Indonesian Journal of Electrical Engineering and Computer Science
Vol. 40, No. 2, November 2025, pp. 926~940
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v40.i2.pp926-940

Federated learning in edge AI: a systematic review of applications, privacy challenges, and preservation techniques

Christina Thankam Sajan, Helanmary M. Sunny, Anju Pratap
Department of Computer and Engineering, Saintgits college of Engineering (Autonomous),
APJ Abdul Kalam Technological University, Thiruvananthapuram, India

Article Info

Article history:

Received Sep 3, 2024
Revised Jul 25, 2025
Accepted Oct 14, 2025

Keywords:

Artificial intelligence
Edge AI
Edge computing
Federated learning
Privacy-preserving

ABSTRACT

Edge artificial intelligence (Edge AI) involves the implementation of AI algorithms and models directly on local edge devices, such as sensors or internet of things (IoT) devices. This allows for immediate processing and analysis of data without the need for continuous dependence on cloud infrastructure. Concerns about privacy have grown importance in recent years for businesses looking to uphold end-user expectations and safeguard business models. Federated learning (FL) has emerged as a novel approach to enhance privacy. To improve generalization qualities, FL trains local models on local data. These models then collaborate to update a global model. Each edge device (like smartphones, IoT sensors, or autonomous vehicles) trains a local model on its own data. This local training helps in capturing data patterns specific to each device or node. Poisoning, backdoors, and generative adversarial network (GAN)-based attacks are currently the main security risk. Nevertheless, the biggest threat to FL's privacy is from inference-based assaults such as model inversion attacks, differential privacy shortcomings and FL utilizes blockchain and cryptography technologies to improve privacy on edge devices. This paper presents a thorough examination of the current literature on this subject. In more detail, we study the background of FL and its different existing applications, types, privacy threats and its techniques for privacy preservation.

Corresponding Author:

Anju Pratap
Department of Computer and Engineering, Saintgits college of Engineering (Autonomous)
APJ Abdul Kalam Technological University
Thiruvananthapuram, Kerala, India
Email: anju.pratap@saintgits.org

1. INTRODUCTION

The phrase "Edge Artificial Intelligence (Edge AI)" refers to the installation of AI applications on hardware found in the real world. Because the AI computation is carried out close to the user at the edge of the network, nearer the data's location, as opposed to centrally at a cloud computing facility or private data center, it is known as "Edge AI". It's particularly useful in scenarios where real-time processing and decision-making are critical, such as autonomous vehicles, healthcare monitoring systems, and industrial automation. Federated learning (FL) in edge AI is a paradigm that uses a variety of decentralized edge devices to train machine learning (ML) models without transferring raw data. This methodology facilitates ongoing education and adjustment to regional circumstances. Each edge device (such as a smartphone or internet of things (IoT) device) in this configuration has a local dataset and takes part in the learning process. A central server receives the learnings (model parameters) that the learning model has acquired from its

training on these local datasets. The server then compiles these modifications to enhance the global model. Until the model achieves an accuracy level that is acceptable, this process is repeated multiple times.

FL is a collaborative process that uses remote data sharing among multiple participants to train a single deep learning model and improve iteratively, much like in a team report or presentation. Each individual downloads the model, which is often a foundation model that has already been trained, from a cloud data center. They train it on their own private data, then summarize and encrypt the model's new configuration. The model updates are sent to the centralized model, which decodes, norms, and combines them before sending them back to the cloud. The collaborative training process carries on iteration after iteration until the model is completely trained. Distributed ML can be secured using FL as an approach to collaborating to execute FL algorithms on multiple devices. The condition is that there are scattered edge devices or servers where the private information is not left local. A decentralized ML technique called FL uses several devices or servers with local data samples to train models without transferring them. Figure 1 shows the basic architecture of FL.

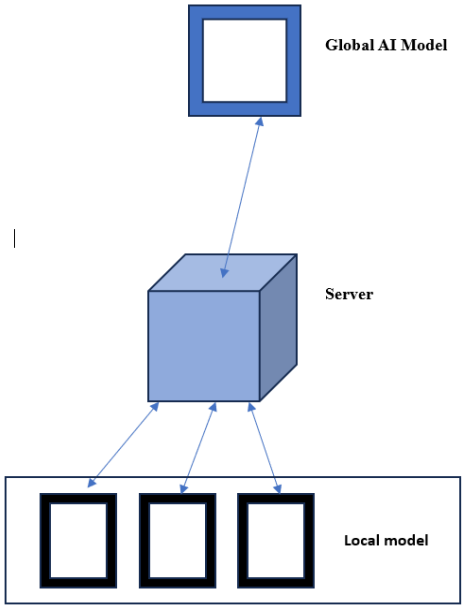

Figure 1. Basic architecture of federated learning

FL in Edge AI leverages the computational power of edge devices like smartphones devices, or edge server to train ML models locally without needing to transmit sensitive data to a central server. Edge devices perform model training using local data. This could include data collected from sensors, user interactions or other resources. Only modifications to models are uploaded to a central server; raw data cannot be sent. User privacy is upheld in this way. FL enables personalized and context-aware intelligence directly on devices without relying heavily on cloud services.

Cloud computing emerged as a unique computer architecture for the Internet based on highly resourced data centers as Information technology advanced after 2000. Development and interest in cloud computing has grown to the point where, by 2020, more than 90 % of all data center traffic will originate from sources in the cloud [1]. The immense potential of edge AI has finally been realized to the most recent strides in AI efficiency, the growth of edge computing, and the explosion of IoT devices. When AI computations are performed in proximity to consumers on a network edge, they are referred to as edge-based AI. This is in contrast to centralized data storage, such as cloud service providers or privately held data warehouses [2]. The successful operation of tasks is enhanced through the 6G services provided for edge computing and autonomous vehicular driving applications. The significant amount of data generated by these applications can be advantageous for the AI and ML industry. By preserving the ability to learn from decentralized data sets, FL, also known as FL, is an essential element in an integrated solution to privacy and technological issues. Training is limited to user devices, and the server receives the locally computed parameter, which aggregates the updated weights to optimize a global model [3]. The emergence of novel technologies like big data, edge computing, fog computing, artificial intelligence of things (AIoT), and fog computing has caused problems for smart city applications, including the disclosure of sensitive and private

data. FL can deal with plenty of smart city concerns, including enormous amounts of data and safeguarding privacy, empowering decision-makers to act swiftly. FL is beneficial in training shared statistics through decentralized devices or servers [4]. The widespread deployment of AI in healthcare poses challenges due to the scattered nature of health data. Privacy concerns can be effectively handled with the use of privacy-preserving algorithms in FL, which was developed to address data fragmentation. To enhance security and computational effectiveness, FL can be paired with other technologies such as edge computing and blockchain [5]. The single point of failure that is the bottleneck of both traditional FL and HFL systems is their reliance on a centralized server to manage the learning process [6].

