Journal Pre-proof

A survey on security and privacy of federated learning

Viraaji Mothukuri, Reza M. Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, Gautam Srivastava

PII: S0167-739X(20)32984-8
DOI: https://doi.org/10.1016/j.future.2020.10.007
Reference: FUTURE 5875

To appear in: *Future Generation Computer Systems*

Received date : 29 December 2019
Revised date : 11 September 2020
Accepted date : 7 October 2020

Please cite this article as: V. Mothukuri, R.M. Parizi, S. Pouriyeh et al., A survey on security and privacy of federated learning, *Future Generation Computer Systems* (2020), doi: https://doi.org/10.1016/j.future.2020.10.007.

A Survey on Security and Privacy of Federated Learning

Viraaji Mothukuri[a], Reza M. Parizi[a], Seyedamin Pouriyeh[b], Yan Huang[a], Ali Dehghantanha[c], Gautam Srivastava[d,e,*]

[a] *The Department of Software Engineering and Game Development, Kennesaw State University, GA 30060, USA*
[b] *The Department of Information Technology, Kennesaw State University, GA 30060, USA*
[c] *Cyber Science Lab, School of Computer Science, University of Guelph, Ontario, Canada*
[d] *Department of Math and Computer Science, Brandon University, MB, Canada*
[e] *Research Center for Interneural Computing, China Medical University, Taichung 40402, Taiwan, Republic of China*

Abstract

Federated learning (FL) is a new breed of Artificial Intelligence (AI) that builds upon decentralized data and training that brings learning to the edge or directly on-device. FL is a new research area often referred to as a new dawn in AI, is in its infancy, and has not yet gained much trust in the community, mainly because of its (unknown) security and privacy implications. To advance the state of the research in this area and to realize extensive utilization of the FL approach and its mass adoption, its security and privacy concerns must be first identified, evaluated, and documented. FL is preferred in use-cases where security and privacy are the key concerns and having a clear view and understanding of risk factors enable an implementer/adopter of FL to successfully build a secure environment and gives researchers a clear vision on possible research areas. This paper aims to provide a comprehensive study concerning FL's security and privacy aspects that can help bridge the gap between the current state of federated AI and a future in which mass adoption is possible. We present an illustrative description of approaches and various implementation styles with an examination of the current challenges in FL and establish a detailed review of security and privacy concerns that need to be considered in a thorough and clear context. Findings from our study suggest that overall there are fewer privacy-specific threats associated with FL compared to security threats. The most specific security threats currently are communication bottlenecks, poisoning, and backdoor attacks while inference-based attacks are the most critical to the privacy of FL. We conclude the paper with much needed future research directions to make FL adaptable in realistic scenarios.

*Corresponding author
Email addresses: vmothuku@students.kennesaw.edu (Viraaji Mothukuri), rparizi1@kennesaw.edu (Reza M. Parizi), spouriye@kennesaw.edu (Seyedamin Pouriyeh), yhuang24@kennesaw.edu (Yan Huang), ali@cybersciencelab.org (Ali Dehghantanha), srivastavag@brandonu.ca (Gautam Srivastava)

Keywords: Artificial Intelligence, Machine Learning, Distributed Learning, Federated Learning, Federated Machine Learning, Security, Privacy.

1. Introduction

In traditional machine learning (ML), the efficiency and accuracy of models rely on computational power and training data of a centralized server. Shortly speaking, in traditional ML, user data is stored on the central server and utilized for training and testing processes in order to develop comprehensive ML models ultimately. Centralized-based ML approaches, in general, are associated with different challenges including computational power and time, and most importantly security and privacy with respect to users' data that has been neglected for a long time. Federated Learning proposed by [1] has recently emerged as a technological solution to address such issues.

Federated Learning (FL) [2] offers a way to preserve user privacy by decentralizing data from the central server to end-devices and enables AI benefits to domains with sensitive data and heterogeneity. This paradigm came to light mainly for two reasons: (1) The unavailability of sufficient data to reside centrally on the server-side (as opposed to traditional machine learning) due to direct access restrictions on such data; and (2) Data privacy protections using local data from edge devices, i.e., clients, instead of sending sensitive data to the server where network asynchronous communication comes into play. Preserving data privacy provides feasibility to leverage AI benefits enabled through machine learning models efficiently across multiple domains. Additionally, computational power is shared among the interested parties instead of relying on a centralized server by iterative local model training processes on end-devices. With its decentralized data concept, FL is one of the growing fields in the area of ML in recent years, as it comes with security and privacy features promising to abide by emerging user data protection laws [3, 4]. In addition to privacy, FL enables ML benefits to smaller domains where sufficient training data is not available to build a standalone ML model.

As phrased by authors in [1] "FL brings the code to the data, instead of the data to the code and addresses the fundamental problems of privacy, ownership, and locality of data". Even before we can fully appreciate the fact that in FL, data stays intact on the users' device and the traditional upload of data over the network can be gracefully skipped, we are sharing the model parameters and global ML model with each and every client, which opens numerous ways to exploit vulnerabilities in the FL environment. As FL is in the initial steps of studies, many researchers in different communities are eagerly working to improve the existing frameworks and ensure privacy and security of user data within FL.

Each time a new technology is introduced and a new ecosystem is created, a range of technical ripple effects typically come into fruition over time. In a less positive way and as good as the FL sounds, the introduction of FL has arguably required more profound research into its confirmation, particularly with respect

to security and privacy aspects. Therefore, we can question what type of security and privacy issues are we facing now and can we imagine occurring in the future as a consequence of the adoption of this technology? This paper aims to provide the answer to these types of questions and shed light on possible unwanted future security and privacy states that we should be mindful of and prepare for.

Privacy-preserving promises of FL attracts different domains that may contain sensitive data. To an extent, FL does solve privacy concerns of sensitive data in ML environments however at the same time model parameter sharing and an increased number of training iterations and communications exposes the federated environment to a new set of risks and opens new avenues for hacking [5] as well as curious attackers to trace vulnerabilities to manipulate ML model output or get access to sensitive user data. To ensure that we out-turn the benefits of FL over risks and utilize the features of FL properly, we have an immediate need to be on top of this area of research to investigate all possible security and privacy attacks on FL. Without precise information and clear vision, FL may be pushed back without giving a fair chance to explore and leverage its benefits.

As seen in recent publications, the majority of work proposed in the FL space aims to apply this new framework in some shape and form to different domains. Our work touches on the issues within FL and could be used as a reference to promote future cybersecurity-related research advancing the acceptance of this framework. To this end, we address the research objectives by identifying and evaluating open security & privacy threats as well as mitigation strategies of FL by answering several specific research questions.

1.1. Contributions

While research already exists on this topic, sufficient progress has not been made concerning understanding FL for its security and privacy risks. This work hopes to contribute a comprehensive overview of FL security in terms of a formal definition, achievements, and challenges, which makes it stand out in comparison to previous works. In doing so, the work can contribute an overall blueprint for data scientists and cybersecurity research on designing FL-based solutions for alleviating future challenges. The outline of the contributions of this paper relative to the recent literature in the field can be summarized as follows:

- Providing a classification and overview of the approaches and techniques in the realization of FL.
- Identifying and examining security vulnerabilities and threats in FL environments both FL-specific and general ML-based attacks related to FL.
- Identifying and evaluating privacy threats, their mitigation techniques, and the trade-off cost associated with privacy-preserving techniques in the FL environments.
- Providing insights into existing defense mechanisms and future directions to enhance the security and privacy of the FL implementation.

1.2. Paper Organization

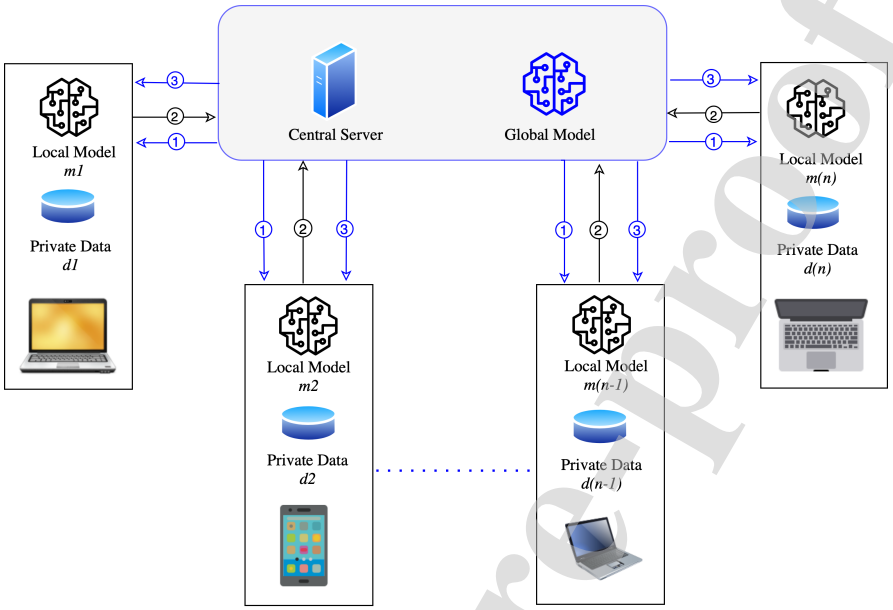
The rest of this paper is structured as follows. Section 2 on page 4 gives background information about FL and its underlying working process. Section 3 on page 6 classifies approaches and techniques related to the FL. Section 4 on page 17 provides the research questions on the security of FL and presents the research results. Similarly, Section 5 on page 28 gives the research questions on the privacy of FL and presents the research results. Section 6 on page 34 gives the related work. Section 7 on page 35 summarizes the future directions for security and privacy in the FL domain. Finally, Section 8 on page 37 gives the concluding remarks.

2. Federated Learning Landscape

In recent years, three major contributing factors have helped in the success of ML. The first and major contributing factor is the availability of a huge amount of data collected over the years (thanks to big data!). The second is computational power, technology has evolved so much that we have moved away from traditional computing devices to highly scalable and integrated microcircuit devices which help to train models faster and deploy directly on devices with less computational costs. For instance, we now have AI-ready smartphones with a pre-installed AI chip to make smartphones much smarter and intelligent and assist humans efficiently in day-to-day tasks. The third contributing factor is Deep Learning Models, which add much-needed intelligence to ML models. Models trained on self-taught deep learning algorithms are showing commendable success rates. For instance, Alpha-Go's [6] board game victory over the world champion has amazed people with its ability to master the game and win over human Intelligence. In spite of the immense success of ML [7], many domains can only wish to leverage its benefits but can not do so because of two major obstacles:

- Concerns on user data privacy and confidentiality and the laws that oversee them.
- Inability to build an ML model due to inadequate data or training cost on ML implementation of the computational cost involved for training an ML model.

Multiple cloud-based companies are providing well-trained ML models [8], which are able to bring ML knowledge and computational power at a cost however still we see that privacy and confidentiality concern remain unaddressed [9, 10, 11, 12, 13, 14]. In an effort to address such obstacles, the community has seen a promising ML-based framework, known as Federated Learning. FL addresses these concerns by providing a highly trained ML model without the risk of exposing training data. FL also tackles the issue of having inadequate data by providing a trust factor among heterogeneous domains. Such privacy-preserving



Step 1: Central Server shares initial model parameters with all the clients.
Step 2: Clients train their local model with initial parameters and share local model with central server.
Step 3: Central Server Aggregates the local models and shares global model with the clients.

Figure 1: FL Process Flow

techniques of FL attracts different communities to leverage it exclusively, preserving client data privacy and availing benefits of having a model trained on larger landscape data. FL is considered as an iterative process wherein each iteration the central ML model is improved. FL implementations can be generalized into the following three steps:

1. *Model selection*: In this step, the central pre-trained ML model (i.e., global model) and its initial parameters are initiated and then the global ML model is shared with all the clients in the FL environment.
2. *Local model training*: After sharing the initial ML model and parameters with all clients, the initial ML model at the client level (called local ML models) is trained with individuals training data.
3. *Aggregation of local models*: Local models are trained at the client level and updates are sent to the central server in order to aggregate and train the global ML model. The global model is updated and the improved model is shared among the individual clients for the next iteration.

FL is in a continuous iterative learning process that repeats the training steps of 2 & 3 above to keep the global ML model updated across all the clients. Figure 1 visualizes the FL architecture and training approach as discussed in the steps above.

In reality, FL has already shown its footprint in mobile applications for next word prediction on keyboards [15, 16, 17, 18] like Gboard by Google on Android mobile phones, and wake word detection [19], which enables voice assisting apps to detect wake word without risking exposure of sensitive user data. In medical domains, FL can be utilized to keep patient data private and enhance ML capabilities in assisting medical practitioners similar to the work in [20] which demonstrates the benefits of FL in the medical domain.

Apart from live production applications, there have been several research proposals with useful application use-cases experimenting with the use of FL for constructing privacy-persevering (or even secure) machine learning solutions in various domains. For instance, the research works in [21, 22] summarize possible applications for wireless communications with FL by avoiding communication overheads. For application use-cases in the security domain, FL has been advocated for malware classification [23], human activity recognition [24], anomaly detection [25], and intrusion detection [26], to name a few. For application use-cases in the intelligent transportation industry, there have several proposals that use an FL-based approach. Data sharing between autonomous cars and driving [27, 28], For preventing data leakage in vehicular cyber-physical systems [29], traffic flow prediction [30], and the detection of attacks in aerial vehicles [31] are examples of such works. For application use-cases in the computer vision domain, Fedvision for object detection using secure FL has been proposed by Webank in [32]. As for application use-cases in the medical domain, the attack detection in Medical cyber-physical systems that maintain sensitive information on patient's health records [33], and managing digital health record with FL [34] are more examples of FL applications. It is worth mentioning here that the primary focus of this paper is to investigate the potential security and privacy-related issues within FL. Thus, examining the applications of FL in different domains is slightly out of the scope with the purpose of this paper.

3. FL categorization of techniques/approaches

FL is under active development and uses a variety of techniques and approaches to realize its underlying technology in practice. For an emerging technology, the categorization of its techniques and approaches is a crucial first step that helps to understand and explore beyond the outlined big picture. This section gathers and gives an overview of the inner workings of such techniques from various points of view which will be used for a deeper understating of security and privacy aspects in later sections. Figure 2 shows our classification. In this classification, we cover the FL implementation network topology used to build FL environment, classification based on data availability and partition, aggregation/optimization algorithms built at the central server for preserving communication bandwidth/costs and aggregation logic, and open-source frameworks to realize FL in practice. Please note this list is not exhaustive and only includes some of the most current and common techniques and approaches in the FL realm. the detailed description of each classification including references for interested readers is given in Figure 2.

