symmetry

MDPI

Article

Applying Federated Learning in Software-Defined Networks: A Survey

Xiaohang Ma [1,†], Lingxia Liao [1,†], Zhi Li [1], Roy Xiaorong Lai [2] and Miao Zhang [3,*]

[1] School of Electronic Information and Automation, Guilin University of Aerospace Technology, Guilin 541004, China; liulianlian@guat.edu.cn (X.M.); liaolx@guat.edu.cn (L.L.); cczhili@guat.edu.cn (Z.L.)
[2] Confederal Networks Inc., Seattle, WA 98055, USA; Roy.lai@confederal.net
[3] Software College, Quanzhou University of Information Engineering, Quanzhou 510006, China
* Correspondence: zm@qziedu.cn; Tel.: +86-186-0108-5488
† These authors contributed equally to this work.

Abstract: Federated learning (FL) is a type of distributed machine learning approacs that trains global models through the collaboration of participants. It protects data privacy as participants only contribute local models instead of sharing private local data. However, the performance of FL highly relies on the number of participants and their contributions. When applying FL over conventional computer networks, attracting more participants, encouraging participants to contribute more local resources, and enabling efficient and effective collaboration among participants become very challenging. As software-defined networks (SDNs) enable open and flexible networking architecture with separate control and data planes, SDNs provide standardized protocols and specifications to enable fine-grained collaborations among devices. Applying FL approaches over SDNs can take use such advantages to address challenges. A SDN control plane can have multiple controllers organized in layers; the controllers in the lower layer can be placed in the network edge to deal with the asymmetries in the attached switches and hosts, and the controller in the upper layer can supervise the whole network centrally and globally. Applying FL in SDNs with a layered-distributed control plane may be able to protect the data privacy of each participant while improving collaboration among participants to produce higher-quality models over asymmetric networks. Accordingly, this paper aims to make a comprehensive survey on the related mechanisms and solutions that enable FL in SDNs. It highlights three major challenges, an incentive mechanism, privacy and security, and model aggregation, which affect the quality and quantity of participants, the security and privacy in model transferring, and the performance of the global model, respectively. The state of the art in mechanisms and solutions that can be applied to address such challenges in the current literature are categorized based on the challenges they face, followed by suggestions of future research directions. To the best of our knowledge, this work is the first effort in surveying the state of the art in combining FL with SDNs.

Keywords: federated learning; software-defined network; incentive mechanism; privacy and security; aggregation

Citation: Ma, X.; Liao, L.; Li, Z.; Lai, R.X.; Zhang, M. Applying Federated Learning in Software-Defined Networks: A Survey. *Symmetry* **2022**, *14*, 195. https://doi.org/10.3390/sym14020195

Academic Editors: Kuo-Hui Yeh, Chien-Ming Chen and Wei-Che Chien

Received: 23 December 2021
Accepted: 14 January 2022
Published: 20 January 2022

1. Introduction

Machine learning (ML) refers to computational approaches using experience to accomplish tasks, such as performance improvement or making accurate predictions. Experience is typically the properties learned from data and made available for subsequent tasks. Therefore, ML approaches are typically data-driven and have attracted increasing attention as mass data's achievements grow, with the development of the internet of things (IoT), cloud computing, and big data [1]. ML approaches are typically centralized. As shown in Figure 1a, a central server is involved to collect data from clients and train a model centrally. The quantity and quality of the centralized training data play the major role in achieving a high-quality model for tasks of such centralized ML approaches.

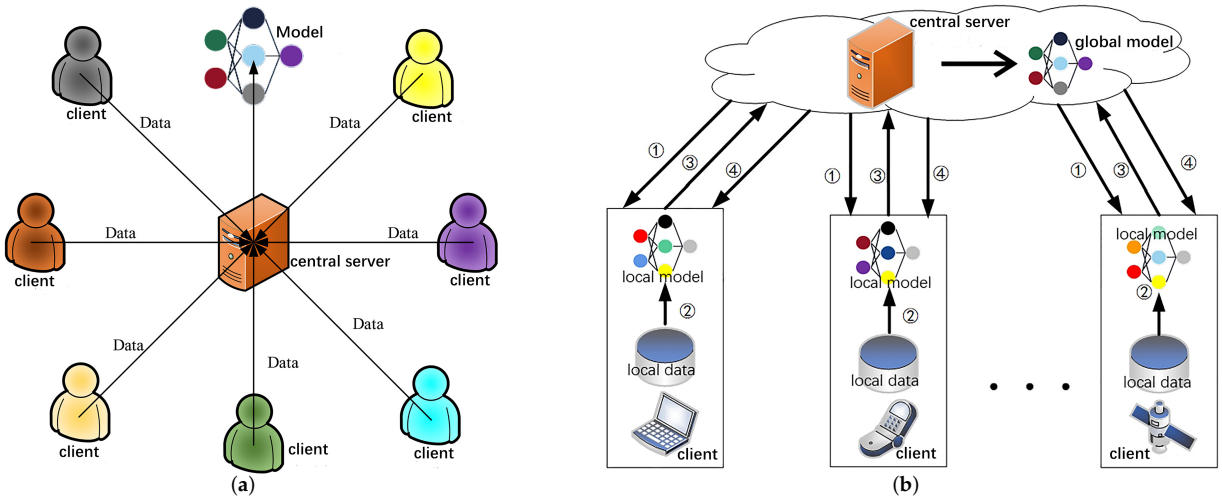

Figure 1. The comparison of traditional machine learning and federated learning: (1) training task distribution; (2) feeding local data; (3) local model transfer; (4) global model delivery. (**a**) Traditional machine learning approaches; (**b**) The architecture and workflow of FL approaches.

However, centralized training data are not always available. Clients, in many cases may not will to share their data due to privacy concerns. For instance, clinical institutions, patients, insurance companies, and pharmacies often do not want to contribute their healthcare data to train a high-quality model for a particular task [2,3]; banks, police offices, and supermarkets also may not want to centralize their data to build a models classifying people, predicting peoples' behaviors, or making recommendations [4]. Therefore, how to obtain a high quality and a sufficient quantity of training data have become issues that limit the further development of the centralized ML approaches.

Federated learning (FL) is a type of distributed ML approaches that do not rely on centralized training data and centralized model training [5]. As shown in Figure 1b, FL approaches also consist of a central server and multiple clients. Given a training task, a FL approach starts from the central server, passing the training task and broadcasting the current global model to each client. Then, the clients participating in FL replace their local models with the received global model, train new local models based on their local data, and upload their local models to the central server. Finally the central server applies aggregation algorithms based on the uploaded local models to produce a higher-quality global model. The central server and clients repeat the above steps until the termination requirements are met [6,7].

In general, FL approaches have three benefits: (1) data security and privacy is greatly improved without data sharing and centralized data storing [8]; (2) central servers do not need to store massive data, saving memory [9]; and (3) the computation pressure of central servers is reduced as clients in FL approaches undertake part of the model's production training [10]. Additionally, FL approaches are able to supply participants with a higher-quality model, although the volume of each local data set contributing to FL is limited [11].

Therefore, FL approaches can be widely used in many fields such as healthcare, smart-city sensing, and digital currency forecasting [12,13] due to its data privacy and security [14,15]. However, FL approaches rely on participants to contribute their local data and local models and cooperation between the central server and participants to produce a global model in a distributed manner. Since conventional networks consist of various devices provided by a huge range of providers, no protocols and specifications are standardized to facilitate such cooperation, leading to significant difficulty for FL approaches in attracting the participation of high-quality clients, detecting intrusion attacks during the process of FL, and dealing with abnormal participant devices. Therefore, FL

approaches have not been widely deployed in practice, although they have been a hot topic in academic research.

Software-defined networks (SDNs) provide a novel networking structure that aims to address the collaboration of network devices. A SDN architecture typically consists of application, control, and data planes and was firstly proposed by Cleanslate Research Group at Stanford University [16]. It decouples the control function from the forwarding function [17]. The control functions are centralized in SDN controllers located in the control plane, while the forwarding functions are performed by SDN switches located in the data plane. SDN architecture provides open and standardized protocols and specifications to enable the collaboration among controllers and between controllers and SDN switches. Regarding the possible primitives for FL approaches, SDN controllers can collect data from SDN switches, program the behaviors of SDN switches, and enforce global policies on SDN switches using open and standardized protocols; SDN controllers can be layered, organized with multiple local controllers in the lower layer and one root controller in the upper layer. The local controllers in the lower layer can be placed in the network edge to efficiently gather local data and local models, and the root controller in the upper layer can act as the central server to aggregate local models and produce a global model through standardized interfaces. Such SDNs with a distributed control plane allow FL approaches to be applied in a more scalable manner. Distributed control planes can also be used to cope with the symmetry and asymmetry in network topology, user distribution, and data storage. Therefore, SDN controllers can manage the client's participation willingness, offer solutions to detect intrusion, handle abnormal devices [18], and balance the data privacy and resource limitations of local model training for each participant in a given FL approach.

Applying FL in SDNs is an interdisciplinary research domain that may produce FL approaches with high performance in communication and management [19]. However, they have not been fully investigated in current research, and some challenges may be faced in restricting thier further development. Aiming to provide secure, efficient, and high-performance FL over SDNs, this paper comprehensively reviews the related mechanisms provided by both FL and SDNs, and discusses the possibilities that combine FL with SDNs to achieve this goal. We summarize three major challenges, including an incentive mechanism for participants, privacy and security strategies for communication, and aggregation methods for global model generation. We categorize the approaches proposed in current research and analyze them based on their challenges.

To the best of our knowledge, this paper is the first effort that comprehensively reviews the possible mechanisms and approaches that may enable FL over SDNs, although FL has been applied in many areas and surveyed for various research purposes. The major contributions of this paper are three-fold:

- We emphasize the importance of combining FL with SDNs as an important paradigm shift towards enabling collaborative ML model training with better data privacy and collaboration. We highlight the unique features of FL and SDNs, and discuss how they can be combined with each other. We discuss SDNs with a distributed control plane, consisting of multiple layered organized controllers, enabling scalable FL approaches with the combination of edge computing and cloud computing.
- We introduce three major challenges consisting of the incentive mechanism for participants, the privacy and security strategies for parameter communication, and aggregation algorithms for the generation of high performance global models. For each challenge, we present readers a comprehensive analysis of existing mechanisms, approaches, and solutions explored in current research.
- Future research issues, such as the evaluation of participants, anomaly detection, and the scalability of FL are also suggested and discussed.

The rest of this paper is organized as follows. While Section 2 presents the related work, Sections 3 and 4 provide the basic concepts and major components of FL and SDN, respectively. Section 5 highlights three major challenges when applying FL over SDNs. As Sections 6–8 provide the possible solutions proposed in current research to address the

challenges, Section 9 suggests some future research directions followed by the conclusion drawn in Section 10.

2. Related Work

FL approaches have been surveyed in many popular research scenarios, as listed in Table 1. In regard to security issues, Lyu et al. [20] summarized the potential threats and defenses, and Yin et al. [21] and Mothukuri et al. [22] offered comprehensive surveys on data privacy and security. While Khan et al. [23] provided a comprehensive survey on FL approaches, applications, and challenges over IoT, Pham et al. presented an exhaustive review on FL over industrial IoT (IIoT) [24], Imteaj et al. [25] surveyed the existing problems and solutions considering resource-constrained IoT devices. For other areas, Gadekallu et al. [26] filled the vacancy in combining big data with FL; Jiang et al. [4] and Xu et al. [2] analyzed the challenges and potentials of FL in smart cities and healthcare, respectively; Xia et al. [27] and Nguyen et al. [28] focused on edge computing and summarized the security and privacy issues when combining FL with edge computing. Applying FL in 6G and wireless networks were also surveyed by references [29–31]. However, no surveys have been made, to the best of our knowledge, in the scenario of applying FL over SDNs, and this survey fills such gap.

Table 1. The existing surveys of federated learning.

Scenario	Reference	Year	Objective
IoT	[23,25]	2021	security, resources, incentive
IIoT	[24]	2021	privacy, resource, data management
big data	[26]	2021	privacy and security, communication
smart city	[4]	2020	
healthcare	[2]	2021	
edge computing	[27,28]	2019-2021	communication, security, resources
6G	[29,30]	2020–2021	communication, security
wireless network	[31]	2020	computing, spectrum management
Others	[20–22]	2020–2021	threats and defense
	[32]	2021	communication
	[33]	2021	incentive mechanisms

3. Federated Learning

This section introduces the basic concepts and components of FL approaches. Three types of FL approaches are categorized based on the data that train their models.

