

Peer-to-Peer Federated Learning: A Comprehensive Survey

Ashley Allen ^{1,*}, Alexios Mylonas ¹ , Stilianos Vidalis ¹ and Nikolaos Pitropakis ^{2,*} 

¹ Cybersecurity and Computing Systems Research Group, Department of Computer Science, University of Hertfordshire, Hatfield AL10 9AB, UK; a.mylonas@herts.ac.uk (A.M.); s.vidalis@herts.ac.uk (S.V.)

² Computing and Mathematics, School of Science and Technology, The American College of Greece, 6 Gravias Street, Aghia Paraskevi, 153 42 Athens, Greece

* Correspondence: a.allen3@herts.ac.uk (A.A.); npitropakis@acg.edu (N.P.)

Abstract

The last five years have seen considerable growth in the topic of peer-to-peer (P2P) federated learning (FL). This framework removes the central coordinating server used in conventional federated learning and instead requires participating nodes to manage model training, peer selection, communication, aggregation, and trust directly. This provides a promising route for privacy-preserving and decentralised machine learning, but it also introduces unresolved challenges in topology selection, participant incentivisation, communication efficiency, security, and evaluation. Existing studies frequently evaluate proposed methods under narrow assumptions, such as static network membership, homogeneous devices, fixed bandwidth, limited topology choices, and public benchmark datasets. Existing surveys also tend to present taxonomies of decentralised federated learning rather than synthesising how topology, incentives, and communication algorithms jointly affect deployability. This paper reviews recent work on peer-to-peer federated learning across three connected dimensions: network topology, incentive mechanisms, and communication algorithms. We compare the topologies, datasets, experimental assumptions, incentive designs, communication strategies, and open issues reported in the literature. The review shows that highly connected topologies tend to improve convergence but increase communication overhead and vulnerability to bottlenecks; sparse and dynamic topologies improve efficiency but create challenges for convergence, reliability, and node drop-out. Incentive mechanisms increasingly combine reward, reputation, validation, and punishment but remain weakly validated under realistic churn, heterogeneous resources, and adversarial behaviour. Communication algorithms reduce bandwidth through gossip, sparsification, prediction, routing, and multi-step aggregation but often trade communication savings against accuracy, robustness, and generalisability. Across all three areas, the field lacks standardised benchmarks, reproducible experimental settings, and realistic evaluation under unstable peer-to-peer conditions. We conclude by identifying cross-cutting research gaps and recommending future work on dynamic topologies, heterogeneous devices, real-world datasets, incentive robustness, and comparable benchmarking.



Academic Editor: Isidoros Perikos

Received: 24 April 2026

Revised: 19 June 2026

Accepted: 22 June 2026

Published: 1 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: survey; federated learning; peer-to-peer; topology; incentive mechanisms; algorithms

1. Introduction

In the past ten years, artificial intelligence (AI) and machine learning (ML) systems have become increasingly prominent in everyday life. Conversational agents are now widely studied in customer-service settings, where they are used to answer routine queries and reduce the need for human intervention in simple interactions [1]. Automated

plagiarism-detection systems have also become common in education, supported by a substantial body of work on computational methods for identifying textual similarity, paraphrase, translation-based plagiarism, and other forms of academic reuse [2]. More recently, general-purpose large language models (LLMs), including GPT-4 and Gemini, have demonstrated strong performance across a wide range of natural language, code-generation, reasoning, and multimodal tasks [3,4]. However, the development and deployment of such large-scale models remains resource intensive, as shown in [5–8]. These requirements make comparable model development impractical for many organisations and motivate alternative training paradigms in which data and computation remain distributed.

A more scalable solution to the problem of model training comes in the form of federated learning (FL), as described by McMahan [9]. In this framework, the model is trained using data that exists on a remote node such as a mobile phone or tablet, with only the gradients sent to a central management server that then combines the totality of gradients into a new model. The new model is then downloaded by the edge client, and the process begins again. This model of gradient transfer has many benefits. The presence of many thousands or millions of edge clients means that there is no need for the central collection of training data. Instead, the data remains on those devices. This has the added benefit of data privacy, meaning that models can be trained against sensitive data without requiring central collation. McMahan also identified several potential shortcomings, such as limited communication with edge devices, the fact that the local training data may not be representative of the total dataset, and that some devices may be over- or under-represented depending on their use cases. An earlier paper from 2016 [10] anticipates many of the issues in McMahan, discussing strategies for ensuring that training rounds are completed where edge devices may enter or leave the network at random. The theory behind federated learning has been employed in several areas, with robust performance seen in language scenarios. FL has been applied in practical language-input settings. For example, work on Google keyboard query suggestions and federated n-gram language modelling shows that FL can be used to improve user-facing language models while avoiding direct collection of raw user text [11,12]. This demonstrates that FL is not only a theoretical privacy-preserving training paradigm but can also be deployed in production-scale settings where user data is sensitive and centrally collecting raw input would be undesirable.

Unfortunately, this new paradigm also provides new opportunities for adversaries to exploit. The sending of gradients only means that the training data for each device is obscured from the central management server. The management server is taking on trust that the remote client will not alter the supplied model in any way and will return the true gradients generated during the training round. Papers such as [13,14] discuss scenarios where malicious clients either deliberately or unknowingly train against a different model to the one provided or supply incorrect data to the central management server in an attempt to “poison” the global model. Such poisoning aims to either degrade performance or more sinisterly flip particular identifiers to produce incorrect responses. The authors in [15] give details of several poisoning attacks, whilst [16,17] discuss potential defences. It should be noted that a significant amount of further literature exists in relation to attacks and defences against centralised federated learning.

The topic of this review paper is decentralised federated learning. Unlike “traditional” federated learning, no central management server is used to combine gradients. Instead, each network member is an autonomous entity that manages model training, peer selection, communication, and model updates. Previous applicable work includes [18] from 2016, [19] from 2010, and [20] from 2015, but most research has developed in parallel with centralised federated learning, with the bulk of papers produced from 2017 onwards. Decentralised

federated learning should therefore be seen as an alternative to the centralised paradigm rather than an offshoot of it.

The lack of a central management server presents unique challenges with respect to data security. Each node is in control of the training data, the model, and the weights that are sent to peers. When all nodes are part of a trusted network, as with [21,22] or [23], the potential for a malicious actor to introduce faked training data, a back-doored model, or to transfer faked weights is minimal. However, when nodes are allowed to join a decentralised network with minimal oversight, the potential for a malicious actor to perform attacks is substantial. The authors of [24] discuss this problem and suggest a sophisticated authentication mechanism as a potential solution. However, with no trusted central management server to perform authentication against, even the most secure system must require a significant amount of trust between parties to already exist.

Additionally, most attacks identified against centralised federated learning are also applicable to decentralised federated learning. The work of [25], for example, looks at the potential for abuse via data poisoning and model swapping. This is where an unwitting user downloads an already compromised model rather than the correct one. Both attacks are relevant in the case of distributed federated learning. In fact, model swapping is more likely where no central management server exists as a single source of truth for clients to connect to and download the correct model. The model reuse attacks in [26] are also more likely in a topology where no definitive source exists. Model performance in decentralised learning is also of concern. In [27], the authors introduce a peer-to-peer algorithm for non-independently and non-identically distributed (non-IID) data. This recognises that the data stored on participant devices will in many cases be unique. This may potentially lead to over- and under-classification of groups within the data depending on the relative frequency of each type. Such scenarios have been widely explored in traditional machine learning and are active topics of research in both types of federated learning.

Finally, many privacy concerns exist when discussing decentralised federated learning. Whilst each node is working on its own local data, it will be receiving weight updates from peers at regular intervals. A node may connect to the network, receive weight updates, and then detach from the network. It can then use techniques such as those in [28–30] to reconstruct the training data. This is a concern in both types of federated learning. With membership of the network ranging from fixed to ephemeral, ensuring that only legitimate nodes receive weight updates is important and remains an open question.

The motivation for this review is therefore to determine whether current peer-to-peer federated learning research provides sufficient evidence for deployment beyond controlled laboratory settings. In particular, we focus on three aspects that determine whether decentralised federated learning systems can operate in practice: the topology over which nodes exchange updates, the incentive mechanisms used to encourage participation and honest behaviour, and the communication algorithms used to transfer model updates efficiently. These aspects are strongly connected. A topology may improve convergence but increase communication cost, an incentive mechanism may reward participation but fail under collusion or heterogeneous resources, and a communication algorithm may reduce bandwidth use but weaken accuracy or robustness.

This review is guided by the following research questions (RQs):

RQ1: What network topologies are used in recent peer-to-peer federated learning studies, and what limitations remain with respect to scalability, convergence, node drop-out, heterogeneity, and real-world deployment?

RQ2: What incentive mechanisms are proposed for peer-to-peer federated learning, and how well do they address participation, fairness, attack resistance, cost, and heterogeneous node capabilities?

RQ3: What communication algorithms are used to exchange model updates in peer-to-peer federated learning, and what trade-offs do they introduce between communication efficiency, accuracy, robustness, and adaptability?

RQ4: To what extent can existing peer-to-peer federated learning studies be compared, reproduced, and generalised across datasets, topologies, node assumptions, and evaluation settings?

The contributions of this paper are as follows:

- We collect and review recent papers on decentralised federated learning and discuss them based on three categories: incentive mechanism, algorithm, and topology. We compare and analyse state-of-the-art incentive mechanisms, network topologies, and recent state-of-the-art algorithms used for weight transfer in decentralised networks.
- We expand beyond the presentation of the reviewed works by comparing their relative strengths, weaknesses, assumptions, datasets, and open issues across three connected dimensions: network topology, incentive mechanisms, and communication algorithms. Rather than claiming to be the first survey of decentralised federated learning, this review contributes a focused cross-category synthesis of how these three dimensions interact and where current evidence remains insufficient for robust comparison or deployment. In particular, we highlight recurring gaps around realistic network conditions, heterogeneous devices, non-IID data, dynamic topologies, incentive robustness, communication efficiency, and the absence of standardised peer-to-peer federated learning benchmarks.
- Our review highlights the absence from the literature of standardised comparison metrics and appropriate datasets, which impedes their direct comparison. Without such metrics, comparison remains largely qualitative.

The rest of the paper is structured as follows. Section 2 provides a brief background on decentralised federated learning. Section 3 discusses our selection methodology and presents our literature review. Section 4 contains a discussion on current open issues. Section 5 presents related work. Section 6 provides a conclusion and discussion around future research directions.

2. Background

This section discusses the fundamentals of distributed federated learning focusing on (a) topology, (b) incentive mechanism, and (c) communication methods and strategies.

Standard federated learning occurs predominantly within a star topology, with nodes communicating directly to the central management server. This is the format used in the original McMahan 2017 paper [9] and subsequent work such as [31,32], as well as in works related to algorithm design, such as [33]. However, such topologies are inherently non-scalable with physical constraints on centralised processing power and network performance, forming an upper bound on the number of client updates that can be accepted. Works such as [34–36] offer improvements on single-layer client–server topologies by introducing one or more sub-aggregation layers, forming a tree topology rather than a star. This alleviates some of the scaling issues identified but at the cost of generating new choke points and targets for attackers.

Decentralised federated learning is theoretically unconstrained in topology type, with some examples given in Figure 1. Common topologies found in the literature include mesh; “all-to-all” topologies, as in [37–39]; and ring topologies, where each node speaks only to its two immediate neighbours, as in [40–43]. Both topologies introduce benefits and drawbacks. Specifically, a full mesh topology ensures that node weights are shared amongst all members of the network. It is, however, much less bandwidth-efficient than even the star topology of centralised federated learning. In graph-theoretic terms, a fully

connected network over n nodes corresponds to the complete graph K_n , which contains $\frac{n(n-1)}{2}$ edges and therefore grows quadratically with the number of nodes [44]. It is also unclear how communication link failures and node drop-outs are handled. Ring topologies meanwhile are more network efficient, with each node communicating only with its two nearest neighbours, regardless of the size of the ring. However, weight propagation from one side of the ring to the other is extremely slow. Moreover, each node is reliant on the honesty of its neighbours; thus, if a malicious node is inserted into the ring, detection may be difficult. A ring lattice graph is a generalised form of both a mesh graph and a ring graph, where the number of edges (k) connected to each node (n) is determined by the following bounds:

$$n - 1 \geq k \geq 2$$

When $k = n - 1$, then the graph is a fully connected mesh, whilst when $k = 2$, the graph is a simple ring.

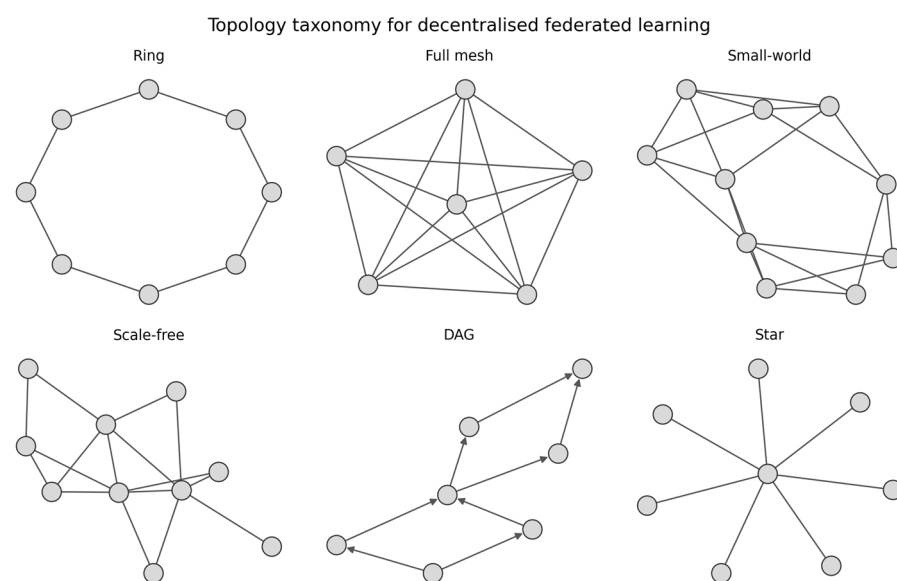


Figure 1. Common topology classes: ring, full mesh, small-world, scale-free, DAG, and star.

Non-standard topologies have also appeared in the literature. The authors in [45] discuss a “one-hop” topology. This is conceptually different to the topologies of the previous paragraph, with inter-node communication concerned only with those nodes that can be reached directly, separate from the actual underlying network topology. Depending on the number of nodes as well as the number of hops between them, this methodology can be significantly more efficient than the “all-to-all” topology. However, there is also the potential for weight updates to propagate much more slowly. Similar work on arbitrary graphs appears in [46,47], with [46] considering machine learning on an arbitrary graph as a subset of social learning on a graph. This introduces a wide range of literature from outside of computer science and suggests potential avenues for future cross-over research. Meanwhile in [47], the authors offer mathematical proof of convergence under specific circumstances and discuss the rate of convergence with respect to the importance of nodes in the network and the quality of their training data. Work on scale-free networks is also discussed. These networks are defined via a power law:

$$P(k) \propto k^{-\tau}$$

where the number of nodes in the network having k edges is proportional to k to the negative power of some constant τ . Scale-free networks are so named because the distribution of nodes with k edges is in the same proportion regardless of the number of nodes.

One of the key benefits of traditional machine learning is that the training data available to the model is always present. This contrasts with federated learning, both centralised and distributed, where nodes may leave or join during the training process. In these environments, training data remains under the control of the nodes in the cluster. They are free to leave and rejoin at will as well as refuse to send their weights after a successful training run. This can have serious consequences for the quality of the model and its subsequent performance against real-life data, as discussed in [9].

For this reason, federated training systems offer incentives to participants to ensure that they complete a training round and return their data. The authors of [48] discuss several mechanisms for incentivising participants. A key metric is the Shapley value [49] for determining fair allocation of incentives based on contribution to the task. Readers interested in the Shapley value can refer to [50]. The calculation of the Shapley value is extremely computationally intensive, and determining contribution relies heavily on knowing the number of participants within a training round. In a decentralised scenario, this may be difficult to ascertain. In addition, no central node exists upon which to perform the necessary calculations. This means that Shapley-based incentives are unlikely to work well in decentralised systems. A similar import from game theory is the Stackelberg game, discussed in [48,51,52]. Examples of use in a distributed scenario can be found in [53,54].

Blockchain-based incentives are popular in the current literature, as they are also decentralised and allow for rewards to be allocated without the need for a central controlling body. The most common types of incentives are “proof of work” [55–57] and “proof of stake” [58–60]. Proof of work rewards speed, whilst proof of stake rewards nodes who guarantee resources to the wider network. Other incentive mechanisms have been envisaged in the recent literature, such as the “proof of accuracy” scheme in [61]. Here, nodes are elected and win the right to update the global model based on their local accuracy. Proof of accuracy is also seen in [62]. Meanwhile, Qi et al. [63] uses reputation scores to prioritise the adoption of weights from honest models. Parameter quality, bid price analysis, and reputation score are all used to give a multi-dimensional representation of each node. This allows the network to award high-quality participants with larger rewards.

A key challenge from the very start with federated learning has been to ensure that the weights generated by nodes at the periphery are communicated to the central server as efficiently as possible. Delays to training rounds may already be present because of node drop-out, so ensuring that the results of a successful training round are incorporated in the central model as quickly as possible is important. One of the most significant early papers on the topic is [10]. Here, the authors discuss strategies for improving communication efficiency. These include (1) performing several training rounds locally before sending weights, (2) sending only the differences between the original model and the trained one, (3) reducing the precision of weights that are sent to the central server, and (4) sparsification of updates, where only a certain percentage of weights is transferred based on a particular requirement, the rest being zeroed. The paper also discussed strategies for client selection that maximise the chances of a successful training round, such as battery life and the strength of the network connection between node and central server. The authors of [64] pursue a similar line of enquiry. They discuss hyperparameter tuning to provide bias towards either nodes with significant model updates or those with low-latency connections to the central server. In this way, either model convergence or speed of training can be prioritised. The literature suggests that the strategies discussed in [10] are the most actively researched. Works such as [65–68] all discuss similar strategies.

When discussing distributed federated learning, new challenges arise. When the central management server is removed, communication becomes much more difficult to manage. Nodes in a traditional federated network have little or no knowledge of each other,

but in a decentralised network, they must maintain their own communication strategies. One of the most popular algorithms is the gossip algorithm, as initially discussed in the context of decentralized federated learning in [69]. Gossip algorithms typically require a node to choose a peer node at random for communication. Weights are then transferred between them (either bidirectionally or unidirectionally). A new training round is then begun; at the end, another node is chosen at random, and communication occurs again. An individual node does not need knowledge of the entire network, just knowledge of at least one peer node. Performance of the algorithm is highly dependent on the connectivity of the graph of nodes. Highly connected graphs overall will converge faster and more reliably than sparsely connected graphs. The mathematical foundation behind this is shown in [70,71], with later discussion in [72–74]. Current decentralised federated learning literature leans heavily on gossip algorithms for communication because they are forgiving of client drop-out and disconnection, whether permanent or temporary. If a peer node is no longer available when a training round is complete, a new one can be chosen quickly at random. Works such as [75–77] apply this knowledge as part of their algorithm design and test against real-world problems. In contrast to gossip-based approaches, graph-based approaches rely on each node sharing its model updates with each node connected to it. Simple examples include ring graphs and fully connected graphs, where each node is directly connected to every other node. Papers such as [47,78,79] discuss training over arbitrary graphs as well as strategies for combatting node drop-out and bandwidth-constrained communication. Graph-based strategies benefit from nodes having a wider level of knowledge around their network, meaning that they can choose where to send model updates and, thus, avoid duplication. However, excessive redundancy of communication paths and increased complexity can mean that resource usage is much higher when compared to gossip communications.