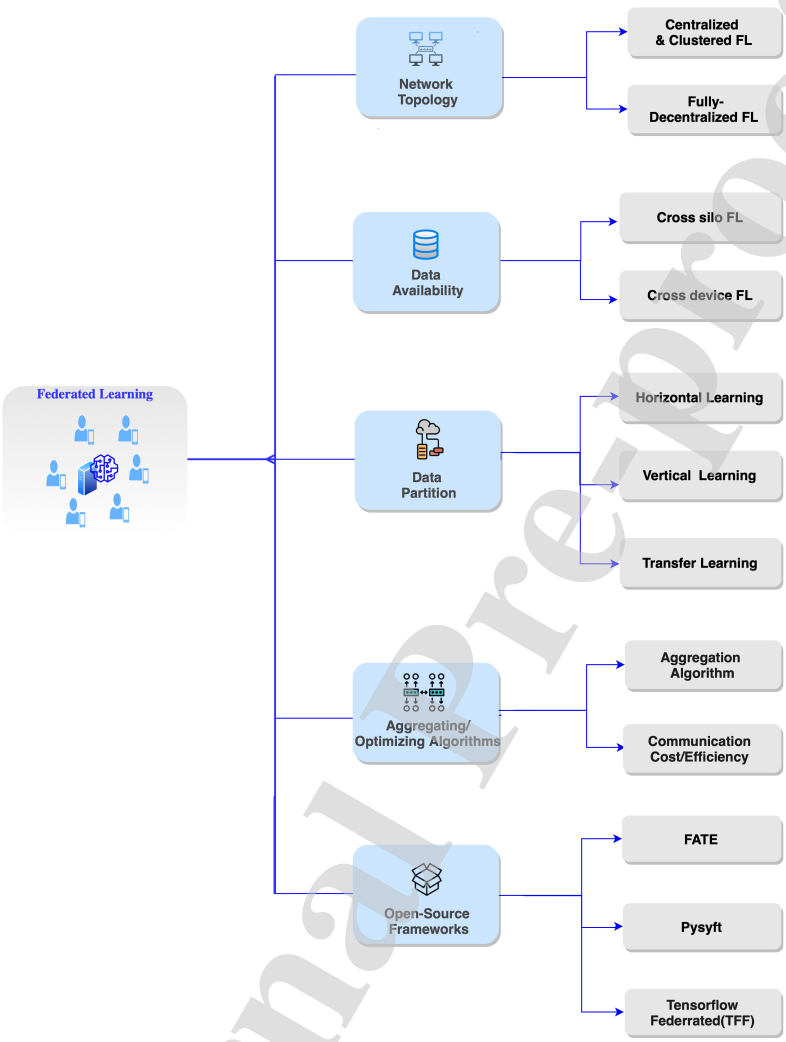


Figure 2: Overview of FL Approaches and Technologies

3.1. Network Topology

The focus of this category is the underlying architecture of FL, the way components are built to achieve a FL environment. Based on network topology, FL can be categorized as centralized or fully decentralized [35].

3.1.1. Centralized and Clustered FL

Even though FL is fundamentally based on a decentralized data approach, there is still a dependency on a centralized server to manage the responsibility of collecting trained models from clients involved in the FL environment, build

a global model, and share it back with all clients. This is mainly preferred to establish a third party system for building a trust factor among clients. Single server and multiple client topology/hub-and-spoke topology [36] are followed which ensures a centralized authority to monitor and manage the continuous learning process. Unlike the traditional centralized server which hosts data and trains a given model on shared data, the centralized server in the FL environment works only on a shared model through synchronous or asynchronous updates from clients. Gboard for Android keyboard developed using Tensorflow federated from Google can be exemplified as a centralized approach of the FL. Figure 1 represents the typical steps of such an approach. The current common implementations of FL in practice use a centralized approach.

The clustering technique is proposed to improvise FL which aims to address heterogeneity in FL clients' data in the centralized network topology. Clustering is one of the techniques which can help in detecting malicious updates. Research work from google in [37] proposes three algorithms to achieve a personalized version of the local model with minimal communication rounds. Authors suggest leveraging the algorithms as a combination or individually based on the use cases of user clustering, data interpolation, and model interpolation. In the user clustering algorithm, Clusters are created with a group of clients with similar data distribution, and with each client, an intermediate model is created to which helps the global model to converge faster. Hypothesis based clustering technique is followed for identifying clusters. A Federated Multi-task learning-based approach is proposed in [38] where FL clusters of clients are identified based on cosine similarity of local models. The initial set of clients are taken and split into clusters recursively based on the stopping criteria set based on the calculated cosine similarity.

Research work in [39] proposes Federated Stochastic Expectation Maximization (FedSEM) to train multiple global ML models and reach a solution. Loss function called Distance-based Federated loss (DF-Loss) is the objective of multi-center FL which is to find the optimal global model among the different global models from a multi-cluster environment. Authors in [40] propose the Iterative Federated Cluster Algorithm (IFCA) framework which tries to minimize the loss function of each FL client and tags the client to a cluster in each training round. Based on the source for averaging IFCA, it is proposed in two variants called model and gradient averaging. IFCA applies random initializing and multiple restarts to identify clients with a specific cluster and reach optimal values. Experimental results of IFCA show that the proposed approach works well in the linear model, convex, and non-convex neural networks.

3.1.2. Fully-decentralized FL

The fully-decentralized approach excludes dependency on the central server for model aggregation. Centralized authority is replaced with algorithms to establish trust and reliability. As demonstrated in [41], there is no concept of a global model and each participant improves their model by sharing information with neighbors. Peer-2-peer topology is followed and central authority is used once to establish a protocol to be followed in the network during training

rounds. For the practical approach of fully-decentralized learning, various add-on technologies or algorithms are proposed. Authors in [42] propose an Adaptive Averaging Algorithm which is based on the Byzantine concept, assuming more than $\frac{2}{3}$rd of systems involved in FL are honest. With this approach, a group of clients from different domains with a common goal collaborate to share data and build an ML model and leverage benefits of high accuracy [43, 44] without the need to rely on a third-party centralized server. Authors in [45] propose a framework called MATCHA to address network delays by providing critical links for the clients to communicate with one another. Research work in [46] demonstrates FL in peer-peer network. To help make this FL approach clear, we have visualized its process in Figure 3.

3.2. Data Partition

FL is extremely useful in building ML models where data is shared across different domains. FL's magic of keeping the data private throughout the training process brings in a variety of domains to leverage the smart features of ML. FL helps to overcome the limitations due to domain-specific constraints on user data and available smart benefits of ML by collaborating and enhancing the

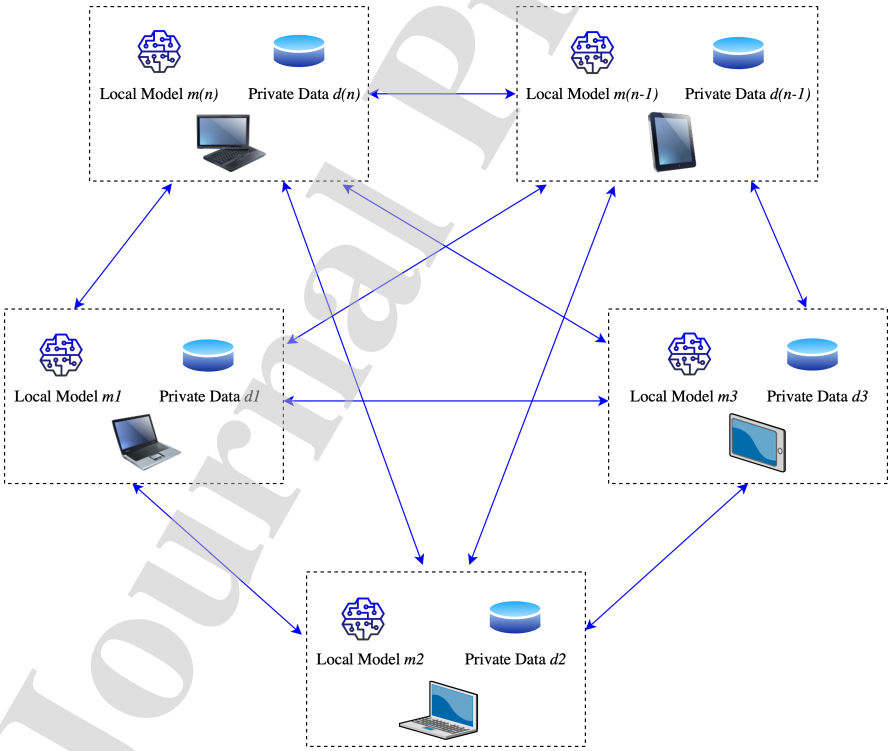


Figure 3: FL Fully-decentralized in P2P Topology

benefits of data with various domains/clients who may have similar/dissimilar areas of interest. In real-time the information, or to be more specific the statistical information derived from a set of user data, is utilized for many applications of various domains. In this era of digitalization, every click a user makes in cyberspace is captured for making derived statistical information, use-cases on such derived data may belong to applications in identical domains or different domains. Similarly, user data to develop such derived statistical information can be collected from domains belonging to dissimilar areas. Categorization of this section can be considered as "pre-work" for setting up an FL environment. The pre-work done on a set of interested parties with valuable user data can be vital for the FL process overall. Distribution characteristics of data, i.e., the differentiating & the colliding factors across heterogeneous data & clients participating in FL can be broadly categorized as Horizontal, Vertical, and Transfer learning (HFL[1], VFL[2], TFL[3]). These subcategories differ based on the data flow between the parties involved in the FL environment. Figure 4, 5 ,6 illustrates data partitioning in FL.

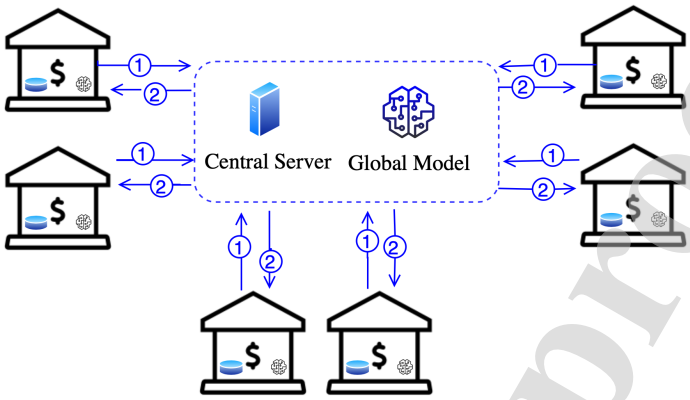
3.2.1. Horizontal Federated Learning

Horizontal federated learning 4 is defined as the circumstances where datasets on the devices share the same features with different instances. In this category of FL, clients have similar features in terms of domain, usage style of derived statistical information, or any other outcome from FL. The classic release of FL from Google [47] falls under this category. ML models which can predict the next possible words while a user key in the text input, works more accurately with real data feed as input from users. In this scenario, the Google keyboard app (known as Gboard) enhances itself with continuous learning from users of Android mobile devices. FL approach is implemented smartly by taking averaged updates from the user usage stats without tagging the user identity. Another example is from the medical domain, where a group of researchers working on an ML model which can analyze the medical images and predict the probability of possible occurrence of cancer cells. Medical images are considered as user sensitive data and cannot be shared as is due to constraints and laws of private medical data. However with FL, information on such private data can be shared securely either through a secure aggregated updates from each client.

3.2.2. Vertical Federated Learning

Vertical Federated learning 5 is an FL approach where the common data between unrelated domains is used to train the global ML model. Participants using this approach prefer to have an intermediate third party organization/resource to provide encryption logic to ensure that the only common data stats are shared. However it is not mandatory to have a 3rd party intermediate

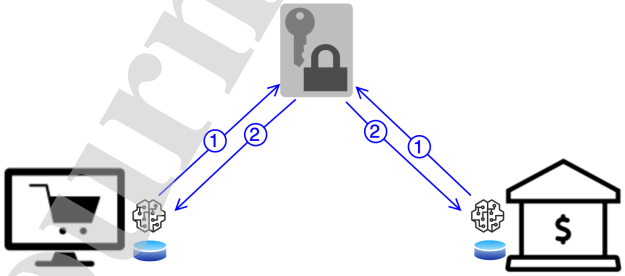
[1]Horizontal Federated Learning (HFL)
[2]Vertical Federated Learning (VFL)
[3]Transfer Federated Learning (TFL)



Step 1: Group of homogeneous clients from same domain contributing to train the global ML model
Step 2: Trained global ML model is downloaded & utilized at each client node, training rounds repeats

Figure 4: Data Partition: Horizontal Federated Learning

entity, research work in [48] demonstrates the implementation of vertical federated learning without 3rd party involved for encryption. Real-time use-cases for vertical federated learning approach would be a scenario where a marketing team of a credit card division in a bank would like to enhance their ML model by learning most purchased items from online shopping domains. Only the common user in the bank and shopping site details are shared to train the ML model, the intermediate encryption logic ensures this secure and restricted share of derived stats. With this liaising of information exchange, banking domains can serve customers better with relevant offers and online shopping domains can revise their points allocation for customers using credit cards.

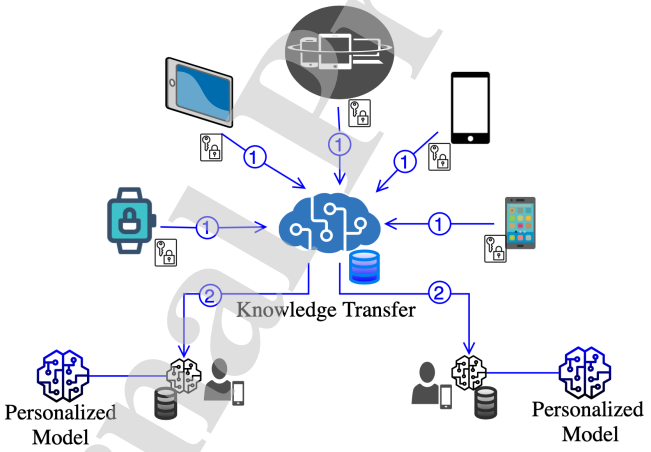


Step 1: Heterogeneous clients contribute to train the global model by sharing encrypted local model updates
Step 2: Trained global ML model is downloaded & utilized at each client node, training rounds repeats

Figure 5: Data Partition: Vertical Federated Learning

3.2.3. Federated Transfer Learning

Federated Transfer learning 6 is an implementation of the existing classic ML transfer learning technique which is a way to train a new requirement on a pre-trained model that is already trained on a similar dataset for solving a completely different problem. In the ML world, training on a pre-trained model gives much better results in comparison to training done on a fresh model built from scratch. In [49, 50, 51, 52, 53], the authors explain about the implementation of FL in transfer learning modes. A real-time example would be similar to vertical federated learning with a few changes. Instead of restricting conditions to share only matching data information, participants can benefit from larger datasets and well-trained ML model stats to serve their individual requirements. As explained in [49], a real-time example is to apply a global model on a cloud to a personalized user which can be further used to provide a personalized model on wearable Internet of Things (IoT) devices of a specific user. This example is illustrated in Figure 6. There are few research works [54, 55, 56] called FedRL, which combine the reinforcement ML algorithms with FL in applications aiming for personalized AI. FedRL utilizes FL to apply transfer learning from a well-trained secure model with which direct transfer learning is not permitted.



Step 1: Heterogeneous clients train a global model on cloud with encrypted techniques, similar to VFL
Step 2: From the pre-trained cloud ML model transfer learning is applied to get personalized individual ML models

Figure 6: Data Partition: Transfer Federated Learning

3.3. Data Availability

As discussed in the work from Google [36], based on the availability of data and the number of client nodes, FL can be categorized as Cross-silo FL and Cross-device FL.

3.3.1. Cross-silo FL

In this case, clients are typically of small-scale numbers ranging from 2-100 devices, usually indexed and are almost always available for training rounds. Training data can be classified into either horizontal or vertical learning. Computation and communication bottlenecks are major issues. Cross-silo FL is more flexible compared to cross-device FL and used in scenarios within organizations or within groups of organizations to train the ML model with their confidential data. As in vertical and transfer learning implementations, an encryption technique is preferred to restrict the inference of information from each client. Built on FATE [57] framework, the research work in [58] demonstrates cross-silo with homomorphic encryption and proposes gradient quantization [59] based batch encrypt algorithm to reduce computation and communication costs in FL environment [60].

3.3.2. Cross-device FL

The FL approach with a large number of clients in a similar domain with similar interests from the global model is known as Cross-device FL. Due to the huge number of clients, it is difficult to track and maintain transaction history logs. Mostly clients connect using unreliable networks where selection/participation in training rounds happens randomly. Similar to data partitioning in horizontal federated learning, resource allocation strategies [61] like client selection/importance [62] , device scheduling [63] are used to choose updates from better contributing clients. Incentive mechanisms like game theory [64] are designed to motivate clients to contribute towards FL. For use-cases with a huge number of clients such as IoT or mobile applications [16], this type is a great fit.

3.4. Aggregation/Optimizing algorithms

Algorithms that contribute towards the binding updates from clients to achieve the target global ML model vary based on the priorities and architecture of FL. Configuring this logic is crucial as it addresses the heterogeneity of clients, varying weights in every update from each client, and communication issues. In the centralized version of FL, there is an aggregation algorithm that works from orchestrating training to optimizing updates. Few proposed algorithms focus on optimal selection of clients, optimizing training rounds for efficient communication and costs of FL. The algorithms which are used for integrating, enhancing, optimizing, aggregating and achieving consensus in different approaches of FL are covered in this subsection.