Edge computing is one way of AI enhancing cybersecurity. Edge computing analyzes data at the network's edges, including individual devices, routers, and firewalls, rather than forwarding it to a central place. This has a number of security benefits, including threat detection and prevention, anomaly detection, enhanced data privacy, adaptive security policies, and fraud detection. Gaining user trust will require addressing difficulties related to performance, data processing, and human monitoring. As edge and IoT adoption grows, solid localized security will become more vital. In cyber security, FL allows businesses to interact and share insights from their data without disclosing the data itself, reducing the dangers connected with data breaches and privacy violations. This decentralized strategy also aids in the creation of more resilient and accurate models by integrating several data sources while maintaining individual data privacy [7].

The remainder of this paper is organized as follows: In section 2 discusses the overview, importance, different types of models, applications, privacy challenges and preservation techniques of FL in Edge AI. In section 3 presents the results and discussions. Finally, section 4 concludes the paper with future research directions.

2. OVERVIEW OF FEDERATED LEARNING IN EDGE AI

FL in Edge AI combines the principles of FL with edge computing, facilitating the immediate training of ML models on edge gadgets while preserving data privacy and reducing communication overhead. FL is a cooperatively decentralized solution that protects privacy while addressing issues with data sensitivity and silos.

In FL, instead of aggregating data in a centralized server, the training process occurs locally on peripheral devices such as edge computing servers, mobile devices and IoT devices. By retaining raw data on edge devices, it guarantees data privacy. The only updates to the model that are sent to the centralized server for aggregation are the weights or gradients. This approach minimizes the risk of data breaches or privacy violations. By performing training locally on edge devices, FL reduces the need for data transmission to a central server. This is particularly beneficial in edge computing environments with limited bandwidth or intermittent connectivity. Edge devices collaboratively contribute to model training by performing local updates based on their respective datasets. These updates are then aggregated to improve the global model, leveraging insights from diverse edge devices. FL enables models to be trained and updated in real-time on edge devices, facilitating quick decision-making and inference without relying on a centralized server [8]. It is inherently scalable as it distributes computation across numerous edge devices. This allows for large-scale deployment of edge AI systems without overburdening any single device or central server. Models trained using FL can adapt to changing data distributions and environmental conditions in real-time, making them well-suited for dynamic edge computing environments. Here's a step-by-step explanation of how it works:

1. Initial model distribution: a central server initializes a global ML model, which could be a neural network or another type of model (global model initialization). The initial version of this model is then sent to all participating edge devices. Each device receives a copy of the same model (model distribution).
2. Local training on edge devices: each edge device has its own local dataset, which could be user-specific data like text messages, images, sensor data, or application usage patterns. Each device trains the received global model on its local data. This training involves several iterations of an optimization algorithm (e.g., stochastic gradient descent) to update the model's parameters based on the local dataset. After local training, each device computes the changes to the model parameters (e.g., gradients or weight updates) based on its local data.
3. Communication with the central server: instead of sharing the actual data, each edge device sends only the model updates (i.e., the changes in the model parameters) to the central server. These updates can be encrypted or processed using techniques like differential privacy to ensure that sensitive information from the local data is not exposed.
4. Aggregation of updates: the central server collects updates from multiple devices. It then aggregates these updates to form a new global model. A common method is to average the updates, but more sophisticated

techniques can be used to ensure robustness against outliers or malicious updates. The central server updates the global model based on the aggregated information and prepares it for the next round of training.

5. Iterative process: the updated global model is then redistributed to all participating edge devices. Steps 2 through 4 are repeated for several iterations (or rounds) until the model converges, meaning that further updates result in minimal improvement.
6. Final model deployment: once the model has reached a satisfactory level of accuracy, it can be deployed for inference on the edge devices, allowing them to make predictions locally based on the trained model.

Overall, FL in Edge AI offers a decentralized and privacy-preserving approach to ML that is well-suited for edge computing environments, including IoT, smart cities, autonomous vehicles, and more. It addresses challenges related to data privacy, communication overhead, scalability, and adaptability, making it a powerful paradigm for deploying AI applications at the edge.

2.1. Importance of federated learning

FL is a training method for deep-learning AI models that involves collaboration. FL takes models to user's devices for training with local data until they mature, instead of centralizing customer data in a single repository. The fully trained models are then sent back to the provider or business. This method ensures that the AI provider doesn't access any end-user data while training, preserving data privacy while still making crucial use of end-user data for model improvement. Edge AI is an AI system that runs AI-driven operations closer to where the actual user data is located instead of on a centralized server. A combination of edge computing and AI is used to enable digital services to use AI capabilities locally without the need for central cloud connectivity. When FL is applied to Edge AI, it enables Edge AI applications to continuously evolve their understanding of end-user dynamics without the need for taking end-user data to their central cloud storage. End users can take advantage of this by not having to share sensitive data with any business.

The combination of FL and Edge AI allows for the development of more robust and privacy-preserving AI systems that can learn and adapt in real-time, precisely where the data is created, at the network's edge. This results in faster response times, a decrease in network latency, and improved data privacy. FL in Edge AI offers significant advantages that cater to the unique demands of edge computing environments. Here's a deeper dive into its importance:

1. Privacy preservation: FL enables AI merely sending raw data to a distant server, models can be trained locally on common devices like smartphones or IoTs sensors. This protects user privacy by keeping sensitive data decentralized and lowering the likelihood of data breaches or privacy violations [9].
2. Reduced latency: FL reduces the requirement to send data to a central server for processing by executing model training and inference on edge devices. This decreases latency and provides real-time responsiveness, making it perfect for low-latency interactions in applications like self-driving cars and augmented reality [10].
3. Bandwidth conservation: huge amounts of data transmitted from edge devices to a central server can strain network bandwidth and result in significant expenditures, particularly in settings with restricted connection. FL alleviates this strain by conducting model changes locally, which saves bandwidth and reduces network congestion.
4. Robustness to connectivity issues: edge devices frequently operate in areas with inconsistent or unpredictable network access. FL is resistant to such obstacles because it allows devices to learn and draw conclusions independently even when they are removed from the network.
5. Adaptability and personalization: FL allows AI models to be tailored and adjusted to specific edge devices or people without sacrificing privacy. Customized advice and services are made possible by this individualized approach, which also improves user experience.

FL in Edge AI offers a privacy-preserving, low-latency, and bandwidth-efficient approach to training AI models directly on edge devices, making it indispensable for multiple applications in the IoTs, health care, smart cities, and various other fields.

2.2. Various federated learning model

In this section, we explain and compare different types of FL, such as horizontal FL (HFL), vertical FL, federated transfer learning (FTL), and cross-silo FL, based on their features as shown in Table 1.

2.2.1. Horizontal federated learning

HFL is a form of FL in which datasets from multiple nodes share the same feature space but utilize various samples. It can also be referred to as sample-based FL or homogeneous FL. It works well when there is significant overlap in the user features of two datasets but not in the total number of users. In order to extract the portion of the data where user attributes are similar but users are not precisely the same for

training, we divide the datasets horizontally (by the user dimension) in this learning process. Multi-task FL reduces communication costs and improves fault tolerance compared to distributed multi-task learning. sensitive information is preserved off the server by using client-specific data division. After calculating the local gradient and uploading it to the server, each client modifies the global model to account for the gradient changes [11]. The Figure 2 shows the architecture of HFL.

The working of HFL consists the following steps:

a. The remote server receives an encoded gradient from the local model.
b. The server handles the safe combination.
c. The model receives updates from the server.
d. The models are updated based on the information from the server.