3. Topology, Incentives, and Communication Algorithms in P2P Federated Learning

This section presents a comparison of topology selection, incentive mechanisms, and communication algorithms with respect to distributed federated learning. We condense each paper to the key contributions. Where appropriate, we extract results and discuss them along with the datasets used. We also identify the key open issues identified in each paper.

For consistency across the review tables, topology categories refer to the graph structures evaluated in each study, incentive categories refer to mechanisms used to encourage participation or discourage low-quality behaviour, and communication categories refer to the update-exchange strategy used to transfer or reduce model information. These categories are not mutually exclusive. Where a paper evaluates more than one topology, incentive mechanism, or communication strategy, it may appear in multiple columns. The X marks in the open-issue tables identify limitations explicitly reported by the cited papers or limitations that follow directly from their experimental assumptions, such as fixed node membership, homogeneous devices, or single-topology evaluation.

3.1. Collection Methodology

Federated learning is a popular research topic at present, with much literature available across the three categories given in Section 2. We have given preference to papers published within the last five years but have included older works where materially relevant. An initial pool of papers was identified based on the following criteria:

- Key search terms such as “distributed federated learning,” and “peer to peer federated learning” were used across relevant search engines, such as Google Scholar, IEEE Xplore, and ACM Digital Library.
- Browsing the latest releases from top-rated security journals such as IEEE Transactions on Network Science and Engineering, IEEE Transactions on Signal Processing, Computers & Security, and IEEE Transactions on Neural Networks and Learning Systems as well as conferences such as NeurIPS and USENIX.

The initial collection of papers was expanded with the following:

- Additional associated papers from the references of those found were also included where appropriate.
- We browsed the journals and conference proceedings where these papers were published if not already collected.

The reviewed corpus contains both recent peer-to-peer federated learning studies and older foundational works that remain necessary for interpreting topology, gossip communication, graph connectivity, and incentive design. To avoid conflating these two roles, Figure 2 separates the full bibliography from the primary reviewed papers. This distinction is important because the review gives preference to recent publications while still retaining older work where it provides essential mathematical, architectural, or historical context.

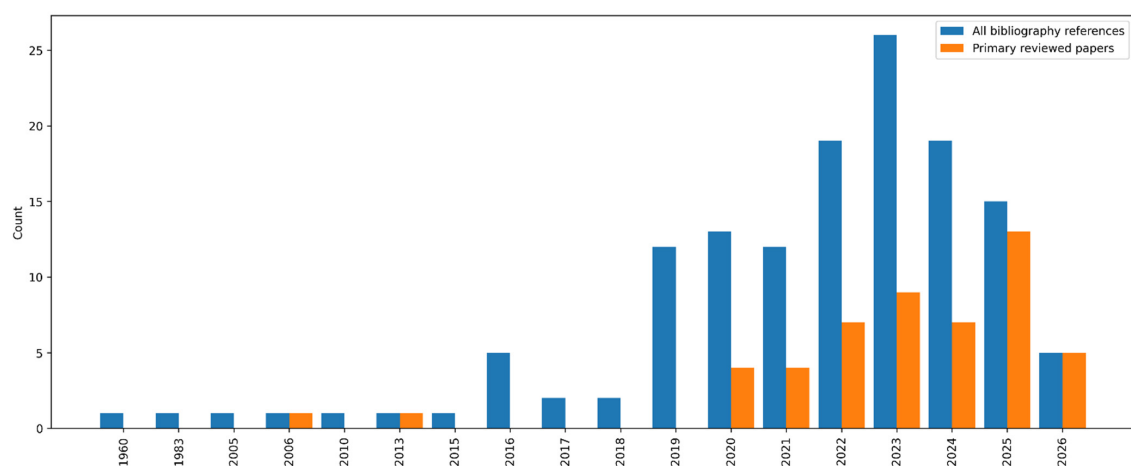


Figure 2. Publication year distribution of references and primary reviewed papers.

The papers included in the main analysis are distributed across the three organising dimensions of this review. Figure 3 summarises this distribution and shows that topology forms the largest category, followed by communication algorithms and incentive mechanisms.

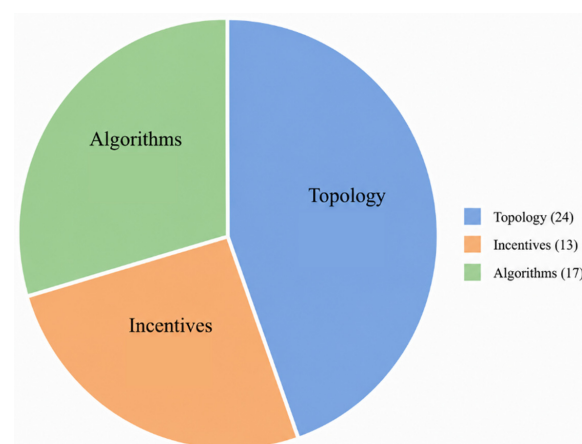


Figure 3. Reviewed papers by survey dimension.

We first examine topology because it determines the communication paths over which model updates move. We then consider incentive mechanisms, which influence whether nodes are willing to participate and behave honestly. Finally, we review communication algorithms, which determine how model updates are exchanged once the topology and participation model are established.

3.2. Topology

Current work focuses on the creation of more efficient or faster performing communication methods, as shall be seen in Section 3.4. These methods are independent of the underlying topology of the nodes forming the decentralised network. This section discusses the impact that topology can have on model convergence and whether the model in fact converges at all.

Traditional federated learning works via a star topology, with remote nodes performing training on their local dataset, then transferring the model weights to a central server for consolidation and distribution, as in [9,80,81]. More advanced models may use multiple layers of consolidation nodes, but the resulting structure remains hierarchical, with weights flowing upward and model updates flowing downward. Jiang et al. analyse decentralised deployments across line, ring, star, and mesh topologies and explicitly connect topology choice to convergence and generalisation under IID and non-IID data [82]. In contrast, decentralised federated learning is fundamentally anti-hierarchical and imposes no logical topology upon the nodes in the network. As shown in Table 1, several distinct topologies are encountered in the current literature, namely (a) mesh, (b) ring, (c) directed acyclic graph, (d) ring lattice, (e) scale-free, and (f) other specialized forms. Recent work is increasingly concerned not simply with choosing one of these structures but with adapting it or understanding the risks it creates. Feng et al. [83] demonstrate that network topology may itself leak information, as aspects of the underlying overlay graph can potentially be inferred from model-update behaviour and convergence patterns. Table 2 depicts the datasets used in the literature, while Table 3 summarises the open issues identified in topology-related papers.

Table 1. Topologies used in the reviewed papers.

Author	Ring	Scale-Free	Mesh	Random	Ring Lattice	Square Lattice	Torus	DAG	Fixed Degree	Star	Line	Time-Variant	Small World	Other/Note
[84]	X	X												
[75]			X											
[73]					X								X	Prism
[72]				X		X								Les Miserables network
[85]	X						X					X		
[86]		X		X									X	
[87]														Topology-free
[88]			X											
[89]														Multi-hop
[90]		X		X										
[47]								X		X		X		
[91]	X								X	X				
[74]	X		X											
[79]							X							
[40]	X													
[92]				X								X		
[93]	X	X							X					
[94]			X						X	X	X			

Table 1. Cont.

Author	Ring	Scale-Free	Mesh	Random	Ring Lattice	Square Lattice	Torus	DAG	Fixed Degree	Star	Line	Time-Variant	Small World	Other/Note
[82]	X		X							X	X			
[83]	X			X						X				27 real-world topologies
[95]			X	X	X					X				Binary tree
[96]			X							X				
[97]												X		
[98]		X		X					X				X	

Note: Categories indicate topology classes evaluated within each paper. Ring lattice denotes regular nearest-neighbour graphs; scale-free denotes approximately power-law degree distributions; random denotes probabilistically generated graphs; small-world denotes clustered networks with long-range links. Papers may appear in multiple categories.

Table 2. Datasets used.

Author	CIFAR-10	CIFAR-100	F-MNIST	FEMNIST	MNIST	EMNIST	Pendigits	HAR	Full-HAR	Synthetic	Other/Note
[84]	X	X	X								
[75]	X			X	X						
[73]											Random geometric graph/autonomous data
[72]										X	
[85]	X		X		X					X	Two synthetic datasets
[86]							X	X	X		
[87]					X						Simplified MNIST
[88]	X	X									
[89]					X						IIoT scenario dataset
[90]			X		X	X					
[47]			X	X	X						MNIST digits
[91]	X				X					X	
[74]										X	
[79]	X				X						
[40]					X						
[92]			X								
[93]	X				X						So2Sat LCZ42
[94]					X						
[82]					X						Breast cancer dataset, TREC_6, cc sub
[83]	X		X		X						SVHN, ImageNet10, Malware
[95]	X										SST-5
[96]					X						Camelyon17-WILDS, ISIC19, Foetal Abdominal Structures, ChestXray
[97]											Cora, CiteSeer, PubMed, Amazon Photo, Amazon Computer, Coauthor CS, Coauthor Physics, ogbn-arxiv
[98]	X				X						Parkinson's disease

Table 3. Open issues identified in topology-related papers.

Author	Scalability	Communication Efficiency	Handling Heterogeneous Devices	Real-World Performance	Handling Node Drop-out	Fault Tolerance	Handling Non-IID Data	Other/Note
[84]	X	X	X					
[75]	X	X	X					
[73]	X			X	X	X		
[72]	X				X	X		
[85]	X	X			X	X		Impact of network latency
[86]	X	X	X			X		Concerns over single points of failure

Table 3. Cont.

Author	Scalability	Communication Efficiency	Handling Heterogeneous Devices	Real-World Performance	Handling Node Drop-out	Fault Tolerance	Handling Non-IID Data	Other/Note
[87]	X	X	X			X		Waveform complexity impacting performance
[88]	X		X			X		
[89]	X	X		X				
[90]	X	X					X	
[47]	X	X					X	
[91]		X					X	
[74]	X	X						
[79]	X							
[40]	X	X			X	X		
[92]	X	X			X	X		
[93]			X					Impact of network latency
[94]	X	X						
[82]			X	X			X	
[83]	X							
[95]	X	X	X	X			X	
[96]	X	X			X	X	X	
[97]	X	X					X	
[98]	X	X					X	

A recurring pattern across the topology literature is that connectivity improves convergence but creates communication and resilience trade-offs. Highly connected graphs can disseminate updates quickly and reduce disagreement between nodes, but they increase the number of communication paths that must be maintained. Sparse graphs reduce communication cost, but they can slow model propagation and make the network more sensitive to node drop-out, bottlenecks, or malicious neighbours. Figure 4 summarises this relationship schematically.

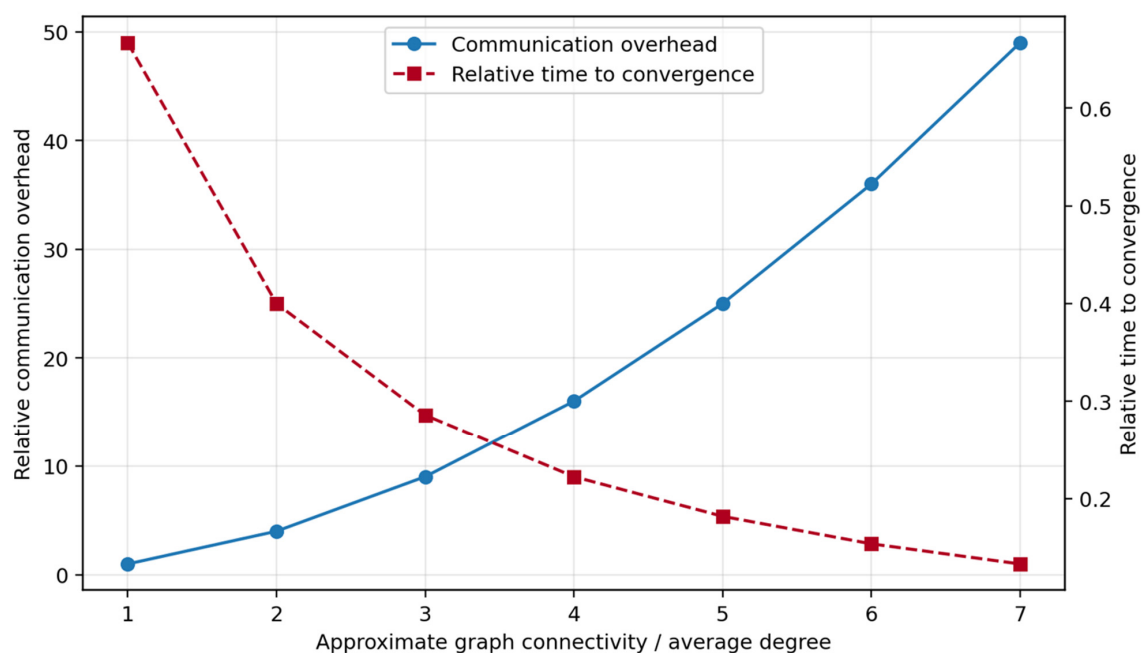


Figure 4. Conceptual relationship between graph connectivity, communication overhead, and convergence time.

The figure illustrates the qualitative trade-off observed across topology-related studies: increasing graph connectivity tends to reduce relative time to convergence but increases communication overhead. The plotted values are illustrative and are intended to summarise the direction of the trade-off rather than report directly comparable empirical measurements from a single experiment.

This trade-off explains why no single topology dominates across the reviewed literature. Fully connected and highly connected networks tend to support faster convergence, but they scale poorly in communication terms. Ring, sparse, random, and gossip-based structures reduce communication burden but may require more rounds to propagate updates and can be more vulnerable to local disruption. Adaptive and fixed-degree designs therefore represent an important middle ground because they allow for connectivity to be tuned.

3.2.1. Mesh

The topology in [75] consists of distinct silos that perform parameter transfer. Once parameters have been averaged across all members in the silo, they are further averaged with aggregated parameters from other silos. In this way, the model can make use of more plentiful bandwidth between local nodes, whilst also being able to aggregate across silos where needed. Aggregation is performed via a pair-wise gossip algorithm. More detailed information on gossip algorithms will be given in the Communication Algorithms section. In most gossip algorithms a peer is chosen at random, and weights exchanged between these two nodes. Across a non-specific number of rounds each node will eventually receive the parameters from each other node, at which point the local model is updated on all nodes. The authors assume that the gossip process occurs on a full mesh network (i.e., a ring lattice graph where $k = n - 1$). The number of communication rounds required to correctly average all weights on such a graph scale is calculated according to $O(\log(n))$, where n is the number of nodes. Where the topology is less connected, the number of communication rounds required can greatly exceed this bound. Further work is required to determine exactly what the lower bound is on an arbitrary graph. The authors also perform experiments where a percentage of clients drop out of the network. It is unclear from the paper whether communication links are rerouted to maintain a fully connected network, or if gaps are allowed to remain, but in-silo convergence takes longer, and the resulting model is less accurate. This would suggest that as in [84], the degree of connectivity is a key indicator of how a model will perform.

In [88], the authors model a fully connected (i.e., mesh) network over which they deploy an enhanced gossip protocol (Performance Enhanced Neighbour Selection) that allows for the identification of similar neighbours. The results in this paper are consistent with other mesh configurations, with performance across a network of fixed size improving in proportion to the size of the training set. Accuracy also improves, though more slowly, in proportion to the number of nodes in the network.

The authors in [91] discuss, amongst other topologies, a “fixed degree” network structure. It should be seen that the fully connected mesh topology represents a subset of the fixed degree network configuration, specifically where in a network with n nodes, each node is connected to every other node. Topologically there is also similarity with the Erdős–Rényi random graphs discussed below. The results suggest, as with other work, that the defining metric determining speed of convergence and algorithm performance is the degree of connectivity. As the degree of connectivity increases, convergence time is reduced, and algorithm performance improves. The downside is that communication overhead increases in proportion with the degree of connection. The most recent work keeps this trade-off central. Selo et al. [95] specifically tests the impact of network topo-

gies on decentralised federated learning and again indicate that increased connectivity improves model convergence but also increases communication burden and potential fragility when highly connected nodes become overloaded or unavailable. Konstantin and Mukhopadhyay [96] also argue against simply reaching for the most connected or most obvious topology, proposing a peer-to-peer framework in which clients select updates that are locally useful and trustworthy rather than relying on a fixed star-like arrangement.

In [74], the concept of “Age of Information” (AoI) is introduced. This is defined as a metric that indicates how far behind current each node is with respect to model updates. This metric is then used as a proxy for determining how scalable networks are. Evidence shows that in the fully connected topology, AoI scales in proportion to $O(\log n)$. This indicates that nodes should remain close to the current model status even as the network massively increases in size.

An important current work is [93], which takes a discussion of topology as its central theme. The authors discuss fully connected meshes, Barabási–Albert scale-free architecture, and fixed degree networks (each node is connected to exactly k neighbours). Further discussion of these latter types can be found in Sections 3.2.4–3.2.6. The paper tests various ML models across these architectures, using three distinct datasets. The results show that the fully connected mesh performs best for convergence speed, but the communications overhead is significant and scales rapidly with increasing numbers of nodes. Scale-free architecture, by contrast, presents communications efficiency, at a cost of resilience should one of the “power nodes” drop-out. A more balanced approach is given by the fixed degree networks. Here, the parameter, k , can be tuned to create a denser or more sparsely connected network, resulting in either enhanced performance or enhanced bandwidth use. Similar results can be seen in [94], where fully connected meshes and fixed degree networks with various values of k exhibit the same behaviour. This would suggest a more general pattern.

3.2.2. Ring Topology

In [85], three separate topologies are tested. These include a basic ring topology, as well as a torus topology, which expands the ring into three dimensions, allowing for connections to four direct neighbours rather than two. The authors also investigate a communication regime in which only subsets of nodes participate in each round, such that the union of communication patterns over time approximates a fully connected graph. Unlike the time-varying topologies discussed later in this section, the focus is on intermittent participation rather than dynamically changing network adjacency. Results from the ring topology test show slower model convergence as the network size increases. Additionally, comparison with a similar sized torus network shows improved performance on the latter, likely due to the increased connectivity between nodes.

Ring networks are also analysed in [91]. As with [85], the results show that the benefits are that no one node assumes an outsize importance in the network and that communication efficiency is high. The drawbacks are again slow convergence and reduced algorithm performance. Additionally, the authors found that maintaining a ring topology is difficult in a real-world scenario and especially as the network grows, as nodes leave the ring fractures into one or more chains, meaning that model updates cannot be transmitted until the ring is recreated.

