Innovative Hybrid Learning Strategies in Machine Learning: An In-Depth Look at Federated, Split Learning and Hybrid Split-Federated Learning

1st Naveen Pathak
Dept. of Electronics and Communication Engineering
Indian Institute of Technology Roorkee, India
naveen_p@ece.iitr.ac.in

2nd Hemkant Nehete
Dept. of Electronics and Communication Engineering
Indian Institute of Technology Roorkee, India
nehete_h@ece.iitr.ac.in

3rd Amit Monga
Dept. of Electronics and Communication Engineering
Indian Institute of Technology Roorkee, India
amit_m@ece.iitr.ac.in

4th Brajesh Kumar Kaushik
Dept. of Electronics and Communication Engineering
Indian Institute of Technology Roorkee, India
bkk23fec@iitr.ac.in

***Abstract*— The protection of data privacy is of paramount importance, particularly in the context of machine learning, where the demand for extensive datasets often clashes with privacy apprehensions. Conventional centralized models present notable challenges in terms of data privacy and scalability. To mitigate these challenges, hybrid learning methodologies, notably Federated Learning (FL) and Split Learning (SL), have been devised. FL and SL amalgamate the advantages of centralized models with collaborative and privacy-preserving approaches, facilitating collective training across multiple devices without centralized data storage. This paper provides a comprehensive exploration of the foundational principles, rationales, and architectures of FL and SL, highlighting their potential to revolutionize model training. Through thorough and meticulous analysis, we demonstrate how FL and SL augment the efficiency and security of model training processes, which holds significant implications in an era where data privacy is paramount. Our findings underscore the transformative influence of FL and SL on the landscape of machine learning, with substantial ramifications for both academic researchers and industry practitioners.**

I. INTRODUCTION

In the contemporary era of data-centricity, safeguarding data privacy is paramount, particularly in sensitive sectors such as finance, personal imagery, and medical records, where individuals voluntarily share their data in return for service utilization. The European General Data Protection Regulation (GDPR) Article 5 delineates the principles of data minimization and purpose limitation, ensuring that user data is only collected for specific purposes and utilized solely for its intended objectives. To combat these challenges and alleviate the computational burden of training a machine learning (ML) model, the concept of federated learning (FL) has been posited. The proliferation of data in today's landscape has rendered machine learning indispensable for enterprises and institutions to extract insights and make data-driven decisions. However, the traditional centralized approach to data gathering and storage has raised apprehensions regarding privacy and security. Machine learning models require access to personal data for effective training, giving rise to potential privacy concerns. Consequently, it is imperative to devise privacy-preserving methodologies that facilitate efficient machine learning model training while upholding the integrity of sensitive information [1-3].

The conventional approach in machine learning has traditionally relied on a centralized model for data collection and storage. However, this model has inherent limitations and drawbacks, particularly with respect to potential privacy breaches. One of the shortcomings of this approach is its heavy reliance on pre-existing knowledge and assumptions, which can introduce biases and inaccuracies into the results. Furthermore, the conventional approach may not effectively handle complex and dynamic data compared to more advanced and adaptive methods. Storing data in a centralized location creates a significant risk because it becomes a single point of failure. This means that if the server's data gets wiped out, it could potentially compromise all the stored data in case of a breach. To reduce these risks, it's crucial to put in place backup and redundancy measures. Additionally, managing large volumes of data can be costly and computationally intensive, and data silos can hinder the development of comprehensive models by keeping valuable datasets locked within specific organizations [4]. To address these concerns, innovative techniques such as federated learning (FL) and split learning (SL) have emerged as privacy-preserving and collaborative approaches to model training. These techniques represent a departure from the traditional centralized model of machine learning. Instead, FL and SL decentralize the learning process by storing data on individual devices, enabling collaborative training on decentralized data. This approach ensures privacy and security while harnessing the collective power of a vast network of devices. In essence, FL and SL facilitate collaborative machine learning model training across multiple devices without the need for centralized data storage.

Federated Learning (FL) was introduced by McMahan *et al.* [1] as a method for training a machine learning (ML) model in a distributed and privacy-preserving manner, without sharing private data among non-trusting parties. FL relies on sharing model parameters that can be aggregated to form a joint model, thus following a client-server architecture. This approach contrasts with other types of distributed ML and Machine Learning as a Service (MLaaS). FL is particularly useful in situations with massively distributed, non-IID, and unbalanced data. The training procedure involves the server initializing an ML model and sending its parameters to participating parties. The goal is to find optimal parameter values for the model to generalize well on the federated database. The server iteratively provides current model parameters to selected clients, who update their local models using methods such as stochastic gradient descent. After a predefined number of local training epochs, clients transmit updates back to the server for aggregation. This process continues until the model sufficiently converges and performs well for all clients. Research into FL training algorithms, communication protocols, and security measures has surged since 2016, driven by major companies like Google, Amazon, and Huawei, particularly for smartphone use and privacy-preserving user recommendations. FL operates on the concept of collaboration without aggregation, where a network of edge devices collaboratively trains a shared model iteratively, each device possessing a distinctive dataset. Each device performs local training on its data and generates a model update, which is then transferred to a central server for aggregation. The aggregated update is used to refine the global model, which is then distributed back to the devices for another round of local training. This iterative process continues until the model converges to an optimal state. The overview of FL can be seen in Figure 1.

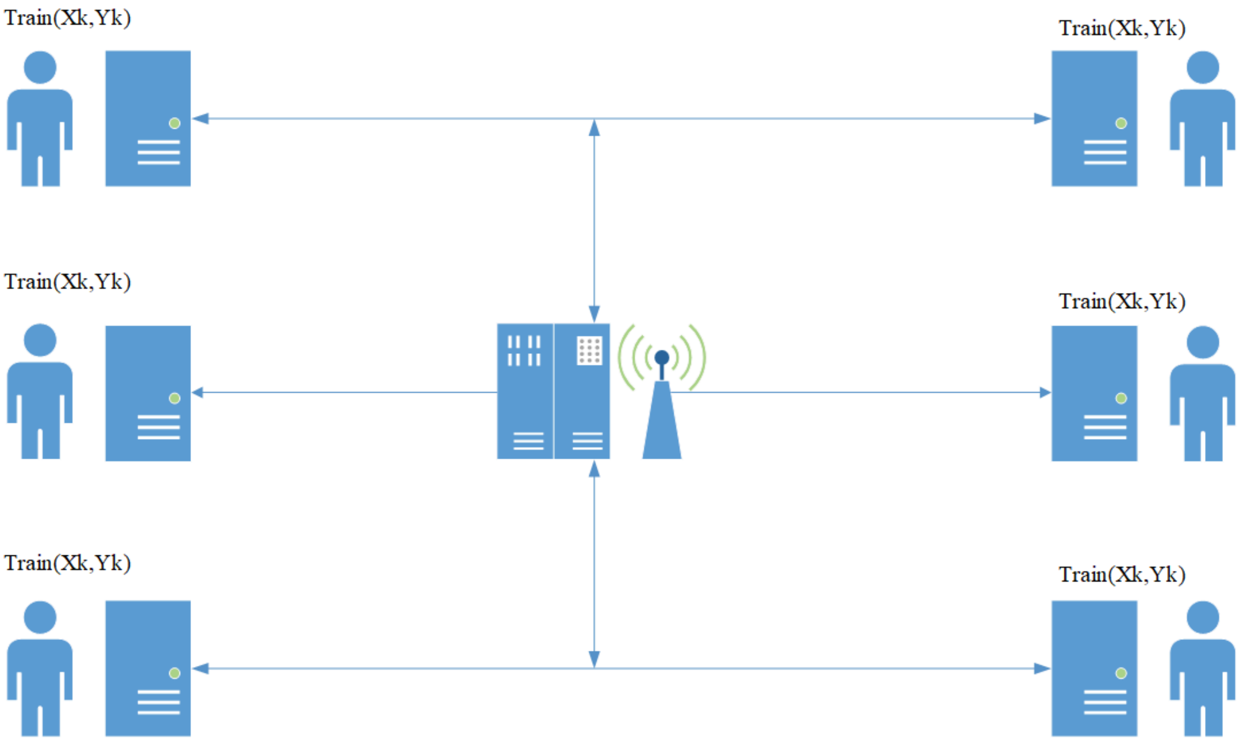

Figure 1: Federated Learning (FL)

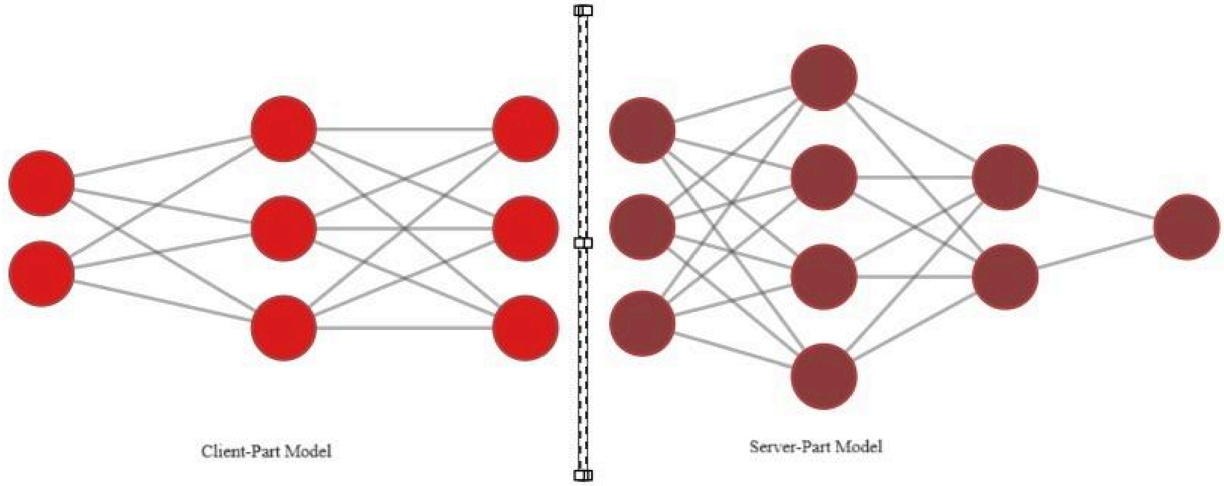

Figure 2: Split Learning (SL)

The concept of SL which is a part of FL, is a technology where a local client and a server work together to learn a global model in a decentralized data environment. In SL, the device handles the initial stages of a deep neural network, while the server manages the more complex stages. This partitioning of data and model ensures data privacy and enables efficient model training. The server hosts a global model, while individual devices retain relevant subsets of data. The local model on the device trains on its local data, and the global model on the server trains on the data it holds. The model parameters are then exchanged, with the device updating its local model with information from the server, and the server incorporating the device's learning progress into the global model. This iterative process, akin to knowledge sharing between a teacher and a student, results in a collaboratively trained model [3-5]. The overview of SL can be seen in Figure 2.

