CSIT (March 2022) 10(1):47–60
https://doi.org/10.1007/s40012-022-00351-0

BRIEF REPORT

Federated learning to comply with data protection regulations

Srinivasa Rao Chalamala[1] · Naveen Kumar Kummari[2] · Ajeet Kumar Singh[1] · Aditya Saibewar[1] · Krishna Mohan Chalavadi[2]

Received: 9 January 2022 / Accepted: 3 February 2022 / Published online: 15 March 2022

Abstract AI is adept at using large quantities of data, sometimes sensitive personal data, and can adversely affect individuals' privacy. Data privacy concerns significantly impact the course of next-generation AI. Users do not trust anyone withholding their data and need privacy-preserving intelligent systems. In addition, several regulations mandate that organizations handle users' data in ways that do not affect their privacy and provide them control on their data. Federated Learning emerged as a privacy-preserving technology for data-intensive machine learning by training the models on-site or on-device. However, several concerns related to federated learning emerged due to: (i) dynamic, distributed, heterogeneous, and collaborative nature of client devices, (ii) membership inference and model inversion attacks affecting the overall privacy and security of FL systems, (iii) the need for strict compliance to data privacy and protection laws, (iv) the vulnerabilities at local client devices leading to data leakage, and (iv) diversity and ubiquity of smart devices collecting real-time multi-modal data leading to lack of standardization efforts for security and privacy management framework. In this paper, we discuss (a) how federated learning can help us withholding privacy, (b) the need for improving security and privacy in federated learning systems, (c) the privacy regulations and their application to federated learning in various business domains, (d) proposed a federated recommender system and demonstrated the performance that matches the central setting.

Keywords Deep learning · Data privacy · Regulations · Federated learning

1 Introduction

Artificial intelligence and big data are receiving significant attention and raising privacy and other ethical concerns. All the concerning stakeholders need to comprehensively understand these issues and find mechanisms to address them to harvest the benefits of these technologies. There is a considerable advancement in multiple domains such as small businesses, automated vehicles, smart cities powered by these ubiquitous intelligent devices. 5G and Edge Cloud is bringing the intelligence closer to the edge, or the cell, instead of centralizing it in a set of servers. When data privacy is a significant concern, there is a high need for security and not trust anyone withholding our data.

On the other hand, federated learning (FL) acts as privacy-preserving learning mechanism which incorporates privacy into intelligent systems. FL can also help build better models, especially when minimal data is available with the service provider for training. When multiple

Srinivasa Rao Chalamal, K. Naveen Kumar have contributed equally to this work.

✉ Srinivasa Rao Chalamala
chalamala.srao@tcs.com

Naveen Kumar Kummari
cs19m20p000001@iith.ac.in

Ajeet Kumar Singh
ajeetk.singh1@tcs.com

Aditya Saibewar
aditya.saibewar@tcs.com

Krishna Mohan Chalavadi
ckm@cse.iith.ac.in

[1] TCS Research, TATA Consultancy Services, Hyderabad, India

[2] CSE Department, IIT Hyderabad, Hyderabad, India

entities have silos of data but have privacy concerns sharing the data, FL can help learn single and better models on these silos on-premises. Despite fewer data, low processing, and computing capabilities of FL clients, they can reap the benefits of the superior performance of deep learning algorithms.

Several countries enacted regulations to protect the privacy of the users. At the same time, some of them mandate restrictions on how the data is collected and processed but do not clearly mention the steps in case of indirect loss of sensitive personal data.

Traditional machine learning trains a model on data stored at a centralized server to make predictions. This leads to typical data privacy leakage issues as the data from different client resources and sensors are stored at a central server for processing. On the contrary, federated learning (FL) allows only the exchange of trained model parameters between the central server and the local client devices. In addition, it computes the local model update at the client device using data privacy protection techniques ensuring privacy. Further, these model updates are sent to the server for aggregation to improve the global model.

Federated Learning allows smarter models, lower latency, and less power consumption while ensuring privacy. In FL firstly, the server initializes the global model by pre-training it with publicly available central training data. Next, it selects K clients out of N clients to distribute the parameters of global model based on their resource information [1]. Further, each selected client trains the global model with their respective local data and uploads the model updates to the server. Finally, the server aggregates all the model updates and updates the global model by the averaged model as shown in Fig. 1. This process is repeats until the global model reaches convergence.

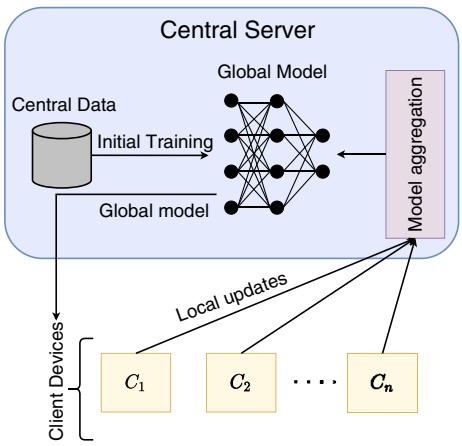


Fig. 1 The central server trains a global model on central data to share with local client devices $(C_1, C_2, \ldots, C_n)$. At each communication round, the clients share the local updates of the trained model. The server performs model aggregation to update the global model and sends it back to the client devices

The distributed nature of federated learning enables data scientists to effectively train a global model with only shared model weights obtained by local model training by decentralized devices as shown in Fig. 1. It implies that although data scientists learn the same globally shared across client devices, there is no exchange of any private data in federated learning. Unlike traditional machine learning with a centralized data resource, federated learning ensures data security and privacy by not sending the data to a server.

In addition, there is a high risk in centralized machine learning where the server holding the data can put the client's privacy at stake. On the contrary, federated learning allows the exchange of minimal model updates required to learn a global model that shows high performance on the test data. These model updates do not contain any private information related to the local data that can threaten the client's privacy. Despite all the advantages mentioned earlier of federated learning, several challenges [2] still need to be addressed. We list these challenges below:

1. *Trade-off between efficiency and privacy:* FL may not result in a better model or a model having equivalent performance as in a centralized setting.
2. *Privacy risks:* FL may not always guarantee privacy as the movement of model updates between the server and local clients can leak sensitive data information. Two primary attacks that can lead to privacy leakage are membership inference attacks and model inversion attacks.
3. *Communication bottlenecks:* The movement of model and its parameters between the server and client devices may impact the FL environment due to communication bottlenecks, which could stall or delay the Federated training process.
4. *Selection of clients among registered clients:* To optimize systems having heterogeneous resources, aggregation frequency must be dynamically adjusted to reduce communication overhead over networks with limited resources. Also, a selection approach needs to be employed to iteratively select a subset of clients for aggregation under the total registered client devices.
5. *System and Statistical heterogeneity:*The local client training in FL comes with a challenge in that all client devices contain heterogeneous computational resources and non-*i.i.d*(independent identically distributed) data. This challenge leads to improper training of the local model, which eventually affects the global model performance on the test data. Hence,

it is necessary to handle the heterogeneity of client devices to ensure robust FL scalability.
6. *Poisoning:* Adversarial clients can modify a part or entire training data resulting in data poisoning attacks. The adversary could also try to tamper the model layers in model poisoning attack to generate a poisoned update. Finally, the adversary aims to corrupt the global model by sending malicious updates.
7. *Aggregator turning malicious:* The central server can turn malicious and estimate private and sensitive information from the model updates received.
8. *Privacy vs performance:* Additional privacy mechanisms such as Differential Privacy (DP) add noise to the training data to ensure privacy. However, with DP, the performance of the model may be compromised. Hence, a trade-off between the right amount of privacy with performance is necessary.