As with the fully connected topology mentioned in Section 3.2.1, ref. [74] looks at the “Age of Information” (AoI) inside ring topologies. Evidence shows that this scales in proportion to $O(\sqrt{n})$. This indicates that the AoI increases much more quickly than in a fully connected graph. Empirical evidence suggests that an update that is generated in a node on one side of the ring will take $\frac{n}{2}$ hops to reach the other side of the ring. Further

investigation is required to determine if this result generalises where other communications algorithms are utilised.

A simple ring topology is utilised in [40]. Node updates are sent in a clockwise direction. This ensures communication between nodes is kept to a minimum, which is useful in scenarios where bandwidth is limited. As with other papers citing similar topologies ([99,100]), the drawbacks are considerable. Increased latency is evident as the network increases, with model updates taking longer to reach every node. Moreover, the network is vulnerable both to node drop-out and malicious nodes. Because of the minimal number of connections between nodes, it is difficult to reroute the flow of data should a node leave or become compromised in some way. Similarly, if nodes are mismatched in performance characteristics there exists the potential for considerable bottlenecks.

3.2.3. Directed Acyclic Graph

The authors of [47] discuss performance across three separate and distinct topologies, one of which is the Directed Acyclic Graph (DAG). The authors use the concept of eigenvector centrality to prioritise model updates sent from nodes with high degrees of connectivity. A high eigenvector-centrality value indicates both a high degree of connectivity and that the nodes it is connected to are also important. To generalise, a highly connected DAG is likely to have more important member nodes and so propagate model updates more quickly. When compared with results relating to other topologies, this is not unexpected.

Further work on DAG-based topologies can be found in [79]. Here, model updates flow through in an asynchronous manner up to a point where the model improvements are such that the model on the genesis nodes (those at the beginning of the graph) can be updated. As with [47], evidence shows that nodes with a higher degree of connectivity show faster convergence and contribute more to model improvements. This comes at the cost of enhanced bandwidth requirements. Experiments show that this requirement can quickly lead to bottlenecks, especially in networks containing heterogeneous client configurations.

3.2.4. Ring Lattice

The authors of [84] discuss the impact of various topologies on model convergence as well as performance deviation between nodes. This second factor is unique to decentralised federated learning. In a centralised network, the result of each training round is deployed to every node in the network. Thus, performance should be broadly the same across all nodes and vary only in accordance with the data tested against. In contrast, except in special circumstances, there will always be discrepancies in performance between the models deployed on each node in a decentralised network. The algorithm is tested with a 10-node network and models three ring lattices with a k of 2, 4, and 6, respectively.

When tested using the F-MNIST and CIFAR10 and CIFAR100 datasets, results show that the greater the degree of connectivity (i.e., the larger k is), the better the model performs. We also see a reduced variance between the highest and lowest performing nodes, indicating that increased connectivity improves model stability. The trade-off is of course the communication bandwidth requirements, with the R3 model ($k = 6$) transmitting and receiving three times the amount of data as compared to the R1 ($k = 2$) model.

Work in [73] focuses on a number of differing network topologies. The authors introduce the concept of R-Regular Ring Networks. In this paper, the authors look at the performance of a simple gossip algorithm across each topology, and their ring network results correspond closely to those in [84], with convergence time increasing as a function of the number of nodes but decreasing as a function of their average degree of connectivity.

3.2.5. Scale-Free

Scale-free networks are often cited as being common across disciplines, although works such as [101] dispute that. This may limit their usefulness in modelling real-world situations, although other research [102] has suggested that real-world data failing to meet a strict mathematical description is not an impediment to their use. The authors of [84] create example scale-free network topologies using the Barabási–Albert method [103] and compare their algorithm performance on the resultant graphs, as well as the ring lattice topology discussed in Section 3.2.4. The results suggest that at least for their algorithm, scale-free networks perform better than ring topologies. It also suggests that scale-free networks with a high degree of connectivity outperform those with a lower connectivity. These results are expected, and they suggest again that connectivity is one of the key metrics relating to algorithm performance. Further analysis suggests that the presence of a few highly connected nodes leads to a lack of resilience should node drop-out occur. Losing a widely connected node, especially in a scale-free network with fewer overall connections, would massively increase the model convergence time. The scale-free network approaches a semi-hierarchical state, with certain nodes taking on more importance inside the network.

Scale-free networks are discussed in [86], alongside small world and regular random graphs. Performance is consistent with the results described in [84], showing much quicker model convergence and improved accuracy over the other two topologies tested. As with other scale-free graph topologies, there may be a lack of resilience in networks where significant node drop-out or communication link failure is expected.

3.2.6. Other and Dynamic Topologies

As well as ring networks, the authors of [73] investigate algorithm performance across several other graph types, specifically small-world networks (as illustrated in Figure 1), random graphs, and prism networks. Small-world networks exhibit tight clusters of nodes together with a proportion of long-range connections that serve to improve the speed of dissemination between clusters.

Experimental results show that such networks demonstrate significant improvements in convergence time when compared to ring networks. This again implies that increasing connectivity is a key factor in improving network performance. The addition of long-range links does, however, duplicate one of the main concerns from the scale-free results, namely the creation of nodes that contribute an outsized amount to model convergence.

A newer direction is to make the topology adaptive rather than pre-selecting a fixed graph. Guo et al. [97] introduce DFed-SST, where semantic and structural features are used to build a topology for decentralised federated graph learning. This is a useful development because it recognises that in graph-learning settings, the local data already contains topology and that simply applying computer-vision-style decentralised optimisation may discard useful information. Guan et al. [98] move in a related direction by modelling decentralised federated learning through network propagation dynamics, treating each node as an independent entity inside the dynamics of the network. This is conceptually consistent with the wider conclusion of this section: the graph is not merely a transport layer but an active part of the learning process.

Along with small-world networks, the authors of [73] also investigate performance on random graphs and prism networks. It should be noted here that the random graph is not strictly random in accordance with the Erdős–Rényi [104] model. Instead, a fully connected base graph, such as a ring, is chosen, and then, edges between pairs of nodes are added with a probability of P , where $0 < P \leq 1$. The prism networks are formed of stacked rings of equal size, such that node n_k^0 (node k in ring 0) is connected to n_k^1 (node k in ring 1) and so on. The results here show that increasing the value of P results in

more connected and therefore more performant networks. For prism networks, increasing the number of nodes and the number of rings both negatively affect performance. Node drop-out increases convergence times in all cases, but random graphs with high degrees of connectivity perform better. Performance for prism networks decreases in relation to the dimensionality (i.e., number of rings) when nodes drop-out. This would suggest that prism networks are not desirable structures for real-world connectivity.

The authors of [72] test their communication algorithms on networks with widely varying topologies. The algorithms are discussed in Section 3.4. Testing was performed on several topologies, including random Erdős–Rényi graphs, square lattice graphs, and the Les Misérables graph [105]. Further analysis is performed at the subgraph level with independent edge sets, clique groups (fully connected subgraphs), and what the authors define as path gossip blocks, where nodes are connected in sequences of length n . As expected, the results show that the degree of connectivity is key to convergence speed. The random Erdős–Rényi graphs showed a direct correlation between the probability of an edge between nodes, p , and the algorithm performance.

A novel topology is introduced in [85]. Here, a subset of nodes form fully connected networks that, when summed over time, form a fully connected graph. This is particularly suited for environments where networks are subject to communication link failures or node drop-out. The results suggest that it provides a good balance of convergence speed and communication load.

In [86], the authors test both small-world topologies and regular random graphs. The latter exhibit a uniform degree (i.e., each node is connected to exactly n peers), but the connections themselves are assigned at random. As expected, the small-world topology exhibits much better performance than the regular random but is less efficient than the scale-free example (see Section 3.2.5). Where the regular random node graph is beneficial is ensuring that the communication load on each node remains broadly similar. More connected topologies tend towards the creation of “super nodes,” which must perform much more work than their peers. These create more significant points of failure. As such, it may be that scale-free topologies especially should be reserved for heterogeneous networks, where nodes with greater computing performance and bandwidth availability are able to act as super nodes. In low power, homogeneous networks a more balanced topology may be preferred.

A novel solution to weight transfer is presented in [87]. Here, a “topology-free” network uses waveform superposition in wireless signals to allow for the encoding and transfer of weights through interference patterns. In this scenario, wirelessly equipped IoT devices transmit their weights and rely on channel interference to combine them in a form that can then be decoded by other devices. Experimental evidence shows that the method works on a simplified MNIST identification task, where the aim is simply to identify 0 or 1. However, more complex scenarios are affected by non-constructive interference and are also reliant on topology assumptions that suggest this approach may not scale to real-world use.

The authors in [89] look at multi-hop networks. In this configuration, each node is connected to its nearest neighbours, forming a star topology with the node at the centre. Expanding this leads to a configuration with a “mesh-like” structure, which is fully locally connected but lacks the long-range connections found in a small world configuration. The results with both the MNIST and IIoT Scenario datasets used indicate that an increase in local connectivity (i.e., increasing the number of neighbouring nodes) improves both convergence speed and accuracy, although strong performance is seen with as few as $n = 2$ neighbours. It is apparent that greater connectivity implies improved performance. This appears to be algorithm-agnostic. The clustered solution uses groups of locally connected nodes that are then connected hierarchically to each other. Consensus is achieved locally

and only then are weights distributed between clusters. Indications are that clustered networks can deal with widely heterogeneous data better than the multi-hop solution, as local consensus mechanisms are able to stabilise local variations before sharing weights between networks. They are, however, slower at overall convergence than the multi-hop solution given the relative sparsity of long-range communication links.

A further example of Erdős–Rényi random graphs appears in [90]. The results are similar to those in [73], with the degree of connectivity highly indicative of the likely speed of convergence and accuracy of results, regardless of the algorithm used. Erdős–Rényi graphs in [90] are discussed in the context of a “social network” topology. Social network graphs are often cited as being real-world examples of scale-free networks, though as discussed in Section 3.2.5, this is disputed. Social networks demonstrate strong local connectivity, with some nodes also providing long-range communication links. Qualitative results indicate performance like the clustered network, with local consensus easily achieved, but consensus between clusters being much slower.

In [47] as well as arbitrary directed graphs, the paper also investigates the performance of star and time-variant star topologies. As might be expected, both show similar convergence properties to traditional federated learning. In fact, the only substantive difference is that the hub node in this case is not trusted. In particular the time-variant experiments mimic almost exactly those envisaged in [9], though with an untrusted hub node. The potential for significant degradation in speed of weight transfer exists should the hub node drop out, plus it also presents a tempting target for anyone looking to subvert the training process. Ultimately, this topology is likely to only be of use in minimal real-world cases given the similarity with traditional federated learning. Star topology is also investigated in [91], with results and concerns identical to those above.

The work in [92] expands [47] by considering more generic time-variant networks. These are constrained by two factors: a) a node must communicate with each of its neighbours within a fixed number of training rounds, and b) a path between any two nodes must exist at some point during the training process to allow model updates from node *a* to reach node *b*. Aside from these requirements, edges connecting nodes are free to alter on a time-variant basis. The results show improved communication usage when compared to a fully connected mesh network. The time variant graph is, however, more vulnerable to node drop-out, which is expected. Scalability is improved when compared to a full mesh but at the cost of resilience.

In [94], along with fully connected and fixed degree networks, the authors investigate star and line topologies. The star results mimic almost exactly those seen in [47]; with an untrusted node acting as coordinator, the network mimics a classic centralised federated learning configuration but without the benefits that provides. The line topology that is tested can be seen as conceptually identical to a ring topology that has suffered a node drop-out. The transfer of model weights is substantially hampered, and there are few real-world scenarios where this would be an appropriate topology.

The following conclusions arise from the study of the literature:

1. Scalability is often discussed as a concern, more specifically around determining whether the proposed solutions will work in real-world scenarios (see, e.g., [75,79,84]).
2. Communication efficiency and potential bottlenecks are also discussed, particularly whether nodes can communicate the required weight updates efficiently and without overburdening them. For instance, the reader can refer to [85,89,92].
3. Difficulty in handling heterogeneous devices is also raised as a concern. Most experiments are conducted with nodes that are identical in terms of memory, central processing unit (CPU) performance, bandwidth availability, etc. It is unclear if the proposed solutions still work when this is not the case, e.g., in [86–88,93].

4. Our literature review suggests that seven works were only evaluated on one specific network topology (see as examples [75,87–89]). As such, it is unclear if they can be generalised and be applied to the other topologies. Amongst the papers reviewed, only a small sample were tested on more than four topologies, again indicating a potential gap regarding generalisation.
5. Regarding node drop-out, mobility, and the impact of network latency, most solutions assume that nodes remain part of the network and remain in the same orientation to all other nodes during the entire training process. It is unclear whether the proposed solutions still work if this assumption is relaxed. Such concerns are explicitly raised in [73,85,92], amongst others. With respect to network latency, most solutions imply a fixed, always-available supply of network bandwidth. Should these connections become unstable, it cannot be determined if the proposed solutions still work, as discussed in [85,93].
6. When handling non-IID datasets, solutions assume a degree of similarity between the training datasets available to each node. Where this is not the case, it cannot be determined whether the proposed solutions are still valid. As indicated in the review, concerns around non-IID datasets are illustrated in [47,90,91].

3.3. Incentive Mechanisms

Many federated learning scenarios rely on third parties to train the model against their local data and share it either with a central server or directly with each other, depending on the architecture model chosen. It should be noted that this is not a prerequisite of federated learning; often, all the endpoints are controlled by the same entity. For example, in a factory setting, models may run on distributed sensors. These sensors are controlled by one entity, so there is no incentive requirement, and model updates are shared freely. However, in other scenarios, the end points are controlled by two or more entities, and so incentive mechanisms must be devised to ensure that updates are shared freely and contributing nodes are rewarded for their contribution to refining the global model. An example would be anonymised health data shared between hospitals and the patient themselves. In Table 4, we depict the incentive mechanisms that we extracted from reviewed papers, while Table 5 summarises the related datasets and Table 6 the identified open issues.

Table 4. Incentive mechanisms extracted from reviewed papers.

Author	Blockchain Smart Contract	Token Based Reverse Auction	Non-Tangible	Resource-Based	Multiple
[106]	X				
[107]					X
[108]	X	X			
[109]	X				
[110]	X				
[111]			X		
[54]				X	
[112]					X
[113]	X				X
[114]	X				X
[115]	X				X
[116]					X
[117]					X

Note: Categories indicate incentive mechanisms evaluated or materially used within each paper. Non-tangible incentives include reputation and trust-based approaches. Resource-based incentives reward contributed computation, storage, bandwidth, or data quality. Papers may appear in multiple categories.

Table 5. Datasets used in testing incentive mechanism-related papers.

Author	CIFAR-10	MNIST	F-MNIST	EMNIST	FEMNIST	Other
[106]	X		X			
[107]						X
[108]		X				
[109]	X	X	X	X		
[110]	X	X				
[111]	X	X	X			
[54]					X	
[112]						X
[113]	X					
[114]	X					
[115]		X				
[116]						X (theoretical only)
[117]	X	X		X		

Table 6. Open issues identified in incentive mechanism-related papers.

Author	Scalability	Privacy	Cost	Attack Resistance	Incentive Mechanism Optimisation	Dependency on 3rd Party Data	Handling Heterogeneous Nodes	Real-World/Large Dataset Performance	Other
[106]	X	X	X	X					
[107]	X	X	X		X		X		X
[108]	X	X			X				
[109]	X	X							X
[110]	X	X		X			X		
[111]		X	X		X			X	
[54]	X	X	X		X			X	
[112]	X		X		X		X	X	
[113]				X	X		X		X
[114]	X	X	X	X	X		X		
[115]	X			X	X	X		X	X
[116]	X		X	X	X			X	
[117]	X	X	X	X	X	X	X		

3.3.1. Blockchain Mechanisms

One of the key challenges of distributed federated learning around incentivisation is determining how to keep track of the incentives given and received. Many recent papers have chosen to use blockchain-based models to provide a record of each transaction. In [106], the authors combine their record-keeping and incentivisation mechanisms into a blockchain smart contract. Here, three distinct entities take part in their *Incentivised Decentralised Machine Learning* (iDML) process. A Learner initiates a task and pays for both the Neighbour and Validator nodes. The Neighbour provides data and resources to train the model, and the Validator confirms whether or not the training process improved model accuracy. Multiple Validators are supported to ensure consensus. iDML is designed to be plugged into existing peer-to-peer federated learning models, giving them the ability to pay their participants. However, significant challenges remain. For example, the smart contract is reliant on the ERC-20 standard and would require alteration to work with other blockchains. There are also major concerns about data privacy. For example, model weights are not written to the chain, but other identifiable data is, and this may allow reconstruction of participant interactions and model provenance. The reputation system used is also relatively simple, and collusion between Neighbour and Validator nodes may allow for the siphoning of tokens for model updates that provide marginal benefit or are in fact malicious. Finally, as with all proof-of-work blockchains, the energy requirement for contract operation is significant.

More recent work continues this line. Wu and Seneviratne [112] propose a blockchain-based framework that automates client registration, update validation, reward distribution, and global state tracking through smart contracts. The smart contract acts as a coordination mechanism without acting as a learning server. Tang et al. [113] take a game-theoretic route for blockchain-based federated learning, treating incentives as a stability problem rather than simply a payment problem. In repeated-round settings, incentives must encourage continued honest behaviour rather than one-off participation.

Liu et al. [114] move the same problem into semi-asynchronous blockchain-based federated edge learning. Edge servers, not only leaf clients, may require incentives where they contribute training, mining, validation, or aggregation resources.

Blockchain provides auditability and decentralised bookkeeping, but it does not, by itself, solve the problem of measuring contribution. If the evaluation metric is weak, then the system simply records weak incentives very reliably. Energy cost, gas cost, and smart contract complexity also remain practical concerns, especially where model training is performed on resource-constrained devices.

In [107], the authors discuss an array of studies pertaining to the healthcare sector and their use of various blockchain reward scenarios for different tasks. Contribution-based incentives are discussed in relation to data quality and diversity. Both improve model accuracy, so it makes good sense to reward those who provide both. Contributions are measured through the evaluation of model updates, with greater improvements leading to greater rewards. Specific metrics can also be applied when searching for specific datasets, for example, diabetes data or COVID-19 detections. Rewards are generally given in the form of generic “credits” that can then be converted to something tangible, such as computing resources or financial rewards. The authors also discuss reward distribution via blockchain smart contracts. These are rules that are triggered by pre-defined events. Smart contracts are particularly suited to fully decentralised systems, as they allow for automatic and fair distribution of rewards. This of course is reliant on the smart contract being written fairly; the ability of nodes to audit each contract is key. Reviewing previous work with respect to blockchain-native solutions, it appears that the consensus is that reward structures tailored towards pre-defined requirements can assist in ensuring that models receive the training data and resources they require.