3.4.1. Aggregation Algorithm

The aggregation algorithm plays a key role in any centralized topology and horizontal federated learning style environment. It is the logic [65] that combines the local model updates from all the clients that participated in the training round. Many proposed algorithms are targeting to achieve enhanced privacy of local model updates or to preserve communication bandwidth or to facilitate

asynchronous updates from clients. Federated averaging varies in each implementation of FL based on the *pre-config* parameters. The current algorithms are discussed below:

- *FedAvg:* Google's implementation [1] of FL, introduced Federated Averaging algorithm (called FedAvg) based on SGD optimization algorithm [4]. As part of FedAvg algorithm logic, the central server acting as a coordinator or orchestrator starts the process of FL training by sharing global parameters and global model to a group of selected clients referred to as mini-batch, who trains the ML model with local training data and global model parameters and shares the trained model weights with the centralized server. The global model is generated by utilizing averaging logic to compute the weighted sum of all the received local model updates at the centralized server. Configurable criteria, number of training rounds is the stopping condition for the coordinator to stop the training rounds and average the local model updates.

- *SMC-Avg:* As explained in [66], secure aggregation is based on the concept of the Secure Multiparty Computation (SMC) algorithm, which aggregates private values of mutually distrustful parties without revealing information about their private values. Designed for addressing challenges of the mobile device-based FL environment. This algorithm has a fault tolerance limit, which means it works well even if $\frac{1}{3}$rd users fail to participate in the designed protocol.

- *FedProx:* A modified version of the FedAvg [1] algorithm is proposed in [67] to tackle heterogeneity in FL. The experimental results of this paper indicated positive results for FedProx in heterogeneous networks. However, the experimental study [65] performed on FedAvg and FedProx demonstrated that FedAvg achieves the highest accuracy among federated averaging algorithms. FedProx considers variation of computational power and different factors in devices participating in the FL training round. FedProx also introduces a proximal term to deal with non-uniformity in local updates.

- *FedMA:* It is proposed by the authors in [68] for constructing a shared model for CNNs and LSTM based ML model updates in FL environments. FedMA does the average of models at the central server by layer-wise matching and averaging hidden elements like neurons and channels in neural networks. Similar to work proposed in [69], this approach works on the same concept of matching neurons and being efficient only on simple neural networks such as fully connected networks. As shown in their respective paper [68], the FedMA approach works well with heterogeneous clients and outperforms FedAvg and FedProx within a few training rounds.

[4] Stochastic gradient descent (SGD)

FedMA can also use communication as a variant that sends global model matching results to clients at the start of each training rounds and adds new neurons in the local model as an update to the global model instead of matching, to achieve better overall performance.

- *Scaffold:* Stochastic Controlled Averaging for FL (Scaffold) proposed in [70] addresses gradient dissimilarity/client-drift issue faced in federated averaging algorithm 3.4.1 for FL approaches where clients are stateful with which algorithm can maintain/control variants at the client and server-side to ensure that client updates are in moving towards with global convergence.

- *Tensor Factorization:* Few applications in the medical domain prefer tensor factorization [71, 72] to limit the amount of information shared. Tensor factorization converts bulk of medical records to meaningful phenotypes. Authors in [72] propose Tensor factorization for privacy preserving computational phenotyping (TRIP) for applications using tensor factorization. TRIP shares summarized data and phenotyping which helps to preserve user-data privacy.

- *Personalisation-based algorithms:* Research work in [73] proposes an approach to segregate neural network layers into the base and personalized layers at each client node. In this approach, the federated averaging is applied only to the base layer updates from FL clients, which helps to omit the heterogeneity of FL clients and focus on the actual learning task. APFL (Adaptive personalized Federated Learning) proposed in [74] suggests learning from the combination of the global model and local model to achieve a personalized version of the ML model at each FL client. Deviating from the classic FL approach of sharing a single local model, this approach proposes three models at each FL client, one local model trained on local data, the second one a local copy of the global model, and third a personalized ML model built using the mixing parameter. Based on the average distribution of data across the FL clients a mixing parameter is calculated at each client node which keeps changing based on the distance between the three (local, global, and personalized) models. Experimental results of APFL perform better than federated averaging algorithm (i.e., FedAvg) in achieving a personalized model for each FL client.

3.4.2. Communication Efficiency and costs

Algorithms focusing on reducing communication efficiency and costs involved during FL training rounds are separately discussed in this part.

- *FedBCD:* The Federated Stochastic Block Coordinate Descent (FedBCD) algorithm is proposed in [75], which is similar to FedAvg algorithm. Two variants called FedBCD-p and FedBCP-s algorithms are proposed for parallel and sequential updating of gradients. FedBCD aims to reduce total communication rounds by skipping updates for every iteration. Clients

share single value for sample instead of model parameters and perform many local updates before sharing parameters with other clients. Evaluation of this approach states promising results in achieving desired accuracy rates.

- *FedAttOpt:* Attentive Federated Aggregation (FedAttOpt) proposed in [76] adds an attention-augmented mechanism to model aggregation at the central server of FL which calculates the attention score based on the contribution of each client. The attention score or the contribution factor of the client is calculated based on the gap in common knowledge between client and global model. FedAttOpt utilises attention score to help client node to train and accumulates useful common knowledge to all the client nodes.

- *Asynchronous FL Training Rounds:* The higher the number of clients the higher the risk of communication bottlenecks and computational costs. There are few research works that address communication efficiency by targeting minimal communication costs during the training rounds of FL. Research work in [77] explores strategies to minimize communication costs in FL by using the layer-wise asynchronous update on neural networks. Similarly, TOR [78] is based on the asynchronous aggregation of models and few strategies to perform even with low-communication bandwidth. Asynchronous aggregation implies to the process where the central server waits for the client device to be online to share updates asynchronously. The authors in [79] explore the Co-op algorithm, which is designed for FL in asynchronous implementation, where aggregation of local models is done on an offline basis based on the availability of clients.

- *Communication costs:* The approach proposed in [80] sets predefined rules for selecting client updates during FL training rounds which helps in minimizing communication costs by eliminating least contributed FL client's updates. Federated Distillation (FD) and Federated Augmentation (FAug) are proposed in [81] to mitigate the communication overhead and costs in FL training rounds. FD shares local model output instead of the whole local model, each client acts as a student who learns from the aggregated knowledge of all other clients. For each label in training, data mean logit vectors are shared with the server to calculate global average values. FAug uses GANs for data augmentation at each client which generates a shareable IID training data.

3.5. Open-Source Frameworks

There are currently a few open-source frameworks for researchers and developers to explore FL solutions. A quick summary of the major state-of-the-art tools is listed in the following.

- *Tensorflow Federated:* Google's TensorFlowFederated (TFF [5]) has a productionized version in Gboard, which enables Android mobile users to predict the next word while using the keyboard on their mobile phones [82, 83, 84, 85, 86], is one of the first attempts in the community to bring FL into reality. TFF provides integration with Google Kubernetes Engine (GKE) [87] or a Kubernetes cluster for orchestrating interaction with clients and the central server of FL. It also provides docker images to deploy a client and connect through gRPCs [88] calls. TFF uses FL specific training datasets generated using LEAF [89] provided in Tensorflow APIs.

- *PySyft:* written in Python on top of the PyTorch framework, Pysyft (Pysyft [6]) provides a virtual hook for connecting to clients through a WebSocket port [90, 91]. An aggregator or orchestrating server maintains pointers to the ML model and sends it to each participating client to train with their local data and gets it back for federated averaging. Federated averaging algorithm averages the model weights and scales them to maintain consistency in irregular coverage of data across clients. Apart from the basic approach of FL, PySyft provides support for asynchronous and synchronous approaches of FL and integration with existing encryption strategies like differential privacy.

- *FATE:* from Webank developers called FATE (FATE [7]), which is being improvised with every release. FATE provides a framework to implement FL in Horizontal, vertical, and transfer learning modes. It can be implemented with either docker images or manual steps. There is an open-source GitHub code, which provides training datasets for simulating known attacks on FL. The authors in [92, 57] utilize such malicious user datasets to explore the impact of attacks in FL. This framework provides production-ready APIs with Kubernetes integration.

There are more frameworks [93, 94, 95, 96, 97] for implementing FL that are being experimented with. Based on the experience of researchers in the field, it seems as though PySyft provides a more stable environment to develop FL solutions as opposed to TFF which is in an early experimental stage.

4. Security in Federated Learning

The FL technology adopters and developers should adhere to information security fundamentals such as **Confidentiality**, **Integrity**, and **Availability**. The decentralized approach of having a huge number of clients for collaborative training and exposure of model parameters makes FL vulnerable to various

[5] TFF `https://www.tensorflow.org/federated/federated_learning`
[6] Pysyft `https://blog.openmined.org/tag/pysyft/`
[7] FATE `https://fate.fedai.org`

attacks and open to risks. Current research on exploring vulnerabilities and proposing frameworks to mitigate the risks is very limited.

We set out to investigate the following research questions on the security aspect of the FL[8]:

- **RSQ1:** What are the source of vulnerabilities in FL ecosystem?
- **RSQ2:** What are the security threats/attacks in FL domain?
- **RSQ3:** What are the unique security threats to FL in comparison to distributed ML solutions?
- **RSQ4:** What are the defensive techniques for security vulnerabilities of FL?

In the following sections, we discuss the results based on each research question and provide an analysis of the strengths and weaknesses of the current works.

4.1. RSQ1: What are the source of vulnerabilities in the FL ecosystem?

A vulnerability can be defined as a weakness in a system which gives an opportunity to curious/malicious attacker to gain unauthorized access [98]. Knowledge of knowing (open) vulnerabilities of a system or framework helps to manage and defend against the possible attacks. Identifying vulnerabilities will help to build a more secure environment by implementing pre-requisites for defending loopholes. Failing to protect usage and exposure of personally identifiable information (PII) or failing to adhere to data protection laws will not just cause bad publicity, it can also cost many more consequences by law. It is a mandatory step for FL developers to scan for all sources of vulnerabilities and tighten defenses to ensure the security and privacy of the data. For a better insight into vulnerabilities, we categorize the source of vulnerabilities in the FL process. Our results show that there are five various sources, listed below, that can be considered as weak points for exploitation.

- *Communication Protocol*: FL implements an iterative learning process with randomly selected clients which involves a significant amount of communication over a given network. The FL approach suggests a mixed network [78], which is based on public-key cryptography that keeps source and message content anonymous throughout the communication. As FL has more rounds of training, a non-secure communication channel is an open vulnerability.
- *Client Data Manipulations*: FL in a larger landscape has numerous clients that are open for attackers to exploit model parameters and training data. Access to the global model may be further vulnerable to data reconstruction attacks.

[8]Research Security Question (RSQ)

- *Compromised Central Server*: The central server should be robust and secure, the central server is responsible for sharing initial model parameters, aggregating local models and sharing global model updates to all the clients. The cloud-based or physical server picked for this job should be checked to ensure that open vulnerabilities of the server are not exploited by curious attackers.

- *Weaker Aggregation Algorithm*: The aggregation algorithm is the central authority. In other words, as the local model's update, it should be intelligent to identify abnormality with client updates, and it should have a configuration to drop updates from suspicious clients. Failing to configure a standardized aggregation algorithm will make the global model vulnerable. Section 3.4 gives a list of algorithms proposed for FL aggregation logic.

- *Implementer's of FL Environment*: Intentionally or unintentionally a team of architects, developers, and deployers involved in the implementation of FL may turn-out to be a source of a security risk. Either due to the confusion or lack of understanding in what is consider as sensitive user data and what is not can be the reason for the breach of security and privacy. The risk from the implementers may be due to the basic fact that they miss taking proper measures to scan for sensitive data and plan the usage of it, hiding of the facts while taking consent of users on data usage.

4.2. RSQ2: What are the security threats/attacks in FL domain?

Threat/attack is the possibility of a vulnerability being exploited by a malicious/curious attacker impacting the system security and violating its privacy policies. In FL, generally, the malicious agent utilizes vulnerabilities [99] to take control of one or more participants (i.e., clients) in order to manipulate the global model ultimately. In such a scenario, the attacker targets different clients with hopes of accessing local data at rest, training procedure, hyperparameters, or updated weights in transit [100] to modify and launch attacks in the global model. The security threats/attacks are classified and their respective descriptions are discussed in the following sub-sections.

4.2.1. Poisoning

An attack with a major possibility of occurrence in FL is known as poisoning [101, 102], as each client in FL has access to the training data the possibility of getting tampered data weights added to the global ML model is very high. Poisoning can occur during the training phase and can impact either the training dataset or the local model in-turn/indirectly tampering the global ML model performance/accuracy. In FL, model updates are taken from a large group of clients. That is, the probability of poisoning attacks from one or more clients' training data is high so is the severity of the threat. Poisoning attack targets at various artifacts in the FL process. Next we present a brief description of poisoning attack classifications:

- *Data Poisoning*: The concept of a data poisoning attack against ML algorithms was for the first time presented by authors in [103] where the attacker targets the vulnerability of the support vector machines algorithm and tries to incorporate malicious data points in the training phase in hopes of maximizing the classification error. Since then, a wide variety of approaches have been proposed to mitigate data poisoning attacks in ML algorithms in different settings including centralized and distributed environments respectively. While the FL environment enables clients to actively contribute to training data and sending model parameters to the server, it provides this opportunity for malicious clients to poison the global model by manipulating the training process. Data poisoning in FL is defined as generating dirty samples to train the global model in hopes of producing falsified model parameters and send them to the server.

  Data injection can be also considered as a subcategory of data poisoning where the malicious client may inject malicious data into client's local model processing. As a result, the malicious agent can take control over multiple client's local models and ultimately manipulate the global model with malicious data.

- *Model Poisoning*: While in data poisoning the malicious agent aims to manipulate the global model using fake data, in model poisoning, the malicious agent targets the global model directly. Model poisoning attacks have been shown to be more effective compared to data poisoning attacks in recent researches [104, 100, 105]. In fact, the effectiveness of model poisoning attacks tends to rise when there is a large-scale FL product in a place with many clients. In general, in the model poisoning attack, the malicious party can modify the updated model before sending it to the central server for aggregation and as a result, the global model can be easily poisoned.

- *Data Modification*: Data tampering/modification attacks may involve changing/altering the training dataset like feature collision [106] which is merging two classes in the dataset in an attempt to fool the ML model to always misclassify the targeted class. Some techniques include simply adding a shade or pattern of another class to a targeted class that can confuse the ML model. Another technique is to do a random label swap of the training dataset. Data injection and Data modification attacks can be considered as a type of ML data poisoning attacks [107] in FL.

4.2.2. Inference

Inference attacks are more of a threat to privacy (as pointed in Section 5.1) yet we are including it here for overall comparison of threats in FL. The severity of inference attacks is highly similar to poisoning attacks as there is a very high possibility of inference attacks from either the participants or a malicious centralized server in the FL process.

4.2.3. Backdoor Attacks

Poisoning and Inference attacks are more transparent compared to backdoor attacks. A backdoor attack is a way to inject a malicious task into the existing model while retaining the accuracy of the actual task. It is difficult and time-consuming to identify backdoor attacks as the accuracy of the actual ML task may not get impacted right away. The authors in [92, 100] experiment on how backdoor attacks are implemented. Furthermore, the authors in [108, 109] suggest model pruning and fine-tuning as a solution to mitigate the risks of backdoor attacks. The severity of backdoor attacks is high as it takes significant time to identify the occurrence of the attack. Moreover, the impact of the attack is high as backdoor attacks have the capability to confuse ML models and predict the false positives confidently. Trojans threats [110, 100, 111, 112, 113] are a similar category of backdoor attacks which try to retain the existing task of ML models while performing a malicious task in stealth mode.