There exist several review papers [3–7] on Federated learning in the literature. These review papers broadly discuss the issues in Federated Learning and the methods addressing these issues. However, these papers do not dwell much on the regulations and their impact on federated learning. These data regulations' impact could be different for various application sectors such as healthcare, retail, banking etc. Hence, this paper, explores these challenges concerning federated learning to comply with data protection regulations. Further, we summarize the main contributions of this paper as follows:

1. Explored federated learning through the lens of data privacy regulations and investigated the threats and defenses on federated learning systems.
2. Explained the data protection and privacy regulations and extended them to comply using federated learning.
3. We investigated the meta-learning approach in FL to deal with *non-i.i.d* and heterogeneous data among client devices.
4. We have demonstrated the use case of federated learning on the session-based recommendation system. We explained the NISER method for private and secure recommendation systems in FL. Also, we have thrown some light on federated reinforcement learning use-case for detecting attacks.

2 Regulations for data protection and privacy

Many countries have laws restricting the collection, use, and disclosure of personal data. Often, the processing and use of personal data is subjected to strict legal requirement and protective measures. These data privacy laws impose restrictions on when to collect the data and under what circumstances organizations may collect sensitive personal information, what purposes it can be used, and whether or not individuals first need to give consent before it is stored, processed, or disclosed. Policies on data ownership, informed consent, confidentiality, and security would be beneficial for identifying liabilities.

General Data Protection Regulations (GDPR) [8], **California Consumer Privacy Act (CCPA)** [9] provides rights to data owners to control their data. The GDPR focuses on creating privacy by default legal framework for the entire EU. CCPA lets consumers know what data is being collected, when it's being sold and shared for a business purpose. The principle user rights of the privacy regulations include the right to be informed, the right to access, the right to deletion, the right to prior consent, right to opt-out. ML algorithms' designs should accommodate sufficient privacy-preserving techniques to comply with rigorous regulations.

India's **NITI Aayog's** [10, 11] document on Resposible AI recommends that AI maintain the privacy and security of data of individuals or entities used for training the system.

The location and ownership of computers that store and access data for AI to use are essential. **Court of Justice of the European Union (CJEU)** [12] ruled in the Schrems II case that the EU-US Privacy Shield is void, terminating free data flows between the EU and the United States, providing impetus to ways to share the data without actually compromising on the privacy. Figure 2 broadly classifies the measures to be taken for data privacy protection in general. These measures include both technical and non-technical approaches. The technical methods help in achieving privacy at different stages of data processing pipeline. Regulations should require that personal data remain in the jurisdiction from which it is obtained, with few exceptions to address rigorous requirements of these regulations, ML algorithms shall accommodate sufficient privacy-preserving techniques which do not require data to be moved from one geography to another and one device to another in the design.

Few national laws require formal audits to determine compliance with privacy protection requirements. Also, audits are often necessary to support investigations into regulatory violations. In some countries, government oversight bodies or other authorities have the right to conduct audits of organizations within their jurisdictions.

2.1 Ethical issues

While the data protection regulations broadly cover the aspects of privacy in handling the users' data, they may not cover some aspects and may result in ambiguity in interpretation based on regional, and cultural practices. In

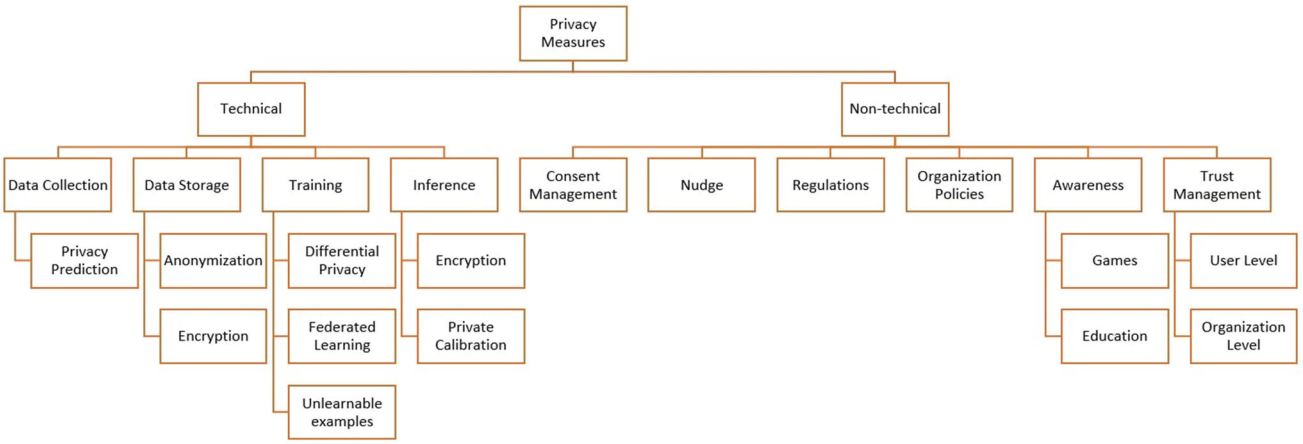


Fig. 2 Measures for data security

addition to adhering to the regulations, organizations may take the initiative and follow ethical principles in handling the users' data for privacy. Recently, EU's HLEG [13]; on AI recommended guidelines to promote Trustworthy AI. Particularly on privacy, it recommends (i) respect for privacy, (ii) quality and integrity of data, and (iii) access to data. Organizations developing AI solutions are recommended to follow human centered approach and values.

3 Attacks on federated learning

Multi-agent collaboration in Federated Learning is an exciting development in deep neural networks (DNN) [14]. This learning method achieves remarkable results in several application areas by offloading the computation-intensive training work to the clients. Functioning as a typical distributed system, the clients (or agents) send local model updates regularly to the central server. After collecting this local information, the server updates the global shared model and communicates the weights back to the clients. This iterative learning is repeated until the global server model reaches convergence. One fundamental difference between Federated Learning (FL) and Distributed Learning is that FL does not allow direct raw data communication. On the contrary, distributed learning does not have any such restriction and communicates the data.

The Federated Learning approach is developed to ensure data privacy and security onto deep learning models. FL allows only model parameters sharing between a central server and connected client devices. However, these model parameters can further leak data information using model inversion attacks and membership inference attacks, causing data privacy threats. The model querying capability thus is a significant vulnerability, and differential privacy and secure aggregation can further address the vulnerabilities.

Primarily the privacy concerns in federated learning arise due to the track of running estimates during training of deep learning models that an attacker can invert [15]. Atanov *et al.* [16] proposed a static batch normalization (sBN) to optimize privacy-constrained deep neural networks. The algorithm ensures normalizing the batch data without keeping track of running estimates during the training phase. After the model convergence, it calculates only the hidden representation statistics from the local client's data. Hence, sBN is highly suitable for the federated learning process as the local client models need to upload only the trained model parameters for each communication round. Further, local clients can only upload their trained model statistics after optimization, reducing the data leakage risk at the server.

However, there has been tremendous research interest in designing novel adversarial attacks and developing defense mechanisms for DNN [24] because of its gross mispredictions, even with minor perturbations [25]. This interest has percolated into the privacy-preserving Federated Learning, as researchers have begun investigating it through the lens of adversarial settings. A summary of related works is shown in Table 1. There is a high possibility to construct a poisoning attack in FL as the client's local data and training process are not accessible by the global server [26]. It is highly impossible to verify the authenticity and trustworthiness of a client's update as shown in Fig. 3. Prediction confidence reduction, misclassification, and targeted mis-classification are the primary goals of adversarial attacks on deep neural networks. These attacks can be divided into poisoning/causative attacks (i.e., training time attacks) and evasion/exploratory attacks (i.e., test time attacks).

