IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 21 May 2023, accepted 1 June 2023, date of publication 9 June 2023, date of current version 11 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3284976

SURVEY

Decentralized Machine Learning Training: A Survey on Synchronization, Consolidation, and Topologies

QAZI WAQAS KHAN[1], ANAM NAWAZ KHAN[1], ATIF RIZWAN[1], RASHID AHMAD[2,3], SALABAT KHAN[2,3], AND DO-HYEUN KIM[1]

[1]Department of Computer Engineering, Jeju National University, Jeju-si, Jeju Special Self-Governing Province 63243, Republic of Korea
[2]Department of Computer Science, COMSATS University Islamabad, Attock Campus, Attock 43600, Pakistan
[3]Bigdata Research Center, Jeju National University, Jejusi, Jeju Special Self-Governing Province 63243, Republic of Korea

Corresponding author: Do-Hyeun Kim (kimdh@jejunu.ac.kr)

This work was supported by the Research Grant of Jeju National University, in 2022.

ABSTRACT Federated Learning (FL) has emerged as a promising methodology for collaboratively training machine learning models on decentralized devices. Notwithstanding, the effective synchronization and consolidation of model updates originating from diverse devices, in conjunction with the appropriate configuration of network topologies, persist as crucial obstacles. This paper provides a comprehensive analysis of the current techniques and methodologies utilized in the synchronization, consolidation, and network topologies of Federated Learning. The present study explores diverse synchronization strategies utilized for the purpose of coordinating model updates from geographically distributed cross-silo edge nodes. The study takes into account several factors, including communication efficiency and privacy preservation. This study delves into the intricacies of model consolidation techniques, such as weighted and personalized aggregation methods, to evaluate their efficacy in consolidation of local model updates into a global model, while taking into consideration statistical heterogeneity and resource constraints. In addition, an examination is conducted on the importance of network topologies in Federated Learning (FL), taking into account their influence on communication efficacy, confidentiality, expandability, resilience, and resource allocation. The survey assesses and contrasts the efficacies and constraints of extant methodologies, discerns deficiencies in present investigations, and provides insights for future progressions. The objective of this survey is to provide a thorough examination of FL synchronization, consolidation, and network topologies, with the intention of offering a valuable reference for individuals engaged in Federated Learning, including researchers, practitioners, and stakeholders. This survey aims to support the advancement of more effective and resilient FL systems.

INDEX TERMS Federated learning, synchronous, asynchronous, semi asynchronous weight aggregation, network topology.

I. INTRODUCTION

The proliferation of Internet of Things (IoT) devices has resulted in a substantial volume of data being produced by the corresponding physical IoT networks. Various kinds of devices, including wearable devices, smartphones, and smart home IoT systems, produce enormous amount of data [1]. The conventional approach involves the transfer of data obtained from these devices to a centralized cloud for the purpose of training machine learning models, which are utilized to extract valuable insights from the data and facilitate decision-making [2]. Nonetheless, the transmission of data to a centralized cloud or server raises concerns regarding the privacy of data sharing [3]. To address the privacy concern at hand, it is recommended to maintain the confidentiality of data through decentralization and to conduct collaborative

The associate editor coordinating the review of this manuscript and approving it for publication was Daniel Augusto Ribeiro Chaves.

training of machine learning models through distribution, as suggested in reference [4]. FL is a novel approach that was initially proposed by Google to facilitate the training of a machine learning model on distributed data while ensuring data privacy [5]. FL aims to facilitate the training of a global model across numerous devices while safeguarding the confidentiality of the personal data of each device, as stated in reference [6]. FL involves training a local model on individual devices and exchanging their knowledge through weights or gradients during each communication round. This facilitates local model aggregation, which ultimately leads to the development of a generalized global model [7]. Numerous limitations arise while performing FL on devices with limited resources [8]. The majority of these intelligent devices exhibit limited computational capacity, thereby necessitating a longer duration for training a local model. The FL system is faced with several challenges, namely unreliable heterogeneous devices, data heterogeneity, and heterogeneous computing resources, as noted in a scholarly source [9]. Due to device heterogeneity, the central server awaits the update of the local model, as some devices exhibit a shorter duration for training a local model, while others require a longer duration [10].

In order to establish effective communication between local devices and a central server, it is necessary to devise an efficient and robust synchronization method that can enhance communication efficiency and minimize the waiting time of the central server, as suggested by [11]. The presence of statistical heterogeneity among clients can impact collaborative training.

Various synchronization techniques have been employed in scholarly works to facilitate communication between the central server and the local client [12]. [13], [14] This paper will examine in detail the synchronization strategies in FL techniques employed in facilitating communication between clients and servers. Specifically, this method involves the server awaiting the arrival of all clients prior to executing the aggregation process. One limitation of the synchronous approach is the presence of a straggler or a worker with slow performance, which causes the server to wait for the completion of the parameter aggregation process for all models. The asynchronous synchronization technique, wherein the server does not wait for all clients, is discussed in the works of [15] and [16]. In this method, the client finishes the local training and sends a wait signal to the server. This approach enables concurrent execution of communication and computation. Nevertheless, the substantial update of the global model results in increased consistency with respect to the local minima, however at the cost of decreased convergence speed. The semi-synchronous approach is utilized, which effectively addresses the limitations of both synchronous and asynchronous methods while simultaneously providing the benefits of both [17], [18].

In FL, an additional challenge to overcome, is the uneven distribution of data among clients. Some clients may have non-IID (non-independent and identically distributed) data, which can adversely affect the effectiveness of the global model. The significance of network topology in federated learning cannot be ignored. In this context, the present survey also provides a comprehensive examination of the aggregation techniques and network topologies that are currently being utilized.

Numerous surveys have been conducted to examine and evaluate the contemporary techniques of synchronization, network topology, and aggregation method. However, there exist certain limitations. The study referenced by [19] centers solely on the asynchronous method and incorporates a limited number of parameters for the purpose of analyzing said method. The study referenced by [20] does not place emphasis on the synchronization technique, but rather delves into the topic of privacy preservation in the context of FL. The cited source exclusively presents a general outline of the concept of federated learning within the context of healthcare. The cited source solely presents a broad introduction to the concept of FL within the context of edge computing. The authors of a separate survey publication [21] have directed their attention towards an alternative viewpoint of FL, with the aim of offering a resolution for Non-Independent and Identically Distributed (Non-IID) data. However, the network topology, aggregation, and synchronization method were not discussed. The cited source solely presents information regarding the network topology, without delving into the synchronization and aggregation technique.

The contribution of this survey paper is listed below:

- Provides a comprehensive in-depth overview of a network topology.
- Discussed the existing Synchronous, Asynchronous, and Semi Asynchronous FL schemes
- Discussed the existing basic and personalized weighted aggregation methods
- Discussed the current challenges of FL and provided the future direction

The taxonomy of a survey paper is presented in Figure 1. In the next section, we provide an overview of existing surveys and their limitations.

II. RELATED WORK

Numerous surveys have been conducted in the scholarly literature regarding the topology, aggregation, and synchronization techniques employed in FL, as outlined in Table 1.

III. BACKGROUND OF FEDERATED LEARNING

The concept of FL is initially introduced by Google's research team in 2016/2017. The primary objective of this novel paradigm is to facilitate machine learning model training using data from mobile devices. The underlying rationale for this concept is to uphold the confidentiality of the user's private information through engagement in a cooperative learning and training. Conducting on-device training of machine learning models is preferable from a data security standpoint, as opposed to transferring data to a centralized

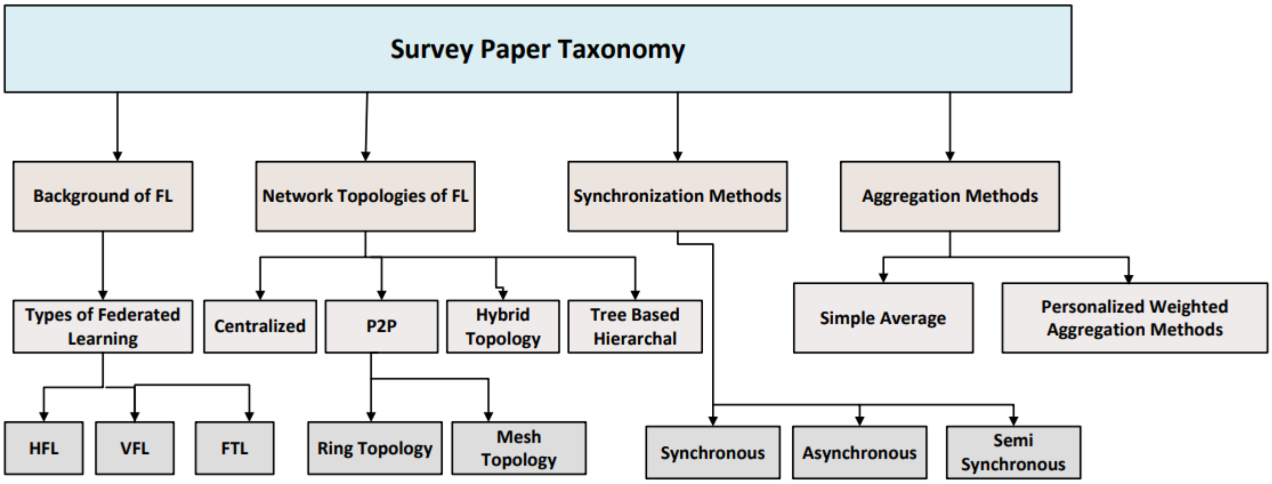

FIGURE 1. Taxonomy of a survey paper.

location. An FL System enables multiple participants to engage in collaborative model training. Within a FL context, there exist two primary stakeholders: the first a participating device who trains a model utilizing its local data, and the second a central server to which each participant transmits their local model updates. The process involves the training of a machine learning model on local data by each participant device, followed by the computation of weights or gradients, and ultimately transferring the local model parameters to the central server. The server received weights from individual clients, conducted weighted aggregation, and subsequently transmitted the aggregated weights back to each respective client. The architectural design of FL is depicted in Figure 2. The Figure depicts the existence of clients who possess data locally and are responsible for conducting model training. These clients subsequently transmit the trained model to a central server.

The central server or recipient device executes the process of model aggregation and subsequently transmits the aggregated model to a device. Various aggregation techniques exist, with Fed-Avg being a prevalent method employed for model aggregation.

The FedAVG method refers to a weight aggregation technique that involves computing the average of the weights of all participants. The process of training a global machine learning model involves multiple rounds of communication between the participant and server. The objective of this collaborative learning approach is to acquire a comprehensive model that reduces a global loss function while simultaneously enhancing the accuracy of predictions. The process of aggregating FL is illustrated in Figure 3. As depicted in Figure 3, it can be observed that three clients transmit a model to a server, which subsequently conducts the weight aggregation process.

FL is typically categorized into three distinct classifications based on particular scenarios. The subsequent segment delves into the intricacies of various forms of FL.

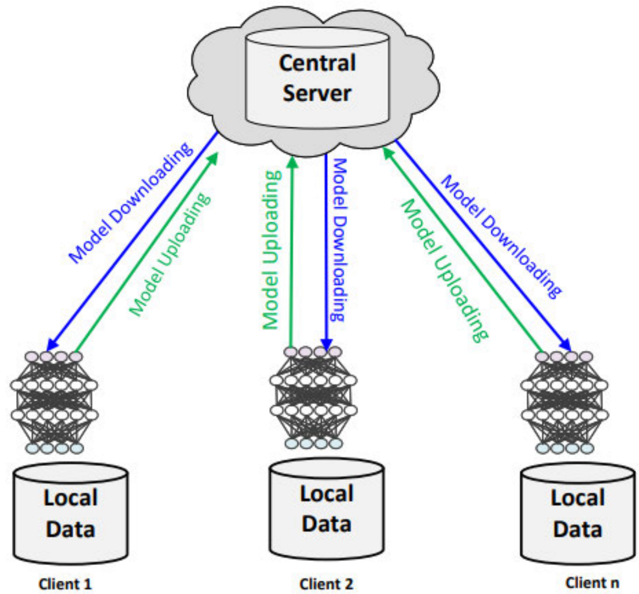

FIGURE 2. Representation of a federated learning architecture with three local clients and central server.