4.2.4. GANs

Generative Adversarial Network-based attacks in FL have been experimented and analyzed by many researchers [114]. With their ability to launch poisoning and inference attacks, GAN based attacks pose a threat to both the security and privacy of a given system, discussed more in Section 5.1.3. Research work in [115] demonstrates how GANs can be used to get training data through inference and use GANs to poison the training data. As all the possibilities of a GAN based threat cannot be foreseen, it is categorized as a high impact and prioritized threat. More information on GAN based attacks can be found in [116, 117]

4.2.5. System disruption IT downtime

Production system downtime is an unavoidable threat in the Information Technology (IT) industry. It is often observed that highly configured and secured applications need to take a downtime phase due to unplanned or planned activity on back-end servers. In FL, the severity of this threat is low as we have a local-global model on each and every client node and the training process can resume after the outage. Even with low severity, this is a considerable threat as downtime can be a well-planned attack to steal information from the FL environment.

4.2.6. Malicious server

In cross-device FL, most of the work is done at the central server. From selecting the model parameters to deploying the global model. Compromised or malicious servers have a huge impact, and honest but curious or malicious servers can easily extract private client data or manipulate the global model to utilize shared computational power in building malicious tasks in the global ML model.

4.2.7. Communication bottlenecks

One of the challenges in training an ML model from multiple heterogeneous device's data is communication bandwidth. In the FL approach, the communication cost is reduced by transferring trained models instead of sending huge amounts of data but still, we do have the need to preserve communication bandwidth. There are few algorithms [78, 1, 118] based on asynchronous aggregation of models and few strategies to perform well even with low-communication bandwidth. There are various research studies [119, 120] on preserving communication bandwidth in FL environment, discussed in section 3.4.2 . The severity of this threat is high, as communication bottlenecks can disrupt the FL environment significantly.

4.2.8. Free-riding Attacks

Few clients play a passive role and are connected to the environment only to leverage the benefits of the global ML model without contributing to the training process. Such passive clients may also insert dummy updates without training the ML model with their local data. Research works in [121] explored the free-riding attack in FL environments and proposed an enhanced version of the anomaly detection technique using autoencoders [122] to identify free-riders. This attack's impact would be more in a smaller landscape FL environment as the absence of client participation can negatively impact global model training. As the probability of this attack is lower, the severity of this attack is medium.

4.2.9. Unavailability

Unavailability or dropout of clients in between training processes may yield inefficient results in training the global model. This is similar to the free-riding attack, but in this scenario clients unintentionally miss participating in the training process due to network issues or any other unexpected roadblocks. The severity of this threat is medium as the probability is lower, and there is an option to opt for aggregation algorithms which can work asynchronously.

4.2.10. Eavesdropping

In FL, we have an iteration of the learning process which involves communication rounds from clients to the central server. Attackers may eavesdrop and extract data through a weak communication channel if one exists. Eavesdropping can be considered a medium severity threat on attacking FL models, since black-box models in general are tough to attack. Attackers would rather takeover a client with weaker security which will readily offer model parameters and white-box global model.

4.2.11. Interplay with data protection laws

This threat has a low possibility of occurrence as a data-scientist configuring the FL environment makes sure that the deployment of the global model is well analyzed before being put into production to all of the clients. The severity of the threat is low, but it is still a considerable threat as intentional or unintentional misconfiguration in FL can lead to a security breach.

4.3. RSQ3 What are the unique security threats to FL in comparison to distributed ML solutions?

Distributed Machine Learning (DML) solutions proposed so far aim to solve challenges of Big Data and computational power while training the ML model. Data and computation power are shared to train a common ML model. A parameter server or multiple server nodes are configured which distribute data or assign a task to client nodes of DML. From an architecture perspective, DML shares a few common properties with FL and there are research works addressing security and privacy concerns in DML. However, FL is unique from existing DML solutions and by default comes with higher security and privacy levels. This section aims to discuss unique threats of FL and common threats shared between FL and DML. This helps to understand existing work on DML [123, 124], and explore reusable/adaptable research ideas from DML to FL. Security and privacy risks specific to DML are out of scope in this paper and are not discussed, we focus only on common risk factors of FL and DML.

Table 1 presents an in-depth summary of the classification of security threats/ attacks. Poisoning threat, Backdoor attacks are for both DML [125] and FL as the data at client nodes can be modified at client nodes which are common in FL and DML architecture. Parameter server [126, 127] of DML and Central server is prone to attacks leading to breach in security. Communication bottleneck threat in DML and FL requires much attention as both the frameworks need to communicate with their respective client nodes.

Table 1: Threats in FL

Threats	Severity	ML Framework	Source of Vulnerability
Poisoning	High	DML/FL	Client Data Manipulations, Compromised Central Server
Inference	High	FL	Client Data Manipulations, Compromised Central Server
Backdoor Attacks	High	DML/FL	Client Data Manipulations
GANs	High	FL	Client Data Manipulations, Compromised Central Server
Malicious Server	High	DML/FL	Compromised Central Server
Communication bottlenecks	High	DML/FL	Weaker Communication bandwidth
Free-riding	Medium	FL	Clients in FL
Unavailability	Medium	FL	Clients in FL
Eavesdropping	Medium	FL	Weaker Communication Protocol
Interplay with data protection laws	Low	FL	Implementer's of FL Environment
System disruption IT downtime	Low	FL	Clients and Centralized Server in FL

Furthermore, Figure 7 visualizes the severity of threats. The severity was calculated based on the probability of adversary taking advantage of the vulnerability and launching a threat and derived from the direct study of current works in the domain by researchers. As can be seen from Figure 7, poisoning attacks have the highest severity. This could be attributed to the fact that the global model can be poisoned from many sources like local model updates, malicious servers, and many others. The higher the possibility of the threat, the higher the impact of the attack on FL. Similar to the poisoning threat, the Inference threat has a high severity since the source of vulnerabilities that it can launch is fairly diverse. The threat of backdoor attacks is of higher severity as well since it is difficult to identify such an attack and the impact has the possibility of destroying the authenticity of the global model. GAN-based attacks also have high severity as well due to their unpredictability and capability

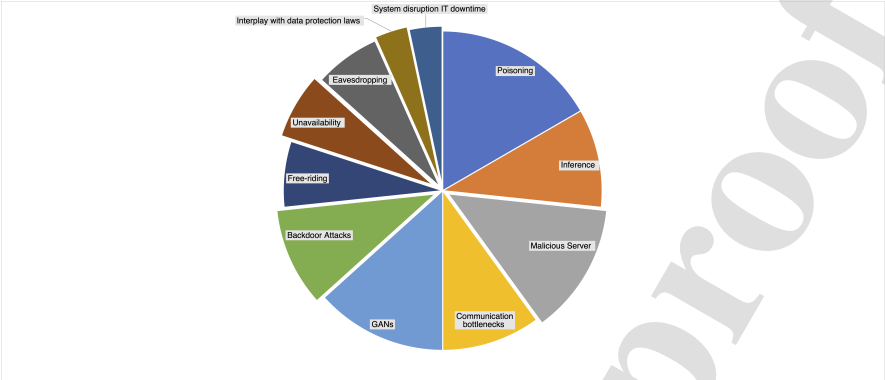


Figure 7: Severity of Threats in FL

to impact the security and privacy of user data. Malicious server threat is of higher severity, due to the open vulnerability of cloud-based/physical servers. System disruption as well as IT downtime is tagged with low severity level as the impact would be less with every client holding the global model individually. Communication bottleneck in FL is a well-researched topic as it can be a showstopper in the learning process. The impact and priority of this threat are always considered high in the FL environment with a huge number of clients posting updates with every iteration of the learning process. The free-riding threat impact is set as medium as this can be possible in fewer scenarios and the impact of it is not severe as the learning process continues with other clients anyway. Unavailability can halt the learning process where the aggregation algorithm is not robust to handle dropouts, the impact of the threat is tagged as the medium as there are algorithms [66], which are designed to handle dropouts. Eavesdropping has medium severity and priority as well. The reason for this is that from the adversary point of view eavesdropping through a communication channel would be the least preferred way to steal information as there is an option to take over a client terminal and gain readily available sets of training data, global model parameters and the global model itself. Interplay with data protection laws can be an initiation loophole which can be created knowing or unknowingly by implementers of FL environment, and the impact stands low as the possibility is less and rectifiable.

4.4. RSQ4: What are the defensive techniques for security vulnerabilities of FL?

Defense techniques help to protect against the known attacks and reduce the probability of risks. There are two classifications of defenses, namely proactive and reactive. Proactive defense is a way to guess the threats and risks associated with and employ a cost-effective defense technique. The reactive defense is later work done when an attack is identified and as part of the mitigation process, where a defense technique is deployed as patch-up in the production

environment. Apart from the defense technique mentioned, there exists additional research work to enhance FL security capability with new add-on technologies/algorithms. Examples of such enhancements are the integration of FL with one more successful technology such as blockchain [128]. Many research works [129, 130, 131] target the combination of FL with one more decentralized applications, mainly blockchain, which provides an immutable ledger to save and persist information. In FL, the use of blockchain technology serves mainly two purposes. One is to provide incentives for major contributors to the global ML model as a motivation to get more contributors from clients. The second purpose is to save global model parameters, weights on a blockchain ledger to ensure the security of the global ML model. In [129, 130], the authors demonstrate how coordination and trust among federated clients are achieved through the use of blockchain technology. As for the incentive mechanism concept where ML training administrative hub encourages clients to proactively contribute to the learning process by giving incentives, there exists ongoing research [132, 133, 134, 135].

Table 2 summarizes the current defense techniques for FL and the types of threats they mitigate.

Table 2: FL defense Techniques

Defenses	Description	Threats	References
Sniper	configure euclidean distance check on global server to exclude adversarial updates	Poisoning	[136]
Knowledge distillation	Transferring knowledge from fully trained model to another model	Eavesdropping Inference GANs	[137]
Anomaly Detection	Monitoring for suspicious updates of clients	Poisoning Trojans Model update poisoning	[138, 139, 140]
Moving target defense	Obfuscating source of vulnerability	Eavesdropping	[141]
Pruning	Reduce the size of Neural network model	Backdoor Attacks Model Computation Communication costs	[109]
Data Sanitization	removing/deleting the data after use	Poisoning Attacks	[142]
Trusted execution Environment	Provides integrity and confidentiality of the code executed on a server	Malicious server	[143, 144]
Fools Gold	Based on the diversity of client updates sybil attacks are identified	sybil-based label flipping backdoor poisoning attacks	[145]
Federated Multi-task Learning	Train models for multiple related tasks simultaneously	Device drop Fault tolerance	[146]

- *Sniper*:

  Different poisoning attacks including data and model poisoning attacks have been investigated in a centralized setting in FL for a long time. However, there are only a few works that explore the poisoning attacks in a distributed environment where multiple malicious participants with the same attack goal aim to incorporate poisoned training samples to the training procedures. Although it seems that distributed poisoning attacks are a bigger threat in FL, the effectiveness of distributed poisoning with multiple attackers is still unclear when compared to the traditional poisoning with a single attacker. Recently, the authors in [136] proposed the Sniper approach where it can recognize legitimate users and decrease

the success rate of poisoning attacks dramatically even when multiple attackers are involved.

- *Knowledge distillation*: It is a variant of the model compression technique, where a fully trained neural network transfers knowledge to a small model step by step on what needs to be done. Knowledge distillation saves computational cost involved in training a model. The concept of sharing the knowledge only instead of model parameters can be leveraged in FL to enhance the security of the client data. The authors in [137] proposed a federated model distillation framework, which provides flexibility to use personalized ML models and uses translators to collect knowledge to be shared with each client.

- *Anomaly detection*: This technique often utilizes statistical and analytical methods in order to identify events that do not conform to an expected pattern or activity. An effective anomaly detection system needs a profile of the normal behavior or events to detect attacks as deviations from the normal behavior profile. In FL environments, different attacks such as data poisoning, model poisoning, or trojans threats can be detected using different anomaly detection techniques. For example, the authors in [139] suggest Auror as a defense against malicious client updates by performing a clustering operation on each client update before the aggregation step. This helps to detect malicious clients' updates. Euclidean Distance is utilized in [138] as the Krum model to detect a deviation in input parameters from each client is used. In a similar work, the authors in [147] discuss identifying abnormal updates from clients in FL. Anomaly detection defense with Autoencoders are proposed in [148, 105] which helps to identify the malicious local model updates. Research work in [148] uses Spectral anomaly detection with variational autoencoders [149, 150, 151]. Furthermore, the research work in [105] proposes LFR (Loss function based rejection) & ERR (Error rate based rejection (ERR)) which are inspired from existing ML defenses such as RONI (reject on negative impact) [152] & TRIM [153] to identify negatively impacting updates from clients.

- *Moving target defense*: The moving target defense concept was introduced in the U.S. National Cyber Leap Year Summit in 2009 [154]. The Federal Cybersecurity Research and Development Program [155] defined moving target defense as a way to deploy diverse mechanisms and strategies that continually change over time in order to increase the cost and complexity for attackers. Moving target defense also increases system resiliency while limits the disclosure of system vulnerabilities and opportunities for attack. Moving target defense is the best of its kind to protect against intrusion at server-level, network-level, and application level. It is a proactive defense architecture built to obscure the vulnerability source from the attackers. defense techniques like IP hopping, pooling of virtual IPs for DNS pings are efficient approaches to moving target defense techniques.

The authors in [141] discuss network-level moving target defense to prevent eavesdropping-based attacks.

- *Federated MultiTask Learning*: Federated Learning provides training ML models collaboratively over a large number of mobile devices considering their local data privacy. This setting can be also extended to a federated multi-task learning environment where multi-task learning drives personalized but shared models among devices. For example, the MOCHA Framework proposed in [146] is designed to speed up the learning process with fault tolerance, making it able to work even in the scenarios of device drop. This framework is proposed to address statistical and system challenges like high communication cost, stragglers and fault tolerance issues in the FL environment. Their experiments show that the framework can handle variability in system heterogeneity and being robust to system drop. Few other works on federated Multi-task learning are given in [156, 157, 158, 159].

- *Trusted Execution Environment (TEE)*: A Trusted Execution Environment (TEE), in general, is defined as a high-level trusted environment for executing code. This technique has been also utilized for privacy-preserving in different ML models where private areas of computing resources are isolated for a particular task [160]. This approach is also applicable in federated learning where we have very limited computing resources. TEE is a tamper-resistant processing environment which provides integrity and confidentiality of the code executed in the secure area of the main processor. Authors in [143] discuss this proactive defense technique. The authors in [144, 161] propose a framework that uses TEE in the federated environment.

- *Data Sanitization*: Sanitizing training data proposed by [142] for the first time is mainly utilized as an anomaly detector in order to filter out training data points that look suspicious. Recent work by [162, 163] aimed to improve data sanitization techniques by utilizing different robust statistics models. In FL environments, the data sanitization technique is one of the common defense techniques against data poisoning attacks, however, work in [164] report that there is a possibility that stronger data poisoning attacks can break data sanitization defense.

- *Foolsgold*: Malicious clients create multiple fake identities and send falsified updates to the central server. This type of attack from compromised clients can break the security and authenticity of the FL environment. Authors in [145] propose a Foolsgold approach that is efficient against Sybil-based, label flipping, and backdoor poisoning attacks.

- *Pruning*: It is a technique in FL that minimizes the size of the ML model to reduce the complexity and improve the accuracy. In FL, clients have relatively low computational power and communication bandwidth. The problem arises when we have large-sized deep neural networks to train

in FL environment as clients usually have a relatively low computational power and communication bandwidth when compared to machines in huge data centers. To address such issues authors in [108, 109] propose a pruning technique. As it is not required to share the full-fledged model in this approach, it helps to address backdoor attacks and communication bottlenecks more efficiently.

One other defense technique proposed for the security of FL that is GAN based is called PDGAN which is proposed in [165] to help defend data poisoning attacks from malicious clients of FL. There has also been a defense technique using a vertical federated learning approach as proposed in [166] known as the Secureboost Framework which is based on gradient boosted tree algorithm.