Black-box adversarial attack in multi-agent communication is first introduced [27] using a computationally expensive surrogate-based approach. Bhagoji *et al.* [19] have focused on targeted model poisoning insted of data

Table 1 Related works on FL privacy & security threats

FL attack	Model(s)	Dataset(s)	Attacker	Attacker's Knowledge	Attack purpose
Information leakage in FL [17]	GAN, CNN	MNIST, AT&T dataset of faces	Client	White-box	Membership Inference Attack
Deep Leakage from Gradients [18]	CNN	MNIST, CIFAR-10, SVHN and LFW	Client	White-box	Membership Inference Attack
Model poisoning attack using boosting [19]	CNN	Fashion-MNIST, UCI Adult Census dataset	Client	White-box	Poisoning local model
Model poisoning attack [20]	CNN	Fashion-MNIST	Client	White-box	Poisoning local model
Comprehensive Privacy Analysis of Deep Learning [21]	ResNet , DenseNet	Texas100, Purchase100, CIFAR100	Client, Server	White-box	Membership Inference Attack
Beyond Inferring Class Representatives: User-Level Privacy Leakage From Federated Learning [22]	CNN	AT&T , MNIST, CIFAR100	Server	White-box	Model Inversion Attack
Inverting Gradients - How easy is it to break privacy in federated learning? [23]	CNN	MNIST, CIFAR100	Server	White-box	Membership Inference Attack
Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning [17]	CNN	MNIST, AT&T	Client	White-box	Model Inversion Attack

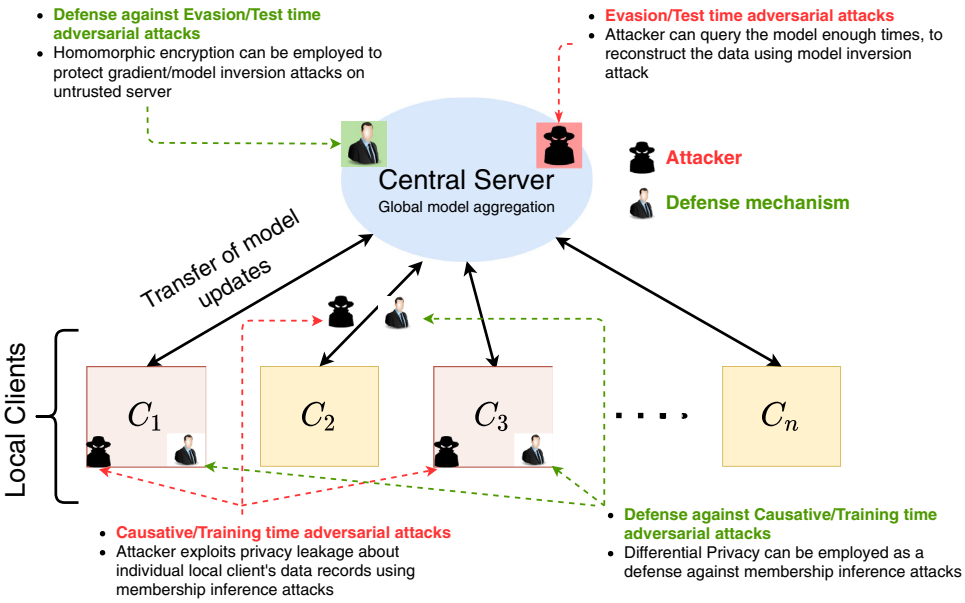


Fig. 3 Overview of privacy and security threats of Federated Learning. There are two types of attacks: causative (training time) evasion (test time) attacks. The attacker exploits privacy leakage about individual local client's data records using membership inference attacks and model/gradient inversion attacks. Differential privacy;and homomorphic encryption can be used to defend against these attacks

poisoning in federated learning. Zizzo *et al.* [28] have taken the first step known in literature towards defense against the threat of evasion attacks during inference in FL. Nonetheless, this can create some empirical robustness; and data privacy across all clients will be compromised as this can leak some local data by inducing a small set of adversarial training data globally shared between all learning clients.

In addition to adversarial attacks, the attacker's motivation extends to privacy leakage in FL. The continuous communication of model updates between the server and the local clients throughout the training process can reveal sensitive data information to a third party or the central server. Aono *et al.* [29] have shown that local data leakage can happen even with a small portion of original gradient information.

Further, the membership inference attacks [30–32] on FL can exploit leakage of local client's data during training. In addition, the attacker can build a shadow model to create a dataset similar to that of the original dataset. Also, model updates in FL can leak information about training data features as deep learning models are designed to internally recognize many data features unrelated to main tasks. On the other hand, model inversion attacks [33, 34] on FL drastically lead to data privacy leakage as the attacker learns the data distribution from other participating client devices. The attacker can also record the model parameters at regular intervals and exploit the difference between the subsequent model updates. Since the model updates are derived from the client's private data, the adversary can reconstruct the data using model inversion. [35] Discusses a good approximation of private client data labels using those labels provides a novel way to retrieve the original batch from the client data.

In addition, the attacker aims to corrupt the entire federated learning process by compromising the benign learning nodes with poisonous data in the form of altered features or false labels. The attacker can control the learning process and send malicious updates to the central server. These poisoned updates used for aggregation cause drastic degradation of the overall performance.

4 Federated learning and regulations

Identifying and clarifying the laws that apply to the respective geographies, businesses is the first step in achieving compliance. In addition, clearly understanding what personal information is protected by the respective privacy acts and what aspects relate to consumer control of private information play important role in achieving compliance.

Article 24 of the GDPR directs organizations to fully comply with all data protection principles. In centralized machine learning, data is collected from end-users, the privacy notice should explain the lawful basis for processing and the purposes of the processing. Similarly, in a federated learning setting, if the central server or aggregator is categorized as a data controller and a data processor, hence it is responsible for demonstrating and ensuring compliance with the regulation's data protection principles.

Current ML system now have several privacy-preserving methods, however implementing these regulatory obligations even in a centralized ML-based system is non-trivial, and maybe technologically impractical.

Federated learning is an alternative for the cloud-centric and centralized ML approaches, and not anymore preferred choice due to challenges of compliance to regulations on vast aggregation and processing of personal data. As indicated in earlier sections, federated learning potentially mitigates data privacy-related risks by enabling collaborative training of ML models while retaining original personal data on their devices.

While Federated learning is deemed to protect the personal information of the client devices or end-users, attacks [5, 6] on the model updates could prove it be otherwise. A central server in the FL setting may not be directly fulfilling the role of the data controller or data processor, However, any compromise on the intermediate model updates received from the end nodes or devices may lead to hefty penalties as the collection of model updates maybe sometimes done in the background without explicit end-user attention or knowledge.

Central server or aggregator as a data controller must implement appropriate technical measures and security measures that demonstrate the data processing activities such as training client nodes, communication, and model aggregations have been performed in accordance with regulations. While the regulations mention differing roles for controllers and processors, there could be some overlap of the responsibilities in a federated learning setting. Both the central server and client nodes may have to fulfill the roles of controller and processor. For example, a client has access to the model(data controller) trained on other's data and further training on local data (data processor). Also, a server may have similar access to the model (data controller) and client updates(data processor). In essence, the security of the users' data is a collective responsibility of both the central server and client nodes. Unfortunately, there are no regulations to control if the end devices turn out to be malicious and cause privacy issues. But, the laws may be directly applicable in the case of central server.

In FL, the workflow for training at multiple clients involves: (i) selection of the global model (ii) selection of initial training data (iii)selection of the number of clients (iv) selection of the clients among the registered clients (v) selection of secure aggregation mechanism (vi)selection of secure communication mechanism (vii) selection of hyper-parameters (viii) selection of privacy-preserving training mechanism (ix) selection of evaluation approach.

Each of the above steps is influenced by the regulations directly or indirectly. For example, selecting a global model could be critical for countering some attacks. The Selection of clients(possibly colluded) and hyper-parameters can impact the security of the model.

