Azerbaijan Journal of High Performance Computing, Vol 5, Issue 2, 2022, pp. 273-285
https://doi.org/10.32010/26166127.2022.5.2.273.285

A Survey on Challenges of Federated Learning

Aliyev S.I.
Azerbaijan State Oil and Industry University, Baku, Azernaijan, aliyevs320@gmail.com

*Correspondence: Aliyev S.I., Azerbaijan State Oil and Industry University, Baku, Azerbaijan, aliyevs320@gmail.com

Abstract

Federated Learning is a new paradigm of Machine Learning. The main idea behind FL is to provide a decentralized approach to Machine Learning. Traditional ML algorithms are trained in servers with data collected by clients, but data privacy is the primary concern. This is where FL comes into play: all clients train their model locally and then share it with a global model in the server and receive it back. However, FL has problems, such as possible cyberattacks, aggregating most appropriately, scaling the non-IID data, etc. This paper highlights current attacks, defenses, pros and cons of aggregating methods, and types of non-IID data based on publications in this field.

Keyword: Federated learning, Challenges of FL, Aggregation methods in FL, Attacks and vulnerabilities, Defenses, non-iid data.

1. Introduction

As Machine Learning spreads more and more, new challenges are faced. Another trend that has been observed during the last years is the IoT (Al Hayajneh, A., Bhuiyan, M. Z. A., & McAndrew, I. 2020). Today, more and more devices are included in business, sciences, etc., which collect data and help people in decision making. To make the most interest in those fields, these devices send data to the server to one global model to make a better Machine Learning model. However, in this case, data privacy issue comes into play. The price of data is increasing, and sometimes sharing this data is not in the interest of the data owner. Federated Learning overcomes this problem (Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. 2020). The main idea behind FL is that the global model can be trained without direct access to data if the devices can train the model locally (Sun, T., Li, D., & Wang, B. 2022). In this case, models are trained in devices, and the trained model is sent to a central server. The central server aggregated these models and sent them back to the client. The process is shown in Figure 1.

Thanks to increased calculation power and storage of devices, this has become even easier nowadays. The calculation and storage of processes in local devices and not on servers is called edge computing and several works were already done in this field.

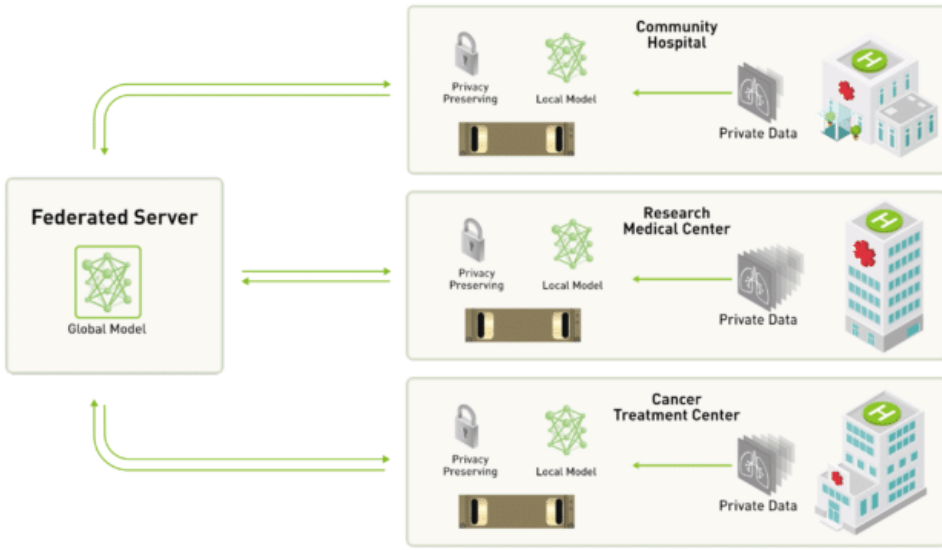

Fig. 1: Federated Learning process

However, FL comes with challenges. One of the main challenges FL face is aggregation (Ek, S., Portet, F., Lalanda, P., & Vega, G.,2020). Evaluation of federated learning aggregation algorithms: application to human activity recognition. As discussed earlier, clients train models and send them to the server to aggregate. However, these aggregations are not easy since most of the ML problems are a black box, and it is hard to determine what is going on during the process. Several aggregation methods have been proposed which will be discussed in the following sections. Although they do what they must, they could be more flawless, and each has disadvantages.