A reverse auction, reputation dependent model is defined in [108]. In this work, a task is defined via specific data requirements, computing resources required, and total budget available. Based on this information, nodes containing training data or computing resources can offer these via a smart contract. Associated with each node is a reputation score that is either incremented or decremented depending on their previous contributions. Once all bids are submitted, winning nodes are selected according to their *unit reputation bid price*, which is the quotient of the bid price and the reputation score for each node. Nodes with the lowest values are selected in ascending order until all requirements are met. After the task is completed, nodes are rewarded. This reverse auction ensures the best value for money for the contracting entity. If nodes do not believe the requirements are worth their participation, they are free to decline to bid, potentially leaving the task unfulfilled. The reputation score element is designed to discourage malicious nodes, as decreases in their reputation score means a proportionally larger unit reputation bid price. However, an entity willing to price resources under the market rate can still influence events. The authors acknowledge that there is further work required in terms of the optimisation of the reputation mechanism. Wang and Zhao [115] extend the incentive discussion by treating federated learning on decentralised infrastructure as a reputation-based collaborative workflow. The framework extends reward allocation by incorporating committee selection, dynamic reputation, penalties, and smart-contract execution. This direction is promising

because it aligns more closely with how decentralised federated learning behaves in practice. Nodes enter with different data, different resources, and different willingness to contribute. The reward model therefore needs to handle heterogeneous and unreliable environments, not a clean laboratory setup where each node behaves identically.

In [109], the authors aim to ensure that nodes are rewarded in proportion to their contribution through the use of their *Peer Truth Serum for Federated Distillation (PTSFD)* algorithm. This is a two-stage process that requires nodes to train against their local datasets, then submit to the blockchain cryptographically hashed predictions, generated by testing against a shared public dataset. Once all predictions are written, a smart contract is used to compare publicly revealed results to the submitted hash. Rewards are then divided up according to their accuracy. A penalty factor for incorrect predictions ensures that nodes submitting random or wildly inaccurate data are not rewarded. The results suggest improved performance over the centralised baselines considered by the authors under their experimental settings, although the size of the improvement depends on the dataset, baseline, and configuration used. However, there are also potential drawbacks with the method. It is reliant on the existence of a shared, non-sensitive dataset that can be accessed by all nodes, which can be used to generate predictions. In the paper, the use of standard datasets, such as MNIST and CIFAR10, ensure this dataset is available. However, for non-laboratory cases using real-world data, such a shared dataset may not exist. This is especially likely in medical or banking settings, so the usefulness of the method may be limited. Coupled with this, the algorithm makes use of 1-bit soft labels. These allow for much more efficient use of bandwidth and storage, but it is a “lossy” method, with useful data potentially being lost, especially in edge cases where label attribution is less certain.

Finally, for blockchain solutions, the authors of [110] introduce *Fair Governance for Federated Learning (FGFL)*. Here, local training is performed and assessed against alignment with the optimal global gradient. Solutions that fall within a threshold on either side are rewarded, whilst those outside are discarded. This should ensure that malicious submissions are not incorporated into the initial model. It does also, however, run the risk of discarding legitimate updates that are in significant disagreement with the ideal gradient. This is most likely where non-IID datasets are in use, so care must be taken to ensure that the discard threshold is set at a suitable level. Uniquely this solution incorporates a reputation system with a temporal decay factor, meaning more recent behaviour is more highly weighted. This may potentially discourage long-term cooperation, so again, care must be taken when setting the decay value.

3.3.2. Other Mechanisms

Many recent papers discuss incentive mechanisms from the perspective of centralised federated learning (for example, [118–121]). It is to be expected that some of the algorithms suggested could also apply to decentralised federated learning, albeit with some adjustment. The key difference is the lack of a “controlling mind” in the distributed case that can assign rewards based on work completed and its quality. In a distributed environment, a blockchain smart contract can take on some of this responsibility. However, where blockchains are not used, further thought must be given. In [111], the reward mechanism is non-tangible and instead allows for nodes to compete in local clusters for the opportunity to take part in cluster to cluster communication, which ultimately forms part of the updates written to the global model for that iteration. This “survival of the fittest” interaction between local nodes avoids the need for costly node-to-node communication, as discussed in Section 3.2. The reward mechanism is relatively simple, with local nodes sharing their model updates with each other, utilising differential privacy techniques. A cosine similarity score is calculated, the highest scoring node is elected as cluster leader, and it is this device

only that shares its updates with other cluster leaders. This has the potential to impact both overall model performance as well as introducing security concerns. Similar issues may also occur if one node has data that is more particularly suited to the task than others. As with many of the other algorithms discussed in this paper, real-world performance has not been assessed. This should be a key driver of future research.

In [54], the incentive mechanism in question applies to Unmanned Aerial Vehicle (UAV) swarms. These autonomous vehicles are seeing increased use in tasks, such as military surveillance, crop irrigation and pest control, and smart city management. These tasks are ideally suited to distributed federated learning. However, as each individual UAV is resource constrained in both battery life and computing power, incentives are required to encourage them to take part in the learning process. This is accomplished via a three-stage Stackelberg game, where each node works in its own interest to discover the most efficient use of resources. Tasks are generated by a publisher, which also determines a resource allocation strategy. Tasks are then passed to clusters by way of a cluster head. These intermediaries are elected by cluster members and determine the unit price for contributions. Individual nodes in each cluster can then determine how much, if any, of their local resources to commit to the task. Clusters are determined by a cosine similarity measure, and cluster members are free to move between clusters after task completion. Clustering allows for a significant reduction in communication overhead via the creation of a semi-hierarchical structure. It is, however, reliant on the honesty of cluster heads as well as vulnerable to cluster head drop-out.

Reputation-based mechanisms attempt to solve the contribution measurement problem by taking historic behaviour into account. Some use blockchain-backed schemes to record behaviour, while others use a non-tangible reward structure. The practical attraction of reputation is obvious: a node that repeatedly provides useful updates should be trusted more than one that only occasionally contributes or whose updates damage convergence. The difficulty is that reputation can be gamed if contribution scoring is weak or if malicious nodes collude.

Shrestha [116] proposes a trust-based incentive mechanism for semi-decentralised federated learning, where trust scores are based on factors such as data quality, model accuracy, consistency, and contribution frequency. This is a useful extension of the reputation models already discussed because it makes the incentive mechanism explicitly multi-dimensional.

A related 2026 contribution, IMFLKD [117], applies incentives to decentralised federated learning with knowledge distillation. It shows that incentives are being adapted to communication-efficient learning paradigms, where what is shared may be predictions, logits, or distilled knowledge rather than full model weights.

Incentive mechanisms must be designed to encourage participation without being excessively generous. The recent literature suggests that the strongest designs are those that combine reward, reputation, and punishment, rather than treating incentives as simple payments. It is still unclear, however, whether these mechanisms scale to highly dynamic peer-to-peer networks where nodes join briefly, train opportunistically, and then disappear. That remains one of the key open questions.

Upon review, the following open issues are found:

1. The cost of blockchain operations is highly dependent on the blockchain used and has the potential to price clients out of the market. An interested reader can refer to [54,106,107,111] for in-depth discussion.
2. The potential exists for nodes to act collaboratively to skew bids and capture incentives. Attack resistance in this scenario is discussed in papers [106,110].
3. Whilst blockchain solutions have shown the ability to scale, it is unknown whether the smart contract solutions that sit on top of them can work at scale and at speed. As

with privacy concerns, this is a near universal point raised in the surveyed papers, with most papers expressing scalability concerns in some form.

4. Incentive mechanisms must be designed to encourage participation without being excessively generous. Useful discussion of these limitations is given in [54,107,108,111].
5. Many of the proposed solutions discussed assume that each node is of the same design. This may mean difficulty ensuring heterogeneous nodes can take part fairly. If this requirement is relaxed, care must be taken to ensure that low-powered nodes can still participate fairly and that high-powered nodes do not monopolise the incentives. Interested readers can refer to the concerns raised in [107,110].
6. Performance outside of laboratory conditions is not discussed, so real-world validation is key to determining the usefulness of the proposed solutions to real-world problems. Current limitations are further discussed in [54,111].

3.4. Communication Mechanisms

Within the scope of our review effort, we identified multiple communication strategies and related primitives as shown in Table 7, their respective datasets in Table 8 and the open issues in Table 9. As discussed in Section 3.2, communication efficiency is impacted considerably by the underlying topology. However, once the underlying topology is fixed, communication efficiency and performance are then managed by the communication algorithm chosen. Challenges arise from the absence of a controlling server, unlike in traditional federated learning. The often-ephemeral nature of node membership also impacts the performance of communication algorithms, as does the potential unreliability of bandwidth between nodes.

Table 7. Communication strategies and related primitives extracted from reviewed papers.

Author	Traditional Gossip	Prediction-Based Parameter Sharing	Parameter Optimised Gossip	Weight/Trust-Based Gossip	Sparsified Gossip	Multi-Step Gossip
[66]	X					
[84]		X				
[122]			X			
[90]				X		
[123]					X	
[124]					X	
[125]					X	
[71]	X					
[76]					X	
[126]	X					
[85]						X
[127]		X				
[128]	X			X	X	
[129]					X	X
[130]			X			
[131]						X
[132]			X			

Note: Categories indicate communication strategies evaluated or materially used within each paper. Parameter-optimised, weight/trust-based, sparsified, low-precision, prediction-based, and multi-step approaches are treated as broad communication strategy classes. Papers may appear in multiple categories.

Table 8. Datasets used in testing algorithm-related papers.

Author	CIFAR-10	CIFAR-100	MNIST	F-MNIST	EMNIST	IOT	Synthetic	Other
[66]			X					
[84]	X	X	X	X				
[122]	X							
[90]			X	X	X			
[123]								Twitter followers graph, Twitter document term graph, Altavista web graph,
[124]	X		X					
[125]						N-BaIoT, UNSW BoT-IoT		
[71]								No datasets used
[76]	X		X					
[126]								Spambase, Pendigits, HAR
[85]	X		X	X			Two synthetic datasets used	
[127]	X			X				
[128]	X		X	X				
[129]	X	X						
[130]								None reported
[131]			X					20 NG
[132]								ogbn-arxiv, ogbn-products, Reddit, ogbn-mag

Table 9. Open issues identified in algorithm-related papers.

Author	Scalability and Performance	Performance on Real-World Data	Dependence on Public Datasets	Balancing Compression and Accuracy	Working with Heterogeneous Devices	Working with Non-IID Data	Handling Dynamic Topologies	Handling Network Congestion/Failure
[66]	X							
[84]	X		X					
[122]	X			X	X			X
[90]	X					X	X	
[123]	X						X	
[124]	X					X	X	
[125]	X	X				X	X	
[71]	X						X	X
[76]				X	X	X		X
[126]	X			X		X		
[85]				X			X	X
[127]	X	X	X	X	X			
[128]	X			X	X	X		
[129]	X			X	X			X
[130]	X	X			X		X	X
[131]	X					X	X	X
[132]	X			X	X	X	X	X

The work in [66] is related to traditional federated learning, but the concepts explored are just as important for decentralised federated learning. The authors discuss the challenges faced when data is transmitted from edge nodes to coordinating server via noisy wireless channels. Neural networks are sensitive to noise when receiving parameter updates. The solution described looks at two specific scenarios: firstly, noise that conforms to statistical parameters and, secondly, noise that is random within a particular band of uncertainty. A “regularizer” function is added that helps by penalizing substantial changes in the loss function, which can be an indicator of excessive statistical variance. In essence, the model encourages robustness over accuracy when the noise variance is high. The results

of tests run against the MNIST dataset suggest that the proposed solution outperforms unaugmented federated learning, particularly in situations of excessive noise. Given what is known about real-world applications of distributed federated learning, further work seems necessary. Particular attention to the performance of data transmission under noisy conditions in decentralised networks may yield interesting results.

The authors of [84] take an unusual approach to the task of communication between nodes. In their work, rather than nodes sharing model weights, each node trains against its own local data. Once completed, the model then generates predictions against a common shared or public dataset. It is then these predictions that are shared with all one-hop neighbours. The local model is then updated via distillation, with the node aiming to minimise the distance between its predictions and the aggregated model. This presents several benefits. Firstly, the amount of data shared is greatly reduced in each communication round, with the results set forming a much smaller space than the parameter set. Secondly, privacy is much more easily preserved, with no weights being transferred, and the predictions developed against a transparent dataset. Finally, the method also supports heterogeneous model architectures during training. As the outcome is determined by fitting to results and not the transfer of parameters, there is no specific requirement that the same models run on each node. This would therefore allow low-powered nodes to interact and cross-train with more powerful devices. Testing against multiple common datasets, such as MNIST and CIFAR10, shows that the CMFD approach from Consensus-Based Distillation in Function Space performs better than the parameter-swapping baseline example. It also leads to greater homogeneity of performance between nodes, especially in weakly connected topologies. Extending this line of work, the authors of [127] introduce CE-ProxyFL, a proxy-based decentralised federated learning framework for wireless networks. This addresses a recurring tension in the literature: a node may need a model that generalises, but it may also need a model that remains personalised to its own data distribution. Proxy-based exchange reduces the need to transmit complete models whilst still allowing for useful cross-node learning.

Compression algorithms are discussed in [122]. Here, the focus is on the transfer of model weights in a compressed manner using a scheme called Low Huffman-coded Delta Quantization (LHDQ). The largest communications overhead in any type of federated learning is the transfer of weights, and LHDQ aims to reduce this overhead using Huffman encoding, where shorter codes are used for more frequent data. In addition, the algorithm uses the delta between parameter changes rather than the parameters themselves. Updates are generally smaller and often correlated, reducing the data load. Overall parameter updates are compressed to around 5/3 bits each. As discussed in [66], the authors apply their algorithm to wireless communication between nodes, testing using both fixed length and adaptive length time slots for transmission. The results suggest a significant reduction in the amount of data transferred, with only a minimal impact on model performance. The impact of noise on the encoding method is not discussed.

The focus in [90] is the optimisation of local model updates by differentially preferring weights from trusted partners. Local training occurs, and then, model parameters are sent to all one-hop neighbours. At this point instead of applying a simple or weighted average model update, the local model can favour updates provided by trusted nodes. Trusted partners may be nodes with strong ongoing data links, or those that have provided high-quality updates previously. Incorporating this trust mechanism allows for the tuning of model improvements based on the selected optimisation objective. This allows for the construction of models that are personalised to each node and more relevant to the data it possesses. Crucially, this factor can be dynamic, allowing for reputation or performance-based metrics as part of the model-sharing cycle. Testing against the MNIST,

Fashion-MNIST, and EMNIST datasets, the algorithm consistently outperforms baseline decentralised algorithms and comes within a few percent of centralised federated learning solutions. Communication efficiency is also markedly improved against the baseline algorithm with dramatic improvements in the number of rounds required to hit accuracy percentiles for each dataset.

Communication efficiency is a key topic, and in [123], the authors introduce Sparse Allreduce, an algorithm for dealing with power-law data, conceptually related to the power-law degree distributions discussed in Section 3.2.5. While the paper is not specifically aimed at peer-to-peer federated learning, it introduces the concept of sparsification as a key driver for communication efficiency. The algorithm concentrates only on the non-zero data points in their assigned set, and it is only these points that are transferred between nodes. Each node communicates in a hybrid round-robin/butterfly network, which balances speed of communication and reliability. It should be noted that the data transferred here are not weights but segments of distributed computing problems. However, the design of the network has clear impact on peer-to-peer federated learning tasks. Data aggregation and distribution and communication optimisation as demonstrated here are key metrics in every decentralised federated learning algorithm. In their work, Zhao and Canny discover that their algorithm becomes network bound at around 64 nodes, with inter-node communication taking up to 80% of available bandwidth. Whilst relating to a fixed topology and applicable only to power law data, this early paper presents a benchmark against which communications efficiency for peer-to-peer networks can be measured. The data here is already sparsified, and future efficiencies may be gained in distributed federated learning tasks through enhanced sparsification of transmitted weights. Work in [132] examines decentralised federated graph learning over non-IID data and jointly optimises network topology and graph sampling. Whilst also relating to topology, the central contribution is communication efficiency. Completion time and communication cost are reduced by exploiting the coupled relationship between graph sampling and the peer-to-peer topology.

More recent work deploys sparsification techniques as part of the FedOpt algorithm [124]. Enhanced privacy measures are also deployed, discussion of which falls outside the scope of this paper. The sparsification process deploys a sparsity fraction, q , which defines the proportion of gradients to retain. The algorithm then identifies the top $q\%$ of both positive and negative gradients. The mean of each set is taken, and the dominant mean determines which group is sparsified. This process is therefore able to tune the number of zeroed parameters that are transferred. The greater the sparsification is, the less data is transferred, as compression is greater. The downside to this is the loss of nuance in the transferred weights that may lead to reduced model accuracy. One key limitation with the paper is that the sparsification percentage remains constant and is independent of training performance and available network bandwidth.

Further work on sparsified communication can be found in [125]. Here, the authors use a variation on *TopK* sparsification. In classic *TopK* sparsification, only the top percentage of values are transferred, with the rest being zeroed. This is a simple algorithm to implement, but it can impact convergence speeds by repeatedly selecting the same parameters in consecutive training rounds. To combat this, the *Sparse Federated Aggregation (SparSFA)* algorithm implements a residual momentum function, which ensures smaller parameters are eventually included in the *TopK* picks. Xu et al. [128] address this limitation by combining model pruning and adaptive communication in decentralised federated learning. The framework avoids treating pruning as a fixed global value by adapting pruning and neighbour selection to device and communication constraints. He et al. [129] go further by jointly optimising routing and pruning in bandwidth-constrained multi-hop wireless

networks, connecting model size, routing path, latency, and convergence into a single communication problem.

The authors in [71] discuss the use of a gossip algorithm to average values across nodes in a pre-defined graph. The paper does not specifically address peer-to-peer federated learning, but the introduction of gossip algorithms is a key precursor for many subsequent works. The paper discusses how to optimise the transmission of values between nodes to reach consensus on every node. The convergence speed is found to relate to the second largest eigenvalue of the matrix, which defines inter-node communication properties. The largest eigenvalue represents the final state where each value is the same, whilst the second largest eigenvalue, λ_2 , represents the speed at which the steady state is approached. Large values of λ_2 indicate a significant memory of the initial state is retained, whilst smaller values of λ_2 imply a quicker dissipation. Results from the paper indicate that a key contributor to the value of λ_2 is the network's connectedness. A fully connected graph allows for quicker convergence, whilst in a sparse graph, the value of λ_2 remains large. This provides a theoretical underpinning for some of the results in Section 3.2. Interestingly, the paper also manages to show the equivalence between the performance of the gossip algorithm and the performance of random walks on the same graph, as λ_2 underpins them both. Much further research is suggested by the authors, for example, in the generalisation of the algorithm to incorporate other functions. It is in that work that gossip algorithms become useful in distributed machine learning research. The work in [130] provides a recent bridge between gossip- and graph-based approaches by proposing graph-based gossiping, using minimum spanning tree and graph colouring to optimise communication scheduling. The importance of this paper is that it treats communication as a network scheduling problem rather than only an algorithmic one and tests against real physical network configurations rather than relying only on simulation.