Also, who is responsible for deploying client functions as in the case of serverless cloud architecture, significantly impacts security and privacy. If the controller is responsible, it is expected to have a homogeneous security architecture across clients. Security architecture could be heterogeneous when participating institutions and users are

responsible for deploying the client functions themselves. FL clients may belong to separate organizations and networks. It is crucial that only authenticated and authorized entities can invoke client functions as their functionality is exposed to the public internet. This allows the clients to have their local training workflows, resulting in a data leak. The data leak at one client could affect the overall privacy of all the users/clients.

Even though appropriate authentication and authorization mechanisms are in place, the federated learning setting assumes all the end nodes or clients are considered trustworthy and this assumption may not hold from a security angle.

The HIPAA (Health Insurance Portability and Accountability Act) [36] enacted for the protection of Protected Health Information (PHI) at each stage of data life cycle including when it created, sourced, used, and maintained by the service entity. HIPAA privacy rules regulate the use and disclosure of individuals' health information and other individually identifiable health information. HIPAA protects patient health data from organizations that provide healthcare services, such as insurance companies and hospitals. Organizations or services such as DNA analysis for health conditions, are not legally counted as healthcare services and can exploit healthcare data and still can evade regulations. Appropriate safeguards must be taken at each workflow stage when the AI models are built using these silos of healthcare data.

Healthcare made significant advances due to data-driven machine learning on medical data, but still, due to privacy constraints, not all the available data is used. Federated learning can solve privacy issues to some extent. WHO recently issued six guiding principles for AI in healthcare through *"Ethics and governance of artificial intelligence for health"*. Protecting human autonomy is one of the essential guiding principles which mandates (i) healthcare systems and medical decisions shall be under the control of humans; (ii) privacy and confidentiality should be protected. FDA [37] recently released its Software as a Medical Device (SAMD) action plan covering privacy and security of medical devices.

Also, GDPR does not make a distinction between "tech companies" and "organizations that provide health services." Under the law, all organizations must obtain informed, explicit consent from the user to collect their data. A few general recommendations that ensure privacy policies and ethical standards are guaranteed include i) educate users on how their data will be used. ii) right to decline or withdraw consent iii) incorporate technical safeguards into their AI systems. Technology companies can take care of the third aspect using federated learning while complying with the local regulations.

Retail businesses often engage in targeted online advertising and sometimes process sensitive personal information. In 2018, California passed the California Consumer Privacy Act (CCPA), which will come into effect from 2023. CCPA enables the right for private citizens to sue businesses for privacy violations. These laws on Consumer Data Protection mandate organizations to publish a privacy policy that informs consumers how the consumers' data is collected, used and shared. In addition, some of these laws introduce Cybersecurity requirements. These laws also, enable consumers to opt-out of data for advertising and from the sale of their data to third parties, which may not be easy for the retailers to comply with as often retailers might be operating heterogeneous and distributed systems. It is challenging to locate and delete a user's data on each device or node. Federated Learning could enable e-commerce or retailers to comply with the privacy regulations to some extent. Still, as seen in the previous sections, FL-based solutions are also vulnerable to attacks, and companies are responsible for cybersecurity risks at information and curated model levels. However, FL may alleviate the problem of right-to-be-forgotten as the data is collected anonymously through model updates. Nevertheless, when the retails need to personalize the services, they may not wholly avoid personal data.

As in the case of retail, banking services also need to protect individuals' privacy by taking measures according to the law of the land. Banking applications such as fraud detection, the financial performance of individuals, personalized banking products are increasingly using ML and need greater attention in terms of privacy compliance. While banks may use silos of data from other banks in training the models in a federated way for a common purpose, they still need to ensure the data is de-identified before processing to avoid any penalties as they can carry information about the location of the bank. Unlike the retail sector, any banking and financial data resides with both the user and the bank. Hence it is difficult to identify where the data leaked from. Also, any custom product and services using personal and sensitive data must ensure that all the regulations are followed in writing and spirit.

It is demonstrated in the recent literature that retaining data and computation on-device in federated learning is insufficient for privacy assurances. Exchange of machine learning model parameters between entities in an FL system can still conceal sensitive information and be exploited in some privacy attack compromising on privacy guarantee.

4.1 Evolving privacy laws and technology

Compliance and privacy laws continue to evolve. Similarly, organizations incorporate new technologies to improve and expand their solutions. These changes will

affect compliance at different stages and scales. Technology must provide for ongoing regulations and compliance monitoring changes to prove effective in the long term.

5 Security of federated learning systems

Federated learning systems shall be strengthened by efficient privacy-preserving techniques to comply with the GDPR and other regulations. In federated learning architecture, the aggregating server and the other participating clients could be malicious. A high-level security architecture for federated learning systems has been depicted in Fig. 4 which showcases different privacy-preserving technologies. These technologies address different issues related to privacy and security in federated learning workflow.

Generally, the summary of the new knowledge learned on the client's data is sent back to the server in FL. For security, this knowledge is encrypted. To prevent the server from estimating individual data samples based on the summary update it has received, a Secure aggregation protocol [38] is developed by Google. Other methods such as FedMA, FedPer are introduced to improve FL privacy preservation in FL workflow. In FedMA [39] layers are independently trained and communicated to the server. FedPer [40] splits the model into the base and personalized layers. In this, only base layer updates are sent to the server are aggregated by the federated server using transfer learning methodologies and the personalized layers are not communicated to the server protecting individuals client's privacy.

Cryptographic methods like homomorphic encryption; [41–43], and secure multi-party computation [44] are common privacy methods used in FL systems. Homomorphic encryption of model updates could prevent both the client and server from extracting knowledge about the training data. With homomorphic encryption, encrypted data from clients are sent to the server. The server works on the encrypted data. Lastly, the encrypted output is decrypted to get the final result. Although these methods provide cover against many attacks, they are computationally expensive as the FL process involves multiple rounds of learning.

Differential privacy [45–48] can be used to provide quantifiable privacy in a database. This can be done by working with different approaches to adding noises to the users' data to provide ample privacy. This can help us ascertain the likelihood of someone leaking private information from the dataset and define the upper bound on how much data can be leaked at most.

One common technique used by differential privacy is to protect local clients' privacy is by adding noise (e.g. Gaussian and Laplacian) to the data. It can be categorized according to the place the noise is added as

1. *Local differential privacy:* the noise is added to each individual data point in the local client training dataset.
2. *Global differential privacy:* the required noise to protect the user privacy is added at the output of the query of the local dataset.