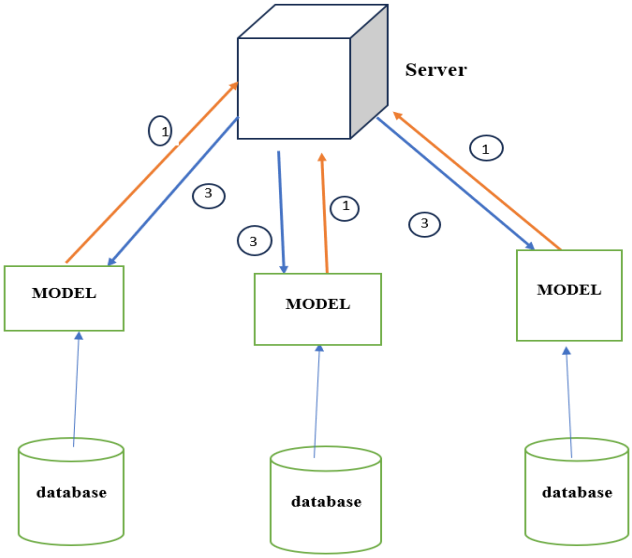

Figure 2. Architecture of HFL

2.2.2. Vertical federated learning

Vertical federated learning (VFL) is a specialized form of FL, designed to enable multiple organizations or entities to collaboratively train ML models without sharing their raw data. It's particularly useful when these organizations possess different types of data about the same set of users or entities. In VFL, the data is partitioned vertically, meaning different organizations hold different features or attributes of the same users. For example, a bank might have financial data about its customers, while a healthcare provider might have medical data about the same individuals. These organizations want to collaborate to build a better model, but they cannot share their raw data due to privacy concerns. The organizations collaborate to train a ML model by sharing encrypted intermediate computations instead of raw data. Each organization contributes to the model by using its local data to compute certain aspects (e.g., gradients or model updates) and shares these with the other parties in a secure manner. VFL employs various cryptographic techniques, such as secure multi-party computation (SMPC) or homomorphic encryption (HE), to ensure that while the computations are shared, the actual data remains private. This allows the organizations to learn from each other's data without actually seeing it. A crucial step in VFL is aligning the data across the different parties. Since each organization has data on the same users but in different forms, they need to ensure they are working with the same users without directly sharing identifiable information. This is often done through secure alignment protocols that match users across datasets based on encrypted identifiers. Figure 3 shows the architecture of vertical FL.

A typical VFL procedure for each learning time frame has seven important steps [12]:

a. Private set intersection (PSI): to align training data samples, the system uses PSI or secure entity alignment to identify common identifiers shared by all participants, including guest and host organizations. PSI is a secure system that identifies common IDs among multiple participants' data. Commonly used PSI approaches include naïve hashing, oblivious polynomial evaluation, and oblivious transfer.

b. Bottom model forward propagation: after aligning data samples, participants will use local data to do forward propagation based on their bottom model. The forward propagation procedure is similar to conventional training, with the exception of determining the loss value.
c. Forward output transmission: each participant must provide their forward output to the label owner. The forward output represents the intermediate outcomes of local neural networks that translate the original properties into features.
d. Backward output transmission: each guest participant receives the gradients of their forward output. The communication cost (transmission bits) for gradients is typically lower than for intermediate outputs.
e. Bottom model backward propagation: participants change their bottom model parameters depending on local data and the label owner's forward outputs.

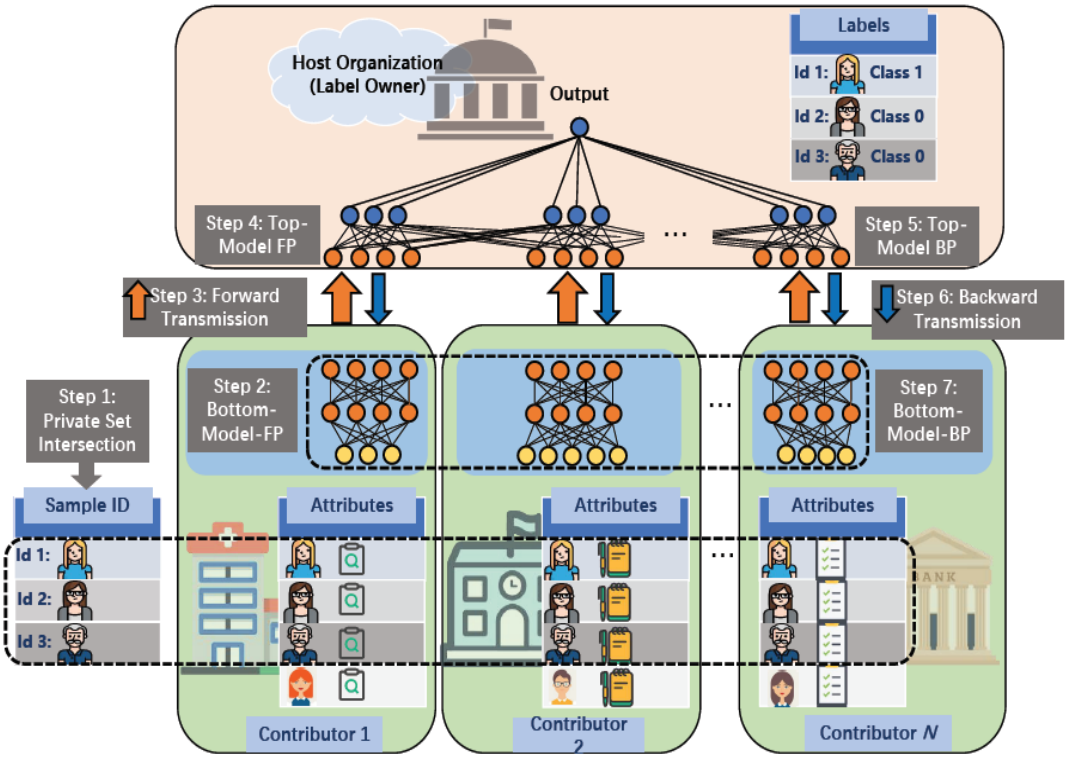

Figure 3. Architecture of VFL [12]

2.2.3. Federated transfer learning

FTL, similar to standard ML, involves adding a new feature to a pre-trained model. Extending vertical FL to include sample instances from non-collaborating organizations is a good example that promotes complementary exchange of information between the domains in a data federation and knowledge sharing without jeopardizing user privacy. This allows a target-domain party to leverage rich labels from the source domain to develop adaptable and powerful models. This methodology delivers the same degree of accuracy as non-privacy-preserving transfer learning methods with very little adjustment to the prevailing structure. It adapts well to secure multi-party ML workloads [13]. Figure 4 shows architecture of FTL. Here are the typical steps involved in FTL:

a. Initial setup and data preparation: identify the participating entities (clients) and their respective datasets. Preprocess and standardize data across all clients to ensure consistency.
b. Pre-trained model selection: select a pre-trained model that will be used as the base model for transfer learning. This model is typically trained on a large, diverse dataset and provides a good starting point.
c. Local model customization: each client fine-tunes the pre-trained model on their local data. This step involves: downloading the pre-trained model, Training the model on local data by adjusting model weights based on local dataset characteristics.
d. Local model updates: after local training, each client computes the model updates (gradients or model parameters) based on their local dataset.

e. Secure aggregation: to protect data privacy, the model updates are securely aggregated. This can be done using techniques like secure multiparty computation (SMC) or differential privacy to ensure that individual updates remain confidential. Clients send their encrypted model updates to a central server or an aggregator.
f. Global model update: the central server or aggregator decrypts and aggregates the local updates to update the global model. This step involves:
   i. Combining the updates from all clients.
   ii. Applying the aggregated updates to the global model.
g. Global model distribution: the updated global model is then distributed back to all clients.
h. Iterative process: steps 3 to 7 are repeated iteratively. In each iteration, the global model becomes more refined as it learns from the diverse local datasets of all clients.
i. Convergence and final model: the process continues until the global model converges to a satisfactory performance level or a predefined number of iterations is reached. The final model is then used by all clients for inference on their local data.
j. Evaluation and deployment: evaluate the final model's performance on a validation dataset. Deploy the final model for real-world use.