The algorithm detailed in [76] makes use of both sparsification and bandwidth awareness to improve both training speed and bandwidth utilisation. In contrast with established algorithms such as *FedAvg*, *GossipFL* transmits only a predetermined fraction of the model parameters. A binary mask is generated from a pre-shared random seed, ensuring that each peer generates the same mask. A random Bernoulli distribution is applied to the parameter set that ensures that only $\frac{1}{c}$ parameters are transmitted, with c being the compression ratio. In conjunction with this sparsification, the algorithm also monitors the network bandwidth available between each pair of nodes and prioritises faster connections whilst still making sure that nodes communicate with diverse peers across multiple training cycles. This results in faster communication and more efficient utilisation of networks. Combined these two enhancements mean that *GossipFL* can improve on baseline methods in terms of resource utilisation whilst also ensuring model convergence at a similar speed. There are, however, potential enhancements that may improve the algorithm further. Using a random mask to determine which parameters to send ensures fairness but does not consider that not all parameters are equal in terms of their contribution to model performance. Taking this fact into consideration when determining which parameters to send may lead to greater performance increases. Similarly, the level of sparsification is fixed regardless of the bandwidth available between nodes. A more advanced algorithm may look to implement some form of adaptive sparsification that is tied to the network resources available.

A comparison of traditional federated learning and gossip-based fully distributed learning is given in [126]. The paper looks at three less commonly used datasets, i.e., Spambase, Pendigits, and HAR, and compares the performance of both styles of federated learning. Additionally, three communication performance-enhancing techniques are also tested to see whether these improve data transmission overheads and convergence speed. As previous work has demonstrated, decentralised federated learning in this scenario

handles node drop-out better than the centralised model, but traditional topologies allow for faster convergence due to the actions of the aggregation server. Additionally, when data is extremely skewed across nodes, convergence in the traditional FL model far outperforms the distributed version. Convergence does, however, eventually occur. When a more realistic real-world scenario is modelled, where node drop-out occurs frequently, the gossip-based decentralised model increases in competitiveness as the rate of drop-out increases. In [131], the authors propose MAR-FL, a peer-to-peer federated learning system based on iterative group aggregation. The main significance is the stated reduction from the quadratic communication behaviour associated with naive peer-to-peer exchange to $O(N \log N)$ whilst retaining tolerance to unreliable clients.

In [85], gossip learning is expanded to encompass both multiple gossip steps and weight compression. Utilising a fixed communication budget, the algorithm, *DeLi-CoCo*, aims to minimise the training loss through balancing multiple gossip steps that contain compressed weights. The key insight is that this process is more bandwidth-efficient than a single gossip step per training iteration that transfers 100% of the weights. However, several practical considerations must be made prior to real-world use. For example, a good measure of network latency is necessary to determine the number of gossip steps per iteration. Care must also be taken when choosing the compression coefficient. If set too high, convergence may fail. Theoretical approaches may assist with initial parameter selection, but it is likely that real-world data will be required to determine the most efficient values. Whilst the paper presents a unique approach to decentralised federated learning, challenges remain. The application of the algorithm to heterogeneous networks, where nodes have significantly differing amounts of computing power, is not considered. Also untested are scenarios where inter-node bandwidth is non-homogeneous. Time-variant graphs, where nodes either leave voluntarily or drop out due to communication problems, are also not discussed.

The literature review indicates the following open issues:

1. Often the experimental design makes several assumptions that mean the results are artificial. Further work should be undertaken to assess solutions in real-world configurations. This is explicitly raised in [125].
2. Some solutions require an external public or shared dataset to train against (e.g., [84]). However, in many scenarios, such a dataset may not exist. Further work would either need to remove this limitation or find some way of creating the shared dataset.
3. Several papers identify the trade-off between model compression and accuracy after a fixed number of training rounds. Additional work is required to determine where the optimal balance lies. Additionally, the creation of more efficient algorithms should be encouraged. Interested readers should review [76,85,122,126].
4. As has been seen in previous sections, there is an assumption in these papers that nodes are comparable in performance across all parameters. Further work is required to test the robustness of solutions where this is not the case. Extended discussion can be found in [76,122].
5. Many papers indicate a theoretical limit on performance, and further work is required to overcome these. Scalability is one of the most commonly cited concerns, with most reviewed papers raising it explicitly.
6. Most solutions assume that an unchanging collection of nodes exists throughout the training process and that they remain in the same configuration with each other. Additional research is required to evaluate solutions where dynamic topologies are involved, as this is the second most cited concern. Further discussion can be found, for example, in [71,85,90,123]. Newer papers strengthen rather than weaken this criticism:

they show that real progress is being made but mainly by making the network layer more explicit rather than pretending it is a stable abstraction.

The dataset choices reported in Tables 2, 5 and 8 also reveal an important limitation in the current literature. Although the reviewed papers cover a range of topology, incentive, and communication settings, their experimental validation remains concentrated around a relatively small set of benchmark datasets. Figure 5 aggregates dataset usage across the three review dimensions.

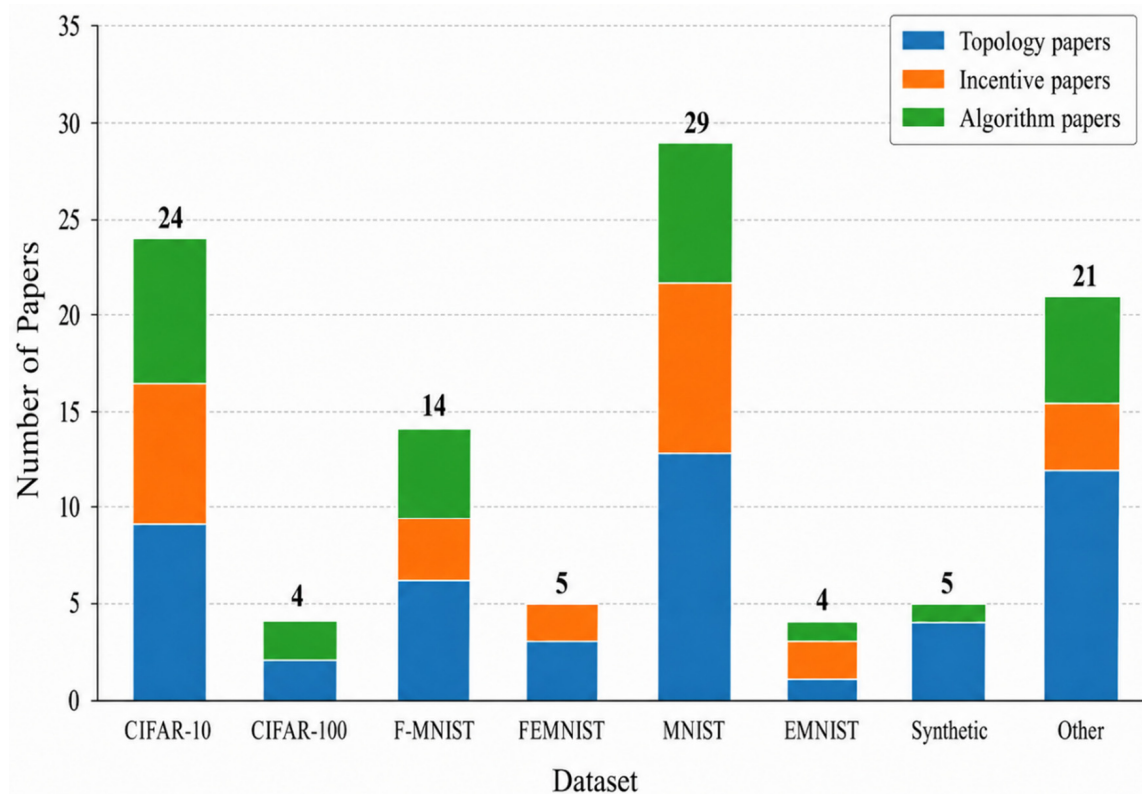


Figure 5. Dataset usage across reviewed papers.

The figure aggregates dataset use across topology-related, incentive-mechanism-related, and communication-algorithm-related papers. Counts indicate how often each dataset category appears in the reviewed studies, with stacked bars showing the contribution from each review dimension.

Figure 5 shows that MNIST and CIFAR-10 remain the dominant benchmarks, with F-MNIST also used frequently. This concentration is useful for comparability, but it limits claims about real-world deployment. Public image-classification benchmarks do not capture the full range of operational challenges faced by peer-to-peer federated learning systems, including heterogeneous devices, unstable bandwidth, privacy-sensitive local data, adversarial behaviour, or multi-gigabyte model-update transfer. The relatively frequent use of “Other” datasets is encouraging, but these datasets are heterogeneous and are not yet standardised enough to support robust cross-paper comparison.

The open issues identified across the reviewed papers are not evenly distributed. Some concerns appear repeatedly across topology, incentive mechanisms, and communication algorithms, while others are more specific to one strand of the literature. Figure 6 aggregates the open issue categories from the preceding tables to show which limitations recur most often.

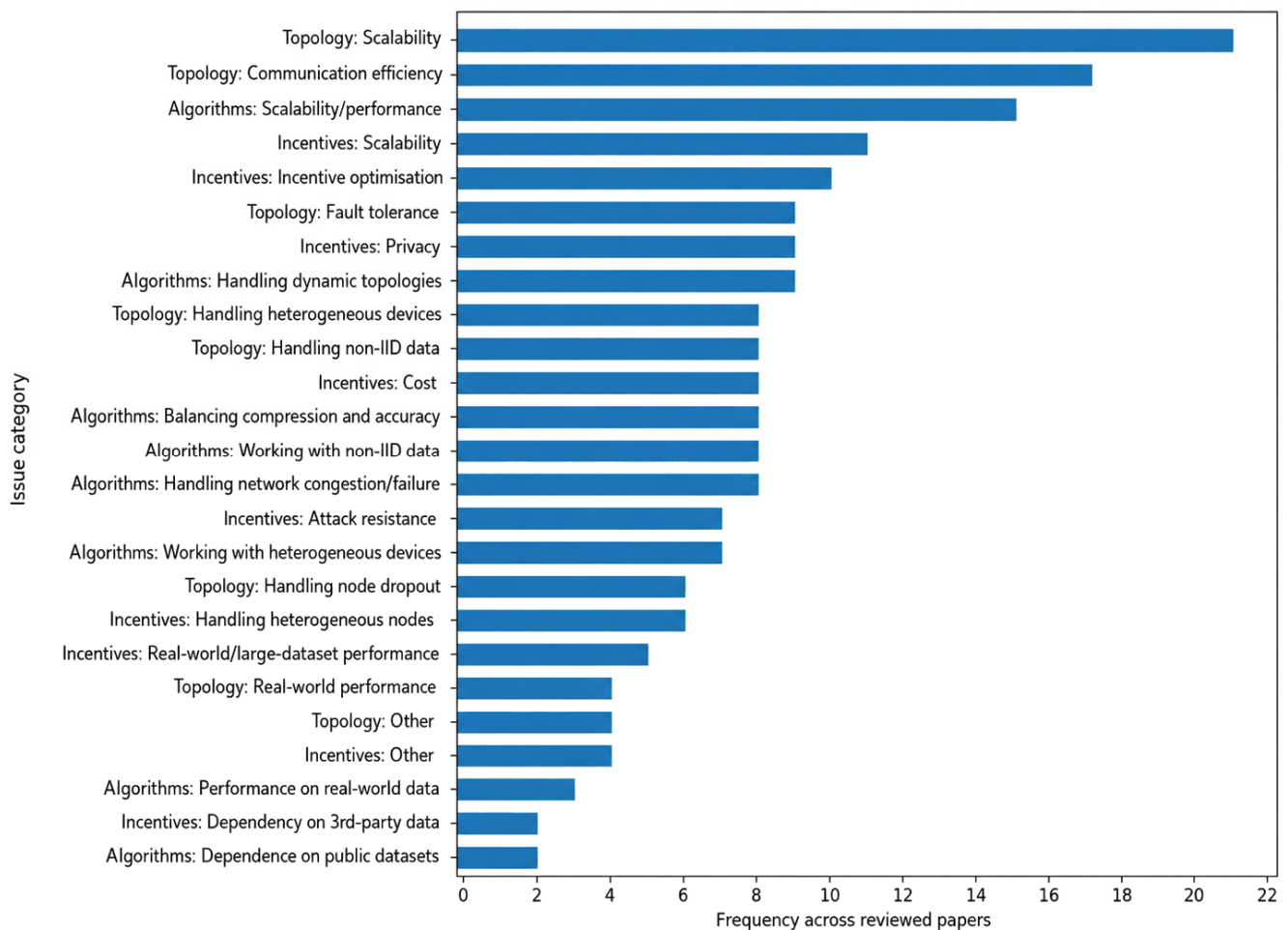


Figure 6. Open issue frequency across reviewed papers.

The figure aggregates open issues identified across topology-related, incentive-mechanism-related, and communication-algorithm-related papers. Frequencies indicate how often each issue category was identified in the reviewed literature.

Figure 6 shows that scalability and communication efficiency remain the most frequently identified concerns. This reflects a central challenge in peer-to-peer federated learning: removing the central server does not remove the need for coordination, communication, and update propagation but instead distributes these burdens across the participating nodes. The chart also shows that heterogeneous devices, non-IID data, dynamic topology, fault tolerance, privacy, and attack resistance recur across the literature. These issues indicate that the field's main challenge is not simply to improve model accuracy but to support learning under unstable, resource-constrained, and partially adversarial conditions.

The preceding tables identify which topologies, datasets, incentive mechanisms, and communication algorithms appear in the reviewed literature. However, coverage matrices alone do not show the scale of the experiments or the magnitude of reported improvements. Table 10 therefore summarises selected quantitative results from papers that report explicit endpoint values or clear within-paper deltas on broadly comparable learning tasks. The table should not be read as a leaderboard: the reviewed studies differ in model architecture, node count, topology, data partitioning, communication budget, baseline choice, stopping criteria, and whether the reported outcome is accuracy, convergence time, transmission time, communication volume, or privacy/security cost. The purpose is instead to show where

comparison is possible and why standardised peer-to-peer federated learning benchmarks remain necessary.

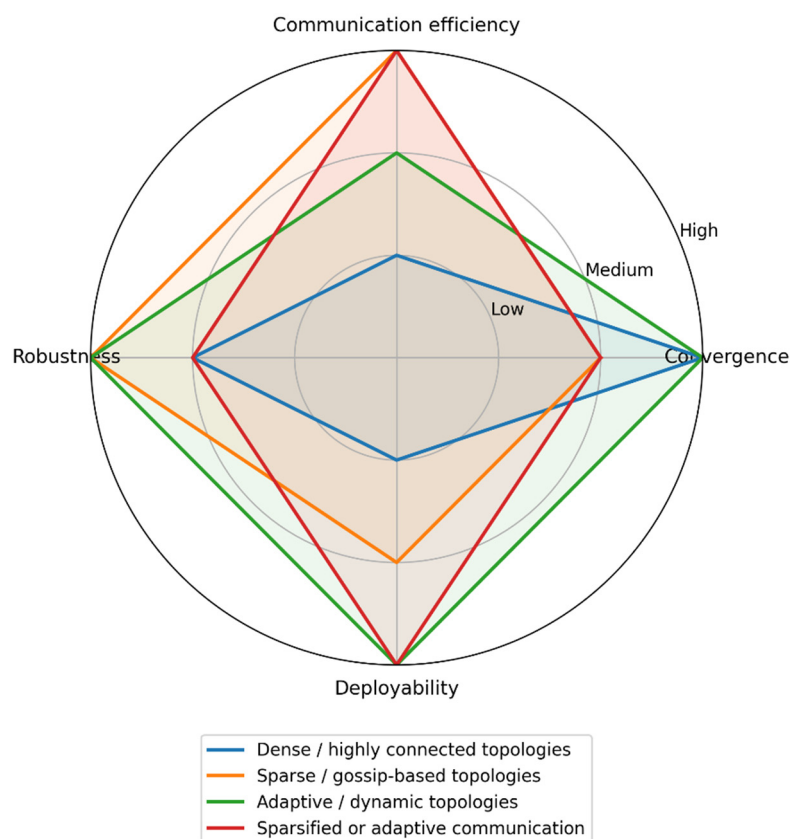
Table 10. Reported experimental characteristics and outcomes.

Ref.	Method/Focus	Dataset and Scale	Topology/Communication Setting	Nearest in-Paper Comparator	Reported Outcome	Best Reported Gain or Trade-Off	Key Insights
[75]	FedDual: pair-wise gossip with privacy/security variants	MNIST and CIFAR-10 with 100 clients; FEMNIST with 3550 clients	Hierarchical clusters with pair-wise gossip aggregation	Free-Noise, LDP-Noise, and PSI-Noise variants	Final accuracies: MNIST 98.92/98.21/98.88%; CIFAR-10 62.71/49.47/55.62%; FEMNIST 70.51/57.79/59.17% for Free-Noise/LDP-Noise/PSI-Noise	PSI-Noise recovers performance relative to LDP-Noise by +0.67 percentage points on MNIST, +6.15 on CIFAR-10, and +1.38 on FEMNIST	Privacy/security mechanisms can reduce accuracy, but mitigation mechanisms may recover part of the loss
[84]	CMFD: consensus-based distillation in function space	10 devices; CIFAR-10 and F-MNIST among evaluated datasets	Ring-lattice networks R1/R2/R3 and Barabási–Albert scale-free networks BA1/BA2.	Parameter averaging and PDMM baselines	On CIFAR-10 R1, CMFD 60.3%, parameter averaging 40.2%, PDMM 51.7%. On CIFAR-10 R3, CMFD 63.3%, parameter averaging 66.2%, PDMM 47.5%	CMFD is +20.1 percentage points over parameter averaging on sparse CIFAR-10 R1 but −2.9 points behind parameter averaging on more connected CIFAR-10 R3	Method performance is topology-dependent; a method may outperform in sparse networks but not in denser ones
[90]	DecDiff + VT: coordination-free decentralised learning over complex networks	MNIST, Fashion-MNIST, EMNIST	Complex decentralised networks under heterogeneous/non-IID conditions	Centralised benchmark, FED, DecHetero, CFA-GE	Average accuracies: centralised 0.9824/0.9181/0.9027; FED 0.9410/0.8939/0.8344; DecDiff + VT 0.9530/0.8904/0.8653 on MNIST/Fashion-MNIST/EMNIST	DecDiff + VT is +1.20 percentage points over FED on MNIST and +3.09 on EMNIST; it is statistically equivalent to FL/FED baseline on Fashion-MNIST according to the source paper	Decentralised methods can approach or exceed some FL baselines, but results depend heavily on heterogeneity and baseline definition
[122]	LHDQ: Low Huffman-coded Delta Quantization	5-client DFL; CIFAR-10; CNN with 545,098 parameters	Fully connected, line, and ring topologies; wireless channel settings	32-bit and 16-bit quantisation baselines	ALT-protocol average accuracies: LHDQ 72.0%, 32-bit 74.1%, 16-bit 73.9%	LHDQ loses about 1.9–2.1 accuracy points but reduces transmission time by 38.4–57.7% and transmitted bits by 9.3–18.5%	Communication savings can be substantial even when accuracy does not improve
[129]	Joint routing and model pruning for bandwidth-constrained multi-hop DFL	F-CIFAR10 with ResNet18 and F-CIFAR100 with ResNet34; 20-client default setting, with additional 5/10/20 and 50-client comparisons	Bandwidth-constrained multi-hop wireless networks	Unpruned systems and benchmark routing algorithms	Compared with no pruning, the proposed optimal pruning scheme improves testing accuracy by about 12% and reduces average model transmission time per client by about 27.8%; the paper also separately reports a 13% latency reduction in its contribution summary.	Proposed routing further improves accuracy by roughly 8% over benchmark routing algorithms	Bandwidth-aware routing and pruning jointly affect latency, convergence, and accuracy

Table 10 illustrates why direct ranking across peer-to-peer federated learning studies remains difficult. Even where common datasets such as MNIST, Fashion-MNIST, or CIFAR-10 are used, the papers differ in node count, topology, data partitioning, local model architecture, communication budget, failure assumptions, and whether the reported outcome is accuracy, loss, convergence time, communication volume, latency, or privacy/security cost. Nevertheless, several cross-paper patterns are visible. First, increased connectivity or topology-aware communication generally improves convergence or stability but usually increases communication burden. Second, compression, pruning, quantisation, proxy exchange, and graph-based scheduling can substantially reduce communication cost but often introduce trade-offs against accuracy, robustness, or additional assumptions such as public data availability. Third, results reported on realistic, physical, or structurally complex network settings remain relatively rare, with [130,132] being notable examples. Finally, even

strong within-paper gains are difficult to compare across papers because reported baselines and stopping criteria differ. This supports the need for standardised peer-to-peer federated learning benchmarks that specify datasets, node counts, topologies, data heterogeneity, communication budgets, drop-out/churn assumptions, and reporting metrics.