The selection between FL and SL is contingent upon the specific requirements of the learning task at hand. FL, due to its emphasis on privacy and communication efficiency, is a promising alternative for handling sensitive data and resource-constrained devices. On the other hand, SL can achieve faster convergence and leverage the server's data partition, making it suitable for scenarios where a global model is desired alongside local models. As these technologies continue to evolve, we can expect further refinements to address remaining challenges in FL and SL, such as communication efficiency, robustness to non-IID (Independent and identically distributed) data, and potential security vulnerabilities. The future of collaborative learning looks promising, as these approaches offer several benefits such as communication efficiency, diverse data, and model generalizability, while addressing privacy and security concerns associated with centralized methods.

The organization of this paper is as follows: Section II provides insights into the research methodology employed in studying FL and SL. Section III delves into the background of FL and SL, elucidating their descriptions, motivations, and architectures. Section IV outlines the applications of FL and SL. In Section V, the strengths and weaknesses of FL and SL are discussed. Section VI explores the Hybrid SL-FL approach. Section VII presents the HSFL architecture. Finally, Section VIII offers the conclusion, and Section IX discusses future prospects.

II. RESEARCH METHODOLOGY

The review paper's research methodology comprises several key sections, including the formulation of research questions, the search process for relevant literature, the establishment of inclusion and exclusion criteria, the methods for data collection and analysis, and the use of a PRISMA flow diagram to illustrate the review process.

A. Research Questions

This research investigates the effectiveness and efficiency of FL, SL, and their hybrid, Hybrid Split-Federated Learning (HSFL) model. The specific research questions addressed in this study are:

RQ1. What is FL? What are the steps involved in FL? What are the architecture and applications of FL?

RQ2. What is SL? What are the steps involved in SL? What is the architecture of SL?

RQ3. How does the performance of FL compare to SL in terms of model accuracy and computational efficiency?

RQ4. What are the limitations of standalone FL and SL models?

RQ5. How does Hybrid SL-FL(HSFL) solve the standalone problem of SL and FL?

RQ6. What is HSFL? What are the steps involved in HSFL? What is the architecture of HSFL?

B. Search Process

In order to conduct a comprehensive literature review, a systematic search was performed across various academic databases, such as IEEE Xplore, ACM Digital Library, SpringerLink, and Google Scholar. The search encompassed the following key terms:

- "Federated Learning"
- "Split Learning"
- "Hybrid Split-Federated Learning"
- "HSFL"
- "distributed machine learning"
- "privacy-preserving machine learning"

Boolean operators (AND, OR) were utilized to combine terms and enhance search precision. Furthermore, the reference lists of pertinent papers were manually examined to uncover any potential studies that might have been overlooked during the initial database search.

C. Inclusion and Exclusion Criteria

This review aims to give readers a thorough understanding of FL, SL, and HSFL, especially in their applications and comparative performance. The main focus is on how these learning methods are implemented and their effectiveness, particularly in real-world scenarios. The criteria for including and excluding relevant studies are as follows:

Inclusion Criteria:

Papers that:

- Focus primarily on FL, SL, or HSFL.
- Discuss the use of FL, SL, or HSFL in training machine learning models using real-world data, including medical data but not limited to it.
- Provide new insights, methodologies, or significant improvements related to FL, SL, or HSFL.
- Present case studies or applications of FL, SL, or HSFL in various domains, such as healthcare, finance, IoT, and others.
- Offer a comparative analysis of FL, SL, and HSFL in terms of performance metrics like accuracy, computational efficiency, and privacy.

Exclusion Criteria

Papers that:

- Require participating clients to share their private data, whether encrypted or not, as this contradicts the core privacy-preserving goals of FL, SL, and HSFL.
- Assume clients possess independent and identically distributed (IID) data, which is not a realistic setting for most real-world applications, particularly in healthcare where data heterogeneity is common.
- Focus solely on federated reinforcement learning, as this area, while related, employs significantly different methodologies and applications compared to supervised and unsupervised learning paradigms.
- Do not present novel ideas but merely describe the implementation of FL, SL, or HSFL in some applications, with the exception of detailed studies in the various sectors.
- Describe a fully decentralized implementation of FL (e.g., using Blockchain or Peer-to-Peer networks), as this diverges from the traditional client-server model typically associated with FL.
- Cover topics unrelated to FL, SL, or HSFL, i.e., papers that were mistakenly retrieved by the search query and do not contribute to the understanding of these learning paradigms.

These criteria ensure that the review remains focused on the most relevant and impactful studies, providing readers with a detailed and accurate understanding of FL, SL, and HSFL.

D. Data Collection and Analysis

Data was systematically extracted from each included paper and organized into a structured spreadsheet to provide a detailed quantitative and qualitative analysis of the reviewed literature. This process facilitated the synthesis of information and supported the subsequent analysis.

Data Extracted from Each Paper:

- Title and Year of Publication: To monitor the sequence and development of research in FL, SL, and HSFL.
- Data Extracted from Each Paper:
    - Title and Year of Publication: This helped us track the chronology and evolution of research in FL, SL, and HSFL.
    - FL, SL, or HSFL Training Algorithm.
    - Aggregation Techniques used in FL, SL, and HSFL.
    - Security or Privacy Protocols.
    - Communication Protocols.
    - Applications of FL, SL, and, HSFL.
    - Advanced Privacy Techniques.
    - Empirical Investigation.
- Machine Learning Models.
- Datasets.
- Research Question/Problem.
- Proposed Hypothesis or Solution.
- Results and Discussion.

This structured approach to data collection and analysis ensures a comprehensive understanding of the contributions, methodologies, and findings of each paper, allowing for a thorough evaluation of the state-of-the-art in FL, SL, and HSFL.

E. PRISMA Flow Diagram

In the study, the PRISMA flow diagram [109] was used to illustrate the process of searching, selecting, and excluding papers based on specific criteria outlined earlier. Out of the initial 267 papers, only 41 were included in the review as can be seen from Figure 3. Upon analyzing the included literature, several observations were made. Figure 4 displays the search engines used to find the papers and the number of papers included in the review. It's worth noting that the total number of papers is greater than 41 because some papers were found on multiple search engines. The majority of the papers were sourced from standard conferences and journal, a platform without direct peer-review, which is expected given that FL is a relatively new research topic.

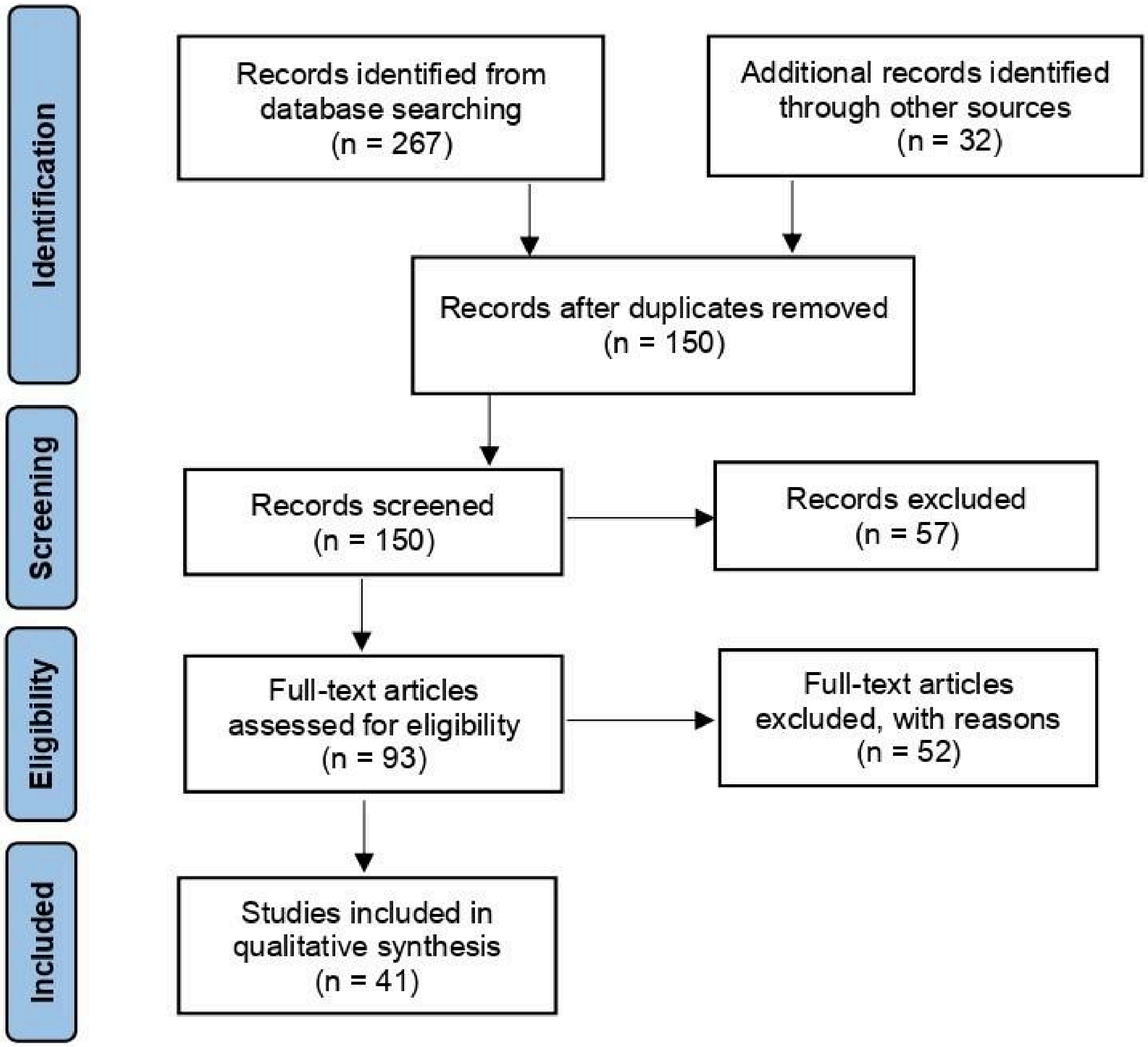
Identification
Screening
Eligibility
Included
Records identified from database searching (n = 267)
Additional records identified through other sources (n = 32)
Records after duplicates removed (n = 150)
Records screened (n = 150)
Records excluded (n = 57)
Full-text articles assessed for eligibility (n = 93)
Full-text articles excluded, with reasons (n = 52)
Studies included in qualitative synthesis (n = 41)