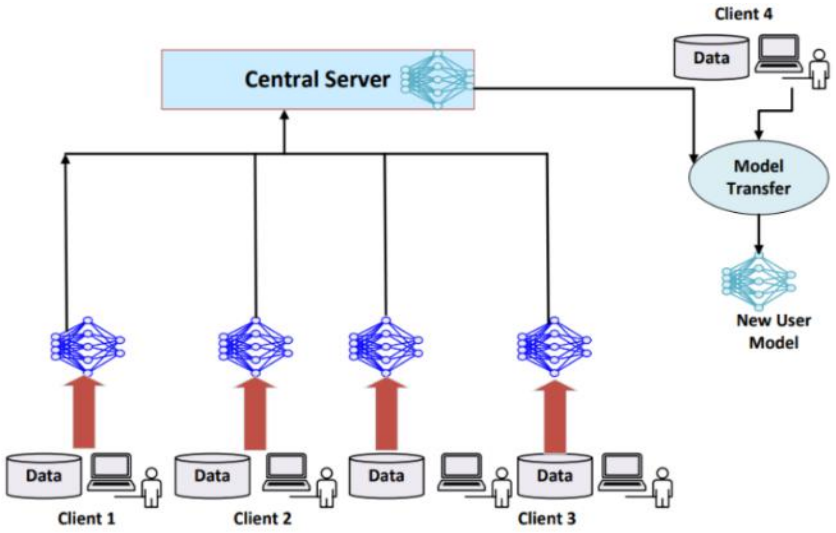

Figure 4. Illustration of federated transfer learning architecture [13]

2.2.4. Cross-silo federated learning

Cross-silo FL is utilized when there are fewer participating devices and they are available for all rounds. The training data may be in horizontal or vertical FL format. Cross-silo is mostly utilized for organizational cases [14]. Cross-silo FL is a collaborative ML strategy in which multiple companies, each with its own data silo, work together to train a common model without sharing their raw data. Instead, they exchange model updates or gradients while keeping their data local, which protects privacy and security. This method allows enterprises to harness collective expertise while protecting sensitive data. Figure 5 shows the architecture of cross-silo FL.

A typical cross-silo FL method comprises multiple rounds. Each round has four steps:

a. The central server provides the clients with the global model from the previous cycle. The downloaded model is initialized at random in the first round.
b. Users train acquired models on private local data sets, resulting in updated local models.
c. Local model updates are uploaded by clients to the central server.
d. In the subsequent round, clients will receive a new global model that the server has created by combining the submitted model revisions.

In the below Table 1 shows the comparison between different types of FL such as HFL, VFL, FTL, and cross-silo FL based on their features like data distribution, model architecture, communication overhead, model accuracy, client heterogeneity and use case.

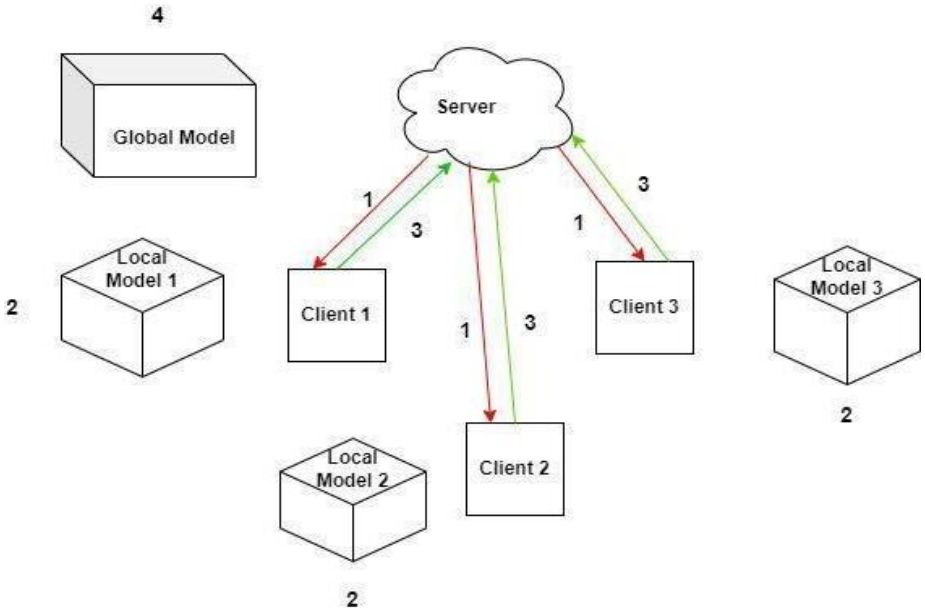

Figure 5. Architecture of cross-silo federated learning

Table 1. Comparison between different types of federated learning

Features	HFL	VFL	FTL	Cross-silo FL
Data distribution	Data is distributed horizontally, with each client having a subset of the total data	Data is distributed vertically, with each client having a subset of the features	Data across clients is often non-IID (non-Independent and Identically Distributed), unbalanced and different feature spaces	IID or non-IID, overlapping feature spaces
Model architecture	Typically uses a single, shared model architecture across all clients	Requires a customized model architecture for each client as each client has a unique set of features	Different architectures, fine-tuning	Same architecture across clients
Communication overhead	High communication overhead	Reduce communication overhead	Lower, but variable	Higher, but predictable
Model accuracy	High model Accuracy	Require additional technique to achieve high model accuracy	Variable, depends on transfer effectiveness	Generally high, consistent improvements
Client heterogeneity	Homogenous client with similar data distribution	Heterogeneous client with different data distribution	High, different data and capabilities	Low to moderate, similar data and resources
Use case	Suitable for applications with horizontally distributed data such as images, videos	Suitable for application with vertically distributed data such as financial transactions, medical records	Personalized, domain-specific adaptations	Collaborative learning across organizations with similar data

2.3. Applications

2.3.1. Healthcare

Distributed intelligence using edge devices is made possible by edge AI, an interdisciplinary technology. It involves models for FL, DL, and ML that are developed and run at the edge of the network, independent of centralized data centers. It also includes data analytics and AI. Edge computing allows for local data processing and analysis, which lowers latency and facilitates quick decision-making [15].