These patterns can be summarised as a set of recurring design trade-offs rather than as a simple ranking of methods. Figure 7 therefore provides a qualitative synthesis of how common peer-to-peer federated learning design choices tend to balance convergence, communication efficiency, robustness, and deployability.



Three-part qualitative scale used: Low = 1, Medium = 2, High = 3. Scores are interpretive synthesis indicators rather than directly comparable empirical measurements.

Figure 7. Qualitative trade-off radar for peer-to-peer federated learning design choices.

The figure summarises recurring design trade-offs across four properties: convergence, communication efficiency, robustness, and deployability. The three-point scale uses Low, Medium, and High ratings. The scores are interpretive synthesis indicators derived from patterns observed across the reviewed literature and should not be read as directly comparable empirical measurements.

Figure 7 extends the conclusion drawn from Table 10 by showing that peer-to-peer federated learning cannot be assessed using accuracy alone. Dense and highly connected topologies tend to favour convergence but impose higher communication cost. Sparse and gossip-based approaches improve communication efficiency and robustness but may slow convergence or produce uneven local models. Adaptive topologies and adaptive communication methods appear more deployable but introduce additional complexity around measurement, tuning, and stability. Future evaluations should therefore report not only accuracy but also communication volume, convergence time, node heterogeneity, churn tolerance, robustness under non-IID data, and the assumptions required for deployment.

4. Open Issues

As discussed in Section 3, many open issues remain across all three research areas. They fall into four recurring categories: deployment realism, communication scalability, incentive robustness, and comparability. Many distinct topologies are present in the literature, and evidence points to convergence speed being directly tied to the degree of connectivity between nodes. Initial configurations such as mesh, ring, and scale-free are easily modelled but do not represent realistic deployments. Thus, whilst results obtained under these conditions are useful in an illustrative sense, they should not be used in lieu of data taken from real-world scenarios. Many of the surveyed papers cite scalability as an open issue, and it is clear to see why when reviewing the most popular topologies. Mesh networks, as discussed in, for example, refs. [88,93,94] are fully fault tolerant, but the growth in complexity as nodes are added is quadratic. As systems scale to hundreds or thousands of nodes, an increasing proportion of each node's available resources must be dedicated to handling inter-node communication. This will directly impact the speed and quality of the results obtained during training. Scale-free networks such as [84,86] are more efficient than full mesh networks but suffer from the problem of "super-nodes". These nodes can act as a bottleneck for communication as the network grows. The potential impact on network performance should they drop out is significantly greater than for other nodes, making them a tempting target for attackers. Additionally, as scale-free networks become more connected, they become asymptotic to a full mesh network, with the communication overheads that this entails. Scale-free networks offer interesting performance metrics, but there are also potential pitfalls with their use that should be investigated. Ring and ring-lattice topologies as seen in [40,73,85] are much less impacted through network growth when communication requirements are considered. However, as the network scales, they suffer from degraded and slow convergence that severely impacts their usefulness. Other topologies discussed in Section 3.2 suffer from some or all of the same problems depending on their degrees of connectivity. The most promising target for future research into a universal solution would appear to be the random Erdős–Rényi topology, as discussed in [72,90]. By changing the probability of a node connection, a good approximation of other topologies can be made. Thus, an algorithm that performs well where $p = 1$ and $p = 0.1$ is likely to work well in real-world scenarios. Whether Erdős–Rényi or something else, designing topologies that maximise connectivity, efficiency, and performance remains a key challenge.

Similar to the challenge of topology design is the incorporation of methods to ensure fault tolerance and robustness to node drop-out. The real-world use cases for decentralised federated learning include those where bandwidth may be limited and at times unavailable. The nodes themselves may also be subject to power failure. Given the potential use cases can involve nodes being placed in out of the way areas, there is also potential for adversarial denial of service attacks that may go undetected for prolonged periods of time. The best-performing topology when discussing resilience is mesh. The loss of multiple nodes can be tolerated as the "all-to-all" configuration ensures that a propagation path between all nodes remains. The worst performing topologies are ring and line, where a significant impact on convergence speed and communication efficiency can be achieved through the drop-out of just a single node. In cases where redundancy is an absolute requirement, such as military or emergency service use, investigating full mesh networks may be justified. For all other scenarios, such redundancy is likely inefficient. The use of time-variant topologies, such as in [92], can improve resilience but require network reachability and introduce additional complexity. These challenges have not been adequately explored thus far. DAG-based solutions as described in [47,79] show potential but can become bottlenecked when nodes with high eigenvector centrality are removed. Further research into ensuring such

nodes and the “super-nodes” present in scale-free topologies are protected from drop-out may yield significant advancements. Experiments that simulate node drop-out have been performed in the literature, most notably in [75,84,88]. However, these experiments are limited in scope and apply only to the topologies discussed in each paper. New research that looks at the wider concept of recovery protocols and resilience is required. Multi-topology experiments may lead to the discovery of rules that can be applied on a broader scope. Additional work looking at Erdős–Rényi graphs may prove most fruitful.

One of the most pervasive of underlying assumptions across all the studies is the non-mobility of the nodes in the network. For many real-world scenarios, this is not a concern, but for applications such as drone swarms or mobile monitoring equipment, ensuring that communication and bandwidth remain available is key. As discussed above, time-variant topologies aim to model this type of scenario. However, the underlying assumptions still indicate that the network bandwidth between each node remains constant. There appear to be no recent studies that investigate the impact of node availability *and* variable bandwidth. The work on the *DeLi-CoCo* algorithm in [85] discusses some of the challenges related to ensuring convergence even in sparsely connected networks, but it requires tuning to ensure this happens. Either extending this work or investigating novel approaches is required to generalise the results obtained.

Many open issues related to the use of incentive mechanisms remain unresolved. At the most fundamental level, incentive mechanisms are produced to encourage data owners to participate in the training process. The criteria for providing rewards can have a significant impact on whether the training process fails or succeeds. As an example, if rewards are provided based solely on volume, nodes are incentivised to send as many updates as possible, regardless of quality. These low-quality updates will at best provide little benefit to the shared model and at worst may lead to a failure of convergence. An example of this type has previously been identified in [111], where the best-performing node is given the ability to update the global model. The papers reviewed cover a significant time span, so it is expected that much of the content will have been superseded by later research. However, concepts relating to auction types are still relevant and provide useful avenues of enquiry for future research around incentivising federated learning nodes. Techniques like Peer Truth Serum [109] aim to provide mitigation in this scenario but they are reliant on public test data that may not be available in every domain. Formal proofs ensuring that incentive mechanisms and beneficial model updates are aligned do not currently exist and should be explored further. Similarly, network owners must ensure that their rewards are pitched to capture all the relevant data the shared model requires. Data diversity and rarity are difficult to reward fairly, especially when those data points are part of a minority class. Dealing with data heterogeneity is a concern across all three areas of investigation in this paper. Any successful incentive mechanism must be able to deal with the unique challenges that non-IID datasets provide. This is a largely unexplored area. Tangentially connected is the need to ensure that any mechanism is also able to incentivise all participants fairly, regardless of their available resources. The “all or nothing” reward mechanisms seen in areas such as Bitcoin mining encourage nodes to commit all available resources to the task. It should be clear, however, that the highest-powered nodes are most likely to win in this scenario, potentially leading to inadvertent skewing of model parameters and the exclusion of potentially important data points from underpowered participants. A formal process for determining a fair reward system does not currently exist save for in a few specific scenarios, such as in [54]. Further research into whether such a process can be deduced would be of value. Finally, we see that most incentive mechanisms are designed with only the current training round in mind. For some tasks, this is reasonable. However, there are tasks such as security and health monitoring where continuity of cooperation

is potentially more important. In [110], the authors try to address this scenario with their decaying reputation score. However, the authors themselves acknowledge that further research in areas such as node participation and long-term incentivisation is needed.

The communication layer as reviewed in this paper also presents several open issues. Gossip-based and gossip-adjacent approaches remain common across the reviewed communication literature, but newer work increasingly combines these with pruning, compression, routing, scheduling, topology optimisation, and graph-sampling strategies. The lack of a central server, as seen in traditional federated learning, makes the coordination of updates extremely difficult. Gossip algorithms such as those in [71,75,76] simplify management at the expense of potentially duplicated updates and inefficient transfer of model weights. A key disadvantage of standard gossip algorithms is the use of random peer selection. This greatly simplifies the communication algorithm. The downside is that redundant communication is likely, and no consideration is given to the quality of updates that are *not* redundant. The Performance-Based Neighbour Selection (*PENS*) algorithm set out in [88] looks to ensure that nodes communicate preferentially with similar nodes through the comparison of training losses. This approach appears to be an outlier, and further work on preferential peer selection is required. Such algorithms need to consider factors including reliability, data value, and historical contribution. Doing so in a secure and auditable way in a decentralised network is a significant undertaking. Similarly, little work appears in the literature related to real-world performance on large networks. Papers such as [75,126] discuss results on networks of 100 nodes. In a smart city context or similar, the number of nodes is likely to be orders of magnitude higher. Gossip algorithms are slow to converge, and in real-world situations, on massive networks, nodes joining and leaving frequently may cause convergence to fail. More research is needed into large-scale performance, including networks with 1000 or more nodes, and into unstable networks where nodes join and leave frequently. The assumption of stable participation may be valid in some scenarios, but it is unlikely to hold in deployments where node drop-out is common.

Work into improving network performance through sparsification and bandwidth-aware communication is seen but is at a rudimentary stage. In [125], the authors discuss the benefits of sparsification and introduce their *SparSEA* algorithm, which allows for the transmission of a subset of model parameters based on those most impactful to the training round. However, the level of sparsification is constant for each node, and it makes no attempt at adjusting this value based on the available bandwidth or computing resources offered by the peer node. In [76], the algorithm is somewhat bandwidth aware. However, it does not apply adaptive bandwidth monitoring when determining the level of sparsification. Here, the model aims to over-utilise those edges that have high bandwidth connections, but the underlying assumption is that these will *remain* high bandwidth throughout the entire process. This assumption may be valid in some scenarios. However, in the real world, where node drop-out may be more common, it is likely to not hold true. Similarly, physical constraints such as bandwidth utilisation and interference for wireless connections are unlikely to remain constant and so should be considered by any advanced algorithm.

A further scaling issue concerns the size of the models themselves. Almost all evaluated systems use comparatively small models and benchmark datasets such as MNIST, Fashion-MNIST, and CIFAR-10. As foundation models continue to grow in size, peer-to-peer federated learning systems may be required to support models whose parameters are measured in hundreds of megabytes or even multiple gigabytes. Existing communication strategies such as gossip, sparsification, quantisation, and compression reduce communication overhead, but it remains unclear whether they can scale efficiently to foundation-model workloads operating across heterogeneous devices and wide-area net-

works. Future research should therefore investigate communication protocols, update compression techniques, topology adaptation strategies, and evaluation methodologies specifically designed for large-scale model transfer under realistic bandwidth constraints.

A recurring issue across the reviewed literature is that of comparability. When testing new algorithms, incentive mechanisms, or topologies, researchers do not currently design studies to enable direct comparison. Each reviewed paper uses a nearly unique combination of design choices and evaluation metrics. This makes it almost impossible to draw comparisons between research in anything other than a qualitative fashion. Most of the papers reviewed have produced significant amounts of numerical data, but even specialist analysis would struggle to extract directly comparable conclusions. Further work into the production of peer-to-peer specific datasets and networks of agreed size, performance, and topology should be encouraged. Only by allowing for ease of comparison between approaches can promising routes of enquiry be identified. At present, direct cross-study comparison remains extremely difficult.

5. Related Work

Security and privacy are important issues in decentralised frameworks. In traditional federated learning, it is assumed that the central server will have the ability to approve or reject clients attempting to join. In the decentralised world, no such authority exists, leaving space for bad actors to compromise user data and attack the model itself. In [75], the authors look to ensure both these threats are neutralised to the greatest possible extent. Firstly, the FedDual algorithm allows nodes to add noise to their gradients using a Gaussian distribution to ensure that sensitive information cannot be retrieved using model inversion attacks. A key trade-off here is between the amount of noise added and the speed of model convergence. Care must be taken to ensure that the Gaussian blurring of parameters does not lead to model convergence becoming impossible. Secondly, the algorithm also allows for signed updates. Weights are signed using a blind signature [133]. When the weights are sent to a neighbour, that node can confirm with the sender that it did in fact create the update about to be merged. This ensures that no unauthorised modifications have been made. Finally, Private Set Intersection (PSI) techniques are used to ensure that data points that overlap between two nodes do not have noise added. This can be performed in two separate ways. The most computationally efficient method is to compare hashes of data points to see if they match. If they do, then it can be surmised that they represent a value in common and so need not be included in the Gaussian masking process. A more secure but less computationally efficient method is to use public key encryption instead of hash values, as in the case of FedDual. It should be noted that both ways require significantly more inter-node communication, as hashes or encrypted values must be matched before weights are shared.

Previous survey papers have discussed distributed federated learning either as their main topic or as part of a broader work on federated learning of all types. Table 11 compares the present review with the most closely related decentralised federated learning surveys. The purpose of this comparison is not to suggest that earlier surveys omit future research challenges but to clarify the narrower contribution of this review: a focused comparison of topology, incentive mechanisms, and communication algorithms using aligned categories for mechanisms, datasets, experimental assumptions, and open issues.

Previous survey papers have discussed distributed federated learning either as their main topic or as part of a broader work on federated learning of all types. In [134], the authors provide a broad overview of decentralised federated learning, categorising prior work according to architecture, topology, security, privacy, and other dimensions. While comprehensive in scope, the review focuses primarily on taxonomy and classification rather

than comparative analysis across topology, incentives, and communication strategies. The research gaps identified include the lack of a standard framework for the evaluation of novel solutions as well as challenges with communication efficiency. The lack of robust security and privacy features for distributed federated learning is also noted. Further work in [135] draws similar conclusions whilst also raising some undiscussed points. Of all the models discussed in the paper, none are considered production ready. This indicates that most research is still in the laboratory stage. The authors also note that there are marked differences between the models in terms of underlying assumptions, topology, and evaluation settings. This makes direct comparison between models a challenging task. Whilst the production of new algorithms should be encouraged, care should be taken to ensure that the conclusions drawn from their testing can be generalised. A further survey on decentralised federated learning is published in [136]. As with the other surveys, there is perhaps an over-reliance on technical taxonomy when comparing frameworks. Real-world examples are also not evident, though theoretical examples are provided. The paper does identify key open issues such as the poor state of security and privacy protections inside distributed federated learning environments.

Table 11. Comparison with related decentralised federated learning surveys.

Work	Main Focus	Treatment of Topology	Treatment of Incentive Mechanisms	Treatment of Communication Algorithms	Treatment of Open Issues	How the Present Review Differs
[134]	Broad survey of decentralised federated learning fundamentals, frameworks, trends, and challenges	Discusses decentralised architectures, frameworks, and topology-related considerations at a broad level	Discussed as part of the wider decentralised FL landscape but not the central comparative focus	Discussed as part of decentralised FL systems and frameworks	Identifies general challenges including evaluation, communication efficiency, security, and privacy	Provides a broad foundational survey of decentralised FL. The present review is narrower but provides a more explicit cross-category comparison of topology, incentives, communication algorithms, datasets, and open-issue mappings.
[135]	Systematic literature review of decentralised and incentivised federated learning frameworks	Discusses assumptions and framework differences, including topology-related differences	Strong focus on incentivised and decentralised frameworks	Discussed where relevant to reviewed frameworks	Notes lack of production readiness and difficulty comparing models because assumptions, topology, and evaluation settings differ	Provides a systematic framework-level review. The present review builds on this by separating topology, incentive mechanisms, and communication algorithms into explicit comparative tables and by drawing cross-category implications.
[136]	Survey of decentralised federated learning	Provides taxonomy of decentralised FL approaches	Not the primary organising dimension	Discusses decentralised FL methods and technical categories	Identifies security, privacy, and other decentralised FL challenges	Provides a decentralised FL taxonomy. The present review places less emphasis on taxonomy alone and more emphasis on experimentally reported topologies, datasets, communication strategies, incentive mechanisms, and unresolved deployment barriers.
This review	Comparative survey of peer-to-peer federated learning across topology, incentives, and communication algorithms	Compares reviewed papers by topology class, dataset, and topology-related open issues	Compares incentive mechanisms by mechanism type, datasets, and unresolved issues	Compares communication algorithms by broad communication type, datasets, and unresolved issues	Synthesizes open issues across topology, incentives, and communication protocols	Provides a focused, table-driven synthesis of how topology, incentives, and communication algorithms interact in peer-to-peer federated learning and identifies the need for standardised benchmarks and realistic evaluation conditions.

Finally, the work in [137] discusses incentivisation in peer-to-peer networks more broadly rather than focusing solely on federated learning systems. In a wide-ranging literature review, the authors focus on three subgroups of incentive mechanisms. These are monetary incentives, or their tokenised equivalent, as demonstrated most clearly by Bitcoin mining. Reputation-based incentives are used to encourage positive behaviours such as collaboration amongst nodes. Finally, service-based incentives are rewards that grant immediate or preferential access to resources based on contribution to the task.