Figure 3: PRISMA Flow Diagram

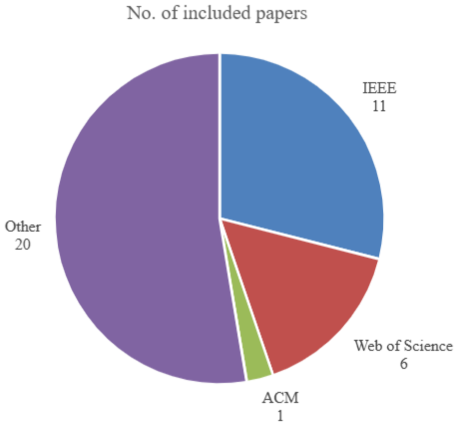

Figure 4: Sources used in this review paper

III. BACKGROUND: FL AND SL

This section offers a comprehensive analysis of the descriptions, motivations, and architectural intricacies of FL and SL within the context of batch processing. Batch processing involves executing data processing tasks in predefined groups or batches. This method is particularly relevant to FL and SL as it enables the efficient handling of extensive datasets and complex computations by processing them in discrete, manageable segments. In the realm of FL, batch processing plays a crucial role in facilitating the aggregation of model updates from multiple decentralized devices without overwhelming the central server. Devices conduct local model training on their respective data in batches and periodically transmit updates to a central server, where these updates are aggregated to improve the global model. This approach ensures that the training process remains scalable and efficient, even with a large number of devices. Similarly, in Split SL, batch processing helps manage the computational load by segmenting the model into multiple components. Data is processed in batches across these segments, with each segment potentially residing on different devices or servers. This segmentation allows for parallel processing and reduces the strain on any single device, thereby facilitating the handling of extensive datasets and complex models.

A. Description: Federated Learning and Split Learning

FL is a decentralized machine-learning technique that has gained significant attention in recent years. It involves training machine learning models on local devices or servers without exposing raw data. This approach addresses privacy concerns while harnessing the vast amounts of data available in the modern world. First introduced by Google in 2017 with the Gboard app, FL has since been applied to various use cases, including personalized recommendation systems, healthcare data analysis, and financial fraud detection. FL presents a promising approach to addressing data privacy concerns in the era of big data and machine learning by allowing data to remain local and private while benefiting from a global model. The primary goal of FL is to train a global model that incorporates insights from diverse data sources without compromising data privacy. This is achieved through a collaborative process where model updates are computed locally and periodically aggregated to refine the global model. FL revolutionizes traditional machine learning by distributing the training process across multiple devices or servers, allowing each to independently compute model updates using local data, which collectively contribute to the overall model. This decentralized model training approach enables large-scale model development while accounting for diverse data distribution patterns across devices or servers. FL prioritizes privacy preservation by localizing data and computations, minimizing the risk of sensitive information exposure, and ensuring user privacy throughout the training process [6]. It facilitates collaborative model aggregation by periodically combining locally computed updates from individual devices, refining the global model with insights from diverse data sources, and maintaining model consistency and accuracy [7]. This technique allows data owners to maintain control over their data while contributing to a robust global model, enhancing overall system performance. As an iterative learning process, FL uses local updates from edge devices to incrementally improve the global model, addressing privacy concerns inherent in centralized machine learning [9].

SL is also a cutting-edge approach to collaborative machine learning that offers unparalleled strategies for distributed model training while maintaining the data privacy of individual users [1]. In an extensive exploration of SL take a closer look at its definition and core concepts, the advantages and limitations it presents, the essential components that make up its architecture, practical implementations of this technology, real-world case studies that demonstrate its effectiveness, and the emerging research directions in this field. SL enables machine learning models to be trained on data that is distributed across multiple devices, allowing for decentralized data processing, which is particularly advantageous when working with sensitive data that cannot be shared with others. The model is divided into two distinct segments: the client-side and server-side components [10]. The client-side segment operates on local data, generating intermediate outputs that are transmitted to the server-side segment for further processing. This split enables collaborative learning without sharing raw data, thereby fostering privacy preservation and efficient model training across distributed environments.

Furthermore, SL enables efficient machine learning training across distributed environments, making it ideal for use cases where data is geographically dispersed. SL has numerous benefits, including increased privacy, improved scalability, and enhanced security. However, it is important to consider certain limitations when implementing this technology, such as the potential for data privacy leakage, high computation at clients, reduced efficiency due to a large number of clients, and poor scalability, especially with non-independent and identically distributed (Non-IID) data.

Overall, both, FL and SL are particularly useful in scenarios where data privacy is a primary concern, such as in healthcare, finance, etc as it ensures that data remains on local devices, reducing the risk of data breaches or leaks. The decentralized approach of FL also reduces network bandwidth requirements, making it more cost-effective than traditional centralized machine-learning approaches.

The initial implementation of FL took place on an Android smartphone using the Gboard application, a Google keyboard app for typing input. The Gboard app features text prediction, which anticipates the next word in a sentence based on the user's typing style. This personalized word prediction is achieved through the use of Recurrent Neural Network (RNN) models, particularly those employing Long Short-Term Memory (LSTM) [1]. As RNN models advanced, the decentralization of data became more prominent, requiring either the transmission of data to the model or the model to the data for personalized training. Natural data labeling is utilized, where the user clicks on the generated predictions are used as labeling inputs. The training process follows the FedAvg algorithm [1], involving the selection of active clients for synchronous FL rounds. The training devices are selected based on their idle time, availability, and connection status. The training process involves sending global model parameters to mobile devices, which then train the model on local data. After each round, the updated model parameters are sent back to the central server for aggregation and averaging. This process continues for a predefined number of rounds, after which the updated and averaged model is sent back to the mobile devices for performance evaluation.

ALGORITHM1:FederatedAveraging (FedAvg) [1]

```
Initialize $w^{0}_{global}$
for each round x = 1,2….
    $m \Leftarrow max(C \bullet K, 1)$
    $S_x \Leftarrow randomly\ selected\ m\ clients$
    for each client $k \in S_x$ in parallel do:
        $w^{x+1,k}_{local} \Leftarrow ClientUpdate\ (k, w^{x}_{global})$
    $w^{x+1}_{global} \Leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w^{x+1,\,k}_{global}$
Client Update ( $k,\ w_{global}$ ) $//\ Run\ on\ client\ k$
$\beta \Leftarrow (split\ P_k into\ batches\ of\ size\ \beta)$
for each local epoch $i$ from 1 to E do:
    for batch $b \in B$ do:
        $w_{local} \Leftarrow w_{local} - \eta\Delta\iota\ (w_{local}: b)$
Return $w_{local}$ to server
```

Algorithm 1 presents the pseudo-code for the FedAvg FL algorithm [1]. Here the global model parameters w_{global} are initialized before any training. For each round, a subset of clients S_x is randomly selected. The size of this subset is determined by $max(C \bullet K, 1)$, where C is the fraction of clients to be used and K is the total number of clients. For each selected client $k\ in\ S_x$, the local model parameters $w_{k,l}$are updated using the learning rate $\eta_{x,k,l}$ and the gradient of the loss function $g_{k,l}(w_{t-1})$. The average of local model parameters across all clients w_{avg} is computed. The global model parameters w_{global} are then updated with the average local model parameters. These steps are repeated for multiple rounds until convergence. The process involves training the model on local data while setting the number of epochs. Both the training and model aggregation occur in parallel, as illustrated in the algorithm. After each epoch, the client updates the model, sending the model parameters to the servers for global model updating. Each device follows the same process with its local data, sending updated model parameters to the central server after each round. The server averages the received parameters, as outlined in the algorithm, continuing until the predefined number of rounds is completed. Following the training rounds, the updated and averaged model is sent to the mobile devices for performance evaluation. There have been numerous revisions made to the FedAvg algorithm, covering improvements in accuracy, fairness, optimization for better performance, and more. In the domain of Federated Learning (FL), the FedAVg model aggregation technique stands as a fundamental approach. Nevertheless, this technique has undergone numerous refinements over time, as outlined in Table 1. In scenarios where the data displays diversity, the effectiveness of the FedAvg algorithm may be constrained. To address this issue and achieve broader data coverage, the FedProx algorithm has been devised, showcasing a significant 22% increase in test accuracy. FedProx exploits the notion of the proximal term to ensure improved convergence around local sub-problems, thereby limiting the impact of variable local updates. Before each update within the FedProx framework, the proximal term is integrated into the weights to promote convergence without introducing bias towards any specific client. For multi-distributed learning scenarios, the MOCHA framework introduces a multi-task learning framework that is well-suited for federated learning. It is noteworthy, however, that the current iteration of MOCHA does not extend to non-convex deep learning models. Two additional techniques that have been developed over time are FedWorse and FedBetter. The FedWorse technique involves considering the global model as the local model of the client with the worst performance, while the FedBetter technique involves considering the global model as the local model of the client with the best performance. In the context of applying federated learning to a large number of clients, challenges emerge due to the distributed nature of training data and the potential for unstable or slow network connections. Consequently, issues related to system, networking, and communication bottlenecks may arise. In response to these challenges, the Stochastic-AFL algorithm has been developed to optimize federated learning problems [8].

Table 1: Techniques of model aggregation in FL [8].

Algorithm	**Description**
FedAvg (Federated Averaging)	● Initialize global model parameters w_{global}. ● Repeat until convergence: For each client (i): Update local model parameters $w_{local}^{(i)}$ by minimizing local loss. ● Compute the average of local model parameters w_{avg}. ● Update global model parameters w_{global}.
FedProx (Proximal Federated Learning)	● Initialize global model parameters w_{global}. ● Repeat until convergence: For each client (i): Update local model parameters $w_{local}^{(i)}$ by minimizing proximal loss. ● Compute the average of local model parameters w_{avg}. ● Update global model parameters w_{global} .
MOCHA (Multi-task Learning Over Client Heterogeneity Algorithm)	● Initialize global model parameters w_{global}. ● Repeat until convergence: For each client (i): Update local model parameters $w_{local}^{(i)}$ by minimizing task-specific loss. ● Compute the average of local model parameters w_{avg}. ● Update global model parameters w_{global} .
Agnostic Federated Learning	● Initialize global model parameters w_{global}. ● Repeat until convergence: For each client (i): Update local model parameters $w_{local}^{(i)}$ by minimizing agnostic loss. ● Compute the average of local model parameters w_{avg} . ● Update global model parameters w_{global} .
FedWorse	Consider the global model as the local model of the client with the worst performance.
FedBetter	Consider the global model as the local model of the client with the best performance.

Minimizing communication overhead is also paramount To ensure the scalability and efficient performance of the system in FL, Communication protocols play a critical role in transmitting model updates between devices and servers, optimizing data transmission, and ensuring system security and reliability. Several communication protocols have been proposed for FL, including Message Queue Telemetry Transport (MQTT), Advanced Message Queuing Protocol (AMQP), ZeroMQ Message Transport Protocol (ZMTP), and socket implementations of TCP and UDP. These protocols are designed to efficiently distribute updates to participating devices while upholding security and reliability. Optimized application layer protocols such as AMQP, MQTT, and ZMTP outperform non-optimized protocols in most network conditions, resulting in a 2.5x reduction in communication time compared to TCP while maintaining accuracy. However, it is important to note that these communication protocols have limitations, and addressing these limitations and employing efficient techniques can further enhance FL's scalability, security, and reliability.

Due to the fact that SL is essentially an evolution of Federated FL, the model aggregation technique and communication protocols used in SL are identical to those used in FL.

B. Motivation and Key Features of FL and SL

FL has emerged as a promising approach to machine learning due to its ability to leverage data from diverse sources while safeguarding privacy and addressing scalability challenges associated with centralized machine learning approaches. The motivation behind FL is driven by the need to balance the benefits of data sharing with the need to protect sensitive information. Some of the key motivations and features of FL include privacy preservation, data sovereignty, scalability and efficiency, edge computing integration, and heterogeneous device support. One of the most significant motivations behind FL is privacy preservation, as it ensures that raw data remains localized on user devices or edge servers, thereby reducing the risk of data breaches or unauthorized access. This approach helps to keep sensitive information under the control of data owners, critical in today's world where data privacy is a growing concern. Another important motivation behind FL is data sovereignty, which empowers data owners to retain control over their data and dictate how it is utilized for model training, ensuring compliance with data regulations and enhancing trust between data owners and service providers [1-2]. Scalability and efficiency are also significant motivations behind FL, as it enables scalable and efficient model training by distributing computation and training tasks across distributed devices or servers, accommodating variations in computational capabilities, network bandwidths, and data distributions across heterogeneous devices, making it suitable for deployment in diverse environments, including IoT networks, mobile devices, and edge servers [3-4]. In summary, the motivation behind FL lies in its ability to address privacy concerns, enhance data sovereignty, improve scalability and efficiency, enable edge computing integration, and support heterogeneous device environments. These features make FL a promising approach to machine learning that can be deployed in a wide range of applications, from healthcare to financial services and beyond.

SL is a promising approach that offers several advantages over traditional centralized machine learning methods. One of the most significant benefits of SL is its ability to mitigate privacy concerns associated with centralized approaches. By processing data locally, SL ensures that sensitive user data remains on-device, making it more challenging for third-party entities to access or exploit it [11]. Additionally, the separation of model segments reduces communication overhead, enabling efficient collaboration even in bandwidth-constrained settings. However, as with any distributed machine learning approach, SL presents some challenges that need to be addressed to ensure its successful implementation. One of the most significant challenges is the need for robust synchronization mechanisms between client and server components to ensure model consistency across distributed segments [12]. The coordination and synchronization strategies must be carefully designed to maintain accuracy and convergence while minimizing communication overhead. Despite its limitations, SL is a promising approach that has the potential to revolutionize the way machine learning is performed by preserving privacy, improving communication efficiency, and ensuring scalability. Future research should focus on developing innovative synchronization mechanisms that can overcome the current limitations and allow for more widespread adoption of SL in real-world scenarios.

C. Architectures and Frameworks for Batch FL and SL

Both FL and SL involve communication between devices and the server, albeit FL generally incurs lower communication overhead than SL. This discrepancy arises from the exchange of only model updates in FL, while in SL, model parameter exchange can be more substantial, particularly in cases of complex model architectures. Techniques such as model compression can mitigate communication costs in both FL and SL. Security and privacy are paramount concerns in any system or process, necessitating the implementation of appropriate measures and safeguards to protect sensitive data and information. Encryption techniques are pivotal in both FL and SL to secure model updates and parameters during exchange. Additionally, the incorporation of differential privacy can further enhance privacy by introducing noise to model updates, rendering it challenging to infer individual data points from aggregated updates. Moreover, scalability is a critical consideration. Both FL and SL are well-suited for large-scale deployments involving numerous participating devices. The capacity to distribute training across devices makes them efficient for scenarios where centralized training may be impractical due to computational resource limitations. Batch processing, in which a subset of devices (batch) is selected for each training round, is a feasible implementation for both FL and SL.

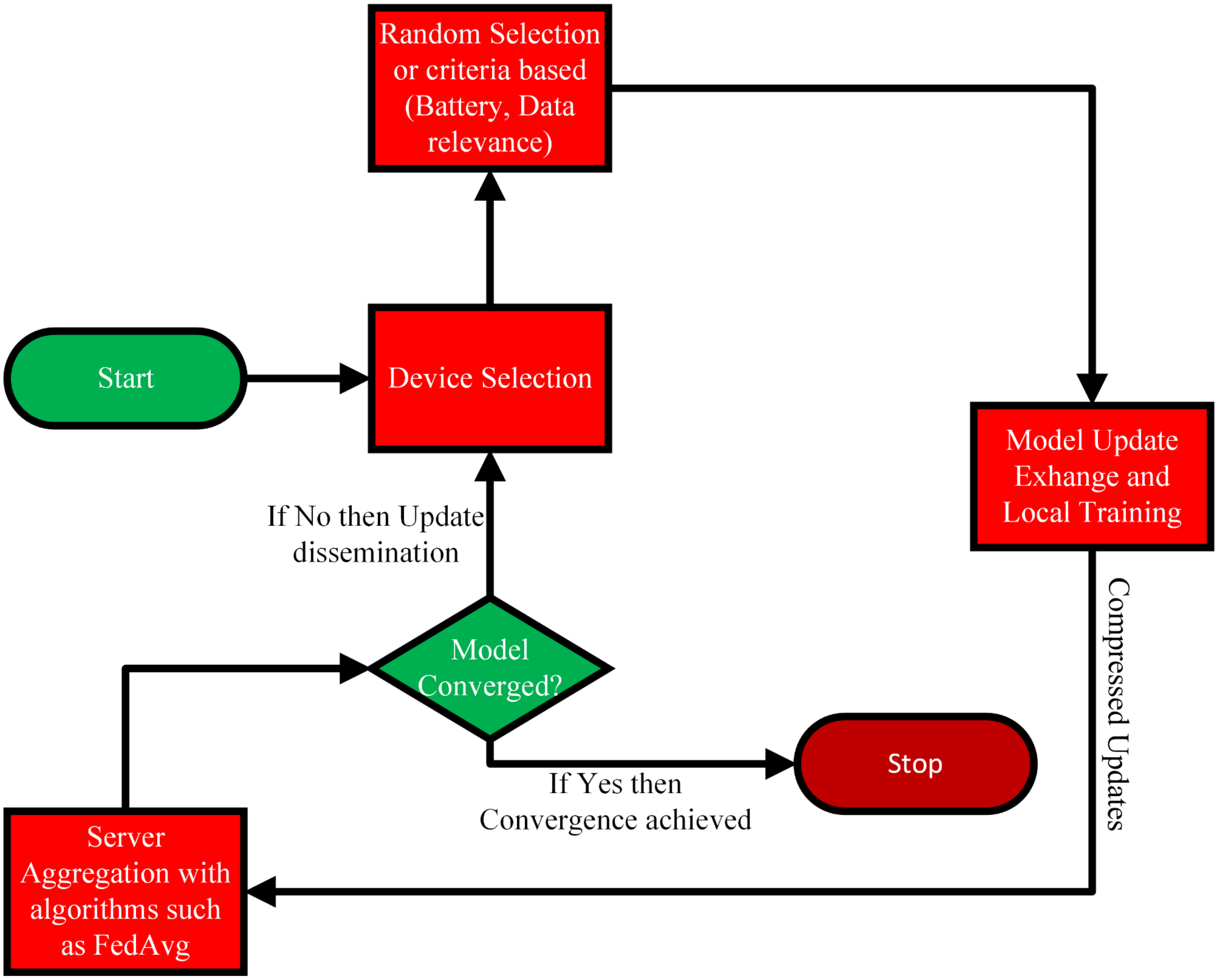
Random Selection or criteria based (Battery, Data relevance)
Start
Device Selection
Model Update Exhange and Local Training
If No then Update dissemination
Model Converged?
Compressed Updates
Stop
If Yes then Convergence achieved
Server Aggregation with algorithms such as FedAvg