A distributed ML technique called FL uses information from several decentralized edge clients to build a global model. Data protection and scalability are only two of its many benefits. When dealing with heterogeneous devices, FL techniques also come with some dangers and computational complexity limitations. Healthcare providers, for example, can use FL to collectively train a model, leveraging the pooled expertise of all participating entities while ensuring that each entity's data stays private [16].

2.3.2. Finance

Edge AI is a technique that makes edge devices capable of distributed intelligence. It enables the deployment and execution of AI and data analytics at the network's edge, away from centralized data centers. This allows for faster decision-making and lower latency [17]. A ML method called FL uses a network of decentralized edge clients to learn from in order to create a global model. Scalability and data privacy are among the benefits it provides. But when it comes to processing complexity, it does cause some problems when dealing with heterogeneous devices.

Furthermore, FL enables Edge AI apps to continuously improve their understanding of end-user dynamics without having to transfer end-user data to a central cloud store. This gives end users an edge because they do not have to disclose sensitive information with any firm. This ensures that no personal data leaves the device. For example, financial organizations might train a model cooperatively using FL, which allows them to exploit the aggregate intelligence of all participating institutions while ensuring that each institution's data stays private [18].

2.3.3. Smart homes

Edge AI is a revolutionary development in AI that fundamentally changes how we think about data processing and device interaction. Whether it's an edge server, smartphone, or other IoT device, the magic happens right there or very close to the data source. This change fundamentally affects how quickly and efficiently devices can function, not just where they are located. Real-time decision-making and quicker replies are made possible by Edge AI, which reduces the latency of transferring data back and forth to remote computers. Furthermore, data privacy benefits greatly from this specialized processing. Edge AI maximizes user control by reducing the inherent dangers associated with frequent data transfers to external servers by storing critical information closer to home [19]. Conversely, FL allows for joint AI training without jeopardizing the privacy of personal information. With this ML technique, all training data is retained on the original device and a shared global model is learnt across several devices. This promotes privacy because no raw data is shared or kept in a central location. One example of this is the Fed Home framework, which is an architecture for in-home health monitoring based on the cloud edge. It creates a shared global model in the cloud by utilizing several houses at the network edges, and it preserves user privacy by storing user data locally.

2.3.4. Telecommunication

In the context of telecommunications, edge AI refers to the application of AI algorithms to network edge devices like switches, routers, and other devices. This facilitates quicker answers and real-time decision-making, both of which are essential in a communications network. Additionally, Edge AI can help safeguard data privacy because it processes data locally rather than transferring it to a central server [20]. FL is becoming increasingly popular in the telecom sector as communication service providers (CSPs) consider how to leverage their data assets while upholding privacy regulations. Over five billion consumers' data are stored by the top 50 carriers worldwide. The utilization of FL is increasingly important in the development of centralized models with distributed training data, as telecommunication companies employ AI/ML technology to extract analytical and predictive capabilities. Significantly more network capacity, reduced latency, faster speeds, and greater efficiency are made possible by 5G and edge computing. 5G Edge computing will disperse data and AI models among numerous nodes, however sharing the data can be difficult due to security, bandwidth, storage, and other limitations. FL is perfect for this kind of setting.

2.3.5. Smart farm

FL-based monitoring systems for smart farms detect animal diseases. Unlike prior studies, which did not use FL for animal disease diagnosis, this technique is based on extensive experimentation with information from the internet of animal health things (IoAHT). These studies on clinical mastitis in cows provide a strong foundation for assessing FL's efficacy in actual agricultural settings. Smart farming involves solar-powered sensors attached to each animal to monitor their health. The information is periodically sent via long-range (LoRa) transmission to edge devices, such gateways, and subsequently to a cloud server. Farmers are able to effectively oversee farm operations and keep an eye on livestock thanks to this infrastructure. While solar-powered sensors offer labor-saving and environmental benefits, they also pose substantial problems. To enhance smart farming automation, a cloud server can use a deep learning (DL) model to analyze data from gateways and detect diseases like mastitis in cattle. This approach protects data privacy and encourages sustainable agricultural practices by enhancing disease prediction in smart farms through the use of FL. Local elements, including sensors, are crucial to improving FL's forecast accuracy [21].

2.4. Privacy threats in federated learning

Privacy flaws are among the most common worries about traditional ML. To preserve their privacy, FL requests that participants contribute the local training model parameters rather than their actual data. The dangers present in FL can be broadly classified into many types of inference-based assaults. The main privacy protection issues and data security risks that FL faced while working at EC include [22]:

a. Model inversion attack: by decentralizing the training process and enabling devices to cooperatively learn a shared model while maintaining local access to the raw data, FL seeks to safeguard data privacy. However, the model parameters and updates exchanged during FL can still leak information. The server gathers updates from client models in FL. An attacker can deduce details about the training set by examining the parameters if they manage to obtain access to either the global model or these updates. The assailant reconstructs the input data using the model parameters. This can be accomplished by optimizing an input so that the observed outputs from the valid mode coincide with the outputs from the model (or intermediate layer outputs). An attacker can exploit these gradients to approximate the training data because FL requires exchanging gradients or model updates [23].
b. Model poisoning: model poisoning attack is also known as adversarial attack. A malicious client might send manipulated model updates to the central server. These updates can be crafted to degrade the performance of the global model. This could be done by training on poisoned data or by deliberately introducing errors into the model updates.
c. Backdoor attacks: in this scenario, an attacker injects a hidden backdoor into the global model. This backdoor activates when the model encounters specific trigger inputs, causing it to behave incorrectly while performing normally on regular data. The backdoor can be implanted by subtly modifying the model updates.
d. Data poisoning: although this is not a refactoring of the model updates themselves, data poisoning involves injecting malicious data into the training dataset. This can indirectly cause the model to learn incorrect or harmful patterns, affecting its performance.
e. Model extraction attack: in FL, a model extraction attack occurs when a hostile party uses their local data to try and rebuild or extract sensitive information about the global model that is being trained. The fact that FL frequently uses proprietary or sensitive models and that its objective is to preserve individual data privacy while learning a global model makes this assault especially worrisome.

2.5. Privacy preservation techniques

Global and local privacy are the two categories into which FL privacy falls. Every iteration, all unreliable third parties' privacy is safeguarded, with the exception of the trusted central aggregation server, using global privacy regulations and locally created model updates. For the server privacy to be protected, model changes are needed for local privacy. Currently, adversarial training (AT), blockchain, disturbance, cryptography, and KD are common technologies for enhancing FL privacy security.

2.5.1. Differential privacy technology

Fuzzy processing is commonly employed in DML to safeguard training dataset's privacy and security. This includes using generalization, noise disturbance, randomization and compression to conceal training data and improve privacy performance to some extent. In FL, DP is typically employed to disguise significant features by adding noise to training data, model parameters, or gradient information. DP can ensure data privacy. DP improves data privacy and security by adding noise into sensitive data. The use of DP in FL to introduce noise disturbances to model parameters provided by FL participants, or to apply generalization methods to disguise critical data features, prevents reverse data retrieval, allowing ML models to tolerate adversarial examples [24].