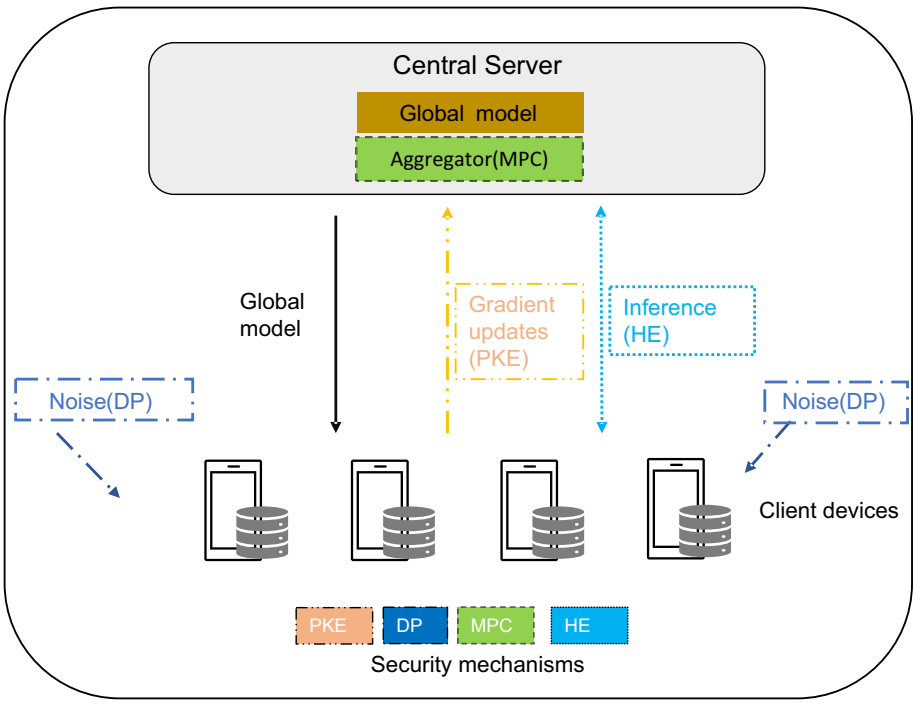


Fig. 4 A high level security architecture for Federated learning systems

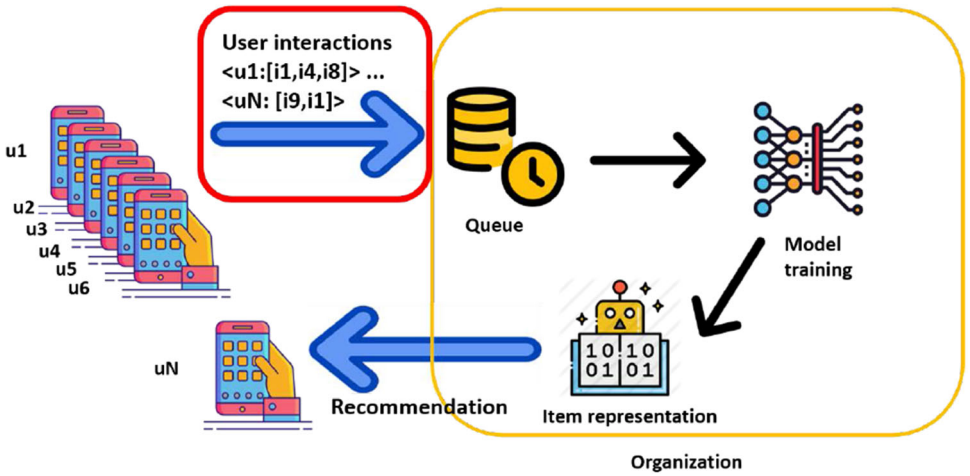


Fig. 5 The current session based recommendation systems without any user privacy. All the interactions has to be shared with the organization/service provider for training the recommendation engine

Federated learning aims to protect data privacy through distributed learning methods that keep the data on device/premises without transferring it to untrusted data processors. But, this does not prevent privacy attacks such as membership inference and model inversion. Likewise, differential privacy aims to improve data privacy protection by measuring the privacy loss in the communication among the elements of federated learning. Differential privacy employs random noise to data or model parameters, thereby masking the gradients. Differential privacy defends against the attacks [49–52], albeit with varying levels of protection-privacy trade-offs. A significant drawback of this method is that the model accuracy is significantly reduced due to added noise. Global differential privacy has been shown to perform better than local differential privacy while keeping the same privacy level. But, an explicit trust is required on people donating or sharing their data and information to the data aggregator. The data aggregator will add suitable noises into the user data to preserve privacy. FedProx [53] is proposed to tackle heterogeneity in federated networks. It provides convergence guarantees when learning from non-identical distributions (i.e. statistical heterogeneity) data. Differential Private versions of FedAvg [54] , and SCAFFOLD [53] were proposed to address the security and heterogeneity issues in federated learning.

5.1 Federated reinforcement learning for detection of attacks

Developing a robust federated learning system is challenging due to data heterogeneity and the vulnerability of client devices to privacy threats. Traditional centralized machine learning defense mechanisms with a high false alarm rate fail to address the threats. In addition, they do not take privacy into account. On the other hand, Federated Reinforcement Learning (FRL) framework is proven to be a robust defense mechanism in detecting attacks to mitigate the risk due to attack in different scenarios [55]. Mowla et al. [56] proposed an adaptive federated reinforcement learning-based defense strategy against jamming attacks. The authors developed a model-free Q-learning directed by an on-device federated jamming detection mechanism with an adaptive exploration-exploitation epsilon-greedy privacy strategy. Wang et al. [57] built a universal anomaly detection model using a federated learning technique where each local client model trains a deep reinforcement learning algorithm. The authors claim that anomaly detection accuracy can be significantly improved by introducing privacy leakage degree and action relation to detection design. We observed very few works that use federated reinforcement learning to detect anomalous client updates at the central server in the literature. In the future, we would like to design and develop a deep reinforcement learning algorithm to detect malicious client updates at the central server. Also, the algorithm helps in the client selection process for model aggregation.

5.2 Meta learning

Federated learning has been shown to struggle to deal with non-IID data and the heterogeneous structure among clients. To this end, one can use meta-learning to extract and propagate internal transferable representations of prior tasks. This has the advantage of preventing over-fitting and improving generalization. This shows that meta-learning can help handle the statistical and systematic challenges of a federated setting [58]. Though meta-learning is helping us address the systemic and statistical heterogeneity, the shared global model still implicitly includes all clients' privacy. Hence, a thorough evaluation of the global meta learner is necessary against various privacy and security [59, 60] threats.

6 Federated learning on session based recommendation system

Recommendation systems have been used heavily by online streaming services, retail services, dating platforms to provide the user with relevant items which are otherwise very difficult to do in their absence.

6.1 Recommendation systems

Different algorithms have been proposed based on the availability of the data and domain constraints. Collaborative filtering is one such class of recommendation algorithms. The recommendation is generated using the rating information from other users and items in these algorithms. But these algorithms do not factor in any content related to users and items, resulting in a cold-start problem. Content-based recommendation algorithms were proposed to deal with this issue that uses additional information about users and/or items. This helps the recommendation system to make more relevant recommendations to users. Both the collaborative and content-based filtering methods generally rely on historical user-item interactions to understand a user's long-term preferences. The underlying assumption here is that historical interactions are equally important to the user's current preference, which might not be accurate. To make more relevant recommendations, both a user's longer-term historical and recent preferences should be considered together. Hence, session-based recommendation algorithms were proposed which rely heavily on the user's most recent interactions rather than only on her historical preferences. It is also appropriate when she appears anonymously rather than logged in. Figure 5 shows a typical workflow of centralized recommendation system without any privacy measures in place.

Popularity Bias. The aforementioned class of recommendation algorithms has been shown to improve the recommendations to the user based on her historical and recent interactions. However these algorithms are known to suffer from the popularity-bias problem wherein popular items get a lot of exposure while less popular ones are under-represented in the recommendations. It has been found that popular SR-GNN [61] suffers from it. It becomes more problematic in an online setting where new items are frequently added to the catalog and are less prevalent in the initial days. To mitigate this, Normalized Item and Session Representation (NISER) [62] for session-based recommendations was proposed. This restricts the item and session-graph representation to lie on unit hyperspace both during training and inference, which helps tackle the popularity bias.

6.2 Privacy issues in recommendation systems

This should be noted that all the methods using the aforementioned algorithms are developed from the private data collected from the users. This private comprises the behavioral information, the contextual information, the domain knowledge, the item metadata, the purchase history, the recommendation feedback, the social data, and so on. Some integrate multiple data sources from other organizations to improve these recommendations. All these users' private information is stored on a central system where recommendation engines are trained. This data centralization poses grievous privacy and security risks. Not only the centralized data is more vulnerable to hacking and other forms of data theft, but it also gives control to whoever controls the server. The user has no control whatsoever over how their data is used once handed over.

6.3 Federated learning for resolving privacy issues

Data decentralization is necessary to prevent such gross mishandling of user's data adhering to privacy laws and regulations. This decentralization can adversely affect the recommendations systems development as there will not be much data to learn from them. Hence, federated learning can be used to train recommendation systems where the data never leaves the user's device, respecting the privacy of her data. Only the model parameter updates are used to communicate with the aggregating server.

