A Survey on Federated Learning Poisoning Attacks and Defenses

Pratham patel (20IT104)

*Department of Information Technology*

*C S Patel Institute of Technology, CHARUSAT, Changa, India*

*20it104@charusat.edu.in*

# Abstract:

Federated Learning (FL) is an innovative decentralized machine learning paradigm that enables collaborative model training across multiple edge devices or nodes while preserving data privacy. In FL, the training process takes place locally on each node using its own data, and only model updates are shared instead of raw data. This approach addresses concerns related to data privacy and security, making it particularly suitable for sensitive applications such as healthcare and finance. This abstract provides an overview of federated learning, emphasizing its key characteristics, advantages, and potential applications. By enabling distributed learning without compromising data privacy, federated learning represents a promising direction for the future of machine learning in a privacy-conscious world.

# Introduction

Federated Learning (FL) has emerged as a groundbreaking paradigm in the field of machine learning, offering a distributed approach to collaborative model training while ensuring data privacy and security. The conventional centralized model training, where data is gathered at a central server, has limitations, such as privacy concerns, communication overhead, and the need for large-scale data transfer. Federated Learning addresses these challenges by enabling model updates to be computed locally on individual edge devices, avoiding the need to share raw data while fostering collaborative learning across multiple participants.

In Federated Learning, the process begins with the distribution of a global model to participating nodes, which can be mobile devices, IoT devices, or any other edge computing entity. Each node then performs model training locally on its own data, learning from its unique data distribution, and generating model updates. These updates, rather than raw data, are sent back to a central server, where they are aggregated to form an improved global model. This iterative process continues, refining the global model with insights from diverse local data, without compromising data privacy.

FL offers several significant advantages over traditional centralized approaches. Firstly, it addresses data privacy concerns by keeping sensitive data localized on edge devices, reducing the risk of data breaches and ensuring compliance with privacy regulations. Secondly, it reduces communication overhead since only model updates are transmitted, alleviating bandwidth constraints and making FL suitable for resource-constrained devices with limited connectivity. Additionally, FL can leverage the diversity of data from multiple sources, resulting in a more robust and representative global model compared to models trained on a single, centralized dataset.

However, Federated Learning also faces several challenges that warrant careful consideration. The heterogeneity of data across nodes may lead to a phenomenon known as non-independent and identically distributed (non-IID) data, affecting model convergence and performance. Balancing the need for preserving data privacy while maximizing the utility of the global model is a delicate trade-off that requires innovative privacy-preserving techniques such as differential privacy and secure aggregation.

Furthermore, the decentralized nature of FL introduces new attack vectors that threaten the integrity and security of the learning process. Adversarial participants may attempt to manipulate model updates to inject malicious behavior, leading to model poisoning attacks. Moreover, FL is susceptible to privacy attacks like model inversion and membership inference, where adversaries infer sensitive information about individual training samples or participants. These attacks underscore the critical importance of developing robust defenses to protect against potential breaches and maintaining the integrity of the collaborative learning environment.

Despite these challenges, Federated Learning holds immense promise across various domains and applications. In healthcare, FL enables collaborative disease prediction models while ensuring the privacy of sensitive patient data. In finance, it facilitates fraud detection and risk assessment without compromising the confidentiality of financial transactions. The Internet of Things (IoT) domain benefits from FL by empowering smart devices to collectively improve their performance without sharing raw sensor data. Furthermore, in natural language processing, FL enables collaborative language models for translation and speech recognition without compromising user privacy.

This research paper aims to explore the working principles of Federated Learning, delve into the challenges it faces, and present real-world applications. Additionally, it investigates potential attacks on FL and examines the state-of-the-art defenses to mitigate these risks. By understanding the intricacies of Federated Learning and its applications, we can harness its transformative potential while addressing its vulnerabilities, thus fostering a privacy-conscious and collaborative future for machine learning and artificial intelligence.

1. **Working of Federated Learning:**

The typical process of federated learning involves the following steps:

1. Initialization: A central server or coordinator selects an initial model or a set of model parameters to be shared with the participating devices. This initial model can be a pre-trained model or randomly initialized weights.
2. Device Training: Each participating device, such as a user's smartphone or edge device, performs local training using its own locally available data. The devices train the model using various machine learning algorithms, such as gradient descent, on their respective datasets.
3. Model Update: After completing the local training, the devices compute the updates to the model parameters based on the differences between the local model's performance and the desired outcome. These updates typically represent the gradients of the loss function.
4. Aggregation: The devices send their model updates securely to the central server or coordinator. The server aggregates the updates from multiple devices using techniques such as weighted averaging or cryptographic protocols to protect the privacy of individual updates.
5. Model Update and Iteration: The server receives the aggregated model updates and applies them to the current global model. It updates the global model parameters by incorporating the aggregated updates from the participating devices.
6. Repeating the Process: Steps 2 to 5 are repeated iteratively for a predefined number of rounds or until the model converges to a satisfactory level of performance. In each round, the participating devices train their local models, send updates to the central server, and receive the updated global model.

By repeating this process, federated learning leverages the collective knowledge and insights from multiple devices while preserving data privacy. The local training occurs on the devices themselves, which helps protect sensitive data as it never leaves the local device. The central server coordinates the training process, but it does not have access to individual device data.

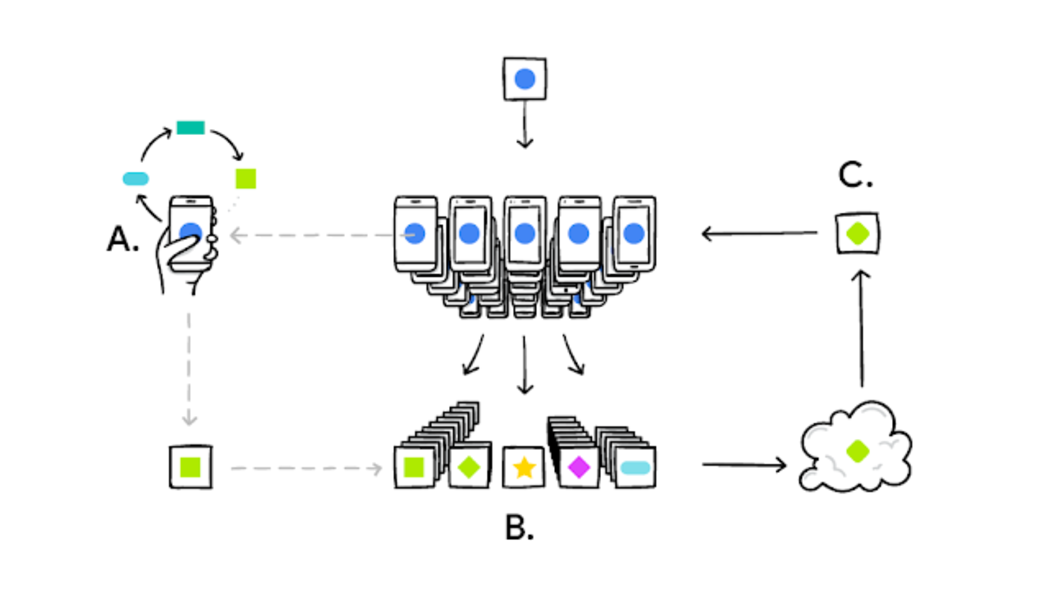


Figure 1: Working process of federated learning

# Components of federated learning:

Federated learning involves several key components that work together to enable the collaborative training of machine learning models while maintaining data privacy. The main components of federated learning are as follows:

1. Central Server or Coordinator: The central server or coordinator initiates and manages the federated learning process. It coordinates communication between the participating devices, receives model updates from the devices, aggregates them, and distributes the updated global model.
2. Participating Devices: These are the local devices that hold the data and perform local training. They can be smartphones, edge devices, or any other devices that have data and computational capabilities. Each participating device trains a local model using its own data, computes the model updates, and sends them to the central server for aggregation.
3. Local Data: Local devices possess their own data, which is typically sensitive and not shared with the central server or other devices. The data can be in various forms, such as user activity logs, sensor readings, or healthcare records. The local data remains on the devices and is used for local model training.
4. Local Model: Each participating device trains a local model using its local data. This model is initialized with the global model provided by the central server. The local model is trained using local computations and updates, and it captures insights from the local data while respecting privacy constraints.
5. Model Updates: After local training, the participating devices generate model updates based on the differences between their local models' performance and the desired outcome. These updates typically represent the gradients of the loss function with respect to the model parameters.
6. Aggregation Algorithm: The central server employs an aggregation algorithm to combine the model updates received from the participating devices. The aggregation can involve techniques such as weighted averaging, where the contribution of each device's update is weighted based on factors like device capabilities or data quality. The aggregation process aims to preserve privacy by not directly accessing or revealing individual device updates.
7. Global Model: The global model represents the collective knowledge and insights learned from the participating devices. The central server incorporates the aggregated model updates into the global model, which is then distributed back to the participating devices for the next round of local training.

These components work in a cyclic manner, with multiple rounds of local training, model updates, aggregation, and global model distribution, leading to collaborative learning while preserving data privacy.

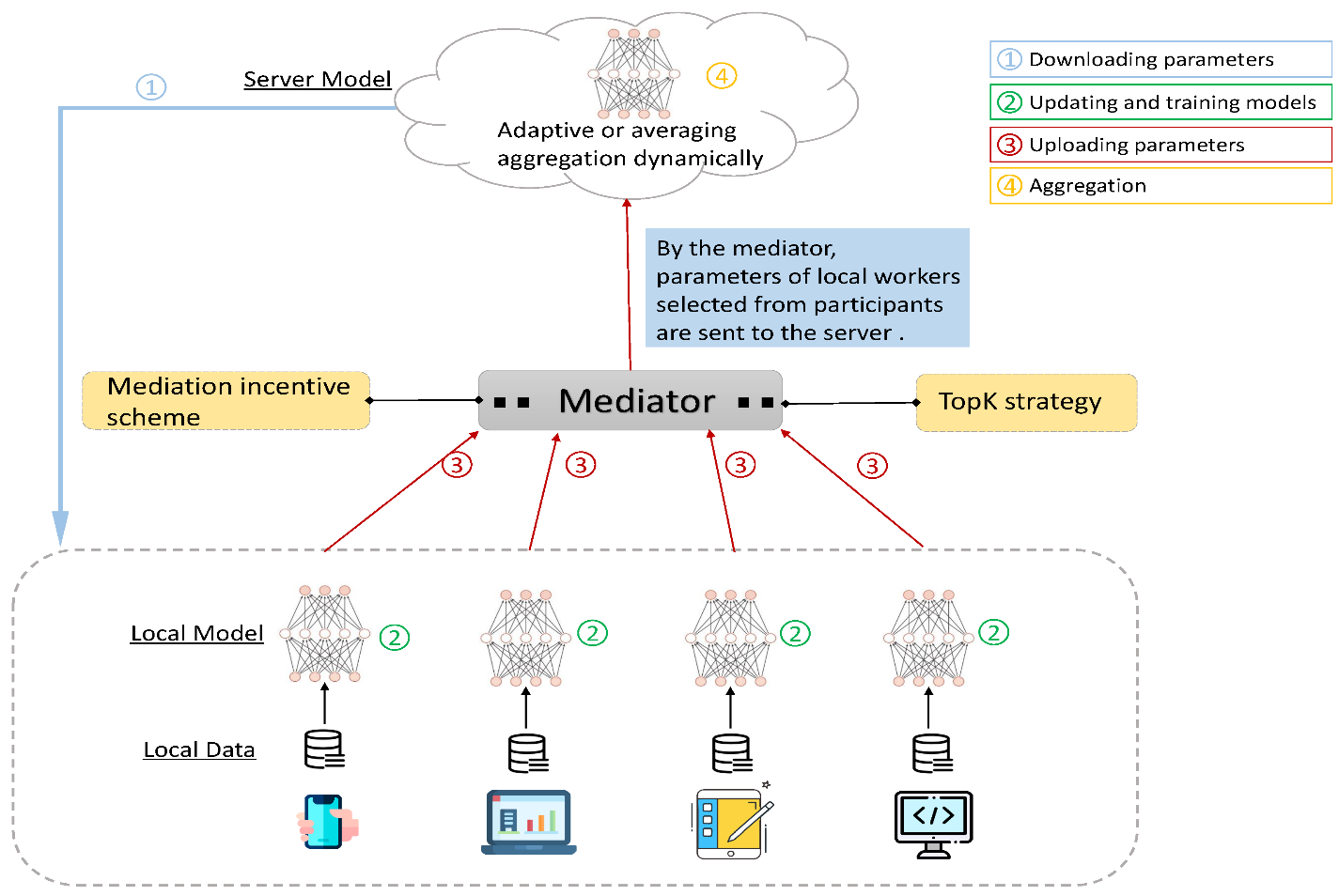


Figure 2: Components structure of federated learning

1. **TRAINING MODEL(ML model/ DL model) IN FEDERATED WAY:**

The Training a machine learning or deep learning model in a federated way involves specific considerations and steps. Here's a high-level overview of the process:

1. Determine the Federated Learning Approach: Choose the appropriate federated learning approach based on your requirements. This can include methods such as Federated Averaging, Federated SGD, or more advanced techniques like Secure Multi-Party Computation (MPC) or Homomorphic Encryption, depending on the level of privacy and security required.
2. Setup Communication and Infrastructure: Establish the communication channels and infrastructure needed for coordination between the central server and participating devices. This typically involves ensuring secure and efficient data transfer protocols and mechanisms for device registration, update exchange, and model distribution.
3. Initialize the Global Model: The central server initializes the global model by selecting a pre-trained model or randomly initializing the model parameters. This initial model is shared with the participating devices.
4. Device Training and Local Model Updates: Each participating device performs local training using its own local data. The device trains the model using standard machine learning or deep learning algorithms, such as gradient descent, stochastic gradient descent (SGD), or more advanced optimization algorithms. After training, the device computes the model updates, which typically involve calculating gradients of the loss function with respect to the model parameters.
5. Secure Model Update Aggregation: The participating devices send their model updates securely to the central server. The central server employs an aggregation algorithm, such as Federated Averaging, to aggregate the updates. The aggregation can include techniques like weighted averaging or cryptographic protocols to ensure privacy and security.
6. Update the Global Model: The central server receives the aggregated model updates and applies them to the current global model. This update process adjusts the global model parameters based on the aggregated updates received from the devices.
7. Iterative Training: Repeat steps 4 to 6 for multiple rounds or until convergence is achieved. In each round, the participating devices perform local training, compute model updates, send updates to the central server, and receive the updated global model. The iterative process allows the model to improve with the collective knowledge of all participating devices.
8. Evaluation and Deployment: After the federated training process, evaluate the performance of the trained model. Validate it on separate evaluation datasets to ensure its effectiveness. Once satisfied with the performance, deploy the model for inference on new data.

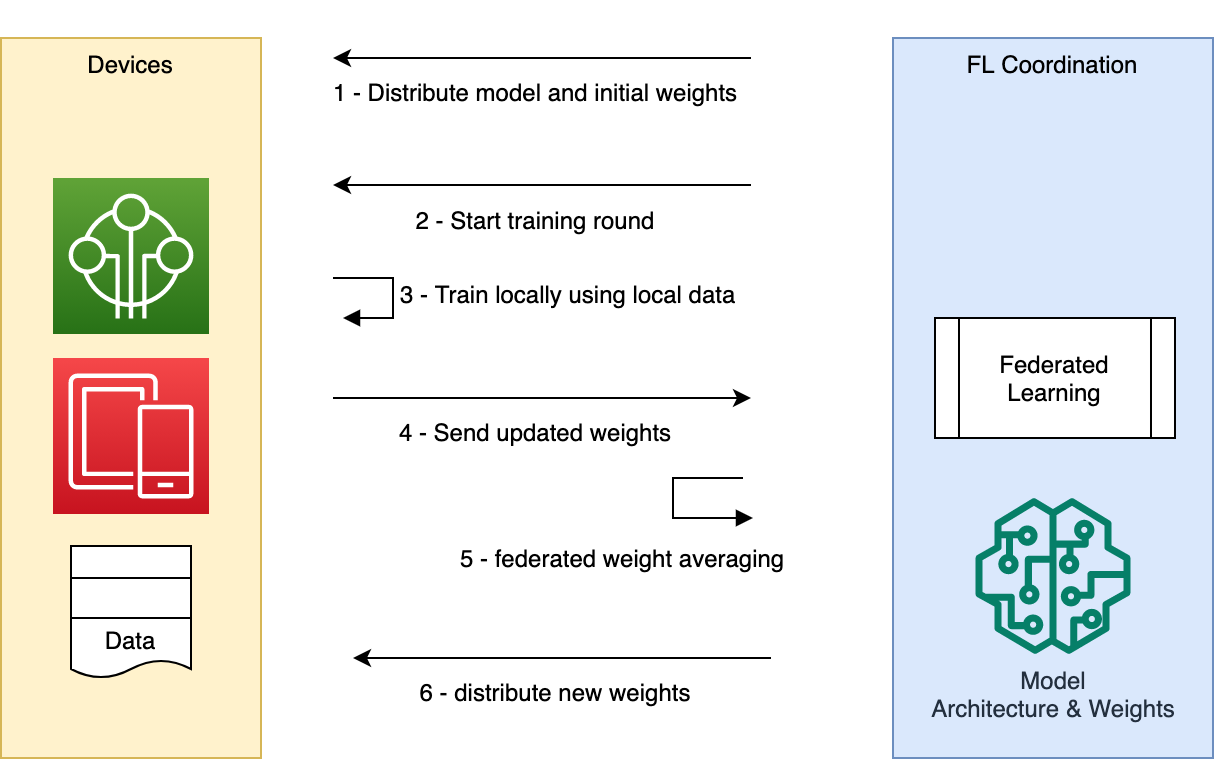


Figure 3: Steps for training model

# Poisoning attacks

Federated learning is designed to preserve the privacy of individual user data, but it is still vulnerable to certain privacy attacks. Here are some common types of attacks that can be performed on federated learning systems:

1. Model Inversion Attacks: In model inversion attacks, an adversary tries to reconstruct or infer sensitive data from the global model. By analyzing the outputs of the model, an attacker can attempt to reverse-engineer private information, such as images, text, or user preferences. This attack exploits the leakage of information through model outputs.

* Brief: Adversaries reconstruct sensitive data from the global model's outputs.
* Advantages: Allows extraction of specific details from the model, enabling precise inference of private information.
* Disadvantages: Requires significant computational resources and access to the model's outputs. Accuracy of the reconstructed data depends on the model's complexity and quality.
* Limitations: May not be effective against models with strong privacy defenses or complex data representations.

1. Membership Inference Attacks: In membership inference attacks, an attacker aims to determine whether a specific data point was used in the training dataset of the global model. By analyzing the model's responses to certain inputs, an adversary can infer whether a particular data point was present in the training data, potentially violating the privacy of individuals.

* Brief: Attackers determine whether specific data points were part of the training dataset.
* Advantages: Enables identification of training data, potentially leading to privacy breaches for individuals.
* Disadvantages: Accuracy of membership inference depends on the attacker's knowledge and the model's behavior. Requires access to the model's responses to specific inputs.
* Limitations: Performance of membership inference can be affected by the presence of data augmentation, randomization techniques, or other privacy-enhancing measures.

1. Model Poisoning Attacks: Model poisoning attacks involve injecting malicious data or maliciously crafted updates into the federated learning process. By manipulating the model updates sent by a participating device, an attacker can introduce biases or compromise the integrity and performance of the global model.

* Brief: Malicious data or updates are injected to compromise the integrity or performance of the global model.
* Advantages: Can manipulate the model to produce desired outcomes, such as biased predictions or system failure.
* Disadvantages: Requires the attacker to have access to the federated learning process and may require collusion with participating devices.
* Limitations: Effective poisoning attacks may be challenging due to limited access to the federated learning system and the need to bypass defenses.

1. Byzantine Attacks: Byzantine attacks refer to scenarios where participating devices intentionally deviate from the protocol or act maliciously. This can include devices sending incorrect or manipulated model updates, dropping out during training rounds, or colluding with other devices to disrupt the federated learning process.

* Brief: Participating devices intentionally deviate from the protocol or act maliciously.
* Advantages: Can disrupt the federated learning process, compromise the global model, or influence the learning outcome.
* Disadvantages: Requires collusion or control over multiple participating devices. Detection and mitigation can be challenging.
* Limitations: Byzantine attacks can be mitigated through techniques like robust aggregation algorithms, redundancy, and fault tolerance mechanisms.

1. Eavesdropping Attacks: Eavesdropping attacks involve intercepting the communication between the central server and participating devices to gather sensitive information. Adversaries attempt to access model updates, gradients, or other data exchanged during the federated learning process to gain insights into private user data.

* Brief: Attackers intercept the communication between the central server and participating devices.
* Advantages: Provides access to sensitive information exchanged during the federated learning process.
* Disadvantages: Requires sophisticated techniques to intercept and decrypt communication. May be mitigated by secure communication protocols.
* Limitations: Secure communication protocols, encryption, and other defenses can mitigate the impact of eavesdropping attacks.

1. Model Stealing Attacks: Model stealing attacks aim to steal the entire global model or a replica of it. An attacker, pretending to be a participating device, may request the model from the central server and use it for their own purposes, potentially undermining the intellectual property of the model owner.

* Brief: Adversaries attempt to steal the entire global model or a replica of it.
* Advantages: Enables unauthorized access to the model's parameters and intellectual property.
* Disadvantages: Requires mimicking the behavior of a participating device and can be challenging if strong security measures are in place.
* Limitations: Model stealing attacks can be mitigated through techniques like watermarked models, secure enclaves, and hardware-based protections.

1. Data Poisoning Attacks: Data poisoning attacks involve manipulating the local data of participating devices to inject biased or malicious samples into the federated learning process. By including carefully crafted data, an attacker can influence the global model to produce desired outcomes that may be detrimental or discriminatory.

* Brief: Manipulated data is injected to influence the global model's behavior or introduce biases.
* Advantages: Can manipulate the learning process and induce specific outcomes or biases in the model.
* Disadvantages: Requires access to participating devices or the central server to inject the poisoned data. Detection can be challenging.
* Limitations: Detection of data poisoning attacks is an ongoing challenge, and robust defenses against poisoning attacks are actively researched.

1. Differential Privacy Breaches: Federated learning often employs differential privacy techniques to protect the privacy of individual data points. However, adversaries may attempt to breach the differential privacy guarantees by analyzing the outputs of the global model and exploiting statistical leakage to infer sensitive information about individuals.

* Brief: Attackers breach the privacy guarantees of differential privacy mechanisms.
* Advantages: Allows inference of sensitive information through statistical leakage in the model's outputs.
* Disadvantages: Requires sophisticated analysis and knowledge of the differential privacy mechanisms in use. Accuracy of the inference may vary.
* Limitations: Differential privacy breaches can be mitigated through tighter privacy budgets, noise injection mechanisms, and adaptive privacy defenses.

1. Free-Riding Attacks:

Free-riding attacks in federated learning refer to scenarios where participating devices maliciously withhold their contribution or make minimal effort during the training process, exploiting the contributions of other honest participants without providing meaningful updates

* Brief: Participating devices intentionally avoid contributing meaningful updates to the global model, relying on others' efforts.
* Advantages: Attackers benefit from the improvements in the global model without investing computational resources or sharing their local data.
* Disadvantages: Free-riding reduces the effectiveness and efficiency of federated learning by burdening honest participants and hindering convergence.
* Limitations: Detecting free-riding attacks can be challenging due to variations in computational resources, network conditions, and varying levels of participation.

The advantages, disadvantages and limitations of poisoning attacks:

|  |  |  |
| --- | --- | --- |
| **ADVANTAGES** | **DISADVANTAGES** | **LIMITATIONS** |
| * Extraction of sensitive information: Privacy attacks can successfully extract sensitive information from the global model or the federated learning process, providing adversaries with access to private data. | * Ethical and legal concerns: Privacy attacks infringe upon users' privacy rights and ethical principles, leading to potential legal ramifications for the attackers. | * Attack complexity: Many privacy attacks require significant computational resources, access to model outputs, or control over multiple participating devices, limiting their practicality and scalability. |
| * Manipulation of outcomes: Attackers can manipulate the global model's behavior or predictions through various attacks, such as model poisoning or data poisoning attacks, to achieve desired outcomes or introduce biases. | * Erosion of user trust: Privacy breaches can erode users' trust in federated learning systems, affecting adoption and participation. | * Detection challenges: Some attacks, such as model inversion or membership inference attacks, may be challenging to detect due to the subtle nature of the information leakage or the complexity of the attack vectors. |
| * Intellectual property theft: Attacks like model stealing enable unauthorized access to the model's parameters and intellectual property, which can be valuable for competitors or malicious entities. | * Reduced system performance: Attacks like free-riding or data poisoning can negatively impact the effectiveness and efficiency of federated learning, leading to degraded system performance. | * Mitigation difficulties: Defending against privacy attacks often involves developing robust defenses, implementing secure protocols, and continuously monitoring the system, which can be challenging and resource-intensive. |
|  | * Data integrity compromise: Attacks targeting the global model, such as model poisoning or Byzantine attacks, can compromise the integrity of the model and the accuracy of predictions. | * Evolving attack techniques: Attackers continuously adapt their strategies and exploit new vulnerabilities, necessitating ongoing research and development of privacy-preserving techniques. |

Table 1: Advantages, disadvantages and limitations of Poisoning attacks

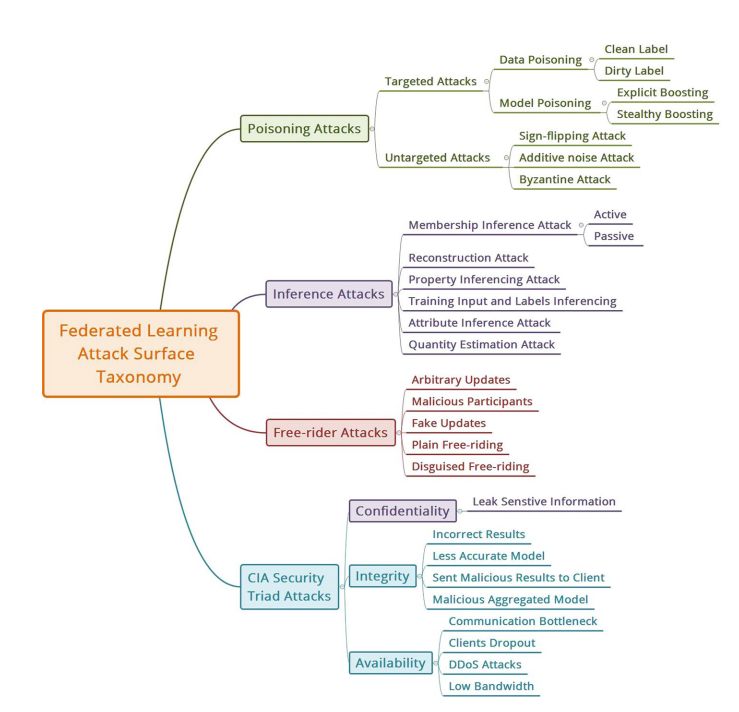
****

Figure 4: Various poisoning attacks

# Defenses against poisoning attacks

Poisoning attacks in the context of Federated Learning (FL) can manifest in various forms, each posing a unique threat to the integrity of the collaborative learning process. To protect against these attacks, researchers have been actively developing a range of defense mechanisms, specifically targeting data poisoning attacks, model poisoning attacks, sign-flipping attacks, additive noise attacks, byzantine attacks, and backdoor attacks.

1. Data Poisoning Attack Defense: Data poisoning attacks aim to corrupt the training process by introducing malicious or misleading data into the training datasets of participating nodes. One defense mechanism involves employing data sanitization techniques, where nodes thoroughly clean and validate their local datasets before contributing updates to the global model. This includes outlier detection, data filtering based on consistency checks, and data scrubbing to identify and remove potentially tainted samples. Furthermore, employing robust aggregation techniques at the central server helps mitigate the impact of poisoned data by using weighted averaging or median aggregation, thereby reducing the influence of outliers and malicious samples on the global model.
2. Model Poisoning Attack Defense: Model poisoning attacks involve adversaries injecting poisoned model updates into the FL system, aiming to manipulate the global model's behavior. One prominent defense approach is to implement robust optimization techniques during model aggregation. Robust aggregation algorithms, such as Robust Federated Optimization, aim to detect and mitigate the impact of poisoned model updates by assigning lower weights to updates that exhibit anomalous behavior during aggregation. Additionally, employing anomaly detection algorithms to identify model updates that deviate significantly from the expected distribution can help identify and neutralize potential model poisoning attempts.
3. Sign-Flipping Attack Defense: Sign-flipping attacks, where adversaries deliberately invert the signs of model gradients during the training process, can severely disrupt the convergence of the global model. To counteract this, secure aggregation methods like Secure Multi-Party Computation (SMPC) or Homomorphic Encryption can be employed to ensure that gradient updates are combined without revealing their individual contributions. Furthermore, validating the consistency and convergence behavior of model updates from different nodes can aid in identifying and excluding adversarial contributions.
4. Additive Noise Attack Defense: Additive noise attacks involve adversaries injecting random noise into their model updates to undermine the learning process. Employing robust aggregation techniques, such as trimmed averaging or rank-based aggregation, can help in identifying and neutralizing noisy updates, preventing their undue influence on the global model. Additionally, introducing adaptive learning rates or model-specific noise variance can enhance the model's resilience against these attacks.
5. Byzantine Attack Defense: Byzantine attacks involve adversarial nodes that deviate from the FL protocol by sending arbitrary or contradictory model updates to the central server. To defend against Byzantine attacks, Byzantine fault-tolerant aggregation algorithms can be employed, where majority voting or weighted voting schemes help in identifying and discounting malicious updates. Additionally, redundancy and replication of model updates from multiple nodes can aid in detecting and isolating Byzantine behavior.
6. Backdoor Attack Defense: Backdoor attacks involve adversaries poisoning the training data with subtle patterns or triggers that, when triggered, lead the model to produce erroneous outputs. To mitigate backdoor attacks, data sanitization techniques can be employed during data aggregation, identifying and filtering out potential backdoor triggers. Additionally, employing regularization techniques like defensive distillation can help in training models robust against these attacks by adding noise to the training process and reducing the model's sensitivity to poisoned data.
7. Robust Aggregation Techniques: One effective defense mechanism involves using robust aggregation techniques during the model update aggregation process at the central server. Traditional aggregation methods, such as averaging, are vulnerable to the influence of poisoned updates, leading to a compromised global model. To counteract this, robust aggregation techniques, such as trimmed averaging, median aggregation, or outlier detection, are employed to identify and discard potentially poisoned updates. These methods prevent the poisoned updates from exerting disproportionate influence on the global model, thereby enhancing its resistance to attacks.
8. Data Filtering and Validation: Pre-processing data at individual nodes is another crucial line of defense against poisoning attacks. By employing data filtering and validation techniques, nodes can identify potential outlier data points or suspicious model updates before sending them for aggregation. Nodes can employ statistical checks or anomaly detection algorithms to verify the validity and consistency of the data before incorporating it into the training process. This approach ensures that only reliable and authentic updates contribute to the global model, minimizing the impact of malicious data.
9. Secure Model Aggregation: Secure aggregation methods play a pivotal role in defending against poisoning attacks. Techniques like Secure Multi-Party Computation (SMPC) or Homomorphic Encryption enable nodes to collaboratively aggregate their model updates without revealing individual contributions. SMPC allows nodes to compute the aggregate without disclosing their updates, safeguarding against adversaries attempting to infer sensitive information from the updates. By preserving the privacy of individual contributions, secure model aggregation reinforces the confidentiality of the learning process, rendering poisoning attacks less effective.
10. Robust Federated Learning Frameworks: Implementing robust FL frameworks can significantly bolster defenses against poisoning attacks. These frameworks include mechanisms to detect and identify adversarial nodes based on their anomalous behavior or updates. Once identified, these nodes can be excluded from future iterations of the learning process, preventing their influence on the global model. Moreover, adaptive federated learning approaches can dynamically adjust the participation of nodes based on their trustworthiness, enhancing the resilience of the FL system against attacks.
11. Federated Averaging with Trust Scores: Introducing trust scores for individual nodes can provide an added layer of defense against poisoning attacks. By quantifying the reliability and reputation of each node, federated averaging can be weighted accordingly during aggregation. Nodes with higher trust scores receive more significant contributions to the global model, while those with lower trust scores have their influence minimized. Trust scores can be dynamically updated over time based on the node's performance and adherence to the FL protocol, further strengthening the system's resilience against adversarial behavior.

By integrating these defense mechanisms into the FL framework, organizations and researchers can enhance the resilience of collaborative learning systems against poisoning attacks. Nevertheless, as poisoning attacks continue to evolve, constant research and innovation are crucial to stay ahead of emerging threats and foster a secure and robust environment for federated machine learning.

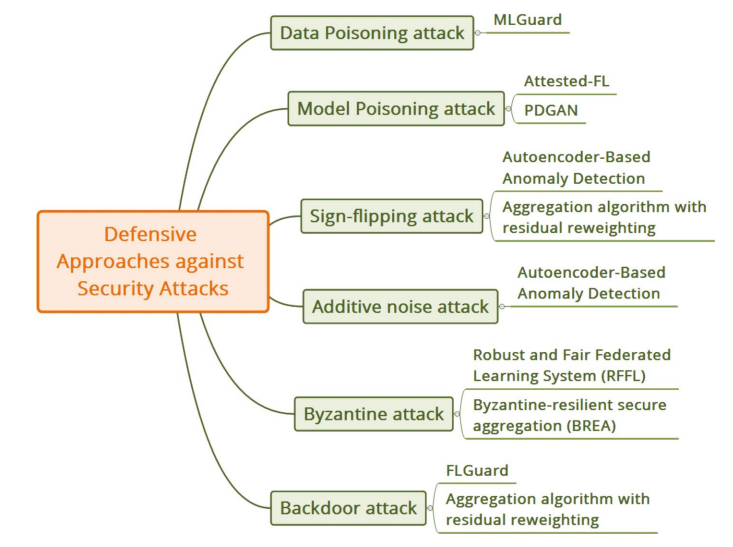
****

Figure 5: Various defense mechanisms solutions against poisoning attacks

# ISSUES/OPEN CHALLENGES IN FEDERATED LEARNING:

There Federated learning is a rapidly evolving field, and several open challenges and issues remain to be addressed to fully realize its potential. Here are some of the key challenges and open issues in federated learning:

1. Privacy and Security: Privacy is a critical concern in federated learning. Protecting sensitive user data and preventing unauthorized access or leakage during the model training and aggregation process is a significant challenge. Developing robust privacy-preserving techniques, secure communication protocols, and defenses against adversarial attacks are active areas of research.
2. Communication Efficiency: Federated learning requires frequent communication between the central server and participating devices. As the number of devices increases, communication overhead and bandwidth requirements become significant. Developing efficient communication protocols, compression techniques, and strategies to handle heterogeneous network conditions are important for scalable and practical federated learning.
3. Data Heterogeneity and Bias: Federated learning involves training models on data from multiple devices, which may have diverse distributions and characteristics. Handling data heterogeneity and ensuring fair representation across devices while mitigating biases are ongoing challenges. Techniques such as adaptive aggregation, weighted sampling, and data preprocessing methods are being explored to address these issues.
4. Model Aggregation and Convergence: Aggregating model updates from multiple devices and achieving convergence of the global model is a non-trivial task. Variations in local data, computation capabilities, and communication delays can lead to synchronization challenges and impact the convergence rate. Developing efficient and robust aggregation algorithms, adaptive learning rates, and convergence monitoring techniques are active research areas.
5. Resource-Constrained Devices: Federated learning involves training models on resource-constrained devices, such as smartphones or IoT devices. These devices often have limited computational power, memory, and energy resources. Developing lightweight and efficient machine learning algorithms, model compression techniques, and adaptive resource allocation methods are crucial for federated learning on such devices.
6. Federated Learning in Non-IID Settings: Federated learning traditionally assumes that data across devices are independently and identically distributed (IID). However, in real-world scenarios, data may exhibit non-IID characteristics due to variations in user preferences, demographics, or data collection processes. Extending federated learning to handle non-IID data is an ongoing research area.
7. Standardization and Interoperability: Federated learning is an emerging field, and there is a lack of standardized frameworks, protocols, and interfaces. Developing common standards and interoperability across different federated learning platforms and frameworks would enable easier collaboration, comparison of results, and wider adoption of federated learning techniques.
8. Fairness and Accountability: Ensuring fairness and accountability in federated learning is a significant challenge. Biases in data collection, model outcomes, or participation can impact the fairness and ethical implications of federated learning systems. Developing techniques for measuring and mitigating biases, establishing transparent and accountable governance mechanisms, and addressing the potential impact on marginalized or underrepresented groups are important considerations.

Addressing these challenges requires interdisciplinary research efforts, collaborations between academia and industry, and a focus on developing robust algorithms, privacy-enhancing techniques, and system-level optimizations.

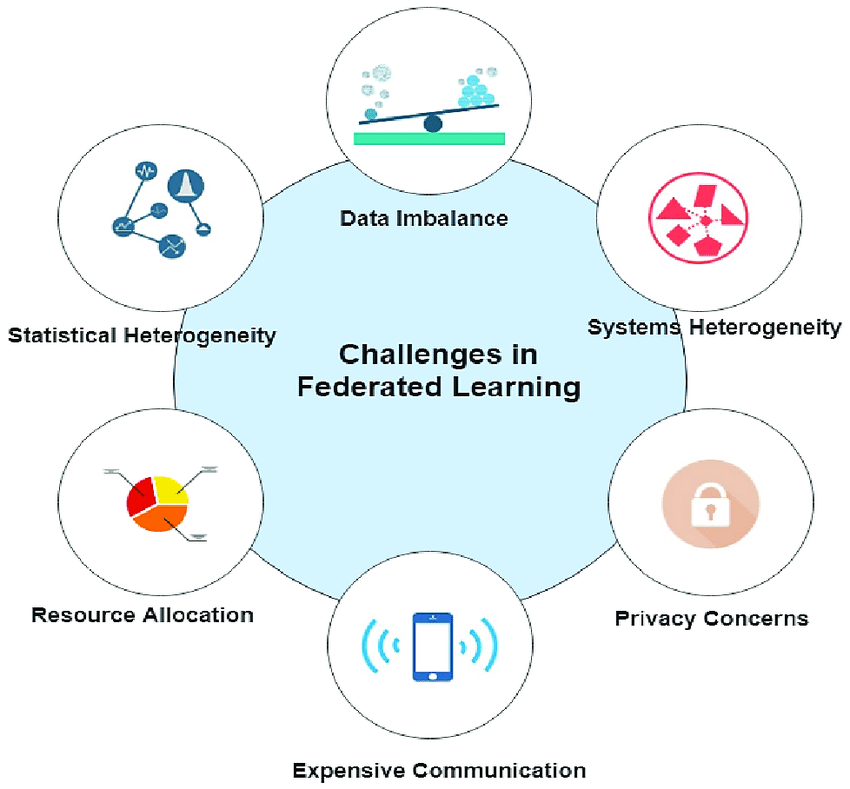


Figure 5: Challenges in federated learning

# Future Aspects:

1. Enhanced Privacy-Preserving Techniques: Advancements in privacy-preserving techniques, such as differential privacy and secure aggregation, will continue to strengthen data protection in federated learning. Improving the privacy guarantees for individual participants while maintaining model utility will be a crucial focus.
2. Handling Non-IID Data: Addressing the challenges posed by non-independent and identically distributed (non-IID) data remains an important future aspect. Developing innovative algorithms and methods to effectively learn from heterogeneous data distributions will be vital for improving model convergence and performance in federated settings.
3. Federated Transfer Learning: Exploring federated transfer learning techniques will enable knowledge transfer between domains and facilitate more efficient and faster model training on edge devices. This approach will help leverage knowledge learned from related tasks to improve the performance of models on new tasks.
4. Federated Learning in Edge Computing: As edge computing continues to evolve, federated learning will play a significant role in enabling AI and ML applications on resource-constrained edge devices. Optimizing federated learning algorithms to accommodate limited computation and communication resources will be a key area of research.
5. Robustness Against Advanced Attacks: Developing more sophisticated defense mechanisms to protect federated learning systems against advanced attacks, including adversarial examples, model inversion, and advanced poisoning attacks, will be crucial to ensure the security and integrity of the learning process.
6. Large-Scale Federated Learning: Scaling federated learning to large and diverse networks will present challenges in terms of communication efficiency, model aggregation, and system robustness. Future research will focus on devising scalable solutions to enable federated learning in massive distributed environments.
7. **Conclusion:**

The future of federated learning holds immense potential, driven by advancements in privacy-preserving techniques, handling non-IID data, expanding into edge computing, and developing robust defense mechanisms against advanced attacks. With ongoing research and collaborative efforts, federated learning will continue to revolutionize the field of distributed machine learning, unlocking new possibilities and applications while safeguarding data privacy and security.

# References

1. H. B. McMahan, E. Moore, D. Ramage, B. A. y Arcas, Feder- ated learning of deep networks using model averaging, arXiv preprint arXiv:1602.05629 2.
2. K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman,

V. Ivanov, C. Kiddon, J. Koneˇcny`, S. Mazzocchi, B. McMahan, et al., Towards federated learning at scale: System design, Proceedings of Ma- chine Learning and Systems 1 (2019) 374–388.

1. Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, B. He, A survey on federated learning systems: Vision, hype and reality for data privacy and protection, IEEE Transactions on Knowledge and Data Engineering.
2. J. Koneˇcny`, B. McMahan, D. Ramage, Federated optimization: Distributed optimization beyond the datacenter, arXiv preprint arXiv:1511.03575.
3. M. Chen, A. T. Suresh, R. Mathews, A. Wong, C. Allauzen, F. Beaufays,

M. Riley, Federated learning of n-gram language models, arXiv preprint arXiv:1910.03432.

1. S. Ramaswamy, R. Mathews, K. Rao, F. Beaufays, Federated learn- ing for emoji prediction in a mobile keyboard, arXiv preprint arXiv:1906.04329.
2. T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage,

F. Beaufays, Applied federated learning: Improving google keyboard query suggestions, arXiv preprint arXiv:1812.02903.

1. M. Liu, S. Ho, M. Wang, L. Gao, Y. Jin, H. Zhang, Federated learning meets natural language processing: A survey, arXiv preprint arXiv:2107.12603.
2. T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, W. Shi, Federated learning of predictive models from federated electronic health records, International journal of medical informatics 112 (2018) 59–67.
3. N. Rieke, J. Hancox, W. Li, F. Milletari, H. R. Roth, S. Albarqouni,

S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein, et al., The future of digital health with federated learning, NPJ digital medicine 3 (1) (2020) 1–7.

1. A. Saeed, T. Ozcelebi, J. Lukkien, Multi-task self-supervised learning for human activity detection, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 3 (2) (2019) 1–30.
2. P. Schmidt, A. Reiss, R. Duerichen, C. Marberger, K. Van Laerhoven, Introducing wesad, a multimodal dataset for wearable stress and affect detection, in: Proceedings of the 20th ACM international conference on multimodal interaction, 2018, pp. 400–408.
3. S. Silva, B. A. Gutman, E. Romero, P. M. Thompson, A. Altmann,

M. Lorenzi, Federated learning in distributed medical databases: Meta- analysis of large-scale subcortical brain data, in: 2019 IEEE 16th inter- national symposium on biomedical imaging, IEEE, 2019, pp. 270–274.

1. A. Supratak, H. Dong, C. Wu, Y. Guo, Deepsleepnet: A model for automatic sleep stage scoring based on raw single-channel eeg, IEEE Transactions on Neural Systems and Rehabilitation Engineering 25 (11) (2017) 1998–2008.
2. J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, F. Wang, Federated learning for healthcare informatics, Journal of Healthcare Informatics Research 5 (1) (2021) 1–19.