6. Conclusions

This paper reviewed recent work in peer-to-peer federated learning across three connected dimensions: topology, incentive mechanisms, and communication algorithms. Although these topics are often considered separately, the evidence reviewed here suggests that they are closely interdependent. Topology shapes the speed, reliability, and resilience with which model updates propagate through the network. Incentive mechanisms affect participation, trust, and the quality of contributed updates. Communication algorithms determine whether the resulting system can operate efficiently under practical bandwidth and resource constraints.

The topology analysis shows that peer-to-peer federated learning is shaped by a persistent trade-off between connectivity, convergence, and communication cost. Highly connected networks can improve update dissemination and reduce disagreement between nodes, but they also increase bandwidth demand and may create new points of fragility when highly connected nodes fail or become overloaded. More sparsely connected structures reduce communication overhead but can increase convergence time and sensitivity to node drop-out. No single topology therefore emerges as universally optimal. Instead, topology choice depends heavily on the deployment environment, expected churn, bandwidth availability, node heterogeneity, and tolerance for delayed convergence.

Incentive design presents a different but equally important constraint. The reviewed mechanisms range from blockchain-based rewards and auction systems to reputation, trust, and resource-based schemes. These approaches provide useful ways to encourage participation and discourage low-quality or malicious contributions, but comparison between them remains difficult because studies use different assumptions, evaluation methodologies, and experimental environments. Many mechanisms demonstrate promising behaviour under controlled conditions, but comparatively little evidence exists regarding their performance under large-scale deployment, heterogeneous resources, significant node churn, collusion, or adversarial participation.

Communication algorithms provide the main route for reducing bandwidth pressure in peer-to-peer federated learning. Gossip-based approaches remain prominent because of their simplicity and resilience to disruption, while more recent work has explored sparsification, prediction-based sharing, routing optimisation, trust-weighted aggregation, and multi-stage communication strategies. These techniques can reduce the volume or frequency of model exchange but typically introduce trade-offs relating to accuracy, convergence speed, robustness, implementation complexity, or dependence on assumptions such as stable links, public datasets, or homogeneous devices.

A recurring issue across all three dimensions is the difficulty of direct comparison between proposed approaches. Studies frequently employ different datasets, node counts, topologies, communication assumptions, failure models, and evaluation metrics. As a result, it is often possible to identify improvements within a specific experimental configuration but much harder to determine whether those improvements generalise across different environments. This lack of standardisation limits the strength of cross-paper conclusions and remains one of the main barriers to assessing progress in peer-to-peer federated learning.

Overall, peer-to-peer federated learning is a viable and increasingly mature alternative to centralised federated learning. Considerable progress has been made in designing topologies, incentive mechanisms, and communication strategies capable of supporting decentralised training. However, the reviewed work also highlights persistent challenges around realistic evaluation, heterogeneous environments, dynamic network conditions, adversarial behaviour, and comparability between studies. Addressing these issues will

be important in determining whether current approaches can move beyond controlled experimental settings and support large-scale real-world deployment.

Author Contributions: Conceptualization, A.A.; methodology, A.A.; software, A.A.; validation, A.A., A.M. and S.V.; formal analysis, A.M. and S.V.; investigation, A.A.; resources, A.A.; data curation, A.A.; writing—original draft preparation, A.A.; writing—review and editing, S.V., A.M. and N.P.; visualization, A.A. and A.M.; supervision, A.M. and S.V.; project administration, N.P.; funding acquisition, A.M. and N.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

References

1. Følstad, A.; Skjuve, M. Chatbots for customer service: User experience and motivation. In *Proceedings of the 1st International Conference on Conversational User Interfaces*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 1–9. [\[CrossRef\]](#)
2. Foltýnek, T.; Meuschke, N.; Gipp, B. Academic Plagiarism Detection: A Systematic Literature Review. *ACM Comput. Surv.* **2019**, *52*, 112. [\[CrossRef\]](#)
3. OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; et al. GPT-4 Technical Report 2024. *arXiv* **2024**, arXiv:2303.08774.
4. Team, G.; Anil, R.; Borgeaud, S.; Alayrac, J.-B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A.M.; Hauth, A.; Millican, K.; et al. Gemini: A Family of Highly Capable Multimodal Models. *arXiv* **2025**, arXiv:2312.11805.
5. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
6. Hoffmann, J.; Borgeaud, S.; Mensch, A.; Buchatskaya, E.; Cai, T.; Rutherford, E.; de Las Casas, D.; Hendricks, L.A.; Welbl, J.; Clark, A.; et al. Training Compute-Optimal Large Language Models. *arXiv* **2022**, arXiv:2203.15556.
7. Luccioni, A.S.; Viguiet, S.; Ligozat, A.-L. Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model. *J. Mach. Learn. Res.* **2023**, *24*, 1–15.
8. Kwon, W.; Li, Z.; Zhuang, S.; Sheng, Y.; Zheng, L.; Yu, C.H.; Gonzalez, J.; Zhang, H.; Stoica, I. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP '23*; Association for Computing Machinery: New York, NY, USA, 2023; pp. 611–626. [\[CrossRef\]](#)
9. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A.Y. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*; PMLR: Scottsdale, AZ, USA, 2017; pp. 1273–1282.
10. Konečný, J.; McMahan, H.B.; Yu, F.X.; Richtárik, P.; Suresh, A.T.; Bacon, D. Federated learning: Strategies for improving communication efficiency. *arXiv* **2016**, arXiv:1610.05492.
11. Yang, T.; Andrew, G.; Eichner, H.; Sun, H.; Li, W.; Kong, N.; Ramage, D.; Beaufays, F. Applied Federated Learning: Improving Google Keyboard Query Suggestions. *arXiv* **2018**, arXiv:1812.02903.
12. Chen, M.; Suresh, A.T.; Mathews, R.; Wong, A.; Allauzen, C.; Beaufays, F.; Riley, M. Federated Learning of N-gram Language Models. *arXiv* **2019**, arXiv:1910.03432.
13. Bagdasaryan, E.; Veit, A.; Hua, Y.; Estrin, D.; Shmatikov, V. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*; PMLR: Scottsdale, AZ, USA, 2020; pp. 2938–2948.
14. Muñoz-González, L.; Biggio, B.; Demontis, A.; Paudice, A.; Wongrassamee, V.; Lupu, E.C.; Roli, F. Towards Poisoning of Deep Learning Algorithms with Back-gradient Optimization. *arXiv* **2017**, arXiv:1708.08689.
15. Tolpegin, V.; Truex, S.; Gursoy, M.E.; Liu, L. Data Poisoning Attacks Against Federated Learning Systems. In *Computer Security—ESORICS 2020*; Chen, L., Li, N., Liang, K., Schneider, S., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 480–501. [\[CrossRef\]](#)
16. Li, S.; Cheng, Y.; Wang, W.; Liu, Y.; Chen, T. Learning to Detect Malicious Clients for Robust Federated Learning. *arXiv* **2020**, arXiv:2002.00211.

17. Cao, D.; Chang, S.; Lin, Z.; Liu, G.; Sun, D. Understanding Distributed Poisoning Attack in Federated Learning. In Proceedings of the 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS), Tianjin, China, 4–6 December 2019; pp. 233–239. [\[CrossRef\]](#)
18. Watcharapichat, P.; Morales, V.L.; Fernandez, R.C.; Pietzuch, P. Ako: Decentralised Deep Learning with Partial Gradient Exchange. In *Proceedings of the Seventh ACM Symposium on Cloud Computing, SoCC '16*; Association for Computing Machinery: New York, NY, USA, 2016; pp. 84–97. [\[CrossRef\]](#)
19. Isdal, T.; Piatek, M.; Krishnamurthy, A.; Anderson, T. Privacy-preserving P2P data sharing with OneSwarm. In *Proceedings of the ACM SIGCOMM 2010 Conference*; Association for Computing Machinery: New York, NY, USA, 2010; pp. 111–122. [\[CrossRef\]](#)
20. Kermarrec, A.-M.; Taïani, F. Want to scale in centralized systems? Think P2P. *J. Internet Serv. Appl.* **2015**, *6*, 16. [\[CrossRef\]](#)
21. Roy, A.G.; Siddiqui, S.; Pölsterl, S.; Navab, N.; Wachinger, C. Braintorrent: A peer-to-peer environment for decentralized federated learning. *arXiv* **2019**, arXiv:1905.06731.
22. Huang, Y.; Bert, C.; Fischer, S.; Schmidt, M.; Dörfler, A.; Maier, A.; Fietkau, R.; Putz, F. Continual Learning for Peer-to-Peer Federated Learning: A Study on Automated Brain Metastasis Identification. *arXiv* **2022**, arXiv:2204.13591.
23. Camajori Tedeschini, B.; Savazzi, S.; Stoklasa, R.; Barbieri, L.; Stathopoulos, I.; Nicoli, M.; Serio, L. Decentralized Federated Learning for Healthcare Networks: A Case Study on Tumor Segmentation. *IEEE Access* **2022**, *10*, 8693–8708. [\[CrossRef\]](#)
24. Friha, O.; Ferrag, M.A.; Benbouzid, M.; Berghout, T.; Kantarci, B.; Choo, K.-K.R. 2DF-IDS: Decentralized and differentially private federated learning-based intrusion detection system for industrial IoT. *Comput. Secur.* **2023**, *127*, 103097. [\[CrossRef\]](#)
25. Gu, T.; Dolan-Gavitt, B.; Garg, S. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. *arXiv* **2017**, arXiv:1708.06733.
26. Ji, Y.; Zhang, X.; Ji, S.; Luo, X.; Wang, T. Model-Reuse Attacks on Deep Learning Systems. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS '18*; Association for Computing Machinery: New York, NY, USA, 2018; pp. 349–363. [\[CrossRef\]](#)
27. Wang, H.; Muñoz-González, L.; Eklund, D.; Raza, S. Non-IID Data Re-Balancing at IoT Edge with Peer-to-Peer Federated Learning for Anomaly Detection. In *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*; Association for Computing Machinery: New York, NY, USA, 2021; pp. 153–163. [\[CrossRef\]](#)
28. Melis, L.; Song, C.; De Cristofaro, E.; Shmatikov, V. Exploiting Unintended Feature Leakage in Collaborative Learning. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 20–22 May 2019; pp. 691–706. [\[CrossRef\]](#)
29. Geiping, J.; Bauermeister, H.; Dröge, H.; Moeller, M. Inverting Gradients—How easy is it to break privacy in federated learning? *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 16937–16947.
30. 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, 20–22 May 2019; pp. 739–753. [\[CrossRef\]](#)
31. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [\[CrossRef\]](#)
32. Mhaisen, N.; Abdellatif, A.A.; Mohamed, A.; Erbad, A.; Guizani, M. Optimal User-Edge Assignment in Hierarchical Federated Learning Based on Statistical Properties and Network Topology Constraints. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 55–66. [\[CrossRef\]](#)
33. Ni, W.; Liu, Y.; Eldar, Y.C.; Yang, Z.; Tian, H. STAR-RIS Integrated Nonorthogonal Multiple Access and Over-the-Air Federated Learning: Framework, Analysis, and Optimization. *IEEE Internet Things J.* **2022**, *9*, 17136–17156. [\[CrossRef\]](#)
34. Liu, L.; Zhang, J.; Song, S.H.; Letaief, K.B. Client-Edge-Cloud Hierarchical Federated Learning. In Proceedings of the ICC 2020—2020 IEEE International Conference on Communications (ICC), Virtual, 7–11 June 2020; pp. 1–6. [\[CrossRef\]](#)
35. Luo, S.; Chen, X.; Wu, Q.; Zhou, Z.; Yu, S. HFEL: Joint Edge Association and Resource Allocation for Cost-Efficient Hierarchical Federated Edge Learning. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 6535–6548. [\[CrossRef\]](#)
36. Cao, Q.; Zhang, X.; Zhang, Y.; Zhu, Y. Layered Model Aggregation based Federated Learning in Mobile Edge Networks. In Proceedings of the 2021 IEEE/CIC International Conference on Communications in China (ICCC), Xiamen, China, 28–30 July 2021; pp. 1–6. [\[CrossRef\]](#)
37. Wang, B.; Fang, J.; Li, H.; Yuan, X.; Ling, Q. Confederated Learning: Federated Learning with Decentralized Edge Servers. *IEEE Trans. Signal Process.* **2023**, *71*, 248–263. [\[CrossRef\]](#)
38. Shi, Y.; Shen, L.; Wei, K.; Sun, Y.; Yuan, B.; Wang, X.; Tao, D. Improving the Model Consistency of Decentralized Federated Learning. *arXiv* **2023**, arXiv:2302.04083.
39. He, C.; Ceyani, E.; Balasubramanian, K.; Annavaram, M.; Avestimehr, S. SpreadGNN: Decentralized Multi-Task Federated Learning for Graph Neural Networks on Molecular Data. *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 6865–6873. [\[CrossRef\]](#)
40. Wang, Z.; Hu, Y.; Yan, S.; Wang, Z.; Hou, R.; Wu, C. Efficient Ring-Topology Decentralized Federated Learning with Deep Generative Models for Medical Data in eHealthcare Systems. *Electronics* **2022**, *11*, 1548. [\[CrossRef\]](#)

41. Hu, Y.; Zhou, Y.; Xiao, J.; Wu, C. GFL: A Decentralized Federated Learning Framework Based On Blockchain. *arXiv* **2021**, arXiv:2010.10996.
42. Liu, W.; Chen, L.; Zhang, W. Decentralized Federated Learning: Balancing Communication and Computing Costs. *IEEE Trans. Signal Inf. Process. Over Netw.* **2022**, *8*, 131–143. [[CrossRef](#)]
43. Tian, Y.; Wang, S.; Xiong, J.; Bi, R.; Zhou, Z.; Bhuiyan, M.Z.A. Robust and Privacy-Preserving Decentralized Deep Federated Learning Training: Focusing on Digital Healthcare Applications. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2023**, *21*, 890–901. [[CrossRef](#)] [[PubMed](#)]
44. West, D.B. *Introduction to Graph Theory*, 2nd ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2001.
45. Salama, A.; Stergioulis, A.; Hayajneh, A.M.; Zaidi, S.A.R.; McLernon, D.; Robertson, I. Decentralized Federated Learning over Slotted ALOHA Wireless Mesh Networking. *IEEE Access* **2023**, *11*, 18326–18342. [[CrossRef](#)]
46. Lalitha, A.; Kilinc, O.C.; Javidi, T.; Koushanfar, F. Peer-to-peer federated learning on graphs. *arXiv* **2019**, arXiv:1901.11173.
47. Wang, X.; Lalitha, A.; Javidi, T.; Koushanfar, F. Peer-to-Peer Variational Federated Learning over Arbitrary Graphs. *IEEE J. Sel. Areas Inf. Theory* **2022**, *3*, 172–182. [[CrossRef](#)]
48. Zeng, R.; Zeng, C.; Wang, X.; Li, B.; Chu, X. A Comprehensive Survey of Incentive Mechanism for Federated Learning. *arXiv* **2021**, arXiv:2106.15406.
49. Rozemberczki, B.; Watson, L.; Bayer, P.; Yang, H.-T.; Kiss, O.; Nilsson, S.; Sarkar, R. The Shapley Value in Machine Learning. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence, IJCAI-ECAI 2022*; International Joint Conferences on Artificial Intelligence Organization: Darmstadt, Germany, 2022; pp. 5572–5579. [[CrossRef](#)] [[PubMed](#)]
50. Fryer, D.; Strümke, I.; Nguyen, H. Shapley Values for Feature Selection: The Good, the Bad, and the Axioms. *IEEE Access* **2021**, *9*, 144352–144360. [[CrossRef](#)]
51. 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)*, Virtual, 2–4 December 2020; pp. 148–155. [[CrossRef](#)]
52. Wang, Z.; Hu, Q.; Li, R.; Xu, M.; Xiong, Z. Incentive Mechanism Design for Joint Resource Allocation in Blockchain-Based Federated Learning. *IEEE Trans. Parallel Distrib. Syst.* **2023**, *34*, 1536–1547. [[CrossRef](#)]
53. Li, C.; Song, M.; Luo, Y. Federated learning based on Stackelberg game in unmanned-aerial-vehicle-enabled mobile edge computing. *Expert Syst. Appl.* **2024**, *235*, 121023. [[CrossRef](#)]
54. He, W.; Yao, H.; Mai, T.; Wang, F.; Guizani, M. Three-Stage Stackelberg Game Enabled Clustered Federated Learning in Heterogeneous UAV Swarms. *IEEE Trans. Veh. Technol.* **2023**, *72*, 9366–9380. [[CrossRef](#)]
55. Gervais, A.; Karame, G.O.; Wüst, K.; Glykantzis, V.; Ritzdorf, H.; Capkun, S. On the Security and Performance of Proof of Work Blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS '16*; Association for Computing Machinery: New York, NY, USA, 2016; pp. 3–16. [[CrossRef](#)]
56. Qu, Y.; Gao, L.; Luan, T.H.; Xiang, Y.; Yu, S.; Li, B.; Zheng, G. Decentralized Privacy Using Blockchain-Enabled Federated Learning in Fog Computing. *IEEE Internet Things J.* **2020**, *7*, 5171–5183. [[CrossRef](#)]
57. Sethi, P. Reinforcement Learning Assisted Adaptive Difficulty of Proof of Work (PoW) in Blockchain-Enabled Federated Learning. *Virginia Tech.* 2023. Available online: <http://hdl.handle.net/10919/116016> (accessed on 4 August 2024).
58. Saleh, F. Blockchain without Waste: Proof-of-Stake. *Rev. Financ. Stud.* **2021**, *34*, 1156–1190. [[CrossRef](#)]
59. Bala, K.; Kaur, P.D. A novel game theory based reliable proof-of-stake consensus mechanism for blockchain. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e4525. [[CrossRef](#)]
60. Nguyen, C.T.; Hoang, D.T.; Nguyen, D.N.; Xiao, Y.; Pham, H.-A.; Dutkiewicz, E.; Tuong, N.H. FedChain: Secure Proof-of-Stake-Based Framework for Federated-Blockchain Systems. *IEEE Trans. Serv. Comput.* **2023**, *16*, 2642–2656. [[CrossRef](#)]
61. Wang, P.; Sun, W.; Zhang, H.; Ma, W.; Zhang, Y. Distributed and Secure Federated Learning for Wireless Computing Power Networks. *IEEE Trans. Veh. Technol.* **2023**, *72*, 9381–9393. [[CrossRef](#)]
62. Li, Z.; Yu, H.; Zhou, T.; Luo, L.; Fan, M.; Xu, Z.; Sun, G. Byzantine Resistant Secure Blockchain Federated Learning at the Edge. *IEEE Netw.* **2021**, *35*, 295–301. [[CrossRef](#)]
63. Qi, M.; Wang, Z.; Chen, S.; Xiang, Y. A Hybrid Incentive Mechanism for Decentralized Federated Learning. *Distrib. Ledger Technol.* **2022**, *1*, 4. [[CrossRef](#)]
64. Chen, M.; Shlezinger, N.; Poor, H.V.; Eldar, Y.C.; Cui, S. Communication-efficient federated learning. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2024789118. [[CrossRef](#)] [[PubMed](#)]
65. Sattler, F.; Wiedemann, S.; Müller, K.-R.; Samek, W. Robust and Communication-Efficient Federated Learning from Non-i.i.d. Data. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 3400–3413. [[CrossRef](#)] [[PubMed](#)]
66. Ang, F.; Chen, L.; Zhao, N.; Chen, Y.; Wang, W.; Yu, F.R. Robust Federated Learning with Noisy Communication. *IEEE Trans. Commun.* **2020**, *68*, 3452–3464. [[CrossRef](#)]
67. Asad, M.; Shaukat, S.; Hu, D.; Wang, Z.; Javanmardi, E.; Nakazato, J.; Tsukada, M. Limitations and Future Aspects of Communication Costs in Federated Learning: A Survey. *Sensors* **2023**, *23*, 7358. [[CrossRef](#)] [[PubMed](#)]