6.4 Private NISER using federated learning

To provide users with better and more relevant recommendations but at the same time preserve the privacy of their data, we adapt the NISER [62] algorithm to the federated setting. We train the recommendation model using the same algorithm used in NISER but at the user's devices.

6.4.1 The method

To train the NISER [62] in federated setting, we define a global model $\mathcal{F}_g$ at server and $\mathcal{N}$ clients of which $\mathcal{C}$ will be chosen to train the global model which will be eventually shared with all $\mathcal{N}$ clients. Each client n will have all past sessions $\mathcal{S}_n$, and set of m items observed ($\mathcal{I}_n$) in the set $\mathcal{S}$. Each session $s \in \mathcal{S}_n$ is a chronologically ordered tuple of item-click events: $s = (i_{s,1}, i_{s,2}, \ldots, i_{s,l})$ where each of the l item-click events $i_{s,j}(j = 1, 2, \ldots, i_{s,l})$ corresponds to an item in $\mathcal{I}_n$ and j denotes the position of the item $i_{s,j}$ in the session s. A session s can be modeled as a grah $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$, where $i_{s,j} \in \mathcal{V}$ is a node in the graph.

Further, $(i_{s,j}, i_{s,j+1}) \in \mathcal{E}_s$ is a directed edge from i_{s_j} to $i_{s,j+1}$. Given s, the goal of SR is to predict the next item $i_{s,j+1}$ by estimating the mdimensional probability vector $\hat{y}_{s,l+1}$ corresponding to the relevance scores for the m items. The K items with highest scores constitute the top-K recommendation list. We follow the steps in algorithm 1 to update the global model $\mathcal{F}_g$ using popular federated averaging [63].

Algorithm 1 Private NISER for Private and Personalized Recommendations using Federated Averaging

On Server
initialize global model $\mathcal{F}_g = \theta_0$
for each round $t = 1, 2, \ldots$ **do**
  $c \leftarrow \max(\mathcal{N} \cdot \mathcal{C}, 1)$
  for each client $m \in c$ **in parallel do**
    $\theta^c_{t+1} \leftarrow \text{ClientUpdate}(c, \theta_t)$
  end for
  $\theta_{t+1} \leftarrow \sum_{c=1}^{K} \frac{n_c}{n} \theta^c_{t+1}$
end for
ClientUpdate(c, θ) : ▷ Run on client c
$\mathcal{B} \leftarrow$(Split $\mathcal{S}_]$ into batches of size B)
for each local epoch i from 1 to E **do**
  for batch $b \in \mathcal{B}$ **do**
    get item embeddings $\mathcal{I}_c$
    normalize $\mathcal{I}_c$
    initialize two adjacency matrices A^{in}_s and A^{out}_s using $\mathcal{I}_c$ for incoming/outcoming edges
    $\theta \leftarrow \theta - \eta \nabla loss(\theta; b)$
    return updated item embeddings for session s $\tilde{\mathcal{I}}_s$ using GNN on graph $\mathcal{G}(A^{in}_s, A^{out}_s, \tilde{\mathcal{I}}_s); \theta)$
  end for
end for
return θ to server

As mentioned in algorithm 1, the randomly initialized global model($\mathcal{F}_g = \theta_0$ is shared with a randomly chosen $N \cdot K$ client for the local training. At each client, the sessions are recorded and item embeddings ($\mathcal{I}_s$) for all the sessions (S_k). Locally, these item embeddings are updated using a graph neural network from the nodes and vertices obtained from the $\mathcal{I}$. This gives a combined embedding of normalized item representation, session representation, and position embedding. This updated item embedding is then used to obtain the relevance score for net clicked item i_k computed as,

$$\hat{y}_k = \frac{exp(\sigma \vec{i}_k^T \tilde{s})}{\sum_{j=1}^{m} exp(\sigma \vec{i}_j^T \tilde{s})} \quad (1)$$

where m is the total number of items present. Local training of the model and subsequent update of the global model by aggregating the model updates will continue to go on until the global model has achieved convergence.

6.4.2 Dataset and evaluation metrics

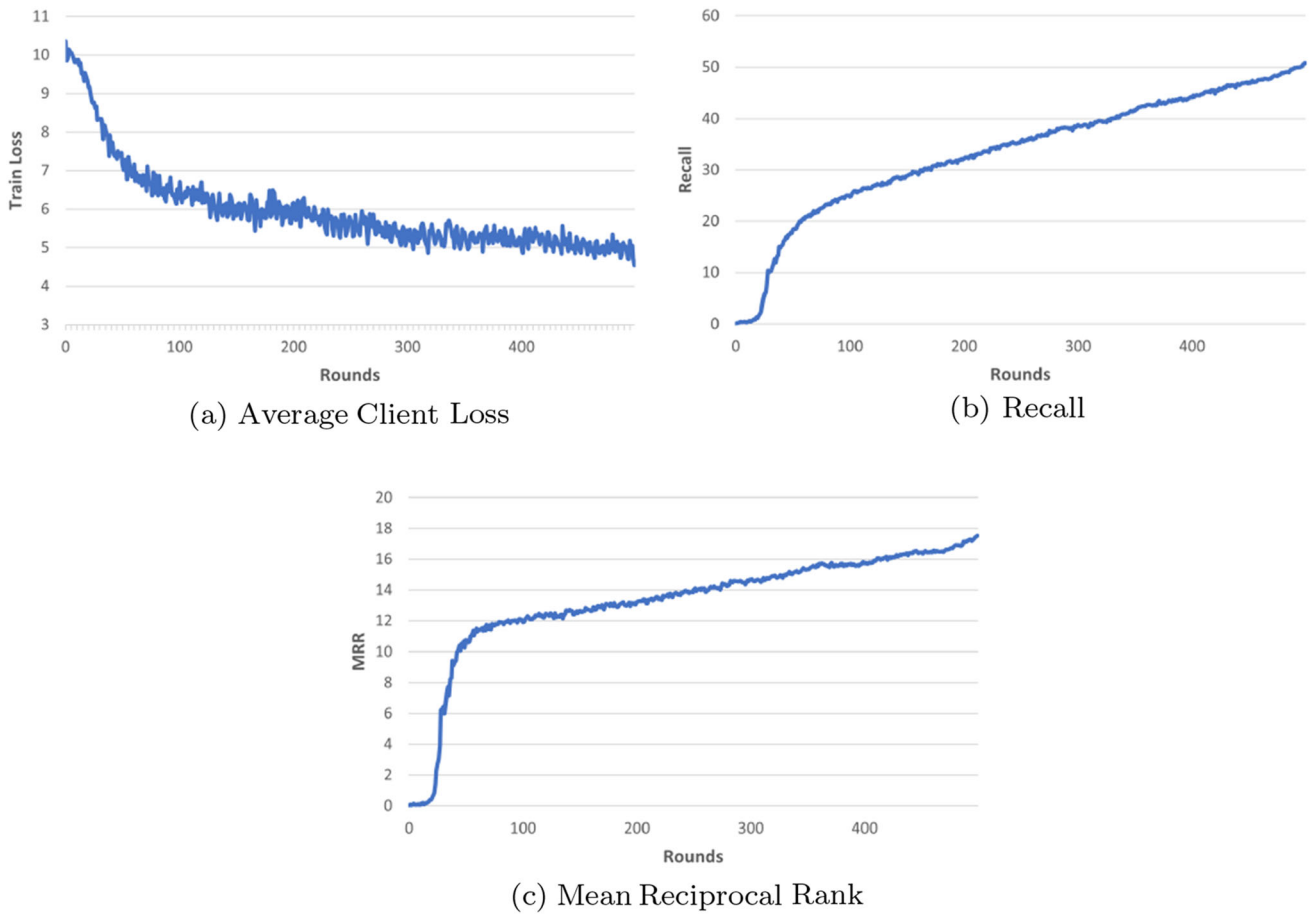
We evaluate the federated model on a publicly available benchmark dataset: Diginetica, transactional data from the CIKM Cup 2016 challenge. This dataset contains 700K train sessions 60859 test sessions over 43K items. The average length of a session is 5.12.

We use two evaluation metrics to evaluate the global model: Recall@K and Mean Reciprocal Rank(MRR@K) as in NISER [62] with K=20. Recall@K represents the proportion of test instances which has desired item in the top-K items. MRR@K is the mean of reciprocal ranks of the desired item in the recommendation list. The large value of MRR indicates the item is in the top of the recommendation list.

6.4.3 Implementation details

We reproduce the NISER [62] model which we treat as central model to compare the federated model with. We follow the same steps in [62] to train and test the model. The central model was trained with item embeddings of size 100, learning rate of 0.001, Adam optimizer and dropout was set to 0.1. 10% of the train set was used for validation set.

For federated model, we simulate the experiments with 100 clients for 500 rounds. At client side, to train the local model we use $epochs = 3$, learning rate is set as 0.001, dropout is 0.1 and SGD optimizer with momentum set to



(a) Average Client Loss (b) Recall (c) Mean Reciprocal Rank

Fig. 6 For federated NISER (global) model, we present across rounds, (a) the average client loss, (b) recall@20 and (c) mean reciprocal rank (MRR@20)

Table 2 Central vs. federated model: a comparison between central and federated model on Diginetica dataset

Model	Recll@20	MRR@20
Central	52.63	18.27
Federated	51.2	17.56

0.9. The gradients from the clients are aggregated using federated averaging [63] at server.

6.4.4 Results and discussion

This section discusses the efficacy of recommendations in federated against the recommendations by the central version of the model. In Fig. 6, we present average client loss (Fig. 6a) across rounds. Also, Fig. 6b and 6c shows the Recall@20 and MRR@20 across 500 rounds, respectively, for the updated global model. As we can see in Table 2, recommendations in both centralized and federated settings are comparable with minimal depreciation in performance. But, the federated model does not require the data to be present at a server as it gets aggregated from the parameters obtained from the client. The client updates the shared global model locally and shares the parameters with the trusted server for updating the global federated model.

In earlier sections, it has been seen that just adapting to a federated setting will not necessarily make it private. The private NISER is only private in ideal conditions when treating all participating parties as honest. For example, there is still the chance of model poisoning by a dishonest client. A dishonest aggregator can still infer private information from the client updates. To improve in such scenarios, algorithms based on meta-learning and differential privacy are in the pipeline to reduce the threats related to models' and users' privacy and security.

7 Conclusion

Federated learning is helpful in building privacy-aware collaborative learning with the cooperation of multiple clients or institutions and a data or central server. FL opens

new opportunities despite strict privacy regulations due to its nature of learning by only allowing model updates to be exchanged while retaining original raw data at the source. With the increase in the adoption of cloud-based services, FL can accelerate the training of AI models from silos of data scattered across institutions. FL can facilitate the personalization of services by training AI models locally. While FL provides the convenience of learning on-site, it is still vulnerable to attacks from malicious entities in the learning workflow. In this paper, we discussed FL's benefits and challenges and the impact of regulations on the learning process. Each of the entities in the FL workflow has to fulfill multiple roles as laid by the regulations based on what kind of data they possess, even if it is temporary. The interpretation of privacy regulations on the federated learning architecture under poisoning, model inversion, inference attacks is crucial to achieving absolute compliance and trust among the users.

References

1. Nishio T, Yonetani R (2019) Client selection for federated learning with heterogeneous resources in mobile edge. In: ICC 2019-2019 IEEE international conference on communications (ICC), pp. 1–7 . IEEE
2. Chen J, Pan L, Wei Z, Wang X, Ngo CW, Chua TS (2020) Zero-shot ingredient recognition by multi-relational graph convolutional network. In: AAAI
3. Kairouz P, McMahan HB, Avent B, Bellet A, Bennis M, Bhagoji AN, Bonawitz K, Charles ZB, Cormode G, Cummings R, D'Oliveira RGL, Rouayheb SYE, Evans D, Gardner J, Garrett Z, Gascón A, Ghazi B, Gibbons PB, Gruteser M, Harchaoui Z, He C, He L, Huo Z, Hutchinson B, Hsu J, Jaggi M, Javidi T, Joshi G, Khodak M, Konecný J, Korolova A, Koushanfar F, Koyejo O, Lepoint T, Liu Y, Mittal P, Mohri M, Nock R, Özgür A, Pagh R, Raykova M, Qi H, Ramage D, Raskar R, Song DX, Song W, Stich SU, Sun Z, Suresh AT, Tramèr F, Vepakomma P, Wang J, Xiong L, Xu Z, Yang Q, Yu FX, Yu H, Zhao S (2021) Advances and open problems in federated learning. Found Trends Mach Learn 14:1–210
4. Li Q, Wen Z, He B (2021) A survey on federated learning systems: Vision, hype and reality for data privacy and protection. ArXiv 1907.09693
5. Enthoven D, Al-Ars Z (2020) An overview of federated deep learning privacy attacks and defensive strategies. ArXiv 2004.04676
6. Lyu L, Yu H, Yang Q (2020) Threats to federated learning: A survey. ArXiv 2003.02133
7. Li T, Sahu AK, Talwalkar AS, Smith V (2020) Federated learning: challenges, methods, and future directions. IEEE Signal Process Magaz 37:50–60
8. GDPR: General Data Protection Regulation (GDPR). https://gdpr-info.eu/ (2018)
9. CCPA: General Data Protection Regulation (GDPR). http://leginfo.legislature.ca.gov/faces/codes_displayText.xhtml?division=3.&part=4.&lawCode=CIV&title=1.81.5 (2018)
10. NITI-Aayog: Principles for Responsible AI. https://www.niti.gov.in/sites/default/files/2021-02/Responsible-AI-22022021.pdf (2021)
11. NITI-Aayog: Principles for Responsible AI. https://www.niti.gov.in/sites/default/files/2021-08/Part2-Responsible-AI-12082021.pdf (2021)
12. European Parliament: The CJEU judgment in the Schrems II case. https://www.europarl.europa.eu/RegData/etudes/ATAG/202652=073/EPRS_ATA(2020)652073_EN.pdf (2020)
13. High-Level Expert Group on Artificial Intelligence : Ethics Guidelines for Trustworthy AI. https://ec.europa.eu/newsroom/dae/redirection/document/56341 (2018)
14. Smith V, Chiang CK, Sanjabi M, Talwalkar A (2017) Federated multi-task learning. arXiv preprint arXiv:1705.10467
15. Mothukuri V, Parizi RM, Pouriyeh S, Huang Y, Dehghantanha A, Srivastava G (2021) A survey on security and privacy of federated learning. Fut Gener Comput Syst 115:619–640
16. Atanov A, Ashukha A, Molchanov D, Neklyudov K, Vetrov D (2019) Uncertainty estimation via stochastic batch normalization. Int Symp Neural Netw. Springer, Cham, pp 261–269
17. Hitaj B, Ateniese G, Perez-Cruz F (2017) Deep models under the GAN: information leakage from collaborative deep learning. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security, pp. 603–618
18. Zhu L, Liu Z, Han S (2019) Deep leakage from gradients. CoRR 1906.08935
19. Bhagoji A.N, Chakraborty S, Mittal P, Calo S (2019) Analyzing federated learning through an adversarial lens. In: International conference on machine learning, pp. 634–643 . PMLR
20. Bhagoji A.N, Chakraborty S, Mittal P, Calo S (2018) Model poisoning attacks in federated learning. In: Proc Workshop Secur Mach Learn (SecML) 32nd Conf Neural Inf Process Syst (NeurIPS)
21. Nasr M, Shokri R, Houmansadr A (2019) 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). https://doi.org/10.1109/sp.2019.00065
22. Wang Z, Song M, Zhang Z, Song Y, Wang Q, Qi H (2018) Beyond inferring class representatives: user-level privacy leakage from federated learning
23. Geiping J, Bauermeister H, Dr-öge H, Moeller M (2020) Inverting gradients–how easy is it to break privacy in federated learning?
24. Ximeng L, Lehui X, Yaopeng W, Xuru L (2020) Adversarial attacks and defenses in deep learning. Chin J Netw Inf Secur 6(5):36
25. Goodfellow I, Shlens J, Szegedy C (2015) Explaining and harnessing adversarial examples. In: International conference on learning representations . 1412.6572
26. Hayes J, Ohrimenko O (2018) Contamination attacks and mitigation in multi-party machine learning. In: Advances in neural information processing systems 31 (NeurIPS 2018)
27. Tu J, Wang T, Wang J, Manivasagam S, Ren M, Urtasun R (2021) Adversarial attacks on multi-agent communication. arXiv preprint arXiv:2101.06560
28. Zizzo G, Rawat A, Sinn M, Buesser B (2020) Fat: Federated adversarial training. arXiv preprint arXiv:2012.01791
29. Aono Y, Hayashi T, Wang L, Moriai S et al (2017) Privacy-preserving deep learning via additively homomorphic encryption. IEEE Trans Inf Forens Secur 13(5):1333–1345
30. Zari O, Xu C, Neglia G (2021) Efficient passive membership inference attack in federated learning. arXiv preprint arXiv:2111.00430