Figure 5: Basic Architecture of FL

Figure 1 below depicts the fundamental architecture of Federated Learning (FL). It comprises five main components: Device Selection, Local Training and Model Update Exchange, Server Aggregation, and Global Model Update. In the training process, selecting a subset of devices for each round is crucial for ensuring the accuracy and effectiveness of the model. This selection can be done in various ways, such as random selection or based on specific criteria like device availability, battery level, data relevance, or a combination of these factors [1]. This approach allows for a more diverse and representative set of data to be used in the training process, leading to improved performance and a more robust model. The chosen devices conduct localized training on their respective data partitions, using the most up-to-date version of the global model transmitted from the server. This local training process is optimized to utilize the computational resources available on the device itself. After completing the training at the local level, the devices send their lightweight model updates to the server. To reduce communication overhead, these updates can be compressed using methods such as Weight Clustering, Knowledge Distillation, etc [3]. In the context of FL, aggregating model updates from all participating devices in a batch is a crucial step. Various techniques are employed for this purpose, which may vary depending on the specific FL implementation. These techniques include averaging, as well as federated averaging with momentum, among others. FL is a process that improves the global model by receiving updates from local training on all participating devices. The convergence of the global model is typically determined by the stability of its performance across successive iterations. Determining if a model has converged in FL involves monitoring the loss curve and performance curve. The criteria for convergence and the decision of whether a model has converged can depend on the specific requirements and constraints (i.e., accuracy) of the FL system being used. This iterative process ensures convergence of the global model, enhancing its accuracy and generalizability.

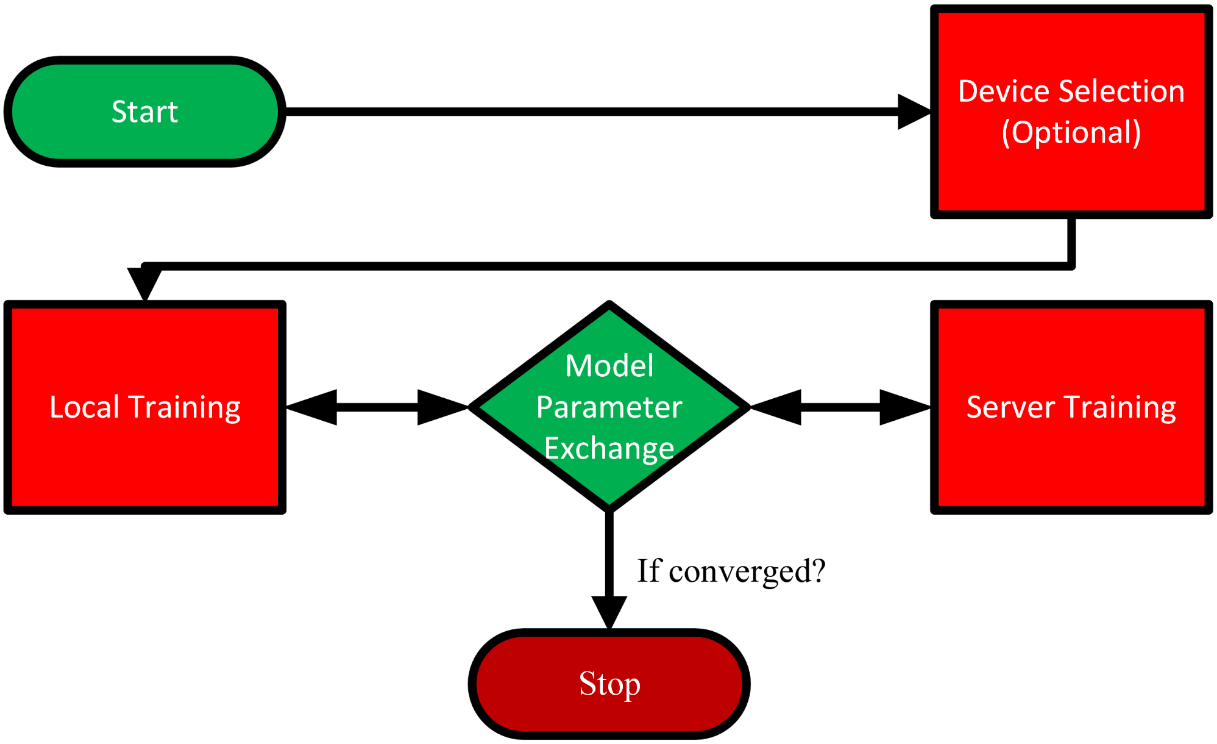

Figure 6: Basic Architecture of SL

Figure 2 depicts the fundamental structure of SL, encompassing optional Device Selection, Local Training, Server Training, and Activation Communication. Each of these elements is further explained below. In the context of SL, it is beneficial yet optional to select a subset of devices for each training round. This selection process can be influenced by various factors, such as data relevance or device computational capacity. By strategically choosing a subset of devices for each round of training, the efficiency of the learning process can be enhanced, ultimately leading to improved results. During the SL process, both the server and devices engage in training simultaneously in each round. The server trains the global model on its own data partition, while devices train on their local models using the most recent activations received from the server. This approach enables the global model to be updated with the latest information from all participating devices while maintaining the privacy of their local data partitions. Instead of exchanging model parameters, devices, and servers communicate specific activations to ensure learning occurs without compromising privacy or efficiency. As part of the communication process, devices utilize the latest activations to conduct forward propagation on their local models. The loss is then calculated based on these activations locally, without exposing any sensitive information to the server. Local execution of backpropagation is carried out to update the model parameters using gradients computed from the loss function. This ensures that the model of each device gradually improves while safeguarding privacy. The process of forward and backpropagation is repeated until the model reaches the optimal solution.

IV. APPLICATIONS OF FL AND SL

FL and SL have gained significant attention in fields such as IoT, healthcare, vehicle IoT, intrusion detection, cybersecurity, robotics, computer vision, NLP, and blockchain. Blockchain technology ensures security and transparency in AI and machine learning frameworks [13]. AI, machine learning, and deep learning are crucial in technologies like self-driving cars, which use AI to detect obstacles and predict traffic [14]. Federated learning-based AI approaches in smart healthcare leverage machine learning, computer vision, IoT, and blockchain technologies [15]. Federated learning also secures IoMT applications in healthcare, integrating medical equipment and devices [16]. In cybersecurity, deep learning techniques are employed for real-time anomaly detection and security attacks in IoT systems [17]. Traditional machine-learning models for IoT malware detection are resource-intensive, necessitating more efficient approaches. Federated learning is a solution for secure communication in 6G networks, with blockchain empowering distributed federated learning frameworks for health data security [18]. Securing

AI-based systems in healthcare using blockchain is crucial, especially for data from NLP, computer vision, and acoustic AI applications. Blockchain enhances trust in AI systems by establishing the provenance of machine learning models, leading to more reliable and secure healthcare solutions [19]. Overall, the integration of federated learning, blockchain technology, and deep learning has the potential to revolutionize industries including IoT, healthcare, cybersecurity, and robotics.

FL and SL have gained significant attention in various fields such as IoT, healthcare, vehicle IoT, intrusion detection, cybersecurity, robotics, computer vision, NLP, and blockchain. In the realm of deep learning, blockchain technology has been utilized to ensure security and transparency in AI and machine learning frameworks [13]. Artificial intelligence, machine learning, and deep learning have been instrumental in the development of technologies such as self-driving cars, which rely on AI to detect obstacles and predict traffic movements [14]. In the context of smart healthcare, federated learning-based AI approaches have been explored, leveraging machine learning, computer vision, IoT, and blockchain technologies [15]. Additionally, federated learning has been applied to secure IoMT applications in smart healthcare, where IoT devices play a crucial role in medical equipment and device integration [16]. In the domain of cybersecurity, the detection of real-time malicious intrusions and attacks in IoT systems has been a key area of focus, with deep learning techniques being employed for anomaly detection and security attacks [17]. Traditional machine-learning models for detecting IoT malware are resource-intensive and computationally demanding, highlighting the need for more efficient approaches in IoT security. Furthermore, federated learning has been implemented as a solution for secure communication systems in the context of 6G networks, with blockchain technology empowering distributed federated learning frameworks for health data security [18]. In the healthcare sector, securing AI-based systems using blockchain technology has been identified as a critical need, particularly in the processing of data from NLP, computer vision, and acoustic AI applications. By establishing the provenance of machine learning models through blockchain technology, trust in AI systems can be enhanced, leading to more reliable and secure healthcare solutions [19]. Overall, the integration of federated learning, blockchain technology, and deep learning has the potential to revolutionize various industries, including IoT, healthcare, cybersecurity, and robotics.

A. IOT

The Internet of Things (IoT) generates a lot of sensitive data, which is difficult to centralize and process. FL and SL have emerged as solutions that allow for collaborative machine learning without centralizing data, thus preserving user privacy and reducing communication and storage costs. Edge Computing provides a platform for high-performance IoT applications but poses challenges such as data privacy, resource constraints, and system scalability. Collaborative Learning, where FL and SL are explored as strategies to address these challenges by training global models collaboratively without centralizing data. However, implementing FL and SL in IoT also poses several challenges, such as limited on-device resources, network bandwidth, intermittent connectivity, system heterogeneity, temporal dynamics, trustworthiness, and the need for standardization and development tools. Addressing these challenges presents opportunities to improve the efficiency and effectiveness of FL and SL in IoT applications.

B. Healthcare

Healthcare organizations use privacy-preserving distributed learning techniques to develop models without sharing raw patient data. Two techniques are FL and SL. FL trains a global model on a central server using locally maintained data. SL splits the model into two parts - one on the user's device and the other on a central server. Both techniques address data governance and privacy issues. However, they face ethical and technical challenges.

C. Internet of Vehicles

FL and SL have various applications in the Internet of Vehicles (IoV), especially in the fields of cooperative driving and collaborative robotics. By allowing local models to be trained on separate vehicles before aggregating them in the cloud, FL and SL can enhance security, accuracy, and learning efficiency in IoV. Additionally, FL is presented as a solution to privacy concerns in IoV, enabling distributed learning on edge devices while keeping private data local. Various systems implementing FL for IoV have been reviewed, and different ML models like neural networks, decision trees, and linear models have been discussed within the federated environment.

D. Smart Intrusion Detection Systems

Smart Intrusion Detection Systems are increasingly utilizing FL and SL techniques to ensure security and privacy. By facilitating collaborative model training across decentralized devices, FL and SL enable anomaly detection without the need to share raw data. The integration of FL and SL with Intrusion Detection Systems (IDS) enhances privacy and security while maintaining high classification accuracy, making it an effective solution for the growing number of connected devices, particularly in smart cities. However, implementing FL and SL in IDS poses challenges such as privacy concerns, data management, and model training, which can be overcome through robustness in anomaly detection and further research in communication-efficient federated IDS, handling non-IID data, and model heterogeneity. The implemented federated-based IDS model is evaluated through client-side performance, where clients assess the global model locally and send results back to the server, achieving a high detection rate. The innovative approach of balancing IoT network security and data privacy through FL and SL offers a scalable and adaptable solution for smart city cybersecurity in the digital transformation of urban environments.

E. Cybersecurity

FL and SL are gaining popularity in the field of cybersecurity due to their ability to overcome challenges related to data silos and privacy concerns. These approaches have practical applications in cybersecurity, including detecting attacks and identifying vulnerabilities. However, privacy threats, communication overhead, data heterogeneity, and security threats are some of the challenges that need to be addressed. Researchers are actively exploring ways to mitigate these threats at each phase of execution, including data and behavior auditing, training, and predicting, by leveraging techniques such as blockchain and secure multi-party computing to enhance security and privacy. By adopting these approaches, the cybersecurity industry can achieve better protection against potential threats while preserving the privacy and security of sensitive data.

F. Robotics and Automation

FL and SL are two distributed machine-learning approaches that have found applications in robotics and automation. They are used to provide tailored recommendations based on historical data and to enhance autonomous and networked industrial systems, such as robots, vehicles, and drones. These approaches enable model training across multiple devices without sharing raw data, preserving privacy, and complying with regulations. In the context of robotics and automation, FedSL has been implemented as a good FL-SL method tailored for RNNs, outperforming previous FL and centralized learning approaches in accuracy and communication efficiency.

G. Computer Vision

The use of FL and SL in Computer Vision (CV) is gaining interest. However, implementing FL and SL in CV has challenges such as dealing with non-IID data, ensuring data privacy, and addressing data scarcity. To address these challenges, the FeSViBS framework introduces a new federated split learning framework for medical imaging classification. It uses Vision Transformers (ViTs) and a block sampling module to improve model generalization. The framework also uses a shared projection network to improve feature augmentation and model robustness. It outperforms other SL and FL methods on three medical imaging datasets, under both IID and non-IID settings. FL and SL techniques have been applied to various CV tasks such as image classification, sequence tagging, question answering, and seq2seq generation.

H. Natural Language Processing (NLP)

FL and SL are increasingly being used in Natural Language Processing (NLP) tasks such as text classification, sequence tagging, question answering, and seq2seq generation. These techniques allow for the development of models without compromising data privacy, thus ensuring compliance with regulations and preserving privacy. The adoption of FL in NLP has been driven by the need to protect sensitive data during machine learning training. It addresses privacy and data breach concerns by training models on user devices, and keeping sensitive data local. FL is also useful in various NLP applications such as fraud detection and medical data analysis. Its viability for privacy-preserving NLP model development has been proven in experiments that show comparable accuracy with centralized approaches. However, the survey identifies algorithmic, systemic, and privacy challenges in applying FL to NLP tasks, highlighting the need for large-scale pre-trained language models and advanced privacy preservation techniques.

I. Blockchain Technology

Blockchain technology combined with Federated-Split Deep Learning can enhance the security and efficiency of electronic health records (EHR) in healthcare. The implemented Federated Learning Framework uses Hyperledger Fabric for blockchain-based storage and LightGBM and N-Gram models for a collaborative learning module to provide tailored treatment recommendations by analyzing EHR. The implemented framework emphasizes secure and private EHR storage to protect sensitive patient information from unauthorized access and cyber-attacks. The implemented BCFL-based framework has the potential to revolutionize the healthcare industry by providing a secure and efficient way of analyzing and utilizing EHR data without compromising patient privacy.

V. STRENGTHS AND WEAKNESSES OF FL AND SL

Collaborative machine learning is a rapidly evolving field that encapsulates two distinct paradigms - FL and SL, each offering unique approaches and characteristics. In this section, we aim to conduct a comprehensive comparative analysis of both FL and SL, delving into their key features, and evaluating their respective strengths and weaknesses across various domains and use cases. Furthermore, we endeavor to introduce a Hybrid FL and SL approach that leverages the advantages of both paradigms, while mitigating their respective disadvantages. Our discussion is intended to provide a deeper understanding of the theoretical underpinnings of these paradigms, as well as their practical implications and limitations.

FL and SL are two distinct approaches to collaborative machine learning that differ fundamentally in their approach. FL emphasizes decentralized model training, where the model is distributed across multiple devices or nodes, and local updates are aggregated to generate a global model [40]. In contrast, SL focuses on splitting the model into segments, with client-side processing performed locally and server-side processing aggregating client outputs [39]. FL leverages the heterogeneity of distributed data sources for model training, allowing each client to contribute unique insights without sharing raw data. This approach is particularly useful in scenarios with a large number of participating clients, such as mobile devices, IoT devices, or edge servers, facilitating collaborative learning across diverse data sources [1]. FL operates on a sequence of decentralized optimization rounds, where each client trains the model locally on its data and sends the updated model parameters to the server. The server then aggregates the updated model parameters to generate a new global model, which is then sent back to the clients for the next iteration [20]. On the other hand, SL emphasizes privacy preservation by keeping sensitive data on-device, minimizing raw data transmission across networks [21]. This approach is particularly suited for applications where data privacy is paramount, such as healthcare, finance, and telecommunications, where sensitive information must be protected [22]. SL operates by splitting the model into two parts: a public part that is shared with the server and a private part that is kept on the client device. The client device performs local computations on the private part of the model and sends the output to the server, which aggregates the outputs to update the public part of the model. Both FL and SL have their unique strengths and limitations, and the choice of approach depends on the specific application requirements. While FL is better suited for scenarios with a large number of clients, SL is more appropriate for situations where data privacy is paramount. Further research is needed to optimize these approaches for different applications and to identify the best practices for collaborative machine learning.

FL is a novel approach to machine learning that has gained significant attention in recent years. One of the key strengths of FL is its scalability, robustness, and adaptability to heterogeneous data sources. By aggregating local updates from distributed clients, FL enables collaborative learning across diverse environments, facilitating efficient model training without the need for centralized data storage [23]. However, despite its many benefits, FL also presents several challenges, including communication overhead, synchronization issues, and privacy concerns. These challenges become particularly pronounced in scenarios with bandwidth constraints or stringent privacy requirements [24].

In contrast, SL offers several advantages, including privacy preservation, communication efficiency, and model interpretability. By processing data locally and transmitting only intermediate outputs, SL minimizes data exposure and communication overhead, making it well-suited for privacy-sensitive applications [25]. Additionally, SL enables transparent model interpretation, as each segment of the model can be analyzed independently, facilitating debugging and model optimization

[26]. However, SL may encounter challenges related to model synchronization, scalability, and resource constraints, particularly in scenarios with large model sizes or distributed architectures [27].

On the contrary, FL and SL are two distinct machine-learning approaches that offer unique advantages and disadvantages. While FL excels in scalability, robustness, and adaptability to diverse data sources, it poses challenges in communication overhead, synchronization, and privacy preservation. Conversely, SL offers privacy preservation, communication efficiency, and model interpretability, but may face challenges in scalability, model synchronization, and resource constraints. By carefully considering the strengths and limitations of these two approaches, machine learning practitioners can choose the most appropriate approach for their specific use case.

VI. HYBRID SPLIT - FEDERATED LEARNING (HSFL)

FL and SL are two promising techniques that have emerged in recent years in the field of machine learning. FL is a decentralized approach that enables multiple parties to collaboratively train a model while keeping their training data locally, which has several advantages such as data privacy, reduced communication costs, and improved scalability. On the other hand, SL allows for model split and training, which can be beneficial in certain scenarios where data cannot be shared or when the data is too large to be transmitted. As shown by Thapa et al. [28] SL could achieve comparable accuracy to FL while reducing the communication overhead significantly. This approach is gaining traction in various applications, such as healthcare, where privacy and security are of utmost importance.

In the healthcare industry, FL has been used for predicting patient outcomes and detecting diseases, while SL has demonstrated its usefulness in edge computing scenarios, where devices have limited computational resources and cannot transmit large amounts of data. Both techniques have unique advantages and applications that make them suitable for specific scenarios. FL has been applied to fraud detection and credit risk analysis in finance. FL has also been used in Industry 4.0 for predictive maintenance of machinery and in smart vehicles to improve autonomous driving capabilities. These applications show the vast potential of FL and SL in various fields.