68. Shao, J.; Li, Z.; Sun, W.; Zhou, T.; Sun, Y.; Liu, L.; Lin, Z.; Mao, Y.; Zhang, J. A Survey of What to Share in Federated Learning: Perspectives on Model Utility, Privacy Leakage, and Communication Efficiency. *arXiv* **2024**, arXiv:2307.10655.
69. Blot, M.; Picard, D.; Cord, M.; Thome, N. Gossip training for deep learning. *arXiv* **2016**, arXiv:1611.09726.
70. Boyd, S.; Ghosh, A.; Prabhakar, B.; Shah, D. Gossip algorithms: Design, analysis; applications. In Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Miami, FL, USA, 13–17 March 2005; Volume 3, pp. 1653–1664. [\[CrossRef\]](#)
71. Boyd, S.; Ghosh, A.; Prabhakar, B.; Shah, D. Randomized gossip algorithms. *IEEE Trans. Inf. Theory* **2006**, *52*, 2508–2530. [\[CrossRef\]](#)
72. Haddock, J.; Jarman, B.; Yap, C. Paving the Way for Consensus: Convergence of Block Gossip Algorithms. *IEEE Trans. Inf. Theory* **2022**, *68*, 7515–7527. [\[CrossRef\]](#)
73. Dhuli, S.; Atik, F.; Chhabra, A.; Singh, P.; Cenkeramaddi, L.R. Performance Analysis of Gossip Algorithms for Large Scale Wireless Sensor Networks. *IEEE Open J. Comput. Soc.* **2024**, *5*, 290–302. [\[CrossRef\]](#)
74. Yates, R.D. The Age of Gossip in Networks. In Proceedings of the 2021 IEEE International Symposium on Information Theory (ISIT), Virtual, 12–20 July 2021; pp. 2984–2989. [\[CrossRef\]](#)
75. Chen, Q.; Wang, Z.; Wang, H.; Lin, X. FedDual: Pair-Wise Gossip Helps Federated Learning in Large Decentralized Networks. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 335–350. [\[CrossRef\]](#)
76. Tang, Z.; Shi, S.; Li, B.; Chu, X. GossipFL: A Decentralized Federated Learning Framework with Sparsified and Adaptive Communication. *IEEE Trans. Parallel Distrib. Syst.* **2023**, *34*, 909–922. [\[CrossRef\]](#)
77. Husnoo, M.A.; Anwar, A.; Haque, M.E.; Mahmood, A.N. Decentralized Federated Anomaly Detection in Smart Grids: A P2P Gossip Approach. *arXiv* **2024**, arXiv:2407.15879.
78. Nguyen, B.X.; Do, T.; Pham, V.; Tran, T.; Tjiputra, E.; Tran, Q.D.; Nguyen, A. Multigraph Topology Design for Cross-Silo Federated Learning. *arXiv* **2022**, arXiv:2207.09657.
79. Yu, G.; Wang, X.; Sun, C.; Wang, Q.; Yu, P.; Ni, W.; Liu, R.P. IronForge: An Open, Secure, Fair, Decentralized Federated Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *36*, 354–368. [\[CrossRef\]](#) [\[PubMed\]](#)
80. Bonawitz, K.; Eichner, H.; Grieskamp, W.; Huba, D.; Ingerman, A.; Ivanov, V.; Kiddon, C.; Konečný, J.; Mazzocchi, S.; McMahan, H.B.; et al. Towards Federated Learning at Scale: System Design. In *Proceedings of the Machine Learning and Systems*; Talwalkar, A., Smith, V., Zaharia, M., Eds.; PMLR: Scottsdale, AZ, USA, 2019; pp. 374–388. Available online: <https://mlsys.org/Conferences/2019/doc/2019/193.pdf> (accessed on 20 March 2026).
81. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and Open Problems in Federated Learning. *Found. Trends Mach. Learn.* **2021**, *14*, 1–210. [\[CrossRef\]](#)
82. Jiang, C.; Fan, J.; Halabi, T.; Haque, I. Performance Analysis of Decentralized Federated Learning Deployments. *arXiv* **2025**, arXiv:2503.11828.
83. Feng, C.; Gao, Y.; Celdran, A.H.; Bovet, G.; Stiller, B. from Models to Network Topologies: A Topology Inference Attack in Decentralized Federated Learning. *arXiv* **2025**, arXiv:2501.03119.
84. Taya, A.; Nishio, T.; Morikura, M.; Yamamoto, K. Decentralized and Model-Free Federated Learning: Consensus-Based Distillation in Function Space. *IEEE Trans. Signal Inf. Process. Over Netw.* **2022**, *8*, 799–814. [\[CrossRef\]](#)
85. Hashemi, A.; Acharya, A.; Das, R.; Vikalo, H.; Sanghavi, S.; Dhillon, I. On the Benefits of Multiple Gossip Steps in Communication-Constrained Decentralized Federated Learning. *IEEE Trans. Parallel Distrib. Syst.* **2022**, *33*, 2727–2739. [\[CrossRef\]](#)
86. Kavalionak, H.; Carlini, E.; Dazzi, P.; Ferrucci, L.; Mordacchini, M.; Coppola, M. Impact of Network Topology on the Convergence of Decentralized Federated Learning Systems. In *Proceedings of the 2021 IEEE Symposium on Computers and Communications (ISCC)*; IEEE: Athens, Greece, 2021; pp. 1–6. [\[CrossRef\]](#)
87. Michelusi, N. Decentralized Federated Learning via Non-Coherent Over-the-Air Consensus. In Proceedings of the ICC 2023–IEEE International Conference on Communications, Rome, Italy, 28 May–1 June 2023. [\[CrossRef\]](#)
88. Onoszko, N.; Karlsson, G.; Mogren, O.; Zec, E.L. Decentralized federated learning of deep neural networks on non-iid data. *arXiv* **2021**, arXiv:2107.08517.
89. Savazzi, S.; Kianoush, S.; Rampa, V.; Bennis, M. A Joint Decentralized Federated Learning and Communications Framework for Industrial Networks. In Proceedings of the 2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Virtual, 14–16 September 2020; pp. 1–7. [\[CrossRef\]](#)
90. Valerio, L.; Boldrini, C.; Passarella, A.; Kertész, J.; Karsai, M.; Iñiguez, G. Coordination-free Decentralised Federated Learning on Complex Networks: Overcoming Heterogeneity. *arXiv* **2023**, arXiv:2312.04504.
91. Wu, S.; Wu, S.; Huang, D.; Huang, D.; Wang, H.; Wang, H. Network Gradient Descent Algorithm for Decentralized Federated Learning. *J. Bus. Econ. Stat.* **2022**, *41*, 806–818. [\[CrossRef\]](#)
92. Zehtabi, S.; Hosseinalipour, S.; Brinton, C.G. Decentralized Event-Triggered Federated Learning with Heterogeneous Communication Thresholds. In Proceedings of the 2022 IEEE 61st Conference on Decision and Control (CDC), Cancún, Mexico, 6–9 December 2022; pp. 4680–4687. [\[CrossRef\]](#)

93. Badie-Modiri, A.; Boldrini, C.; Valerio, L.; Kertész, J.; Karsai, M. Initialisation and Topology Effects in Decentralised Federated Learning. *arXiv* **2024**, arXiv:2403.15855.
94. Lu, Y.; Yu, Z.; Suri, N. Privacy-preserving Decentralized Federated Learning over Time-varying Communication Graph. *ACM Trans. Priv. Secur.* **2023**, *26*, 1–39. [[CrossRef](#)]
95. Selo, R.; Kundroo, M.; Kim, T. Impact of network topologies on decentralized federated learning. *Clust. Comput.* **2025**, *28*, 996. [[CrossRef](#)]
96. Konstantin, M.; Mukhopadhyay, A. Don't Reach for the Stars: Rethinking Topology for Resilient Federated Learning. *arXiv* **2025**, arXiv:2508.05224.
97. Guo, L.; Yuan, Z.; Li, X.; Zhu, Y.; Qu, M.; Wang, W. DFed-SST: Building Semantic- and Structure-aware Topologies for Decentralized Federated Graph Learning. *arXiv* **2025**, arXiv:2508.11530.
98. Guan, B.; Yu, L.; Li, Y.; Jia, Z.; Hui, Z.; Jin, Z. Decentralized federated learning model based on network propagation dynamics. *Int. J. Mach. Learn. Cyber.* **2025**, *16*, 10889–10911. [[CrossRef](#)]
99. Han, L.; Huang, X.; Li, D.; Zhang, Y. RingFFL: A Ring-Architecture-Based Fair Federated Learning Framework. *Future Internet* **2023**, *15*, 68. [[CrossRef](#)]
100. Chung, W.-C.; Lin, Y.-H.; Luo, J.-A. Ring-Based Decentralized Federated Learning with Cosine Similarity Grouping. In Proceedings of the 2024 International Conference on Consumer Electronics—Taiwan (ICCE-Taiwan), Taichung, Taiwan, 9–11 July 2024; pp. 113–114. [[CrossRef](#)]
101. Broido, A.D.; Clauset, A. Scale-free networks are rare. *Nat. Commun.* **2019**, *10*, 1017. [[CrossRef](#)] [[PubMed](#)]
102. Holme, P. Rare and everywhere: Perspectives on scale-free networks. *Nat. Commun.* **2019**, *10*, 1016. [[CrossRef](#)] [[PubMed](#)]
103. Barabási, A.L.; Albert, R. Emergence of Scaling in Random Networks. *Science* **1999**, *286*, 509–512. [[CrossRef](#)] [[PubMed](#)]
104. Erdős, P.; Rényi, A. On the Evolution of Random Graphs. *Publ. Math. Inst. Hung. Acad. Sci.* **1960**, *5*, 17–61.
105. Peixoto, T.d.P. Lesmis—Les Misérables Coappearances. Available online: <https://networks.skewed.de/net/lesmis> (accessed on 5 October 2024).
106. Yu, H.; Chen, H.-Y.; Lee, S.; Vishwanath, S.; Zheng, X.; Julien, C. iDML: Incentivized Decentralized Machine Learning. *arXiv* **2023**, arXiv:2304.05354.
107. Myrzashova, R.; Alsamhi, S.H.; Shvetsov, A.V.; Hawbani, A.; Wei, X. Blockchain Meets Federated Learning in Healthcare: A Systematic Review with Challenges and Opportunities. *IEEE Internet Things J.* **2023**, *10*, 14418–14437. [[CrossRef](#)]
108. Ouyang, K.; Yu, J.; Cao, X.; Liao, Z. Towards Reliable Federated Learning Using Blockchain-Based Reverse Auctions and Reputation Incentives. *Symmetry* **2023**, *15*, 2179. [[CrossRef](#)]
109. Witt, L.; Zafar, U.; Shen, K.; Sattler, F.; Li, D.; Wang, S.; Samek, W. Decentralized and Incentivized Federated Learning: A Blockchain-Enabled Framework Utilising Compressed Soft-Labels and Peer Consistency. *IEEE Trans. Serv. Comput.* **2024**, *17*, 1449–1464. [[CrossRef](#)]
110. Gao, L.; Li, L.; Chen, Y.; Xu, C.; Xu, M. FGFL: A blockchain-based fair incentive governor for Federated Learning. *J. Parallel Distrib. Comput.* **2022**, *163*, 283–299. [[CrossRef](#)]
111. Kanagavelu, R.; Anil, C.G.; Wang, Y.; Fu, H.; Wei, Q.; Liu, Y.; Goh, R.S.M. Fed-SHARC: Resilient Decentralized Federated Learning based on Reward driven Clustering. In Proceedings of the 2024 IEEE Conference on Artificial Intelligence (CAI), Singapore, 25–27 June 2024; pp. 581–586. [[CrossRef](#)]
112. Wu, B.; Seneviratne, O. Blockchain-based Framework for Scalable and Incentivized Federated Learning. *arXiv* **2025**, arXiv:2502.14170.
113. Tang, W.; Liu, E.; Ni, W.; Qu, X.; Huang, B.; Li, K.; Niyato, D.; Jamalipour, A. Game-Theoretic Incentive Mechanism for Blockchain-Based Federated Learning. *IEEE Trans. Mob. Comput.* **2025**, *24*, 10363–10376. [[CrossRef](#)]
114. Liu, X.; Liu, J.; Wei, X.; Wang, Y. Incentive Mechanism Design for Semi-Asynchronous Blockchain-Based Federated Edge Learning. *ITU J. Future Evol. Technol.* **2025**, *6*, 119–131. [[CrossRef](#)]
115. Wang, Y.; Zhao, Z. Managing Federated Learning on Decentralized Infrastructures as a Reputation-based Collaborative Workflow. *arXiv* **2025**, arXiv:2502.20882.
116. Shrestha, A.K. Trust-Based Incentive Mechanisms in Semi-Decentralized Federated Learning Systems. *arXiv* **2026**, arXiv:2602.08290.
117. Ying, X.; Yan, K.; Gao, X.; Huang, J. IMFLKD: An incentive mechanism for decentralized federated learning based on knowledge distillation. *Sci. Rep.* **2026**, *16*, 10567. [[CrossRef](#)] [[PubMed](#)]
118. Xu, H.; Nanda, P.; Liang, J. Designing Incentive Mechanisms for Fair Participation in Federated Learning. In Proceedings of the 2023 IEEE International Conference on High Performance Computing & Communications, Data Science & Systems, Smart City & Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), Melbourne, Australia, 17–21 December 2023; pp. 98–105. [[CrossRef](#)]
119. Ali, A.; Ilahi, I.; Qayyum, A.; Mohammed, I.; Al-Fuqaha, A.; Qadir, J. Incentive-Driven Federated Learning and Associated Security Challenges: A Systematic Review. *TechRxiv* **2023**. [[CrossRef](#)]

120. 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.* **2022**, *10*, 1035–1044. [\[CrossRef\]](#)
121. Cong, M.; Yu, H.; Weng, X.; Qu, J.; Liu, Y.; Yiu, S.M. A VCG-based Fair Incentive Mechanism for Federated Learning. *arXiv* **2024**, arXiv:2008.06680.
122. Barhoush, M.; Ayad, A.; Kohankhaki, M.; Schmeink, A. Communication-efficient Decentralised Federated Learning via Low Huffman-coded Delta Quantization Scheme. In Proceedings of the 2024 International Wireless Communications and Mobile Computing (IWCMC), Ayia Napa, Cyprus, 27–31 May 2024; pp. 31–36. [\[CrossRef\]](#)
123. Zhao, H.; Canny, J. Sparse Allreduce: Efficient Scalable Communication for Power-Law Data. *arXiv* **2013**, arXiv:1312.3020.
124. Asad, M.; Moustafa, A.; Ito, T. FedOpt: Towards Communication Efficiency and Privacy Preservation in Federated Learning. *Appl. Sci.* **2020**, *10*, 2864. [\[CrossRef\]](#)
125. Wang, H.; Muñoz-González, L.; Hameed, M.Z.; Eklund, D.; Raza, S. SparSFA: Towards robust and communication-efficient peer-to-peer federated learning. *Comput. Secur.* **2023**, *129*, 103182. [\[CrossRef\]](#)
126. Hegedűs, I.; Hegedűs, I.; Danner, G.; Danner, G.; Jelasity, M.; Jelasity, M. Decentralized learning works: An empirical comparison of gossip learning and federated learning. *J. Parallel Distrib. Comput.* **2020**, *148*, 109–124. [\[CrossRef\]](#)
127. Park, J.; Kim, S.; Lee, J.; Niyato, D. Communication-Efficient Decentralized Federated Learning for Generalization and Personalization Over Wireless Networks. *IEEE Wirel. Commun. Lett.* **2025**, *14*, 4207–4211. [\[CrossRef\]](#)
128. Xu, Y.; Xiao, M.; Wu, J.; Gao, G.; Li, D.; Xu, H.; Zhang, T. Enhancing Decentralized Federated Learning with Model Pruning and Adaptive Communication. *IEEE Trans. Ind. Inf.* **2025**, *21*, 70–84. [\[CrossRef\]](#)
129. He, X.; Li, W.; Lv, T.; Yu, X. Joint Routing and Model Pruning for Decentralized Federated Learning in Bandwidth-Constrained Multi-Hop Wireless Networks. *arXiv* **2026**, arXiv:2603.15188.
130. Nguyen, H.; Nguyen, H.-T.; Donta, P.K.; Pirttikangas, S.; Lovén, L. Graph-based Gossiping for Communication Efficiency in Decentralized Federated Learning. *arXiv* **2025**, arXiv:2506.10607.
131. Mulitze, F.; Woisetschläger, H.; Jacobsen, H.A. MAR-FL: A Communication Efficient Peer-to-Peer Federated Learning System. *arXiv* **2025**, arXiv:2512.05234.
132. Wang, S.; Liu, J.; Xu, H.; Tang, C.; Ma, Q.; Huang, L. Towards Communication-Efficient Decentralized Federated Graph Learning over Non-IID Data. *arXiv* **2025**, arXiv:2509.08409.
133. Chaum, D. Blind Signatures for Untraceable Payments. In *Advances in Cryptology*; Chaum, D., Rivest, R.L., Sherman, A.T., Eds.; Springer: Boston, MA, USA, 1983; pp. 199–203. [\[CrossRef\]](#)
134. Beltrán, E.T.M.; Pérez, M.Q.; Sánchez, P.M.S.; Bernal, S.L.; Bovet, G.; Gil Pérez, M.; Pérez, G.M.; Celdrán, A.H. Decentralized Federated Learning: Fundamentals, State of the Art, Frameworks, Trends, and Challenges. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 2983–3013. [\[CrossRef\]](#)
135. Witt, L.; Heyer, M.; Toyoda, K.; Samek, W.; Li, D. Decentral and Incentivized Federated Learning Frameworks: A Systematic Literature Review. *IEEE Internet Things J.* **2023**, *10*, 3642–3663. [\[CrossRef\]](#)
136. Gabrielli, E.; Pica, G.; Tolomei, G. A Survey on Decentralized Federated Learning. *arXiv* **2023**, arXiv:2308.04604.
137. Ihle, C.; Trautwein, D.; Schubotz, M.; Meuschke, N.; Gipp, B. Incentive Mechanisms in Peer-to-Peer Networks—A Systematic Literature Review. *ACM Comput. Surv.* **2023**, *55*, 1–69. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.