The communication overhead of SMPC is much higher than that of DP. DP algorithms have been developed in existing studies. An algorithm for DP-protection FL optimization on the client side. To provide a user-level DP training process for large neural networks, a user-level privacy protection is additionally incorporated to the FL averaging method. Both sought to protect private data by masking user-uploaded model parameters during training, weighing model performance against privacy loss. These methods were tested on genuine datasets. This demonstrated that with enough devices participating in federated training, privacy protection can be achieved with low overhead. Both approaches ensured high model correctness.

However, this technique neglected to consider the possibility that incorporating DP in FL with fewer participants may impair overall model accuracy. DP noise was substituted into a neural network by pruning a specific layer, with the purpose of safeguarding sensitive data from leaking while maintaining model accuracy. There is a novel privacy-preserving learning framework based on graph neural networks (GNNs). The framework provides a formal privacy guarantee by utilizing edge-local DP to protect node features and edge privacy. The system combines a GNN with a privacy utility to secure user data privacy within a budget.

2.5.2. Cryptography technologies

FL employs encryption to protect model parameter data, which must be uploaded in plaintext. This strategy enhances the privacy and security characteristics of FL systems. Currently, the most widely used encryption algorithms are SMPC and HE [25]. Each encryption technology has distinct technological characteristics. SMPC can keep user input data confidential and allow for cooperative calculation on private data, but the compute overhead is substantial. Security-wise, HE schemes operate similarly to SMPC. This requires fewer processing resources than SMPC.

SMPC, also known as MPC, was originally created to protect the inputs of numerous participants. In the FL framework, the SMPC is used to secure clients' model modifications. SMPC ensures that each participant in the FL system only knows its own inputs and outputs, with no knowledge of other clients. Using SMPC to build a FL security model can increase efficiency by reducing security needs. The SMPC is a lossless approach that allows many parties to perform collaborative computations on sensitive data. SMPC guarantees data confidentiality and privacy protection. Despite being a research-oriented approach, the SMPC-based FL privacy protection strategy has significant obstacles. The key issue is balancing the effectiveness of the FL system with privacy concerns. SMPC encryption and decryption can be time-consuming, potentially impacting model training. Developing a lightweight SMPC system remains a big challenge. As a result, many scholars prefer higher education technology. HER technologies are preferred by researchers when creating FL security protocol frameworks due to their lower computation consumption, even with the same security performance scheme.

HE is a method of encrypting plaintext and ciphertext that enables third parties to process data while protecting sensitive information. It has been included in the FL framework to protect private information from enemies. Early HE algorithms relied on single-key arithmetic, which increased the risk of private key leakage and malicious client access to other clients' data. However, there is also a need to address security issues with present cryptography approaches in FL, such as member inference attacks and reverse attacks that cause privacy leaks in training data. Additional computing is also required. There is still space for technological innovation in HE, including improving processing efficiency, interaction logic complexity, and secret sharing systems to reduce communication delays and bandwidth costs.

2.5.3. Adversarial training

In Florida, information privacy-preserving strategies built on disruptive technologies and encryption have gained popularity recently. These technologies are mostly concerned with secure local computing, parameter encryption, and raw data. The risk of privacy leakage in distributed learning is decreased when computation results are transferred to a third party. Generative adversarial networks (GANs) can be used by malicious attackers to steal data from trustworthy parties. Computed gradient information gives the attacker the ability to reverse some or all of the private data. GANs have been proposed as a way to steal personal information on FL systems recently. A great deal of study has been done to protect privacy in Florida from adversarial attacks. The primary objectives include detection and defense. There are three basic directions for defending against adversarial attacks [26].

- During the testing phase, make changes to the input sample or the training protocol.
- Change the neural network, for instance, by adding or deleting sublayers and raising the activation or loss function.
- Identify or completely categorize hostile samples.

Using potential limits, the latent-boundary-guided AT approach trained DNN models on adversarial samples. By introducing disturbances to potential characteristics, superior adversarial sample samples were generated. The trade-off between adversarial toughness and standard accuracy was improved by this tactic. Generally speaking, AT improves user data privacy. By using AT samples, the chance of drawing conclusions from the real training data is decreased. The trade-off between standardization and robustness has garnered significant attention in recent studies on enhancing AT robustness. Additionally, this creates new opportunities for the advancement of federal confrontation training in the future.

2.5.4. Blockchain

Conventional, centralized FL frameworks rely on a single aggregating server, resulting in a single point of failure. The central node incurs higher costs and performs less efficiently when communication is heavy. Participants are not motivated to engage in cooperative learning due to a lack of incentives. There are also insufficient security mechanisms to detect malevolent users that breach the model. To remedy these issues, academics have coupled blockchain and FL. The blockchain's participating nodes replace the central server, reducing the risk of single-point failures.

Next, miner nodes calculate local device model parameters without downloading raw data. The local device model modifications are then verified and recorded using the blockchain's consensus mechanism.

Local devices upload aggregate model parameters, while global updates are applied to fresh blocks. The local devices download the global model from blockchain blocks [27].

In blockchain, privacy-preserving Byzantine robust FL (PBFL) employed cosine similarity to identify gradients uploaded by malicious clients. Secure aggregation was achieved using full HE. To limit the effects of poor clients and central servers, PBFL often uses a blockchain system to facilitate the execution of transparent PBFL rules and processes. The integration of blockchain and FL technology has significantly improved the traditional FL area. However, even after integrating these two technologies, there are still issues related to blockchain. Traditional blockchain consensus mechanisms and network topology can lead to issues including slow transaction confirmation, limited throughput, and complex communication structures. As a result, there are delays in the blockchain network's model parameter aggregation for every FL cycle. Every FL participant makes use of a different local device. Different devices may experience different latency delays when updating a model on the blockchain network. This can result in decreased prediction accuracy for the trained global model.

With the present problems with blockchain-based FL frameworks, there is a growing movement toward the current decentralized FL architecture approach. In low-bandwidth or high-latency networks, decentralized training performs better than centralized training for federated systems. The suggested asynchronous FL architecture, in conjunction with blockchain technology, eliminates the possibility of model parameter manipulation. The asynchronous FL expedites global aggregation at the same time. Asynchronous FL frameworks based on blockchain technology can solve issues with existing FL technology development by striking a compromise between efficiency, security, and anonymity.

2.5.5. Knowledge distillation

The idea behind Knowledge distillation (KD) technology was to transmit knowledge from large models to small models. Conventional ML and deep learning techniques are susceptible to privacy breaches. Since KD in FL allows model training without centrally gathering potentially sensitive raw data, it can improve privacy. This offers more robust privacy protections for predictive model construction. Even with the benefits of anonymity, KD in Florida is not risk-free. For example, KD usually uses a proxy dataset, and the quality, size, and feature distribution of the publicly available shared dataset can have a big influence on how well the model performs in terms of generalization and accuracy. There's also a significant chance of privacy leaking.

Mechanisms for selective information exchange for federated distillation have been proposed to reduce these hazards. The objective of these systems is to discern exact and accurate knowledge from local and ensemble forecasts, in that order. This method routinely beats baseline and improves the Federated Distillation framework's capacity for generalization.

