider the data type and accuracy level of IoT devices to achieve an effective selection process. Methods for feature dimensionality reduction include principal component analysis (PCA) [118], feature selection based on information gain [126], and model-based feature selection [127].

Table 4. Tree-based model training in vertical federated learning.

Training Stage	Incremental learning.
	Model compression and pruning.
	Parallel and asynchronous computing.
	Sampling and subsampling.
Prediction Stage	Client inference.
	Compression model and quantization.

We can consider the reasonable arrangement of time and resources in different components, allocate more data to the important parts, and allocate less data to the parts that have little impact on the final results, thus improving the overall training efficiency of the federated model. Yamamoto et al. [21] balanced resource allocation in local and global computing. They suggest that the leaf node weights contribute more to prediction performance. Therefore, the determination of the tree structure in their framework does not require multiple communication of gradient histograms, but is determined and shared by one party according to local data. The weights of leaf nodes are calculated by an aggregator after collecting the results for each participant, which can optimize the cost function based on the global distribution.

Xu et al. [40] suggested that in the previous training process of the federated learning decision tree model, the passive parties need to establish a histogram of gradient and Hessian values to send to the active party during each split, resulting in a huge communication overhead. In the framework they proposed, the passive party places the hashed sample IDs into bins at the beginning of model training so as to establish a histogram for each feature. The passive party only needs to communicate with the server once during the whole training phase, and then the server and the active party interact to determine the best split point. In this process, the server only needs to transfer the dot product and feature information triplets, which greatly reduces the traffic in the training process.

On the other hand, the training at different stages can be parallelized, which is very common in vertical federated learning. In the process of vertical federated learning, due to the restriction that the label is located on one side, more calculations need to be completed by the side with the label, which makes other participants remain in the waiting state for a long time. Fu et al. [39] proposed VF2Boost to improve the training efficiency through use of a parallel protocol. They divided the samples into small batches for separate processing and transmission, thereby shortening the waiting time of both parties and avoiding message queue congestion caused by the transmission of a large amount of ciphertext in a short period of time. They also parallelized the data encryption, gradient transmission, and node-splitting stages in the training process, achieving an acceleration of 1.90–2.21×.

In the prediction stage, the existing scheme uses a multi-interactive reasoning framework to calculate the reasoning results. The decision path is determined by sequentially searching the decisions on the current node and moving to the next node, requiring a large number of communications between both parties. In order to reduce the number of communications and to improve efficiency, it is preferred to use local models and parameters in federated learning to first predict, and then communicate with other parties, rather than asking each node separately. A participant may judge a sample based solely on local

information and fall into one or more leaf nodes, while the active party can combine the results of multiple participants and associate them with the final prediction [32,37].

The Fed-EINI model proposed by Chen et al. [32] emphasizes the interpretability and efficiency of the models. They proposed encrypting decision paths while making feature meanings and importance public information. In the original SecureBoost model's inference process, when a tree node splits based on the features of the passive party, the active party asks the passive party how to proceed. After obtaining the decision path information, the active party continues with the prediction. In Fed-EINI, the passive party (defined as the host) and the active party (defined as the guest) use their own node information to infer possible prediction results, i.e., predicting which leaf node the instance will fall into. By multiplying the output vectors obtained by different participants, the final prediction result can be obtained. As shown in Figure 5, we can clearly see the difference between the two approaches. Fed-EINI provides a more secure and interpretable inference framework, avoiding the leakage of feature path decision information by encrypting the matrix calculations. It has higher accuracy and efficiency.

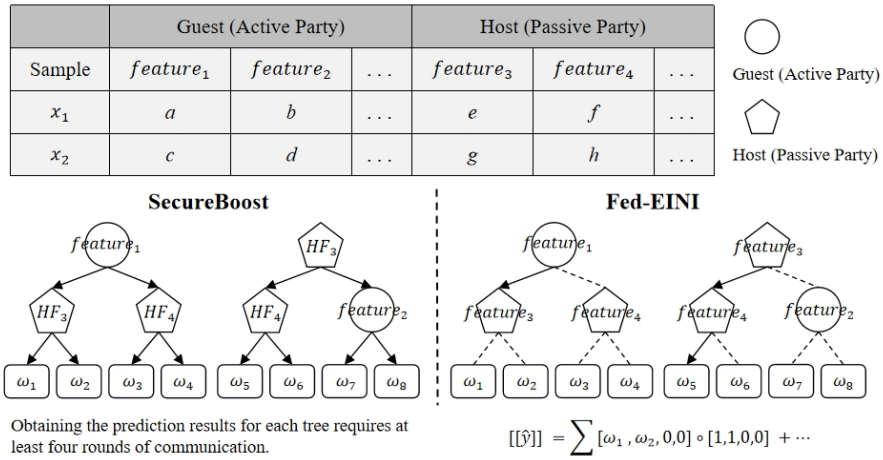

Figure 5. Comparison between standard SecureBoost and Fed-EINI.