31. Hu H, Salcic Z, Sun L, Dobbie G, Zhang X (2021) Source inference attacks in federated learning. arXiv preprint arXiv: 2109.05659
32. Song L, Shokri R, Mittal P (2019) Membership inference attacks against adversarially robust deep learning models. In: 2019 IEEE security and privacy workshops (SPW), pp. 50–56. IEEE
33. Chen S, Kahla M, Jia R, Qi G.-J (2021) Knowledge-enriched distributional model inversion attacks. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 16178–16187
34. Wang Z, Song M, Zhang Z, Song Y, Wang Q, Qi H (2019) Beyond inferring class representatives: user-level privacy leakage from federated learning. In: IEEE INFOCOM 2019-IEEE conference on computer communications, pp. 2512–2520 . IEEE
35. Yin H, Mallya A, Vahdat A, Alvarez J.M, Kautz J, Molchanov P (2021) See through gradients: image batch recovery via GradInversion
36. Portability H.-H.I, Act A (1996) Health information privacy. https://www.ncbi.nlm.nih.gov/books/NBK9573/
37. FDA: Artificial intelligence and machine learning in software as a medical device. https://www.fda.gov/media/145022/download (2021)
38. Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan H.B, Patel S, Ramage D, Segal A, Seth K (2017) Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security. CCS '17, pp. 1175–1191. Association for Computing Machinery, New York, NY, USA . https://doi.org/10.1145/3133956.3133982
39. Wang H, Yurochkin M, Sun Y, Papailiopoulos D, Khazaeni Y (2020) Federated learning with matched averaging. ArXiv 2002.06440
40. Arivazhagan M.G, Aggarwal V, Singh A, Choudhary S (2019) Federated learning with personalization layers. ArXiv 1912.00818
41. Phong LT, Aono Y, Hayashi T, Wang L, Moriai S (2018) Privacy-preserving deep learning via additively homomorphic encryption. IEEE Trans Inf Forens Secur 13(5):1333–1345. https://doi.org/10.1109/TIFS.2017.2787987
42. Gentry C (2010) Computing arbitrary functions of encrypted data. Commun ACM 53:97–105
43. Acar A, Aksu H, Uluagac AS, Conti M (2018) A survey on homomorphic encryption schemes. ACM Comput Surveys (CSUR) 51:1–35
44. Canetti R, Feige U, Goldreich O, Naor M (1996) Adaptively secure multi-party computation. In: STOC '96
45. Dwork C (2006) Differential privacy. In: Bugliesi M, Preneel B, Sassone V, Wegener I (eds) Automata, languages and programming. Springer, Berlin, Heidelberg, pp 1–12
46. Dwork C (2008) Differential privacy: a survey of results. In: TAMC
47. Dwork C, Roth A (2014) The algorithmic foundations of differential privacy. Found Trends Theor Comput Sci 9:211–407
48. Dwork C, Kenthapadi K, McSherry F, Mironov I, Naor M (2006) Our data, ourselves: privacy via distributed noise generation. In: EUROCRYPT
49. Wei K, Li J, Ding M, Ma C, Yang HH, Farokhi F, Jin S, Quek TQS, Vincent Poor H (2020) Federated learning with differential privacy: algorithms and performance analysis. Trans Inf. For Sec 15, 3454–3469 . https://doi.org/10.1109/TIFS.2020.2988575
50. Naseri M, Hayes J, Cristofaro E.D (2020) Toward robustness and privacy in federated learning: experimenting with local and central differential privacy. ArXiv 2009.03561
51. Truex S, Liu L, Chow K.-H, Gursoy M.E, Wei W (2020) Ldp-fed: federated learning with local differential privacy. In: Proceedings of the third ACM international workshop on edge systems, analytics and networking
52. Hu R, Guo Y, Li H, Pei Q, Gong Y (2020) Personalized federated learning with differential privacy. IEEE Internet Things J 7:9530–9539
53. Bonawitz K, Eichner H, Grieskamp W, Huba D, Ingerman A, Ivanov V, Kiddon C, Konecný J, Mazzocchi S, McMahan H.B, Overveldt T.V, Petrou D, Ramage D, Roselander J (2019) Towards federated learning at scale: system design. ArXiv 1902.01046
54. Sahu AK, Li T, Sanjabi M, Zaheer M, Talwalkar AS, Smith V (2020) Federated optimization in heterogeneous networks
55. Nguyen TG, Phan TV, Hoang DT, Nguyen TN, So-In C (2021) Federated deep reinforcement learning for traffic monitoring in SDN-based IoT networks. IEEE Trans Cogn Commun Netw 7(4):1048–1065
56. Mowla NI, Tran NH, Doh I, Chae K (2020) AFRL: Adaptive federated reinforcement learning for intelligent jamming defense in FANET. J Commun Netw 22(3):244–258
57. Wang X, Garg S, Lin H, Hu J, Kaddoum G, Piran MJ, Hossain MS (2021) Towards accurate anomaly detection in industrial internet-of-things using hierarchical federated learning. IEEE Internet Things J
58. Chen F, Luo M, Dong Z, Li Z, He X (2018) Federated meta-learning with fast convergence and efficient communication. arXiv: 1802.07876
59. Shokri R, Stronati M, Song C, Shmatikov V (2017) Membership inference attacks against machine learning models. In: 2017 IEEE symposium on security and privacy (SP)
60. Song C, Ristenpart T, Shmatikov V (2017) Machine learning models that remember too much. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security, CCS
61. Wu S, Tang Y, Zhu Y, Wang L, Xie X, Tan T (2019) Session-based recommendation with graph neural networks. In: AAAI
62. Gupta P, Garg D, Malhotra P, Vig L, Shroff G.M (2019) NISER: Normalized item and session representations to handle popularity bias. arXiv: 1909.04276 Retrieval
63. McMahan HB, Moore E, Ramage D, Hampson S, y Arcas BA (2017) Communication-efficient learning of deep networks from decentralized data. In: AISTATS