3. DISCUSSION AND FUTURE SCOPE

The systematic review reveals that FL has emerged as a significant paradigm in distributed ML, particularly in scenarios where data privacy is a concern, and data is distributed across multiple devices or locations. FL allows for the training of models across decentralized data sources without the need to transfer raw data to a central server, making it especially relevant in the context of Edge AI. The review identifies VFL as a key type where different entities possess different feature spaces for the same sample set. This is particularly useful in scenarios where organizations with complementary data (e.g., banks and insurance companies) can collaborate without sharing raw data. HFL is another common type where different entities or devices have data with the same feature space but different samples. This is typical in scenarios like mobile devices where similar data types (e.g., user activity data) are distributed across different devices.

FTL combines FL with transfer learning to address situations where both the feature space and the sample set differ across participants. This is useful when organizations with limited data can benefit from the knowledge transferred from another domain. Cross-silo FL involves a smaller number of participants, typically organizations or institutions, who collaborate over long periods. Cross-silo FL is applicable in sectors like healthcare and finance where collaboration is essential but the number of participating entities is relatively small and stable. The review identifies model inversion attacks as a significant threat, where adversaries attempt to reconstruct input data by exploiting the model updates shared during FL. This type of attack can compromise the privacy of participants' data, especially in scenarios with sensitive information.

Model poisoning is another identified threat where malicious participants intentionally corrupt the model by sending harmful updates. This can degrade the model's performance or even introduce biases, posing a significant challenge in maintaining the integrity of the FL process. Backdoor attacks involve embedding hidden triggers in the model that cause it to behave maliciously under certain conditions. This type of attack is particularly dangerous as it can go unnoticed during training and only activates under specific inputs. In data poisoning attacks, adversaries inject malicious data into the training process to skew

the model's output. This type of attack is challenging to detect and can significantly undermine the trustworthiness of the model.

Model extraction attacks focus on stealing the model by analyzing the outputs of the FL process. These attacks can lead to intellectual property theft and reduce the competitive advantage of the entities involved in FL. We also discussed privacy preservation techniques in FL such as Differential privacy is highlighted as a critical technique for preserving privacy in FL by adding controlled noise to the model updates. This ensures that the contribution of any single participant's data is obfuscated, reducing the risk of data leakage through model inversion attacks. The review discusses the use of cryptographic methods, such as SMPC and HE, which allow participants to perform computations on encrypted data, ensuring that the model updates remain confidential even during aggregation.

AT is mentioned as a method to make models more robust against adversarial attacks by training the model on data that includes adversarial examples. This approach can help mitigate the risk of model poisoning and backdoor attacks. The integration of blockchain technology with FL is explored as a means to enhance security and transparency. Blockchain can provide a decentralized and immutable record of model updates, ensuring that the contributions of each participant are verifiable and tamper-proof. KD is discussed as a technique to reduce the complexity of FL models while maintaining performance. By distilling the knowledge from a large model to a smaller one, the risk of privacy leakage can be minimized, as the distilled model reveals less about the underlying data.

Challenges and future directions:

1. Scalability: one of the key challenges identified is the scalability of FL, particularly in large-scale networks with numerous and diverse participants. The review suggests the need for more efficient communication protocols and aggregation techniques to make FL more scalable and responsive [28].
2. Robustness to heterogeneous data: the heterogeneity of data across different participants poses a challenge in FL, as varying data distributions can negatively impact the global model's performance. Future research should focus on developing algorithms that are robust to these differences, ensuring consistent model performance across diverse datasets.
3. Incentive mechanisms: effective incentive mechanisms are necessary to encourage participation in FL. Since FL relies on voluntary collaboration, particularly in cross-silo scenarios, exploring fair and transparent reward systems could drive greater adoption and sustained participation [29].
4. Regulatory and ethical considerations: as FL continues to evolve, addressing regulatory and ethical challenges related to data privacy, security, and bias is crucial. The review suggests that future work should focus on developing guidelines and frameworks that ensure FL implementations comply with legal standards and ethical principles [16].

4. CONCLUSION

The systematic review provides a comprehensive examination of the current state of FL in Edge AI, emphasizing its potential and challenges. The findings demonstrate that while FL offers significant advantages in terms of privacy preservation and decentralized learning, it faces several challenges, particularly regarding scalability, data heterogeneity, and security threats. The discussion underscores the importance of ongoing research and development in privacy-preserving techniques and the integration of emerging technologies like blockchain to unlock the full potential of FL in various applications. FL reduces the hazards connected with centralized data collecting and processing by decentralizing the training process and maintaining data localized on edge devices. New technological development lessens privacy concerns for edge devices. There are several privacy issues with traditional machine-learning methods. An efficient FL approach can help us enhance edge device performance. Numerous industries, including telecommunications, healthcare, and smart cities, can benefit from FL. FL enhances privacy on edge devices by combining blockchain and cryptographic technology. The future of FL in Edge AI will depend on addressing these challenges and refining the techniques to ensure robust, secure, and efficient decentralized learning systems.

FUNDING INFORMATION

The authors would like to express sincere gratitude to the institution management for providing financial support for this research. Their financial assistance through research schemes, YRF and ERF, has allowed us to get fruitful research experiences.

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] R. Singh and S. S. Gill, "Edge AI: a survey," *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 71–92, 2023, doi: 10.1016/j.iotcps.2023.02.004.

[2] J. Yang, T. Baker, S. S. Gill, X. Yang, W. Han, and Y. Li, "A federated learning attack method based on edge collaboration via cloud," *Software: Practice and Experience*, vol. 54, no. 7, pp. 1257–1274, Jul. 2024, doi: 10.1002/spe.3180.

[3] M. Al-quraan, G. S. Member, L. Mohjazi, S. Member, S. Muhaidat, and S. Member, "Edge-native intelligence for 6g communications driven by federated learning : a survey of trends and challenges," in *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2023, pp. 1–24.

[4] S. Pandya *et al.*, "Federated learning for smart cities: a comprehensive survey," *Sustainable Energy Technologies and Assessments*, vol. 55, p. 102987, Feb. 2023, doi: 10.1016/j.seta.2022.102987.

[5] H. Li *et al.*, "Review on security of federated learning and its application in healthcare," *Future Generation Computer Systems*, vol. 144, pp. 271–290, Jul. 2023, doi: 10.1016/j.future.2023.02.021.

[6] Y. Liu, X. Yuan, Z. Xiong, J. Kang, X. Wang, and D. Niyato, "Federated learning for 6G communications Challenges methods and future directions," *China Communications*, pp. 105–118, 2020.

[7] J. P. Singh, "Advancing edge security: AI and ML innovations for robust cyber defense," *International Journal of Marketing and Technology*, vol. 14, no. 2, 2024.

[8] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, Mar. 2021, doi: 10.1016/j.knosys.2021.106775.

[9] H. Li, L. Ge, and L. Tian, "Survey: federated learning data security and privacy-preserving in edge-internet of things," *Artificial Intelligence Review*, vol. 57, no. 5, p. 130, Apr. 2024, doi: 10.1007/s10462-024-10774-7.

[10] M. Shaheen, M. S. Farooq, and T. Umer, "AI-empowered mobile edge computing: inducing balanced federated learning strategy over edge for balanced data and optimized computation cost," *Journal of Cloud Computing*, vol. 13, no. 1, p. 52, Mar. 2024, doi: 10.1186/s13677-024-00614-y.