5. Privacy in Federated Learning

FL promotes privacy by default through the lessening footprint of user data in the network (a central server). Although this privacy-aware machine learning framework (i.e., FL) sounds ideal in theory, it is not immune to attacks nor the current developments in its enabling technology are mature enough to be expected to solve all privacy issues by default, at least not for the time being. Inspired by this fact, this section is dedicated to explore the existing privacy issues and the current relevant achievement in federated learning technology with the hopes to provide more insights for its future development.

Along with the motivation and the stated objectives, this section specifically aims to answer the following privacy-specific research questions[9]:

- **RPQ1**: What are the privacy threats/attacks in FL domain?
- **RPQ2**: What are the techniques to mitigate identified threats in RPQ1 and enhance the general privacy-preserving feature of FL?
- **RPQ3**: What are the unique privacy threats to FL compared to distributed ML solutions?
- **RPQ4**: What is the associated cost with the privacy-preserving techniques identified in RPQ2?

In the following sections, we discuss the results based on each research question obtained from a thorough analysis of all current works in the FL domain.

5.1. RPQ1: What are the privacy threats/attacks in FL domain?

FL aims to guarantee participants' privacy by asking participants to share local training model parameters instead of their actual data. However, according to recent research, FL still has some privacy threats because the adversaries can

[9]Research Privacy Question (RPQ)

partially reveal each participants' training data in the original training dataset based on their uploaded parameter. Such critical threats in FL can be generalized into different categories of inference based attacks

5.1.1. Membership Inference Attacks

As the name denotes, an inference attack is a way to infer training data details. Membership Inference attack aims [167] to get information by checking if the data exists on a training set. The attacker misuses the global model to get information on the training data of the other users. In such cases, the information on the training data set is inferred through guesswork and training the predictive model to predict original training data. The authors in [107] explore the vulnerability of the neural network to memorize their training data which is prone to passive and active inference attacks.

5.1.2. Unintentional Data Leakage & Reconstruction through Inference

Is a scenario where updates or gradients from clients leak unintended information at the central server. The authors in [168] exploit unintentional data leakage vulnerability and successfully reconstruct data of other clients through an inference attack.

Research work in [117] explores how private data from an honest client can be revealed unintentionally using GANs based inference attacks. The advisory client generates data similar to training data using GANs and retrieves sensitive information from other clients in FL. Malicious/curious clients in [169] use the global model and parameters to reconstruct the training data of other clients.

5.1.3. GANs-based Inference Attacks

GANs are generative adversarial networks that have gained much popularity in big data domains in recent years and are also applicable to FL based approaches. Specifically for FL, the authors in [114] propose the mGAN-AI framework for exploring GAN-based attacks on FL. mGAN-AI attacks are experimented on a malicious central server of the FL environment. It explores user-level privacy leakage against the federated learning by the attack from a malicious server. mGAN-AI framework's passive version analyzes all the client inputs and the active version works on isolating a client by sending the global update to only to the isolated instance. The inference attack gains the highest accuracy with mGAN-AI framework because it does not interfere with the training process.

Nevertheless, it is possible to have potential adversaries among FL clients, who may use old local data as their contributions only in exchange for the global model. After they obtained the global model, they may use inference techniques to deduce other client's information. Such behaviors are difficult to be distinguished due to the limited knowledge about clients' profiles and reputation. Furthermore, collaborative training with parameter-only updates also makes the FL server hard to evaluate the effects of each client's contribution.

5.2. RPQ2: What are the techniques to mitigate identified threats in RPQ1 and enhance the general privacy-preserving feature of FL?

This section covers the mitigation strategies for the identified privacy threats. The approach of retaining data at the client device level is the major built-in privacy feature of FL. The state-of-the-art algorithms to enhance privacy-preserving and mitigate threats in FL are mainly based on two categories: Secure Multi-party Computation (SMC) and Differential Privacy (DP).

5.2.1. Secure Multi-party Computation

The concept of Secure Multi-party Computation (SMC), also referred to as MPC, was first introduced to secure the inputs of multi-participant while they jointly compute a model or a function [170]. In SMC, communication is secured and protected with cryptographic methods. Recently, SMC has been utilized to secure updates from clients in the FL framework. Different from the conventional version of SMC, in FL, the computing efficiency is increased immensely since it only needs to encrypt the parameters instead of the large volume of data inputs. This performance feature makes the SMC a preferable choice in the FL environment.

The authors in [171] explore the possibility of information leakage from client updates at the central server. They combine encryption with asynchronous stochastic gradient descent which efficiently prevents data leakage of clients at the central server. Encrypting client updates to ensure that there is no information leakage. It is worth noting that the encryption technique is expensive to use in a larger landscape environment and may impact the efficiency of the ML model.

The work in [172] combines the advantages of two privacy-preserving approaches to mitigate the risk of client data exposure. The main technique was the integration of homomorphic encryption and differential privacy. Focusing on privacy issues in FL environment, this paper suggests to apply client level differential privacy and encrypting the model update. Experiments performed with this approach claim to have high accuracy while ensuring protection against the honest but curious server and other users of FL. The authors in [173] propose a privacy-enhanced FL approach, which provides privacy even after completion of the training process. This approach is similar to the integration of encryption and client level DP proposed in [172].

Therefore, as a promising solution, there are several remaining challenges for SMC based solutions. The main challenge is the trade-off between efficiency and privacy. SMC based solutions need more time expense than typical FL frameworks which may negatively affect the model training. Such a problem will be enhanced in data-freshness aware training tasks since longer training time means data value loss. Besides, how to design a lightweight SMC solution for FL clients is still an open problem.

5.2.2. Differential Privacy

Differential Privacy (DP) is a widely used privacy-preserving technique in industry and academia. The main concept of DP is to preserve privacy by adding

noise to personal sensitive attributes [174]. Therefore, each user's privacy is protected. Meanwhile, the statistic data quality loss caused by the added noise of each user is relatively low compared with the increased privacy protection. In FL, to avoid inverse data retrieval, DP is introduced to add noise to participants' uploaded parameters. DPGAN framework is proposed in [175] that utilizes DP to make GAN based attacks inefficient in inferencing training data of other users in a deep learning network. Similarly, there is DPFedAvgGAN framework [176] for FL specific environments. The authors in [177] explain about DP benefits and definitions of DP properties such as randomization, composition, and exemplifies DP implementation in a multi-agent system, reinforcement learning, transfer learning, and distributed ML. Both works in [178, 172] combine secure multiparty computation and differential privacy to achieve a secured FL model with high accuracy. The authors in [179] improve privacy guarantee of the FL model by combining the shuffling technique with DP and mask user data with an invisibility cloak algorithm. However, such solutions bring uncertainty into the upload parameters and may harm the training performance. Furthermore, these techniques make the FL server more difficult to evaluate the client's behavior to calculate payoff.

5.2.3. VerifyNet

VerifyNet [180] is a privacy preserving and verifiable FL framework. It gets listed as a preferred mitigation strategy to preserve privacy as it provides double-masking protocol which makes it difficult for attackers to infer training data. It provides a way for clients to verify central server results which ensures the reliability of central server. Apart from providing security and privacy enhancements, the VerifyNet framework is robust to handle multiple dropouts. Only issue with this framework is the communication overhead as the central server has to send verifiable proofs with each client.

5.2.4. Adversarial Training

Evasion attacks from an adversarial user aims to fool ML models by injecting adversarial samples into the machine learning models. Examples of adversarial data are projections of imperfection to data in real world. The attacker tries to impact the robustness of the FL model with perturbed data. Adversarial training, which is a proactive defense technique, tries all permutations of an attack from the beginning of the training phase to make the FL global model robust to known adversarial attacks. The authors in [181] discuss how to make the learning model robust to attacks with adversarial training. Evaluation results demonstrate that adversarial training remains vulnerable to black-box attacks. Thus, they further introduce Ensemble Adversarial Training, a technique that augments training data with perturbations. Adversarial Training improves the privacy of the user data as addition of adversarial samples minimizes the threat of revealing actual training data through inference.

Few other defense works proposed for preserving privacy in FL include FEDXGB [182], which proposes defense against user drop-out & reconstruction/leakage of private training data issues of FL. FEDXGB makes use of secret

sharing techniques as well as proposes secure boost protocol for training rounds and a secure predict protocol for secure prediction on the trained global model. There are also a few defense techniques based on GANs such as in [183] called Anti-GAN which helps to prevent inference attacks with the use of WGANs [184, 185] to generate fake training data. This occurs at each client node which makes it difficult to inference actual training data from the global ML model. Similar defense techniques of fake data generation at client node is proposed with FedGP [186] with a slight modification to allow data share instead of model sharing.

5.3. RPQ3: What are the unique privacy threats to FL compared to distributed ML solutions?

FL offers user-data privacy by default resulting in very few privacy threats that are specific to FL. As discussed and experimented in [187], FL performs well in comparison with DML in terms of protecting the privacy of user data. In DML solutions with a parameter server, launching an inference attack (as discussed in Section5.1) to steal information from other clients will be the least preferred approach as the data is readily accessible either on the parameter server or through the updates from clients. However, for DML applications, e.g., [8], where a well-trained ML model is outsourced as a paid service there is a high possibility of inference based attack [188, 189].

GANs based inference attacks (discussed in Section 5.1.3) are feasible in the FL environment but a less suitable approach for DML solutions. This makes GANs based inference attacks specific to only FL. GANs involve two neural networks to launch an attack and DML provides space for attackers to launch more simple and straightforward attacks. GANs are complex and demand more computational resources which makes it a less preferred approach in DML solutions. However, in FL's high-level privacy environment GANs based inference attacks [114] are worth the efforts yielding expected results which wouldn't have been achievable with less complex strategies proposed for DML. Detailed strategies of inference attacks in FL and ML with centralized data solutions are discussed in [107]. The findings from this particular research suggest the possible strategies of passive and active inference attack in stand-alone ML and FL environment. In the FL version, an attacker can be at an aggregator server or a client observing updates to the global model trying to infer data from other clients. And in the standalone version, an attacker utilizes BlackBox inference to retrieve sensitive information.

Few FL proposals [190] include buyers/end-users who only leverage the final product of FL without involving in FL training rounds. For such 3rd party end-users of FL existing research work on the privacy of standalone and distributed ML solutions can be leveraged and adapted in FL for protecting the privacy of the hosted ML model. We recommend readers to sections 5.2 and 5.4 to explore the defenses and costs proposed for add-on privacy of FL, and combine additional privacy protection measures if required. Moreover, the training phase of FL provides much-needed privacy to the user data by skipping most of the

privacy threats observed in DML solutions making the existing work on the privacy of DML less suitable for FL adaption.

5.4. RPQ4: What is the associated cost with the privacy-preserving techniques identified in RPQ2?

Every add-on enhancement comes with its own set of additional costs and implications. The cost here defines an overhead or a consequence incurred due to the enhancement approach implemented. Secure Multi-party Computation and Differential Privacy boost the privacy protection capability of FL, but at the expense of higher cost with respect to accuracy and efficiency. In cryptography-based methods in Secure Multi-party Computation, each client is required to encrypt all the uploaded parameters. Therefore, each client needs to spend additional computational resources to perform the encryption. This could be concerning if the client device is computationally constrained such as is common with IoT devices. Therefore, to enhance the privacy of user-data using encryption, the efficiency of the ML model could be compromised as a trade-off.

In an empirical work [191] related to cost analysis of FL, the authors simulate on Reddit datasets to understand the accuracy of DP-enabled FL global model. Their experimental results show varied results, DP-FL, and nonDPFL environments show almost the same accuracy for the dataset with similar vocabulary size. For experiments with participants who have varying lengths of vocabulary size, DP-FL accuracy is less when compared to nonDPFL. Experiments from of this paper conclude that DP in the heterogeneous environment has a negative impact on accuracy.

DP based methods add noise to the uploaded parameters to enhance privacy protection during the communication and on the server. However, the added noise will perturb the accuracy inevitably. And it may further influence the convergence the global aggregation. That means there is a trade-off between privacy protection strength and accuracy loss as well as efficiency (convergence time). If the FL model preserves more privacy, it loses more accuracy and costs more time to converge. On the contrary, if the FL model needs to preserve a certain degree of accuracy or converge time, it needs to estimate if the privacy protection level is acceptable or not. Table 3 summarizes the privacy-preserving techniques and their associated characteristics discussed in this section.

Table 3: Approaches to enhance privacy preservation in FL

Approach	Cost	Methodology	Ref
Secure Multi-party Computation	Efficiency loss due to encryption	Encrypt uploaded parameters	[170, 173, 192]
Differential Privacy	Accuracy loss due to added noise in client's model	Add random noise to uploaded parameters	[191, 174, 177, 193]
Hybrid	Subdued cost on both efficiency and accuracy	Encrypt the manipulated parameter	[194, 178, 172]
VerifyNet	Communication overhead	Double-masking protocol Verifiable aggregation results	[180]
Adversarial Training	Computation power, training time for adversarial samples	Include adversarial samples in training data	[181, 195, 196]

6. Other Literature Review Work

There have been a handful of survey-like studies proposed in the literature, which aim to explore aspects limited to challenges/problems, architecture style of FL for different domains/use-cases of Federated Learning by providing collective insights and views.

Authors in [197] focus on four challenges in FL which are expensive communication, systems heterogeneity, statistical heterogeneity, and privacy concerns. The authors suggest ways to deal with each challenge but lack in-depth categorization of possible breaks in the integrity of FL environment. The authors in [198] categorize existing FL models and provides a summary of each category. It emphasizes building a robust FL environment by evaluating the issues around FL from the system perspective. A taxonomy of FL is defined based on data privacy levels, machine model, data partition across domains and basic building blocks of FL. The focus of the paper is limited to categorizing FL from different perspectives with a brief overview of privacy. Another survey work in [199] gives a detailed architecture and implementation of FL in horizontal, vertical, and transfer learning approaches with relevant uses cases. This work gives a great summary of definitions and information on existing research works in the FL space with details on implementing encryption techniques utilized in the vertical federated learning approach.However, the focus stays on implementation details of FL in different scenarios and lack insights into possible risks in FL.

There are few papers focusing on domain-specific areas related to FL, which aim only to introduce FL in different real-world domains with possible useful use-case scenarios. One such work is a comprehensive survey on FL in mobile devices presented in [200] which gives insights of FL with mobile edge networks and issues such as communication cost, privacy & security of data, and resource allocation in mobile IoT devices specific to FL implementation. The authors of this survey provide a comparison of FL implementation using different approaches. Few other works suggest importance-based updating [201], model compression [202, 203] in IoT specific implementation of FL. The strategies for FL clients resource allocation and participation selection are discussed to mitigate communication bottlenecks and communication costs incurring in a larger landscape IoT-FL environment. FL in the autonomous vehicle gives feasibility for the ML model to learn from other autonomous cars configured in the FL environment instead of relying on a central database to increase the efficiency of the decision-making skills of AI. Research work in [204] focuses on vehicular IoT devices in FL and provides an examination of recent achievements and challenges in this field.

Open problems in FL are extensively discussed in paper [36] published by Google. The authors start with summarizing various works, approaches, and definitions of FL and summarize open challenges/issues with interesting suggestions to researchers for further enhancing the efficiency of FL. A few of the suggestions are to implement neural architectural search (NAS) [205] in FL to address redundancy in neural architecture for few clients which may lead to ad-

ditional computation. FL specific hyperparameter optimization [206] to address communication efficiency and cost in each training round as well as federated fairness to avoid bias [207] in FL are other suggested areas of future research. Although open problems of FL are explored in detail in this paper, the authors do not discuss the privacy and security risks of FL in depth as we did in this work.

In summary, the current state-of-the-art research surveys and reviews in the field provide remarkable work from various perspectives of FL with different goals and concentration. Despite their importance, the examination of security and privacy of FL has not been sufficiently addressed in the literature, resulting in a missing piece of work with an organized review of studies to dive deep into the cybersecurity aspect of the FL. Our research work endeavored to provide this focused research to fill the gap in an attempt to help the community and newcomers with in-depth information and knowledge of FL security and privacy.