To further enhance the benefits of both FL and SL, a hybrid approach called SplitFed Learning has been introduced, aiming to combine the strengths of both techniques Thapa et al. [28]. This hybrid model, as discussed by Turina et al. [29], can reduce the computational power required for each client running FL and enable parallelization in SL, all while maintaining high prediction accuracy, especially with unbalanced datasets during training. Overall, SplitFed Learning has the potential to improve the efficiency and effectiveness of distributed machine learning systems [28-29]. In the context of scalable SL, Pal et al. [30] highlight the advantages of server-side local gradient averaging and learning rate acceleration (as described in their paper "Scalable SL for Federated Edge Intelligence"), which can lead to a greater reduction in the leakage of sensitive information compared to baselines. This emphasizes the importance of privacy preservation in distributed learning architectures. Their work builds upon previous research in the field of distributed machine learning, including the work of Bonawitz et al. [5] on FL and the work of McMahan et al. [1] on communication-efficient learning. In recent years, there has been a growing interest in exploring the potential of combining FL and SL in wireless unmanned aerial vehicle (UAV) networks. One notable study in this area was conducted by Liu et al. [31], who gave a hybrid split and FL framework that allows users to choose between split training or federated training based on their respective characteristics. This approach considers the diverse nature of users with varying resources and data distributions and offers a flexible solution that can be tailored to the needs of each user. Other studies in this field include the work of Wang et al. [32] and Zhang et al. [33], who both investigated the potential of FL in UAV networks. Wang et al. [32] implemented a FL framework that can be used to support collaborative training of deep neural networks across multiple UAVs, while Zhang et al. [33] demonstrated the feasibility of using FL to train UAVs in a distributed manner, without the need for a centralized server. Overall, these studies highlight the potential of FL and SL in wireless UAV networks and suggest that these techniques could play an important role in enabling more efficient and flexible training of machine learning models in this context.

Challenges arise when implementing FL or supervised learning in device-to-device (D2D)-enabled heterogeneous networks, as discussed by Cheng et al. [34] in their paper titled " FL: Challenges, Methods, and Future Directions ". To address these challenges, hybrid distributed machine learning (ML) architectures like hybrid split FL (HSFL) and hybrid federated SL (HSFL) have been introduced, aiming to improve scalability and reduce training delays. This was mentioned in a recent research paper titled "A Hybrid Distributed Machine Learning Architecture for Device-to-Device-Enabled Heterogeneous Networks" by Zhang et al. [33]. In the realm of edge intelligence algorithms, Yin et al. [35] proposes a hybrid federated SL framework in wireless networks, combining the advantages of FL and SL to reduce communication burden and increase computing efficiency. The implemented framework is based on the assumption that the edge nodes can perform both SL and FL operations, and it consists of two phases: a split phase and a federated phase. In the split phase, the edge nodes perform SL to extract features from the data, and in the federated phase, the features are aggregated and used for training a global model. The implemented framework is evaluated using a real-world dataset, and the results show that it outperforms existing FL and SL algorithms in terms of communication efficiency and accuracy. This hybrid approach could potentially be useful in a wide range of applications, such as smart cities, industrial IoT, and healthcare. Similarly, Yang et al. [36] introduced Robust Split FL (RoS-FL) for U-shaped medical image networks. RoS-FL aims to balance computational cost, model privacy, and parallel training effectively. The authors implemented a novel architecture that leverages the characteristics of U-shaped networks to improve the robustness of FL. The implemented method was evaluated on a public medical imaging dataset, and the results showed that RoS-FL outperformed existing FL methods in terms of accuracy and convergence rate. Overall, the study highlights the potential of RoS-FL for improving the efficiency and privacy of FL in medical image analysis. According to Zhang et al. [29], significant efforts have been made to enhance the privacy and efficiency of communications in federated-split-learning architectures. This highlights the importance of finding ways to optimize the exchange of information in distributed machine-learning systems, especially when dealing with sensitive data. The authors discuss several methods for improving the performance of federated-split-learning, including the use of compression techniques, secure aggregation, and differential privacy mechanisms. These techniques can help ensure that data is kept private and secure while also allowing for faster and more efficient communication between different nodes in the network. Examining the trade-offs between different distributed learning approaches, a new hybrid Federated SL architecture has been suggested to maximize efficiency and privacy benefits Zhou et al. [32]. Overall, the literature indicates a growing interest in hybrid approaches that combine the strengths of FL and SL to address various challenges in distributed machine learning applications Verbraeken et al. [37] and Zhou et al. [32]. These hybrid models offer promising solutions for improving efficiency, privacy preservation, and scalability in diverse network settings [31 - 37].

VII. HSFL ARCHITECTURE

Here's a simplified working diagram HSFL as shown by the Figure 7:

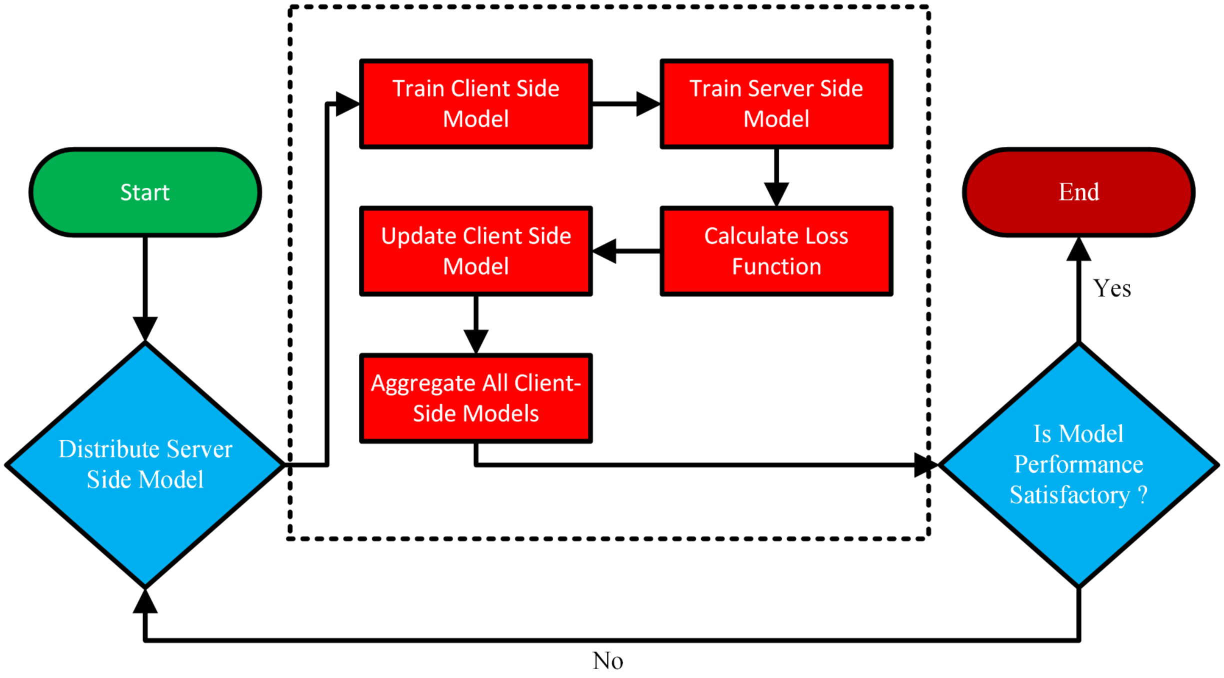

Figure 7: Basic Architecture of HSFL

Hybrid SL-FL (HSFL) is a distributed learning architecture that combines the benefits of both FL and SL in terms of efficiency and privacy [38]. The architecture allows for decentralized model training while maintaining data privacy, as raw data does not leave individual clients. It also reduces the computational power required for each client running FL and enables SL parallelization while maintaining a high prediction accuracy with unbalanced datasets during training. Furthermore, HSFL provides a better accuracy-privacy tradeoff in specific privacy approaches than Parallel SL.

ALGORITHM 2: Cluster-HSFL Workflow

```
Server Side:
  1: Initialize the server-side model
  2: for each round t = 1,2,3,4….:
      ●  Distribute the server-side model to each user cluster.
      ●  For each user in each cluster, train the client-side model with user data.
      ●  Aggregate the trained client-side models at the edge server.
      ●  Train the server-side model with the aggregated data.
      ●  Calculate the loss function to evaluate model performance.
      ●  For each user in each cluster, update the client-side model based on the loss function.
      ●  Aggregate the updated client-side models at the edge server.
  3: Return updated server-side model

ClientUpdate (user, server-side model): // Run on each user
  1: Split the user data into batches of size β
  2: For each local epoch $i$ from 1 to $E$:
      ●  For each batch $b \in B$:
            ○  Update the local model parameters $w_{local} \Leftarrow w_{local}$ - ηΔι $w_{local}$ ; b).
      ●  Return $w_{local}$ to the server.
```

Figure 7 and Algorithm 2 represents a flowchart for a machine learning model training and deployment process of HSFL. It shows the iterative and parallel processes involved in training and deploying a machine learning model, with checks for convergence to ensure model stability before deployment. Algorithm 1 represents the HSFL workflow, where the server-side model is distributed to different user clusters. Each user within these clusters trains their respective client-side models with their own locally available data. These locally trained client-side models are then sent to an edge server where they contribute to training a unified server-side model. At this stage, a loss function is calculated to evaluate the performance metrics of the trained model. The feedback from this evaluation is subsequently sent back to all user clusters, leading to updates and improvements in each individual's client-side models. Finally, all improved client-side models are aggregated at the edge server for a comprehensive update. This iterative process continues until the model's performance is deemed satisfactory. The final model is a result of aggregating all the client-side models. This approach enables the efficient utilization of decentralized data and preserves the privacy of user data. It also provides a distributed learning framework that combines the benefits of both split learning and federated learning. By distributing the server-side model to different user clusters, the algorithm reduces communication costs and addresses the issue of unbalanced data distribution. Moreover, the iterative nature of the process allows for continuous improvement of the model, leading to better performance and increased accuracy.

VIII. CONCLUSION

This review paper comprehensively examines FL and SL, focusing on their architectures and aggregation methods. We delve into the structural nuances of both FL and SL, highlighting how these approaches enable decentralized and privacy-preserving machine learning. The paper elucidates the mechanisms of model training, data handling, and the iterative processes that define each method. Various architectural models within FL and SL are discussed, demonstrating their applicability in diverse scenarios and their impact on communication efficiency and computational load. Additionally, we explore the different aggregation strategies employed in these learning frameworks, emphasizing their roles in enhancing model accuracy and convergence speed. The review also covers the integration of FL and SL into hybrid frameworks, showcasing their combined strengths in addressing the limitations inherent in standalone implementations. By providing a thorough overview of current methodologies, challenges, and advancements, this paper aims to serve as a valuable resource for researchers and practitioners seeking to understand and leverage these powerful machine-learning techniques. In conclusion, this paper presents a detailed synthesis of the state-of-the-art in Federated Learning and Split Learning, offering insights into their architectural designs, aggregation techniques, and practical implementations. Through this comprehensive review, we hope to facilitate further research and development in decentralized and privacy-preserving machine learning.

IX. REFERENCES

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Aguera y Arcas. "Communication-efficient learning of deep networks from decentralized data." in Proceedings of the Twentieth International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, USA, 2017, vol. 54, pp. 1-11.

[2] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," in *ACM Transactions on Intelligent Systems and Technology,* vol. 10, no. 2, pp. 1–19, 2019.

[3] Y. Liu, J. Peng, J. Kang, A. M. Iliyasu, D. Niyato, and A. A. A. El-Latif, "A secure federated learning framework for 5G networks," *IEEE Wireless Communications*, vol. 27, no. 4, pp. 24-31, 2020.

[4] F. Haddadpour, M. M. Kamani, A. Mokhtari, and M. Mahdavi, "Federated learning with compression: Unified analysis and sharp guarantees," in Proceedings of the 24th International Conference on Artificial Intelligence and Statistics, San Diego, California, USA, 2021, vol. 130, pp. 2350-2358.

[5] M. S. H. Abad, E. Ozfatura, D. Gunduz, and O. Ercetin, "Hierarchical federated learning across heterogeneous cellular networks," in IEEE International Conference on Acoustics, Speech and Signal Processing, Barcelona, Spain, 2020, pp. 8866-8870.

[6] B. Jones, "Financial fraud detection using FL: Opportunities and challenges," in Proceedings of the International Conference on Machine Learning, Long Beach, California, vol. 25, 2019, pp. 1117-1651.

[7] C. Wang, "FL for wireless communications: A survey," in I*EEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2861–2895, 2021.

[8] M. P. Sah, and A. Singh, "Aggregation techniques in federated learning: Comprehensive survey, challenges, and opportunities," in Proceedings of International Conference on Advance Computing and Innovative Technologies in Engineering, Greater Noida, India, 2022, pp. 1962–1967.

[9] M. A. B. Syed, Q. Rhaman, and S. Sushil, "Federated learning in manufacturing: A systematic review and pathway to industry 5.0," in 5th International Conference on Sustainable Technologies for Industry 5.0, Dhaka, Bangladesh, 2023, pp. 1-6.

[10] J. Smith and A. Johnson, "Split Learning: A decentralized approach to collaborative machine learning," in J*ournal of Artificial Intelligence Research*, vol. 45, pp. 567–589, 2020.

[11] S. Lee, H. Kim, and C. Park, "Privacy-preserving machine learning via split learning," in *IEEE Access*, vol. 7, pp. 165640–165652, 2019.

[12] M. Li, J. Zhao, and J. Zhang, "Split learning for privacy-preserving federated learning in IoT networks," in *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11991–12002, 2020.

[13] M. Shafay, R. W. Ahmad, K. Salah, I. Yaqoob, R. Jayaraman, and M. Omar, "Blockchain for deep learning: review and open challenges," *Cluster Computing*, vol. 26, no. 1, pp. 197-221, 2023

[14] M. Soori, B. Arezoo, and R. Dastres, "Artificial intelligence, machine learning and deep learning in advanced robotics, a review," *Cognitive Robotics*, vol. 3, pp. 54-70, 2023.

[15] A. Rahman, M.S. Hossain, G. Muhammad, D. Kundu, T. Debnath, M. Rahman, M. S. I. Khan, P. Tiwari, and S. S. Band, "Federated learning-based AI approaches in smart healthcare: concepts, taxonomies, challenges, and open issues," *Cluster Computing*, vol. 26, no. 4, pp. 2271–2311, 2023.

[16] S. Rani, A. Kataria, S. Kumar, and P. Tiwari, "Federated learning for secure IoMT-applications in smart healthcare systems: A comprehensive review," *Knowledge-Based Systems*, vol. 274, p.110658, 2023.

[17] I. A. Kandhro, S. M. Al Enezi, F. Ali, A. Kehar, K. Fatima, M. Uddin, and S. Karuppayah, "Detection of real-time malicious intrusions and attacks in iot empowered cybersecurity infrastructures," *IEEE Access*, 2023.

[18] C. Roche, P. Wall, and D. Lewis, "Ethics and diversity in artificial intelligence policies, strategies and initiatives," *AI and Ethics*, vol. 3, no. 4, pp. 1095–1115, 2023.

[19] D. Sirohi, N. Kumar, P.S. Rana, S. Tanwar, R. Iqbal and M. Hijjii, "Federated learning for 6G-enabled secure communication systems: A comprehensive survey," *Artificial Intelligence Review*, vol. 56, pp. 11297–11389, 2023.

[20] M. Li, J. Zhao, and J. Zhang, "Split learning for privacy-preserving federated learning in IoT Networks," in *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11991–12002, 2020.

[21] S. Lee, H. Kim, and C. Park, "Privacy-preserving machine learning via split learning," *IEEE Access*, vol. 7, pp. 165640–165652, 2019.

[22] M. Geppert, V. Larsson, J. L. Schönberger, and M. Pollefeys, "Privacy-preserving partial localization," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Orleans, Louisiana, 2022, pp. 17337-17347.

[23] K. U. Chouhan, N. P. K. Jha, R. S. Jha, S. I. Kamaluddin, and N. Giri, "Mobile keyword prediction using federated learning," in *International Journal for Research in Applied Science & Engineering Technology*, vol. 11, no. 4, 2023.

[24] S. Banabilah, M. Aloqaily, E. Alsayed, N. Malik, and Y. Jararweh, "FL review: Fundamentals, enabling technologies, and future applications," in *Information Processing & Management*, vol. 59, no. 6, p. 103061, 2022.

[25] Q. Meng, M. Huang, Y. Xu, N. Liu, and X. Xiang, "Decentralized distributed deep learning with low-bandwidth consumption for smart constellations," *Space: Science & Technology*, 2021.

[26] D. Lee, J. Lee, H. Jun, H. Kim, and S. Yoo, "Triad of split learning: Privacy, accuracy, and performance," in International Conference on Information and Communication Technology Convergence, Jeju Island, South Korea, 2021, pp. 1185–1189.

[27] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu, "1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs," in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Florence, 2014, pp. 1058–1062.

[28] 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*, M. H. ur Rehman and M. M. Gaber, Eds., Cham, Switzerland: Springer, 2021, ch. 4, pp. 79-109.

[29] V. Turina, Z. Zhang, F. Esposito, and I. Matta, “Federated or split? A performance and privacy analysis of hybrid split and federated learning architectures,” in IEEE International Conference on Cloud Computing, Chicago, USA, 2021, pp. 250–257.

[30] S. Pal, M. Uniyal, J. Park, P. Vepakomma, R. Raskar, M. Bennis, M. Jeon, and J. Choi, “Server-side local gradient averaging and learning rate acceleration for scalable split learning,” in Proceedings of the Association for the Advancement of Artificial Intelligence, Palo Alto, California, vol. 36, pp. 1-9, 2022.

[31] X. Liu, Y. Deng, and T. Mahmoodi, “Energy efficient user scheduling for hybrid split and federated learning in wireless UAV networks,” in Proceedings of IEEE International Conference on Communications, Seoul, South Korea 2022, pp. 1–6.

[32] Y. Zhou, B. Rao, and W. Wang, “UAV swarm intelligence: Recent advances and future trends,” in *IEEE Access*, vol. 8, pp. 183856–183878, 2020.

[33] S. Zhang, Y. Zeng, and R. Zhang, “Cellular-enabled UAV communication: A connectivity-constrained trajectory optimization perspective,” in *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 2580–2604, 2019.

[34] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” in *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[35] X. Liu, Y. Deng, and T. Mahmoodi, “Wireless distributed learning: A new hybrid split and federated learning approach,” in *IEEE Transactions on Wireless Communications*, vol. 22, no. 4, pp. 2650–2665, April 2023.

[36] Y. Gao, L. Zhang, J. Zhao, and Z. Jiang, "Improved U-Net with channel and spatial attention for coronary angiography segmentation," in 16th International Conference on Complex Medical Engineering, Zhongshan, China, 2022, pp. 123-126.

[37] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, “A survey on distributed machine learning,” in *ACM Computing Surveys*, vol. 53, no. 2, Article no. 30, 2021.

[38] S. Zhang, H. Tu, Z. Li, S. Liu, S. Li, W. Wu, and X. S. Shen, "Cluster-HSFL: A Cluster-based hybrid split and federated learning," in Proceedings of International Conference on Communications in China, China, 2023, pp. 1–2.

[39] R. Gupta and K. N. Sivarajan, "An overview of split learning: Techniques, applications, and challenges," in Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 2018, pp. 112–128.

[40] J. Konecný, H. B. McMahan, F. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in Proceedings of the 29th Conference on Neural Information Processing Systems, Barcelona, Spain, 2016, pp. 1–10.

[41] B. Jones, "Financial fraud detection using FL: Opportunities and challenges," in Proceedings of the International Conference on Machine Learning, Long Beach, California, vol. 25, 2019, pp. 1117-1651.