Another problem FL face is cyberattacks (Blanco-Justicia, A., Domingo-Ferrer, J., Martínez, S., Sánchez, D., Flanagan, A., & Tan, K. E.,2021). When many devices can access the server, one must consider that it opens doors for different attacks. These attacks may be targeted on the server or the global model (Zhang, J., Zhu, H., Wang, F., Zhao, J., Xu, Q., & Li, H. 2022). Some of these attacks are very hard to detect and, when not prevented, can drastically decrease the global model's performance. In the third section of this paper, vulnerabilities of FL, different attack threads, and defense measures will be discussed.

Handling non-IID data is also challenging for FL (Gao, L., Fu, H., Li, L., Chen, Y., Xu, M., & Xu, C. Z. 2022). Most machine learning algorithms are designed to work with IID data. So, the FL algorithms must be modified to work effectively with non-IID data. Since there are different types of non-IID data handling them requires different methods. In the fourth section, the types of non-IID data will be discussed.

2. Aggregation Methods

2.1. Federated SGD

The Federated SGD method is the most basic and naive approach (Chen, Y., Sun, X., & Jin, Y.,2019). It starts by randomly initializing the neural network's weights randomly in the server. This neural network is sent to each client. Training is done in clients and sent back to the central node. This is called a communication round. After each communication round, averaging takes place in the server.

$$w_{t+1} \leftarrow w_t - \alpha \sum_{k=1}^{K} g_k \frac{n_k}{n}$$

Where α is the step length, gk is gradients, nk is the data in client k, and n is the number of clients.

There are two main approaches to federated averaging:

1. Update gradients in client: In this approach, gradients are calculated, and weights are updated. Then the newly updated model is sent back to the server, and all new weights are averaged.

2. Compute gradients in client: In this approach, gradients are calculated in the client, and only new gradients are sent to the server. Gradients are averaged in the server and used to update models.

Overall FedSGD method performs poorly because of its naive approach. The reason is that all weights are calculated separately, and different neurons may be optimized for different purposes in different clients.

2.2. Federated Averaging (FedAVG)

The Federated averaging method is similar to Federated SGD, where gradients are updated in clients. This is done by:

$$For\ each\ client\ k, w_{t+1}^{k} \leftarrow w_t - \alpha\, g_k$$

$$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n}\, w_{t+1}^{k}$$

When we do this, we can update gradients in clients multiple times before being sent back to the central node. When updating is done multiple times, it is called the Federated Averaging algorithm (Sannara, E. K., Portet, F., Lalanda, P., & German, V. E. G. A. 2021). The main advantage of the FedAVG algorithm is that each client can update the weights parallel, and it becomes a faster algorithm.

2.3. Federated Learning with Personalized Layers (Feder)

FedPer algorithm works similarly to FedAVG for computing weights in aggregating model. The main difference is how layers are approached. In the FedPer algorithm,

unlike FedAVG, not all layers are aggregated. In neural networks, higher levels play a more critical role in decision making, whereas lower-level layers extract more available features. FedPer algorithm can be considered an adaptation of transfer learning (Pan, S. J., & Yang, Q. 2010) since it uses the same idea to freeze already trained lower-level layers and train only higher-level layers for prediction. In the FedPer algorithm, higher levels are client specific and are not aggregated in the server (Wu, Q., He, K., & Chen, X.,2020). Only lower levels are trained in the server, which helps clients handle different inputs and extract general information. Figure 2 demonstrates the structure of the FedAVG algorithm.

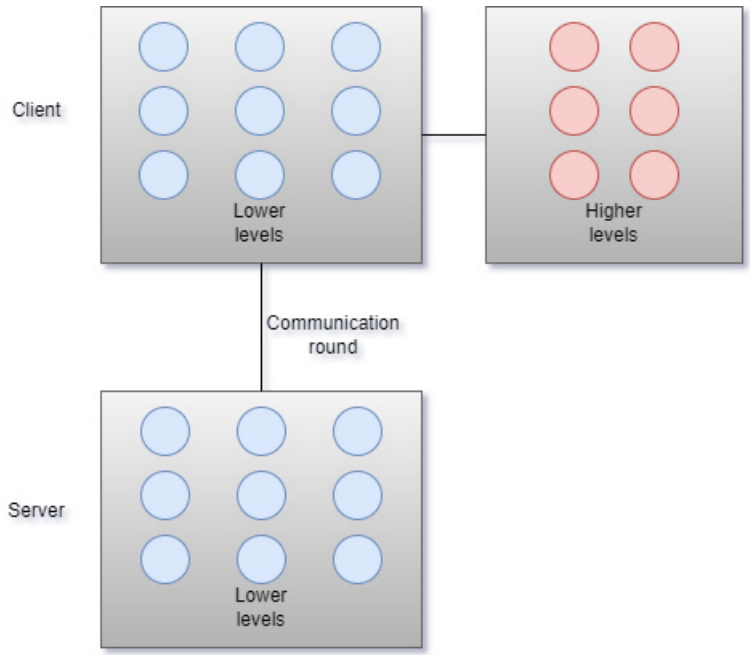

Fig. 2: FedAVG algorithm structure

Figure 2 shows that the model on the server cannot make the decision by itself since it misses higher and output layers.

2.4. Federated Matched Averaging

FedMA algorithm modifies the neural network so that new similar neurons can be merged and new neurons not similar to others can be added (Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., & Khazaeni, Y. 2020). This is done by layer-wise aggregation process. For the FedMA algorithm, the number of neurons in a layer is a sub-problem to solve rather than a hyper-parameter to be defined. The main idea behind FedMA is that all clients have similar neurons and can be merged. This can be done by non-parametric clustering algorithms in which neurons in the same cluster can

be combined into one global neuron. Beta-Bernoulli Process – Maximum a Posteriori (BBP-MAP) (Yurochkin, M., Agarwal, M., Ghosh, S., Greenewald, K., Hoang, N., & Khazaeni, Y. 2019) is used to calculate 2D permutation matrix which helps us to identify which neurons should be combined. The Hungarian method is applied to the matrix to select neurons to be merged and neurons to be added. Modified layers are then sent back to clients incrementally. The idea is shown in Figure 3.

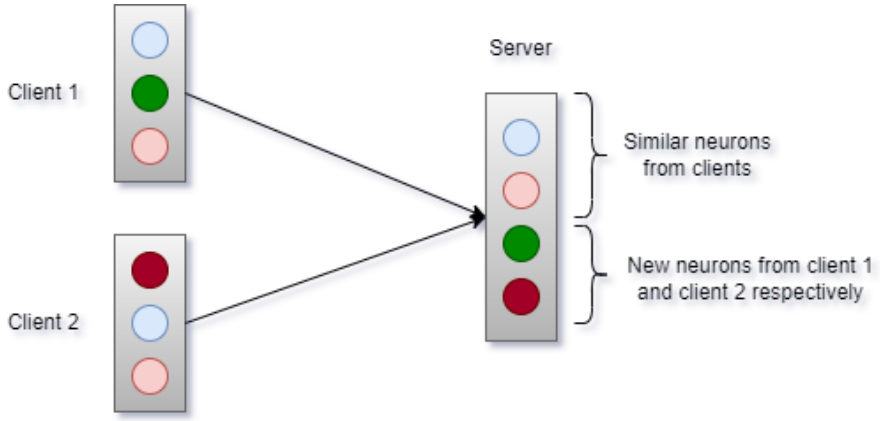

Fig. 3: FedMA algorithm

3. Security and Attacks

The number of clients can be huge in the federated learning model. When there are numerous clients in one model, it can open doors for attacks on clients, servers, models, etc. Thus security, privacy, and integrity in one of the most critical topics in federated learning, and these things must be considered while designing the FL model (Gosselin, R., Vieu, L., Loukil, F., & Benoit, A. 2022). This section will discuss vulnerabilities, attacks, and possible defenses according to papers and publications.

3.1. Vulnerabilities

We can define vulnerabilities as a weakness in the system against malicious attackers to gain unauthorized access. Detecting them is usually the very first step when it comes to security. After vulnerabilities are detected in defense, anti-attack actions can be taken accordingly, thus tightening the gaps in the defense mechanism of the FL model (Liu, P., Xu, X., & Wang, W. 2022). We can classify the vulnerabilities into three main groups. These groups are very similar to vulnerabilities in distributed systems.

1. Attacking through communication channels: During the FL process, weights are exchanged between clients and servers via communication channels. The man-in-the-middle can access and manipulate these weights during the exchange process.

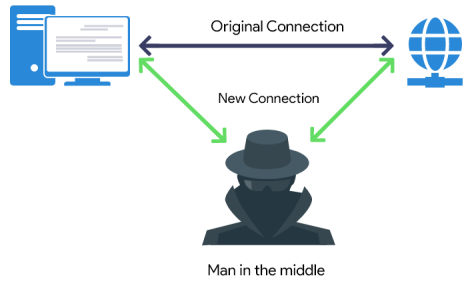

Fig. 4: Man-in-the-middle attacks

Thus, securing communication lines is crucial to secure and safe messages during communication rounds.

2. Byzantine nodes: In FL, the number of clients is huge; some are likely Byzantine nodes. These nodes are "bad" nodes and send the wrong models and gradients to the server. As a result, the performance of the FL model decreases drastically. These attacks are known as data poisoning. (Bhagoji, A. N., Chakraborty, S., Mittal, P., & Calo, S., 2019)

3. Failure of the single point: All clients send updated weights or gradients to the server. So, the server is the cornerstone of the model and must be robust and secure to reduce the risk of failure of the FL model. Thus, the security of the server is crucial both physically and electronically. Security software, constant updates, and all other security measures must be done to guarantee the server's security (Lyu, L., Yu, H., & Yang, Q. 2020).

3.2. Possible Attacks

Cyberattacks are widespread to attack and decrease the performance of the FL model. In this section, the most common ones will be discussed.

1. Poisoning attacks: Poisoning attacks are the most common attacks on FL models. The main idea behind this attack is to send malicious and rigged data to the server to reduce the performance of the global model (Zhang, J., Zhu, H., Wang, F., Zhao, J., Xu, Q., & Li, H., 2022). The demonstration is shown in Figure 5. There are two main types of poisoning attacks: Data poisoning and Model poisoning.

- In the data poisoning approach, the attacker includes malicious data to send to the server. These data create bias in the global model. Sometimes, to make it undetectable, attackers modify model parameters but do it repeatedly, making it harder for the FL model to detect if the data is poisonous. (Tolpegin, V., Truex, S., Gursoy, M. E., & Liu, L., 2020)
- Model poisoning is more direct. It sends a modified and infected model to the server without inserting data.

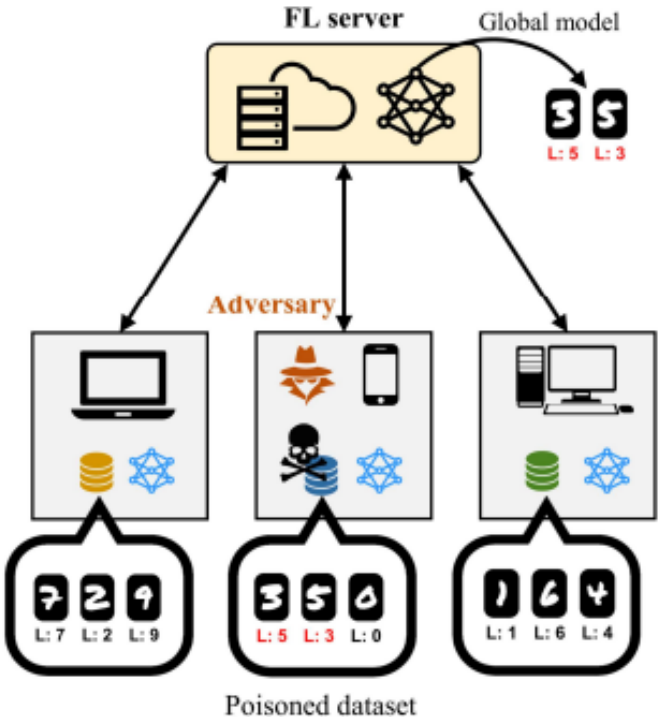

Fig. 5: Model poisoning attack

2. Backdoor attacks: These attacks are similar to poisoning attacks. However, the only difference is that Backdoor attacks do not aim to reduce global model accuracy. These attacks focus only on one label. Attackers create instances very similar to one label with minor changes and label it entirely differently (Wu, Q., He, K., & Chen, X.,2020). By doing so, overall model accuracy remains mostly the same but significantly impacts classification. An example is shown in figure 6.

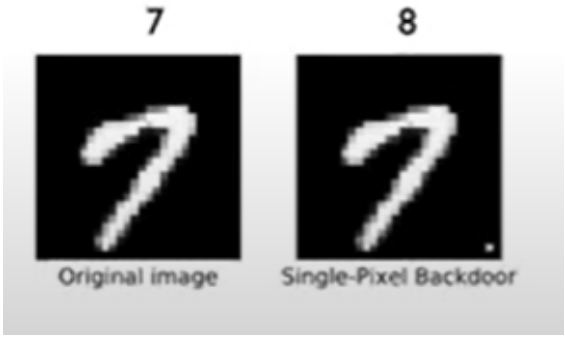

Fig. 6: Example of backdoor attack

3. Generative Adversarial Network (GAN) Based Attacks: It is a type of poisoning attack. In this attack, attackers use Generative Adversarial Network (GAN) to create realistic data and manipulate the labels in a way they want. (Zhang, J., Chen, B., Cheng, X., Binh, H. T. T., & Yu, S. 2020). These GANs are optimized in the client; weights are modified to manipulate the weights on the global model. These attacks are tough to detect since the generated data are very realistic.

4. Communication attacks: As mentioned in the previous section, the weights or gradients can be obtained by attackers and modified during the communication round. There are two main types of communication attacks:

- Man-in-the-middle attacks: In this attack, attackers can reach data during communication and replace them with malicious data. Usually, attackers create a fake network between clients and servers. Thus, every update goes through the attackers' hands (Wang, D., Li, C., Wen, S., Nepal, S., & Xiang, Y.(2020).
- Communication bottleneck attacks: Here, attackers increase the number of clients, which causes the dropout of some users. Additionally, removing clients based on their connection status leads to biases in the global shared model over time and impacts the aggregate of individual updates. Furthermore, methods that reduce communication overhead, such as compression, can be used destructively to introduce noise into individual updates and lower their quality. (Chen, Y., Sun, X., & Jin, Y. 2019)

5. Free-rider attacks: In these attacks, free-riders try to acquire an updated global model without participating in the FL process (Lin, J., Du, M., & Liu, J. 2019). It is done by mimicking minor local updates. By this, an attacker can still obtain the global model. There can be several reasons which motivate attackers, such as:

- The attacker wants to save computational resource
- Do not want to share with the global model his local data

3.3. Defenses

In this section, defense measures will be discussed. Since there are different attack types, different defense mechanisms are also required against them (Rodríguez-Barroso, N., Jiménez-López, D., Luzón, M. V., Herrera, F., & Martínez-Cámara, E., 2023)

• Anomaly detection: Anomaly or outlier detection is a process of identifying events or records which does not fit the overall pattern of activity using analytical and statistical methods. Several methods and machine learning algorithms, such as DBSCAN, Isolation Tree, One-Class SVM, etc., have proven to be effective in anomaly detection. Anomaly detection can be effectively used to analyze new upcoming data and decide whether new updating should be considered by classifying it as an anomaly.

Chen proposed a couple of approaches to using anomaly detection in FL. One approach was using a validation dataset (Chen, Y., Su, L., & Xu, J.,2017). The FL checked whether new updates received by specific clients decreased the metric in the

validation set. If it does, then it is marked as an anomaly. The other method is called Sniper, and it is based on graphs and cliques in a graph. Cao, D., Chang, S., Lin, Z., Liu, G., & Sun, D. ,2019)

Although anomaly detection algorithms succeed in poisoning attacks, they suffer from identifying and preventing backdoor attacks.

• Federated filtering: Federated filtering is another defense measure that is applied in order to protect FL. This is done by transferring the global model to some small model with steps that should be done. As a result, it increases the quality of security and client data privacy. Transferring to some small models also improves the time performance of the model (Zhu, Z., Hong, J., & Zhou, J.,2021)

• Defense against Backdoor attacks: The primary preventive measure against backdoor attacks is called Pruning. This is done by reducing the complexity of the model in trade for little accuracy. Because backdoor attacks are complex to carry out, it will make more sense to use small models. Small models will also be helpful in communication costs (Liu, K., Dolan-Gavitt, B., & Garg, S., 2018).

• Defense against GAN attacks: The defense against this attack must be better developed and documented. However, methods such as advanced Byzantine actor detection and ML using model distillation were proposed. (Benmalek, M., Benrekia, M. A., & Challal, Y. ,2022)(Hayes, J., & Ohrimenko, O. ,2018),(Li, D., & Wang, J. ,2019)

4. Non-IID Data

Non-IID data stands for Non-Independent and identically distributed data. It is the opposite of IID data, where each data sample is independent and identically distributed data. The main reason behind it there are numerous clients connected to the server, and data distribution in one client can be completely different from the distribution of another client (Chen, Y., Sun, X., & Jin, Y.,2019), (Criado, M. F., Casado, F. E., Iglesias, R., Regueiro, C. V., & Barro, S. (2022). For example, in supervised learning, each client has a data sample (x, y) where x is an attribute and y is a label with some distribution Pk (x, y) (Kairouz, P., McMahan, H. B., Avent, B., Bellet, A. et al. 2023). In Non-IID data, the distribution Pk is different for each client k. Zhu (Zhu, H., Xu, J., Liu, S., & Jin, Y. 2021) categorized Non-IID data into several categories.

1. They differ in attributes. In this category, attributes or features among clients are different. These features can, between clients, be the same, different, and partially the same. Examples are as follows:

• The same - In this type, labels are the same and attributed differ by distribution. We can show EMNIST dataset as an example in which a different person writes each number. As a result, we have different features like width, slant, and skewness, even for the same labels.

• Different - In this type, different clients contain different information about one label, like the first client containing features x1, x2, and the second containing features x3, x4 for the same label. For example, gastroenterologists and hepatologists have

entirely different feature sets, but labels can be identical.

• Partially-same - In this category, different clients can have feature set which, in some points, overlaps. An excellent example of it is CCTV cameras. In one room, two different cameras can have views from different angles. In this case, the same label partially has the same features. (Li, Q., Diao, Y., Chen, Q., He, B., 2021)

2. Differing on labels: In this category, label distributions differ from client to client. There are two closely related types of distributions:

• Labels have different distributions: The reason behind them is very loose. The different clients can have different training datasets. For example, client 1 has images mostly labeled as "Football," and client 2 has an image mostly labeled as "Tennis."

• Preference: In this type exact same data sample can be labeled differently by two clients. For example, client 1 provides positive feedback for the same movie, whereas client 2 provides negative feedback, as demonstrated in Figure 7. (Garcia-Molina, H., Joglekar, M., Marcus, A., Parameswaran, A., & Verroios, V. ,2016).

Fig. 7: Differing in preference

3. Differing in time: This type of data differs depending on some period. For example, some information is provided by client 1 for the first two weeks of some months and client 2 for the second two weeks of the month. (Zhu, H., Xu, J., Liu, S., & Jin, Y. ,2021)

Dealing with non-IID data is considerably different from IID data. Thus, some measurements must be considered when dealing with non-IID data. For example, calculating the gradients for linear models may be a little problematic since parameters are proportional to input data (Aono, Y., Hayashi, T., Wang, L., & Moriai, S. (2017). That is why additional methods, such as homomorphic encryption (HE), is used in linear methods. In contrast, neural network mostly depends on the algorithm itself. Results show that FedAVG algorithms perform relatively better with shallow neural network models. (Aono, Y., Hayashi, T., Wang, L., & Moriai, S. (2017)

References

Al Hayajneh, A., Bhuiyan, M. Z. A., & McAndrew, I. (2020). Improving Internet of Things (IoT) security with software-defined networking (SDN). *Computers, 9*(1), 8.

Aono, Y., Hayashi, T., Wang, L., & Moriai, S. (2017). Privacy-preserving deep

learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security, 13*(5), 1333-1345.

Benmalek, M., Benrekia, M. A., & Challal, Y. (2022). Security of Federated Learning: Attacks, Defensive Mechanisms, and Challenges. *Revue des Sciences et Technologies de l'Information-Série RIA: Revue d'Intelligence Artificielle, 36*(1), 49-59.

Bhagoji, A. N., Chakraborty, S., Mittal, P., & Calo, S. (2019, May). Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning* (pp. 634-643). PMLR.

Blanco-Justicia, A., Domingo-Ferrer, J., Martínez, S., Sánchez, D., Flanagan, A., & Tan, K. E. (2021). Achieving security and privacy in federated learning systems: Survey, research challenges and future directions. *Engineering Applications of Artificial Intelligence, 106,* 104468.

Cao, D., Chang, S., Lin, Z., Liu, G., & Sun, D. (2019, December). Understanding distributed poisoning attack in federated learning. In *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)* (pp. 233-239). IEEE.

Chen, Y., Su, L., & Xu, J. (2017). Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems, 1(*2), 1-25.

Chen, Y., Sun, X., & Jin, Y. (2019). Communication-efficient federated deep learning with layer-wise asynchronous model update and temporally weighted aggregation. *IEEE transactions on neural networks and learning systems, 31*(10), 4229-4238.

Criado, M. F., Casado, F. E., Iglesias, R., Regueiro, C. V., & Barro, S. (2022). Non-IID data and Continual Learning processes in Federated Learning: A long road ahead. *Information Fusion, 88,* 263-280.

Ek, S., Portet, F., Lalanda, P., & Vega, G. (2020, September). Evaluation of federated learning aggregation algorithms: application to human activity recognition. In *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers* (pp. 638-643).

Gao, L., Fu, H., Li, L., Chen, Y., Xu, M., & Xu, C. Z. (2022). FedDC: Federated Learning with Non-IID Data via Local Drift Decoupling and Correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10112-10121).

Garcia-Molina, H., Joglekar, M., Marcus, A., Parameswaran, A., & Verroios, V. (2016). Challenges in data crowdsourcing. *IEEE Transactions on Knowledge and Data Engineering, 28*(4), 901-911.

Gosselin, R., Vieu, L., Loukil, F., & Benoit, A. (2022). Privacy and Security in Federated Learning: A Survey. *Applied Sciences, 12*(19), 9901.

Hayes, J., & Ohrimenko, O. (2018). Contamination attacks and mitigation in multi-party machine learning. *Advances in neural information processing systems, 31.*

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., ... & Zhao, S. (2021). Advances and open problems in federated learning. *Foundations and*

Trends® in Machine Learning, 14(1–2), 1-210.

Li, D., & Wang, J. (2019). Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581.*

Li, Q., Diao, Y., Chen, Q., & He, B. (2022, May). Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)* (pp. 965-978). IEEE.

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

Lin, J., Du, M., & Liu, J. (2019). Free-riders in federated learning: Attacks and defenses. *arXiv preprint arXiv:1911.12560.*

Liu, K., Dolan-Gavitt, B., & Garg, S. (2018, September). Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses* (pp. 273-294). Springer, Cham.

Liu, P., Xu, X., & Wang, W. (2022). Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives. *Cybersecurity, 5*(1), 1-19.

Lyu, L., Yu, H., & Yang, Q. (2020). Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133.*

Lyu, L., Yu, H., Ma, X., Chen, C., Sun, L., Zhao, J., ... & Philip, S. Y. (2022). Privacy and robustness in federated learning: Attacks and defenses. *IEEE transactions on neural networks and learning systems.*

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering, 22*(10), 1345-1359.

Rodríguez-Barroso, N., Jiménez-López, D., Luzón, M. V., Herrera, F., & Martínez-Cámara, E. (2023). Survey on federated learning threats: concepts, taxonomy on attacks and defences, experimental study and challenges. *Information Fusion, 90,* 148-173.

Sannara, E. K., Portet, F., Lalanda, P., & German, V. E. G. A. (2021, March). A federated learning aggregation algorithm for pervasive computing: Evaluation and comparison. In *2021 IEEE International Conference on Pervasive Computing and Communications (PerCom)* (pp. 1-10). IEEE.

Su, H., Maji, S., Kalogerakis, E., & Learned-Miller, E. (2015). Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision* (pp. 945-953).

Sun, T., Li, D., & Wang, B. (2022). Decentralized federated averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

Tolpegin, V., Truex, S., Gursoy, M. E., & Liu, L. (2020, September). Data poisoning attacks against federated learning systems. In *European Symposium on Research in Computer Security* (pp. 480-501). Springer, Cham.

Wang, D., Li, C., Wen, S., Nepal, S., & Xiang, Y. (2020). Man-in-the-middle attacks against machine learning classifiers via malicious generative models. *IEEE Transactions on Dependable and Secure Computing, 18*(5), 2074-2087.

Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., & Khazaeni, Y. (2020).

Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440.*

Wu, Q., He, K., & Chen, X. (2020). Personalized federated learning for intelligent IoT applications: A cloud-edge based framework. *IEEE Open Journal of the Computer Society, 1,* 35-44.

Yurochkin, M., Agarwal, M., Ghosh, S., Greenewald, K., Hoang, N., & Khazaeni, Y. (2019, May). Bayesian non-parametric federated learning of neural networks. In *International Conference on Machine Learning* (pp. 7252-7261). PMLR.

Zhang, J., Chen, B., Cheng, X., Binh, H. T. T., & Yu, S. (2020). Poisongan: Generative poisoning attacks against federated learning in edge computing systems. *IEEE Internet of Things Journal, 8*(5), 3310-3322.

Zhang, J., Zhu, H., Wang, F., Zhao, J., Xu, Q., & Li, H. (2022). Security and Privacy Threats to Federated Learning: Issues, Methods, and Challenges. *Security and Communication Networks, 2022.*

Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., & Chandra, V. (2018). Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582.*

Zhu, H., Xu, J., Liu, S., & Jin, Y. (2021). Federated learning on non-IID data: A survey. *Neurocomputing, 465,* 371-390.

Zhu, Z., Hong, J., & Zhou, J. (2021, July). Data-free knowledge distillation for heterogeneous federated learning. In *International Conference on Machine Learning* (pp. 12878-12889). PMLR.

Submitted: 05.10.2022
Accepted: 23.11.2022