7. Future Directions in FL Security and Privacy

Federated learning has a set of challenges that need further research. Based on our observations and results, we have identified the following issues that could form future avenues of research.

7.1. Zero-day adversarial attacks and their supporting techniques

Current defense efforts in FL are designed to protect against known vulnerabilities and specific predefined malicious activities, making them less useful in detecting attacks outside their design parameters when tested. Although, this phenomenon applies to virtually any ML application's defense mechanisms, the probability is more in FL as we do not have many versions in production that would have demonstrated the possibility of various attacks. Current achievements using advanced deep learning have shown promising solutions in combating such attacks [208, 209].

7.2. Trusted Traceability

A major challenge of FL is traceability of the global ML model throughout the lifecycle of the underlying ML process. For instance, if a prediction value is changed in the global ML model we will need to have backward tracking ability to identify which clients aggregation values resulted in that change. If the logic behind ML model behavior is a black-box, then we are forced to lose grip to logical reality and blindly rely on human-made AI. There are a few preliminary works leveraging blockchain technology [210, 211, 212] with FL to provide and trace transaction updates to the global ML model [129, 130, 131], hoping to achieve a more transparent tracing of the training process in deep learning ML models.

7.3. Well-Defined Process with APIs

FL is a fairly new approach that requires a detailed analysis of all the pros and cons tagged with different approaches. Standardized techniques need to be defined to support the emerging requirements of FL in different domains. As privacy is a key factor in FL further research needs to be done focusing on enhancing privacy and standardizing approaches for each requirement and define a process (with generic APIs) to implement such enhanced approaches.

7.4. Optimize trade-off between Privacy Protection Enhancement and Cost

Current research work shows how to enhance privacy protection in FL at the expense of sacrificing efficiency or accuracy. However, there are no research works on finding the proper encryption level for SMC and the quantity of added noise. If the encryption level or the quantity of noise is not enough, the participants still suffer from the risk of privacy leakage. On the contrary, if the encryption level is too high or too much noise added to the parameters, the FL model severally suffers from low accuracy.

7.5. Build FL Privacy Protection Enhanced Frameworks in Practice

There are currently some FL frameworks that can be utilized to implement FL-based systems such as TensorFlow Federated, PySyft, and FATE (as discussed in Section 3.5). Apart from PySyft, there are no frameworks, libraries, or toolboxes that can integrate and execute SMC or DP at the moment. Thus, developing FL Privacy Protection Enhanced Frameworks could be a pressing research direction that can benefit both academic research and FL adoption in the industry.

7.6. Client selection and Training plan in FL

Training plan and strategy for client selection for training rounds are crucial in FL. Research work in [213] suggests optimal approaches, but still there is a need to have a standardized approach for each ML algorithm use-case in FL.

7.7. Optimization techniques for different ML Algorithms

Based on different ML algorithms, there is a need to have pre-defined and standardized optimizing algorithms to build the FL model. There are many proposed aggregation/optimizing algorithms (as discussed in Section 3.4) suggesting to optimize or enhance FL but still there is a need to have dedicated research to provide FL specific optimizing algorithms for all the current ML applications/use-case. This helps future implementers/adaptors of FL to develop FL specific solutions with ease.

7.8. Vision on training strategies and parameters

Research work in [214] proposes an optimal strategy that helps the central server to set an optimal trigger point to stop/restart the training rounds. Similar research work needs to be done with respect to different models & domains of ML applications which can help in understanding FL specific hyperparameters and possible trigger conditions to configure in FL training rounds. As FL training rounds are time, cost and computational consuming having vision on setting optimal values will help in establishing robust and cost-efficient FL solutions.

7.9. Ease in Migrating and Productionising

It is noticeable that there is no simple and straight forward approach to productionise FL environment. Research work in [215] proposes many factors to be considered while moving to production, but still, there is a need for well-established guidelines for implementing a new use-case in FL or migrating an existing ML environment to decentralized FL approach.

8. Conclusion

Federated learning is a new technology that advocates on-device AI through decentralized learning. FL was proposed to extend machine learning benefits to domains with sensitive data. In this paper, we provide a comprehensive study on the security and privacy achievements, issues, and impacts in the FL environment. With the evaluation and results on security & privacy, we are hoping to give new perspectives and bring the community's attention towards building risk-free FL environments, suited for mass adoption. Through the future directions section, we outline the areas in FL which require in-depth research and investigation. FL is relatively a newly launched framework in the market which needs further research to identify the suitable enhancement top-ups which fit different FL environment styles.

References

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), 2017.

[2] M. Aledhari, R. Razzak, R. M. Parizi, F. Saeed, Federated learning: A survey on enabling technologies, protocols, and applications, IEEE Access 8 (2020) 140699–140725.

[3] White house report. consumer data privacy in a net- worked world: A framework for protecting privacy and promoting innovation in the global digital economy. journal of privacy and confidentiality, 2013 (2013).

[4] General data protection regulation, web.
URL https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679

[5] A. Narayanan, V. Shmatikov, Robust de-anonymization of large sparse datasets, in: 2008 IEEE Symposium on Security and Privacy (sp 2008), 2008, pp. 111–125. doi:10.1109/SP.2008.33.

[6] J. X. Chen, The evolution of computing: Alphago, Computing in Science & Engineering 18 (4) (2016) 4.

[7] State of art papers in ml, web.
URL https://paperswithcode.com/sota

[8] M. Ribeiro, K. Grolinger, M. A. M. Capretz, Mlaas: Machine learning as a service, in: 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), 2015, pp. 896–902. doi:10.1109/ICMLA.2015.152.

[9] L. Caviglione, W. Mazurczyk, S. Wendzel, S. Zander, Emerging and unconventional: New attacks and innovative detection techniques, Security and Communication Networks 2018.

[10] R. Keidel, S. Wendzel, S. Zillien, E. S. Conner, G. Haas, Wodicof-a testbed for the evaluation of (parallel) covert channel detection algorithms., J. UCS 24 (5) (2018) 556–576.

[11] L. Caviglione, W. Mazurczyk, S. Wendzel, S. Zander, Emerging and unconventional: New attacks and innovative detection techniques, Security and Communication Networks 2018.

[12] K. Cabaj, M. Gregorczyk, W. Mazurczyk, Software-defined networking-based crypto ransomware detection using http traffic characteristics, Computers & Electrical Engineering 66 (2018) 353–368.

[13] Z. Lv, W. Mazurczyk, S. Wendzel, H. Song, Recent advances in cyber-physical security in industrial environments, IEEE Transactions on Industrial Informatics.

[14] K. Cabaj, L. Caviglione, W. Mazurczyk, S. Wendzel, A. Woodward, S. Zander, The new threats of information hiding: The road ahead, IT Professional 20 (3) (2018) 31–39.

[15] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, D. Ramage, Federated learning for mobile keyboard prediction (2018). arXiv:1811.03604.

[16] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, F. Beaufays, Applied federated learning: Improving google keyboard query suggestions (2018). arXiv:1812.02903.

[17] F. S. Beaufays, M. Chen, R. Mathews, T. Ouyang, Federated learning of out-of-vocabulary words (2019).
URL https://arxiv.org/abs/1903.10635

[18] S. Ramaswamy, R. Mathews, K. Rao, F. Beaufays, Federated learning for emoji prediction in a mobile keyboard (2019). arXiv:1906.04329.

[19] D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, J. Dureau, Federated learning for keyword spotting, ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)doi:10.1109/icassp.2019.8683546.
URL http://dx.doi.org/10.1109/ICASSP.2019.8683546

[20] Nvidia clara federated learning to deliver ai to hospitals while protecting patient data, web.
URL https://blogs.nvidia.com/blog/2019/12/01/clara-federated-learning/

[21] S. Niknam, H. S. Dhillon, J. H. Reed, Federated learning for wireless communications: Motivation, opportunities and challenges (2019). arXiv:1908.06847.

[22] M. Chen, H. V. Poor, W. Saad, S. Cui, Wireless communications for collaborative federated learning in the internet of things, ArXiv abs/2006.02499.

[23] K. Lin, W. Huang, Using federated learning on malware classification, in: 2020 22nd International Conference on Advanced Communication Technology (ICACT), 2020, pp. 585–589.

[24] K. Sozinov, V. Vlassov, S. Girdzijauskas, Human activity recognition using federated learning, in: 2018 IEEE Intl Conf on Parallel Distributed Processing with Applications, Ubiquitous Computing Communications, Big Data Cloud Computing, Social Computing Networking, Sustainable Computing Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom), 2018, pp. 1103–1111.

[25] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, A. Sadeghi, DÏot: A federated self-learning anomaly detection system for iot, in: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), 2019, pp. 756–767.

[26] B. Cetin, A. Lazar, J. Kim, A. Sim, K. Wu, Federated wireless network intrusion detection, in: 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 6004–6006.

[27] Y. Lu, X. Huang, K. Zhang, S. Maharjan, Y. Zhang, Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles, IEEE Transactions on Vehicular Technology 69 (4) (2020) 4298–4311.

[28] S. Samarakoon, M. Bennis, W. Saad, M. Debbah, Federated learning for ultra-reliable low-latency v2v communications, in: 2018 IEEE Global Communications Conference (GLOBECOM), 2018, pp. 1–7.

[29] Y. Lu, X. Huang, Y. Dai, S. Maharjan, Y. Zhang, Federated learning for data privacy preservation in vehicular cyber-physical systems, IEEE Network 34 (3) (2020) 50–56.

[30] Y. Liu, J. J. Q. Yu, J. Kang, D. Niyato, S. Zhang, Privacy-preserving traffic flow prediction: A federated learning approach, IEEE Internet of Things Journal (2020) 1–1.

[31] N. I. Mowla, N. H. Tran, I. Doh, K. Chae, Federated learning-based cognitive detection of jamming attack in flying ad-hoc network, IEEE Access 8 (2020) 4338–4350.

[32] Y. Liu, A. Huang, Y. Luo, H. Huang, Y. Liu, Y.-Y. Chen, L. Feng, T. Chen, H. Yu, Q. Yang, Fedvision: An online visual object detection platform powered by federated learning, in: AAAI, 2020.

[33] W. Schneble, G. Thamilarasu, Attack detection using federated learning in medical cyber-physical systems, in: 2019 28th International Conference on Computer Communication and Networks (ICCCN), 2019, pp. 1–8. doi:10.1109/ICCCN.2019.8847161.

[34] S. Lu, Y. Zhang, Y. Wang, Decentralized federated learning for electronic health records, in: 2020 54th Annual Conference on Information Sciences and Systems (CISS), 2020, pp. 1–5.

[35] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, J. Liu, Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent, in: Advances in Neural Information Processing Systems, 2017, pp. 5330–5340.

[36] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al., Advances and open problems in federated learning, arXiv preprint arXiv:1912.04977.

[37] Y. Mansour, M. Mohri, J. Ro, A. T. Suresh, Three approaches for personalization with applications to federated learning (2020). arXiv:2002.10619.

[38] F. Sattler, K.-R. Muller, W. Samek, Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints, IEEE Transactions on Neural Networks and Learning Systems (2020) 1–13doi:10.1109/tnnls.2020.3015958.
URL http://dx.doi.org/10.1109/TNNLS.2020.3015958

[39] M. Xie, G. Long, T. Shen, T. Zhou, X. Wang, J. Jiang, Multi-center federated learning (2020). arXiv:2005.01026.

[40] A. Ghosh, J. Chung, D. Yin, K. Ramchandran, An efficient framework for clustered federated learning (2020). arXiv:2006.04088.

[41] P. Vanhaesebrouck, A. Bellet, M. Tommasi, Decentralized collaborative learning of personalized models over networks (2016). arXiv:1610.05202.

[42] L. MuÃ±oz-GonzÃ¡lez, K. T. Co, E. C. Lupu, Byzantine-robust federated machine learning through adaptive model averaging (2019). arXiv:1909.05125.

[43] Z. Jiang, A. Balu, C. Hegde, S. Sarkar, Collaborative deep learning in fixed topology networks (2017). arXiv:1706.07880.

[44] J. Daily, A. Vishnu, C. Siegel, T. Warfel, V. Amatya, Gossipgrad: Scalable deep learning using gossip communication based asynchronous gradient descent (2018). arXiv:1803.05880.

[45] J. Wang, A. K. Sahu, Z. Yang, G. Joshi, S. Kar, Matcha: Speeding up decentralized sgd via matching decomposition sampling (2019). arXiv:1905.09435.

[46] A. Lalitha, O. C. Kilinc, T. Javidi, F. Koushanfar, Peer-to-peer federated learning on graphs (2019). arXiv:1901.11173.

[47] H. B. McMahan, E. Moore, D. Ramage, B. A. y Arcas, Federated learning of deep networks using model averaging.

[48] S. Yang, B. Ren, X. Zhou, L. Liu, Parallel distributed logistic regression for vertical federated learning without third-party coordinator, arXiv preprint arXiv:1911.09824.

[49] Y. Chen, J. Wang, C. Yu, W. Gao, X. Qin, Fedhealth: A federated transfer learning framework for wearable healthcare, CoRR abs/1907.09173. arXiv:1907.09173.
URL http://arxiv.org/abs/1907.09173

[50] Y. Liu, T. Chen, Q. Yang, Secure federated transfer learning (2018). arXiv:1812.03337.

[51] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on knowledge and data engineering 22 (10) (2009) 1345–1359.

[52] H. Yang, H. He, W. Zhang, X. Cao, Fedsteg: A federated transfer learning framework for secure image steganalysis, IEEE Transactions on Network Science and Engineering (2020) 1–1.

[53] Y. Liu, Y. Kang, C. Xing, T. Chen, Q. Yang, A secure federated transfer learning framework, IEEE Intelligent Systems (2020) 1–1.

[54] C. Nadiger, A. Kumar, S. Abdelhak, Federated reinforcement learning for fast personalization, in: 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), 2019, pp. 123–127.

[55] H. Lim, J. Kim, C. Kim, G. Hwang, H. Choi, Y. Han, Federated reinforcement learning for controlling multiple rotary inverted pendulums in edge computing environments, in: 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), 2020, pp. 463–464.

[56] B. Liu, L. Wang, M. Liu, Lifelong federated reinforcement learning: A learning architecture for navigation in cloud robotic systems, IEEE Robotics and Automation Letters 4 (4) (2019) 4555–4562.

[57] Fate framework from webank, web.
URL https://fate.fedai.org

[58] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, Y. Liu, Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning.

[59] D. Alistarh, D. Grubic, J. Li, R. Tomioka, M. Vojnovic, Qsgd: Communication-efficient sgd via gradient quantization and encoding, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 1709–1720.

[60] Y. Feng, X. Yang, W. Fang, S.-T. Xia, X. Tang, Practical and bilateral privacy-preserving federated learning (2020). arXiv:2002.09843.

[61] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, S. Cui, Performance optimization of federated learning over wireless networks, in: 2019 IEEE Global Communications Conference (GLOBECOM), 2019, pp. 1–6.

[62] Z. Tao, Q. Li, esgd: Communication efficient distributed deep learning on the edge, in: USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18), USENIX Association, Boston, MA, 2018.
URL https://www.usenix.org/conference/hotedge18/presentation/tao

[63] W. Shi, S. Zhou, Z. Niu, Device scheduling with fast convergence for wireless federated learning (2019). arXiv:1911.00856.

[64] Y. Sarikaya, O. Ercetin, Motivating workers in federated learning: A stackelberg game perspective, IEEE Networking Lettersdoi:10.1109/lnet.2019.2947144.
URL http://dx.doi.org/10.1109/lnet.2019.2947144

[65] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, M. Jirstrand, A performance evaluation of federated learning algorithms, in: Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning, DIDL '18, ACM, New York, NY, USA, 2018, pp. 1–8. doi:10.1145/3286490.3286559.
URL http://doi.acm.org/10.1145/3286490.3286559

[66] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, K. Seth, Practical secure aggregation for federated learning on user-held data (2016). arXiv:1611.04482.