[11] Q. W. Khan, A. N. Khan, A. Rizwan, R. Ahmad, S. Khan, and D.-H. Kim, "Decentralized machine learning training: a survey on synchronization, consolidation, and topologies," *IEEE Access*, vol. 11, pp. 68031–68050, 2023, doi: 10.1109/ACCESS.2023.3284976.

[12] K. Wei *et al.*, "Vertical federated learning: challenges, methodologies and experiments," *arXiv*, 2022, [Online]. Available: http://arxiv.org/abs/2202.04309.

[13] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang, "A secure federated transfer learning framework," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 70–82, Jul. 2020, doi: 10.1109/MIS.2020.2988525.

[14] C. Huang, J. Huang, and X. Liu, "Cross-silo federated learning: challenges and opportunities," *arXiv*, 2022, [Online]. Available: http://arxiv.org/abs/2206.12949.

[15] A. Manocha, S. K. Sood, and M. Bhatia, "Edge intelligence-assisted smart healthcare solution for health pandemic: a federated environment approach," *Cluster Computing*, vol. 27, no. 5, pp. 5611–5630, Aug. 2024, doi: 10.1007/s10586-023-04245-x.

[16] Q. Wu, X. Chen, S. Member, Z. Zhou, and J. Zhang, "Fedhome: cloud-edge based personalized federated learning for in-home health monitoring," *IEEE Transactions on Mobile Computing*, vol. 21, no. 8, pp. 2818–2832, 2020.

[17] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106854, Nov. 2020, doi: 10.1016/j.cie.2020.106854.

[18] J. White and P. Legg, "Federated learning: data privacy and cyber security in edge-based machine learning," in *Data Protection in a Post-Pandemic Society*, Cham: Springer International Publishing, 2023, pp. 169–193.

[19] M. Khan, F. G. Glavin, and M. Nickles, "Federated learning as a privacy solution - an overview," *Procedia Computer Science*, vol. 217, pp. 316–325, 2023, doi: 10.1016/j.procs.2022.12.227.

[20] U. Mangla, "Application of federated learning in telecommunications and edge computing," in *Federated Learning*, Cham: Springer International Publishing, 2022, pp. 523–534.

[21] T. Oh, S. Chung, B. Lunt, R. McMahon, and R. Rutherfoord, "SusFL: energy-aware federated learning-based monitoring for sustainable smart farms," in *SIGITE 2017 - Proceedings of the 18th Annual Conference on Information Technology Education*, 2017, pp. 39–40.

[22] N. Bouacida and P. Mohapatra, "Vulnerabilities in federated learning," *IEEE Access*, vol. 9, pp. 63229–63249, 2021, doi: 10.1109/ACCESS.2021.3075203.

[23] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, May 2020, doi: 10.1109/MSP.2020.2975749.

[24] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, Feb. 2021, doi: 10.1016/j.future.2020.10.007.

[25] R. Gosselin, L. Vieu, F. Loukil, and A. Benoit, "Privacy and security in federated learning: a survey," *Applied Sciences*, vol. 12, no. 19, p. 9901, Oct. 2022, doi: 10.3390/app12199901.

[26] A. Brecko, E. Kajati, J. Koziorek, and I. Zolotova, "Federated learning for edge computing: a survey," *Applied Sciences*, vol. 12, no. 18, p. 9124, Sep. 2022, doi: 10.3390/app12189124.

[27] E. Badidi, "Edge AI and blockchain for smart sustainable cities: promise and potential," *Sustainability*, vol. 14, no. 13, p. 7609, Jun. 2022, doi: 10.3390/su14137609.

[28] E. Badidi, "Edge AI for early detection of chronic diseases and the spread of infectious diseases: opportunities, challenges, and future directions," *Future Internet*, vol. 15, no. 11, p. 370, Nov. 2023, doi: 10.3390/fi15110370.

[29] H. Kaur, V. Rani, M. Kumar, M. Sachdeva, A. Mittal, and K. Kumar, "Federated learning: a comprehensive review of recent advances and applications," *Multimedia Tools and Applications*, vol. 83, no. 18, pp. 54165–54188, Nov. 2023, doi: 10.1007/s11042-023-17737-0.

BIOGRAPHIES OF AUTHORS

Christina Thankam Sajan holds a bachelor of technology (B.Tech.) degree in Computer Science and Engineering from Saintgits College of Engineering (Autonomous), Kottayam, Kerala, India, under APJ Abdul Kalam Technological University. Her academic journey is marked by a minor degree in Robotics and Automation. Currently, she is pursuing a Master of Technology (M.Tech.) degree in Computer Science and Systems Engineering at the same institution. As a dedicated young researcher, she is a recipient of the prestigious Young Research Fellowship (YRF) at Saintgits College of Engineering. Her academic journey is marked by her active engagement in innovative projects, including the development of an IoT-based RWS waste management system, a project with significant potential for smart city initiatives. Her current research focuses on sign language recognition using federated learning techniques, aiming to enhance privacy while improving accuracy. Her areas of expertise include artificial intelligence, ML, the IoT, and pattern recognition. In the future, she hopes to research advances in AI and obtain a Ph.D., continuing her career in computer science and systems engineering. She can be contacted at email: cts.se2325@saintgits.org.

Helanmary M. Sunny is a computer engineer currently pursuing postgraduate studies in Computer Science and Systems Engineering at Saintgits College of Engineering. She graduated from MBCCET Idukki, Kerala, India, in 2023 with First Class with Distinction, and she holds an NPTEL certification in Data Analytics with Python. She has technical expertise in Python, TensorFlow, and the flask framework, skills she has applied in various research and development projects. Notably, she developed a smart parking system aimed at optimizing parking in smart cities and reducing traffic accidents, utilizing IoT technology. She has been an active member of IEEE and ISTE organizations for four years, further enriching her professional network and experience. As a passionate researcher, she was awarded the Young Research Fellowship at Saintgits College of Engineering. Her research interests encompass federated learning, generative AI, the IoT, and data science. She is currently working on sign language detection using HFL to enhance both privacy and accuracy. Looking ahead, she aims to explore advancements in generative AI and pursue a Ph.D., continuing her journey in the field of computer science and systems engineering. She can be contacted at email: helanmary.se2325@saintgits.org.

Dr. Anju Pratap is a professor and former HoD in the Department of Computer Science and Engineering, Saintgits College of Engineering (Autonomous), Kerala. Prior to that, she was working as a post-doctoral researcher in National Institute of Advanced Industrial Science and Technology, Tokyo, Japan. She did her doctoral and post-doctoral research in applied artificial intelligence techniques for the common good, including psychological assessments and cancer research. She has 10+ years of research and training experience in AI and ML. She is an active member and trainer of Intel® Unnati AIIoT Lab and advanced skill acquisition programmes. She is heading the "Informatics and Computational Intelligence" stream in the department of her college. She has funded projects in the area of Applied AI for nurses, virtual learning etc. She is one among the expert judging panel of International Project Awards conducted by CSI-InAPP during years 2020,21 and 22, as well as in Darsana Ignite awards during 2021 and 2022. She won the "Innovative Faculty" Runner Up award from Project Management of India (PMI) during the year 2021. She can be contacted at email: anju.pratap@saintgits.org.