A. TYPES OF FEDERATED LEARNING

The FL system offers the capability to leverage existing data across various domains and systems. The presence of heterogeneous environments may result in varying data distributions across devices that are situated at different locations [31]. Prior to constructing the FL system, it is imperative to comprehend the distribution of the data. The three primary categories of FL systems are Horizontal Federated Learning (HFL), Vertical Federated Learning (VFL), and Federated Transfer Learning (FTL).

1) HORIZONTAL FL

In this type of system, the data from multiple devices may have different instances but have the same number of feature sets. In HFL, each participant device may have data from a

TABLE 1. Detail of existing Surveys.

Existing Studies	Limitations
[19] Perform a detailed analysis of the asynchronous method and gradient compression method	only focus on the asynchronous method
[20] Discuss the advantages of FL regarding privacy-preserving.	not focus on the synchronization method, and discuss the privacy-preserving in FL
[22] Discuss the research questions regarding the security and privacy of FL	Not focus on the synchronization method and discuss few aggregations method
[23] Provides the detail of challenges and future direction of FL for health	Not provide a detailed analysis of the aggregation and synchronization method
[24] Discuss issues related to communication, and discuss a different solution to enhance the communication efficiency of FL.	Focus on the communication Broadly and discuss the sparsification, quantization, client selection and compression method for efficient communication
[25] Provides detail of quantization and sparsification methods for communication efficiency and discuss a straggler problem in FL	Do not focus on the synchronization method.
[26] Discuss the client selection, quantization and sparsification and Load balancing for efficient communication.	Not provide a detailed analysis of existing synchronization method.
[27] Discuss the privacy method for FL	Only focus on privacy
[28] Provides the detail of Scale of the federation, privacy mechanism, Machine Learning Model, and motivation of the federation	Not discuss the existing communication methods.
[29] Discuss the applications and challenges of FL for smart cities and provides a possible solution.	Not discuss the existing communication methods.
[30] overview the different network topologies of FL	Not discussed focus on aggregation and synchronization method

homogenous environment [32]. Each worker trains a machine learning model and subsequently transfers it to a server. The concept of HFL entails the possibility of varying numbers of instances across devices, whereby the device with a higher number of instances may require a longer computational time, leading to the server waiting for a model. HFL is particularly well-suited for collaborative learning in cases where the distribution of data is similar to eq1. In an initial investigation of FL, a proposed approach involved utilizing mobile devices as clients to facilitate collaborative training among them [5]. A previous scholarly investigation [33] introduced a HFL framework aimed at improving the performance of the gradient boosting method.

The two primary architectures of HFL are peer2peer and client-server, as documented in reference [34]. The architecture of client-server relies on a centralized computing model, where each client transmits a model to a central server. The aggregation of individual client weights is executed by the central server, which subsequently disseminates

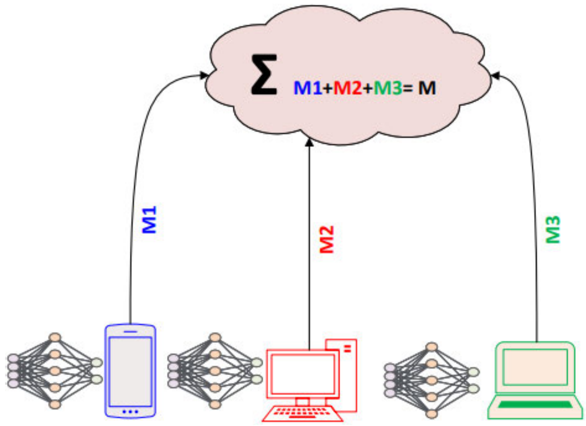

FIGURE 3. Representation of aggregation process of federated learning with three local clients and central server.

the aggregated weight to all clients. The architecture that follows the client-server model presents a challenge in terms of a single point of failure. In the event of a central server failure, all clients relying on it will experience a disruption in service [35]. The peer-to-peer architecture is characterized by a lack of a central server and a reliance on decentralized computing. This architectural approach allows for dynamic client selection for aggregation and enables the establishment of a neighborhood topology to facilitate communication and aggregation [36].

2) VERTICAL FL

VFL involves devices that share the same data distribution space while exhibiting distinct feature spaces [15]. The training process in VFL incorporates a mechanism for feature-based learning, which is necessitated by the presence of diverse feature spaces. The identity space remains constant in VFL, while the feature space varies. The VFL environment enables multiple entities to collaboratively train a unified global model utilizing distinct feature spaces while maintaining data privacy. Prior to commencing collaborative training, data alignment is conducted. This data alignment method finds the common sample IDs. In VFL, each device only exchanges the intermediate results and keeps the data and model private. In VFL, each device has its own model when collaborated training is completed. Vertical Federated Learning consists of two types of architecture such as the third-party coordinator and without third-party coordinator architecture [37].

a: ARCHITECTURE WITH THIRD-PARTY COORDINATOR

In the context of collaborative training for a global model, it is common to have two distinct nodes, namely N_1 and N_2, which undertake the task of training a model on their respective local data-sets. The given scenarios entail that N_1 possesses labeled data which is imperative for the purpose of training. It is assumed that Nodes N_1 and N_2 are engaged in a training process characterized by honesty, albeit with a sense of curiosity towards each other. A third-party entity with the designation of N_3 has been registered as a third-party coordinator with the purpose of ensuring the confidentiality of data. The underlying premise is that N_3 possesses a level of integrity akin to that of a trustworthy governing body.

- The first step is ID alignment. VFL system uses encryption-based ID alignment mechanisms to ensure the common IDS without compromising the data privacy of N_1 and N_2. The model is trained on these common instances.
- The third-party node generates the encrypted key pair and shares this key with nodes N_1 and N_2.
- The node N1 and N2 perform encryption of intermediate results.
- N_1 and N_2 compute the encrypted gradients and add a mask. Both nodes send these encrypted results to N_3.
- N_3 performs the decryption of the gradient and sends it to nodes N_1 and N_2. Both nodes unmask the gradient and update the model.

b: ARCHITECTURE WITHOUT THIRD-PARTY COORDINATOR

This type of architecture has no central coordinator between the clients. The nodes N_1 and N_2 are honest, but they are curious about data privacy. Protecting the data in VFL consists of these main steps.

- The first step is ID alignment. VFL system uses encryption-based ID alignment mechanisms to ensure the common IDS without compromising the data privacy of N_1 and N_2.
- N_1 generates an encrypted key and sends the key to N_2.
- Both node N_1 and N_2 initialize their model
- N_1 and N_2 node compute partial linear predictors and N_2 send its result to N_1.
- Node N_1 computes the residual (local gradients) and sends the encrypted to N_2.
- Node N_2 computes the encrypted gradient and shares the masked gradient with N_1.
- Node N_1 performs the decryption of the masked gradient and shares it with N_2.

3) FEDERATED TRANSFER LEARNING

Multiple devices are located at different locations, but they have a small intersection with each other [38]. Consequently, a model trained on comparable types of data can be utilized in other environments by transferring the pre-training knowledge. Clients have distinct IDs and feature spaces in the real world. The devices are dispersed across numerous geographic locations. However, interaction between the devices is minimal. The global model is trained on comparable data from other environments, and its knowledge is then transferred to another environment. The FTL methodology encompasses two distinct approaches, namely feature-based FTL and parameter-based FTL, as outlined in reference [39]. The objective of Feature-based Faster Than Light (FTL) is to acquire a proficient feature representation and reduce the disparity between domains for the intended domain. The approach of parameter-based FTL prioritizes the hyper-parameters prior to the distribution and utilization of shared parameters between the source and target domains. The architecture of a FTL is depicted in Figure 4.

Some other emerging paradigms share common standard features with FL. The particulars of the various emerging paradigms are presented in Table 2. Split Learning is a distributed machine learning methodology that involves central training of the model on a server while maintaining the privacy of client device data to ensure its confidentiality. Distributed Machine Learning is a paradigm that employs either data parallelism or model parallelism to train a machine learning model across multiple devices. The Model Parallelism technique involves partitioning a given model, such as a Neural Network model, into K segments and distributing them across k devices. Data Parallelism is a technique that involves distributing the data across multiple devices. Mobile Edge Computing (MEC) is an architectural framework for distributed computing that aims to bring computing resources in close proximity to end-user devices. The computing resources encompass storage, networking, and processing capabilities. Privacy-preserving machine learning refers to a methodology aimed at mitigating the risk of data leakage in machine learning. This approach involves the collective training of a model by several participants through the utilization of encryption techniques. The utilization of encryption techniques facilitates the safeguarding of data during communication with a centralized cloud by the device.

IV. NETWORK TOPOLOGIES

The network topology in a FL system determines the interconnectivity of participant devices within the FL network. The selection of network topology is a crucial aspect to be taken into account while deploying FL, as it can considerably influence the efficiency and efficacy of the training procedure. The selection of network topology for federated learning is contingent upon the distinct demands of the application, encompassing the magnitude and intricacy of the network, the quantity and character of the data being analyzed, and the communication and computation resources that are at disposal. The network topologies that are utilized to establish the FL networks are depicted in Figure 5, which represents the current cutting-edge approaches in this field. Various network topologies are applicable for FL, such as:

- Centralized: In this topology, all participating devices or clients connect directly to a central server, which coordinates the training process and aggregates the client's model updates. This simple and straightforward approach can be vulnerable to single points of failure and may not scale well with many clients.
- Decentralized: In this topology, the participating devices or clients are organized into a peer-to-peer network, with each device communicating directly with its neighbours. This can be more resilient to failures and more scalable

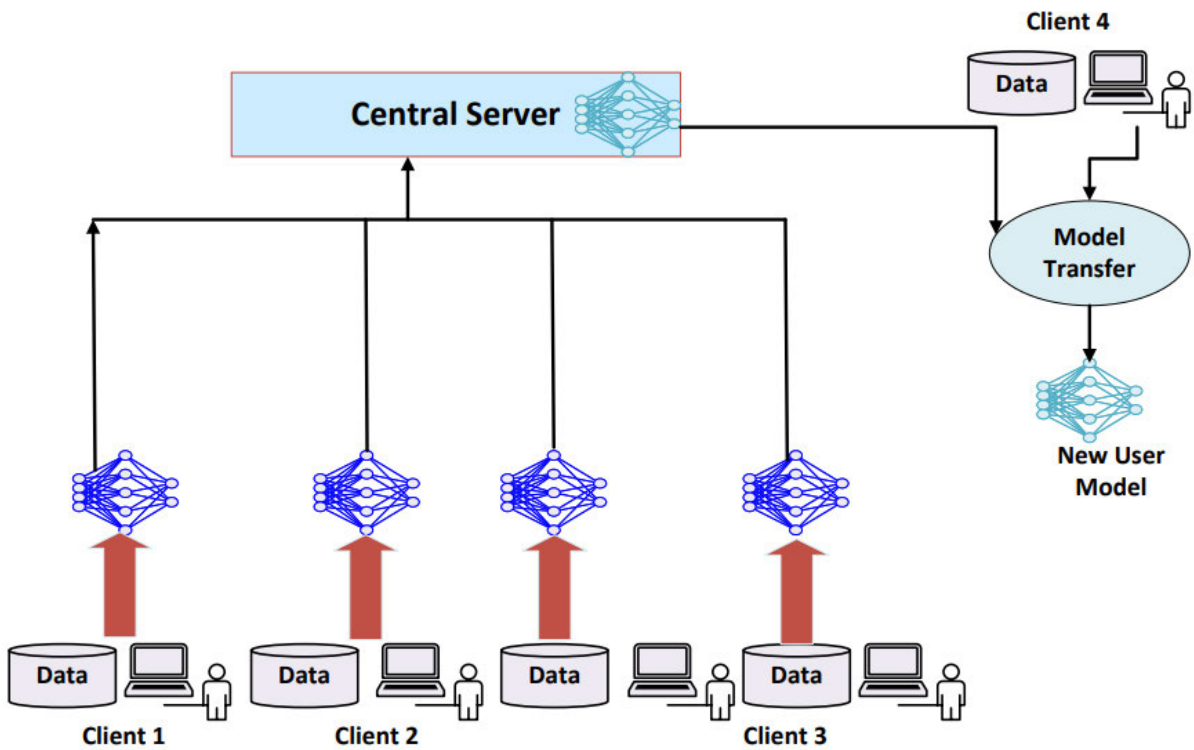

FIGURE 4. Representation of federated transfer learning architecture with four local clients and central server.

than a centralized topology, but it can also be more complex to implement and may require more communication between devices.

- Hybrid: A hybrid topology combines elements of both centralized and decentralized approaches. For example, a central server may be used to coordinate the training process while the participating devices communicate with each other directly to exchange model updates. This can offer both approaches benefit while mitigating some drawbacks.
- Tree-based (Hierarchical Structure): In this topology, the devices or nodes in the network are organized in a tree structure, with some nodes serving as parent nodes and others as child nodes. The parent nodes communicate with their child nodes, and the child nodes communicate with their parent nodes, forming a hierarchy of communication. This topology can be efficient for large networks and can reduce communication overhead compared to a peer-to-peer topology.

A. CENTRALIZED (START TOPOLOGY)

It is a centralized fL paradigm [43] that collaboratively trains a learning model on each client with the presence of a central entity called a server.It is commonly assumed that all clients are truthful and that the central server is both secure and inquisitive about the model of each client. FL is a privacy-preserving approach that involves the exchange of model parameters or gradients between clients and a central server to facilitate collaborative training. The present collaborative training program comprises of the subsequent stages.

- The central server initializes the model's parameters and shares them with all clients.
- The clients that receive an initial parameter from the server start the training on their local data,
- After completing the local training, each client sends their local model to a central server.
- The server receives the model from clients and performs the aggregation of each client's parameter.
- The server sends an updated global model to all clients, and the clients start training on a received global model.

Within the context of FL, it is customary for clients and servers to exchange two distinct categories of parameters, namely gradients, and weights. The client situated in the immediate vicinity undertakes the training process locally and calculates the weights locally. The weights of a model, which have been computed by the client, are transmitted to a server for the purpose of aggregation. The server is responsible for performing the aggregation of the weights and subsequently transmitting them back to the client. One of the benefits of utilizing this approach is its reduced communication expenses. Nonetheless, the absence of convergence assurance is a characteristic of this particular method of weight sharing at the local level. A distinct parameter-sharing paradigm involves a scenario where a model is trained by a local client, which subsequently transmits the gradient to a server for gradient aggregation.

TABLE 2. Existing collaborative learning paradigm comparison with federated learning.

Related concepts	Primary features	Comparison with FL common features	Comparison with FL different features
Split Learning [40]	It has a model-splitting design. It has a client-server architecture. It works in collaborative learning, and it is communication efficient.	It can work with large datasets. No data sharing	FL performs the entire model training and FL peer2peer architecture may not have a central server. FL is privacy-preserving
Distributed Machine Learning [41]	The main primary feature of DML is the data and model parallel scheme. Homogeneous data distribution It has multiple local clients, and there is an aggregator for aggregation	Both can work on large-scale data-sets. Both work on the concept of distributed learning (computing)	FL peer2peer architecture may not have a central server. FL environment can have both homogeneous and heterogeneous data. In FL, each client has separate data, and there is no exchange of data-sets is performed. FL is privacy-preserving
Mobile Edge Computing [25]	Mobile edge computing has a three-layer architecture that consists of cloud servers, edge servers and end users. It has slow latency	It is privacy preservation. Large scale data distributed computing	In FL, each client has separate data; no dataset exchange is performed. FL peer2peer architecture may not have a central server.
Privacy-Preserving Machine Learning [42]	Centralized Learning (Computing) It works with a combination of machine learning methods and privacy preservation.	It has a privacy-preserving feature that is common to FL	FL based on distributed computing paradigm. It has many participants that have separate private data. FL can work on large-scale datasets.

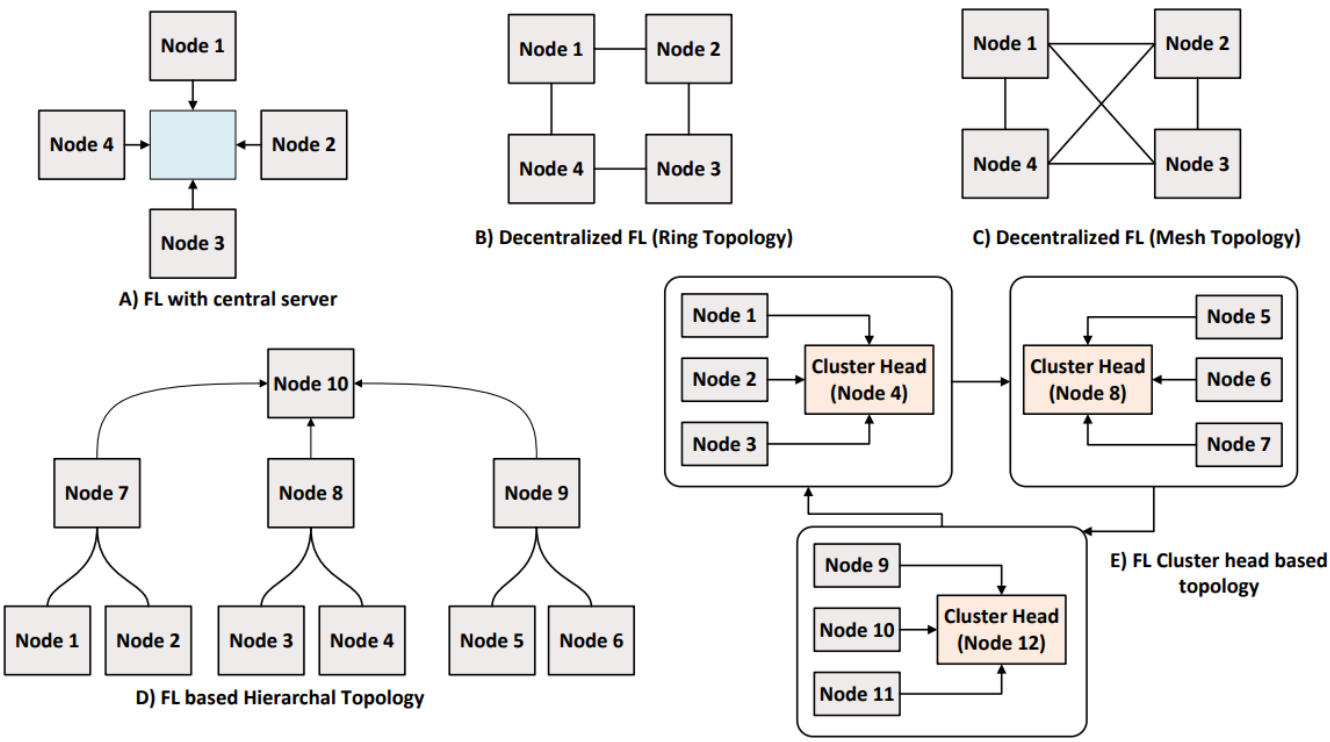

FIGURE 5. Representation of centralized, decentralized, hierarchical and hybrid network topology for federated learning.

The server then returns the global gradient model to the client. Gradient sharing offers the benefit of ensuring convergence. However, it incurs additional expenses in terms of communication.

B. PEER-TO-PEER TOPOLOGY (DECENTRALIZED)

The Peer to Peer (P2P) model is a type of decentralized FL paradigm, as referenced in source [44]. Unlike the traditional client-server architecture, P2P does not rely on a central

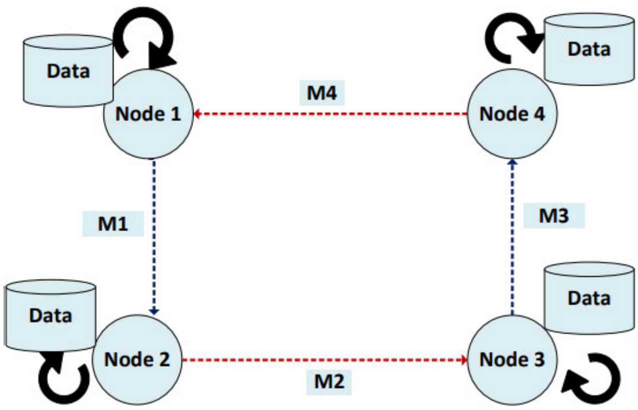

FIGURE 6. Representation of sequential model transfer strategy of ring topology.

server. In P2P FL each client trains a model on its data and shares the model according to network topology. The network topology may take the form of a ring or a mesh. In a ring topology, clients engage in a process of training a model on their respective local data and subsequently transmitting it to the next node in the network. In a mesh network topology, each node is interconnected with other nodes, allowing for the exchange of data and information between clients. The security of this architecture surpasses that of the client-server model due to the absence of a central server and a single point of failure.

1) RING TOPOLOGY

The Ring Topology is characterized by the interconnection of participant devices with two adjacent devices, namely the previous and next device. The apparatus conducts on-site training and transmits a model to a subsequent interconnected node. Two strategies are typically considered for transferring models to other nodes in order to share them. The two techniques under consideration are sequential model transfer and probability-based node selection.

a: SEQUENTIAL MODEL TRANSFER STRATEGY

The Sequential Model Transfer protocol is a method of organizing a client in a sequential manner to establish connections between each node and its subsequent node. The initial client denoted as N_1 in a given sequence executes the process of local training and subsequently transmits a model to the succeeding client, N_2. The N_2 client executes parameter aggregation utilizing its local parameter and subsequently receives the parameter that is transmitted to the subsequent client, N_3, in a circular fashion. This process continues until the termination condition is met. The process of transferring a sequential model is depicted in Figure 6.

b: DYNAMIC PROBABILITY BASED NODE SELECTION

The Dynamic Node Selection Protocol is designed to dynamically select a node based on probabilistic considerations. N_i, a client operating within a specific locality, trains a model and subsequently employs a node selection process based on probability. As an illustration, a given node denoted as N_i performs a computation of a parameter that is local in nature and subsequently chooses another node, N_j, based on a probability criterion. The parameter aggregation is performed by the client N_j, who subsequently selects a node N_k through a probabilistic approach. The collaborative training persists until the specified termination condition is met. The criterion for termination may encompass a global communication round, global threshold loss, or accuracy. The process of selecting a node based on probability is depicted in Figure 7. This figure shows that in a i^{th} iteration all interconnected devices send a model to a $Node_5$. In a $i+1$ iteration $Node_4$ is selected to receive a model. All Nodes send a model to $Node_4$.

2) MESH TOPOLOGY

The Mesh Topology refers to a network architecture in which edge devices are interconnected with each other without the presence of a central server. Within a Mesh network comprised of edge devices, each individual device conducts localized training and disseminates a model to all other edge devices present within the network. In a network, every edge device is bestowed with a model from all other devices present within the network, which is then combined with its own local model. The Mesh Topology exhibits various advantages in comparison to alternative network topologies.

- Mesh Topology is Robust and Fault Tolerance. Because in a Mesh, each device receives a model from all other devices in a network, and if some device goes down, it not affect the overall communication of a network.
- Another benefit of a Mesh Topology is that it reduces the dependency of edge devices on a central server, making the FL system more efficient and Scalable.

However, Mesh topology has more communication overhead and required more communication between the devices in collaborative training.

3) HYBRID NETWORK TOPOLOGY

In the literature, researchers present many combinations of network topologies to achieve better communication efficiency and fast convergence. The hybrid network topologies combine at least two topologies to form an efficient FL network.

a: COMBINATION OF RING AND START TOPOLOGY

In this type of topology, the edge devices within a cluster are connected to central cluster heads. Each device trains a model and sends it to a cluster head. Multiple cluster heads are connected to each other in a ring. Each cluster head performed the weight aggregation and sent it to the next cluster heads. The other topology is also formed in a vice versa. In the Same cluster, each device is interconnected to the next and previous node. After each communication round initial node in a ring send a model to a central server.

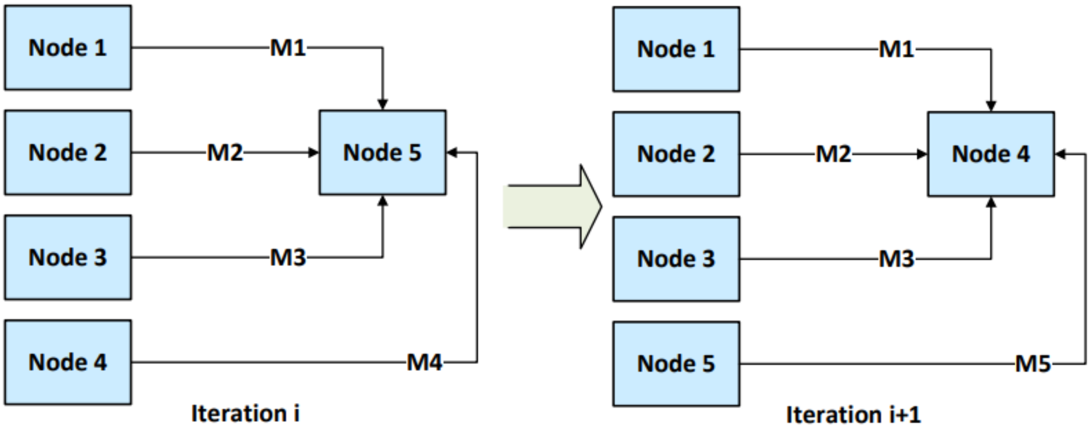

FIGURE 7. Representation of dynamic probability based node selection of ring topology.

TABLE 3. Comparison of centralized, decentralized and semi decentralized learning.

Techniques	Current Status	Problems
Central	HetViz [13], CoLearn [45] FedMTL [46] DisFL [47]	Dependency on a central entity and prone to the reliability issue scalability
Decentralized	Gossip Learning GossipSGD [48] Gossip as Alternative [49] SL for healthcare [50] Splited [51] Swarm Learning [52] SL for healthcare [53] Swarm Reinforcement learning [54]	Purely decentralized models prone to the scalability issue Synchronization problem Vulnerable to noise nodes
Semi-decentralized	ConFL [55] TT-HF [56]	Purely decentralized models prone to the scalability issue Synchronization problem Vulnerable to noise nodes

4) HIERARCHICAL TOPOLOGY

Hierarchical Topology provides a more organized structure in FL. In this network there are multiple layers of edge node between the server and bottom layer edge devices. The edge devices that are at a bottom layers they performed the local training and send a model to bottom layer node. The bottom layer receive a model and performed the aggregation of a weights and send to a central server. Some Benefits of a hierarchical topology are listed below:

- Enhanced Privacy: Hierarchical Topology enhanced the privacy of the FL network by aggregating the local updates of a model at a higher level before sending it to a central server. It reduces the data transmission and helps to protect the sensitive data.
- Scalability: Hierarchical Topology provides the benefit of Scalability. It is a more structured approach to organize the edge devices in an FL network as the number of devices increases.
- Enhanced Communication Efficiency: It reduces the communication overhead by enabling a local aggregation of a model at a higher level before sending a model to a central server.

C. COMPARISON OF CENTRALIZED, DECENTRALIZED AND SEMI-CENTRALIZED LEARNING SYSTEM

Centralized models like vanilla FL and variants mainly rely on a central entity and are vulnerable to faults, stragglers, attacks and prone to scalability issues [13], [45], [46], [47]. The swarm learning algorithm dynamically selects a leader to be aggregated in each communication round, the all-reduce-based strategies are vulnerable to attacks and trust issues. [48], [49]. Table 3 shows the comparative analysis of centralized, decentralized, and semi-decentralized.

V. CHALLENGES RELATED TO COMMUNICATION ISSUES AND OVERVIEW OF EXISTING SYNCHRONIZATION METHODS

This section discussed the existing challenges of FL related to communication, and it also discussed the detail of existing synchronization methods.

A. CHALLENGES RELATED TO COMMUNICATION

In FL, multiple participants participate in collaborative training. Each participant may have heterogeneity in computing resources, network bandwidth, and data distribution. In the next sections, we discussed the detail of these challenges.

1) HETEROGENEITY IN COMPUTING AND NETWORK BANDWIDTH RESOURCES

Communication is performed multiple times between the local client devices and the server. A real-world environment needs fast communication between the server and clients [38]. Unfortunately, communication is affected due to the heterogeneity of device computation capacity and network bandwidth [57]. The devices that participate in collaborative learning require high computational power and internet bandwidth [58]. Smart Edge devices have fewer computational resources and storage resources [59]. Training of the Machine Learning model depends on an edge device computing capability. These devices have fewer CPUs [60]. The local training time of each device varies due to the resource's heterogeneity. Besides the computational issues, the client device required a high network bandwidth to upload and download the model from a server [61]. The device with less network bandwidth and computational resources than other devices causes a straggler problem at a server [62]. Due to a heterogeneous environment, the ideal network bandwidth and resources are not possible to perform fast communication [63].

2) HETEROGENEITY IN DATA SIZE

The FL environment has multiple nodes that perform their local training, and the distribution of data varies between each client [64]. Each device has only its local data, and each client's data collection environment may differ. Due to the variation in data distribution, the local training time of each device is different from others [31]. The device that has more data samples takes more time to train. The server waits for a client that takes longer time in local training and uploading the model [51]. In the next section, we discussed the existing synchronizations methods.

B. SYNCHRONIZATION METHODS

Distributing training is a collaborating learning mechanism where multiple devices or nodes perform the training, and that is controlled by a central entity [65]. To synchronize the worker devices with the master central node, three main types of methods are used to ensure consistency in collaborating learning, including Bulk Synchronous Parallel (BSP), Asynchronous Parallel (ASP) and Stale Synchronous Parallel (SSP) [66]. BSP [67] is a computation framework for distributing learning that divides the computation into a sequence of super steps. In FL, each device computes local gradients or weights and sends them to a central server. The server waits for all devices to be before the aggregation of weights. However, the convergence speed of BSP is fast, but it has a straggler problem that slows down the whole communication process. Asynchronous Parallel [68] performs the communication between the client devices and server asynchronously. The approach of ASP is opposite to BSP, and it speeds up the communication process. In an ASP, each client device computes the local gradients and weights and transmits them to a server, and the server performs the weight aggregation and sends a global model. However, the lack of coordination with the other client device can prevent the old version or state of the model from reducing the performance of the global model. Stale Synchronous Parallel (SSP) [69] combines ASP and BSP. This method switches based on the policy Between ASP and BSP during the collaborating training. The staleness parameter restricts the iterations between the slow and fast worker and ensures that it does not exceed a staleness threshold. Figure 8 shows the block diagram of the synchronous, asynchronous, and semi-asynchronous methods. This figure shows that when devices are communicated with the server using synchronous synchronization central server has to wait for all devices to perform the aggregation. While on the other hand, in the case of asynchronous, whenever a model training is computed it sends a model to a central server, and the central server does not wait for the other devices to perform the aggregation. In a semi-asynchronous scheme, the central server caches the received model and aggregates a received model after a certain period.

In an existing study, the researcher proposed various synchronization methods to synchronize the clients and server in an FL environment [69]. There are commonly three types of FL synchronization methods used in literature. These are synchronous, asynchronous, and semi asynchronous. This survey explored these methods and discussed some drawbacks of these existing methods.

1) SYNCHRONOUS METHOD

Reference [70] design a novel partial synchronization parallel method that reduces the traffic by transferring the gradient simultaneously at the relay node. This method breaks the constraints of traditional transmission and exploits the broadcast characteristic of relay nodes. A paper [71] proposed a hierarchical clustering method that initially clusters the available clients based on the computing resources of devices. The devices perform the local training, and the local model is transferred to a server, where the server aggregates the weights simultaneously according to a cluster category. Reference [72] proposed a probabilistic synchronous parallel that ensures that only some workers participate in knowledge aggregation. According to this method, the nodes that pass the probabilistic sampling test will be allowed to participate in aggregation. Reference [73] proposed a synchronous synchronization-based device selection method. A device with minimum expected delay and expected contribution towards a global model will be selected for aggregation. It reduces the waiting time of the server. Table 4 shows the comparative analysis of the existing synchronous synchronization method.

2) ASYNCHRONOUS METHOD

Reference [74] proposed a buffered asynchronous method that stores a client update in a buffer. A server performs the aggregation when the number of arrived models equals the size of the buffer. This buffer size is a tunable parameter that can increase or decrease according to the waiting time of the server and convergence rate. In a paper [75] asynchronous method to scale the training in a distributed environment and reduce the difference between the worker's local model. They found the client's local and global models and introduced an oscillating weight factor between the global and local models as the degree of staleness. A paper [76] proposed a synchronization method based on the asynchronous mechanism; it performs online learning on live streaming data of local clients and simultaneously passes the parameter to a server after performing the local training. In a paper, [77]

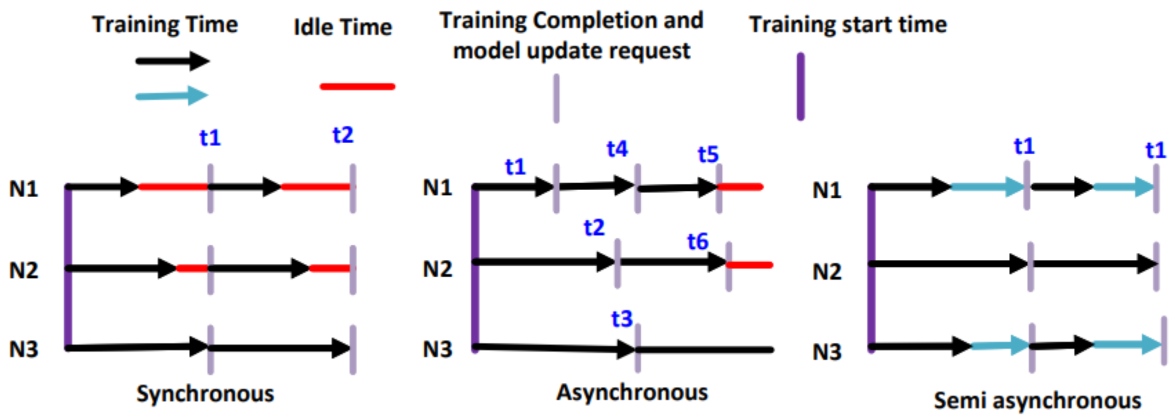

FIGURE 8. Representation of synchronous, asynchronous and semi synchronous federated learning synchronization method.

TABLE 4. Comparative analysis of synchronous synchronization method.

Ref	Method detail	objective	Model and data distribution	Application area and dataset	Convergence Analysis
[70]	Each mobile device calculates the stochastic gradient based on the local parameter and transmits the update to the adjacent relay base station. Meanwhile, the mobile base station broadcasts the global parameter to all relay base station. The additional overhead of the Relay base station. If a relay-based station goes down or has some latency, it may also affect the communication	Enhanced communication efficiency and reduced communication overhead.	CNN and Logistic Regression & IID	Computer Vision & Cifar10, Mnist	yes
[71]	Make a cluster of devices based on their computing capacity and form and ring topology for intra-cluster communication. Each cluster communicates with the central server. If there is a straggler in a ring, it affects the communication	to tackle the problems of straggler effects and outdated models	CNN & IID and Non-IID	Computer Vision & MNIST, EMNIST-Letters, CIFAR10, and CIFAR100	yes
[72]	Introduced a probabilistic synchronous parallel method that synchronize the workers based on sampling probability and ensures that o only some proportions of the workers participate in aggregation, and the node who participates in aggregation must have passed the probabilistic sampling test.	To enhance the performance and scalability of FL systems	CNN &IID and Non-IID	Computer Vision & FEMNSIT	yes
[73]	They select a device at a federated server that significantly impacts the model's convergence speed and has less training loss. To execute the FL, the devices are selected based on the expected contribution towards a global model and expected delay time.	This study aims to enhance convergence speed and reduce communication delays.	CNN & IID	Computer Vision & MNIST, CIFAR-10, electromyogram	yes

presents an FL synchronization method based on tiers for non-IID data. The nodes are inserted into tiers based on the latency. The intra-tier synchronous mechanism is used asynchronous based communication performed in cross-tiers. It reduces the effect of stragglers in an intra-tier. However, performing the communication to a cross-tier node with more latency affects the communication. Reference [78] proposed a synchronization method that performs the aggregation of nodes in a tree structure. First, aggregate the few nodes and then aggregate the updated weights of a global model with a new arrive model. Their method tackles the problem of staleness in asynchronous synchronization. Table 5 describes the comparative analysis of the existing asynchronous synchronization method.

3) SEMI ASYNCHRONOUS METHOD

A paper [79] presents a semi-asynchronous method to increase the round efficiency and convergence rate in a scenario where clients are dropping most frequently. This method categorizes the local model into three types, date, deprecated and Tolerable. Semi-Asynchronous Protocol for Fast Federated Learning (SAFA) only requires the up-to-date and deprecated clients to synchronize with the server synchronously, while the tolerable clients stay asynchronous with the server training scheme. A study [80] proposed a semi-synchronous method for fast convergence and reducing the minimum waiting or idle time. They introduced a time interval for weight aggregation, and the time interval is based on the slow worker.

Reference [81] present a semi-synchronous method to improve scheduling efficiency. The local model arrived at the server in an asynchronous mode, waiting for a certain window size to perform the aggregation. The drawback of this method is that there are many slow workers, and in that case, at a certain time may, some need nodes arrive there and perform the aggregation. Reference [82] present a method that selects the

TABLE 5. Comparative analysis of asynchronous synchronization method.

Ref	Method detail	objective	Model and data distribution	Application area and dataset	Convergence Analysis
[74]	This study builds a system that has a buffered at a Federated server, and this buffered stores the clients at a buffered when client updates arrive at a server. The size of a buffer was fixed, and aggregation was performed when k number of clients reached a buffer. The nodes with more data and knowledge may always be ignored because of latency.	This study aims to enhance the collaborating training speed, performance, and privacy. Because fully synchronous updates are affected when we have a slow worker, and fully asynchronous (individual client updates), cause an undesirable level of privacy.	LSTM, CNN /IID	NLP and computer vision & CelebA, Sent140 and CIFAR-10	yes
[75]	The server finds the difference between the local client model and the global model and introduces an oscillating weight factor between local and global models as the degree of the staleness of each worker.	To reduce the synchronization overhead and increase the convergences of a global model.	VGG-16, ResNet-50 & IID	Computer vision & CIFAR10	Not provide the theoretical convergence analysis.
[76]	This method performs the update of the model when a client reaches a server and does not wait for the other node. However, there can be many updates that make a global model inconsistent. It learns a global feature representation on the server and uses a dynamic learning step size for training local clients. This method has issue of Staleness.	Reduce the delay and latency of collaborating learning	CNN and LSTM & Non-IID	Computer Vision and behavioral context recognition in-the-wild & FitRec Dataset, Air Quality Dataset, ExtraSensory Dataset and Fashion Mnist	yes
[77]	This study makes partition of clients into multiple tiers according to a latency response. This method performs the synchronous update in an intra-tier asynchronous update into cross-tier. clients are partitioned into tiers based on their response latencies, and the tiers asynchronously update the global model. FedAT combines synchronous intra-tier training and asynchronous cross-tier training. However, to perform the communication to a cross-tier the node with more latency affects the communication.	To resolve the issue of a network communication bottleneck and biased tiering favours towards the fastest tier.	CNN & Non-IID	Computer vision and NLP & CIFAR-10, Fashion-MNIST, and Sentiment140	yes
[78]	Performed the communication between the nodes in a different tree-structured and sent the model to a server asynchronously.	To improve the efficiency in case of unbalanced computation/communication resources between clients	Ridge and linear regression & IID	Finance and Natural Language Processing & UCICreditCard, GiveMeSome-Credi, news20, yearprediction-MSD	yes

K nodes from the m nodes based on the priority of the node. The priority of the node depends on the data distribution and computing resources. It reduces the effect of stragglers. In a study [83] presents a semi-asynchronous method in which the central server aggregates the k number of local device's models according to their arrival order. The nodes that come in a certain time T server aggregate the weights and do not wait for other nodes. Reference [84] present a new method to cluster the nodes based on their similarity. An intra-cluster node communicates in a synchronous mode, and cross-cluster communication can be performed in an asynchronous mode. But if there is some node that is straggler, it will slow down the whole communication process. Table 6 describes a comparative analysis of the semi-synchronous synchronization method. In Table 7 we performed the performance comparison of the existing synchronization method.

VI. CHALLENGES RELATED TO MODEL AGGREGATION AND OVERVIEW OF EXISTING AGGREGATION METHODS

This section discusses the challenges of FL related to model aggregation and describes the detail of existing aggregation methods.

A. CHALLENGES RELATED TO MODEL AGGREGATION

In collaborative training, the data distribution of participant clients may vary. The clients with low prediction performance on local data may affect the performance of a global model. In the next section, we discussed these challenges.

1) NON-INDEPENDENT IDENTICALLY DISTRIBUTED (NON-IID)

Another problem is a non-independent identically distributed data distribution. Non-IID data refers that the data instances are Non-Independent and not Identically distributed which means that the data instances are not drawn from the same probability distribution. In Federated Learning the participant's device data distribution varies. Regarding non-IID data distribution FedAvg does not give a convergence guarantee because of each device's feature and class label heterogeneity [63]. In the case of non-iid data distribution, data is not distributed equally amongst all the clients in terms of feature space and the number of classes [85]. A client that trains a local model on a high quantity of non-iid data may skew the global model towards his direction rather than the client

TABLE 6. Comparative analysis of the semi-synchronization method.

Ref	Method detail	objective	Model and data distribution	Application area and dataset	Convergence Analysis
[79]	Semi-Asynchronous SAFA classifies the nodes into three categories and performs the communication asynchronously with tolerable nodes. If there are more nodes in a Tolerable node category in that case, there will be more updates can occur globally that may make an inconsistent the direction of global minima	Increased the convergence rate and round efficiency in a scenario when clients are dropping offline frequently	CNN, SVM /IID	Computer vision and intrusion detection & Boston, MNIST, KDD Cup '99	yes
[80]	In this policy, every learner continues training up to a specific synchronization time point, and the time interval for an aggregation depends on a slow worker.	increased a convergence speed with a minimum idle time of the local model and energy-efficient	CNN & IID	Computer Vision & CIFAR10, CIFAR100 and Extended MNIST	yes
[81]	Local models arrive at the server and wait for a certain time window. Aggregation is performed amongst the clients that arrive in the time window.	scheduling efficiency and the stability of accuracy	MLP & IID	Computer vision & MNIST	Not provide the theoretical convergence analysis
[82]	The server will select k nodes for each iteration from m nodes based on the priority, and the priority criterion is set to be data distribution and computing resources.	Reduce the impact of stragglers and build an effective client selection mechanism to accelerate the global model.	CNN & IID data	Computer Vision & MNIST, FMNIST, CIFAR10	Not provide the theoretical convergence analysis
[83]	The server aggregates a certain number of local models by their arrival order in each round. They set the m and T parameters that control the number of nodes and the staleness of the client. If a certain number of m nodes arrive within a threshold of the T server, perform the aggregation, and send these workers.	Mitigate the impact of straggler (synchronous) and reduce the massive communication of clients with the server (asynchronous)	CNN & IID data and non-IID	Computer Vision & MNIST, CIFAR10	yes
[84]	This study used a clustering-based method to make a different cluster of nodes based on similarity. They combine asynchronous and synchronous synchronization methods to update the parameters.	combine the advantages of synchronous and asynchronous strategies to reduce the waiting time of slow workers and fast convergence.	LSTM, Logistic Regression & IID	Computer vision and NLP & MNIST, FEMNIST, Synthetic, and Sentiment140	Not provide the theoretical convergence analysis

that has a low quantity of non-iid data. After the aggregation, it affects the performance of other clients [86]. There is a possibility that the distribution of class labels is not equally represented by all clients. The client with a greater number of instances in each class than the other client with a smaller number of instances in each class requires more attention than others in terms of weights [87]. However, when we have an imbalanced class distribution of each client in this scenario assigning weights based on data distribution and performance to a model of each client is better while performing a model aggregation [88].

2) CORRUPTED NODES

FL's communication and aggregation mechanism needs to be robust to malicious nodes [89]. The server performs the aggregation of weights or gradients. However, the presence of weights of a malicious node changes the direction and decreases the performance of the global model [90]. FedAvg takes the average of all client's weights or gradients, and these resulting average weights send back to all clients. However, whenever we have extremely large and extremely less weights for some clients or all clients. The impact of one device's node is greater when performing an aggregation. A single node can change the convergence trend of the global opposite to opposite direction.

B. AGGREGATION METHODS

1) BASIC AVERAGE BASED AGGREGATION METHOD

A paper [91] presents an FL weight aggregation method that takes the average of all clients' weights in each server round. These aggregate weights are sent back to each client. It is the first method that was introduced for weight aggregation. However, the drawback of this method is that it is not robust to malicious or corrupted nodes' weight. Another study [92] presents a method for weight aggregation that is somehow robust to the malicious node. Their method used a Geometric median instead of a simple average to aggregate the weights of all clients. This method deals better with the weights of a malicious node by using the median instead of the average, which reduces the impact of large or low weights. However, the convergence speed of this method is slower than the FedAvg weight aggregation method. [93] proposed a median-based gradient, trimmed mean-based gradient method for aggregation. They take the median of the gradients for aggregation, and this method also uses a trimmed mean for aggregation. In the trimmed mean method, they first remove some gradient values and then take the mean of gradients.

TABLE 7. Performance comparative analysis of synchronization methods.

Method Name	Dataset	Performance
Partial Synchronization Parallel [70]	Mnist	97% accuracy of CNN
Probabilistic Synchronous Parallel [72]	FEMNSIT	The highest prediction accuracy is 80%
Communication-efficient FL [73]	MNIST	accuracy of this method is 90%
FedBuff [74]	CelebA	This method has 90% accuracy
SHAT [75]	CIFAR10	ResNet-50 and VGG-16 model has 93.99% and 92.28% accuracy
ASO-Fed [76]	Fashion MNIST	ASO-Fed has a 95% accuracy
FedAT [77]	Fashion-MNIST	Highest accuracy is achieved on Fashion-MNIST data 99.99% with 2 clients and 1 Non IID ratio
AFSGD-VP [78]	news20	Highest accuracy of this method is 85.5% with 8 clients
SAFA [79]	MNIST	lower global loss of their method is 0.01
SemiSync FL [80]	CIFAR-10	This method has 80% accuracy
HySync [81]	MNIST	97% accuracy with 4 clients and IID sata
Time Efficient Federated Learning [82]	MNIST	This method has almost 96% accuracy
FedSA [83]	MNIST	On a IID data it has almost 99% accuracy and 97% accuracy on a Non IID
CSAFL [84]	MNIST	10K This method has 98% accuracy

TABLE 8. Comparative analysis of basic average based aggregation method.

Ref	Method detail	objective	Model and data distribution	Application area and dataset
[91]	Fedavg method performs the average of a client's weight at the server. Not robust to malicious and corrupted node	Efficient learning of deep networks from decentralized data	CNN, LSTM & iid and non-iid	Computer vision & CIFAR10, MNIST
[92]	A robust aggregation method that used the geometric median as an average to perform the aggregation of weights at the server.	Developed a method that will be robust in case of corrupt and malicious clients.	LSTM, iid,	Computer Vision and Natural Language Processing & EMNIST and Sent140 dataset
[93]	This study used the median as the average to aggregate the value of all clients at the server. the complexity of calculating the median scales up with the number of clients and the model's parameters. Slow convergence	To build a robust aggregation method in case of malicious node	CNN & IID	Computer Vision & MNIST
[94]	Trimming some of the values of clients before averaging them	To avoid the impact of byzantine attacks or malicious client	MLP & iid	Computer vision & MNIST

A paper [94] proposed a method that is based on the Byzantine-resilient aggregation algorithm, and their aggregation rule protects the model from the attack. T means the method removes some weights and, after removing the weights, performs the aggregation. Table 8 describes the comparative analysis of the basic average based aggregation method.

2) WEIGHTED AND PERSONALIZED WEIGHTED AGGREGATION METHOD

In a paper, [95] proposed a method that performs the personalized layer wised aggregation. They assigned the weights to each layer based on its importance. This method is a better-personalized weight aggregation model to deal with a heterogeneous client in collaborating training. But this method has communication and computation overhead. A study [96] presents a precision weighted average that takes the variance of gradients while of a weighted average of local parameters of clients. The author argues that using the FedAvg in the presence of heterogeneous data distribution is not better. While their Precision-weighted average method leverages the heterogeneity of the data in the presence of feature diversity.

In a paper, [97] proposed an aggregation method that assigned the weights to each client based on the reputation

TABLE 9. Comparative analysis of weighted and personalized weighted aggregation method.

Ref	Method detail	objective	Model and data distribution	Application area and dataset
[95]	Evaluates the importance of each layer from different clients to achieve a layer-wise personalized model. pFedLA parameterized the weights during the training phase via a set of dedicated hyper networks. communication cost overhead	describes the importance of each layer from different clients, and the personalized model aggregation for clients with heterogeneous data.	CNN & Non-IID	Computer Vision & EMNIST, FashionMNIST, CIFAR10, CIFAR100
[96]	The Precision-weighted FL approach combines the weights from each client into a globally shared model where the aggregation is achieved by averaging the weights by the inverse of their estimated variance. To estimate the inverse of the maximum likelihood variance, they use the raw second-moment estimate (uncentered variance). With a batch size less batch size, such as less than 10, this method is sensitive to the noise introduced by individual sources, degrading the performance of the method.	Enhanced decentralized method performance and convergence speed in resource-constrained environments like mobile and IoT devices.	Convolutional Neural Networks & IID and Non-IID	Computer Vision & MNIST, Fashion-MNIST, CIFAR10
[97]	A user's reputation score is computed according to the performance metrics of their trained local models during each training round, assigned the weights to each client-based reputation score. The reputation score is computed using local models' performance, temporary global model performance, and last communication round performance. When computing a reputation score, they subtract the accuracy of local models from the average of all models. If there are some nodes that have a bad performance, it has more impact on averaging.	To develop a method that is robust to heterogeneous devices' impact in aggregation.	CNN, MLP & IID and Non-IID	Computer Vision & Fashion-MNIST, MNIST &
[98]	Use a gaussian distribution to find the impact of each client and assign a weight to each based on the gaussian distribution function.	Accelerate the global model's convergence speed and build a reliable in case of network instability and offline attacks.	CNN and LSTM & IID	Computer Vision & FEMNIST
[99]	Assigning a weight to each client based on the contribution of participating nodes is first measured by the angle between the local gradient vector and the global gradient vector, and then, weight is quantified by a designed non-linear mapping function subsequently.	to accelerate model convergence under the presence of nodes with non-IID	CNN & iid and non-iid	Computer Vision & FEMNIST and MNIST
[100]	Assign a weight to each client based on the training data class distribution and validation data class distribution. to improve communication efficiency in FL and fast convergence.	to improve the communication efficiency in FL	CNN & non-iid	Computer Vision & Fashion-MNIST, CIFAR-10 and MNIST
[101]	Each client computes the inference loss on the global model and sends it to a server with local updates. the server aggregates them by processing contribution-aware model aggregation with received inference loss	To handle the Heterogeneous Data and fast convergence	LeNet-5, Resnet18 and CNN & non iid	Computer Vision & MNIST, FMNIST and CIFAR-10
[102]	propose a method to involve all clients for aggregation in each round and assign higher weights to latest models and lower ones to previously trained model	To make use of the previously trained local models, thereby enhancing the accuracy and convergence of the central model.	CNN and LSTM & IID & NON-IID	Computer Vision & HAR, MNIST
[103]	regard weight assignment in FL as agnostic, and optimize the global model via a minimax optimization scheme	To avoid the biasedness of global models can towards different clients	LSTM	Computer Vision & FEMNIST and Adult
[104]	assign higher weights to devices with poor performance so that the accuracy distribution develops in a more uniform direction	To enhance the resulting fairness, flexibility, and efficiency	LSTM & Non-IID	NLP & (Sent140)
[105]	Dynamically adjust the aggregation weight of branches based on the accuracy	To reduce the impact of heterogeneity and autonomy of edge nodes and enhance efficiency.	CNN & iid	Computer vision & private data (Synthetic data)

score. The performance metrics of the local model are used to compute the reputation score of the client in each communication round. The node with better performance contributes more and is assigned a higher reputation score than the other node with less reputation score. In a study, [98] proposed a method to accelerate the global model convergence and make

TABLE 10. Performance comparative analysis of aggregation methods.

Method Name	Dataset	Performance
Fedavg [91]	MNIST	99% accuracy
RFA [92]	EMNIST dataset	The highest accuracy is almost 65%
Byzantine-Robust [93]	MNIST	94% accuracy
Trimmed mean aggregation rules [94]	MNIST	93% accuracy
HeurpFedLA [95]	EMNIST	94.11% accuracy
Precision-Weighted Federated Learning [96]	MNIST	99.0% accuracy
Reputation-enabled Federated Learning [97]	MNIST	93% accuracy
Efficient and Robust Aggregation [98]	FEMNIST	This method has almost 76% accuracy
FedAdp [99]	MNIST	95% accuracy
Fast-convergent federated learning [100]	MNIST	99.01% accuracy
AST_FedAvg [102]	MNIST	This method has 98.1% accuracy
SCAFFOLD [103]	EMNIST	80.1%
Q-FFL [104]	Sent140	This method has almost 70% accuracy
Efficient federated learning [105]	Synthetic (Private data)	This method has almost 99% accuracy

them reliable for offline attacks and network instability. They used a gaussian distribution to assign weight to clients based on potential contribution. In a paper, [99] proposed a method that adaptively assigned a weight to each client to accelerate the global model in the presence of non-iid data. They assume that a local client's contribution depends on that client's data distribution. The contribution of the local client is computed by computing the angle between the global and local gradient vectors. This method calculated the contribution in each round and assigned a weight to clients.

In a paper, [100] proposed a method that assigned a weight to a classification layer based on the class distribution of that client. The aggregation of weight is performed by assigning a weighted score to a classification layer based on the contribution. They assume that the existing method that aggregates the weights without considering the contribution of the client's class distribution may decrease the performance of the model after aggregation. A study [101] proposed an aggregation algorithm that contribution-aware model aggregation rather than average like a FedAvg. Each client downloads a model from the server, computes the inference loss, and then sends it back to a server with local updates. The server selects a node based on an inference loss and performs the aggregation. A paper [102] proposed a weighted aggregation method and assigned a weight to each client based on the data size of the client. They used the asynchronous synchronization method and categorized the neural network layer into deep and shallow layers. In this method, the parameters of shallow layers are updated more frequently than deep ones.

A paper [103] proposed a method that reduces the biasedness towards a specific client by optimizing a global model towards the target distribution by forming a mixture of the client distribution. In a paper, [104] proposed a method that assigned a weight-based performance to the local client. In this method, they calculated the dynamic weight of fairness and assigned a weight to each client. The weighted aggregation is performed based on the dynamic fair weight. A study [105] proposed a method that is based on a multi-branch neural network. In an aggregation process, their method merges multiple branches. This method performs the weighted average based on the weight of each branch. Table 9 shows the Comparative Analysis of Weighted and Personalized Weighted Aggregation method. In Table 10 performed the performance comparison of the existing synchronization method.

VII. FUTURE DIRECTION

- When the computing resources of the devices have extreme variation, in that case, there is a possibility that some of the Hybrid synchronization methods (a combination of asynchronous and synchronous) may perform better to mitigate the impact of the straggler node. However, there is a possibility that the node discarded in each communication round may contribute more than the other client devices. Therefore, there is a need for adaptive synchronization, in which the straggler node can contribute to the global model also it has a low impact on server waiting time.

- In client-server communication, the straggler problem occurs due to device resources and network bandwidth. Therefore, making a cluster of nodes based on their locality or a similar data distribution and computing resources may reduce the impact of a straggler node.
- In an aggregation phase, a malicious node's presence decreases a global model's performance. FedAvg is not robust to the malicious node. There is a need for some cryptographic security methods, such as client-server authentication, before starting the communication between the server and client. Also, need some encoding method that encodes the actual weights of the model before transferring to a server.
- Network Topology plays an important role in communication efficiency and convergence. There is a need for an adaptive topology of the network that adaptively connects the edge devices based on the data distribution and resource heterogeneity to reduce the communication delay and enhance the convergence.

VIII. CONCLUSION

The objective of this survey is to present a comprehensive analysis of contemporary techniques for synchronization, network topology and aggregation in FL. FL represents a new paradigm that requires further attention to optimize the performance, communication efficiency and convergence speed of the FL methodology. The present article provides a summary of the current approach to synchronization and aggregation, while also examining the constraints associated with these established techniques. The presence of heterogeneous devices in a FL setting is a pertinent concern that has a notable impact on both communication efficiency and convergence performance. Based on a thorough examination of the relevant literature, it can be inferred that semi-asynchronous techniques are more suitable for the FL setting due to the constraints posed by device resources and the heterogeneity of data. Personalized weighted aggregation methods are a suitable approach for addressing the challenges posed by heterogeneous feature spaces and class distributions. However, further enhancements are necessary to attain optimal performance levels in the context of the FL System for heterogeneous environments. The present survey aims to assist the research community in providing guidance on a synchronization and aggregation technique that is currently in use. The scope of this survey is limited to the examination of aggregation and synchronization techniques. In subsequent discussions, it is anticipated that issues pertaining to model sparsification, compression, and node selection will be addressed.

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

REFERENCES

[1] M. M. U. Rathore, A. Paul, A. Ahmad, and G. Jeon, "IoT-based big data: From smart city towards next generation super city planning," *Int. J. Semantic Web Inf. Syst.*, vol. 13, no. 1, pp. 28–47, 2017.

[2] H. P. Sajjad, K. Danniswara, A. Al-Shishtawy, and V. Vlassov, "SpanEdge: Towards unifying stream processing over central and near-the-edge data centers," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2016, pp. 168–178.

[3] M. Adams, "Big data and individual privacy in the age of the Internet of Things," *Technol. Innov. Manage. Rev.*, vol. 7, no. 4, pp. 12–24, Apr. 2017.

[4] Y. Hu, D. Niu, J. Yang, and S. Zhou, "FDML: A collaborative machine learning framework for distributed features," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 2232–2240.

[5] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016, *arXiv:1610.02527*.

[6] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016, *arXiv:1610.05492*.

[7] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecný, S. Mazzocchi, B. McMahan, and T. Van Overveldt, "Towards federated learning at scale: System design," in *Proc. Mach. Learn. Syst.*, vol. 1, 2019, pp. 374–388.

[8] Y. Dong, X. Chen, L. Shen, and D. Wang, "EaSTFLy: Efficient and secure ternary federated learning," *Comput. Secur.*, vol. 94, Jul. 2020, Art. no. 101824.

[9] J. Pang, Y. Huang, Z. Xie, Q. Han, and Z. Cai, "Realizing the heterogeneity: A self-organized federated learning framework for IoT," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3088–3098, Mar. 2021.

[10] A. Reisizadeh, I. Tziotis, H. Hassani, A. Mokhtari, and R. Pedarsani, "Straggler-resilient federated learning: Leveraging the interplay between statistical accuracy and system heterogeneity," *IEEE J. Sel. Areas Inf. Theory*, vol. 3, no. 2, pp. 197–205, Jun. 2022.

[11] R. Xu, N. Baracaldo, Y. Zhou, A. Anwar, and H. Ludwig, "HybridAlpha: An efficient approach for privacy-preserving federated learning," in *Proc. 12th ACM Workshop Artif. Intell. Secur.*, Nov. 2019, pp. 13–23.

[12] Y. Liu, X. Yuan, Z. Xiong, J. Kang, X. Wang, and D. Niyato, "Federated learning for 6G communications: Challenges, methods, and future directions," *China Commun.*, vol. 17, no. 9, pp. 105–118, Sep. 2020.

[13] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[14] T. R. Gadekallu, Q.-V. Pham, T. Huynh-The, S. Bhattacharya, P. K. R. Maddikunta, and M. Liyanage, "Federated learning for big data: A survey on opportunities, applications, and future directions," 2021, *arXiv:2110.04160*.

[15] T. Chen, X. Jin, Y. Sun, and W. Yin, "VAFL: A method of vertical asynchronous federated learning," 2020, *arXiv:2007.06081*.

[16] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Differentially private asynchronous federated learning for mobile edge computing in urban informatics," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 2134–2143, Mar. 2020.

[17] F. Liang, Q. Yang, R. Liu, J. Wang, K. Sato, and J. Guo, "Semi-synchronous federated learning protocol with dynamic aggregation in Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4677–4691, May 2022.

[18] L. Yu, X. Sun, R. Albelaihi, and C. Yi, "Latency aware semi-synchronous client selection and model aggregation for wireless federated learning," 2022, *arXiv:2210.10311*.

[19] C. Xu, Y. Qu, Y. Xiang, and L. Gao, "Asynchronous federated learning on heterogeneous devices: A survey," 2021, *arXiv:2109.04269*.

[20] X. Yin, Y. Zhu, and J. Hu, "A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions," *ACM Comput. Surveys*, vol. 54, no. 6, pp. 1–36, Jul. 2022.

[21] X. Ma, J. Zhu, Z. Lin, S. Chen, and Y. Qin, "A state-of-the-art survey on solving non-iid data in federated learning," *Future Gener. Comput. Syst.*, vol. 135, pp. 244–258, Oct. 2022.

[22] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Gener. Comput. Syst.*, vol. 115, pp. 619–640, Feb. 2021.

[23] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning for healthcare informatics," *J. Healthcare Informat. Res.*, vol. 5, no. 1, pp. 1–19, Mar. 2021.

[24] O. Shahid, S. Pouriyeh, R. M. Parizi, Q. Z. Sheng, G. Srivastava, and L. Zhao, "Communication efficiency in federated learning: Achievements and challenges," 2021, *arXiv:2107.10996*.

[25] Q. Xia, W. Ye, Z. Tao, J. Wu, and Q. Li, "A survey of federated learning for edge computing: Research problems and solutions," *High-Confidence Comput.*, vol. 1, no. 1, 2021, Art. no. 100008.
[26] Z. Jiang, W. Wang, B. Li, and Q. Yang, "Towards efficient synchronous federated training: A survey on system optimization strategies," *IEEE Trans. Big Data*, vol. 9, no. 2, pp. 437–454, Apr. 2023.
[27] A. E. Ouadrhiri and A. Abdelhadi, "Differential privacy for deep and federated learning: A survey," *IEEE Access*, vol. 10, pp. 22359–22380, 2022.
[28] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3347–3366, Apr. 2021.
[29] Z. Zheng, Y. Zhou, Y. Sun, Z. Wang, B. Liu, and K. Li, "Applications of federated learning in smart cities: Recent advances, taxonomy, and open challenges," *Connection Sci.*, vol. 34, no. 1, pp. 1–28, Dec. 2022.
[30] J. Wu, S. Drew, F. Dong, Z. Zhu, and J. Zhou, "Topology-aware federated learning in edge computing: A comprehensive survey," 2023, *arXiv:2302.02573*.
[31] A. Ghosh, J. Hong, D. Yin, and K. Ramchandran, "Robust federated learning in a heterogeneous environment," 2019, *arXiv:1906.06629*.
[32] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.
[33] Q. Li, Z. Wen, and B. He, "Practical federated gradient boosting decision trees," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, 2020, pp. 4642–4649.
[34] P. Singh, M. K. Singh, R. Singh, and N. Singh, "Federated learning: Challenges, methods, and future directions," in *Federated Learning for IoT Applications*. Cham, Switzerland: Springer, 2022, pp. 199–214.
[35] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Horizontal federated learning," in *Federated Learning*. Springer, 2019, pp. 49–67.
[36] A. Karras, C. Karras, K. C. Giotopoulos, D. Tsolis, K. Oikonomou, and S. Sioutas, "Peer to peer federated learning: Towards decentralized machine learning on edge devices," in *Proc. 7th South-East Eur. Design Autom., Comput. Eng., Comput. Netw. Social Media Conf. (SEEDA-CECNSM)*, Sep. 2022, pp. 1–9.
[37] Y. Liu, Y. Kang, T. Zou, Y. Pu, Y. He, X. Ye, Y. Ouyang, Y.-Q. Zhang, and Q. Yang, "Vertical federated learning," 2022, *arXiv:2211.12814*.
[38] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang, "A secure federated transfer learning framework," *IEEE Intell. Syst.*, vol. 35, no. 4, pp. 70–82, Jul. 2020.
[39] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "FedHealth: A federated transfer learning framework for wearable healthcare," *IEEE Intell. Syst.*, vol. 35, no. 4, pp. 83–93, Jul. 2020.
[40] C. Thapa, M. A. P. Chamikara, and S. A. Camtepe, "Advancements of federated learning towards privacy preservation: From federated learning to split learning," in *Federated Learning Systems: Towards Next-Generation AI*. Cham, Switzerland: Springer, 2021, pp. 79–109.
[41] J. Liu, J. Huang, Y. Zhou, X. Li, S. Ji, H. Xiong, and D. Dou, "From distributed machine learning to federated learning: A survey," *Knowl. Inf. Syst.*, vol. 64, no. 4, pp. 885–917, Apr. 2022.
[42] R. Xu, N. Baracaldo, and J. Joshi, "Privacy-preserving machine learning: Methods, challenges and directions," 2021, *arXiv:2108.04417*.
[43] G. H. Lee and S.-Y. Shin, "Federated learning on clinical benchmark data: Performance assessment," *J. Med. Internet Res.*, vol. 22, no. 10, Oct. 2020, Art. no. e20891.
[44] H. Elayan, M. Aloqaily, and M. Guizani, "Deep federated learning for IoT-based decentralized healthcare systems," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Jun. 2021, pp. 105–109.
[45] X. Wang, W. Chen, J. Xia, Z. Wen, R. Zhu, and T. Schreck, "HetVis: A visual analysis approach for identifying data heterogeneity in horizontal federated learning," *IEEE Trans. Vis. Comput. Graphics*, vol. 29, no. 1, pp. 310–319, Jan. 2023.
[46] N. Rieke, J. Hancox, W. Li, F. Milletarì, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein, S. Ourselin, M. Sheller, R. M. Summers, A. Trask, D. Xu, M. Baust, and M. J. Cardoso, "The future of digital health with federated learning," *npj Digit. Med.*, vol. 3, no. 1, p. 119, Sep. 2020.
[47] P. Kairouzz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, and R. G. D'Oliveira, "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, nos. 1–2, pp. 1–210, 2019.
[48] Y. Chen, K. Yuan, Y. Zhang, P. Pan, Y. Xu, and W. Yin, "Accelerating gossip SGD with periodic global averaging," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 1791–1802.
[49] I. Hegedűs, G. Danner, and M. Jelasity, "Gossip learning as a decentralized alternative to federated learning," in *Proc. Distrib. Appl. Interoperable Syst., 19th IFIP WG Int. Conf., (DAIS), Held 14th Int. Federated Conf. Distrib. Comput. Techn., (DisCoTec)*, Kongens Lyngby, Denmark: Springer, Jun. 2019, pp. 74–90.
[50] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, "Split learning for health: Distributed deep learning without sharing raw patient data," 2018, *arXiv:1812.00564*.
[51] C. Thapa, P. C. M. Arachchige, S. Camtepe, and L. Sun, "SplitFed: When federated learning meets split learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, 2022, pp. 8485–8493.
[52] S. Warnat-Herresthal, H. Schultze, K. L. Shastry, S. Manamohan, S. Mukherjee, V. Garg, R. Sarveswara, K. Händler, P. Pickkers, N. A. Aziz, and S. Ktena, "Swarm learning for decentralized and confidential clinical machine learning," *Nature*, vol. 594, no. 7862, pp. 265–270, 2021.
[53] M. Becker, "Swarm learning for decentralized healthcare," *Der Hautarzt*, vol. 73, no. 4, pp. 323–325, Apr. 2022.
[54] X. Zhu, F. Zhang, and H. Li, "Swarm deep reinforcement learning for robotic manipulation," *Proc. Comput. Sci.*, vol. 198, pp. 472–479, May 2022.
[55] Z. Wang, Q. Xu, K. Ma, X. Cao, and Q. Huang, "Confederated learning: Going beyond centralization," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 2939–2947.
[56] F. Po-Chen Lin, S. Hosseinalipour, S. Shams Azam, C. G. Brinton, and N. Michelusi, "Semi-decentralized federated learning with cooperative D2D local model aggregations," 2021, *arXiv:2103.10481*.
[57] S. Caldas, J. Konečny, H. B. McMahan, and A. Talwalkar, "Expanding the reach of federated learning by reducing client resource requirements," 2018, *arXiv:1812.07210*.
[58] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.
[59] M. Asad, A. Moustafa, T. Ito, and M. Aslam, "Evaluating the communication efficiency in federated learning algorithms," in *Proc. IEEE 24th Int. Conf. Comput. Supported Cooperat. Work Design (CSCWD)*, May 2021, pp. 552–557.
[60] M. Hartmann, U. S. Hashmi, and A. Imran, "Edge computing in smart health care systems: Review, challenges, and research directions," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 3, p. e3710, Mar. 2022.
[61] C. He, M. Annavaram, and S. Avestimehr, "Group knowledge transfer: Federated learning of large CNNs at the edge," in *Proc. NIPS*, vol. 33, 2020, pp. 14068–14080.
[62] D. Rothchild, A. Panda, E. Ullah, N. Ivkin, I. Stoica, V. Braverman, J. Gonzalez, and R. Arora, "FetchSGD: Communication-efficient federated learning with sketching," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 8253–8265.
[63] Z. Ji, L. Chen, N. Zhao, Y. Chen, G. Wei, and F. R. Yu, "Computation offloading for edge-assisted federated learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9330–9344, Sep. 2021.
[64] D. Li and J. Wang, "FedMD: Heterogenous federated learning via model distillation," 2019, *arXiv:1910.03581*.
[65] J. Qi, F. Lin, Z. Chen, C. Tang, R. Jia, and M. Li, "High-quality model aggregation for blockchain-based federated learning via reputation-motivated task participation," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 18378–18391, Oct. 2022.
[66] X. Zhao, A. An, J. Liu, and B. X. Chen, "Dynamic stale synchronous parallel distributed training for deep learning," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 1507–1517.
[67] H. Xu, J. Li, H. Xiong, and H. Lu, "FedMax: Enabling a highly-efficient federated learning framework," in *Proc. IEEE 13th Int. Conf. Cloud Comput. (CLOUD)*, Oct. 2020, pp. 426–434.
[68] J. Yuan, M. Xu, X. Ma, A. Zhou, X. Liu, and S. Wang, "Hierarchical federated learning through LAN-WAN orchestration," 2020, *arXiv:2010.11612*.
[69] S. S. Diwangkara and A. I. Kistijantoro, "Study of data imbalance and asynchronous aggregation algorithm on federated learning system," in *Proc. Int. Conf. Inf. Technol. Syst. Innov. (ICITSI)*, Oct. 2020, pp. 276–281.

[70] Z. Qu, S. Guo, H. Wang, B. Ye, Y. Wang, A. Y. Zomaya, and B. Tang, "Partial synchronization to accelerate federated learning over relay-assisted edge networks," *IEEE Trans. Mobile Comput.*, vol. 21, no. 12, pp. 4502–4516, Dec. 2022.

[71] G. Li, Y. Hu, M. Zhang, J. Liu, Q. Yin, Y. Peng, and D. Dou, "FedHiSyn: A hierarchical synchronous federated learning framework for resource and data heterogeneity," in *Proc. 51st Int. Conf. Parallel Process.*, Aug. 2022, pp. 1–11.

[72] J. Zhao, R. Han, Y. Yang, B. Catterall, C. H. Liu, L. Y. Chen, R. Mortier, J. Crowcroft, and L. Wang, "Federated learning with heterogeneity-aware probabilistic synchronous parallel on edge," *IEEE Trans. Services Comput.*, vol. 15, no. 2, pp. 614–626, Mar. 2022.

[73] M. Chen, N. Shlezinger, H. V. Poor, Y. C. Eldar, and S. Cui, "Communication-efficient federated learning," *Proc. Nat. Acad. Sci. USA*, vol. 118, no. 17, 2021, Art. no. e2024789118.

[74] J. Nguyen, K. Malik, H. Zhan, A. Yousefpour, M. Rabbat, M. Malek, and D. Huba, "Federated learning with buffered asynchronous aggregation," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2022, pp. 3581–3607.

[75] Y. Ko and S.-W. Kim, "SHAT: A novel asynchronous training algorithm that provides fast model convergence in distributed deep learning," *Appl. Sci.*, vol. 12, no. 1, p. 292, Dec. 2021.

[76] Y. Chen, Y. Ning, M. Slawski, and H. Rangwala, "Asynchronous online federated learning for edge devices with non-iid data," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2020, pp. 15–24.

[77] Z. Chai, Y. Chen, L. Zhao, Y. Cheng, and H. Rangwala, "FedAT: A communication-efficient federated learning method with asynchronous tiers under non-iid data," in *Proc. 51st Int. Conf. Parallel Process. (ICPP)*, Aug. 2022, pp. 1–11, Art. no. 8, doi: 10.1145/3545008.3545065.

[78] B. Gu, A. Xu, Z. Huo, C. Deng, and H. Huang, "Privacy-preserving asynchronous vertical federated learning algorithms for multiparty collaborative learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6103–6115, Nov. 2022.

[79] W. Wu, L. He, W. Lin, R. Mao, C. Maple, and S. Jarvis, "SAFA: A semi-asynchronous protocol for fast federated learning with low overhead," *IEEE Trans. Comput.*, vol. 70, no. 5, pp. 655–668, May 2021.

[80] D. Stripelis, P. M. Thompson, and J. L. Ambite, "Semi-synchronous federated learning for energy-efficient training and accelerated convergence in cross-silo settings," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 5, pp. 1–29, Oct. 2022.

[81] G. Shi, L. Li, J. Wang, W. Chen, K. Ye, and C. Xu, "HySync: Hybrid federated learning with effective synchronization," in *Proc. IEEE 22nd Int. Conf. High Perform. Comput. Commun., IEEE 18th Int. Conf. Smart City, IEEE 6th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Dec. 2020, pp. 628–633.

[82] J. Hao, Y. Zhao, and J. Zhang, "Time efficient federated learning with semi-asynchronous communication," in *Proc. IEEE 26th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2020, pp. 156–163.

[83] Q. Ma, Y. Xu, H. Xu, Z. Jiang, L. Huang, and H. Huang, "FedSA: A semi-asynchronous federated learning mechanism in heterogeneous edge computing," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3654–3672, Dec. 2021.

[84] Y. Zhang, M. Duan, D. Liu, L. Li, A. Ren, X. Chen, Y. Tan, and C. Wang, "CSAFL: A clustered semi-asynchronous federated learning framework," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–10.

[85] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5132–5143.

[86] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-iid data silos: An experimental study," in *Proc. IEEE 38th Int. Conf. Data Eng. (ICDE)*, May 2022, pp. 965–978.

[87] M. Yang, X. Wang, H. Zhu, H. Wang, and H. Qian, "Federated learning with class imbalance reduction," in *Proc. 29th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2021, pp. 2174–2178.

[88] Z. Chen, C. Yang, M. Zhu, Z. Peng, and Y. Yuan, "Personalized retrogress-resilient federated learning toward imbalanced medical data," *IEEE Trans. Med. Imag.*, vol. 41, no. 12, pp. 3663–3674, Dec. 2022.

[89] L. Wang, S. Xu, X. Wang, and Q. Zhu, "Addressing class imbalance in federated learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 11, 2021, pp. 10165–10173.

[90] S. Li, E. Ngai, F. Ye, and T. Voigt, "Auto-weighted robust federated learning with corrupted data sources," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 5, pp. 1–20, Oct. 2022.

[91] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[92] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Trans. Signal Process.*, vol. 70, pp. 1142–1154, 2022.

[93] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5650–5659.

[94] T. Wang, Z. Zheng, and F. Lin, "Federated learning framew ork based on trimmed mean aggregation rules," Tech. Rep., 2021.

[95] X. Ma, J. Zhang, S. Guo, and W. Xu, "Layer-wised model aggregation for personalized federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10082–10091.

[96] J. Reyes, L. Di Jorio, C. Low-Kam, and M. Kersten-Oertel, "Precision-weighted federated learning," 2021, *arXiv:2107.09627*.

[97] Y. Wang and B. Kantarci, "Reputation-enabled federated learning model aggregation in mobile platforms," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2021, pp. 1–6.

[98] Y. Lu and L. Fan, "An efficient and robust aggregation algorithm for learning federated CNN," in *Proc. 3rd Int. Conf. Signal Process. Mach. Learn.*, Oct. 2020, pp. 1–7.

[99] H. Wu and P. Wang, "Fast-convergent federated learning with adaptive weighting," *IEEE Trans. Cognit. Commun. Netw.*, vol. 7, no. 4, pp. 1078–1088, Dec. 2021.

[100] Z. Ma, M. Zhao, X. Cai, and Z. Jia, "Fast-convergent federated learning with class-weighted aggregation," *J. Syst. Archit.*, vol. 117, Aug. 2021, Art. no. 102125.

[101] H. Zeng, T. Zhou, Y. Guo, Z. Cai, and F. Liu, "FedCav: Contribution-aware model aggregation on distributed heterogeneous data in federated learning," in *Proc. 50th Int. Conf. Parallel Process.*, Aug. 2021, pp. 1–10.

[102] Y. Chen, X. Sun, and Y. Jin, "Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 10, pp. 4229–4238, Oct. 2020.

[103] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4615–4625.

[104] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair resource allocation in federated learning," 2019, *arXiv:1905.10497*.

[105] Q. Wang, Q. Li, K. Wang, H. Wang, and P. Zeng, "Efficient federated learning for fault diagnosis in industrial cloud-edge computing," *Computing*, vol. 103, no. 10, pp. 2319–2337, Oct. 2021.

QAZI WAQAS KHAN received the Bachelor of Science (B.Sc.) degree in statistics and computer science from the University of the Punjab, Lahore, Pakistan, in 2017, and the M.C.S. (two years) and M.S. degrees in computer science from COMSATS University Islamabad, Attock Campus, Punjab, Pakistan, in 2019 and 2022, respectively. He is currently pursuing the Ph.D. degree in computer engineering with Jeju National University, South Korea. His research interests include applied machine learning, federated learning, data mining, natural language processing, computer vision, and the Internet of Things. He received the Merit Fee Waiver Scholarship for the M.S.C.S. degree. He received the Fee Waiver PEEF Scholarship for the B.Sc. and M.C.S. degrees.

ANAM NAWAZ KHAN received the B.S. and M.S. degrees in computer science from COMSATS University Islamabad, Pakistan, in 2016 and 2019, respectively. In the meantime, she joined various reputable academic institutions and later decided to expand her knowledge by continuing her academic career. In 2020, she joined the Department of Computer Engineering, Jeju National University, Republic of Korea, as a Ph.D. Research Fellow. She is currently working on integrating AI-enabled IoT platforms and edge computing to provide autonomous intelligent services for smart environments. Her research interests include the development and optimization of artificial intelligence and machine learning-based applications in smart spaces.

ATIF RIZWAN received the Bachelor of Science (B.Sc.) degree from the University of the Punjab, Lahore, Punjab, Pakistan, in 2015, and the M.C.S. (two years) and M.S. degrees in computer science from COMSATS University Islamabad, Attock Campus, Punjab, Pakistan, in 2018 and 2020, respectively. He is currently a Ph.D. Researcher with the Department of Computer Engineering, Jeju National University, Republic of Korea. He has good industry experience in mobile and web application development and testing. He received a fully-funded scholarship for the entire duration of the Ph.D. studies. His research interests include applied machine learning, data and web mining, analysis, optimization of core algorithms, and the IoT-based applications.

RASHID AHMAD received the B.S. degree from the University of Malakand, Pakistan, in 2007, the M.S. degree in computer science from the National University of Computer and Emerging Sciences (NUCES), Islamabad, Pakistan, in 2009, and the Ph.D. degree in computer engineering from Jeju National University, Republic of Korea, in 2015. Since 2016, he has been with COMSATS University Islamabad, Attock Campus, Pakistan, where he is currently an Assistant Professor with the Department of Computer Science. His research work is focused on the application of prediction and optimization algorithms to build the IoT-based solutions. His research interests include machine learning, data mining, and related applications.

SALABAT KHAN received the Ph.D. degree from NUCES FAST, Islamabad. He is currently a Brain-Pool Overseas Researcher with the Big Data Research Center, Jeju National University, South Korea. He is also an Associate Professor with COMSATS University Islamabad, Attock Campus, Pakistan. His research interests include evolutionary computation, medical image processing, machine learning, federated learning, decentralized cooperative learning, big data analytics, recommender systems, and the IoT. He is a reviewer of several prestigious international journals and conferences.

DO-HYEUN KIM received the B.S. degree in electronics engineering and the M.S. and Ph.D. degrees in information telecommunication from Kyungpook National University, South Korea, in 1988, 1990, and 2000, respectively. He was with the Agency of Defense Development (ADD), from 1990 to 1995. Since 2004, he has been with Jeju National University, Republic of Korea, where he is currently a Professor with the Department of Computer Engineering. From 2008 to 2009, he was a Visiting Researcher with the Queensland University of Technology, Australia. His research interests include sensor networks, M2M/IoT, energy optimization and prediction, intelligent service, and mobile computing.