[67] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks (2018). arXiv:1812.06127.

[68] Anonymous, Federated learning with matched averaging, in: Submitted to International Conference on Learning Representations, 2020, under review.
URL https://openreview.net/forum?id=BkluqlSFDS

[69] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, T. N. Hoang, Y. Khazaeni, Bayesian nonparametric federated learning of neural networks (2019). arXiv:1905.12022.

[70] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, A. T. Suresh, Scaffold: Stochastic controlled averaging for federated learning (2019). arXiv:1910.06378.

[71] Y. Kim, J. Sun, H. Yu, X. Jiang, Federated tensor factorization for computational phenotyping, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 887–895. doi:10.1145/3097983.3098118.
URL https://doi.org/10.1145/3097983.3098118

[72] J. Ma, Q. Zhang, J. Lou, J. C. Ho, L. Xiong, X. Jiang, Privacy-preserving tensor factorization for collaborative health data analysis, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 1291–1300.

[73] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, S. Choudhary, Federated learning with personalization layers (2019). arXiv:1912.00818.

[74] Y. Deng, M. M. Kamani, M. Mahdavi, Adaptive personalized federated learning (2020). arXiv:2003.13461.

[75] Y. Liu, Y. Kang, X. Zhang, L. Li, Y. Cheng, T. Chen, M. Hong, Q. Yang, A communication efficient collaborative learning framework for distributed features (2019). arXiv:1912.11187.

[76] J. Jiang, S. Ji, G. Long, Decentralized knowledge acquisition for mobile internet applications, World Wide Webdoi:10.1007/s11280-019-00775-w.

[77] Y. Chen, X. Sun, Y. Jin, Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation, IEEE Transactions on Neural Networks and Learning Systems (2019) 1–10.

[78] G. D. A. Chaum, David, Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms, Springer US, Boston, MA, 2003, pp. 211–219. doi:10.1007/978-1-4615-0239-5_14.
URL https://doi.org/10.1007/978-1-4615-0239-5_14

[79] Y. Wang, Co-op: Cooperative machine learning from mobile devices.

[80] T. Nishio, R. Yonetani, Client selection for federated learning with heterogeneous resources in mobile edge, ICC 2019 - 2019 IEEE International Conference on Communications (ICC)doi:10.1109/icc.2019.8761315.
URL http://dx.doi.org/10.1109/ICC.2019.8761315

[81] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, S.-L. Kim, Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data (2018). arXiv:1811.11479.

[82] Tff:open source for federated learning from google, web.
URL https://www.tensorflow.org/federated/federated_learning

[83] G. Sannino, G. De Pietro, A deep learning approach for ecg-based heartbeat classification for arrhythmia detection, Future Generation Computer Systems 86 (2018) 446–455.

[84] C. Wang, Z. Zhao, L. Gong, L. Zhu, Z. Liu, X. Cheng, A distributed anomaly detection system for in-vehicle network using htm, IEEE Access 6 (2018) 9091–9098.

[85] S. Lou, G. Srivastava, S. Liu, A node density control learning method for the internet of things, Sensors 19 (15) (2019) 3428.

[86] S. Liu, W. Bai, G. Srivastava, J. T. Machado, Property of self-similarity between baseband and modulated signals, Mobile Networks and Applications (2019) 1–11.

[87] Tensorflow federated with google kubernetes engine, web.
URL https://github.com/tensorflow/federated/tree/master/docs/tutorials

[88] grpcs- remote procedure callsl, web.
URL https://grpc.io

[89] S. Caldas, P. Wu, T. Li, J. Konecný, H. B. McMahan, V. Smith, A. Talwalkar, LEAF: A benchmark for federated settings, CoRR abs/1812.01097. arXiv:1812.01097.
URL http://arxiv.org/abs/1812.01097

[90] Pysyft: Open source framework for federated learning, web.
URL https://github.com/OpenMined/PySyft

[91] T. Ryffel, A. Trask, M. Dahl, B. Wagner, J. Mancuso, D. Rueckert, J. Passerat-Palmbach, A generic framework for privacy preserving deep learning (2018). arXiv:1811.04017.

[92] Z. Sun, P. Kairouz, A. T. Suresh, H. B. McMahan, Can you really backdoor federated learning? (2019). arXiv:1911.07963.

[93] The clara training framework, web.
URL https://developer.nvidia.com/clara

[94] Paddlefl, web.
URL https://github.com/PaddlePaddle/PaddleFL

[95] Uberhorovod, web.
URL https://eng.uber.com/horovod/

[96] G. Ulm, E. Gustavsson, M. Jirstrand, Functional federated learning in erlang (ffl-erl), in: J. Silva (Ed.), Functional and Constraint Logic Programming, Springer International Publishing, Cham, 2019, pp. 162–178.

[97] Federatd learning with crypten, web.
URL https://crypten.ai

[98] Owsap defination for vulnerability, web.
URL https://www.owasp.org/index.php/Category:Vulnerability

[99] J. Men, G. Xu, Z. Han, Z. Sun, X. Zhou, W. Lian, X. Cheng, Finding sands in the eyes: vulnerabilities discovery in iot with eufuzzer on human machine interface, IEEE Access 7 (2019) 103751–103759.

[100] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, V. Shmatikov, How to backdoor federated learning, in: International Conference on Artificial Intelligence and Statistics, 2020, pp. 2938–2948.

[101] J. Feng, Q.-Z. Cai, Z.-H. Zhou, Learning to confuse: Generating training time adversarial data with auto-encoder (2019). arXiv:1905.09027.

[102] L. Muñoz-González, B. Biggio, A. Demontis, A. Paudice, V. Wongrassamee, E. C. Lupu, F. Roli, Towards poisoning of deep learning algorithms with back-gradient optimization, Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security - AISec '17doi:10.1145/3128572.3140451.
URL http://dx.doi.org/10.1145/3128572.3140451

[103] B. Biggio, B. Nelson, P. Laskov, Poisoning attacks against support vector machines, arXiv preprint arXiv:1206.6389.

[104] A. N. Bhagoji, S. Chakraborty, P. Mittal, S. Calo, Analyzing federated learning through an adversarial lens, in: International Conference on Machine Learning, 2019, pp. 634–643.

[105] M. Fang, X. Cao, J. Jia, N. Z. Gong, Local model poisoning attacks to byzantine-robust federated learning, arXiv preprint arXiv:1911.11815.

[106] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, T. Goldstein, Poison frogs! targeted clean-label poisoning attacks on neural networks, in: Advances in Neural Information Processing Systems, 2018, pp. 6103–6113.

[107] M. Nasr, R. Shokri, A. Houmansadr, Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning, in: 2019 IEEE Symposium on Security and Privacy (SP), 2019, pp. 739–753.

[108] K. Liu, B. Dolan-Gavitt, S. Garg, Fine-pruning: Defending against backdooring attacks on deep neural networks, in: International Symposium on Research in Attacks, Intrusions, and Defenses, Springer, 2018, pp. 273–294.

[109] Y. Jiang, S. Wang, B. J. Ko, W.-H. Lee, L. Tassiulas, Model pruning enables efficient federated learning on edge devices (2019). `arXiv:1909.12326`.

[110] C. Xie, K. Huang, P.-Y. Chen, B. Li, Dba: Distributed backdoor attacks against federated learning, in: International Conference on Learning Representations, 2019.

[111] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, X. Zhang, Trojaning attack on neural networks.

[112] M. Zou, Y. Shi, C. Wang, F. Li, W. Song, Y. Wang, Potrojan: powerful neural-level trojan designs in deep learning models, arXiv preprint arXiv:1802.03043.

[113] A. Koloskova, S. U. Stich, M. Jaggi, Decentralized stochastic optimization and gossip algorithms with compressed communication, arXiv preprint arXiv:1902.00340.

[114] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, H. Qi, Beyond inferring class representatives: User-level privacy leakage from federated learning, in: IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, 2019, pp. 2512–2520.

[115] J. Zhang, J. Chen, D. Wu, B. Chen, S. Yu, Poisoning attack in federated learning using generative adversarial nets, in: 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), 2019, pp. 374–380.

[116] Generative adversarial networks, web.
URL https://developers.google.com/machine-learning/gan

[117] B. Hitaj, G. Ateniese, F. Perez-Cruz, Deep models under the gan: Information leakage from collaborative deep learning, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS '17, ACM, New York, NY, USA, 2017, pp. 603–618.

[118] J. Kone, H. B. McMahan, F. X. Yu, P. Richtrik, A. T. Suresh, D. Bacon, Federated learning: Strategies for improving communication efficiency (2016). arXiv:1610.05492.

[119] L. WANG, W. WANG, B. LI, Cmfl: Mitigating communication overhead for federated learning, in: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), 2019, pp. 954–964. doi:10.1109/ICDCS.2019.00099.

[120] X. Yao, C. Huang, L. Sun, Two-stream federated learning: Reduce the communication costs, in: 2018 IEEE Visual Communications and Image Processing (VCIP), 2018, pp. 1–4.

[121] J. Lin, M. Du, J. Liu, Free-riders in federated learning: Attacks and defenses (2019). arXiv:1911.12560.

[122] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, H. Chen, Deep autoencoding gaussian mixture model for unsupervised anomaly detection, in: International Conference on Learning Representations, 2018.
URL https://openreview.net/forum?id=BJJLHbb0-

[123] R. Zhang, Q. Zhu, Security of distributed machine learning: A game-theoretic approach to design secure dsvm (03 2020).

[124] R. Zhang, Q. Zhu, A game-theoretic approach to design secure and resilient distributed support vector machines, IEEE Transactions on Neural Networks and Learning Systems 29 (11) (2018) 5512–5527.

[125] Y. Chen, Y. Mao, H. Liang, S. Yu, Y. Wei, S. Leng, Data poison detection schemes for distributed machine learning, IEEE Access 8 (2020) 7442–7454.

[126] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, B.-Y. Su, Scaling distributed machine learning with the parameter server, in: 11th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 14), 2014, pp. 583–598.

[127] M. Li, L. Zhou, Z. Yang, A. Li, F. Xia, D. G. Andersen, A. Smola, Parameter server for distributed machine learning.

[128] P. J. Taylor, T. Dargahi, A. Dehghantanha, R. M. Parizi, K.-K. R. Choo, A systematic literature review of blockchain cyber security, Digital Communications and Networks 6 (2) (2020) 147 – 156.

[129] H. Kim, J. Park, M. Bennis, S. Kim, Blockchained on-device federated learning, IEEE Communications Letters 24 (6) (2020) 1279–1283.

[130] U. Majeed, C. S. Hong, Flchain: Federated learning via mec-enabled blockchain network, in: 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), 2019, pp. 1–4. doi:10.23919/APNOMS.2019.8892848.

[131] K. Salah, M. H. U. Rehman, N. Nizamuddin, A. Al-Fuqaha, Blockchain for ai: Review and open research challenges, IEEE Access 7 (2019) 10127–10149. doi:10.1109/ACCESS.2018.2890507.

[132] Y. Zhao, J. Zhao, L. Jiang, R. Tan, D. Niyato, Mobile edge computing, blockchain and reputation-based crowdsourcing iot federated learning: A secure, decentralized and privacy-preserving system (2019). arXiv:1906.10893.

[133] L. U. Khan, N. H. Tran, S. R. Pandey, W. Saad, Z. Han, M. N. H. Nguyen, C. S. Hong, Federated learning for edge networks: Resource optimization and incentive mechanism (2019). arXiv:1911.05642.

[134] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, W. Luo, Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive, IEEE Transactions on Dependable and Secure Computing (2019) 1–1doi:10.1109/TDSC.2019.2952332.

[135] J. Kang, Z. Xiong, D. Niyato, S. Xie, J. Zhang, Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory, IEEE Internet of Things Journal.

[136] D. Cao, S. Chang, Z. Lin, G. Liu, D. Sun, Understanding distributed poisoning attack in federated learning, in: 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS), 2019, pp. 233–239.

[137] D. Li, J. Wang, Fedmd: Heterogenous federated learning via model distillation (2019). arXiv:1910.03581.

[138] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, J. Stainer, Machine learning with adversaries: Byzantine tolerant gradient descent, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 119–129.

[139] S. Shen, S. Tople, P. Saxena, A uror: defending against poisoning attacks in collaborative deep learning systems, in: A uror: defending against poisoning attacks in collaborative deep learning systems, 2016, pp. 508–519. doi:10.1145/2991079.2991125.

[140] R. Ito, M. Tsukada, H. Matsutani, An on-device federated learning approach for cooperative anomaly detection, arXiv preprint arXiv:2002.12301.

[141] R. Colbaugh, K. Glass, Moving target defense for adaptive adversaries, in: 2013 IEEE International Conference on Intelligence and Security Informatics, 2013, pp. 50–55. doi:10.1109/ISI.2013.6578785.

[142] G. F. Cretu-Ciocarlie, A. Stavrou, M. E. Locasto, S. J. Stolfo, A. D. Keromytis, Casting out demons: Sanitizing training data for anomaly sensors, 2008 IEEE Symposium on Security and Privacy (sp 2008) (2008) 81–95.

[143] M. Sabt, M. Achemlal, A. Bouabdallah, Trusted execution environment: What it is, and what it is not, in: 2015 IEEE Trustcom/BigDataSE/ISPA, Vol. 1, 2015, pp. 57–64. doi:10.1109/Trustcom.2015.357.

[144] F. Mo, H. Haddadi, Efficient and private federated learning using tee.

[145] C. Fung, C. J. M. Yoon, I. Beschastnikh, Mitigating sybils in federated learning poisoning, CoRR abs/1808.04866. arXiv:1808.04866.
URL http://arxiv.org/abs/1808.04866

[146] V. Smith, C.-K. Chiang, M. Sanjabi, A. S. Talwalkar, Federated multi-task learning, in: Advances in Neural Information Processing Systems, 2017, pp. 4424–4434.

[147] S. Li, Y. Cheng, Y. Liu, W. Wang, T. Chen, Abnormal client behavior detection in federated learning (2019). arXiv:1910.09933.

[148] S. Li, Y. Cheng, W. Wang, Y. Liu, T. Chen, Learning to detect malicious clients for robust federated learning (2020). arXiv:2002.00211.

[149] D. P. Kingma, M. Welling, An introduction to variational autoencoders, Foundations and Trends® in Machine Learning 12 (4) (2019) 307–392. doi:10.1561/2200000056.
URL http://dx.doi.org/10.1561/2200000056

[150] J. An, S. Cho, Variational autoencoder based anomaly detection using reconstruction probability, 2015.

[151] T. Kieu, B. Yang, C. Guo, C. S. Jensen, Outlier detection for time series with recurrent autoencoder ensembles., in: IJCAI, 2019, pp. 2725–2732.

[152] M. Barreno, B. Nelson, A. D. Joseph, J. D. Tygar, The security of machine learning, Machine Learning 81 (2) (2010) 121–148.

[153] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, B. Li, Manipulating machine learning: Poisoning attacks and countermeasures for regression learning, in: 2018 IEEE Symposium on Security and Privacy (SP), IEEE, 2018, pp. 19–35.

[154] U. national cyber, National cyber leap year summit 2009 cochairs report (2009 (accessed Jun 21, 2020)).
URL https://www.nitrd.gov/nitrdgroups/index.php?title=File:National_Cyber_Leap_Year_Summit_2009_CoChairs_Report.pdf

[155] F. C. Research, D. Program, Nitrd csia iwg cybersecurity game-change research & development recommendations (2014 (accessed Jun 21, 2020)).
URL https://www.nitrd.gov/pubs/CSIA_IWG_%20Cybersecurity_%20GameChange_RD_%20Recommendations_20100513.pdf

[156] R. Li, F. Ma, W. Jiang, J. Gao, Federated multitask learning, in: 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 215–220.

[157] T. Yu, T. Li, Y. Sun, S. Nanda, V. Smith, V. Sekar, S. Seshan, Learning context-aware policies from multiple smart homes via federated multi-task learning, in: 2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI), 2020, pp. 104–115.