3.1. Basic Concepts and Components

FL refers to distributed ML approaches that solve the trouble of data silos and privacy. FL approaches typically consist of one central server and multiple clients, as shown in Figure 1b. FL approaches generate models in an iterative manner. Given a training task, a FL approach typically contains the following four steps.

- Initializing the training task. The central server passes the training task and broadcasts the current global model to each client, as shown by (1) in Figure 1b.

- Local model training. Clients replace their local models with the received global model and continue to train new local models based on their local data, as shown by (2) in Figure 1b The training process continues until the number of iterations reaches the maximum.
- Local model uploading. Clients upload their local models to the central server, as shown by (3) in Figure 1b.
- Aggregating local models. The central server applies aggregation algorithms to generate a global model based on the uploaded local models, as shown by (4) in Figure 1b.

The central server and clients repeat the above four steps until the termination conditions are fulfilled. As clients only need to upload their local model instead of sharing their local data during the training process of FL, FL ensures their data privacy and security. As each client only owns a very limited volume of data, the global model, aggregated by the central server and based on all the local models trained by clients, can achieve a better quality than the local models. By participating in the FL, clients gain a global model with better general accuracy than their local ones.

3.2. Classification of FL

Although criteria such as network topology and ML models can be applied to categorize FL approaches, this paper categorizes the existing FL approaches according to the partitioning of data used to train local models. Such a criterion has been widely used in current research.

The partitioning of data can be done based on sample space and feature space. While sample space refers to the name or identity of samples in the data set, feature space consists of all the features provided by the samples. Given three samples, A, B, and C, where sample A has features of weight, height, and birth date; sample B has features of age and education; and sample C has features of weight, height, and income per year. Then the samples of A, B, and C form the sample space, and the features of weight, height, birth date, age, education, and income per year construct the feature space. When multiple clients participate in FL, each client contributes its local data to training a local model. Since each local data set may have various sample space and feature space, FL approaches can be categorized into horizontal FL, vertical FL, and federated transfer learning.

- Horizontal federated learning. Horizontal FL approaches refer to the approaches that have data sets with numerous features overlapped but few similar users. As illustrated by Figure 2a, horizontal FL approaches extract the same features from different users in the original data set to train local models and aggregate global models in FL. For instance, we might apply a horizontal FL approach in two real banks. Each bank has various customers because they are located in two different cities. However, the types of information provided by each customer are similar, as the services provided by each bank are similar. This demonstrates a use-case in which the data shares the same feature space but varies in sample space. It meets the requirements for horizontal FL approaches.
  Many currently proposed FL approaches fall into the category of horizontal FL, and the most famous one among them was proposed by Google, wherein a global model was aggregated with the help of Android mobile phones that collected information from different clients [34]. Horizontal FL was also used in the healthcare domain [2]. Our previous work also applied horizontal FL to classify elephant flows over IoTs [35].
- Vertical federated learning. Contrary to horizontal FL, vertical FL is suitable for scenes where data sets have the same users but with few user features in common, as presented in Figure 2b. An intermediate third party is meaningful for the participants of vertical FL, as it can offer authentication and encryption for the training tasks. Consider applying vertical FL to a supermarket and a bank in the same region. Although the majority of users of the supermarket and the bank are the same due to the same region in which they are located, such users have widely various features because

supermarkets and banks provide completely different services to customers. However, such two completely different institutions can contribute their data to produce a global model in a vertical manner, though they are prohibited from communicating and do not know what information each other provided. Cheng et al. [36] studied vertical FL and proposed a privacy-preserving system called SecureBoost. This system compiled information of multiple groups with common user samples but different feature sets to enhance model training. However, Yang et al. [37] reported their belief that third parties are not necessary.

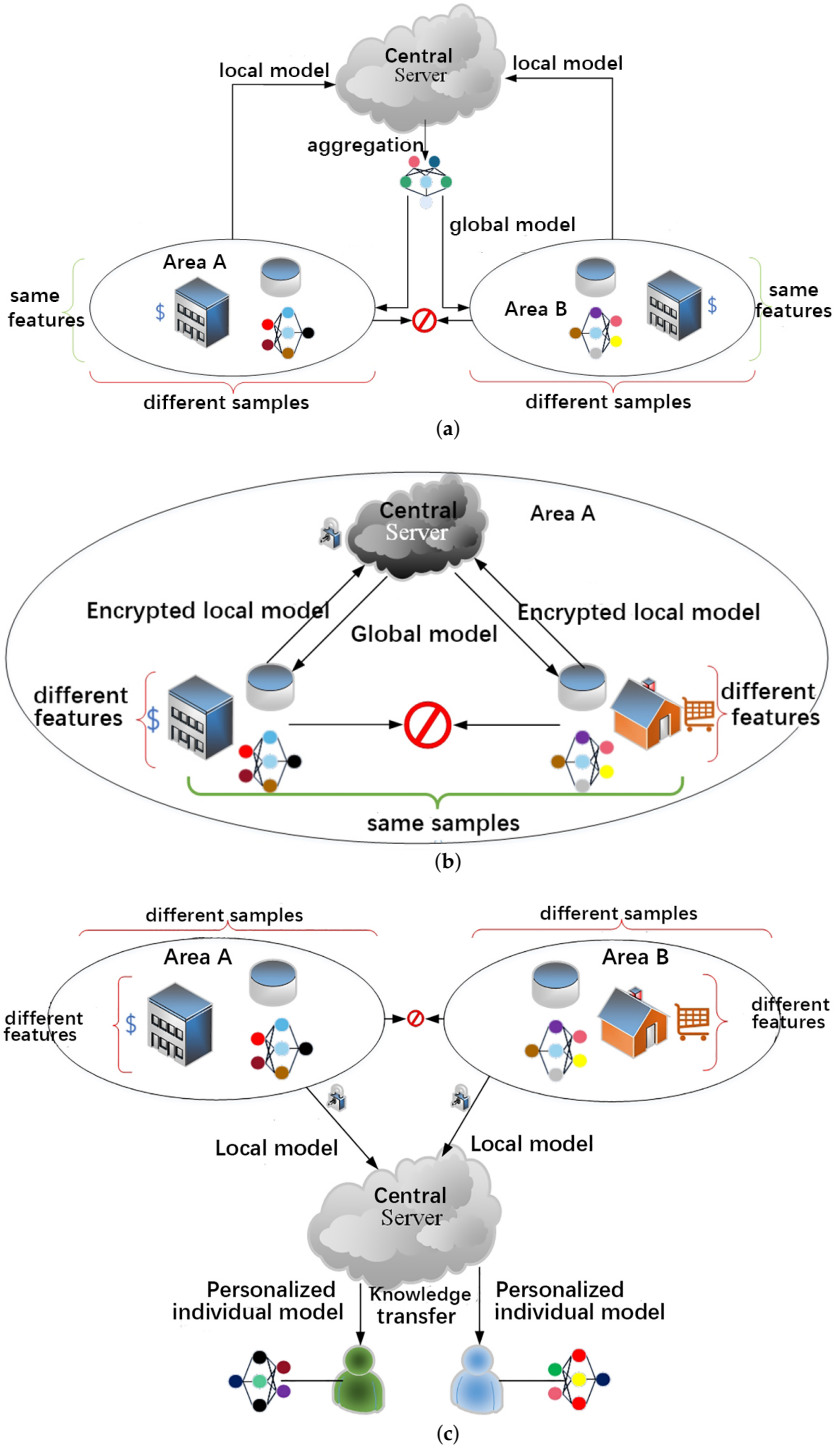

Figure 2. The structure of horizontal FL, vertical FL, and federated transfer learning. (**a**) Horizontal federated learning; (**b**) Vertical federated learning; (**c**) Federated transfer learning.

- Federated transfer learning. When two data sets have very little overlap between the user features and the user samples, the training task can neither meet the requirements of horizontal FL or vertical FL. In this case, federated transfer learning can be involved. As illustrated in Figure 2c, given a supermarket in city A and a bank in city B, their customers and services have almost no overlap. Federated transfer learning will find similarities between these two different data sets when they participate in the model training. References [38,39] studied federated transfer learning.

4. Software-Defined Networks

As illustrated in Figure 3, SDNs provide a novel networking structure that is composed of application, control, and data planes [40]. While the application plane covers a large number of network applications serving users, the control plane hosts SDN controllers coordinating and managing the devices and flows all over the entire network, and the data plane consisting of underlying network devices such as SDN switches and hosts. SDN switches are dumb devices that have no control function. They rely on SDN controllers in the control plane to generate forwarding rules. Accordingly, SDN controllers completely decouple control functions from packet forwarding, and can fully control the behaviors of SDN switches and flows all over the entire network.

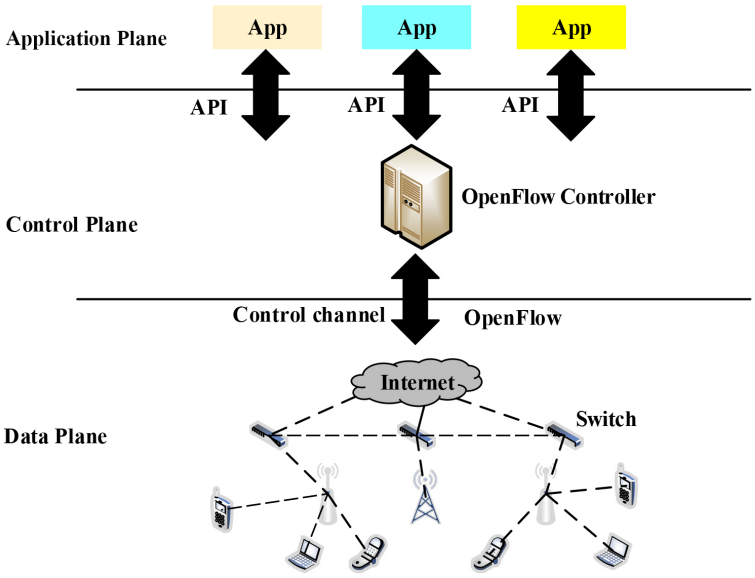

Figure 3. Architecture of software-defined networks.

SDN architecture also defines three types of interfaces: the southbound, northbound, and west/eastbound interfaces. While the southbound interface lies between the control and data planes, the northbound interface is between the control and application planes, and the west/eastbound interface is inside the control plane. The southbound interface is for the control plane's management of flows and configuration of switches in the data plane. The northbound interface is for applications' retrieval of information from the global view maintained by controllers and conveyance of their desired configurations and behaviors to the data plane. The west/eastbound interface enables compatibility and interoperability among different controllers. SDNs have potential in many fields due to their openness, flexibility, and programmability.

SDNs are able to effectively overcome some difficulties of FL including participants, time efficiency, and management. In many networking scenarios, FL may face the challenge of insufficient participants by default if there is inadequate remuneration. In SDNs, SDN controllers can be super participants of FL since they have built-in mechanisms to collect data from clients and much more computation and storage resources to handle data than do clients. Regarding network intrusions that terribly impact the global model and data

privacy of clients in FL, traditional networks do not provide mechanisms to coordinate the clients and central server to detect intrusions timely. SDNs have controllers and switches cooperating, using standardized interfaces, overcoming the abnormalities of detection and defense more efficiently and effectively. Additionally, the central server in FL over traditional networks does not have the capability to supervise clients. However, SDNs controllers can supervise SDN switches all over the network, and one SDN controller can supervise others using open and standardized interfaces if multiple controllers are included in the control plane.

5. Federal Learning in Software-Defined Networks

This section discusses the combination of FL and SDNs. Three challenges are highlighted.

5.1. Combination of FL and SDNs

SDNs may have a centralized control plane and a distributed control plane [41]. While a centralized control plane consists of only one controller that supervises all the devices in the data plane, adistributed control plane consists of multiple controllers, each managing a subset of controllers over the network. The controllers in a distributed control plane are often organized in layers organized, as illustrated in Figure 4. A layered distributed control plane typically consists of a root controller in the top layer and multiple local controllers in the lower layer. Layered-distributed control planes can greatly improve the computation and scalability of SDNs. When applying FL in such SDNs, the local controllers are the participants that play the major role in collecting local data and training local models, and the root controller acts as the central server that takes the responsibility for aggravating global models. More importantly, controllers have stronger computation and storage capability than switches and hosts, which improves the time efficiency of both local and global models in the training process of FL.

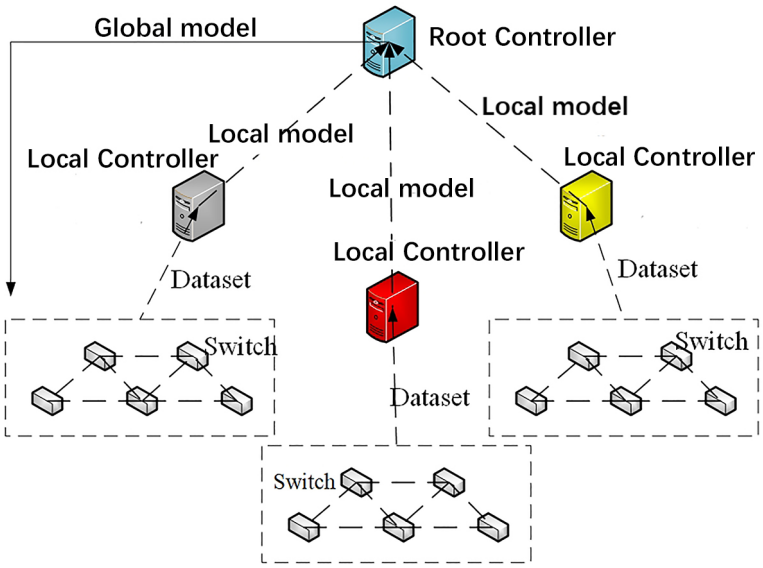

Figure 4. SDN with a distributed control plane.

5.2. Challenges in Applying FL in SDNs

The combination of FL and SDN generates a huge advantage in coordinating central servers with participants. However, it still faces many challenges. The rest of this section summarizes the three major challenges that should be addressed in combining FL with SDNs.

- Incentive mechanisms. Incentive mechanisms are used to encourage clients to participate in the training of FL and to make more contributions to global model aggregation. Although FL over SDNs can force all the controllers in the SDN control plane to be

participants, incentive mechanisms may be involved to distribute rewards based on the contribution of participants, so that clients may be willing to contribute more and higher-quality data and resources to train local models and to improve the global model. Additionally, incentive mechanisms allow the central server and participants to balance the resources competed for by the FL and the other tasks running on SDN controllers.

- Security and privacy. Security and privacy have always been the key issues in the research of FL. Although the local controllers, as the participants in FL approaches, only need to transfer local models instead of local data, there is still a risk of privacy leakage. When applying FL in SDNs, the openness and programmability of SDNs, on the one hand, make FL more flexible and manageable; on the other hand, they also increase the security and privacy risks. In particular, the hidden security troubles in SDN architecture, controllers, and interfaces may completely destroy the aggregation, the related models, and the data.
- Global model aggregation. The performance of FL approaches is directly affected by the aggregation mechanisms involved. Although SDN controllers can use = builtin protocols and interfaces to alleviate this problem, such mechanisms' speed and quality are the major concerns when deploying FL in SDNs.

5.3. Dealing with Asymmetry in FL and SDNs

Controllers in a layered, distributed control plane are often symmetrically organized. Given a data plane with a huge number SDN switches, the simplest way to manage it is to deploy a layered, distributed control plane consisting of multiple controllers, each of which manages a partition of the data plane to form a symmetrical structure. However, each network partition may present an asymmetry due to the number of distributed users, the quantity of switches attached, and the volume of user data stored. Such asymmetries can cause huge difficulties in selecting participants, designing incentive mechanisms, and developing aggregation algorithms for FL approaches. However, since SDN architecture allows controllers to cooperate with each others and to manage the behavior of switches and flows using open and standardized protocols and interfaces, applying FL in SDNs provides more builtin primitives to overcome these difficulties.

6. Incentive Mechanisms

In this section, we summarize the currently proposed incentive mechanisms and further categorize the mechanisms into game theory, contract theory, auction theory, and fairness theory. The major relevant solutions are listed in Table 2.

6.1. Game Theory

Game theory is an important branch of economics. It studies the strategies of multiple individuals or groups in a game. Game theory is widely used to generate incentive mechanisms in many fields such as edge computing [42], device communication [43], and mobile systems [44]. It also has been used to generate incentive mechanisms in FL. We summarize the major models based on game theory for incentive systems in FL approaches.

- The Stackelberg model is a classical game theory model that stipulates a leader and followers. In FL, the central server is the leader, and the clients are the followers. The followers may report their available resource and unit price to the leader, and the central server gives each follower unequal returns based on its budget to achieve the goal that minimizes or maximizes a particular objective. The leader and followers can be defined variously. While Sarikaya et al. [45] defined the model owner (server) as the leader and the workers (mobile devices) as the followers to improve FL efficiency; Khan et al. [46] set the base station as the leader and the participating users as followers to improve the accuracy of the global model; Feng et al. [47] specified the mobile devices as leaders and the model owner as the follower to determine the training data sizes for leaders and followers when applying FL in a communication

framework; and Sun et al. [48] applied the Stackelberg model to adjust the number and the level of customers participating in the global model aggregation in FL. Although many researchers believed that the Stackelberg game existed between leaders and followers, Pandey et al. [49], Xiao et al. [50], and Hu et al. [51] reported their belief that the Stackelberg game should also exist between the followers. They divided the Stackelberg game into two stages: the leader's releasing rewards stage and the followers' maximizing their benefits stage. Zhan et al. [52] introduced deep reinforcement learning based on the two-stage Stackelberg model to determine the optimal pricing and training strategies for the central server and the participants (edge nodes), respectively. However, many researchers believed that it was difficult to find the best strategy based on game theory alone due to the complex networking environment.

Table 2. The existing research on incentive mechanisms.

Approach		Reference	Objectives
Game	Stackelberg	[45,46,48–52]	Leader: central server Follower: participants
		[47]	Leader: mobile devices Follower: model owner
	auction	[53–57]	Buyer: central server Seller: clients
	NC game	[58–60]	stimulus sharing by trading
	Shapley	[61,62]	calculate contribution
	Nash equilibrium	[63,64]	collaboration strategy
contract theory		[65]	decentralized incentive
		[66,67]	based on data and work
		[68,69]	dynamic renewal contract
		[70–72]	multi-dimensional information
		[73]	value- driven incentive
		[74]	built on Ethereum
fairness theory		[75,76]	reputation incentive protocol
		[77–79]	reporting costs truthfully
		[76]	rewards and punishments

- Auction theory is a type of game theory that includes auctioneers and bidders. When applying auction theory to encourage clients to participate in FL, the auctioneers release the task, the bidders bid for the necessary cost in FL training, and the auctioneers determine the winners. Le et al. [53,54] let the base station play the role of auctioneer and the mobile clients act as bidders. The base station released the task and the mobile clients sent bids to the base station for the necessary cost. The base station determined the winner from the bidders based on the relevant algorithm and paid the

reward. Yu et al. [55] also attracted devices with high-quality data through dynamic payoff sharing based on auction theory to eliminate the mismatch between returns and contributions. However, bidding price is not the only element in auctions. More than 50% of results may be influenced by the other factors [80], and therefore, it is important to consider the multiple dimensions of the data. For instance, Zend et al. [56] brought a multi-dimensional incentive framework for FL to derive optimal strategies for edge devices, Jiao et al. [57] designed a reverse multi-dimension auction system to better evaluate the value of each participant.

- Non-cooperative game is a type of game in which individual players compete with each other. Applying non-cooperative gamed in FL implies the central server and clients do not cooperate. As Tang et al. [58] maximized the efficiency and balanced participants' situations and budgets through a non-cooperative game, Pan et al. [59] proposed applying a non-cooperative game to determine the best security strategy for nodes and to attract participants, and Chai et al. [60] involved a non-cooperative game in internet of vehicles (IoVs) networks based on blockchain.
- The Shapley value is used in cooperative game theory to fairly distribute both gains and costs to several actors working in coalition. Cooperative game theory involves a coalition for all players in the game. The gains and costs are computed by the coalition. While Song et al. [61] calculated the Shapley value of data providers in FL to determine their returns, Sim et al. [62] proposed a reward scheme that computed the Shapley value to weigh the contributions of devices to allot rewards.
- The Nash equilibrium is a fundamental game theory model introduced by John Nash [81]. In this model, none of the players can unilateterally change their strategy to increase their payoff. It is a cooperative model in which all strategies are mutual best responses to each other, and no player has any motivation to deviate unilaterally from the given strategy. Gupta et al. [63,64] applied such a game model to enable FD over IoT networks.

6.2. Contract Theory

Contract theory typically studies how economic actors construct contractual agreements. In FL, contract theory can be involved in designing incentive mechanisms that allow the participants to reach a consensus after evaluating themselves. Applying contract theory should consider the requirements of local model training, global model aggregation, and the payment to participants. In particular, the central server firstly customizes a set of contracts according to stipulated requirements, then the clients choose their preferred contract to sign, complete the local model training, and upload the local model. Finally, the server aggregates the received local models and pays the clients. As contract theory was used to attract devices with high-quality data to join the FL in [68], it was also used by [65,69] to encourage participants to increase their contributions for aggregation. Tian et al. [66] conducted a comprehensive survey on incentive mechanisms based on contract theory.

However, it is very challenging to develop incentive mechanisms, based only on contract theory, that achieve a high performance in a complex network environment. Therefore, contract theory is often combined with other mechanisms to address performance issues. For instance, the combination of contract theory and game theory was used by Lim et al. [67] to design a hierarchical incentive mechanism for FL; multi-dimensional contracts were used by Ding et al. [70,71] and Wu et al. [72] to effectively handle user information; blockchain technology, on the basis of contract theory, was also applied in [73,74] to customize all participants and to complete model aggregation. Penalties were issued to participants with poor performance.

6.3. Fairness Theory

Fairness is desired in incentive systems. Game and contract theories focus on the contributions that participants can provide in FL, and based on which the incentive systems

are likely to generate unfair rewards due to the differences between devices under various networking circumstances.

In order to minimize the unfairness and improve the overall social benefits, Qu et al. [77] proposed a fair incentive system such that clients could truthfully report their contributions and lossed in FL training. However, it is not practical to assume clients will be fully honest. Therefore, Wang et al. [78] computed client contributions by deleting models. It better reflected the value of local models in training a global model, and hence improving the fairness of the FL, although it decreased the time efficiency for model aggregation.

Fairness is often evaluated based on reputation. Incentive mechanisms based on reputation in FL typically estimate the customers' reputation value regarding their behaviors after the training is completed. Zhao et al. [75] proposed an incentive mechanism based on reputation for a blockchain-based FL. They initialized all participants with the same reputation and updated the reputation after each iteration. The participants that provided high-quality models had their reputations increased, or decreased otherwise. Rehman et al. [79] proposed a similar scheme. Their system included three types of participants: edge devices, fog nodes, and cloud servers. In the system proposed by Zhao et al. [75], the leader was generated randomly, and each participant could grade the others. Gao et al. [76] presented an incentive system that combined reputation and contribution to avoid unfairness.

7. Security and Privacy

Security and privacy has been extremely popular in the research of both FL and SDNs. The purpose of FL is to avoid direct access to participants' data that threaten privacy. However, some studies, such as [82,83], showed that inference attacks can infer private information from gradients and parameters uploaded by the clients (gradients and parameters are the major components in training and constructing local models), and free-riding attacks can result in malicious devices that always contribute meaningless models receiving a high-quality global model. Such security issues do not help to train a global model in FL, and instead they may significantly reduce the accuracy of the final model.

The rest of this section reviews the proposed approaches related to protecting clients' privacy and security in FL. According to the protection provided, we divide the approaches into three categories: information encryption, decentralized FL, and intrusion detection, as listed in Table 3.

7.1. Information Encryption

Encryption is a common method of enhancing data security and privacy. Although encryption has a profound impact on various fields, we summarize the three most common encryption methods in FL: differential privacy (DP), secure multi-party computing (SMC), and homomorphic encryption (HE).

- Differential privacy is a common encryption method that protects privacy by adding noise to sensitive data and parameters to cause deviations [84]. For instance, Wei et al. [85] added Gaussian noise and adjusted its variance to protect a global DP against FL differential attacks in a stochastic gradient descent; Geyer et al. [86] encrypted private information by adding Gaussian noise and proposed a differential privacy scheme to hide user contributions during FL training with less loss of model accuracy; and Triastcyn et al. [87] added noise after obtaining the privacy loss boundary determined by data distribution to provide a Bayesian differential privacy mechanism. Zhu et al. [88] studied hybrid DP and ML algorithms in FL.
  Some research showed that a trusted third party is needed to add noise to FL. This class of approach is called global differential privacy. Global differential privacy has the advantage that less noise is added and high accuracy global model can be achieved. It has been applied to FL in healthcare [89] and personal [90] scenarios.
  In contrast with global differential privacy, local differential privacy allows clients to encrypt data locally before uploading it. Cao et al. [91] reported their belief that

local differential privacy was more credible than global privacy and developed a protection system with local differential privacy that could add Gaussian noise before model upload to enhance the security of a power system. While Lu et al. [92] applied local differential privacy in asynchronous FL in the field of the IoVs, Zhao et al. [93] designed four local differential privacy mechanisms to disrupt the gradient for IoVs away from threats. Some researchers reported believing that added noise is directly influenced by the dimensionality of the data and that each dimension does not have the same importance. Therefore, Liu et al. [94] proposed a Top-k concept that selects the most crucial dimensions based on contributions, Truex et al. [95] applied local differential privacy based on data with compressed dimensions, and Sun et al. [96] designed a parameter shuffling system for multi-dimensional data to mitigate privacy degradation caused by local differential privacy.

- Secure multi-party computing (SMC) is a sub-field of cryptography and can be used to solve the problem of collaborative computing between a group of untrusted parties under the premise of protecting private information. The application of SMC in FL allows multiple participants to jointly calculate the value of an objective function without revealing their own input. Zhu et al. [97] analyzed the relationship between FL and SMC, and Bonawitz et al. [98] designed a secure communication protocol that could specify a constant number of iterations and tolerate a number of clients exiting the FL. Kanagavelu et al. [99] proposed a two-phase MPC-enabled FL framework, wherein participants elected a committee to offer SMC services to reduce the overhead of SMC and maintain the scalability of FL. In [100], Sotthiwat et al. suggested encrypting the key parameters of multi-dimensional gradients to take advantage of SMC encryption while limiting communication cost.
- Homomorphic encryption (HE) is a cryptographic technique based on the computational complexity of mathematical problems. Using HE in FL firstly requires clients to encrypt their local models or parameters before uploading them to the central server, then clients offer the central server a scheme for the encrypted data. After the central server aggregates the data using the provided scheme and broadcasts it to the clients, clients finally decrypt the received data to get the final model.
  Gao et al. [39] proposed an end-to-end privacy protection scheme using HE for federated transfer learning, Zhou et al. in [101] combined secret sharing and HE to protect privacy. While Zhang et al. [102] introduced HE in Bayesian approaches to protect the privacy of vertical FL, references [103,104] added a coordinator, apart from clients and the central server, and let the coordinator take responsibility for specifying the security protocol including keys and processing functions. Stripelis et al. [105] proposed a HE mechanism based on CKKS to reduce the difficulty of encryption and decryption. CKKS is an approximation HE scheme, proposed in 2017 [106]. Ma et al. [107] designed a multi-key HE protocol on the basis of CKKS to allow collaborating participants to encrypt data with different keys and decryption.
  HE is secure, but its communication loss is also significant. Many researchers believe that communication is a bottleneck restricting the further development of HE in FL. In order to reduce such communication costs, Zhang et al. [108] proposed encrypting a batch of quantitative gradients at one time instead of encrypting a single gradient with full precision. This idea could effectively reduce the total amount of ciphertext and communication loss. Jiang et al. [109] devised a symmetric key system that appropriately introduced sparse technology according to the encrypted object.
- Encryption combinations combine multiple encryption methods to further improve the security of FL. The studies in [110–112] designed encryption mechanisms combining DP and SMC. They allowed local clients to encrypt their local models or parameters through local differential privacy before uploading, and let the central server complete the encryption task of secure aggregation based on SMC. They found that the noise added by local differential privacy was determined by the local clients. However, the study in [113] put forward an encryption combination wherein the added noise

was generated by neighboring clients, and the local clients did not know the noise being added to their models. In addition to combining SMC and DP, the association of HE and DP is also an outstanding encryption approach. Hao et al. [114] proposed adding noise to each local gradient before encryption. A lightweight additive HE was used to reduce cost, and the DP was used to avoid the threat of collusion.

Table 3. The existing studies on security mechanisms.

Approach		Ref	Objectives
encryption	DP	[91–95]	local differential privacy
		[85–87,89,90]	global differential privacy
	SMC	[98,100]	high-dimensional information
		[99]	elect a committee
	HE	[39]	comparing with secret sharing
		[101,102]	Paillier HE
		[105]	constructed with CKKS
		[103,104]	third-party encryption
		[107]	Multiple key HE
		[108]	quantify and unify
		[109]	symmetric key
	DP+SMC	[110–113]	First stage: DP; Second stage: SMC
	DP+HE	[114]	First stage: DP; Second stage: DP
decentralized	blockchain	[115–118]	manage FL frameworks
		[119]	committee consensus
		[120]	incorporates re-encryption.
		[121]	based on deferential data.
	other	[36]	tree-boosting system.
		[122]	tree-based models
intrusion detection		[123]	multiple global models
		[124,125]	based on classical ML
		[126]	introduce teachers and students.
		[127–130]	deep learning detection
		[131,132]	edge devices collaboration
		[133]	decentralized asynchronous FL

7.2. Decentralized Federated Learning

The central server is necessary for classical FL approaches to publish tasks and aggregate models. However, a reliable and fair central server trusted by all clients is difficult to find, in reality. Moreover, the central server may collude with some clients to leak the privacy of other devices, thus, leading to decentralized FL approaches for privacy protection.

Blockchain is a popular technology in decentralized FL. Blockchain originated from Bitcoin. Blockchain has the characteristics of openness and transparency and can replace the third-party server in FL. Rahman et al. [115] presented a fully decentralized hybrid FL framework. They employed blockchain to manage the devices participating in FL and to protect clients from the threats of tampering and unauthorization. They also provided a framework supporting HE and SMC encryption. Majeed et al. [116] proposed an FL architecture based on blockchain. The proposed architecture could store the results of each iteration and prevent models from being poisoned by attacks. Li et al. [119] proposed a committee consensus mechanism in charge of model checking and aggregating to avoid the attacks of malicious nodes with slightly increased cost. Qi et al. [117] introduced blockchain-based FL in IoVs, where workers voted a miner to supervise the aggregation in each iteration. This miner had the ability to stop unreliable updates. Similarly, Li et al. [120], Mugunthan et al. [118], and Lu et al. [121] studied blockchain technology to protect privacy in FL. Decentralized FL also was also shown to be usable in improving digital currency price prediction in blockchain [12,13].

Current research also proposed some decentralized FL approaches that did not rely on blockchain. For instance, Cheng et al. [36] proposed a lossless tree-boosting system for vertical FL. It could protect the privacy of clients without reducing the accuracy of the model. Wu et al. [122] proposed a privacy protection mechanism based on decision trees. The mechanism included two encryption methods, HE and SMC. It could be extended to train random forest and gradient boosting decision tree models.

7.3. Intrusion Detection Mechanism

Intrusion detection is an effective measure to find threats in time and to improve system security. Intrusion detection typically relies on ML to train models for anomaly identification. However, the training of models generally requires a large amount of attacked data. Since FL can handle a large volume of data without data sharing, intrusion detection problems can be well addressed using FL approaches.

Li et al. [127] considered an industrial cyber–physical system covering multiple different networks and consisting of multiple intelligent subsystems including cloud computing and industrial control systems. They involved the deep learning of participants to train local intrusion detection models. The cloud server took charge of the aggregation to produce a global intrusion detection model. The global intrusion detection model was trained by a massive volume of attached data from distinct networks, and its efficiency in detecting anomalies was much higher than those local models trained by local data. In reference [128], Zhao et al. applied a long short-term memory (LSTM) to train intrusion detection models over a FL structure, while Schneble et al. [124] considered possible attacks, including denial of service, data modification, data injection, and eavesdropping, and proposed an intrusion detection system using FL for medical cybe–physical systems; Sun et al. [123] suggested a segmented FL intrusion detection approach that maintained multiple global models. Participants were evaluated after a certain number of rounds. While the participants with good performance could stay and train continuously, the ones with poor performance would be moved to other branch to initialize new, centralized FL training.

Liu et al. [131] proposed a collaborative intrusion detection approach that relied on decentralized FL. The approach relied on blockchain to improve the efficiency and security in intrusion detection model training. Ren et al. [125] replaced the central aggregation server with blockchain and designed an intrusion detection algorithm for lightweight network devices. They constructed an intrusion detection model with five support vector machines. Wittkopp et al. [126] introduced the teacher–student concept in their intrusion detection

system wherein clients chose to be a teacher or student. The teacher model (well-trained model) would extract available data locally to train student models. They showed that the blockchain was capable of replacing the central aggregation server. Mothukuri et al. [129] took advantage of the data from decentralized devices to detect intrusions. The local models were trained based on a gated recurrent unit algorithm. Similarly, Attota et al. [132], Preuveneers et al. [130], and Cui et al. [133] also designed related intrusion detection schemes based on decentralized FL and used generative adversarial networks to improve efficiency.

8. Global Model Aggregation

Global model aggregation is one of the most important components in the process of FL. Aggregation determines the overall performance of FL. In this section, we discuss the performance of global model aggregation from three perspectives: aggregation algorithms, communication efficiency, and node selection. We summarize the relevant research in Table 4.

8.1. Aggregation Algorithms

Aggregation algorithms play a major role in FL. They directly influence the performance of the global models. Current research has proposed many aggregation algorithms, for instance, multi-party secure computing and the homomorphic encryption, mentioned above. Since weight-based aggregation has been widely used in current research, the rest of this section focuses on aggregation approaches based on weights.

Table 4. The existing research on improving aggregation performance.

Approach		Ref	Description
aggregation	Reputation	[134,135]	calculate clients reputation
	FedAvg	[136]	counting data amount
	timeliness	[137]	time-weighted aggregation
	FedCA-TDD	[138]	data class on clients
	FedCA-VSA		accuracy on validation
	FedMA	[139]	model distance
	FedDist	[140]	
	FedAtt	[141]	
communication	quantization	[142–145]	quantized before upload
	sparseness	[146–148]	Top-k
		[149]	correct errors in aggregation
	other	[150]	federated dropout
		[151]	signal superposition
		[152]	maximum mean discrepancy

Table 4. *Cont.*

Approach		Ref	Description
nodes selection	Probability	[143,153]	probabilistic selection
	capacity	[154]	client CUP, memory
		[155]	data volume
		[156–159]	communication, computing
		[160]	training delay
	reputation	[161,162]	local model performance
	malicious	[163]	malicious behavior

FedAvg is shorthand for a federated averaging algorithm. It is based on a stochastic gradient descent (SGD) algorithm that was proposed in [136] by Google. It has become a classic FL aggregation algorithm in current literature. In FedAvg, each local model is given a weight by the central server based on the amount of local data contributing to the local training. The global model is aggregated by weighting local models.

Time efficiency is the time cost in training a local model for a client. Time efficiency also can be used to generate weights, besides the amount of data. Chen et al. [137] proposed considering the amount of local data together with the time efficiency of training the local model to generate weights for the aggregation of the global model in an asynchronous FL approach. The higher time efficiency a local model has, the larger the weight the local model has in the global model aggregation.

Reputation-enabled aggregation gives weights to local models based on their reputations. The reputation of a client can be computed variously. Wang et al. [134] reported their belief that clients with more contributions deserved higher weights during FL aggregation. They designed an aggregation method based on clients' reputation that was influenced by clients' contribution in each iteration. They also only allowed the customers with reputation reaching the predefined threshold to participate in aggregating global models. The central server calculated the ecological distribution of reputation from qualified clients to assign weights. Deng et al. [135] designed a similar aggregation algorithm. They let the central server set the aggregation weight according to the learning quality of the nodes. The learning quality of the node was calculated by the global loss of the training model and current node loss. They also introduced a forgetting function in consideration of freshness of nodes.

FedCA-TDD and FedCA-VSA are the approaches proposed by Ma et al. in [138]. Ma et al. reported that aggregating local models by averaging could not achieve a good enough global model, especially for non-independent and identically distributed (non-iid) data. They designed two schemes to redistribute weights. FedCA-TDD set weights based on the number of different classes of data on clients, while FedCA-VSA calculated the accuracy of the local models based on validation data set to determine the corresponding weight.

FedMA is an aggregation approach proposed by Wang et al. [139]. FedMA considers the non-iid data and the mismatch between data and local models. FedMA applied a convolutional neural network (CNN) and LSTM. FedMA matched and averaged the neurons and hidden parameters of the neural network model layer by layer. FedMA was demonstrated to be suitable for aggregating models with data bias.

FedDist is an aggregation approach proposed by Sannara et al. in [140]. It also considered non-iid data, and its local models were sent to the central server for model aggregation based on weights. However, FedDist generated weights through computing the Euclidean distance between client models and the specific neurons added. Compared with FedAvg and FedMA, FedDist had better generalization accuracy.

FedAtt was proposed by Ji et al. in [141]. It considered the contribution of a client model to the global model. Weights were assigned to minimize the distance between the server model and the client models.

8.2. Communication Efficiency

Since the aggregation in FL has to coordinate the central server and participants to exchange models and parameters, communication plays a major role in global aggregation in FL. Efficient communication can greatly reduce the entire time cost in aggregating a global model. Given the same time duration, FL with high communication efficiency may train a global model with higher quality due to completing more iterations. Gradient quantization and gradient sparseness are two major approaches to improving communication efficiency. Gradient refers to the derivatives of functions, and captures the local slopes of functions. In ML, gradients can be used to predict the effect of taking a small step from a point in any direction [164]. In FL, gradients are the major parameters of local models that need to be transferred to the central server for global model aggregation.

Gradient quantization is a common method of compressing gradients. Konečný et al. [142] proposed two schemes to reduce communication overhead. One was called sketched updates. It compressed the trained model through a combination of quantization, random rotation, and subsampling before uploading the model to the central server. The other allowed updates from restricted space and using few variables. Reisizadeh et al. [143] also considered compressing information in a quantitative way. Mills et al. [144] provided CE-FedAvg to quansify modela and reduce the number of communication rounds to reduce communication cost. An effective and proven optimizer named Adam [165] was introduced. Sun et al. [145] utilized lazily aggregated quantized (LAQ) gradients to reduce the amount of data uploaded by the nodes.

Gradient sparseness is also a popular data compression scheme. Liu et al. [146] proposed a Top-k selection-based gradient compression scheme in their proposed anomaly detection framework, where only gradients with a certain degree of importance could be uploaded. Similarly, Sattler et al. [147] proposed sparse ternary compression (STC) that extended the existing Top-k gradient sparse compression technology. They considered the compression of the uplink (client to server) and downlink (server to client) communication, and the robustness of the system. Li et al. [149] focused on the accuracy after compressing data through gradient sparseness. They developed the general gradient sparse (GGS) framework to correct errors in FL aggregation process. In the study presented in [148], Han et al. proposed a fairness-aware bidirectional Top-k GS (FABtop-k) approach, giving to k the sparsity that directly affected the cost of communication.

Besides gradient quantization and gradient sparseness, there are also other solutions that can reduce the amount of communication data. Yang et al. [151] utilized the signal superimposition characteristics of multiple access channels and combined computation and communication to improve communication efficiency. Caldas et al. [150] followed the idea of dropout, proposed in [166], to decrease communication cost. They let clients train a smaller sub-model (the subset of the global model) instead of training an update to the global model. In reference [152], Yao et al. introduced maximum mean discrepancy (MMD) to allow the FL to have high-quality communication without sacrificing accuracy.

8.3. Nodes Selection

Nodes selection is the selection of clients for participation in FL. Nodes selection is significant for FL, because the number of participants in FL may be huge, coordinating all participants is time consuming. Since more participants do not always lead to better global models in practice, what FL approaches really need is clients that contribute high-quality data and models. Therefore, nodes selection is extremely critical in achieving high-quality global models and reducing communication cost. Currently proposed nodes selection mechanisms are typically based on the probability, node capacity, node reputation, and malicious nodes.

Probability mechanisms take advantage of probability to determine the nodes participating in FL. It is simply and easily implemented. Reisizadeh et al. [143] randomly selected the clients participating in FL to reduce the aggregation overhead. However, it did not help the FL to improve the global model's performance, as the nodes were randomly selected. Chen et al. [153] assigned participants higher probabilities of participating in FL aggregation if the local models provided had more influence on the global model. Although their probabilities were slim, the nodes with small probabilities still had a chance to participate in FL aggregation.

Node capacity-based selection mechanisms choose the nodes with stronger resources and computation capabilities to participate in FL, since the nodes with larger capacity generally can train better local models in a shorter time. AbdulRahman et al. [154] proposed to involve a linear regression algorithm to predict the clients' CPU and memory usage and determine whether or not the clients were eligible to participate in FL aggregation. He et al. [160] selected the nodes with low delay in local model training and local model transmission to participate in FL. In the research of [155], Gupta et al. utilized data volume, computation power, and network bandwidth as parameters to calculate the most suitable clients for FL aggregation. Chen et al. [156] considered computing resources and communication conditions to evaluate participants in their asynchronous FL approach. Similarly, references [157–159] made use of computing resources and communication conditions to make decisions.

Node reputation-based selection approaches estimate the contribution of nodes as the nodes' reputations before aggregation. However, it is difficult to evaluate the contribution of a node in an iteration or with all its historical contributions after aggregation as the reputational basis of nodes for selection. Kang et al. [161] used the efficiency of participants to determine reputation, and Wang et al. [162] compared a local model with other local models, such that the node reputation was determined by the results of these comparisons.

Malicious nodes are not allowed to participate in FL as they can cause catastrophic damage to the aggregation. Therefore, it is necessary to eliminate malicious participants, and Li et al. [163] designed a framework for detecting and preventing harmful clients from joining FL aggregation.

9. Future Research Direction

This section further summarizes the potential research directions in the future research of FL over SDNs.

9.1. Estimating Participants

Since it is critical to encourage different parties to contribute their data and participate in FL to improve the quality and performance of FL, while such encouragement is typically based on the estimation of data and models provided by a participant, how to estimate participants is significant for FL. Although currently proposed FL approaches use contribution, reputation, and fairness to estimate a participant, such methods often rely on a pre-define strategy to compute the contribution, reputation, and fairness—no widely accepted strategies have been provided—thus, it is almost impossible to measure the correctness of such strategies. Therefore, participant evaluation is still an open issue in future FL research.

When applying FL over SDNs, controllers in SDN control planes are the participants of FL because they can use the open and standardized protocols and specifications to collaborate with other controllers and switches. Therefore, being a controller becomes a dimension by which to estimate the contribution of a participant [167]. Such estimation can provide guidance for the reward distribution in the incentive mechanism, the weight of FL aggregation, and the selection of nodes. Besides considering the amount of training data provided and the performances of local models, evaluating local controllers should also consider computation loss, time loss, and the distance between the local model and

the global model. The data shift between data sets and the fairness in the evaluation also should be considered [135].

9.2. Anomaly Detection

As more and more organizations and industries, such as healthcare, manufacturing, finance, and retail provide valuable information and services online, their systems and infrastructure are faced with a growing number of anomaly attacks. These attacks aim to disrupt services and steal sensitive or competitive information, user credentials, and so on. Since FL aims to protect data privacy, FL approaches are more frequently involved in generating models and completing tasks for such organizations and industries. Therefore, FL approaches are often subject to anomaly attacks.

Anomaly detection can identify anomaly attacks, and has been studied in diverse research areas and domains. Chandola et al. conducted a comprehensive survey on anomaly detection techniques, approaches, and applications [168]. Although anomaly detection typically includes two parts: detection of anomalous flows and detection of anomalous equipment, the challenge is defining anomalous behaviors in a dynamic manner in evolving environments [168]. Although anomalies can be detected using conventional anomaly detection methods, applying deep ML algorithms to detect anomalies is a hot topic in current research [169,170]. As deep ML algorithms, such as artificial neural networks, are widely used in FL to train local models and aggregate global models, using FL to detect anomalies is an improvement to data privacy, model accuracy, and detection efficiency. Besides, detecting anomalies is also the first layer of filtering participants, such that the devices with anomalous behaviors should never be aggregated by participating in FL. When applying FL over SDNs, SDN architecture has its own security issues. For instance, flow table overflow attacks can cause switches to fail [171]. Accordingly, anomaly detection should consider these security issues in both FL and SDNs.

9.3. Improving the Scalability of FL over SDN

As FL approaches are distributed ML methods, scalability is the big benefit. However, the main obstacle in scaling FL is the significant communication cost regarding the number of communication rounds and the amount of exchanged data per round between the central server and participants. Compressing the transmission information can effectively reduce the amount of exchanged data per round given the total rounds unchanged. However, compression algorithms are computation resource-consuming [172]. When applying compression algorithms in FL, we have to balance computational resource and bandwidth usage. To significantly reduce the number of communication rounds, edge computing may be applied to trade-off local computation for less communication via local updates and one step of synchronization for global convergence.

Regarding the scalability of FL over SDN, a distributed control plane with layer-organized controllers is a good choice. Local controllers in the lower layer can act as the edge computing servers to take more responsibility in running data compression algorithms and to reduce the total rounds of a FL approach.

Developing data compression algorithms for SDN networks can take the advantage of SDN's open and standardized protocols and specification and provide a general way to compress information between the central server and the participants in FL. Data compression can be seen as a type of data encryption. However, data encryption always adds extra burden to hosts, links, and controllers. The higher the security of an encryption mechanism, the larger the bandwidth the mechanism consumes. Designing information encryption algorithms in FL has to balance the resource consumed and the enhancement of the confidentiality of the data [173].

10. Conclusions

In this survey we have discussed the major challenges, technologies, and solutions in applying FL over SDNs, and have aimed to provide a comprehensive overview of the state

of the art in FL technologies in the current literature. After giving the basic concepts and major components of FL and SDN architecture, we highlighted that applying FL over SDNs could encourage more participants to contribute more data and resources, and provide more flexible and effective mechanisms for participant collaborations. Motivated by these advantages, we summarized three major challenges, including incentive mechanisms for participants, privacy and security strategies for parameter communication, and aggregation algorithms for the generation of high-performance global models when applying FL over SDNs. For each of the challenges, we surveyed the related mechanisms, solutions, and related applications, and discussed their advantages and disadvantages. Since FL is a class of distributed ML approaches relying on participants to contribute their data, to train local models, and to share local models, we further suggested that estimation of the contributions, reputations, and fairness of participants encouraging more participants with high-quality data and resources is still the primary concern in current research. As FL is often used to provide data privacy in many user scenarios with sensitive data, FL approaches are typically subject to anomaly attacks more frequently than are other ML approaches. As SDNs also have their own security issues, how to design anomaly detection approaches for FL approaches over SDNs has become significant to future research. Since scalability is always a concern for distributed approaches, we have suggested applying FL over SDNs with layer-organized, distributed control planes, together with data compression and edge computing technology, to effectively scale FL approaches over SDNs.

Author Contributions: Conceptualization, X.M. and L.L.; methodology, X.M. and L.L.; writing—original draft preparation, X.M.; writing—review and editing, L.L., Z.L., R.X.L. and M.Z.; funding, L.L. and Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Nature Science Foundation of China under Grant number 61962016.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Boutaba, R.; Salahuddin, M.A.; Limam, N.; Ayoubi, S.; Shahriar, N.; Felipe, E.S.; Oscar, M.C. A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities. *J. Internet Serv. Appl.* **2018**, *9*, 1–99. [CrossRef]
2. Xu, J.; Glicksberg, B.S.; Su, C.; Walker, P.; Bian, J.; Wang, F. Federated learning for healthcare informatics. *J. Healthc. Infor. Res.* **2021**, *5*, 1–19. [CrossRef]
3. Gupta, D.; Kayode, O.; Bhatt, S.; Gupta, M.; Tosun, A.S. Hierarchical Federated Learning based Anomaly Detection using Digital Twins for Smart Healthcare. *arXiv* **2021**, arXiv:2111.12241.
4. Jiang, J.C.; Kantarci, B.; Oktug, S.; Soyata, T. Federated learning in smart city sensing: Challenges and opportunities. *Sensors* **2020**, *20*, 6230. [CrossRef]
5. Konečný, J.; McMahan, H.B.; Ramage, D.; Richtarik, P. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv* **2016**, arXiv:1610.02527.
6. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [CrossRef]
7. Guo, H.; Liu, A.; Lau, V.K.N. Analog gradient aggregation for federated learning over wireless networks: Customized design and convergence analysis. *IEEE Internet Things J.* **2020**, *8*, 197–210. [CrossRef]
8. Zhang, C.; Xie, Y.; Bai, H.; Yu, B.; Li, W.; Gao, Y. A survey on federated learning. *Knowl.-Based Syst.* **2021**, *216*, 106775. [CrossRef]
9. Li, Q.; Wen, Z.; Wu, Z.; Wang, N.; He, B. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Trans. Knowl. Data Eng.* **2021**. [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. Zhu, H.; Zhang, H.; Jin, Y. From federated learning to federated neural architecture search: A survey. *Complex Intell. Syst.* **2021**, *7*, 639–657. [CrossRef]

12. Karasu, S.; Altan, A.; Saraç, Z.; Hacioğlu, R. Prediction of Bitcoin prices with machine learning methods using time series data. In Proceedings of the 2018 206th IEEE Signal Processing and Communications Applications Conference (SIU), Izmir, Turkey, 2–5 May 2018; pp. 1–4.
13. Altan, A.; Karasu, S.; Bekiros, S. Digital currency forecasting with chaotic meta-heuristic bio-inspired signal processing techniques. *Chaos Solitons Fractals* **2019**, *126*, 325–336. [CrossRef]
14. Savazzi, S.; Nicoli, M.; Rampa, V. Federated learning with cooperating devices: A consensus approach for massive IoT networks. *IEEE Internet Things J.* **2020**, *7*, 4641–4654. [CrossRef]
15. Wang, S.; Tuor, T.; Salonidis, T.; Leung, K.K.; Makaya, C.; He, T.; Chan, K. Adaptive federated learning in resource constrained edge computing systems. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1205–1221. [CrossRef]
16. Hamdan, M.; Hassan, E.; Abdelaziz, A.; Elhigazi, A.; Mohammed, B.; Khan, S.; Vasilakos, A.V.; Marsono, M.N. A comprehensive survey of load balancing techniques in software-defined network. *J. Netw. Comput. Appl.* **2021**, *174*, 102856. [CrossRef]
17. Foerster, K.T.; Schmid, S.; Vissicchio, S. Survey of consistent software-defined network updates. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 1435–1461. [CrossRef]
18. Huang, C.H.; Lee, T.H.; Chang, L.; Lin, J.R.; Horng, G. Adversarial attacks on SDN-based deep learning IDS system. In Proceedings of the 2019 International Conference on Mobile and Wireless Technology, Singapore, 25–27 June 2018; pp. 181–191.
19. Balasubramanian, V.; Aloqaily, M.; Reisslein, M.; Scaglione, A. Intelligent Resource Management at the Edge for Ubiquitous IoT: An SDN-Based Federated Learning Approach. *IEEE Netw.* **2021**, *35*, 114–121. [CrossRef]
20. Lyu, L.; Yu, H.; Ma, X.; Sun, L.; Zhao, J.; Yang, Q.; Yu, P.S. Privacy and robustness in federated learning: Attacks and defenses. *arXiv* **2020**, arXiv:2012.06337.
21. Yin, X.; Zhu, Y.; Hu, J. A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. *Acm Comput. Surv. (Csur)* **2021**, *54*, 1–36. [CrossRef]
22. Mothukuri, V.; Parizi, R.M.; Pouriyeh, S.; Huang, Y.; Dehghantanha, A.; Srivastava, G. A survey on security and privacy of federated learning. *Future Gener. Comput. Syst.* **2021**, *15*, 619–640. [CrossRef]
23. Khan, L.U.; Saad, W.; Zhu, H.; Hossain, E.; Hong, C.S. Federated learning for internet of things: Recent advances, taxonomy, and open challenges. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1759–1799. [CrossRef]
24. Pham, Q.V.; Dev, K.; Maddikunta, P. K.R.; Gadekallu, T.R.; Huynh-The, T. Fusion of federated learning and industrial internet of things: A survey. *arXiv* **2021**, arXiv:2101.00798.
25. Imteaj, A.; Thakker, U.; Wang, S.; Li, J.; Amini, M.H. A survey on federated learning for resource-constrained iot devices. *Internet Things J.* **2021**, *9*, 1–24. [CrossRef]
26. Gadekallu, T.R.; Pham, Q.V.; Huynh-The, T.; Bhattacharya, S.; Maddikunta, P.K.R.; Liyanage, M. Federated Learning for Big Data: A Survey on Opportunities, Applications, and Future Directions. *arXiv* **2021**, arXiv:2110.04160.
27. Xia, Q.; Ye, W.; Tao, Z.; Wu, J.; Li, Q. A Survey of Federated Learning for Edge Computing: Research Problems and Solutions. *High-Confid. Comput.* **2021**, *1*, 100008. [CrossRef]
28. Nguyen, D.C.; Ding, M.; Pham, Q.V.; Pathirana, P.N.; Le, L.B.; Seneviratne, A.; Li, J.; Niyato, D.; Poor, H.V. Federated learning meets blockchain in edge computing: Opportunities and challenges. *IEEE Internet Things J.* **2021**, *8*, 12806–12825. [CrossRef]
29. Liu, Y.; Yuan, X.; Xiong, Z.; Kang, J.; Wang, X.; Niyato, D. Federated learning for 6G communications: Challenges, methods, and future directions. *China Commun.* **2020**, *17*, 105–118. [CrossRef]
30. Yang, Z.; Chen, M.; Wong, K.K.; Poor, H.V.; Cui, S. Federated learning for 6G: Applications, challenges, and opportunities. *arXiv* **2021**, arXiv:2101.01338.
31. Niknam, S.; Dhillon, H.S.; Reed, J.H. Federated learning for wireless communications: Motivation, opportunities, and challenges. *IEEE Commun. Mag.* **2020**, *58*, 46–51. [CrossRef]
32. Shahid, O.; Pouriyeh, S.; Parizi, R.M.; Sheng, Q.Z.; Srivastava, G.; Zhao, L. Communication Efficiency in Federated Learning: Achievements and Challenges. *arXiv* **2017**, arXiv:2107.10996.
33. Zhan, Y.; Zhang, J.; Hong, Z.; Wu, L.; Li, P.; Guo, S. A Survey of Incentive Mechanism Design for Federated Learning. *IEEE Trans. Emerg. Top. Comput.* **2021**. [CrossRef]
34. McMahan, H.B.; Moore, E.; Ramage, D.; Arcas, B.A.Y. Federated learning of deep networks using model averaging. *arXiv* **2016**, arXiv:1602.05629.
35. Ma, X.; Liao, L.X.; Li, Z.; Chao, H. Asynchronous Federated Learning for Elephant Flow Detection in Software Defined Networking Systems. In Proceedings of the 2021 International Conference on Robotics, Intelligent Control and Artificial Intelligence, Guilin, China, 3–5 December 2021. *in press*.
36. Cheng, K.; Fan, T.; Jin, Y.; Chen, T.; Papadopoulos, D.; Yang, Q. Secureboost: A lossless federated learning framework. *IEEE Intell. Syst.* **2021**, *36*, 87–98. [CrossRef]
37. 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.
38. Liu, Y.; Kang, Y.; Xing, C.; Chen, T.; Yang, Q. A secure federated transfer learning framework. *IEEE Intell. Syst.* **2020**, *35*, 70–82. [CrossRef]
39. Gao, D.; Liu, Y.; Huang, A.; Ju, C.; Yu, H.; Yang, Q. Privacy-preserving heterogeneous federated transfer learning. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 2552–2559.

40. Liao, L.X.; Chao, H.C.; Chen, M.Y. Intelligently modeling, detecting, and scheduling elephant flows in software defined energy cloud: A survey. *J. Parallel Distrib. Comput.* **2020**, *146*, 64–78. [CrossRef]
41. Bannour, F.; Souihi, S.; Mellouk, A. Distributed SDN control: Survey, taxonomy, and challenges. *IEEE Commun. Surv. Tutor.* **2017**, *20*, 333–354. [CrossRef]
42. Liu, Y.; Xu, C.; Zhan, Y.; Liu, Z.; Guan, J.; Zhang, H. Incentive mechanism for computation offloading using edge computing: A Stackelberg game approach. *Comput. Netw.* **2017**, *129*, 399–409. [CrossRef]
43. Li, P.; Guo, S. Incentive mechanisms for device-to-device communications. *IEEE Netw.* **2015**, *29*, 75–79. [CrossRef]
44. Yang, D.; Xue, G.; Fang, X.; Tang, J. Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing. In Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, Istanbul, Turkey, 22–26 August 2012; pp. 173–184.
45. Sarikaya, Y.; Ercetin, O. Motivating workers in federated learning: A stackelberg game perspective. *IEEE Netw. Lett.* **2019**, *2*, 23–27. [CrossRef]
46. Khan, L.U.; Pandey, S.R.; Tran, N.H.; Saad, W.; Han, Z.; Nguyen, M.N.H.; Hong, C.S. Federated learning for edge networks: Resource optimization and incentive mechanism. *IEEE Commun. Mag.* **2020**, *58*, 88–93. [CrossRef]
47. Feng, S.; Niyato, D.; Wang, P.; Kim, D.I.; Liang, Y.C. Joint service pricing and cooperative relay communication for federated learning. In Proceedings of the 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Atlanta, GA, USA, 14–17 July 2019; pp. 815–820.
48. Sun, W.; Xu, N.; Wang, L.; Zhang, H.; Zhang, Y. Dynamic digital twin and federated learning with incentives for air-ground networks. *IEEE Trans. Netw. Sci. Eng.* **2020**, *9*, 321–333. [CrossRef]
49. Pandey, S.R.; Tran, N.H.; Bennis, M.; Tun, Y.K.; Manzoor, A.; Hong, C.S. A crowdsourcing framework for on-device federated learning. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 3241–3256. [CrossRef]
50. Xiao, G.; Xiao, M.; Gao, G.; Zhang, S.; Zhao, H.; Zou, X. Incentive Mechanism Design for Federated Learning: A Two-stage Stackelberg Game Approach. In Proceedings of the 2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS), Beijing, China, 2–4 December 2020; pp. 148–155.
51. Hu, R.; Gong, Y. Trading Data For Learning: Incentive Mechanism For On-Device Federated Learning. In Proceedings of the GLOBECOM 2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; pp. 1–6.
52. Zhan, Y.; Li, P.; Qu, Z.; Zeng, D.; Guo, D. A learning-based incentive mechanism for federated learning. *IEEE Internet Things J.* **2020**, *7*, 6360–6368. [CrossRef]
53. Le, T.H.T.; Tran, N.H.; Tun, Y.K.; Han, Z.; Hong, S.C. Auction based incentive design for efficient federated learning in cellular wireless networks. In Proceedings of the 2020 IEEE Wireless Communications and Networking Conference (WCNC), Seoul, Korea, 6–10 April 2020; pp. 1–6.
54. Le, T.H.T.; Tran, N.H.; Tun, Y.K.; Nguyen, M.N.H.; Pandey, S.R.; Han, Z.; Hong, S.C. An incentive mechanism for federated learning in wireless cellular network: An auction approach. *IEEE Trans. Wirel. Commun.* **2020**, *7*, 6360–6368.
55. Yu, H.; Liu, Z.; Liu, Y.; Chen, T.; Cong, M.; Weng, X.; Niyato, D.; Yang, Q. A sustainable incentive scheme for federated learning. *IEEE Intell. Syst.* **2021**, *35*, 58–69. [CrossRef]
56. Zeng, R.; Zhang, S.; Wang, J.; Chu, X. Fmore: An incentive scheme of multi-dimensional auction for federated learning in mec. In Proceedings of the 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), Singapore, 29 November–1 December 2020; pp. 278–288.
57. Jiao, Y.; Wang, P.; Niyato, D.; Lin, B.; Kim, D.I. Toward an automated auction framework for wireless federated learning services market. *IEEE Trans. Mob. Comput.* **2020**, *20*, 3034–3048. [CrossRef]
58. Tang, M.; Wong, V.W.S. An Incentive Mechanism for Cross-Silo Federated Learning: A Public Goods Perspective. In Proceedings of the IEEE INFOCOM 2021-IEEE Conference on Computer Communications, Vancouver, BC, Canada, 10–13 May 2021; pp. 1–10.
59. Pan, Q.; Wu, J.; Bashir, A.K.; Li, J.; Yang, W. Al-Otaibi, Y.D. Joint Protection of Energy Security and Information Privacy for Energy Harvesting: An Incentive Federated Learning Approach. *IEEE Trans. Ind. Infor.* **2021**. [CrossRef]
60. Chai, H.; Leng, S.; Chen, Y.; Zhang, K. A hierarchical blockchain-enabled federated learning algorithm for knowledge sharing in internet of vehicles. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 3975–3986. [CrossRef]
61. Song, T.; Tong, Y.; Wei, S. Profit allocation for federated learning. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 2577–2586.
62. Sim, R.H.L.; Zhang, Y.; Chan, M.C.; Low, B.K.H. Collaborative machine learning with incentive-aware model rewards. *Int. Conf. Mach. Learn. Pmlr* **2020**, *119* 8927–8936.
63. Gupta, D.; Kayode, O.; Bhatt, S.; Gupta, M.; Tosun, A.S. Learner's Dilemma: IoT Devices Training Strategies in Collaborative Deep Learning. In Proceedings of the IEEE 6th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 2–16 June 2020; pp. 1–6.
64. Gupta, D.; Bhatt, S.; Bhatt, P.; Gupta, M.; Tosun, A.S. Game Theory Based Privacy Preserving Approach for Collaborative Deep Learning in IoT. *arXiv* **2021**, arXiv:2103.15245.
65. Bao, X.; Su, C.; Xiong, Y.; Hu, Y.; Huang, W. Flchain: A blockchain for auditable federated learning with trust and incentive. In Proceedings of the 2019 5th International Conference on Big Data Computing and Communications (BIGCOM), Qingdao, China, 9–11 August 2019; pp. 151–159.

66. Tian, M.; Chen, Y.; Liu, Y.; Xiong, Z.; Leung, C.; Miao, C. A Contract Theory based Incentive Mechanism for Federated Learning. *arXiv* **2021**, arXiv:2108.05568.
67. Lim, W.Y.B.; Xiong, Z.; Miao, C.; Niyato, D.; Yang, Q.; Leung, C.; Poor, H.V. Hierarchical incentive mechanism design for federated machine learning in mobile networks. *IEEE Internet Things J.* **2020**, *7*, 9575–9588. [CrossRef]
68. Kang, J.; Xiong, Z.; Niyato, D.; Yu, H.; Liang, Y.C.; Kim, D.I. Incentive design for efficient federated learning in mobile networks: A contract theory approach. In Proceedings of the 2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS), Singapore, 28–30 August 2019; pp. 1–5.
69. Lim, W.Y.B.; Garg, S.; Xiong, Z.; Niyato, D.; Leung, C.; Miao, C.; Guizani, M. Dynamic contract design for federated learning in smart healthcare applications. *IEEE Internet Things J.* **2020**, *8*, 16853–16862. [CrossRef]
70. Ding, N.; Fang, Z.; Huang, J. Incentive mechanism design for federated learning with multi-dimensional private information. In Proceedings of the 2020 18th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT), Philadelphia, PA, USA, 16 July–18 October 2020; pp. 1–8.
71. Ding, N.; Fang, Z.; Huang, J. Optimal contract design for efficient federated learning with multi-dimensional private information. *IEEE J. Sel. Areas Commun.* **2020**, *39*, 186–200. [CrossRef]
72. Wu, M.; Ye, D.; Ding, J.; Guo, Y.; Yu, R.; Pan, M. Incentivizing differentially private federated learning: A multi-dimensional contract approach. *IEEE Internet Things J.* **2021**, *8*, 10639–10651. [CrossRef]
73. Weng, J.; Weng, J.; Zhang, J.; Li, M.; Zhang, Y.; Luo, W. Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive. *IEEE Trans. Dependable Secur. Comput.* **2021**, *18*, 2438–2455. [CrossRef]
74. Zhang, Z.; Dong, D.; Ma, Y.; Ying, Y.; Jiang, D. Refiner: A reliable incentive-driven federated learning system powered by blockchain. *Proc. Vldb Endow.* **2021**, *14*, 2659–2662. [CrossRef]
75. Zhao, Y.; Zhao, J.; Jiang, L.; Tan, R.; Niyato, D.; Li, Z.; Lyu, L.; Liu, Y. Privacy-preserving blockchain-based federated learning for IoT devices. *IEEE Internet Things J.* **2020**, *8*, 1817–1829. [CrossRef]
76. Gao, L.; Li, L.; Chen, Y.; Zheng, W.; Xu, C.Z. FIFL: A Fair Incentive Mechanism for Federated Learning. In Proceedings of the 50th International Conference on Parallel Processing, Lemont, IL, USA, 9–12 August 2021; pp. 1–10.
77. Cong, M.; Yu, H.; Weng, X.; Qu, J.; Liu, Y.; Yiu, S.M. A VCG-based Fair Incentive Mechanism for Federated Learning. *arXiv* **2020**, arXiv:2008.06680.
78. Wang, G.; Dang, C.X.; Zhou, Z. Measure contribution of participants in federated learning. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 2597–2604.
79. Rehman, M.H.; Salah, K.; Damiani, E.; Svetinovic, D. Towards blockchain-based reputation-aware federated learning. In Proceedings of the 2020 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, ON, Canada, 6–9 July 2020; pp. 183–188.
80. Che, Y.K. Design competition through multidimensional auctions. *Rand J. Econ.* **1993**, *24*, 668–680. [CrossRef]
81. Nash, J. Non-cooperative games. *Ann. Math.* **1951**, *54*, 286–295. [CrossRef]
82. Luo, X.; Wu, Y.; Xiao, X.; Ooi, B.C. Feature inference attack on model predictions in vertical federated learning. In Proceedings of the 2021 International Conference on Data Engineering (ICDE), Chania, Greece, 19–22 April 2021; pp. 181–192.
83. Nasr, M.; Shokri, R.; Houmansadr, A. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In Proceedings of the 2019 IEEE symposium on security and privacy (SP), San Francisco, CA, USA, 19–23 May 2019; pp. 739–753.
84. Xiong, P.; Zhu, T.Q.; Wang, X.F. A survey on differential privacy and applications. *Jisuanji Xuebao/Chin. J. Comput.* **2014**, *37*, 101–122.
85. Wei, K.; Li, J.; Ding, M.; Chuan, M.; Yang, H.H.; Farokhi, F.; Jin, S.; Quek, T.Q.S.; Poor, H.V. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3454–3469. [CrossRef]
86. Geyer, R.C.; Klein, T.; Nabi M. Differentially private federated learning: A client level perspective. *arXiv* **2017**, arXiv:1712.07557.
87. Triastcyn, A.; Faltings, B. Federated learning with bayesian differential privacy. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 2587–2596.
88. Zhu, T.; Philip, S.Y. Applying differential privacy mechanism in artificial intelligence. In Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 7–10 July 2019; pp. 1601–1609.
89. Choudhury, O.; Gkoulalas-Divanis, A.; Salonidis, T.; Sylla, I.; Park, Y.; Hsu, G.; Das, A. Differential privacy-enabled federated learning for sensitive health data.*arXiv* **2019**, arXiv:1910.02578.
90. Hu, R.; Guo, Y.; Li, H.; Pei, Q.; Gong, Y. Personalized federated learning with differential privacy. *IEEE Internet Things J.* **2020**, *7*, 9530–9539. [CrossRef]
91. Cao, H.; Liu, S.; Zhao, R.; Xiong, X. IFed: A novel federated learning framework for local differential privacy in Power Internet of Things. *Int. J. Distrib. Sens. Netw.* **2020**, *16*, 1550147720919698 [CrossRef]
92. Lu, Y.; Huang, X.; Dai, Y.; Maharjan, S.; Zhang, Y. Differentially private asynchronous federated learning for mobile edge computing in urban informatics. *IEEE Trans. Ind. Infor.* **2019**, *16*, 2134–2143. [CrossRef]
93. Zhao, Y.; Zhao, J.; Yang, M.; Wang, N.; Lyu, L.; Niyato, D.; Lam, K.Y. Local differential privacy-based federated learning for internet of things. *IEEE Internet Things J.* **2020**, *8*, 8836–8853. [CrossRef]

94. Liu, R.; Cao, Y.; Yoshikawa, M.; Yoshikawa, M.; Chen, H. Fedsel: Federated sgd under local differential privacy with top-k dimension selection. In Proceedings of the DASFAA 2020: Database Systems for Advanced Applications, Jeju, Korea, 21–24 May 2020; pp. 485–501.
95. Truex, S.; Liu, L.; Chow, K.H.; Gursoy, M.E.; Wei, W. LDP-Fed: Federated learning with local differential privacy. In Proceedings of the Third ACM International Workshop on Edge Systems, New York, NY, USA, 27 April 2020; pp. 61–66.
96. Sun, L.; Qian, J.; Chen, X. Ldp-fl: Practical private aggregation in federated learning with local differential privacy. *arXiv* **2020**, arXiv:2007.15789.
97. Zhu, H. On the relationship between (secure) multi-party computation and (secure) federated learning. *arXiv* **2020**, arXiv:2008.02609.
98. Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; Seth, K. Practical secure aggregation for privacy-preserving machine learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; pp. 1175–1191.
99. Kanagavelu, R.; Li, Z.; Samsudin, J.; Yang, Y.; Yang, F.; Goh, R.S.M.; Cheah, M.; Wiwatphonthana, P.; Akkarajitsakul, K.; Wang, S. Two-phase multi-party computation enabled privacy-preserving federated learning. In Proceedings of the 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID), Melbourne, Australia, 11–14 May 2020; pp. 410–419.
100. Sotthiwat, E.; Zhen, L.; Li, Z.; Zhang, C. Partially Encrypted Multi-Party Computation for Federated Learning. In Proceedings of the 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGRID), Melbourne, Australia, 10–13 May 2021; pp. 828–835.
101. Zhou, Z.; Tian, Y.; Peng, C. Privacy-Preserving Federated Learning Framework with General Aggregation and Multiparty Entity Matching. *Iwireless Commun. Mob. Comput.* **2021**, *2021*, 14. [CrossRef]
102. Zhang, J.; Chen, B.; Yu, S.; Deng, H. PEFL: A privacy-enhanced federated learning scheme for big data analytics. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa Village, HI, USA, 9–13 December 2019; pp. 1–6.
103. Zhang, X.; Fu, A.; Wang, H.; Zhou, C.; Chen, Z. A privacy-preserving and verifiable federated learning scheme. In Proceedings of the 2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–6.
104. Hardy, S.; Henecka, W.; Ivey-Law, H.; Nock, R.; Patrini, G.; Smith, G.; Thorne, B. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. *arXiv* **2017**, arXiv:1711.10677.
105. Stripelis, D.; Saleem, H.; Ghai, T.; Dhinagar, N.; Gupta, U.; Anastasiou, C.; Steeg, G.V.; Ravi, S.; Naveed, M.; Thompson, P.M. Secure Neuroimaging Analysis using Federated Learning with Homomorphic Encryption. *arXiv* **2021**, arXiv:2108.03437
106. Cheon, J.H.; Kim, A.; Kim, M.; Song, Y. Homomorphic encryption for arithmetic of approximate numbers. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Hong Kong, China, 19 May–9 July 2017; pp. 409–437.
107. Ma, J.; Naas, S.A.; Sigg, S.; Lyu, X. Privacy-preserving Federated Learning based on Multi-key Homomorphic Encryption. *arXiv* **2021**, arXiv:2104.06824.
108. Zhang, C.; Li, S.; Xia, J.; Wang, W. Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning. In Proceedings of the 2020 USENIX Annual Technical Conference (USENIXATC 20), Online, 15–17 July 2020; pp. 493–506.
109. Jiang, Z.; Wang, W.; Liu, Y. FLASHE: Additively Symmetric Homomorphic Encryption for Cross-Silo Federated Learning. *arXiv* **2021**, arXiv:2109.00675.
110. Mou, W.; Fu, C.; Lei, Y.; Hu, C. A Verifiable Federated Learning Scheme Based on Secure Multi-party Computation. In Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications, Nanjing, China, 5 August–22 November 2021; pp. 198–209.
111. Truex, S.; Baracaldo, N.; Anwar, A.; Steinke, T.; Ludwig, H.; Zhang, R.; Zhou, Y. A hybrid approach to privacy-preserving federated learning. In Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, New York, NY, USA, 11–15 November 2019; pp. 1–11.
112. Xu, R.; Baracaldo, N.; Zhou, Y.; Ali, A.; Heiko, L. Hybridalpha: An efficient approach for privacy-preserving federated learning. In Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, New York, NY, USA, 11–15 November 2019; pp. 13–23.
113. Mugunthan, V.; Polychroniadou, A.; Byrd, D.; Balch, T.H. SMPAI: Secure Multi-Party Computation for Federated Learning. 2019. Available online: https://www.jpmorgan.com/content/dam/jpm/cib/complex/content/technology/ai-research-publications/pdf-9.pdf (accessed on 13 January 2022).
114. Hao, M.; Li, H.; Xu, G.; Liu, S.; Yang, H. Towards efficient and privacy-preserving federated deep learning. In Proceedings of the 2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–6.
115. Rahman, M.A.; Hossain, M.S.; Islam, M.S.; Liu, S.; Yang, H. Secure and provenance enhanced Internet of health things framework: A blockchain managed federated learning approach. *IEEE Access* **2020**, *8*, 205071–205087. [CrossRef] [PubMed]
116. Majeed, U.; Hong, C.S. FLchain: Federated learning via MEC-enabled blockchain network. In Proceedings of the 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), Matsue, Japan, 18–20 September 2019; pp. 1–4.
117. Qi, Y.; Hossain, M.S.; Nie, J.; Li, X. Privacy-preserving blockchain-based federated learning for traffic flow prediction. *Future Gener. Comput. Syst.* **2021**, *117*, 328–337. [CrossRef]

118. Mugunthan, V.; Rahman, R.; Kagal, L. BlockFLow: An Accountable and Privacy-Preserving Solution for Federated Learning. *arXiv* **2020**, arXiv:2007.03856.
119. Li, Y.; Chen, C.; Liu, N.; Huang, H.; Zheng, Z.; Yan, Q. A blockchain-based decentralized federated learning framework with committee consensus. *IEEE Netw.* **2020**, *35*, 234–241. [CrossRef]
120. Li, Z.; Liu, J.; Hao, J.; Wang, H.; Xian, M. CrowdSFL: A secure crowd computing framework based on blockchain and federated learning. *Electronics* **2020**, *9*, 773. [CrossRef]
121. Lu, Y.; Huang, X.; Dai, Y.; Maharjan, S.; Zhang, Y. Blockchain and federated learning for privacy-preserved data sharing in industrial IoT. *IEEE Trans. Ind. Inform.* **2019**, *16*, 4177–4186. [CrossRef]
122. 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.
123. Sun, Y.; Ochiai, H.; Esaki, H. Intrusion Detection with Segmented Federated Learning for Large-Scale Multiple LANs. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–8.
124. Schneble, W.; Thamilarasu, G. Attack detection using federated learning in medical cyber-physical systems. In Proceedings of the 28th International Conference on Computer Communications and Networks (ICCCN), Valencia, Spain, 29 July–1 August 2019; pp. 1–8.
125. Ren, T.; Jin, R.; Lou, Y. Network Intrusion Detection Algorithm Integrating Blockchain and Federated Learning. *Netinfo Secur.* **2021**, *21*, 27–34.
126. Wittkopp, T.; Acker, A. Decentralized federated learning preserves model and data privacy. In Proceedings of the 2020 International Conference on Service-Oriented Computing, Dubai, United Arab Emirates, 20 September–22 November 2019; pp. 176–187.
127. Li, B.; Wu, Y.; Song, J.; Lu, R.; Li, T.; Zhao, L. DeepFed: Federated Deep Learning for Intrusion Detection in Industrial Cyber–Physical Systems. *IEEE Trans. Ind. Inform.* **2020**, *17*, 5615–5624. [CrossRef]
128. Zhao, R.; Yin, Y.; Shi, Y.; Xue, Z. Intelligent intrusion detection based on federated learning aided long short-term memory. *Phys. Commun.* **2020**, *42*, 101157. [CrossRef]
129. Mothukuri, V.; Khare, P.; Parizi, R.M.; Xue, Z. Federated Learning-based Anomaly Detection for IoT Security Attacks. *IEEE Internet Things J.* **2021**. [CrossRef]
130. Preuveneers, D.; Rimmer, V.; Tsingenopoulos, I.; Spooren, J.; Joosen, W.; Ilie-Zudor, E. Chained anomaly detection models for federated learning: An intrusion detection case study. *Appl. Sci.* **2018**, *8*, 2663. [CrossRef]
131. Liu, H.; Zhang, S.; Zhang, P.; Zhou, X.; Shao, X.; Pu, G.; Zhang, Y. Blockchain and Federated Learning for Collaborative Intrusion Detection in Vehicular Edge Computing. *IEEE Trans. Veh. Technol.* **2021**, *70*, 6073–6084. [CrossRef]
132. Attota, D.C.; Mothukuri, V. Parizi, R.M.; Pouriyeh, S. An ensemble multi-view federated learning intrusion detection for iot. *IEEE Access* **2021**, *9*, 117734–117745. [CrossRef]
133. Cui, L.; Qu, Y.; Xie, G.; Zeng, D.; Li, R.; Shen, S.; Yu, S. Security and Privacy-Enhanced Federated Learning for Anomaly Detection in IoT Infrastructures. *IEEE Trans. Ind. Infor.* **2021** doi:10.1109/TII.2021.3107783. [CrossRef]
134. Wang, Y.; Kantarci, B. Reputation-enabled Federated Learning Model Aggregation in Mobile Platforms. In Proceedings of the ICC 2021-IEEE International Conference on Communications, Chongqing, China, 5 September–15 October 2021; pp. 1–6.
135. Deng, Y.; Lyu, F.; Ren, J.; Chen, Y.C.; Yang, P.; Zhou, Y.; Zhang, Y. FAIR: Quality-Aware Federated Learning with Precise User Incentive and Model Aggregation. In Proceedings of the IEEE INFOCOM 2021-IEEE Conference on Computer Communications, Colombia, BC, Canada, 10–13 May 2021; pp. 1–10.
136. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, Seattle, WA, USA, 9–11 May 2017; pp. 1273–1282.
137. Chen, Y.; Sun, X.; Jin, Y. Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 4229–4238. [CrossRef] [PubMed]
138. Ma, Z.; Zhao, M.; Cai, X.; Jia, Z. Fast-convergent federated learning with class-weighted aggregation. *J. Syst. Archit.* **2021**, *117*, 102125. [CrossRef]
139. Wang, H.; Yurochkin, M.; Sun, Y.; Papailiopoulos, D.; Khazaeni, Y. Federated learning with matched averaging. *arXiv* **2020**, arXiv:2002.06440.
140. Sannara, E.K.; Portet, F.; Lalanda, P.; Vega, G. A Federated Learning Aggregation Algorithm for Pervasive Computing: Evaluation and Comparison. In Proceedings of the 2021 IEEE International Conference on Pervasive Computing and Communications (PerCom), Kassel, Germany, 22–26 March 2021; pp. 1–10.
141. Ji, S.; Pan, S.; Long, G.; Li, X.; Jiang, J.; Huang, Z. Learning private neural language modeling with attentive aggregation. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8.
142. Konečný, J.; McMahan, H.B.; Yu, F.X.; Richtarik, P.; Suresh, A.T.; Bacon, D. Federated learning: Strategies for improving communication efficiency. *arXiv* **2016**, arXiv:1610.05492.
143. Reisizadeh, A.; Mokhtari, A.; Hassani, H.; Jadbabaie, A.; Pedarsani, R. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In Proceedings of the 2020 International Conference on Artificial Intelligence and Statistics, Tianjing, China, 26–28 June 2020; pp. 2021–2031.
144. Mills, J.; Hu, J.; Min, G. Communication-efficient federated learning for wireless edge intelligence in IoT. *IEEE Internet Things J.* **2019**, *7*, 5986–5994. [CrossRef]

145. Sun, J.; Chen, T.; Giannakis, G.B.; Yang, Q.; Yang, Z. Lazily aggregated quantized gradient innovation for communication-efficient federated learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**. [CrossRef] [PubMed]
146. Liu, Y.; Garg, S.; Nie, J.; Zhang, Y.; Xiong, Z.; Kang, K.; Hossain, M.S. Deep anomaly detection for time-series data in industrial iot: A communication-efficient on-device federated learning approach. *IEEE Internet Things J.* **2020**, *8*, 6348–6358. [CrossRef]
147. Sattler, F.; Wiedemann, S.; Müller, K.R.; Samek, W. Robust and communication-efficient federated learning from non-iid data. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 3400–3413. [CrossRef]
148. Han, P.; Wang, S.; Leung, K.K. Adaptive gradient sparsification for efficient federated learning: An online learning approach. In Proceedings of the 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), Singapore, 29 November–1 December 2020; pp. 300–310.
149. Li, S.; Qi, Q.; Wang, J.; Sun, H.; Li, Y.; Yu, F.R. Ggs: General gradient sparsification for federated learning in edge computing. In Proceedings of the 2020 IEEE International Conference on Communications(ICC), Dublin, Ireland, 7–11 June 2020; pp. 1–7.
150. Caldas, S.; Konečny, J.; McMahan, H.B.; Talwalkar, A. Expanding the reach of federated learning by reducing client resource requirements. *arXiv* **2018**, arXiv:1812.07210
151. Yang, K.; Jiang, T.; Shi, Y.; Ding, Z. Federated learning via over-the-air computation. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 2022–2035. [CrossRef]
152. Yao, X.; Huang, C.; Sun, L. Two-stream federated learning: Reduce the communication costs. In Proceedings of the 2018 IEEE Visual Communications and Image Processing (VCIP), Taichung, Taiwan, 9–12 December 2018; pp. 1–4.
153. Chen, M.; Poor, H.V.; Saad, W.; Cui, S. Convergence time optimization for federated learning over wireless networks. *IEEE Trans. Wirel. Commun.* **2020**, *20*, 2457–2471. [CrossRef]
154. AbdulRahman, S.; Tout, H.; Mourad, A.; Talhi, C. FedMCCS: Multicriteria client selection model for optimal IoT federated learning. *IEEE Internet Things J.* **2020**, *8*, 4723–4735. [CrossRef]
155. Gupta, M.; Goyal, P.; Verma, R.; Shorey, R.; Saran, H. FedFm: Towards a Robust Federated Learning Approach For Fault Mitigation at the Edge Nodes. *arXiv* **2021**, arXiv:2111.01074
156. Chen, Z.; Liao, W.; Hua, K.; Lu, C.; Yu, W. Towards asynchronous federated learning for heterogeneous edge-powered internet of things. *Digit. Commun. Netw.* **2021**, *7*, 317–326. [CrossRef]
157. Zhao, Y.; Gong, X. Quality-aware distributed computation and user selection for cost-effective federated learning. In Proceedings of the 2021 IEEE Conference on Computer Communications Workshops(INFOCOM WKSHPS), Vancouver, BC, Canada, 10–13 May 2021; pp. 1–6.
158. Nishio, T.; Yonetani, R. Client selection for federated learning with heterogeneous resources in mobile edge. In Proceedings of the 2019 IEEE International Conference on Communications(ICC), Shanghai, China, 20–24 May 2019; pp. 1–7.
159. Lai, F.; Zhu, X.; Madhyastha, H.V.; Chowdhury, M. Oort: Informed participant selection for scalable federated learning. *arXiv* **2020**, arXiv:2010.06081.
160. He, W.; Guo, S.; Qiu, X.; Chen, L.; Zhang, S. Node selection method in federated learning based on deep reinforcement learning. *J. Commun.* **2021**, *42*, 62–71.
161. Kang, J.; Xiong, Z.; Niyato, D.; Zou, Y.; Zhang, Y.; Guizani, M. Reliable federated learning for mobile networks. *IEEE Wirel. Commun.* **2020**, *27*, 72–80. [CrossRef]
162. Wang, Y.; Kantarci, B. A Novel Reputation-Aware Client Selection Scheme for Federated Learning within Mobile Environments. In Proceedings of the 2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Online, 14–16 September 2020; pp. 1–6.
163. Li, S.; Cheng, Y.; Wang, W.; Liu, Y.; Chen, T. Learning to detect malicious clients for robust federated learning. *arXiv* **2020**, arXiv:2002.00211.
164. Kochenderfer, M.J.; Wheeler, T.A. *Algorithms for Optimization*; Mit Press: London, UK, 2019.
165. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
166. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
167. Song, Z.; Sun, H.; Yang, H.H.; Wang, X.; Zhang, Y.; Quek, T.Q.S. Reputation-based Federated Learning for Secure Wireless Networks. *IEEE Internet Things J.* **2021**, *9*, 1212–1226. [CrossRef]
168. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. *Acm Comput. Surv. (Csur)* **2009**, *41*, 1–58. [CrossRef]
169. Kimmell, J.C.; Abdelsalam, M.; Gupta, M. Analyzing Machine Learning Approaches for Online Malware Detection in Cloud. *arXiv* **2021**, arXiv:2105.09268.
170. Pang, G.; Shen, C.; Cao, L.; Hengel, A.V.D. Deep learning for anomaly detection: A review. *Acm Comput. Surv. (Csur)* **2021**, *54*, 1–38. [CrossRef]
171. Chica, J.C.C.; Imbachi, J.C.; Vega, J.F.B. Security in SDN: A comprehensive survey. *J. Netw. Comput. Appl.* **2020** *159*, 102595. [CrossRef]
172. Sharma, N.; Batra, U. Performance analysis of compression algorithms for information security: A Review. *Eai Endorsed Trans. Scalable Inf. Syst.* **2020**, *7*, 163503. [CrossRef]
173. Xu, J.; Du, W.; Jin, Y.; He, W.; Cheng, R. Ternary compression for communication-efficient federated learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**. [CrossRef] [PubMed]