Gradient quantization is a technique that quantifies gradients to lower accuracy values and reduces the number of transmitted gradients. The efficiency improvement and effectiveness of quantization in neural network training was considered in [128]. However, the existing literature rarely discusses the possibility of low precision training for GBDT. BitBoost [129] utilizes quantization gradients to improve efficiency, but it quantifies gradients in a deterministic manner. On this basis, Shi et al. [130] adopted a random quantization scheme and an adaptive method that is conducive to maintaining good model accuracy and achieving significant acceleration on different computing platforms. After investigation, we have reason to believe that quantitative training can be combined with histogram-based schemes. Using histograms can reduce the amount of statistical data and reduce the communication costs. Quantization training can reduce the size of the histogram, compress the histogram by converting floating-point arithmetic numbers into low precision values before sending, and then decode it into high precision values. Therefore, we believe that it has universal benefits for distributed training.

Summary. We summarize the above studies as follows:

- Our efficiency evaluation of the federated decision tree is divided into the training stage and the prediction stage. We summarized the existing improvement plans and provided an overview of the experimental results.
- When training a decision tree model in a federated environment, many factors, such as safety, accuracy, and efficiency need to be considered. Currently, most solutions focus on one or two of them, and there is room for improvement.

5. Discussion

In this survey, we consider the intersection of decision trees and federated learning, exploring the amalgamation of these two powerful paradigms in contemporary machine learning research.

This survey begins by establishing a fundamental understanding of federated learning, which is a collaborative approach where multiple participants cooperate to train a global model using local data, aiming to enhance training accuracy. Then, it introduces the collaborative training process of decision trees in various federated settings. Subsequently, it provides a comprehensive summary, classification, and analysis of research schemes utilizing decision trees as the underlying model, focusing on both security and efficiency aspects.

Combining our previous research findings with the practicality of federated learning, we discuss its research feasibility in various domains, such as blockchain, the Internet of Things (IoT), and fog computing.

Blockchain, as a distributed ledger, can be integrated well with federated learning. The immutable and decentralized nature of blockchain ensures transparency and security in the decision-making process. Each participant's decision tree model updates are recorded in a tamper-proof manner, providing a transparent and verifiable history of the federated learning process. This not only enhances the integrity of the models but also establishes a trust layer among participants. Fu et al. [131] proposed a verifiable decision tree prediction scheme for decision tree prediction. The integration of decision tree models with blockchain technology offers a robust solution to the security challenges associated with cloud-based machine learning services. The Merkle tree, hash function, and commitments are leveraged to generate efficient verification proofs, ensuring the integrity of decision tree predictions. The transparency and immutability of the blockchain provide a secure platform for clients to verify the correctness of results, addressing concerns related to malicious attacks or computational failures [132,133].

Apart from the previously mentioned types of attacks, phishing attacks [134] are a common form of network threat, particularly in the realm of blockchain, posing a significant security risk. Attackers employ social engineering techniques to deceive miners, leading them to add malicious blocks to the blockchain, potentially disrupting the entire blockchain system. Current efforts to combat phishing attacks include consensus protocols, but they may fail when miners attempt to add new blocks. Zero-trust policies are gradually being introduced as a method, but their deployment is still ongoing and requires a considerable amount of time. A more accurate approach to phishing attack detection involves the use of machine learning models with specific features to automatically classify attempts as phishing attacks or legitimate ones [135]. In the context of preventing phishing attacks in blockchain, federated decision trees may be employed to consolidate information from various miners or nodes, collaboratively constructing a model capable of identifying phishing attacks. This enables the entire model to gain a more comprehensive understanding of behavior patterns within the network.

Next, we suggest that federated learning and machine learning are very valuable in the context of software-defined networking (SDN) for the Internet of Things (IoT) and fog computing. The centralized control plane of SDN provides a global view of the network topology, aiding in achieving flexibility and simplifying the complexity of the network nodes. However, the current SDN architecture faces significant security threats [136,137], especially from distributed denial of service (DDoS) attacks. FL offers advantages in terms of real-time responsiveness and adaptability for DDoS detection. Its capability to update models in real-time enables it to adapt to new attack patterns, enhancing the timeliness of detection. Furthermore, FL avoids the need to transmit large amounts of raw data to a central server, thereby reducing the burden on network transmissions. This is particularly crucial for IoT devices and edge computing scenarios as these environments often have limited bandwidth and resources.

In addition, we consider the possibility of constructing cross-domain and cross-modal federated decision trees. Current research on federated decision trees primarily focuses on data from a single domain with similar data distributions. Future research could extend federated decision trees to scenarios involving cross-domain and cross-modal data, such as jointly modeling and making decisions on data originating from different domains or featuring different data types. This extension aims to enhance the capability of training downstream tasks.

6. Conclusions

Federated learning is an emerging and rapidly advancing technology wherein multiple participants collaborate to train models. Currently, extensive efforts have been invested in the development of federated learning systems. This survey considers the federated learning approach with the decision tree as the foundational model. It analyzes and introduces the establishment of federated decision tree models, privacy protection technologies, and efficiency evaluation schemes. The comparison of existing work across the technical dimensions of accuracy, security, and efficiency enables this survey to serve primarily as a theoretical reference for future endeavors in federated learning based on decision tree models. Finally, we also consider the development direction of the integration of federated decision trees with technologies such as blockchain and the Internet of Things.

Author Contributions: Investigation, Z.W.; writing—original draft, Z.W.; writing—review and editing, K.G.; project administration, K.G. All authors have read and agreed to the published version of the manuscript.

Funding: National Defense Basic Scientific Research Program of China under grant number JCKY2020602B008.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhuang, J.; Yu, J.; Ding, Y.; Qu, X.; Hu, Y. Towards Fast and Accurate Image-Text Retrieval with Self-Supervised Fine-Grained Alignment. *IEEE Trans. Multimed.* **2023**, *26*, 1361–1372. [CrossRef]
2. Peng, W.; Hu, Y.; Yu, J.; Xing, L.; Xie, Y. APER: Adaptive evidence-driven reasoning network for machine reading comprehension with unanswerable questions. *Knowl.-Based Syst.* **2021**, *229*, 107364. [CrossRef]
3. Yu, J.; Jiang, X.; Qin, Z.; Zhang, W.; Hu, Y.; Wu, Q. Learning dual encoding model for adaptive visual understanding in visual dialogue. *IEEE Trans. Image Process.* **2020**, *30*, 220–233. [CrossRef] [PubMed]
4. Yu, J.; Zhang, W.; Lu, Y.; Qin, Z.; Hu, Y.; Tan, J.; Wu, Q. Reasoning on the relation: Enhancing visual representation for visual question answering and cross-modal retrieval. *IEEE Trans. Multimed.* **2020**, *22*, 3196–3209. [CrossRef]
5. Gai, K.; Guo, J.; Zhu, L.; Yu, S. Blockchain meets cloud computing: A survey. *IEEE Commun. Surv. Tut.* **2020**, *22*, 2009–2030. [CrossRef]
6. Yu, J.; Zhu, Z.; Wang, Y.; Zhang, W.; Hu, Y.; Tan, J. Cross-modal knowledge reasoning for knowledge-based visual question answering. *Pattern Recognit.* **2020**, *108*, 107563. [CrossRef]
7. Zaeem, R.N.; Barber, K.S. The effect of the GDPR on privacy policies: Recent progress and future promise. *Acm Trans. Manag. Inf. Syst.* **2020**, *12*, 1–20. [CrossRef]
8. Gai, K.; Xiao, Q.; Qiu, M.; Zhang, G.; Chen, J.; Wei, Y.; Zhang, Y. Digital twin-enabled AI enhancement in smart critical infrastructures for 5G. *Acm Trans. Sens. Netw.* **2022**, *18*, 1–20. [CrossRef]
9. Zhang, Y.; Gai, K.; Xiao, J.; Zhu, L.; Choo, K.-K.R. Blockchain-empowered efficient data sharing in Internet of Things settings. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 3422–3436. [CrossRef]
10. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated machine learning: Concept and applications. *Acm Trans. Intell. Syst. Technol.* **2019**, *10*, 1–19. [CrossRef]
11. McMahan, H.B.; Moore, E.; Ramage, D.; Arcas, B.A.Y. Federated learning of deep networks using model averaging. *arXiv* **2016**, arXiv:1602.05629.
12. Li, Q.; Wen, Z.; He, B. Federated learning systems: Vision, hype and reality for data privacy and protection. *arXiv* **2019**, arXiv:1907.09693.
13. Li, Z.; Huang, C.; Gai, K.; Lu, Z.; Wu, J.; Chen, L.; Choo, K.K.R. AsyFed: Accelerated Federated Learning With Asynchronous Communication Mechanism. *IEEE IoT J.* **2022**, *10*, 8670–8683. [CrossRef]

14. Gascón, A.; Schoppmann, P.; Balle, B.; Raykova, M.; Doerner, J.; Zahur, S.; Evans, D. Secure linear regression on vertically partitioned datasets. *IACR Cryptol. ePrint Arch.* **2016**, *2016*, 892.
15. Cellamare, M.; van Gestel, A.J.; Alradhi, H.; Martin, F.; Moncada-Torres, A. A federated generalized linear model for privacy-preserving analysis. *Algorithms* **2022**, *15*, 243. [CrossRef]
16. Zhu, H.; Jin, Y. Multi-objective evolutionary federated learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 1310–1322. [CrossRef] [PubMed]
17. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Zhao, S. Advances and open problems in federated learning. *Found. Trends Mach. Learn.* **2021**, *14*, 1–210. [CrossRef]
18. Yurochkin, M.; Agarwal, M.; Ghosh, S.; Greenewald, K.; Hoang, T.N.; Khazaeni, Y. Bayesian nonparametric federated learning of neural networks. In Proceedings of the ICML. PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 7252–7261.
19. Liu, Y.; James, J.; Kang, J.; Niyato, D.; Zhang, S. Privacy-preserving traffic flow prediction: A federated learning approach. *IEEE Internet Things J.* **2020**, *7*, 7751–7763. [CrossRef]
20. Liu, Y.; Ma, Z.; Yang, Y.; Liu, X.; Ma, J.; Ren, K. Revfrf: Enabling cross-domain random forest training with revocable federated learning. *IEEE Trans. Dependable Secur. Comput.* **2021**, *19*, 3671–3685. [CrossRef]
21. Hou, J.; Su, M.; Fu, A.; Yu, Y. Verifiable privacy-preserving scheme based on vertical federated random forest. *IEEE Internet Things J.* **2021**, *9*, 22158–22172. [CrossRef]
22. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the SIGKDD, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
23. Cheng, K.; Fan, T.; Jin, Y.; Liu, Y.; Chen, T.; Papadopoulos, D.; Yang, Q. Secureboost: A lossless federated learning framework. *IEEE Intell. Syst.* **2021**, *36*, 87–98. [CrossRef]
24. Tian, Z.; Zhang, R.; Hou, X.; Liu, J.; Ren, K. Federboost: Private federated learning for gbdt. *arXiv* **2020**, arXiv:2011.02796.
25. Benhamou, E.; Ohana, J.; Saltiel, D.; Guez, B. *Planning in Financial Markets in Presence of Spikes: Using Machine Learning GBDT*; Université Paris-Dauphine: Paris, France, 2021.
26. Zhang, X.; Yan, C.; Gao, C.; Malin, B.A.; Chen, Y. Predicting missing values in medical data via XGBoost regression. *Healthc. Inform. Res.* **2020**, *4*, 383–394. [CrossRef] [PubMed]
27. Memon, N.; Patel, S.B.; Patel, D.P. Comparative analysis of artificial neural network and XGBoost algorithm for PolSAR image classification. In Proceedings of the TPAMI, Tepzur, India, 17–20 December 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 452–460.
28. Grinsztajn, L.; Oyallon, E.; Varoquaux, G. Why do tree-based models still outperform deep learning on tabular data? *arXiv* **2022**, arXiv:2207.08815.
29. Popov, S.; Morozov, S.; Babenko, A. Neural oblivious decision ensembles for deep learning on tabular data. *arXiv* **2019**, arXiv:1909.06312.
30. Chen, Y. Attention augmented differentiable forest for tabular data. *arXiv* **2020**, arXiv:2010.02921.
31. Luo, H.; Cheng, F.; Yu, H.; Yi, Y. SDTR: Soft decision tree regressor for tabular data. *IEEE Access* **2021**, *9*, 55999–56011. [CrossRef]
32. Chen, X.; Zhou, S.; Yang, K.; Fao, H.; Wang, H.; Wang, Y. Fed-EINI: An efficient and interpretable inference framework for decision tree ensembles in federated learning. *arXiv* **2021**, arXiv:2105.09540.
33. Liu, Y.; Ma, Z.; Liu, X.; Ma, S.; Nepal, S.; Deng, R. Boosting privately: Privacy-preserving federated extreme boosting for mobile crowdsensing. *arXiv* **2019**, arXiv:1907.10218.
34. Wu, Y.; Cai, S.; Xiao, X.; Chen, G.; Ooi, B.C. Privacy preserving vertical federated learning for tree-based models. *arXiv* **2020**. arXiv:2008.06170
35. Zhao, L.; Ni, L.; Hu, S.; Chen, Y.; Zhou, P.; Xiao, F.; Wu, L. Inprivate digging: Enabling tree-based distributed data mining with differential privacy. In Proceedings of the INFOCOM, Honolulu, HI, USA, 15–19 April 2018; pp. 2087–2095.
36. Yamamoto, F.; Ozawa, S.; Wang, L. eFL-Boost: Efficient Federated Learning for Gradient Boosting Decision Trees. *IEEE Access* **2022**, *10*, 43954–43963. [CrossRef]
37. Liu, Y.; Liu, Y.; Liu, Z.; Liang, Y.; Meng, C.; Zhang, J.; Zheng, Y. Federated forest. *IEEE Trans. Big Data* **2020**, *8*, 843–854. [CrossRef]
38. Maddock, S.; Cormode, G.; Wang, T.; Maple, C.; Jha, S. Federated Boosted Decision Trees with Differential Privacy. In Proceedings of the CCS, Nagasaki, Japan, 30 May–2 June 2022; pp. 2249–2263.
39. Fu, F.; Shao, Y.; Yu, L.; Jiang, J.; Xue, H.; Tao, Y.; Cui, B. Vf2boost: Very fast vertical federated gradient boosting for cross-enterprise learning. In Proceedings of the SIGMOD, Xi'an, China, 20–25 June 2021; pp. 563–576.
40. Xu, Y.; Hu, X.; Wei, J.; Yang, H.; Li, K. VF-CART: A communication-efficient vertical federated framework for the CART algorithm. *Eur. J. Inform. Syst.* **2023**, *35*, 237–249. [CrossRef]
41. Gai, K.; Zhang, Y.; Qiu, M.; Thuraisingham, B. Blockchain-enabled service optimizations in supply chain digital twin. *IEEE Trans. Serv. Comput.* **2022**, *16*, 1673–1685. [CrossRef]
42. Xie, T.; Gai, K.; Zhu, L.; Guo, Y.; Choo, K. Cross-Chain-Based Trustworthy Node Identity Governance in Internet of Things. *IEEE Internet Things J.* **2023**, *10*, 21580–21594. [CrossRef]
43. Xie, T.; Gai, K.; Zhu, L. Wang, S.; Zhang, Z. RAC-Chain: An Asynchronous Consensus-based Cross-chain Approach to Scalable Blockchain for Metaverse. *Acm Trans. Multimed. Comput. Commun. Appl.* **2023**. [CrossRef]
44. Pelttari, H. Federated learning for mortality prediction in intensive care units *arXiv* **2022**, arXiv:2205.15104.

45. Yang, M.W.; Song, L.Q.; Xu, J.; Li, C.; Tan, G. The tradeoff between privacy and accuracy in anomaly detection using federated xgboost. *arXiv* **2019**, arXiv:1907.07157.
46. De Souza, L.A.C.; Rebello, G.A.F.; Camilo, G.F.; Guimarães, L.C.; Duarte, O.C.M. DFedForest: Decentralized federated forest. In Proceedings of the Blockchain, Rhodes, Greece, 2–6 November 2020; pp. 90–97.
47. Yamamoto, F.; Wang, L.; Ozawa, S. New approaches to federated XGBoost learning for privacy-preserving data analysis. In Proceedings of the NeurIPS, Bangkok, Thailand, 23–27 November 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 558–569.
48. Wang, Z.; Yang, Y.; Liu, Y.; Liu, X.; Gupta, B.B.; Ma, J. Cloud-based federated boosting for mobile crowdsensing. *arXiv* **2020**, arXiv:2005.05304.
49. Li, Q.; Wu, Z.; Wen, Z.; He, B. Privacy-preserving gradient boosting decision trees. In Proceedings of the AAAI, Austin, TX, USA, 7–12 February 2020; pp. 784–791.
50. Li, Q.; Wen, Z.; He, B. Practical federated gradient boosting decision trees. In Proceedings of the AAAI, Austin, TX, USA, 7–12 February 2020; pp. 4642–4649.
51. Chen, W.; Ma, G.; Fan, T.; Kang, Y.; Xu, Q.; Yang, Q. Secureboost+: A high performance gradient boosting tree framework for large scale vertical federated learning. *arXiv* **2021**, arXiv:2110.10927.
52. Law, A.; Leung, C.; Poddar, R.; Popa, R.A.; Shi, C.; Sima, O.; Yu, C.; Zhang, X.; Zheng, W. Secure collaborative training and inference for xgboost. In Proceedings of the PPMLP, New York, NY, USA, 9 November 2020; pp. 21–26.
53. Zhang, J.; Zhao, X.; Yuan, P. Federated security tree algorithm for user privacy protection. *J. Comput. Appl.* **2020**, *40*, 2980.
54. Le, N.K.; Liu, Y.; Nguyen, Q.M.; Liu, Q.; Liu, F.; Cai, Q.; Hirche, S. Fedxgboost: Privacy-preserving xgboost for federated learning. *arXiv* **2021**, arXiv:2106.10662.
55. Wang, R.; Ersoy, O.; Zhu, H.; Jin, Y.; Liang, K. Feverless: Fast and secure vertical federated learning based on xgboost for decentralized labels. *IEEE Trans. Big Data* **2022**, 1–19. [CrossRef]
56. Han, Y.; Du, P.; Yang, K. Fedgbf: An efficient vertical federated learning framework via gradient boosting and bagging. *arXiv* **2022**, arXiv:2204.00976.
57. Yao, H.; Wang, J.; Dai, P.; Bo, L.; Chen, Y. An efficient and robust system for vertically federated random forest. *arXiv* **2022**, arXiv:2201.10761.
58. Li, X.; Hu, Y.; Liu, W.; Feng, H.; Peng, L.; Hong, Y.; Ren, K.; Qin, Z. OpBoost: A vertical federated tree boosting framework based on order-preserving desensitization. *arXiv* **2022**, arXiv:2210.01318.
59. Zhao, J.; Zhu, H.; Xu, W.; Wang, F.; Lu, R.; Li, H. SGBoost: An Efficient and Privacy-Preserving Vertical Federated Tree Boosting Framework. *TIFS* **2022**, *18*, 1022–1036. [CrossRef]
60. Chen, H.; Li, H.; Wang, Y.; Hao, M.; Xu, G.; Zhang, T. PriVDT: An Efficient Two-Party Cryptographic Framework for Vertical Decision Trees. *TIFS* **2022**, *18*, 1006–1021. [CrossRef]
61. Zhang, X.; Mavromatics, A.; Vafeas, A.; Nejabati, R.; Simeonidou, D. Federated Feature Selection for Horizontal Federated Learning in IoT Networks. *IEEE Internet Things J.* **2023**, *10*, 10095–10112. [CrossRef]
62. Kwatra, S.; Torra, V. A k-anonymised federated learning framework with decision trees. In Proceedings of the DPM and CBT, Darmstadt, Germany, 4–8 October 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 106–120.
63. Kalloori, S.; Klingler, S. Cross-silo federated learning based decision trees. In Proceedings of the SAC, Brno, Czech Republic, 24–26 August 2022; pp. 1117–1124.
64. Xu, Y.; Lu, Z.; Gai, K.; Duan, Q.; Lin, J.; Wu, J.; Choo, K.R. Besifl: Blockchain empowered secure and incentive federated learning paradigm in iot. *IEEE Internet Things J.* **2021**, *10*, 6561–6573. [CrossRef]
65. Gai, K.; Tang, H.; Li, G.; Xie, T.; Wang, S.; Zhu, L.; Choo, K.R. Blockchain-based privacy-preserving positioning data sharing for IoT-enabled maritime transportation systems. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 2344–2358. [CrossRef]
66. Gai, K.; She, Y.; Zhu, L.; Choo, K.R.; Wan, Z. A blockchain-based access control scheme for zero trust cross-organizational data sharing. *Acm Trans. Internet Technol.* **2023**, *23*, 1–25. [CrossRef]
67. Gai, K.; Wu, Y.; Zhu, L.; Choo, K.R.; Xiao, B. Blockchain-enabled trustworthy group communications in UAV networks. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 4118–4130. [CrossRef]
68. Peng, Z.; Xu, J.; Chu, X.; Gao, S.; Yao, Y.; Gu, R.; Tang, Y. Vfchain: Enabling verifiable and auditable federated learning via blockchain systems. *IEEE Trans. Netw. Sci. Eng.* **2021**, *9*, 173–186. [CrossRef]
69. Zhu, L.; Liu, Z.; Han, S. Deep leakage from gradients. In Proceedings of the NeurIPS, Vancouver, Canada, 8–14 December 2019; pp. 14774–14784.
70. Yin, H.; Mallya, A.; Vahdat, A.; Alvarez, J.M.; Kautz, J.; Molchanov, P. See through gradients: Image batch recovery via gradinversion. In Proceedings of the CVPR, Nashville, TN, USA, 19–25 June 2021; pp. 16337–16346.
71. Fu, C.; Zhang, X.; Ji, S.; Chen, J.; Wu, J.; Guo, S.; Zhou, J.; Liu, A.; Wang, T. Label inference attacks against vertical federated learning. In Proceedings of the USENIX Security, Boston, MA, USA, 10–12 August 2022; pp. 1397–1414.
72. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the ICCV, Venice, Italy, 22–29 October 2017; pp. 618–626.
73. Bagdasaryan, E.; Veit, A.; Hua, Y.; Estrin, D.; Shmatikov, V. How to backdoor federated learning. In Proceedings of the AISTATS, PMLR, Palermo, Italy, 26–28 August 2020; pp. 2938–2948.
74. Xie, C.; Huang, K.; Chen, P.; Li, B. Dba: Distributed backdoor attacks against federated learning. In Proceedings of the ICLR, Addis Ababa, Ethiopia, 26–30 April 2020.

75. Andreina, S.; Marson, G.A.; Möllering, H.; Karame, G. Baffle: Backdoor detection via feedback-based federated learning. In Proceedings of the ICDCS, Washington, DC, USA, 7–10 July 2021; pp. 852–863.
76. Zhou, X.; Peng, B.; Li, Y.F.; Chen, Y.; Tang, H.; Wang, X. To Release or Not to Release: Evaluating Information Leaks in Aggregate Human-Genome Data. In Proceedings of the ESORICS, Athens, Greece, 20–22 September 2011; Springer: Berlin/Heidelberg, Germany, 2011; Volume 11, pp. 607–627.
77. Weng, H.; Zhang, J.; Xue, F.; Wei, T.; Ji, S.; Zong, Z. Privacy leakage of real-world vertical federated learning. *arXiv* **2020**, arXiv:2011.09290.
78. Ribeiro, M.T.; Singh, S.; Guestrin, C. Why should i trust you? Explaining the predictions of any classifier. In Proceedings of the SIGKDD, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144.
79. Blanchard, P.; Mahdi, E.; Guerraoui, R.; Stainer, J. Machine learning with adversaries: Byzantine tolerant gradient descent. *NeurIPS* **2017**, *30*, 118–128.
80. Taheri, R.; Javidan, R.; Shojafar, M.; Pooranian, Z.; Miri, A.; Conti, M. On defending against label flipping attacks on malware detection systems. *Neural Comput. Appl.* **2020**, *32*, 14781–14800. [CrossRef]
81. Xia, Q.; Tao, Z.; Hao, Z.; Li, Q. FABA: An algorithm for fast aggregation against byzantine attacks in distributed neural networks. In Proceedings of the IJCAI, Macao, China, 10–16 August 2019.
82. Xie, C.; Koyejo, S.; Gupta, I. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In Proceedings of the ICML, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6893–6901.
83. Li, L.; Xu, W.; Chen, T.; Giannakis, G.B.; Ling, Q. RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In Proceedings of the AAAI, Honolulu, HI, USA, 29–31 January 2019; Volume 33, pp. 1544–1551.
84. Yang, S.; Ren, B.; Zhou, X.; Liu, L. Parallel distributed logistic regression for vertical federated learning without third-party coordinator. *arXiv* **2019**, arXiv:1911.09824.
85. Zhang, Y.; Zhu, H. Additively homomorphical encryption based deep neural network for asymmetrically collaborative machine learning. *arXiv* **2020**, arXiv:2007.06849.
86. Paillier, P. Public-key cryptosystems based on composite degree residuosity classes. In Proceedings of the EUROCRYPT, Prague, Czech Republic, 2–6 May 1999; Springer: Berlin/Heidelberg, Germany, 1999; pp. 223–238.
87. Goldreich, O. Secure multi-party computation. In *Manuscript Preliminary Version*; Citeseer: University Park, PA, USA, 1998; Volume 78.
88. Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H.B.; Patel, S.; Ramage, D.; Segal, A.; Seth, K. Practical secure aggregation for federated learning on user-held data. *arXiv* **2016**, arXiv:1611.04482.
89. Fredrikson, M.; Jha, S.; Ristenpart, T. Model inversion attacks that exploit confidence information and basic countermeasures. In Proceedings of the CCS, Denver Colorado, USA, 12–16 October 2015; pp. 1322–1333.
90. Shokri, R.; Stronati, M.; Song, C.; Shmatikov, V. Membership inference attacks against machine learning models. In Proceedings of the SP, San Jose, CA, USA, 22–24 May 2017; pp. 3–18.
91. Mohassel, P.; Zhang, Y. Secureml: A system for scalable privacy-preserving machine learning. In Proceedings of the SP, San Jose, CA, USA, 22–24 May 2017; pp. 19–38.
92. Papernot, N.; Abadi, M.; Erlingsson, U.; Goodfellow, I.; Talwar, K. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv* **2016**, arXiv:1610.05755.
93. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A. Calibrating noise to sensitivity in private data analysis. In Proceedings of the TCC, New York, NY, USA, 4–7 March 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 265–284.
94. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep learning with differential privacy. In Proceedings of the CCS, Vienna, Austria, 25–27 October 2016; pp. 308–318.
95. Dey, A.K.; Martin, C.F.; Ruymgaart, F.H. Input recovery from noisy output data, using regularized inversion of the Laplace transform. *IEEE Trans. Inf. Theory* **1998**, *44*, 1125–1130. [CrossRef]
96. McHutchon, A.; Rasmussen, C. Gaussian process training with input noise. *NeurIPS* **2011**, *24*, 1341–1349.
97. Awan, J.; Kenney, A.; Reimherr, M.; Slavković, A. Benefits and pitfalls of the exponential mechanism with applications to hilbert spaces and functional pca. In Proceedings of the ICML, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 374–384.
98. Liu, X.; Li, Q.; Li, T.; Chen, D. Differentially private classification with decision tree ensemble. *Appl. Soft Comput.* **2018**, *62*, 807–816. [CrossRef]
99. Xiang, T.; Li, Y.; Li, X.; Zhong, S.; Yu, S. Collaborative ensemble learning under differential privacy. In Proceedings of the WI, Santiago, Chile, 7 November 2018; pp. 73–87.
100. Fletcher, S.; Islam, M.Z. A Differentially Private Decision Forest. *AusDM* **2015**, *15*, 99–108.
101. Yang, S.; Li, N.; Sun, D.; Du, Q.; Liu, W. A differential privacy preserving algorithm for greedy decision tree. In Proceedings of the ICBASE, IEEE, Zhuhai, China, 24–26 September 2021; pp. 229–237.
102. Mironov, I. Rényi differential privacy. In Proceedings of the CSF, IEEE, Santa Barbara, CA, USA, 21–25 August 2017; pp. 263–275.
103. Shi, L.; Shu, J.; Zhang, W.; Liu, Y. HFL-DP: Hierarchical federated learning with differential privacy. In Proceedings of the GLOBECOM, IEEE, Madrid, Spain, 7–11 December 2021; pp. 1–7.
104. Wu, Z.; Li, Q.; He, B. Practical vertical federated learning with unsupervised representation learning. *TBD arXiv* **2022**, arXiv:2208.10278.

105. Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H.B.; Patel, S.; Ramage, D.; Segal, A.; Seth, K. Practical secure aggregation for privacy-preserving machine learning. In Proceedings of the CCS, Dallas, TX, USA, 31 October 2017; pp. 1175–1191.
106. Bittau, A.; Erlingsson, U.; Maniatis, P.; Mironov, I.; Raghunathan, A.; Lie, D.; Rudominer, M.; Kode, U.; Tinnes, J.; Seefeld, B. Prochlo: Strong privacy for analytics in the crowd. In Proceedings of the SOSP, Shanghai, China, 29–31 October 2017; pp. 441–459.
107. Erlingsson, U.; Feldman, V.; Mironov, I.; Raghunathan, A.; Song, S.; Talwar, K.; Thakurta, A. Encode, shuffle, analyze privacy revisited: Formalizations and empirical evaluation. *arXiv* **2020**, arXiv:2001.03618.
108. Sun, L.; Qian, J.; Chen, X.; Yu, P.S. Ldp-fl: Practical private aggregation in federated learning with local differential privacy. *arXiv* **2020**, arXiv:2007.15789.
109. Erlingsson, U.; Feldman, V.; Mironov, I.; Raghunathan, A.; Talwar, K.; Thakurta, A. Amplification by shuffling: From local to central differential privacy via anonymity. In Proceedings of the SODA, SIAM, San Diego, CA, USA, 6–9 January 2019; pp. 2468–2479.
110. Liu, R.; Cao, Y.; Chen, H.; Guo, R.; Yoshikawa, M. Flame: Differentially private federated learning in the shuffle model. In Proceedings of the AAAI, Virtual, 2–9 February 2021; Volume 35, pp. 8688–8696.
111. McMahan, H.B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the AISTATS, PMLR, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
112. Weinberg, A.I.; Last, M. Selecting a representative decision tree from an ensemble of decision-tree models for fast big data classification. *J. Big Data* **2019**, *6*, 1–17. [CrossRef]
113. Kwatra, S.; Torra, V. A Survey on Tree Aggregation. In Proceedings of the FUZZ-IEEE, IEEE, Luxembourg, Luxembourg, 11–14 July 2021; pp. 1–6.
114. Kargupta, H.; Park, B. A fourier spectrum-based approach to represent decision trees for mining data streams in mobile environments. *TKDE* **2004**, *16*, 216–229. [CrossRef]
115. Miglio, R.; Soffritti, G. The comparison between classification trees through proximity measures. *Comput. Stat. Data. An.* **2004**, *45*, 577–593. [CrossRef]
116. Caruana, R.; Niculescu-Mizil, A.; Crew, G.; Ksikes, A. Ensemble selection from libraries of models. In Proceedings of the ICML, Banff, AL, Canada, 4–8 July 2004; p. 18.
117. Tian, Y.; Feng, Y. Rase: Random subspace ensemble classification. *J. Mach. Learn. Res.* **2021**, *22*, 2019–2111.
118. Chen, M.; Shlezinger, N.; Poor, H.V.; Eldar, Y.C.; Cui, S. Communication-efficient federated learning. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2024789118. [CrossRef] [PubMed]
119. Chen, H.Y.; Chao, W.L. Fedbe: Making bayesian model ensemble applicable to federated learning. *arXiv* **2020**, arXiv:2009.01974.
120. Antunes, R.S.; da Costa, C.A.; Küderle, A. Federated learning for healthcare: Systematic review and architecture proposal. *ACM Trans. Intell. Syst. Technol.* **2022**, *13*, 1–23. [CrossRef]
121. Kasturi, A.; Ellore, A.R.; Hota, C. Fusion learning: A one shot federated learning. In Proceedings of the ICCS, Amsterdam, The Netherlands, 3–5 June 2020; Springer:Berlin/Heidelberg, Germany, 2020; pp. 424–436.
122. Li, M.; Chen, Y.; Wang, Y.; Pan, Y. Efficient asynchronous vertical federated learning via gradient prediction and double-end sparse compression. In Proceedings of the ICARCV, Shenzhen, China, 13–15 December 2020; pp. 291–296.
123. Chiti, F.; Fantacci, R.; Picano, B. A matching theory framework for tasks offloading in fog computing for IoT systems. *IEEE Internet Things J.* **2018**, *5*, 5089–5096. [CrossRef]
124. Arisdakessian, S.; Wahab, O.A.; Mourad, A.; Otrok, H.; Guizani, M. A survey on iot intrusion detection: Federated learning, game theory, social psychology and explainable ai as future directions. *IEEE Internet Things J.* **2022**, *10*, 4059–4092. [CrossRef]
125. Wehbi, O.; Arisdakessian, S.; Wahab, O.A.; Otrok, H.; Otoum, S.; Mourad, A.; Guizani, M. FedMint: Intelligent Bilateral Client Selection in Federated Learning with Newcomer IoT Devices. *IEEE Internet Things J.* **2023**, *10*, 20884–20898. [CrossRef]
126. Li, Y.; Feng, Y.; Qian, Q. FDPBoost: Federated differential privacy gradient boosting decision trees. *J. Inf. Secur. Appl.* **2023**, *74*, 103468. [CrossRef]
127. Hu, Y.; Zhang, Y.; Gong, D.; Sun, X. Multi-participant federated feature selection algorithm with particle swarm optimizaiton for imbalanced data under privacy protection. *IEEE Trans. Artif. Intell.* **2022**, *4*, 1002–1016. [CrossRef]
128. Courbariaux, M.; Bengio, Y.; David, J. Binaryconnect: Training deep neural networks with binary weights during propagations. *NeurIPS* **2015**, *28*, 3123–3131.
129. Devos, L.; Meert, W.; Davis, J. Fast gradient boosting decision trees with bit-level data structures. In Proceedings of the ECML-PKDD, 19–23 Spetember 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 590–606.
130. Shi, Y.; Ke, G.; Chen, Z.; Zheng, S.; Liu, T. Quantized Training of Gradient Boosting Decision Trees. *arXiv* **2022**, arXiv:2207.09682.
131. Fu, M.; Zhang, C.; Hu, C.; Wu, T.; Dong, J.; Zhu, L. Achieving Verifiable Decision Tree Prediction on Hybrid Blockchains. *Entropy* **2023**, *25*, 1058. [CrossRef] [PubMed]
132. Zhang, J.; Fang, Z.; Zhang, Y.; Song, D. Zero knowledge proofs for decision tree predictions and accuracy. In Proceedings of the CCS, Virtual Event, USA, 9–13 November 2020; pp. 2039–2053.
133. Wang, H.; Deng, Y.; Xie, X. Public Verifiable Private Decision Tree Prediction. In Proceedings of the Inscrypt, Guangzhou, China, 11–14 December 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 247–256.
134. Wen, H.; Fang, J.; Wu, J.; Zheng, Z. Transaction-based hidden strategies against general phishing detection framework on ethereum. In Proceedings of the ISCAS, Daegu, Republic of Korea, 22–28 May 2021; pp. 1–5.

135. Joshi, K.; Bhatt, C.; Shah, K.; Parmar, D.; Corchado, J.M.; Bruno, A.; Mazzeo, P.L. Machine-learning techniques for predicting phishing attacks in blockchain networks: A comparative study. *Algorithms* **2023**, *16*, 366. [CrossRef]
136. Ali, M.N.; Imran, M.; din, M.S.U.; Kim, B.S. Low rate DDoS detection using weighted federated learning in SDN control plane in IoT network. *Appl. Sci.* **2023**, *13*, 1431. [CrossRef]
137. Kazmi, S.H.A.; Qamar, F.; Hassan, B.; Nisar, K.; Chowdhry, B.S. Survey on joint paradigm of 5G and SDN emerging mobile technologies: Architecture, security, challenges and research directions. *Wirel. Pers Commun* **2023**, *130*, 2753–2800. [CrossRef]