[158] S. Caldas, V. Smith, A. Talwalkar, Federated kernelized multi-task learning.

[159] F. Sattler, K.-R. Müller, W. Samek, Clustered federated learning: Model-agnostic distributed multi-task optimization under privacy constraints (2019). arXiv:1910.01991.

[160] O. Ohrimenko, F. Schuster, C. Fournet, A. Mehta, S. Nowozin, K. Vaswani, M. Costa, Oblivious multi-party machine learning on trusted processors, in: 25th {USENIX} Security Symposium ({USENIX} Security 16), 2016, pp. 619–636.

[161] Y. Chen, F. Luo, T. Li, T. Xiang, Z. Liu, J. Li, A training-integrity privacy-preserving federated learning scheme with trusted execution environment, Information Sciences 522 (2020) 69–79.

[162] Y. Shen, S. Sanghavi, Learning with bad training data via iterative trimmed loss minimization, in: International Conference on Machine Learning, 2019, pp. 5739–5748.

[163] B. Tran, J. Li, A. Madry, Spectral signatures in backdoor attacks, in: Advances in Neural Information Processing Systems, 2018, pp. 8000–8010.

[164] P. W. Koh, J. Steinhardt, P. Liang, Stronger data poisoning attacks break data sanitization defenses, ArXiv abs/1811.00741.

[165] Y. Zhao, J. Chen, J. Zhang, D. Wu, J. Teng, S. Yu, PDGAN: A Novel Poisoning Defense Method in Federated Learning Using Generative Adversarial Network, 2020, pp. 595–609. doi:10.1007/978-3-030-38991-8_39.

[166] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, Q. Yang, Secureboost: A lossless federated learning framework, CoRR abs/1901.08755. arXiv:1901.08755.
URL http://arxiv.org/abs/1901.08755

[167] S. Truex, L. Liu, M. Gursoy, L. Yu, W. Wei, Demystifying membership inference attacks in machine learning as a service, IEEE Transactions on Services Computing PP (2019) 1–1. doi:10.1109/TSC.2019.2897554.

[168] L. Melis, C. Song, E. De Cristofaro, V. Shmatikov, Exploiting unintended feature leakage in collaborative learning, in: 2019 IEEE Symposium on Security and Privacy (SP), 2019, pp. 691–706. doi:10.1109/SP.2019.00029.

[169] A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, R. Rogers, Protection against reconstruction and its applications in private federated learning, arXiv preprint arXiv:1812.00984.

[170] R. Canetti, U. Friege, O. Goldreich, M. Naor, Adaptively secure multiparty computation, Tech. rep., Cambridge, MA, USA (1996).

[171] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, S. Moriai, Privacy-preserving deep learning via additively homomorphic encryption, IEEE Transactions on Information Forensics and Security 13 (5) (2018) 1333–1345. doi:10.1109/TIFS.2017.2787987.

[172] M. Hao, H. Li, G. Xu, S. Liu, H. Yang, Towards efficient and privacy-preserving federated deep learning, in: ICC 2019 - 2019 IEEE International Conference on Communications (ICC), 2019, pp. 1–6. doi:10.1109/ICC.2019.8761267.

[173] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, S. Liu, Efficient and privacy-enhanced federated learning for industrial artificial intelligence, IEEE Transactions on Industrial Informatics (2019) 1–1.

[174] C. Dwork, Differential Privacy, Springer US, Boston, MA, 2011, pp. 338–340.

[175] L. Xie, K. Lin, S. Wang, F. Wang, J. Zhou, Differentially private generative adversarial network (2018). arXiv:1802.06739.

[176] S. Augenstein, H. B. McMahan, D. Ramage, S. Ramaswamy, P. Kairouz, M. Chen, R. Mathews, B. A. y Arcas, Generative models for effective ml on private, decentralized datasets (2019). arXiv:1911.06679.

[177] T. Zhu, P. S. Yu, Applying differential privacy mechanism in artificial intelligence, in: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), 2019, pp. 1601–1609. doi:10.1109/ICDCS.2019.00159.

[178] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, Y. Zhou, A hybrid approach to privacy-preserving federated learning, in: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, AISec'19, Association for Computing Machinery, New York, NY, USA, 2019, p. 1–11. doi:10.1145/3338501.3357370.
URL https://doi.org/10.1145/3338501.3357370

[179] B. Ghazi, R. Pagh, A. Velingker, Scalable and differentially private distributed aggregation in the shuffled model, arXiv preprint arXiv:1906.08320.

[180] G. Xu, H. Li, S. Liu, K. Yang, X. Lin, Verifynet: Secure and verifiable federated learning, IEEE Transactions on Information Forensics and Security 15 (2020) 911–926. doi:10.1109/TIFS.2019.2929409.

[181] F. TramÃšr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, P. McDaniel, Ensemble adversarial training: Attacks and defenses (2017). arXiv:1705.07204.

[182] Z. Wang, Y. Yang, Y. Liu, X. Liu, B. B. Gupta, J.-F. Ma, Cloud-based federated boosting for mobile crowdsensing, ArXiv abs/2005.05304.

[183] X. Luo, X. Zhu, Exploiting defenses against gan-based feature inference attacks in federated learning (2020). arXiv:2004.12571.

[184] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein gan (2017). arXiv:1701.07875.

[185] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. C. Courville, Improved training of wasserstein gans, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 5767–5777.
URL http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.pdf

[186] A. Triastcyn, B. Faltings, Federated generative privacy, IEEE Intelligent Systems (2020) 1–1.

[187] K. Chandiramani, D. Garg, N. Maheswari, Performance analysis of distributed and federated learning models on private data, Procedia Computer Science 165 (2019) 349 – 355, 2nd International Conference on Recent Trends in Advanced Computing ICRTAC -DISRUP - TIV INNOVATION , 2019 November 11-12, 2019.

doi:https://doi.org/10.1016/j.procs.2020.01.039.
URL http://www.sciencedirect.com/science/article/pii/S1877050920300478

[188] R. Shokri, M. Stronati, C. Song, V. Shmatikov, Membership inference attacks against machine learning models, in: 2017 IEEE Symposium on Security and Privacy (SP), 2017, pp. 3–18.

[189] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, M. Backes, Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models, arXiv preprint arXiv:1806.01246.

[190] X. Bao, C. Su, Y. Xiong, W. Huang, Y. Hu, Flchain: A blockchain for auditable federated learning with trust and incentive, in: 2019 5th International Conference on Big Data Computing and Communications (BIGCOM), 2019, pp. 151–159.

[191] E. Bagdasaryan, O. Poursaeed, V. Shmatikov, Differential privacy has disparate impact on model accuracy, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 15479–15488.

[192] J. Zhang, B. Chen, S. Yu, H. Deng, Pefl: A privacy-enhanced federated learning scheme for big data analytics, in: 2019 IEEE Global Communications Conference (GLOBECOM), IEEE, 2019, pp. 1–6.

[193] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, H. Vincent Poor, Federated learning with differential privacy: Algorithms and performance analysis, IEEE Transactions on Information Forensics and Security 15 (2020) 3454–3469.

[194] J. Zhang, J. Wang, Y. Zhao, B. Chen, An efficient federated learning scheme with differential privacy in mobile edge computing, in: X. B. Zhai, B. Chen, K. Zhu (Eds.), Machine Learning and Intelligent Communications, Springer International Publishing, Cham, 2019, pp. 538–550.

[195] J. Hayes, O. Ohrimenko, Contamination attacks and mitigation in multi-party machine learning, in: Advances in Neural Information Processing Systems, 2018, pp. 6604–6615.

[196] J. Zhang, J. Chen, D. Wu, B. Chen, S. Yu, Poisoning attack in federated learning using generative adversarial nets, in: 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), IEEE, 2019, pp. 374–380.

[197] T. Li, A. K. Sahu, A. Talwalkar, V. Smith, Federated learning: Challenges, methods, and future directions, IEEE Signal Processing Magazine 37 (3) (2020) 50–60.

[198] Q. Li, Z. Wen, B. He, Federated learning systems: Vision, hype and reality for data privacy and protection, CoRR abs/1907.09693. arXiv:1907.09693.
URL http://arxiv.org/abs/1907.09693

[199] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: Concept and applications (2019). arXiv:1902.04885.

[200] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y. Liang, Q. Yang, D. Niyato, C. Miao, Federated learning in mobile edge networks: A comprehensive survey, IEEE Communications Surveys Tutorials (2020) 1–1.

[201] Z. Tao, Q. Li, esgd: Communication efficient distributed deep learning on the edge, in: {USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 18), 2018.

[202] H. Wang, S. Sievert, Z. Charles, S. Liu, S. Wright, D. Papailiopoulos, Atomo: Communication-efficient learning via atomic sparsification (2018). arXiv:1806.04090.

[203] S. Caldas, J. Konecny, H. B. McMahan, A. Talwalkar, Expanding the reach of federated learning by reducing client resource requirements (2018). arXiv:1812.07210.

[204] Z. Du, C. Wu, T. Yoshinaga, K. A. Yau, Y. Ji, J. Li, Federated learning for vehicular internet of things: Recent advances and open issues, IEEE Open Journal of the Computer Society 1 (2020) 45–61.

[205] A. Gaier, D. Ha, Weight agnostic neural networks (2019). arXiv:1906.04358.

[206] K. Bonawitz, F. Salehi, J. Konecny, B. McMahan, M. Gruteser, Federated learning with autotuned communication-efficient secure aggregation, 2019 53rd Asilomar Conference on Signals, Systems, and Computersdoi:10.1109/ieeeconf44664.2019.9049066.
URL http://dx.doi.org/10.1109/IEEECONF44664.2019.9049066

[207] M. Mohri, G. Sivek, A. T. Suresh, Agnostic federated learning, arXiv preprint arXiv:1902.00146.

[208] M. Saharkhizan, A. Azmoodeh, A. Dehghantanha, K. R. Choo, R. M. Parizi, An ensemble of deep recurrent neural networks for detecting iot cyber attacks using network traffic, IEEE Internet of Things Journal (2020) 1–1.

[209] H. Karimipour, A. Dehghantanha, R. M. Parizi, K. R. Choo, H. Leung, A deep and scalable unsupervised machine learning system for cyber-attack detection in large-scale smart grids, IEEE Access 7 (2019) 80778–80788.

[210] A. Yazdinejad, R. M. Parizi, A. Dehghantanha, K.-K. R. Choo, P4-to-blockchain: A secure blockchain-enabled packet parser for software defined networking, Computers & Security 88 (2020) 101629. doi:https://doi.org/10.1016/j.cose.2019.101629.

[211] A. Yazdinejad, R. M. Parizi, A. Dehghantanha, K. R. Choo, Blockchain-enabled authentication handover with efficient privacy protection in sdn-based 5g networks, IEEE Transactions on Network Science and Engineering (2019) 1–1doi:10.1109/TNSE.2019.2937481.

[212] E. Nyaletey, R. M. Parizi, Q. Zhang, K.-K. R. Choo, Blockipfs–blockchain-enabled interplanetary file system for forensic and trusted data traceability, in: 2nd IEEE International Conference on Blockchain (IEEE Blockchain-2019), 2019.

[213] T. Nishio, R. Yonetani, Client selection for federated learning with heterogeneous resources in mobile edge, in: ICC 2019 - 2019 IEEE International Conference on Communications (ICC), 2019, pp. 1–7.

[214] P. Jiang, L. Ying, An optimal stopping approach for iterative training in federated learning, in: 2020 54th Annual Conference on Information Sciences and Systems (CISS), 2020, pp. 1–6.

[215] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecný, S. Mazzocchi, H. B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, J. Roselander, Towards federated learning at scale: System design, ArXiv abs/1902.01046.

- Providing a classification and overview of the approaches and techniques in the federated learning domain.
- Identifying and examining security vulnerabilities and threats in the federated learning environments.
- Identifying and evaluating privacy threats, their mitigation techniques, and the trade-offs cost associated with privacy-preserving techniques in the federated learning environments.
- Providing insights into existing defense mechanisms and future directions to enhance the security and privacy of the federated learning implementation.

Viraaji Mothukuri is a Research Assistant in the College of Computing and Software Engineering (CCSE) at Kennesaw State University GA, USA. She has several years of experience in Java and middleware technologies working with WIPRO and JP Morgan companies. Her research interests include Machine Learning, Hyperledger Fabric, Blockchain systems, and Decentralized Applications. She has a professional certification on Machine Learning to her credit.

Reza M. Parizi is the director of Decentralized Science Lab (dSL) at Kennesaw State University, GA, USA. He is a consummate AI technologist and software security researcher with an entrepreneurial spirit. He is a senior member of IEEE, IEEE Blockchain Community, and ACM. Prior to joining KSU, he was a faculty at New York Institute of Technology. He received a Ph.D. in Software Engineering in 2012 and M.Sc. and B.Sc. degrees in Software Engineering and Computer Science respectively in 2008 and 2005. His research interests are R\&D in decentralized AI, blockchain systems, smart contracts, and emerging issues in the practice of secure software-run world applications.

Seyedamin Pouriyeh is an Assistant Professor of Information Technology at Kennesaw State University, GA, USA. He received an M.Sc. in Information Technology Engineering from Shiraz University, and his Ph.D. in Computer Science from the University of Georgia in 2009 and 2018 respectively. His primary research interests span Federated Machine Learning, Blockchain, and Cyber Security.

Yan Huang is currently an Assistant Professor in the Department of Software Engineering & Game Development at Kennesaw State University, GA, USA. He received his Ph.D. degree in the Department of Computing Science at Georgia State University, and B.S., M.S. degrees from Heilongjiang University. His current research focuses on Cyber Security & Privacy, Federated Learning, and IoT.

Ali Dehghantanha is the director of Cyber Science Lab in the School of Computer Science, University of Guelph (UofG), Ontario, Canada. He has served for more than a decade in a variety of industrial and academic positions with leading players in Cyber-Security and Artificial Intelligence. Prior to joining UofG, he has served as a Sr. Lecturer in the University of Sheffield, UK and as an EU Marie-Curie International Incoming Fellow at the University of Salford, UK. He has PhD in Security in Computing and a number of professional certifications including CISSP and CISM. His main research interests are malware analysis and digital forensics, IoT security and application of AI in the Cyber Security.

Dr. Gautam Srivastava was awarded his B.Sc. degree from Briar Cliff University in U.S.A. in the year 2004, followed by his M.Sc. and Ph.D. degrees from the University of Victoria in Victoria, British Columbia, Canada in the years 2006 and 2011, respectively. He then taught for 3 years at the University of Victoria in the Department of Computer Science, where he was regarded as one of the top undergraduate professors in the Computer Science Course Instruction at the University. From there in the year 2014, he joined a tenure-track position at Brandon University in Brandon, Manitoba, Canada, where he currently is active in various professional and scholarly activities. He was promoted to the rank Associate Professor in January 2018. Dr. G, as he is popularly known, is active in research in the field of Data Mining and Big Data. In his 8-year academic career, he has published a total of 43 papers in high-impact conferences in many countries and in high-status journals (SCI, SCIE) and has also delivered invited guest lectures on Big Data, Cloud Computing, Internet of Things, and Cryptography at many Taiwanese and Czech universities. He is an Editor of several international scientific research journals. He currently has active research projects with other academics in Taiwan, Singapore, Canada, Czech Republic, Poland and U.S.A. He is constantly looking for collaboration opportunities with foreign professors and students. Assoc. Prof. Gautam Srivastava received *Best Oral Presenter Award* in FSDM 2017 which was held at the National Dong Hwa University (NDHU) in Shoufeng (Hualien County) in Taiwan (Republic of China) on November 24-27, 2017.

Viraaji – Conceptualization, Data curation, Formal analysis; Reza - Funding acquisition, Investigation; Seyedamin - review & editing; Yan - review & editing; Ali - Methodology, Project administration; Gautam - Roles/Writing – original draft, review & editing;

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: