On the Security and Privacy of Federated Learning: A Survey with Attacks, Defenses, Frameworks, Applications, and Future Directions

Daniel M. Jimenez-Gutierrez, Yelizaveta Falkouskaya, José L. Hernandez-Ramos, Aris Anagnostopoulos, Ioannis Chatzigiannakis, Andrea Vitaletti

[a]Department of Computer, Control and Management Engineering, Sapienza University of Rome, Via Ariosto, 25, Rome, 00185, Rome, Italy

Abstract

Federated Learning (FL) is an emerging distributed machine learning paradigm enabling multiple clients to train a global model collaboratively without sharing their raw data. While FL enhances data privacy by design, it remains vulnerable to various security and privacy threats. This survey provides a comprehensive overview of more than 200 papers regarding the state-of-the-art attacks and defense mechanisms developed to address these challenges, categorizing them into security-enhancing and privacy-preserving techniques. Security-enhancing methods aim to improve FL robustness against malicious behaviors such as byzantine attacks, poisoning, and Sybil attacks. At the same time, privacy-preserving techniques focus on protecting sensitive data through cryptographic approaches, differential privacy, and secure aggregation. We critically analyze the strengths and limitations of existing methods, highlight the trade-offs between privacy, security, and model performance, and discuss the implications of non-IID data distributions on the effectiveness of these defenses. Furthermore, we identify open research challenges and future directions, including the need for scalable, adaptive, and energy-efficient solutions operating in dynamic and heterogeneous FL environments. Our survey aims to guide researchers and practitioners in developing robust and privacy-preserving FL systems, fostering advancements safeguarding collaborative learning frameworks' integrity and confidentiality.

Keywords:
Federated Learning, Privacy-Preserving, Security Mechanisms, Adversarial Attacks, Robustness, Defense Mechanisms.

1. Introduction

Machine Learning (ML) has revolutionized numerous fields [1] by enabling computers to learn from data and make informed decisions without being explicitly programmed for every scenario. This capability has become increasingly crucial in today's data-driven world, where the volume, velocity, and variety of information far exceed human capacity for manual analysis. ML applications span a wide range of industries, including healthcare [2], finance [3], manufacturing [4], and entertainment [5]. It offers solutions to previously intractable problems and opens new frontiers for innovation. As organizations and researchers seek to leverage the power of ML, they often face challenges related to data accessibility and *privacy concerns*.

Federated Learning (FL) [6] has emerged as a powerful paradigm enabling multiple clients (local nodes, parties, participants) to train ML models collaboratively without sharing raw data. While FL enhances data privacy, it also introduces unique *security* and *privacy* challenges that do not exist in traditional centralized learning settings, including vulnerabilities exacerbated by non-IID (non-Independent and Identically Distributed) data, where client datasets exhibit statistical heterogeneity in label, feature, or quantity distributions. Non-IID data amplifies security risks such as poisoning attacks, as adversaries can exploit skewed local updates to manipulate the global model, and privacy risks like membership inference, where attackers infer participation of specific data points by exploiting distributional disparities [7].

The distributed nature of FL makes it vulnerable to various types of attacks, including model poisoning, backdoor attacks, adversarial manipulations, data and gradient leakage, and model update inference, with non-IID conditions further undermining conventional defenses like differential privacy (DP) and robust aggregation. Addressing these challenges is crucial to ensure the robustness, reliability, and trustworthiness of FL systems, especially as they become increasingly adopted in sensitive domains such as healthcare [8, 9], finance [10], and telecommunications [11], among others.

1.1. Motivation

Our survey seeks to present a comprehensive and interconnected overview of security and privacy in FL. We provide a cohesive perspective on FL's security and privacy landscape by thoroughly examining various factors such as attacks, privacy issues, and defense strategies. This integrated approach enables a deeper comprehension of how FL security and privacy components are interrelated and influence each other. Through synthesizing insights from the field, our work aims to offer a complete understanding of the current state of FL security and privacy, helping foster a more detailed and nuanced awareness of the challenges and possibilities in this area.

Table 1 shows a detailed examination of existing surveys (found following our literature review process explained in Sec-

Table 1: Summary of previous Surveys related to privacy and security in FL (✔: Included, ◆: Partially included, ✘: Not included)

Survey	Publication Year	Security Taxonomy	Privacy Taxonomy	Security Attacks/Defenses	Privacy Attacks/Defenses	Top-tier venues	Frameworks	Fields of Application	Future Directions
[12]	2024	◆	◆	✔	✔	✘	✔	◆	✔
[13]	2024	✘	✘	✔	✔	✘	✘	✘	✔
[14]	2023	◆	◆	◆	◆	✘	✘	✘	✔
[15]	2023	✘	✘	✔	✘	✘	✘	✔	✘
[16]	2023	✘	✘	✔	✘	✘	◆	◆	✔
[17]	2023	◆	✔	✔	✔	✘	✘	✘	✔
[18]	2022	◆	✘	✔	✘	✘	✘	✘	◆
[19]	2022	◆	◆	✔	◆	✘	◆	◆	✔
[20]	2022	◆	✔	◆	✔	✘	◆	✘	◆
[21]	2021	◆	◆	✔	✔	✘	✘	✘	✔
[22]	2021	✘	✔	✘	✔	✘	✘	✘	✔
[23]	2021	✔	✔	✔	✔	✘	✔	◆	✔
[24]	2021	✘	✘	◆	◆	✘	◆	◆	✔
[25]	2021	✘	✘	✘	✔	✘	✘	◆	◆
[26]	2020	✘	✘	◆	✔	✘	✔	✘	✘
Ours	2025	✔	✔	✔	✔	✔	✔	✔	✔

tion 1), revealing significant gaps in integrating these topics. Despite the growing volume of literature, we observe a fragmented landscape: most prior surveys treat either privacy or security in isolation, often listing threats or defenses without organizing them under a shared conceptual framework. Others omit practical concerns like system frameworks, application domains, or scalability trade-offs. For example, Hu et al. [12] and Hallaji et al. [13] primarily address security and privacy attacks/defenses but lack coverage of frameworks and application fields. Similarly, Nair et al. [14], and Neto et al. [15] offer insights into specific areas like security defenses or privacy concerns without integrating these aspects into a broader taxonomy or discussing future directions. Surveys by Liu et al. [20] and Gong et al. [18] heavily focus on security attacks but do not comprehensively address privacy or the application of frameworks.

We find that:

- Only **1 out of 16 surveys** attempt to cover both privacy *and* security perspectives.
- Fewer than half provide any structured taxonomy of attacks or defenses.
- Practical dimensions — such as frameworks and real-world FL applications — are omitted in 12 out of 16 surveys.
- None of the existing surveys unify attacks, defenses, and system-level concerns into a single integrated view.

This work aims to build upon and extend the valuable research done in previous studies by offering a comprehensive and systematized approach to threats, defenses, and frameworks in FL. We present an extensive catalog that consolidates and expands upon the diverse sets of threats and defenses discussed in the existing literature, providing a multi-faceted categorization of attacks and their corresponding solutions. Additionally, we examine relevant frameworks, including privacy and security considerations for FL systems to offer a holistic view of the FL landscape.

1.2. Contribution

Our survey addresses this gap by thoroughly reviewing FL's security and privacy landscape. Table 1 compares our work with previous surveys, highlighting our study's unique coverage and depth. Our survey distinguishes itself by providing a holistic approach integrating a broad spectrum of critical areas. We cover security and privacy taxonomies, security and privacy attacks/defenses, and include discussions on top-tier venues, frameworks, and fields of application. By offering this comprehensive coverage and systematically describing attacks from different perspectives, our survey provides a deeper understanding of the various facets of security and privacy in FL. Notably, our work is among the few that addresses all these aspects in a unified framework, thereby offering a complete and cohesive overview for researchers and practitioners.

Specifically, our contributions are as follows:

1. *Comprehensive Taxonomies:* We provide detailed taxonomies of security and privacy threats and the corresponding defense mechanisms in FL. These taxonomies serve as a structured framework for understanding the diverse challenges and solutions in the field.
2. *Inclusion of Frameworks and Applications:* Our survey is among the few to cover FL frameworks and real-world fields of application for FL. This inclusion offers practical insights into how security and privacy measures are implemented and tested in real-world scenarios.
3. *Future Directions and Open Challenges:* We identify vital open challenges and outline promising future directions, offering valuable guidance for researchers looking to address the existing gaps in the literature.

Overall, our survey is distinguished by its broad scope and integrated approach, making it a valuable resource for researchers and practitioners seeking a comprehensive understanding of security and privacy in FL.

1.3. Relevant Papers Retrieval

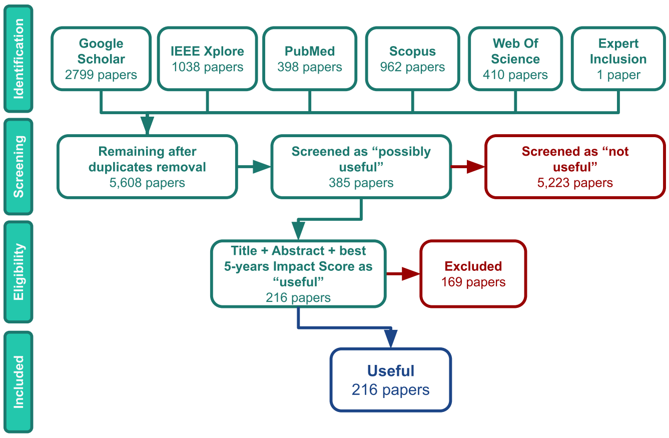
Identification
Screening
Eligibility
Included
Google Scholar 2799 papers
IEEE Xplore 1038 papers
PubMed 398 papers
Scopus 962 papers
Web Of Science 410 papers
Expert Inclusion 1 paper
Remaining after duplicates removal 5,608 papers
Screened as "possibly useful" 385 papers
Screened as "not useful" 5,223 papers
Title + Abstract + best 5-years Impact Score as "useful" 216 papers
Excluded 169 papers
Useful 216 papers

Figure 1: PRISMA flow for gathering relevant references

This work conducted a literature review following the PRISMA methodology [27] to retrieve and comprehensively analyze FL security and privacy literature. Figure 1 depicts each stage of the methodology to retrieve the most relevant papers. The literature review addressed three key research questions: identifying recent attacks and threats, exploring countermeasures, and evaluating FL frameworks for real-world applications. Using six reputable databases (Google Scholar, IEEE Xplore, PubMed, Scopus, Web of Science), 59 search queries were employed across three themes: attacks, defenses, and frameworks (see Table 2). After removing duplicates, 2,002 papers were screened based on keywords, titles, abstracts, and impact scores, ultimately narrowing the selection to 217 high-quality papers with an impact score above 5.

Table 2: Example search queries by topic

Topic	Example search queries
Attacks	"federated learning attacks" "federated learning data poisoning" "federated learning backdoor attacks"
Defenses	"federated learning differential privacy" "Federated learning secure multiparty computation" "federated learning homomorphic encryption"
Frameworks	"federated learning frameworks" "Federated learning flower" "real-world applications of federated learning"

1.4. Road Map

Figure 2 outlines the structure of this survey. In the Section 2, we provide the FL background. Section 3 defines the taxonomy of attacks and defenses for security in FL. Next, in Section 4, we provide the same for privacy in FL. Section 5 lists the standardized frameworks for FL. Section 6 showcases the most relevant applications of FL. Then, in Section 7, we provide exciting future directions. Finally, we conclude in Section 8.

For the reader's convenience, the acronyms used in this work are listed in Table 3.

Acronym	Description
ADIs	Adversarial dominating inputs
AFR	Anonymous Free-Rider
ALIE	A Little Is Enough attack
AutoGM	Auto-Weighted GeoMed
BFT	Byzantine Fault Tolerance
C-GANs	Cross-Client GANs
CPA	Cocktail Party Attack
DDP	Dynamic Differential privacy
DFL	Decentralized federated learning
DLG	Deep leakage from gradients
DP	Differential Privacy
E2EGI	End-to-End Gradient Inversion Attack
FC	Fully connected
FL	Federated learning
FOLTR	Federated online learning to rank
FR	Free-Rider
GAN	Generative Adversarial Network
GDPR	Data Protection Regulation
GeoMed	Geometric Median
GS	Gradient stalking
HIPAA	Health Insurance Portability and Accountability Act
HE	Homomorphic Encryption
IPM	Inner Product Manipulation
IoT	Internet of Things
LDP	Local differential privacy
MAB	Adversarial Multi-Armed Bandit
MarMed	Marginal Median
MCS	Mobile crowdsensing
MeaMed	Mean Around Median
MitM	Man-in-the-Middle
ML	Machine learning
MPC	Secure Multiparty Computation
OT	Oblivious Transfer
PASS	Parameter Audit-based Secure and Fair FL Scheme
PID	Privacy-aware and incremental defense
PMIAs	Poisoning membership inference attacks
RoFL	Robustness of secure FL
SCA	Sybil-Based Collusion Attacks
SFL	Split Federated Learning
SFR	Selfish Free-Rider
SR	Systematic review
SS	Secret sharing
TFF	Tensorflow Federated
VQA	Visual question-answering
ZKP-FL	Zero-knowledge proof-based FL
ZKPs	Zero-knowledge proofs

Table 3: Acronyms employed in this paper

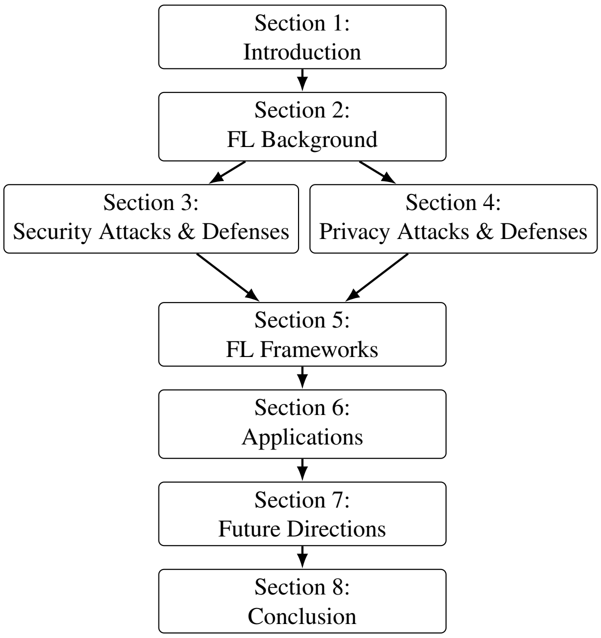

Figure 2: Overview of the paper structure. Each section builds on previous content, progressing from foundational concepts to attack and defense taxonomies, frameworks, applications, and future directions.

2. FL Background

FL [6] is an ML technique for cooperatively training models on several clients in a decentralized way, preserving data privacy and ownership for the client/server owner [28]. FL is hugely advantageous for highly decentralized data, especially with the growing prevalence of IoT devices for continuously capturing data and monitoring users' patterns.

Fig. 3 depicts a high-level view of the framework and how the clients interact with the central server. IoT devices, institutions (i.e., hospitals, companies, etc.), documents, or vehicles will collect user data and train a local deep-learning model that mirrors a previously received global model [29]. Following the completion of the local training phase, the models collaborate to train a global model utilizing their updates rather than the raw data provided by the users. These model updates indicate changes in the models' weights during training and do not reflect private or personal information about the users.

All clients will send updates to a central server, compiling and using them to aggregate the global model weights [30]. Once the global model training procedure finishes, each client will receive a new copy of the updated global model. As a result, the models will be trained and updated regularly without sharing personal information. Thus, the framework will enable a decentralized architecture in which models get distributed among clients without requiring a centralized server to operate the model and serve users. It will also protect users' privacy by processing and analyzing their data on clients without disclosing it.

The collaborative model training process in FL involves *aggregating model updates* from multiple decentralized clients while preserving data privacy. Aggregation algorithms are pivotal in this context, serving as the cornerstone for combining these distributed updates into a global model. These algorithms are essential to ensure that the federated model achieves the desired convergence and accuracy while safeguarding the privacy and security of the individual clients' data.

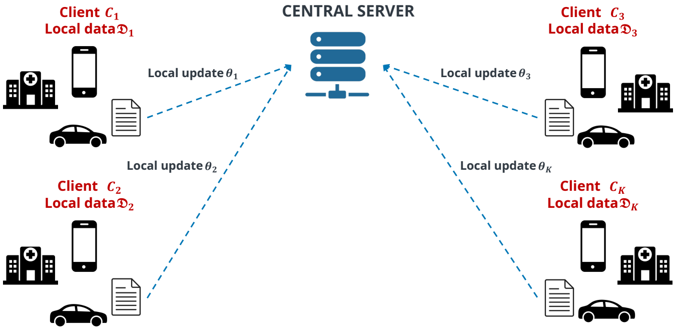

Figure 3: FL framework overview

FedAvg [6] is the most employed *aggregation algorithm* that operates within a client-server architecture, where the server orchestrates the training process, and the clients conduct local training on their data. Each client independently trains the model using its local data and transmits model updates to the server. The server aggregates these updates to construct a global model. FedAvg's advantages include scalability to accommodate a large user base through decentralized training and improved efficiency through the ease of computation in a centralized server. However, in FL settings, one should consider challenges such as client heterogeneity, communication overhead during update aggregation, and potential network connectivity limitations.

From a mathematical point of view [6], FL is defined by a set of K clients, denoted as $C_1, C_2, ..., C_K$. Each client C_i has its dataset $\mathcal{D}_i$ containing features (x) and labels (y) for certain examples (individuals, samples). FL aims to train a global model θ in a decentralized manner, where the model parameters are updated by aggregating the local updates from each client while keeping the data on the clients. The loss minimized during the FL process is $L(\theta) = \sum_{i=1}^{K}(1/K) * L(\theta_i)$ where $L(\theta)$ is the global loss function to be minimized and $L(\theta_i)$ is the local loss function for client C_i. This function quantifies the discrepancy between the predictions of the global model θ and the ground truth labels for the samples in client C_i's dataset $\mathcal{D}_i$.

2.1. Types of FL

FL is currently in an active development phase and employs diverse techniques and methodologies to bring its core technology into practical implementation. When dealing with a nascent technology like FL, initially categorizing these techniques and approaches is a pivotal starting point, enabling a more profound comprehension and exploration beyond the broader conceptual framework. Depending on data partition and scalability, FL gets divided into different categories [19, 23, 31].

2.1.1. Data Partition

FL systems are commonly categorized based on the data distribution across the sample and feature spaces. This categorization typically divides FL systems into three main types: horizontal FL, vertical FL, and hybrid FL. Each category represents distinct approaches to handling data distribution in FL scenarios.

Horizontal FL: In horizontal FL, clients share a common feature space but have limited overlap in the sample space, making it suitable for cross-device settings where users collaborate on a shared task. Local models are trained independently with consistent architectures, and the global model is updated by averaging local weight updates.

Mathematically, horizontal FL is represented by contemplating the same features across clients but with different examples. For example, suppose clients C_1 and C_2 have data on different users for a recommendation system. In that case, $\mathcal{D}_1 = \{(x_1, y_1), (x_2, y_2), \dots, (x_{n_1}, y_{n_1})\}$ with feature space X and n_1 the number of examples of C_1, and $\mathcal{D}_2 = \{(x'_1, y'_1), (x'_2, y'_2), \dots, (x_{n_2}, y_{n_2})\}$ with feature space X and n_2 the number of examples of C_2. Here, x_i and x'_i represent the same features for different examples [32, 22].

Vertical FL: In vertical FL, datasets from different nodes share the same or similar sample space but differ in the feature space. Entity alignment techniques identify overlapping samples by matching entity descriptions, enabling collaborative training of models like gradient-boosting decision trees. Privacy-preserving methods align entities across clients, facilitating joint gradient training. This approach is often seen in collaborations between different companies.

Mathematically, vertical FL is represented by considering the same set of examples across clients but with different features. For example, if clients C_1 and C_2 have data on patients where C_1 has medical records, and C_2 has genetic information, then $\mathcal{D}_1 = \{(x_1, y_1), (x_2, y_2), \dots, (x_{n_1}, y_{n_1})\}$ with feature space X_1 and n_1 the number of examples of C_1, and $\mathcal{D}_2 = \{(x'_1, y'_1), (x'_2, y'_2), \dots, (x_{n_2}, y_{n_2})\}$ with feature space X_2 and n_1 the number of examples of C_2. Here, x_i and x'_i represent different feature sets for the same examples [22].

Hybrid FL: In numerous other use cases, while conventional FL systems predominantly concentrate on a single type of data partition, the data distribution among the clients often exhibits a hybrid combination of horizontal and vertical divisions. One specific example of this type of FL is *Transfer FL* [33], which involves horizontal and vertical data partitioning, making it a hybrid approach. The latter allows models to learn from shared features (vertical) and data from different clients (horizontal) to improve performance and generalization.

Let clients C_1 and C_2 possess datasets D_1 and D_2 such that:

$$\begin{aligned} D_1 &= \{(x_i^{(1)}, y_i^{(1)})\}_{i=1}^{n_1}, \quad x_i^{(1)} \in \mathcal{X}_1, \\ D_2 &= \{(x_j^{(2)}, y_j^{(2)})\}_{j=1}^{n_2}, \quad x_j^{(2)} \in \mathcal{X}_2, \end{aligned}$$

where $\mathcal{X}1$ and $\mathcal{X}2$ are the feature spaces of C_1 and C_2, respectively. In Hybrid FL, there exist subsets Sshared $\subseteq S_1 \cap S_2$ (shared samples) and Xshared $\subseteq \mathcal{X}_1 \cap \mathcal{X}_2$ (shared features).

2.1.2. Scale of Federation

FL fashion can be classified into two types based on the extent of federation: cross-silo FL and cross-device FL. The distinctions between these types revolve around the number of clients and the volume of data stored within each client.

Cross-silo: The clients are typically organizations or data centers. A limited number of clients are generally involved, each with a substantial volume of data and computational resources. For instance, Amazon aims to offer user-item recommendations by leveraging shopping data from many data centers worldwide [34].

Cross-device: There is typically a more significant number of clients, each with a comparatively modest amount of data and computational capacity, often consisting of mobile devices. Google Keyboard exemplifies a cross-device FL, where the enhancement of query suggestions in Google Keyboard can benefit from the application of FL [35].

2.2. Split FL

Split FL (SFL) is a distributed machine learning approach that utilizes a split model architecture, dividing the model between clients and a central server. This design enhances privacy by avoiding raw data sharing and is suitable for resource-constrained environments due to its distributed computations, which reduce the burden on individual clients [36, 37]. SFL offers high scalability and efficiency in large-scale distributed setups, but comes with limitations, including slower performance compared to traditional FL due to its relay-based training process and increased communication overhead [38].

In SFL, for a client C_i, the training process proceeds as follows:

1. *Forward Pass:* The client computes activations up to the cut layer $a_i = f_{\theta_c}(x_i)$ and sends a_i to the server.
2. *Server Computation:* The server completes the forward pass $\hat{y}_i = f_{\theta_s}(a_i)$.
3. *Backward Pass:* The server computes the gradient $\nabla_{a_i}\mathcal{L}$ and sends it to the client, which then computes $\nabla_{\theta_c}\mathcal{L} = \nabla_{a_i}\mathcal{L} \cdot \nabla_{\theta_c} f_{\theta_c}(x_i)$.

The overall optimization objective is:

$$\min_{\theta_c, \theta_s} \frac{1}{K} \sum_{i=1}^{K} \mathbb{E}_{(x,y)\sim D_i} \left[\mathcal{L}\left(f_{\theta_s}(f_{\theta_c}(x)), y\right)\right] \tag{1}$$

2.3. Non-IID Data Impact on FL

In FL, non-IID [39] data refers to data that is not uniformly distributed across clients, meaning that different clients may have significantly different data distributions due to factors like user preferences, geographical location, or client usage patterns. Those disparities arise across three dimensions:

- *Label Distribution Skew:* Differences in $P(y|x)$ (the conditional probability distribution of labels y given features x) between clients. For instance, hospitals specializing in different diseases with imbalanced diagnostic labels.

- *Feature Distribution Skew:* Variation in $P(x)$ (the marginal probability distribution of features x) across clients. For example, smartphones in different regions capture distinct visual patterns (e.g., urban vs. rural environments).
- *Quantity Skew:* Disparities in dataset sizes n_k (where $n\_k = |\mathcal{D}_k|$ denotes the number of samples at client k) among clients. For example, IoT devices with varying storage capacities collect unequal data points.

Non-IID data poses serious privacy and security challenges as it can make models more vulnerable to inference attacks (e.g., membership and property inference) [40] since adversaries can exploit statistical discrepancies to extract sensitive information about client data. Additionally, non-IID data exacerbates the impact of poisoning attacks [41], where adversarial clients can more effectively manipulate global model updates by injecting biased gradients. On the defense side, traditional DP and robust aggregation methods, such as median or trimmed mean-based aggregation, often lose effectiveness in non-IID settings, as the variability in data distributions can lead to excessive noise or biased updates. Furthermore, anomaly detection methods [42] that rely on outlier detection may struggle to distinguish between natural variations due to non-IID data and actual adversarial behavior.

3. Security in FL

Based on the papers assessed, we propose a taxonomy of FL security and privacy attacks and defenses (see Fig. 4), providing a structured framework for understanding this evolving field. Following such a taxonomy, we describe the main attacks and defenses for secure FL in this section. For the attacks, we outline specific mechanisms, degrees of harm, and specific examples of manifestations in the real world. At the end of each subsection, we provide some lessons learned after analyzing the papers regarding attacks and defenses for secure FL.

3.1. Security Attacks/Threats

In FL, security attacks and threats involve adversarial strategies to compromise models' integrity, availability, confidentiality, and underlying data. These attacks can be categorized based on several criteria. In particular, to enhance clarity and reduce overlaps, we define five key dimensions for categorizing security attacks and threats in FL: target specificity, phase affected, intent, nature of the adversary, and execution style. For target specificity, *targeted attacks* aim to disrupt specific system elements, while *untargeted attacks* seek to cause general disruption or degrade overall performance. Phase affected clarifies whether the disruption happens mainly during model training (such as poisoning or Sybil attacks) or only becomes relevant at inference time (like evasion). Furthermore, attacks are categorized by intent; *malicious attacks* aim to cause harm, whereas *exploitative attacks* seek personal gain without direct harm. Additionally, the nature of the adversary plays a crucial role: *insider attacks* come from within the system, while *outsider attacks* originate from outside [14]. Finally, execution style clarifies whether the attacker must engage in multiple rounds or *continuous* participation to achieve success or can accomplish the attack in a single, *one-shot* instance.

This survey categorizes attacks based on their specific nature and tactics, offering a detailed taxonomy and examining their impacts on FL systems. To provide a structured overview, we present a comprehensive overview in Table 4 summarizing various attack types and categorizing them based on the mentioned dimensions. Although we define five primary dimensions–target specificity, phase affected, intent, nature of the adversary, and execution style–real-world attacks can exhibit traits spanning more than one category. For instance, a poisoning attack might initially appear *untargeted* but also target a specific class or region of the data. Likewise, an *insider* adversary could collaborate with *outsider* entities or extend the attack from training into inference phases. In Table 4, we classify each attack according to its most typical or principal form while recognizing that adversaries can mix methods or adopt hybrid strategies. The following sections will discuss critical security threats in FL, detailing their nature, objectives, and potential impacts and providing examples from the literature.

Fig. 5 shows a clear upward trend in the number of papers published on various security attacks over time. In 2019 and 2020, very few papers focused on Sybil and GAN-based attacks, respectively. From 2021 onwards, there's a noticeable diversification in the types of attacks studied, with a significant increase in overall research output. Poisoning attacks have become increasingly prominent, dominating the research landscape, especially in 2023 and 2024. Other attack types like backdoor, dropout, evasion, and free-riding have emerged in the later years, indicating an expansion in the scope of security research. 2024 shows the highest number of papers across multiple attack categories, suggesting a growing interest and concern in security attacks.

3.1.1. Byzantine attacks

A Byzantine attack refers to a broad category of malicious or faulty behaviors within distributed and FL systems. The term originates from the Byzantine Generals Problem [43], which highlights the challenge of achieving consensus in a distributed network when some clients act unpredictably due to malice or faults. In FL, these attacks disrupt the learning process, degrade model performance, or compromise system integrity. For instance, *model and data poisoning* attacks involve adversaries injecting harmful updates to skew the global model, as shown in empirical studies where such attacks significantly increase error rates [44]. Similarly, *Sybil attacks* manipulate aggregation by introducing multiple fake identities, amplifying the attacker's influence [45]. *Backdoor attacks*, on the other hand, secretly alter model behavior for specific inputs, such as embedding triggers that activate malicious outcomes. Real-world examples include tampering with IoT device models to misclassify security threats or injecting biased data in healthcare applications to compromise diagnostic accuracy. In practical terms, adversaries can perform Byzantine behaviors by intercepting local gradient updates and introducing arbitrary deviations before sending them to the server. Minimal Python scripts can scale or randomize these updates, allowing the attacker to bypass naive filters.

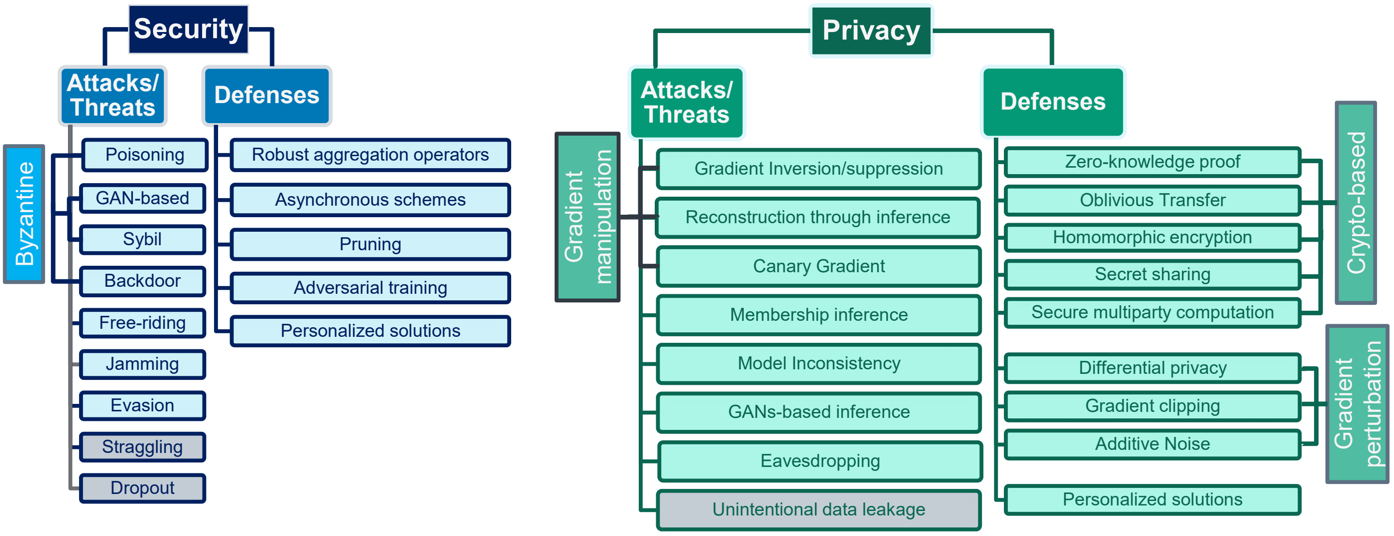

Figure 4: Security and privacy taxonomy for attacks and defenses in FL

Table 4: Categorization of Security Attacks in FL

Attack Type	Target Specificity	Phase Affected	Intent	Nature of Adversary	Execution Style
Data Poisoning	Targeted/Untargeted	Training	Malicious	Insider/Outsider	Continuous
Model Poisoning	Targeted/Untargeted	Training	Malicious	Insider	Continuous
GAN-based	Targeted/Untargeted	Training	Malicious	Outsider	Continuous
Sybil	Targeted/Untargeted	Training/Inference	Malicious	Insider/Outsider	Continuous
Backdoor	Targeted	Training	Malicious	Insider	Continuous
Free Riding	Untargeted	Training	Exploitative	Insider	Continuous
Jamming	Untargeted	Training/Inference	Disruptive	Outsider	One-Shot
Evasion	Targeted/Untargeted	Inference	Disruptive	Outsider	One-Shot
Straggling	Untargeted	Training/Inference	Disruptive	Insider	Continuous
Dropout	Untargeted	Training/Inference	Disruptive	Insider	Continuous

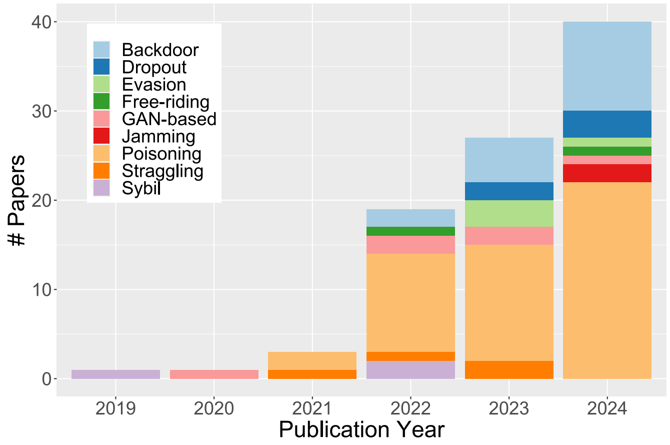

Figure 5: Papers related to security attacks over time

Some open-source prototypes demonstrate how two or three malicious clients can systematically skew the global model. Moreover, robust aggregation methods (e.g., Bulyan, Krum) typically detect large outliers but may fail against subtle manipulations. Integrating cryptographic checks (e.g., commitments) or analyzing multi-round consistency across updates can significantly reduce the success rate of Byzantine exploits. The following paragraphs explore these attack mechanisms and their consequences in detail.

Poisoning attacks. Poisoning attacks in FL involve injecting malicious data or manipulating model updates to compromise the integrity of the learning process. Such attacks can decrease overall model performance and allow the attacker to introduce biases, insert backdoors, or create specific targeted vulnerabilities.

Data poisoning refers to attacks where malicious clients alter their data or model's parameters sent to the global model to degrade its performance. *Untargeted data poisoning* involves general disruptions, such as adding random noise, random label flipping, and random input data poisoning, which can cause a significant drop in model accuracy and robustness. For instance, an attacker injecting noisy data into medical diagnosis models can lead to incorrect patient assessments. *Targeted data poisoning*, on the other hand, seeks to cause specific errors or misclassifications, such as targeted label flipping in autonomous driving systems, where stop signs are misclassified as speed limits, posing safety risks. Another way to categorize data poisoning attacks is based on whether the attacker can modify the labels of the poisoned data. *Clean-label poisoning* assumes that attackers cannot change data labels due to integrity constraints but instead subtly manipulate features, such as modifying image pixels to induce incorrect classifications. These attacks are especially dangerous in security-sensitive domains like biometric authentication, where small perturbations in face recognition models can allow unauthorized access while remaining undetectable by traditional defenses.

In contrast, *dirty-label poisoning* involves directly manipulating the data labels. In this scenario, the attacker introduces samples into the training dataset with incorrect labels, misleading the model during training. Dirty-label poisoning is generally easier to detect than clean-label poisoning because the data and its labels' inconsistencies are more apparent [46]. Table 5 provides an overview of data poisoning attacks, categorized by their targeted or untargeted nature and whether they involve clean or dirty labels. The table includes relevant papers for each category illustrating key research and findings.

Table 5: Classification of Data Poisoning Attacks

	Clean-label	**Dirty-label**
Targeted	Targeted data manipulation [47] [46] [48]	Targeted label flipping [49] [50] [48] [51]
Untargeted	Random data manipulation [48], Adversarial samples [52]	Random label-flipping [53] [48]

Model poisoning involves the deliberate manipulation of model parameters or updates sent to a central server, targeting the integrity of the model itself rather than the training data. This type of attack is particularly effective, often surpassing data poisoning in impact, especially against systems employing Byzantine-robust defense mechanisms. Fang et al. [44] demonstrated that non-directional attacks, which craft local model parameters to deviate significantly from expected values, can lead to aggregated updates that degrade global model performance. For example, their experiments showed that introducing perturbations maximized deviation from the typical update path, resulting in substantial global model errors. Baruch et al. [54] highlighted that even minimal poisoning–where only a small fraction of malicious updates is introduced–can bypass robust defenses by exploiting gradient variance. This approach requires limited knowledge of client data and subtly shifts the mean of aggregated gradients to evade detection. Real-world implications include attacks on recommendation systems, where subtle manipulations degrade ranking accuracy without triggering alarms. Wang et al. [55] extended these findings to federated online learning to rank (FOLTR) systems, showing that sophisticated poisoning strategies outperform data poisoning even under robust defenses. They also noted that deploying such defenses without active attacks can degrade system performance, underscoring the need for adaptive defenses that balance security and functionality.

Implementation-wise, data-poisoning attacks often involve straightforward label manipulation or pixel-level perturbations in the local dataset. Publicly available code, such as in [44], shows how a simple gradient-scaling procedure can overpower benign updates in an aggregation function like FedAvg. Model poisoning goes a step further, directly adjusting weight tensors to embed ”invisible triggers”. Defenders typically integrate robust aggregator pipelines (e.g., Krum or Trimmed Mean, see Section 3.2) and anomaly monitors that track suspicious gradient magnitudes or label discrepancies across rounds. Additionally, partial local data checks (for example, removing highly implausible labels) can disrupt stealthy poisoning attempts before they aggregate into a global parameter shift.

Generative Adversarial Network-based (GAN) attacks. GANs have been employed to execute both model and data poisoning attacks in FL. In such scenarios, an adversary masquerades as a benign client and trains a GAN to replicate prototypical samples from other clients' datasets. The global model parameters serve as the discriminator's parameters, enabling the GAN to produce realistic yet manipulated samples. These samples are then used to generate poisoning updates, which are scaled and submitted to the central server [56]. According to Zhang et al. [57], any internal client can initiate GAN-based poisoning attacks. For instance, their PoisonGAN model demonstrated that even under attack, the global model retained over 80% accuracy on both poisoning and primary tasks [48]. This highlights the dual threat of maintaining task performance while embedding malicious objectives. Real-world implications include adversaries exploiting GANs to bypass detection mechanisms, as seen in cases where vague or noisy poisoned data undermines anomaly detection systems [58]. These examples underline the significant harm GAN-based attacks pose, which compromise FL systems' integrity and privacy without easily detectable anomalies. From the implementation perspective, a GAN-based attack typically involves pairing the server's global model (as a discriminator) with a locally trained generator that refines malicious updates to appear "benign." Minimal modifications to PyTorch or TensorFlow scripts let attackers pass generator outputs as legitimate gradients. Potential defenses could include incremental offset detection that flags suspiciously consistent gradient distortions and clustering techniques for client updates with significant divergences.

Sybil attacks. The Sybil attack involves a malicious client creating multiple fake identities to gain disproportionate influence or control over the system. While not specifically a poisoning attack, it can facilitate or amplify poisoning attacks by increasing the number of fake clients that submit malicious or biased updates [15]. For example, model poisoning attacks using fake clients can significantly reduce the test accuracy of the global model, even against classical defenses [59]. Fung et al. [60] demonstrated this in their experiment where two Sybil nodes inserted a backdoor, causing 96.2% of digit 1s in the MNIST dataset to be misclassified as 7s in the final model. This highlights the severe impact even a few Sybils can have on model integrity. In real-world scenarios, such attacks are particularly concerning due to their ability to bypass detection mechanisms by preserving overall model utility [60]. Furthermore, another study [61] revealed how Sybil nodes could inject backdoor triggers into data, disrupting training processes in FL systems.

Employing Sybil clients can be as simple as registering multiple "fake" clients that communicate identical or slightly modified updates, all controlled by one adversary. Code examples [60] illustrates that only two Sybils can drastically corrupt a federated model. FoolsGold [60], or other similarity-based approaches track the cosine distance among client updates, penalizing suspicious clusters. Some frameworks incorporate blockchain-based identity management or limit how many new clients can join per round, raising barriers for mass Sybil infiltration. These countermeasures reduce the effectiveness and stealth of Sybil-based manipulations.

Backdoor attacks. Backdoor attacks are a form of targeted poisoning attack in which an adversary deliberately corrupts the global model, making it perform well on the main task while exhibiting malicious behavior when triggered by specific conditions, such as a particular label, image modification, or feature [23]. These attacks are particularly concerning in FL due to the decentralized nature of training, where malicious updates can propagate vulnerabilities across the entire system. For example, in real-world scenarios like next-word prediction models used in mobile applications, backdoor triggers could manipulate outputs for sensitive contexts, such as political events [62]. Liu et al. [62] demonstrated that backdoor attacks could accelerate FL convergence by crafting local updates that mimic global data distributions and injecting backdoors during later stages when benign updates have minimal impact. However, these attacks face challenges such as detection risks and limited persistence. Dai et al. [63] addressed these issues by proposing the Chameleon attack, which uses poisoned datasets and contrastive learning to enhance backdoor durability. This method ensures the backdoor remains effective even after attackers stop participating, as seen in applications like IoT devices with weak security measures [63]. Similarly, Zhang et al. [64] highlighted that fixed backdoor triggers often fail under global training dynamics. Their A3FL approach adapts triggers adversarially to maintain effectiveness in evolving models. These examples underscore the significant harm of backdoor attacks, which can compromise model integrity and user trust in critical applications like autonomous vehicles or healthcare systems. Implementing a backdoor often involves a "trigger pattern" integrated into a small fraction of the local training set (e.g., a tiny corner pixel pattern in image classification). Attack scripts typically swap labels for these trigger-laden inputs and train locally to ensure the global model learns to misclassify only when the pattern appears. FLAME [65] and other advanced defenses add mild noise or rely on "clean validation" to detect unexpected performance spikes on specific triggers. Another method is partial neuron pruning, removing neurons that show abnormally high activation for certain triggers. Adopting these defenses usually increases training overhead but significantly reduces successful backdoor injection rates.

3.1.2. Free-Riding

Free-riding occurs when a client benefits from the final aggregated model without contributing to its training due to reasons such as lack of data, privacy concerns, or insufficient computational resources. In the context of Free-Rider (FR) attacks, these can be categorized into Anonymous Free-Rider (AFR) and Selfish Free-Rider (SFR) attacks based on the adversary's control over private data and computing resources [66]. AFR attackers, lacking private datasets or computational resources, typically contribute stochastic Gaussian noise to the central server, resembling a generic Gaussian attack [67]. This behavior undermines model accuracy by introducing noise into the aggregation process. In contrast, SFR attackers possess private data and computational abilities but choose not to contribute these resources. For instance, SFR attackers may employ advanced strategies like delta weights attacks, generating gradient updates by subtracting two global models from previous rounds [68], or submit systematically crafted fake parameters [69]. While delta weights attacks ensure convergence of the aggregated model, they maintain stealth by mimicking benign updates [70]. Even simpler methods, such as consistently returning the same global model parameters, can degrade model performance and reduce fairness in FL [23]. These attacks pose significant threats in real-world scenarios, especially in sensitive domains like healthcare or finance, where FL's integrity is crucial [67]. From an implementation perspective, a free-rider can bypass local training entirely by returning either unchanged or random parameters while continuing to download global updates. These minimal modifications exploit the aggregator's inherent trust in each client. PASS [66] and similar auditing approaches evaluate each client's historical gradient contributions against their impact on model improvements. Clients that fail to provide meaningful updates risk detection or a reduced aggregation weight. These scoring mechanisms discourage free-riders by linking model benefits to local effort.

3.1.3. Jamming Attacks

Jamming attacks pose a severe security threat in wireless networks, particularly decentralized FL (DFL) environments [71]. These attacks involve adversaries emitting interference signals to disrupt communication between legitimate nodes, hindering the exchange of critical data such as local model parameters. For example, in real-world scenarios like airport operations, jamming has led to significant disruptions in communication systems, delaying processes and compromising operational efficiency [72]. In blockchain-based decentralized FL, jamming attacks prevent normal miners from receiving necessary data, excluding them from proof-of-work computations. This gives malicious miners an advantage in controlling the blockchain by increasing the probability of generating a longer malicious block stream, especially when the number of attackers surpasses normal miners [73]. Additionally, targeted jamming in decentralized FL can isolate nodes by disrupting key communication links. This isolation fragments the network, delaying learning processes and degrading model accuracy due to insufficient data exchange [72]. For instance, simulations of such attacks on multi-hop wireless networks have demonstrated significant reductions in DFL performance by exploiting vulnerabilities in connectivity and model sharing [72]. Realistic jamming can be emulated by imposing network drop rates or forced timeouts in each training round. Indeed, attackers might saturate specific channels, delaying or preventing the arrival of local updates to the server. From the defense perspective, coded computations (e.g., CodedPaddedFL [74]) or asynchronous protocols allow partial aggregation even if a subset of updates is lost or late. Additionally, some FL systems introduce fallback communication channels to bypass jammed links. These solutions provide a certain level of robustness to partial network disruption.

3.1.4. Evasion Attacks

Evasion attacks exploit weaknesses in model predictions during inference by introducing carefully crafted adversarial inputs, such as pixel perturbations, without altering the training process [75]. For instance, unnoticeable changes to a panda image can cause GoogLeNet to misclassify it as a gibbon with 99.3% confidence [76]. These attacks undermine the reliability of FL systems by reducing model accuracy and trustworthiness. In real-world scenarios, evasion attacks can deceive spam filters or recommendation systems trained via FL, leading to financial or operational harm [77].

In VFL, Pang et al. [78] demonstrate the susceptibility of VFL systems to ADIs, which manipulate joint inference outcomes to prioritize an attacker's input. They employ gradient-based methods and grey-box fuzz testing to uncover vulnerabilities in privacy-preserving features, revealing that adversaries can exploit these to skew results. For example, ADIs could be used in financial applications to favor fraudulent transactions. To address these threats, Kim et al. [77] analyze internal evasion attacks across learning methods, showing that personalized federated adversarial training enhances robustness by 60% compared to standard approaches. This demonstrates that tailored defenses can mitigate attack impacts even under constrained resources, though challenges remain in balancing accuracy and security. To carry out such an attack, malicious entities could leverage adversarial example implementations to craft feature-level perturbations. Only minor changes to the inference pipeline could be enough to cause misclassifications in the global model. To mitigate the impact, using personalized adversarial training [77] allows for retraining on adversarial variants each round, though at a higher computational cost.

3.1.5. Straggling

Sometimes, due to various factors like limited computing resources, background processes, memory constraints, or unstable wireless communication, certain edge devices, known as stragglers, might perform significantly slower than others, thereby deteriorating the FL process. This vulnerability can also be exploited by adversaries through free-riding attacks, where malicious clients intentionally delay or avoid computations to degrade system performance [79]. Waiting for model updates from these slower clients at each learning step can slow down model convergence and degrade accuracy. For instance, attackers may inject noise into updates or mimic benign clients to amplify delays, leading to inefficient resource utilization as faster clients idle [80]. Ignoring updates from stragglers risks model accuracy and client drift – a phenomenon where local models diverge significantly due to non-identically distributed data. Real-world manifestations include healthcare FL systems where malicious clients disrupt timely updates, jeopardizing critical applications like disease prediction. In terms of implementation, simple modifications in local training scripts can pause or throttle GPU usage, slowing progress. Therefore, the design of asynchronous or coded protocols is required to reduce reliance on a strict round barrier. If certain clients are repeatedly late or absent, they can be down-weighted or removed from the aggregator's pipeline. Nonetheless, balancing the fair inclusion of actual slow clients against malicious stragglers remains a key design challenge in practical FL settings.

3.1.6. Dropout

User dropout in FL refers to the scenario in which some clients drop out or become inactive during training. This phenomenon can occur due to network issues, client failures, or intentional withdrawal. Honest clients may become demotivated to engage in the training process if the collaborative framework does not guarantee fairness for all clients [81]. Beyond these general challenges, dropout can also manifest as an attack, where malicious clients intentionally withdraw at critical training stages to disrupt the global model's convergence. Such targeted dropout attacks can exacerbate biases in the model if specific clients with unique data distributions are affected, leading to skewed performance [74]. For instance, in real-world scenarios like healthcare applications, the dropout of clients representing minority populations could result in a poorly performing model on underrepresented groups. In code, dropout simulates a failure to send updates by skipping the aggregator's communication calls. Defensive solutions require tracking dropout patterns over time to determine if certain clients drop out from training at crucial convergence stages. The integration of partial reweighting or client selection [82] may reduce the damage, though guaranteeing fairness if many dropouts occur remains non-trivial.

We would like to note that while straggling and dropout are not traditionally categorized as intentional attacks in FL (highlighted in gray on the taxonomy of Fig. 4), they represent significant challenges that can hinder the overall learning process. However, it is important to note that these phenomena could also be exploited by adversaries in a malicious context. An attacker could deliberately induce straggling by compromising clients or resources or cause dropout by intentionally withdrawing specific clients to disrupt the training process.

3.2. Security Defenses

This section provides an overview of security mechanisms designed to enhance the robustness of FL systems against various adversarial threats. It highlights key strategies, including robust aggregation operators, anomaly detection techniques, and adversarial training.

3.2.1. Robust Aggregation Operators

FedAvg is one of the most popular algorithms used in FL to aggregate client model updates. However, several studies have shown that this method can be sensitive to various types of attacks, including model poisoning attacks, where some clients might send malicious updates, or data poisoning attacks, where the data used to train local models is manipulated to bias the global model [83] [84]. Robust aggregation operators have been developed to enhance security and defend against such attacks. These operators are designed to minimize the impact of malicious or noisy updates, thereby improving the resilience of the FL system.

- **Trimmed Mean** involves calculating the average of model updates after removing a specified percentage of the highest and lowest values. This method helps mitigate the impact of outliers but can be circumvented by poisoning attacks that exploit high empirical variance among client updates, as demonstrated by "A Little Is Enough" [85]. This solution also mitigates the reduction in performance caused by non-IID data by removing extreme values from clients whose distributions differ significantly from the rest.

- **Median**-based algorithms replace the arithmetic mean with the median of model updates, choosing the value representing the distribution's center. This approach is less sensitive to extreme values and more resistant to adversarial attacks compared to methods like FedAvg. This approach also improves the model performance under high non-IID data since it aims to avoid the influence of highly different distributions (a.k.a outliers). However, it is vulnerable to attacks such as IPM, which can negatively impact the inner product between the true gradient and the aggregated gradients [85]. **GeoMed (Geometric Median)** minimizes the sum of Euclidean distances to all points, offering a central point that is less sensitive to outliers compared to the mean [86]. Its more computation-efficient version is called Medoid [84]. GeoMed can tolerate up to half of the malicious clients and estimate true parameters, showing convergence properties in gradient descent methods. However, GeoMed is sensitive to model poisoning attacks and less robust with imbalanced datasets. To address these issues, Li et al. [87] proposed **Auto-Weighted GeoMed (AutoGM)**, which automatically excludes extreme outliers and re-weights remaining points based on a user-specified skewness threshold. AutoGM maintains high performance even with up to 30% of nodes engaging in model poisoning or 50% experiencing data poisoning attacks. **Marginal Median (MarMed)** [84] focuses on the median of marginal distributions of data points, filtering out extreme values to provide a stable estimate of central tendency. This approach, similar in robustness to the geometric median but with a distinct handling of data, helps maintain the integrity of the aggregation process against adversarial manipulations. **Mean Around Median (MeaMed)** [84] is a trimmed average method that centers calculations around the median, effectively reducing the impact of outliers and adversarial data. Blending the strengths of both the mean and median offers a balanced approach to maintaining performance and robustness in distributed learning scenarios vulnerable to Byzantine attacks.

- **Krum**, introduced by Blanchard et al. [83], selects a model update vector that is least affected by outliers by minimizing the sum of squared distances to its $n - f$ closest neighbors, where f is the maximum number of Byzantine workers tolerated. **Multi-Krum** (or m-Krum) extends this approach by considering multiple vectors, thus enhancing robustness by aggregating d parameter vectors instead of just one. Despite its effectiveness in mitigating high-severity attacks, Han et al. [88] found that Krum struggles with RNNs due to variability in local models caused by sequential data and recurrent structures. Additionally, Krum's reliance on strong assumptions, such as bounded absolute skewness, may not always be realistic, and it is vulnerable to newer attacks like IPM and "A Little Is Enough" (ALIE), which exploit empirical variances between client updates [85].

- **Bulyan** enhances existing Byzantine-robust aggregation techniques, such as Krum and GeoMed, by first compressing gradient updates from each client into a more compact form. This reduces the impact of noise and malicious data. After compression, Bulyan employs a robust aggregation technique to combine the compressed updates, focusing on reliable information while filtering out outliers and adversarial contributions, thereby improving accuracy and resilience against Byzantine faults [89].

- **Clustering** aggregation calculates pairwise cosine distances between parameter updates and groups clients based on cosine similarities using agglomerative clustering with average linkage. While this method shows robustness in some scenarios, it only considers the relative directions of updates, ignoring their magnitudes. Attackers can exploit this by amplifying their updates without altering directions, disrupting model convergence. To address this, Li et al. [85] proposed **ClippedClustering**, which applies norm-based clipping to updates. Updates are scaled if their norm exceeds a server-determined threshold, set automatically using the median of historical update norms, improving defenses under IID local datasets. However, ClippedClustering significantly degrades performance with non-IID datasets, highlighting the need for tailored defense strategies.

- **Zeno** [90] scores and ranks updates based on their alignment with a reference gradient, filtering out suspicious updates dynamically. Zeno is particularly effective in resisting Byzantine attacks because it relies not solely on traditional statistical measures like medians or means. Instead, it actively evaluates the credibility of each update, allowing it to reject harmful contributions dynamically. In contrast to previous work, Zeno++ [91] removes several unrealistic restrictions on worker-server communication, now allowing for fully asynchronous updates from anonymous workers, for arbitrarily stale worker updates, and for the possibility of an unbounded number of Byzantine workers.

- **Anomaly Detection** It employs various statistical and analytical methods to identify events that deviate from expected behavior, which is crucial for detecting Byzantine attacks. Effective anomaly detection systems require a normal behavior profile to recognize malicious activity. Techniques might include clustering to group similar updates and identify outliers, Euclidean distance metrics used in methods like Krum for detecting deviations in input parameters, Autoencoders that reconstruct data to flag abnormal updates, and other methods [23]. For example, Jiang et al. [92] proposed monitoring the average loss reported by clients to identify and exclude potentially compromised updates caused by Sybil attacks. Pan et al. [93] proposed integrating advanced anomaly detection techniques with a unique model update aggregation strategy, enabling the identification and neutralization of backdoor influences in a single update cycle, avoiding the need for extensive data access or communication between clients. Since non-IID data can make normal client updates appear like anomalies, which attackers may exploit, adaptive anomaly detection methods such as the one proposed by Jiang et al. [92] and Pan et al. [93] help to differentiate between natural variations in data and adversarial manipulations.

3.2.2. Asynchronous Schemes

To mitigate the straggler issue, various asynchronous schemes have been proposed. These schemes update the global model based on the time difference between the current round and the previous round when the client first received the global model [80]. For example, Lu et al. [94] proposed FedAAM, which employs an adaptive weight allocation algorithm that assigns dynamic weights to client updates based on their contribution, considering factors such as the timeliness and quality of the updates. The framework introduces two asynchronous global update rules based on a differentiated strategy, allowing the global model to update with varying client contributions depending on their performance and the frequency of their updates. Additionally, FedAAM integrates global momentum by using the historical global update direction, which helps smooth the global update process and manages the asynchrony among clients, thereby improving training efficiency and convergence behavior. However, Schlegel et al. [74] report that these schemes generally do not converge to the global optimum. They further propose two schemes to avoid this problem. CodedPaddedFL combines one-time padding with gradient codes to ensure straggler resiliency while maintaining privacy, achieving an 18x speed-up for 95% accuracy on the MNIST dataset. CodedSecAgg, based on Shamir's secret sharing, provides both straggler resiliency and robustness against model inversion attacks, outperforming the state-of-the-art LightSecAgg by a speed-up factor of 6.6-18.7 for similar accuracy.

3.2.3. Pruning

Pruning can serve as both an optimization strategy and a potential security measure in FL. It is a technique used to reduce the size and complexity of machine learning models by removing less important or dormant neurons and connections. This process helps address the computational and communication constraints typical in FL environments, where clients often have limited resources [23]. Additionally, selective pruning can enhance security and mitigate backdoor attacks by removing neurons that are not activated by clean data. However, this defense method may be less effective if attackers use pruning-aware methods [62].

3.2.4. Adversarial Training

Adversarial training in FL is a defense mechanism in which each client generates adversarial examples locally during training to enhance the robustness of their model updates against adversarial attacks. These adversarially trained local models are then aggregated by the central server, allowing the global model to learn to resist adversarial inputs without directly exposing client data, thereby improving security in a distributed and privacy-preserving manner. Li et al. [95] formulated the training process as a min-max optimization problem, addressing the unique challenges of decentralized data and model training. They also provided a detailed convergence analysis, demonstrating that the minimum loss can converge to a small value under appropriate conditions, and introduced gradient approximation techniques to enhance training effectiveness, particularly for non-IID clients.

3.2.5. Personalized Solutions

This section covers customized solutions that do not fit the categories discussed above. It highlights the most novel and promising methods from recent research, showcasing innovative approaches to enhancing security in FL environments.

FoolsGold It is a defense method specifically designed to counter targeted poisoning sybil attacks. It identifies clients with similar behavior and characteristics of Sybil clones. It then adapts the learning rates of these clients based on the similarity of their contributions, effectively reducing the influence of malicious updates and mitigating the attack [60] [96]. This technique also provides a way to tackle non-IID data issues presented in Section 2.3, detecting overrepresented gradients and down-weighting contributions from clients that exhibit unusually high similarity, ensuring fairer aggregation despite heterogeneous data variations. However, when legitimate updates are similar, these methods also tend to penalize them, causing significant drops in model performance [97]. Some experiments have shown that FoolsGold might completely fail to train the model, potentially eliminating important local models. Additionally, the method may encounter limitations when integrated with large language models due to the substantial cache requirements needed to memorize intermediate results, such as models from previous FL training rounds [88].

FL-Defender [97]: Proposed by Jebreel et al., FL-Defender is a defense mechanism designed to combat targeted poisoning attacks, specifically addressing label-flipping (a type of poisoning attack) and backdoor attacks. Similarly to FoolsGold, FL-Defender extracts last-layer gradients from workers' updates and calculates cosine similarities to detect attack patterns, followed by dimensionality reduction using PCA to focus on the most relevant features. During aggregation, it reweights updates based on historical deviations, minimizing the influence of poisoned data while preserving model performance and maintaining low computational overhead. Its aggregation method is similar to Krum and Trimmed Mean. However, FL-Defender adds an adaptive component by re-weighting updates rather than rejecting them outright.

FLAME [65]: It is a robust defense framework designed to counter backdoor attacks while preserving the benign performance of the global model, even in non-IID data settings. Unlike traditional defenses that rely on limited attacker models or degrade performance with excessive noise, FLAME dynamically estimates and injects the optimal amount of Gaussian noise to eliminate backdoors. Its clustering-based approach effectively separates malicious updates from benign ones, ensuring robust aggregation despite data heterogeneity. Additionally, weight clipping limits the influence of outlier updates, enabling FLAME to maintain high model accuracy while efficiently removing adversarial backdoors.

PAMPAS [98]: Ching et al. proposed to combat GAN attacks by partitioning the model between users and edge servers, with users training only part of the model to enhance security and efficiency. Their approach seeks to optimize model partitioning to resist GAN attacks and minimize total training time while addressing the trade-offs between computation, transmission, and maintaining data privacy.

PPFDL [99]: Xu et al. proposed a solution designed to reduce the negative impact of irregular users (Users who join and leave the training process frequently or unpredictably) on training accuracy by prioritizing high-quality data contributions. The approach ensures the confidentiality of user information using Yao's garbled circuits and additively homomorphic cryptosystems. PPFDL is also robust against user dropout, allowing the training to continue as long as some users remain online.

LeadFL [100]: Zhu et al. proposed a client-side defense mechanism against backdoor and poisoning attacks, which introduces a novel regularization term in local model training to nullify the Hessian matrix of local gradients. Additionally, the regularization helps to tackle non-IID issues explored in Section 2.3 by neutralizing adversarial gradient patterns, improving robustness against backdoor and targeted attacks in heterogeneous data settings. Unlike existing defenses, LeadFL specifically targets the Hessian matrix to enhance robustness against bursty adversarial patterns, effectively handling the high variance in malicious client activity that many server-side defenses struggle with. Designed to work alongside existing server-side defenses, LeadFL enhances overall security by complementing other mechanisms rather than functioning as a standalone solution.

PASS [66]: To address Free-Rider attacks in FL, the paper introduces the Parameter Audit-based Secure and Fair FL Scheme (PASS). PASS employs a privacy-preserving strategy (PASS-PPS) incorporating weak DP with a Gaussian mechanism and a parameter prune mechanism to protect data during parameter auditing. Additionally, PASS utilizes a novel contribution evaluation method to accurately measure each client's performance, ensuring fairness in the training process and deterring both AFR and Selfish SFR attacks.

Sageflow [80]: It introduces a staleness-aware grouping method that integrates seamlessly with robust aggregation rules such as Multi-Krum. This approach enhances resilience against adversaries through entropy filtering and loss-weighted averaging, effectively managing non-IID data distributions and outperforming previous methods like Zeno+ in practical scenarios.

FedRLChain [81]: It leverages blockchain technology to address critical challenges in Federated Reinforcement Learning. This framework features a novel verification algorithm to counter malicious client actions, an aggregation weight scheme to avoid bias in the global model, and an enhanced FedAvg algorithm for improved convergence speed.

In Table 6, we provide a relation between the defense mechanisms and the corresponding attacks or threats they aim to mitigate in FL. Certain defenses, like robust aggregation operators and anomaly detection, address various attacks such as poisoning, GAN-based, Sybil, backdoor, free-riding, jamming, and evasion. Asynchronous schemes, pruning, and personalized solutions focus more specifically on addressing straggling and dropout issues related to client heterogeneity and connectivity.

	Attacks/Threats								
Defenses	**Poisoning**	**GAN-based**	**Sybil**	**Backdoor**	**Free-riding**	**Jamming**	**Evasion**	**Straggling**	**Dropout**
Robust aggregation operators	✓	✓	✓	✓	✓		✓		
Asynchronous schemes	✓					✓		✓	✓
Pruning	✓			✓				✓	
Adversarial training	✓	✓		✓			✓		
Personalized solutions	✓		✓	✓	✓			✓	✓

Table 6: Relationship of defense mechanisms and attacks for secure FL

Thus, Table 6, together with the analysis of the papers assessed, reveals key insights into the current security landscape in FL by illustrating the effectiveness of various defense mechanisms against different types of attacks. A notable observation is the dominance of robust aggregation operators and anomaly detection techniques, which address the broadest range of threats. This suggests that adversarial manipulations, particularly poisoning, Sybil, and backdoor attacks, remain central concerns in FL security. However, these methods alone are not sufficient. For example, while robust aggregation enhances resilience against model and data poisoning, it does not directly counter jamming attacks, which disrupt communication rather than manipulate training data.

Table 6 also highlights an ongoing challenge: no single defense mechanism can comprehensively mitigate all security threats, emphasizing the need for hybrid approaches. Personalized solutions and adversarial training offer promising advances by tailoring security mechanisms to specific attack vectors, but they remain underexplored in the context of free-riding and evasion attacks. The increasing sophistication of FL attacks necessitates continuous refinement of defense strategies, integrating multiple techniques to address emerging adversarial tactics holistically.

Lessons learned: The analysis of security attacks and defenses in FL reveals several critical insights. First, the growing sophistication of adversarial strategies highlights the need for adaptive and multi-layered defense mechanisms. While robust aggregation operators and anomaly detection remain foundational defenses, adversaries continuously develop novel poisoning and backdoor attack strategies that evade traditional filtering methods. This aspect highlights the limitations of static, threshold-based defenses and motivates the need for more adaptive, real-time techniques. Second, client heterogeneity and participation dynamics significantly impact FL security, as attackers can exploit phenomena like straggling, dropout, and free-riding, emphasizing the importance of personalized and incentive-aligned solutions. However, most current solutions assume honest or uniformly distributed clients, leaving a gap in defending against adversarial heterogeneity and collusion.. Additionally, adversarial tactics such as Sybil-based collusion and GAN-powered attacks' rapid evolution underscores the necessity for continuous monitoring and adaptive countermeasures. However, there is no standardized framework to evaluate these defenses across diverse threat types. Finally, while many defenses focus on protecting global model integrity, there is a growing need for client-side security solutions to detect and mitigate threats locally before aggregation.

3.3. Comparative Analysis of Defenses for Secure FL

In this subsection, we analyze quantitative and experimental studies to evaluate the effectiveness of the previous defenses in specific scenarios. For instance, studies such as Li et al. [101] and Zhang et al. [102] offer a detailed quantitative comparison of various FL defense mechanisms across different attacks. Under untargeted model poisoning attacks like the Fang et al. [44] attack, Bulyan demonstrates superior robustness, achieving up to 85% global model accuracy with a 20% adversarial client ratio, compared to Trimmed Mean (78%) and Krum (75%). However, Bulyan's computational complexity ($O(dn^2)$ may hinder scalability in large-scale FL systems. For targeted backdoor attacks, FLTrust, which uses a small trusted dataset, achieves over 90% accuracy on benign tasks while suppressing backdoor success rates below 5%, outperforming Trimmed Mean, which achieves 85% benign accuracy but struggles with backdoor suppression. FLAME emerges as a strong candidate in highly heterogeneous data settings by dynamically adds noise to mitigate backdoors while maintaining model performance at around 88% accuracy. Trimmed Mean balances simplicity and effectiveness for scenarios prioritizing low overhead. Thus, the choice of defense depends on the attack type and system constraints: Bulyan is recommended for untargeted attacks in smaller systems, while FLTrust and FLAME are preferred for targeted attacks or non-IID data distributions.

Beyond robust aggregation, the literature reports competitive results for the remaining four defense families in our taxonomy. *Asynchronous schemes* such as FedAAM [94] and CodedPaddedFL [74] address stragglers and jamming by updating the global model as soon as partial gradients are available. On MNIST, CodedPaddedFL provides around 95% accuracy while delivering an 18× reduction in time compared with synchronous FedAvg in settings with slow or jammed clients; the cost is roughly a two-fold increase in uplink bandwidth due to coded padding. Moreover, *Pruning-based defenses* remove dormant or highly suspicious neurons after each aggregation round. [103] shows that neuron pruning can cut a Fashion-MNIST backdoor attack success rate (ASR) from 99.7% to 2%. *Adversarial training* hardens the model against inference-time manipulations. For example, pFedDef [77] improves robustness PGD perturbations by roughly 60% on CIFAR datasets while maintaining competitive clean accuracy. Finally, *personalised solutions* mitigate Sybil and free-rider behaviour. For example, Sageflow [80] further combines personalised weighting with entropy filtering, yielding a 12% improvement in convergence speed under mixed Sybil-plus-straggler settings.

These results confirm that each defense family excels under specific threat models and resource budgets: asynchronous protocols prioritise liveness, pruning targets stealthy backdoors, adversarial training bolsters prediction-time robustness, and personalised auditing enforces fairness against Sybil or free-riding behaviour. A balanced deployment should therefore mix complementary mechanisms—for example, pairing Bulyan with FedAAM for integrity *and* liveness, or coupling Trimmed-Mean with pFedDef to resist both poisoning and evasion—rather than relying on robust aggregation alone.

4. Privacy in FL

Following the taxonomy depicted in Fig. 4, we describe the main attacks and defenses for privacy in FL in this section. For the attacks, we provided specific mechanisms, degrees of harm, and specific examples of manifestations in the real world. At the end of each subsection, we provide some lessons learned after analyzing the papers regarding attacks and defenses for privacy in FL.

4.1. Privacy Attacks/Threats

In ML, privacy attacks and threats refer to techniques or strategies used by adversaries to compromise the privacy of individuals or sensitive data during the training or inference phase of ML models. These attacks aim to exploit vulnerabilities in the ML process to gain unauthorized access to private information or infer sensitive attributes of individuals [104]. In the FL area, privacy attacks refer to attempts by adversaries to compromise data privacy during the training process. These attacks allow extracting sensitive information from local or aggregated global models to infer, reconstruct, or cause data leakage. In particular, we categorize such attacks based on the following four main dimensions. First, *method of inference* distinguishes between *passive* attacks (e.g., gradient inversion), which rely on observing shared updates without injecting malicious behavior, and *active* attacks (e.g., canary gradient), which manipulate or perturb updates to increase data leakage. Second, *phase affected* differentiates between leaks that occur predominantly during training (such as gradient-based reconstruction) and those emerging at inference time (like membership inference on final model outputs). Third, *adversary's role* clarifies whether an attacker is an *insider*–a legitimate FL client with access to local computations–or an *outsider* who intercepts or eavesdrops on communication, for instance, through man-in-the-middle tactics. Lastly, the *attack scope* specifies whether an attack is *single-round*, occurring once (e.g., a single instance of eavesdropping), or *multi-round*, gradually accumulating sensitive information over multiple iterations (as in repeated gradient inversion attempts). Thus, Table 7 summarizes how each known privacy threat fits into these four dimensions. When a threat spans multiple categories (for example, exhibiting both passive and active behaviors), we explicitly mark that overlap.

The following sections explore the most relevant attacks on privacy in FL by defining their nature, objectives, consequences, and examples proposed in the literature.

4.1.1. Gradient Manipulation

Gradient manipulation in FL involves exploiting shared gradients to infer or reconstruct sensitive data, posing significant privacy risks. This includes techniques like gradient inversion, reconstruction through inference, and canary attacks, highlighting vulnerabilities in FL's gradient-sharing mechanisms.

Gradient inversion attacks. These attacks exploit gradients or weight updates shared during the aggregation process in FL to reconstruct private data, posing significant privacy risks [15]. These attacks typically leverage optimization techniques or linear relationships between gradients and inputs to infer sensitive information. For instance, Kariyappa et al. [105] introduced the Cocktail Party Attack (CPA), which uses independent component analysis to recover private inputs from aggregated gradients, demonstrating its scalability to large batch sizes. This highlights how gradient inversion can compromise privacy even in high-dimensional settings. Li et al. [106] proposed the End-to-End Gradient Inversion (E2EGI) attack, which iteratively reconstructs training data by reversing gradients, showcasing its potential to breach privacy across multiple iterations. Pasquini et al. [107] further explored two variants: a passive optimization-based approach that infers private training sets without active interference and an active attack that manipulates model updates to amplify privacy leakage. These methods underline the nuanced mechanisms attackers employ to exploit gradients.

The consequences of gradient inversion attacks are severe. In real-world scenarios, such attacks can expose sensitive medical images or financial records used in FL systems, violating privacy regulations and enabling misuse [15]. However, Huang et al. [108] argue that practical risks may be mitigated by factors like large batch sizes and local iterations, which reduce reconstruction fidelity. Similarly, Boenisch et al. [109] observed that gradient inversion often suffers from local minima and requires extensive iterations for meaningful data recovery, limiting its feasibility in some production environments. Implementation typically requires intercepting aggregated gradients and running a local optimization loop that refines random inputs until the gradients match observed signals. Encrypting or clipping gradients partially hinder this by reducing the attacker's visibility or precision, though some accuracy trade-offs may arise.

In contrast, *gradient suppression attacks* involve maliciously suppressing gradients during aggregation to manipulate global model updates [107]. By isolating individual updates, attackers can amplify specific patterns in user data, increasing exposure risks. Such attacks can infer the presence of specific data points in user datasets, enabling targeted tracking [110]. While their mechanisms differ from gradient inversion, suppression attacks similarly exploit vulnerabilities in gradient-sharing protocols. Implementation of gradient suppression often involves intercepting or nullifying certain gradient components before sending them to the server, typically by modifying the local backward pass. A partial defense strategy is to rely on cryptographic checks that ensure gradient consistency across dimensions, thereby preventing an attacker from selectively masking or removing critical features.

Reconstruction through inference. It is a privacy-threatening scenario where an adversary attempts to reconstruct or infer sensitive information about the training data of individual clients by analyzing the model updates or outputs shared during the FL process [19]. Such attacks exploit the inherent vulnerability of gradient-sharing mechanisms in FL. For instance, adversaries can reverse-engineer specific data points or patterns from gradients using techniques like gradient inversion, as demonstrated by Chen et al. [111]. They identified two distinct types of reconstruction attacks. The first, called extraction attack, focuses on accurately reconstructing a single training sample with minimal computational cost. This attack leverages advanced optimization techniques to improve reconstruction accuracy, posing significant risks to data privacy. The second type, manipulating reconstructed data, allows adversaries to recover private training data and labels from gradients and subsequently modify this data to execute targeted attacks on models. For example, in healthcare FL applications, attackers could reconstruct sensitive medical images shared across hospitals and manipulate them to mislead diagnostic models [112]. Attackers mostly rely on final model outputs or partial gradient snapshots for offline reconstruction, requiring minimal changes to the FL pipeline. Defensive measures like gradient masking or cryptographic aggregation reduce the granularity of the information available, limiting reconstruction success.

Canary Gradient. A canary gradient attack is a privacy breach in FL where an attacker exploits gradients or weight updates shared during the aggregation process to infer sensitive information. Its name originates from using canaries in coal mines

Table 7: Categorization of Privacy Attacks in FL

Privacy Attack	Method (Passive vs. Active)	Phase Affected	Adversary's Role	Attack Scope	Goal / Effect
Gradient Inversion	Passive	Training	Insider	Multi-Round	Reconstruct private data
Gradient Suppression	Active	Training	Insider	Multi-Round	Amplify data leakage patterns
Membership Inference	Passive or Active	Inference	Outsider / Insider	Single-Round	Check if a data sample was used in training
Canary Gradient	Active	Training	Insider	Multi-Round	Insert small triggers to deduce sensitive info
Model Inconsistency	Active	Training	Insider (Server)	Single-Round	Compare user updates across different models
GAN-based Inference	Active	Training	Insider	Multi-Round	Generate synthetic data that reveals distribution
Eavesdropping	Passive or Active	Training / Inference	Outsider	Single-Round or Multi-Round	Intercept model updates or network traffic
Unintentional Data Leakage	Passive	Training	Outsider / Insider	Single-Round or Multi-Round	Exploit unintended gradient exposure

to detect poisonous gases [110]. In this attack, the adversary injects small, carefully crafted perturbations into a client's gradients or weight updates and observes the server's response to deduce private data. For instance, such attacks have been shown to reconstruct sensitive client data under certain conditions, raising concerns about FL's privacy guarantees [113]. Maddock et al. [113] propose CANIFE, a method to evaluate empirical privacy risks in FL by introducing adversarially crafted "canary" samples. These samples are used to measure model exposure to privacy breaches, revealing that the empirical per-round privacy loss is significantly tighter than theoretical bounds. This approach highlights vulnerabilities in FL systems, such as susceptibility to gradient inversion attacks in real-world scenarios, which theoretical DP guarantees may underestimate. By offering a realistic assessment of privacy risks, CANIFE underscores gaps in current defenses and emphasizes the importance of robust threat models for FL. To carry out a canary attack, an adversary could inject imperceptible 'signatures' into local gradients, then checks if these signatures reappear in the global model's updates. Clipping (see Section 4.2 or encrypting gradients dilutes such signatures, minimizing the attacker's ability to confirm the presence of sensitive data.

4.1.2. Membership Inference

In this privacy threat, an adversary seeks to determine whether a specific data point was part of a client's training dataset used in the FL process. Such an attack primarily aims to verify the membership status of individual data points, discerning whether they belong to a client's private training data [22]. This breach of privacy may lead to the disclosure of identities or sensitive attributes, undermining the confidentiality and anonymity of data contributors. Zhang et al. [114] highlights two types of membership inference attacks with distinct mechanisms and implications. Poisoning membership inference attacks involve adversaries injecting carefully crafted malicious samples into the training data to detect membership. For instance, by observing how poisoned examples alter model loss, attackers can infer membership, posing severe risks in healthcare FL systems where patient data is highly sensitive. Black-box membership inference attacks, such as Memguard [115], operate without direct access to training data or models. These attacks generate adversarial queries to exploit model predictions and infer membership, which could compromise user anonymity in recommendation systems. Pasquini et al. [107] emphasize that adversaries can leverage model updates or query responses to enhance their guesses. The harm caused by these attacks extends beyond privacy breaches, as they can facilitate further privacy threats like attribute inference attacks, creating a cascading effect on the overall security of FL systems. Specifically, membership inference often queries the global model's confidence scores for specific inputs. Small modifications to the local or server-side scripts can track these score patterns, exposing training-set membership. Defensive techniques such as local DP or randomizing confidence outputs inhibit the attack's reliability.

4.1.3. Model Inconsistency

This attack exploits a vulnerability in the FL protocol caused by incorrect usage of secure aggregation and a lack of parameter validation. Specifically, a malicious server distributes different versions of the model to different users within the same training round. The server can analyze behavioral differences in user updates, even though these updates are securely aggregated. The attack leverages the fact that varying model parameters can induce detectable differences in gradient updates, which may reveal sensitive training data. For instance, Pasquini et al. [110] demonstrated that this approach enables inference of private information regardless of the number clients. Real-world implications include risks to applications like healthcare and IoT, where sensitive data is prevalent. Zhang et al. [116] further noted that inconsistencies between global and local models could reflect attack-related information, potentially guiding personalized FL algorithms to improve fault diagnosis accuracy. These findings underscore the severe privacy risks of model inconsistency attacks and their potential to compromise FL systems at scale. The implementation of such an attack only requires slight server-side changes–assigning slightly different model parameters to each client in a single round. A recommended mitigation is verifying model consistency across clients or leveraging secure multi-party aggregation to ensure identical parameter distributions.

4.1.4. GANs-based Inference

A GANs-based inference attack in FL uses GANs to infer sensitive information about the training data held by individual clients. This attack aims to create a generator network that can produce data samples indistinguishable from the data used for training in the local client models [15]. The attacker can effectively determine whether a specific data point was part of a client's training dataset, conducting membership inference. For example, in medical diagnosis scenarios, such attacks could allow a malicious client to infer sensitive patient conditions from gradients shared during FL updates [117]. The consequences of such attacks are severe, as they breach privacy by revealing which data points were used during training, leading to potential misuse or discrimination risks. Huang and Xiang [117] introduce Cross-Client GANs (C-GANs) attacks, where a malicious client reconstructs samples resembling the distribution of other clients' training data. This enables adversaries to compromise privacy by leaking benign clients' sensitive data, as demonstrated in experiments involving reconstructed images from medical datasets. Attackers usually train a local generator alongside the global model, refining synthetic samples that mimic real client data. By limiting gradient visibility with cryptographic or noisy protocols, such as DP, defenders hamper the generator's ability to converge on sensitive distributions.

4.1.5. Eavesdropping

In this threat, an attacker intercepts the communication between the clients and the central server. It is done by sniffing the network traffic or by compromising the devices of the clients or the server [118]. The objectives of an eavesdropping attack often include stealing the FL model, inferring sensitive information from the client's data, and disrupting the FL process. Guo et al. [119] identifies four types of eavesdropping attacks: Passive Eavesdropping, where an attacker monitors communication without altering data; Active Eavesdropping, involving data modification, such as injecting malicious gradients; Man-in-the-Middle (MitM) Attacks, which intercept and relay messages, posing significant risks to data integrity and confidentiality; and Network Sniffing, capturing network traffic to extract sensitive information. Practical eavesdropping often exploits unsecured Wi-Fi or inadequate encryption. An attacker needs little more than packet-capture tools to observe local updates. Configuring Transport Layer Security (TLS) or implementing fully HE (FHE) for parameter exchanges effectively thwarts passive intercepts.

4.1.6. Unintentional Data Leakage

The latter is not precisely an attack, but it is more of a vulnerability that attackers can exploit once discovered. It occurs when private training data leaks through the gradient-sharing mechanism deployed in FL systems. The objective of this is to recover batch data from the shared aggregated gradients. The latter can be catastrophic and lead to users' private data reconstruction by eavesdropping on shared gradients. The risk of confidential data leaking from the training gradients in standard FL, especially the vertical case, is high. For Nair et al.[120], these vulnerabilities are a concern in FL due to the potential for data leakage and adversarial attacks during gradient transfer operations. These threats can occur when gradients are transferred between clients in the FL system. Ziz et al. [121] introduce the CAFE (catastrophic data leakage in vertical FL) attack, an advanced data leakage attack in FL that aims to recover private data from shared aggregated gradients. It addresses the limitations of existing approaches regarding scalability and theoretical justification for data recovery. The attack algorithm consists of three steps: (1) Recovering the loss gradients concerning the outputs of the first fully connected (FC) layer. (2) Using the recovered gradients as a learned regularizer to improve the performance of the data leakage attack. (3) Using the updated model parameters to perform the data leakage attack. Such leakage arises when partial gradient details or intermediate layer outputs inadvertently reveal private features. Minimizing or masking these signals (via secure aggregation or randomization) lowers the precision with which attackers can reassemble original training data.

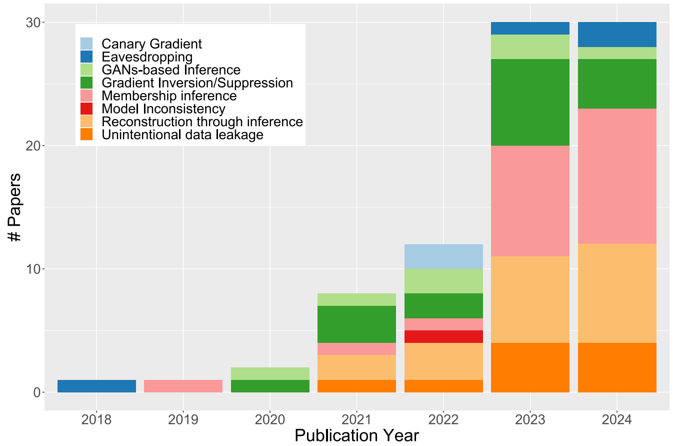

Figure 6: Papers related to privacy attacks over time

Leveraged on the literature review done, we retrieved a list of privacy threats included in the current literature. Fig. 6 demonstrates a notable increase in privacy attack research publications, reflecting heightened awareness and concern in this field. The early years of 2018 until 2020 saw minimal research output with a narrow focus on attack types. However, a marked shift occurred since 2021, with a substantial rise in the quantity and range of studies. Attacks like Membership inference and Reconstruction through inference gained significant traction, particularly in 2023 and 2024. Concurrently, research on GANs-based Inference and Gradient Inversion/Suppression expanded considerably, with a noticeable peak in 2024. The emergence of Unintentional data leakage as a research focus in recent years adds to the diversifying landscape. The year 2024 stands out with the most comprehensive coverage across various attack categories, indicating an intensified focus on privacy protection research. This progression highlights the dynamic nature of privacy threats and showcases the academic community's proactive stance in addressing evolving challenges in data privacy safeguarding.

4.2. Privacy Defenses

Simultaneously, as the range and intricacy of attacks and threats on FL grow, novel defenses are also emerging to counteract their harmful impacts [17]. The following sections explore the most relevant defenses on privacy in FL by defining their concept, advantages, disadvantages, and the attacks each type of defense can defend against, proposed in the most recent and pertinent literature.

4.2.1. Zero-knowledge Proof

Zero-knowledge proof-based FL (ZKP-FL) scheme leverages zero-knowledge proof for both the computation of local data and the aggregation of local model parameters, aiming to verify the computation process without requiring the plaintext of the local data. Xing et al. [122] provided that on a blockchain, ZKP-FL allows clients in an FL system to prove to the central server that they have computed the correct model updates without revealing their underlying data. It helps to protect against attacks that aim to infer sensitive information from the federated model or the FL process. The FLAG framework [123] utilizes ZKP-FL techniques to provide secure computation without revealing sensitive information. It protects against data leakage, inference attacks, and unauthorized access to sensitive information. It provides a lightweight and efficient framework for secure aggregation in FL. However, implementing ZKP-FL protocols requires additional computational resources, which may introduce complexity and overhead due to the need for safe communication and encryption.

4.2.2. Oblivious Transfer

Oblivious Transfer (OT) in FL refers to a cryptographic protocol that allows a client to obtain one out of multiple potential values from a server without revealing the chosen value to the server. OT ensures privacy and confidentiality in FL by enabling clients to securely access and retrieve information from the server without compromising sensitive data. Rathee et al. [124] introduced ELSA (Ensemble Learning with Semi-honest Aggregators) as a defense mechanism in FL to protect the privacy of individual gradients during aggregation. ELSA employs l2-norm bounding to defend against boosted gradients from malicious clients. It consists of multiple layers that incorporate semi-honest and adversarial privacy defenses. ELSA's l2-norm protection is relatively simple compared to other securities, making it suitable for working over secret shares. ELSA cannot guarantee fairness, as it cannot distinguish between a malicious server and a malicious client, potentially leading to some honest clients' inputs not being used in the computation. ELSA's privacy defenses aim to protect the privacy of individual gradients during aggregation, limiting information leakage from the global aggregate.

4.2.3. Homomorphic Encryption (HE)

HE is a cryptographic technique that enables computations to be performed on encrypted data, producing an encrypted result that, when decrypted, matches the result of the same operations performed on the (original) plaintext. Singh et al. [125] discuss using HE to protect crucial data in the healthcare system, allowing computations to use encrypted data without decrypting it, preserving privacy. It enables secure data sharing and analysis in FL without revealing sensitive information to the central server or other clients. Nevertheless, it is a complex technology that requires specialized knowledge and infrastructure for implementation. The solution protects against unauthorized access, data leakage, and inference attacks. HE is a critical component of the SoK [126] defense strategy in FL. It offers robust privacy guarantees by ensuring data remains encrypted throughout the computation process. However, it can introduce notable drawbacks, including computational overhead and increased communication costs due to the complexity of processing encrypted data. It serves as a robust defense against various threats, including eavesdropping and data inference attacks, ultimately enhancing the privacy of FL systems.

4.2.4. Secret Sharing

Secret sharing (SS) is used in FL to distribute sensitive information, such as model parameters, among multiple clients. It involves dividing the secret into shares and distributing them to different clients, ensuring that no single client can access the complete secret. tMK-CKKS [127] with secret sharing provides information-theoretic security, making it impervious to collusion attacks by up to *t-1* clients working in concert with the server. Even when many clients join forces, they cannot deduce details about the master's secret. It involves the distribution of the master public key among all clients for encryption purposes. Furthermore, individual secret keys for each client are generated using a linear secret-sharing scheme. Notably, the decryption of aggregated ciphertexts necessitates the cooperation of only a specific threshold value, *t* clients. This careful balance of secret sharing and threshold requirements enhances the FL system's privacy.

4.2.5. MPC

MPC allows multiple clients to collaboratively compute a function on their private inputs without disclosing those inputs to one another. Bangalore et al. [123] proposed FLAG that scales to 1000s of clients, requires only a constant number of rounds, outperforms prior work in computational cost, and has competitive communication cost. However, it may introduce computational overhead due to the need for secure protocols and encryption. It helps defend against attacks that aim to compromise the privacy of user-held data during the aggregation process in FL. Mansouri et al. [126] proposed SoK, a defense mechanism in FL that focuses on the MPC of data from multiple sources without revealing individual inputs. It provides privacy protection and prevents adversaries from inferring sensitive information. SOK defense offers advantages in FL, like ensuring that individual data contributions remain confidential, making it difficult for attackers to carry out these attacks by obfuscating individual data contributions. Some downsides may include additional computational overhead and communication costs, specialized cryptographic knowledge, and careful design to ensure efficiency and scalability. SoK was specially designed to prevent membership inference and data reconstruction attacks.

4.2.6. DP

Nagy et al. [128] proposed a privacy-preserving FL framework for natural language processing incorporating local differential privacy (LDP) as a robust defense mechanism. LDP safeguards the privacy of individual data contributions, introducing noise to the model updates before sharing it with the server. The critical advantage of LDP is that it is exceedingly challenging for potential attackers to infer sensitive information about individual data contributors. However, introducing noise can impact the accuracy of the trained ML models, necessitating a careful balance between privacy preservation and model utility. LDP is a defense against attacks to uncover sensitive information about individual data contributors. Dynamic differential privacy (DDP), proposed by Guo et al. [119], involves dynamically adjusting the privacy budget and noise scale during model training, allowing for higher-quality models with a fixed privacy budget. It helps to get higher quality models and real-time privacy tracking, preventing the privacy budget from being exceeded, which could lead to the leakage of sensitive information. One drawback of this method is that it requires careful adjustment and injection of noise in each iteration, which adds complexity to the FL process. It is particularly effective at defending against eavesdropping attacks.

4.2.7. Gradient Clipping

Gradient-based defenses in FL involve modifying the gradients during training to protect against adversarial attacks. These defenses aim to make the model more robust by perturbing the gradients or adding noise. Chen et al. [111] proposed FedDef as an optimization-based input perturbation defense in FL that aims to preserve privacy and FL model performance by transforming private data into pseudo data that is dissimilar to the original data while maintaining similar gradients. Users download the global model from the server and use FedDef during local training to transform their data and gradients. However, the computational and memory overhead of FedDef needs to be considered, compared with HE in terms of performance. FedDef defends against reconstruction attacks, such as inversion attacks, extraction attacks, and attacks that manipulate reconstructed data. Gradient clipping restricts the size of updates from individual clients, ensuring that large updates from skewed or non-IID data do not disproportionately impact the model performance.

Li et al. [129] study *gradient clipping* and *sparsification* as defense mechanisms in FL. *Gradient clipping* involves setting a threshold for the magnitude of gradients during training and scaling them down if they exceed the threshold. This technique helps mitigate privacy leakage risks by controlling gradient magnitudes, making it harder to infer sensitive information from them. However, striking the right balance between privacy and model performance is crucial, as overly stringent thresholds can hinder learning. On the other hand, *gradient sparsification* enhances privacy by selectively transmitting only a subset of gradients that exceed a specified threshold. The latter reduces the information shared with the central server, minimizing the risk of privacy breaches. While it offers strong privacy protection, it may introduce computational overhead and affect the learning process by discarding some information.

4.2.8. Personalized Solutions

The references reviewed offer tailor-made solutions created to improve the security of FL environments. The most relevant are mentioned below.

CrowdFL. CrowdFL [130, 131] is an innovative approach that combines mobile crowdsensing (MCS) with FL to address privacy concerns while harnessing the computational power of clients. In this system, participants can perform local data processing using the FL framework, ensuring that sensitive sensing data remains on their clients. Only encrypted training models are uploaded to the server, preserving clients' privacy. CrowdFL offers scalability by leveraging MCS's large-scale data collection capabilities and reduces deployment costs by eliminating the need for extensive centralized infrastructure. This integration of FL into MCS enhances privacy and makes it a cost-effective and scalable solution for privacy-preserving mobile crowdsensing applications.

Soteria. Soteria [132, 111] is a defense mechanism proposed against model inversion attacks in FL. The defense focuses on perturbing data representation to severely degrade the quality of reconstructed data while maintaining FL performance. It aims to improve the privacy of FL systems by addressing data representation leakage from gradients, which has been identified as the essential cause of privacy leakage. After applying the protection, it provides a certified robustness guarantee to FL and a convergence guarantee to FedAvg. The privacy of the FL system is significantly improved with the implementation of Soteria defense. Soteria is designed to defend against model inversion attacks in FL, specifically the deep leakage from gradients (DLG) and gradient stalking (GS) attacks. These attacks aim to infer private data by exploiting the vulnerability of FL to inference attacks. Soteria ensures that the perturbed representations remain similar to the true representations for effective learning but degrade the quality of reconstructed data, mitigating the exacerbated privacy risks caused by non-IID data distributions.

FLTrust. [133, 114]: FLTrust is a defense mechanism designed to enhance the privacy of FL, aiming to detect and mitigate malicious clients in the FL process by evaluating their trustworthiness based on their behavior and contributions to the model training process. FLTrust utilizes trust scores to assess the reliability of clients and make informed decisions regarding their inclusion in the FL system. It relies on trust scores, which may not always accurately reflect the true intentions of clients. False positives or false negatives in trust evaluation can impact the fairness and effectiveness of the defense mechanism. FLTrust defends against attacks involving malicious clients in FL, such as Byzantine poisoning adversarial attacks, local models, and poisoning attacks in Byzantine-robust FL. This method addresses non-IID data issues (see Section 2.3) using a trusted server-side dataset to evaluate and assign trust scores to client updates, ensuring that malicious or biased updates are down-weighted.

RoFL. [134, 135]: Robustness of secure FL (RoFL) is an FL system that incorporates constraints on clients' updates to mitigate severe attacks. It extends secure aggregation with privacy-preserving input validation. RoFL efficiently includes conditions such as norm bounds on clients' updates and provides secure FL protocols in the single-server setting. RoFL can enforce restrictions such as L_2 and L_∞ bounds on high-dimensional encrypted model updates. RoFL achieves practicality even at a large scale. However, it incurs considerable overhead in terms of computational resources and, notably, bandwidth. It also uses a fairness-aware optimization approach to ensure balanced contributions from clients, improving overall model performance on non-IID datasets.

Prio. [136, 124]: It primarily focuses on preserving privacy while collecting essential aggregate statistics. This system leverages specialized zero-knowledge proofs, known as SNIPs, to enforce diverse defenses against malformed gradients while ensuring the confidentiality of clients' data against the influence of at most one malicious server. It not only safeguards sensitive information but also provides the flexibility to implement various defenses against irregular gradients, enhancing the overall security of the FL process. However, it's essential to consider potential drawbacks, such as the potential for computational overhead and increased communication costs due to using zero-knowledge proofs. In essence, Prio defense plays a crucial role in defending FL against attacks that manipulate gradients and compromise the integrity of aggregated statistics, all while preserving privacy.

	Attacks/Threats							
Defenses	**Gradient Inversion/suppression**	**Reconstruction through inference**	**Membership inference**	**Canary Gradient**	**Model Inconsistency**	**GANs-based inference**	**Eavesdropping**	**Unintentional data leakage**
Zero-knowledge proof					✓			
Oblivious Transfer		✓						
Homomorphic encryption	✓							✓
Secret sharing	✓							
MPC	✓	✓	✓					
Differential privacy	✓	✓	✓	✓				✓
Gradient clipping					✓			
Additive Noise	✓		✓	✓				
Personalized solutions	✓		✓		✓			

Table 8: Relationship of defense mechanisms and attacks for private FL

Table 8 highlights the intricate interplay between privacy threats and defense mechanisms in FL, revealing critical gaps and strengths in existing approaches. Notably, DP and MPC emerge as versatile solutions, countering a broad spectrum of attacks, including gradient inversion, membership inference, and unintentional data leakage. However, their effectiveness often comes at the cost of computational overhead and accuracy trade-offs, limiting real-world adoption. HE and secret sharing offer robust protection against gradient-based attacks but are less effective against inference threats like model inconsistency or GAN-based inference. Personalized solutions provide targeted defenses but lack the generalizability required for large-scale deployments. Moreover, attacks such as model inconsistency and GAN-based inference remain under-defended, signaling a need for novel countermeasures. These observations suggest that while FL privacy defenses are evolving, a holistic approach that balances security, efficiency, and adaptability remains a key challenge.

Lessons learned: The analysis of privacy threats and defenses in FL highlights a dynamic and evolving landscape. Attacks such as gradient inversion, membership inference, and reconstruction-based techniques continue to pose significant risks, exposing systemic weaknesses in current gradient-sharing protocols and the lack of standardized privacy auditing frameworks.. The rise of sophisticated attack strategies, including adversarial GANs and model inconsistency exploits, emphasizes the need for stronger, adaptive defenses , particularly those capable of operating under heterogeneous client behaviors and long training cycles. On the defense side, privacy-preserving techniques like DP, HE, and MPC show promise but often introduce computational overhead and may degrade model utility. Despite these limitations, hybrid protocols (e.g., DP-MPC) offer promising trade-offs, yet their scalability and deployment in real-world FL settings remain largely under-explored. Another challenge is the fragmented evaluation of defense strategies; indeed, most studies lack unified benchmarks or attack coverage, making it difficult to assess robustness across multiple threat vectors. Personalized solutions, such as FLTrust and Soteria, offer targeted protection against specific attack vectors but may require careful tuning to balance security and efficiency. Lastly, while many efforts focus on protecting server-side aggregation or model parameters, client-side privacy preservation (e.g., during local training or device compromise) remains insufficiently addressed, creating opportunities for attack vectors beyond the current scope of most defenses.

4.3. Comparative Analysis of Defenses for Private FL

In this subsection, we examine quantitative and experimental research to assess how well previous privacy defenses perform under particular attack scenarios [137, 138, 139, 140]. DP is effective against inference attacks by adding noise to gradients, but it often reduces accuracy, especially for underrepresented classes; newer methods like DP-MPC improve efficiency significantly (16-182x faster) while maintaining privacy guarantees. HE ensures strong confidentiality by enabling computations on encrypted data, achieving high accuracy (e.g., 99.95% on MNIST) but at a significant computational cost. MPC, particularly when combined with DP, enhances communication efficiency (56-794x) and speed, making it suitable for both privacy and efficiency scenarios. Gradient Clipping stabilizes training and reduces the risks of exploding gradients, but can slightly degrade performance if thresholds are too restrictive. ZKPs provide strong privacy and verifiability in trustless environments, such as blockchain-based FL, but can be computationally expensive. DP-MPC is recommended for applications prioritizing privacy without excessive overhead due to its balance of efficiency and privacy. Despite its cost, HE is ideal for accuracy-critical tasks like medical imaging, while gradient clipping with DP is effective for resource-constrained scenarios. In decentralized systems requiring trustless operations, ZKPs are valuable but should be used selectively due to their computational demands.

FL inherently involves trade-offs between privacy, security, and model performance. Strengthening privacy mechanisms often reduces model utility, while enhancing security may increase computational overhead [141, 142]. Therefore, selecting appropriate techniques depends on the specific application needs—whether prioritizing efficiency, accuracy, or privacy. Understanding these trade-offs is essential for designing robust and practical FL systems [143].

5. FL Frameworks

FL has witnessed the emergence of several frameworks designed to facilitate its application and address various aspects of privacy and security. Table 9 compares multiple features evaluated for the mentioned FL frameworks. The comparison involves the privacy and security support, attack simulation capacity, FL implemented types, and documentation and tutorials provided for the users. This comparative analysis enables us to provide insights into the strengths and weaknesses of each FL framework in terms of privacy, security, functionality, and user-friendliness, aiding researchers and practitioners in selecting the most suitable framework for their specific needs. The following paragraphs overview the most relevant FL frameworks, highlighting their characteristics, advantages, and limitations. At the end of the section, we also provide some lessons learned from analyzing the frameworks.

CRYPTEN [144]: CrypTen is a privacy-preserving ML framework implemented in Python and compatible with both Linux and Windows. Built on PyTorch, it provides MPC primitives, enabling collaborative computations on private data without exposing sensitive information. Its API closely resembles PyTorch, offering tensor computations, automatic differentiation, and modular neural networks, which simplify the integration of secure MPC techniques into ML workflows. CrypTen supports horizontal FL but lacks vertical and split FL capabilities. While it excels in secure aggregation and secret sharing, it does not implement DP or advanced attack simulations. The framework is open-source, well-documented, and user-friendly, making it accessible to ML practitioners. However, its reliance on an honest-but-curious threat model and limited support for advanced privacy mechanisms may restrict its application in certain adversarial scenarios.

FATE [145]: FATE is a flexible FL framework compatible with Linux and Windows, supporting popular programming languages like Python and Java. It offers comprehensive, secure computation protocols and diverse ML algorithms, including HE and MPC. FATE supports horizontal and vertical FL but lacks split FL capabilities. Its modular design provides end-to-end usability with pre-built components and user-friendly visualization tools, simplifying the implementation of privacy-preserving techniques. Additionally, FATE includes robust documentation, case studies, and tutorials to guide users. However, it does not implement DP or advanced attack simulations.

FEDML [146, 147, 148]: FedML, also referred to as TensorOpera AI, is a versatile and robust FL platform compatible with Linux, macOS, and Windows, developed in Python. It supports three distinct computing paradigms: on-device training, distributed computing, and single-machine simulation, making it adaptable to various FL scenarios. FedML offers a flexible API design with comprehensive baseline implementations for optimizers, models, and datasets. Security and privacy are addressed through the FEDML-HE module, which employs HE techniques. Additionally, its FEDMLSecurity component includes FedMLAttacker for simulating all type of attacks and FedMLDefender for testing defensive strategies. It does not implement advanced privacy mechanisms such as Zero-knowledge proof or MPC. Despite these limitations, its extensive documentation and strong attack simulation features make it a recommended tool for research and practical applications in FL.

FEDSCALE [149]: FedScale is an FL benchmarking suite compatible with Linux, macOS, and Windows, implemented in Python. It provides scalable runtime and realistic datasets that support diverse FL tasks, such as image classification, object detection, and language modeling. Its high-level APIs simplify the implementation, deployment, and evaluation of FL algorithms, enabling researchers to benchmark FL at scale with minimal effort. FedScale employs DP techniques to enhance security and privacy but lacks support for other mechanisms. It supports horizontal FL but does not implement vertical or split FL. While its documentation is somewhat limited, it includes essential resources for experimentation.

FL AND DP [150]: The Federated Learning (FL) and Differential Privacy (DP) framework is cross-platform, supporting Linux, Windows, and macOS, and is implemented in Python, Java, and C++. It emphasizes data privacy by integrating DP and holomorphic encryption techniques to quantify and mitigate privacy loss during distributed learning. The framework excels in ensuring privacy-preserving communication but lacks advanced security features. It supports horizontal and vertical FL but does not implement split FL. While the framework provides detailed documentation to guide users, its lack of a unified vision and a well-defined methodological workflow may limit its usability and effectiveness.

FLOWER [151, 152]: Flower is an open-source FL framework compatible with Linux, macOS, and Windows, implemented in Python. It is designed for large-scale FL experiments, supporting up to 15 million clients using only a pair of high-end GPUs, showcasing its scalability and efficiency. Flower excels in handling heterogeneous FL cross-device scenarios, making it suitable for diverse real-world applications. The framework prioritizes privacy by implementing various secure aggregation protocols, ensuring the server cannot inspect individual client models. However, it lacks support for advanced privacy techniques like FoolsGold or Geomed and does not include attack simulation features such as data poisoning or backdoor attacks. Flower supports horizontal and vertical FL but does not implement split FL. Detailed documentation and an active community enhance its usability by providing comprehensive guidance on installation, usage, and API references. Despite its limitations in attack simulations, Flower's scalability and flexibility make it a recommended tool for FL research and experimentation.

FLUTE [153]: The FLUTE (Federated Learning Under True Environment) framework is an open-source tool compatible with Linux, macOS, and Windows, implemented in Python (version 3.6 or higher). It is designed for high-performance

Table 9: FL frameworks features comparison (✔: Complete, ◆: Under development/incomplete, ✘: Unknown/Not implemented)

	CRYPTEN	FATE	FEDML	FEDSCALE	FL AND DP	FLOWER	FLUTE	NVFLARE	OPENFL	PaddleFL	PYSYFT	TFF	XFL
Privacy and security support	◆	◆	✔	◆	✔	✔	◆	◆	◆	◆	✔	◆	◆
❄ Differential privacy	✘	✘	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
❄ FoolsGold	✘	✘	✔	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘
❄ GeoMed	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘
❄ Homomorphic encryption	✘	✔	✔	✘	✔	✔	✘	✔	✘	✘	✔	◆	✔
❄ Krum	✘	✘	✔	✘	✘	✔	✘	✘	✘	✘	✘	✘	✘
❄ Multi-Krum	✘	✘	✔	✘	✘	✔	✘	✘	✘	✘	✘	✘	✘
❄ Norm difference clipping	✘	✘	✔	✘	✘	✔	✘	✘	✘	✘	✘	✔	✘
❄ RFA (geometric median)	✘	✘	✔	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘
❄ Secret Sharing	✔	✘	✘	✘	✔	✘	✘	✘	✘	✘	✔	✘	✘
❄ Secure Aggregation	✘	✘	✘	✘	✘	✔	✘	✘	✘	✔	◆	✘	✘
❄ MPC	✔	✔	✘	✘	✔	✘	✘	✘	✘	✔	✔	✔	✔
❄ Trimmed Mean	✘	✘	✘	✘	✘	✔	✘	✘	✘	✘	✘	✘	✘
❄ Zero-knowledge proof	✘	✘	✘	✘	✔	✘	✘	✘	✘	✘	✘	✘	✘
Attacks simulation	✘	✘	✔	✘	◆	✘	✘	✘	✘	✘	✘	◆	✘
❄ Data Poisoning	✘	✘	✔	✘	✔	✘	✘	✘	✘	✘	✘	✔	✘
❄ Model poisoning	✘	✘	✔	✘	✔	✘	✘	✘	✘	✘	✘	✘	✘
❄ Byzantine	✘	✘	✔	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘
❄ Label flipping	✘	✘	✔	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘
❄ Backdoor	✘	✘	✔	✘	✘	✘	✘	✘	✘	✘	✘	✔	✘
FL types	◆	✔	✔	◆	✔	✔	◆	✔	◆	✔	✔	◆	✔
❄ Horizontal FL	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
❄ Vertical	✘	✔	✔	✘	✔	✔	✘	✔	◆	✔	✔	✘	✔
❄ Split FL	✘	✘	✔	✘	✘	✘	✘	✔	✘	✘	◆	✘	✘
Open source	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✘
Documentation and tutorials	✔	✔	✔	✔	✔	✔	◆	✔	◆	◆	✔	✔	◆

FL research, enabling rapid prototyping and large-scale simulations of novel FL algorithms. FLUTE supports local and global DP methods, emphasizing data security and preservation. However, it lacks other advanced privacy mechanisms and does not include attack simulation features. While it supports horizontal FL, it does not implement vertical or split FL. The framework's documentation is available but incomplete, with no tutorials to assist new users.

NVFLARE [154]: The NVIDIA FLARE (a.k.a NVFLARE) framework is a tool compatible with Linux and Windows. FLARE primarily supports Python as the programming language for developing FL workflows. Its exceptional features encompass state-of-the-art FL algorithms and approaches, allowing researchers to apply their data science workflows seamlessly using popular training libraries such as PyTorch, TensorFlow, XGBoost, or NumPy. Its lightweight, flexible, and scalable nature distinguishes the framework, rendering it suitable for real-world FL scenarios. It ensures secure and privacy-preserving multiparty collaboration by implementing HE or DP techniques. Comprehensive documentation provided by NVIDIA FLARE aids users in harnessing the framework's potential for both research and practical applications.

OPENFL [155]: OpenFL is a Python-based FL framework compatible with Linux, macOS, and Windows. It supports developing and training ML and DL algorithms using TensorFlow, PyTorch, and other ML/DL frameworks, enhancing its adaptability for diverse use cases. As an open-source platform, OpenFL offers flexibility and customization options for researchers and developers. However, it lacks support for advanced privacy-preserving techniques such as secure aggregation or HE, limiting its security features. It supports horizontal FL but has an incomplete implementation of vertical FL. While documentation is available on its official website, it is incomplete and lacks specialized tutorials to guide new users effectively.

PaddleFL [156]: PaddleFL is an open-source framework built on PaddlePaddle, supporting horizontal and vertical FL with privacy-preserving techniques like DP and Secure Aggregation. It is compatible with multiple platforms and languages but lacks split FL and attack simulation support. Despite its scalability, PaddleFL's usability is limited by sparse documentation and a predominantly Chinese-speaking community.

PYSYFT [157]: PySyft is an FL library compatible with Linux, macOS, and Windows, primarily implemented in Python and extending popular DL frameworks like PyTorch. Its mission is to democratize privacy-preserving techniques in ML, making them accessible to researchers and data scientists. PySyft supports privacy-enhancing methods such as MPC and DP. However, it lacks advanced security protocols and does not provide attack simulation capabilities. PySyft implements horizontal vertical and an uncompleted version of split FL. Its comprehensive documentation includes detailed procedures, implementation guides, and example workflows, empowering users to effectively utilize the framework for privacy-focused FL projects.

TFF [158, 159]: TensorFlow Federated (TFF) is a Python-based framework that integrates with TensorFlow, supporting horizontal FL and incorporating privacy mechanisms like MPC and DP. It includes a simulation environment for testing attacks but lacks support for vertical and split FL and advanced privacy techniques like secure aggregation. Comprehensive documentation aids usability, though its limitations may restrict its use in highly adversarial settings.

XFL [160]: XFL is a versatile framework compatible with multiple platforms and languages, offering a user-friendly interface and pre-built algorithms for horizontal and vertical FL. It supports privacy-preserving techniques like HE, DP, and MPC but lacks support for split FL and attack simulations. While XFL simplifies deployment via Docker, incomplete documentation somewhat limits its usability.

Lessons learned: Given the previous details of each FL framework, FEDML emerges as the most complete solution since it incorporates many security and privacy methods, all the most common FL types, and vast documentation and tutorials. In addition, it highlights that since it is the only framework that includes a comprehensive suite of attack simulations. It is perfect for quickly testing new defenses proposed by security and privacy FL researchers. Nevertheless, FLOWER and FL AND DP are also relevant frameworks for researchers to consider due to the implementation of various security and privacy protocols. However, they lack support for adversarial testing modules and fine-grained control over threat modeling, which limits their effectiveness for evaluating defenses under dynamic or adaptive adversaries. A notable gap is that none of the surveyed frameworks implement geometric median-based defenses like GeoMed, despite their empirical robustness against poisoning. In addition to missing implementations of advanced defenses (e.g., FLAME, Pruning-based defenses), we also find limited support for simulating realistic deployment conditions such as heterogeneous participation, client drift, or colluding Sybil attacks—factors increasingly relevant in real-world FL. We also observe that support for vertical and split FL is incomplete across most frameworks, which may hinder applications in finance, healthcare, or IoT, where data distributions are often partitioned by feature. Finally, the lack of standardized interfaces for measuring privacy-utility trade-offs (e.g., formal accounting of DP budgets vs. accuracy degradation) further limits reproducibility. Future framework development should prioritize modular threat modeling, integrated attack-defense testbeds, and support for adaptive, personalized, and hybrid defense strategies.

6. Main Applications

We explore how various domains are employing this transformative FL technology. This chapter delves into real-world applications where FL plays a pivotal role, providing a deeper understanding of its practical significance. By highlighting relevant use cases, ranging from healthcare and finance to intrusion detection, we unveil the diverse scenarios where FL is making a substantial impact. Additionally, at the end of the section, we provide a paragraph of lessons learned based on analyzing the main applications of secure and private FL.

Based on our literature review, we obtained the fields of applications that use FL in a private and secure context.Fig. 7 depicts the participation of each field over the papers analyzed. The top three fields applying secure and private FL are text prediction, healthcare, and the financial sector. The latter have been the most employed fields for a long time. Nevertheless, it is relevant to highlight that the intrusion detection systems field has gained strong participation among researchers in recent years. The following subsections describe how FL was employed for each field, emphasizing some challenges, attacks, and defenses utilized.

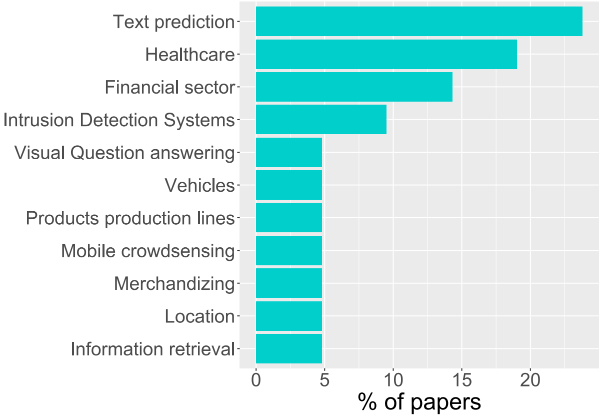

Figure 7: Main applications using privacy and secure FL

Text prediction. Privacy and security are relevant in FL for text prediction because sensitive user data, such as personal messages or search queries, is processed locally on devices. Without strong privacy measures and secure communication (e.g., encryption), there is a high risk of exposing personal information, which could lead to breaches of user confidentiality or misuse of private data by malicious actors. Advancements in FL for text prediction emphasize privacy and security through techniques like DP and local DP [161, 128]. However, these methods often struggle to balance privacy and model performance, as stringent privacy measures can reduce prediction accuracy. Given the sensitive nature of textual data, ensuring security and privacy is vital for maintaining user trust and compliance with data protection regulations. Vulnerabilities in this field can stem from the decentralized nature of the data, with common attacks including poisoning attacks [162, 128], GAN-based inference [163, 135], and gradient inversion/suppression [164, 165]. To address these risks, defenses such as DP, employed by Qi et al. [166], can secure text data while preserving model utility.

Healthcare. FL helps with privacy and security in healthcare by allowing institutions (i.e., hospitals) to train models collaboratively without sharing sensitive patient data. Techniques such as DP and HE safeguard patient information during model updates, addressing risks like data leakage and unauthorized access [167, 168]. However, they can also introduce computational overhead and potentially compromise model accuracy. This domain encounters significant security and privacy challenges due to the sensitive nature of medical data. Common attacks include poisoning attacks [169, 170, 171], where adversaries inject malicious data to undermine model integrity [114], and gradient inversion/suppression [172, 106], which attempts to recover private medical information from shared gradients [108]. Membership inference attacks pose additional risks by revealing whether a specific patient was used in model training [173].

To mitigate these threats, cryptographic techniques like HE and MPC ensure data privacy while allowing computations on encrypted data [174, 125]. DP also plays a crucial role in limiting the risk of sensitive information being memorized or inferred from model updates. Moreover, robust aggregation operators defend against poisoning and other adversarial manipulations, ensuring the integrity of the global model. By combining cryptographic techniques with privacy-preserving methods, healthcare FL systems can effectively protect against multifaceted attacks while maintaining accuracy and compliance with healthcare regulations, as highlighted by Singh et al. [108].

Financial sector. FL enhances privacy and security in the financial sector by enabling institutions (i.e., banks and Fintech enterprises) to collaborate on fraud detection and risk management without sharing sensitive customer data. Ensuring security and privacy is critical in finance due to the sensitive nature of financial data and regulatory requirements, fostering customer trust and compliance with regulations like GDPR, which enable safer financial services. FL is particularly valuable for fraud detection and risk management but is susceptible to various attacks, notably GAN-based poisoning attacks [56, 175], which can degrade model performance and compromise privacy by manipulating training data, as highlighted by Qiao et al. [78]. Moreover, MPC ensures that no single client gains access to sensitive financial data during joint model training.

Intrusion detection systems. FL enhances privacy and security in intrusion detection systems by facilitating distributed model training without sharing raw logs. Techniques such as MPC and optimization-based input perturbation [111] guard against inference and poisoning attacks [176]. However, challenges like deployment complexity and potential impacts on model accuracy persist. Security and privacy are vital in this domain, as adequate intrusion detection safeguards sensitive environments from unauthorized access, ensuring compliance with security standards and fostering user trust in system reliability. In IoT networks, FL encounters significant security challenges, including label-flipping attacks [177], where malicious clients manipulate labels to mislead the global model, as Yang et al.[111] noted.

Visual question-answering. In visual question-answering (VQA) models using FL, several security and privacy concerns arise due to the complexity of the task and the diverse data types involved. One prominent attack in vertical FL VQA is the ADI, where adversaries manipulate input data, such as images, to dominate the learning process and reduce the contributions of other clients, as explored by Pang et al. [78]. GAN-based inference is another risk, where attackers attempt to reconstruct private information, such as images or questions, from model updates. Anomaly detection can be employed to defend against these threats by identifying and excluding manipulated data or adversarial inputs before they influence the model. DP can obscure sensitive images or question details to protect clients' local data [178]. Additionally, robust aggregation operators ensure that adversarial contributions, like poisoned data, do not degrade the overall model performance. Vertical FL VQA systems can leverage these defenses to maintain privacy and security, enabling collaborative model training without exposing sensitive visual or textual information.

Vehicles. Under this field, using FL with secure and private defenses is critical because connected cars generate sensitive data about drivers' locations, routes, driving behaviors, and vehicle diagnostics, and protecting this information prevents unauthorized tracking, behavior profiling, and potential safety vulnerabilities. In the vehicle field, ADI attacks, such as random or bounded mutation, can manipulate vehicle data and degrade model performance, as Pang et al. [78] reported. Additionally, model inconsistency may arise from adversarial updates across clients. To defend against these, robust aggregation operators reduce the impact of malicious updates, while DP and additive noise protect sensitive vehicle data from being inferred through model updates. These defenses ensure secure and accurate FL models in vehicle-related tasks.

Products production line. In this field, privacy and security in FL are functional because manufacturers can collaboratively improve their production models and optimize processes while securely keeping sensitive proprietary data (like manufacturing parameters, quality control metrics, and production recipes) within their facilities, preventing industrial espionage and maintaining competitive advantages. FL faces label-flipping and backdoor attacks in product production lines, which can compromise model accuracy and reliability in assembly processes [179]. To counter these threats, robust aggregation operators filter out harmful contributions from adversaries, while anomaly detection identifies and excludes suspicious data. Additionally, DP protects sensitive production metrics during model training [180].

Mobile crowd-sensing. Privacy and security in mobile crowd-sensing FL are crucial to protect sensitive location and behavioral data, preventing unauthorized tracking and identity breaches while enabling valuable insights for urban planning and services. FL is susceptible to eavesdropping and membership inference attacks in mobile crowd-sensing, which can compromise client privacy. Additionally, poisoning attacks can manipulate model updates, degrading performance [130]. To defend against these threats, secure aggregation methods based on the threshold Paillier cryptosystem protect model confidentiality while DP obscures individual contributions. Robust aggregation operators also help mitigate the impact of adversarial updates.

Merchandising. Privacy and security in FL regarding this field are relevant because retailers handle sensitive customer purchasing patterns, inventory strategies, and pricing data across multiple locations. Thus, protecting such information prevents competitors from accessing valuable business intelligence while fostering peer-to-peer modeling to optimize merchandising decisions and customer experience across store networks. In this field, FL faces threats like poisoning attacks, where malicious clients corrupt the global model by manipulating their local data, and membership inference attacks, which attempt to deduce the presence of specific data samples in the training dataset [114]. To defend against these attacks, robust aggregation operators can filter out adversarial updates, ensuring that only reliable contributions influence the global model. Additionally, employing DP techniques helps obscure individual shopping histories, protecting sensitive merchandising data from exposure [7].

Location. FL's privacy and security are paramount in location services as they protect users' sensitive movement patterns and visited places while leveraging multiparty learning to improve location-based services without exposing individual data. FL is vulnerable to location tracking attacks in the location field, where adversaries attempt to infer users' movements or patterns from shared model updates. Membership inference attacks can also expose sensitive information about individuals based on their location check-ins [114]. Implementing DP techniques can obscure individual check-in data to mitigate these threats, protecting user privacy while allowing practical model training. Additionally, robust aggregation operators can help filter out adversarial contributions, ensuring that only trustworthy data influences the global model [7].

Information retrieval. In information retrieval, FL's privacy measures protect users' sensitive search patterns and interests while permitting collaborative improvement of search systems without exposing personal data. In this field, FL faces challenges such as insufficient training data, where individual users may lack enough interactions to achieve high search effectiveness. The latter can be exploited through model inversion attacks, where adversaries infer sensitive user data from shared model parameters [55]. DP techniques can be employed to protect individual search interactions, ensuring user privacy while still allowing model training. Additionally, robust aggregation operators can help mitigate the effects of malicious updates, enhancing the reliability of the global model.

Lessons learned: The analysis of secure and private FL across main domains highlights its transformative potential and persistent challenges. The widespread adoption in fields like text prediction, healthcare, and finance underscores the necessity of FL for protecting sensitive user data while enabling collaborative model training. However, ensuring privacy and model performance remains a central challenge, as strict privacy mechanisms often introduce accuracy and computational efficiency trade-offs. The rise of FL in intrusion detection systems and mobile crowd-sensing indicates an increasing awareness of its role in security-sensitive environments, though these applications face threats like poisoning attacks and adversarial data manipulation. Across all domains, the decentralized nature of FL introduces vulnerabilities such as gradient inversion and membership inference attacks, emphasizing the need for robust defense mechanisms. Notably, emerging vehicles, manufacturing, and information retrieval applications demonstrate FL's adaptability yet reveal unique domain-specific risks, from adversarial attacks in autonomous driving to industrial espionage in production lines. Another key limitation is the lack of standardized, reproducible evaluation pipelines across domains, making it difficult to compare defense effectiveness or understand trade-offs across threat models. Furthermore, many application areas lack publicly available benchmarks that reflect realistic attack scenarios, hindering the development and validation of domain-adaptive security mechanisms. A key lesson is that while FL enhances data privacy, both its privacy and security largely depend on continuous advancements in cryptographic techniques, adversarial defenses, and efficient aggregation strategies tailored to each field's requirements.

7. Future Directions

While significant strides have been made in addressing FL's security and privacy challenges, several areas remain ripe for exploration and improvement. The complexity and evolving nature of FL environments necessitate ongoing research to refine existing techniques and develop novel solutions. This section outlines vital areas for future work, highlighting the need for advanced methods to enhance the robustness of FL systems against emerging threats. It emphasizes the importance of addressing limitations in current approaches and exploring innovative strategies that balance security, privacy, and efficiency.

7.1. Security Future Directions

Security in FL remains a significant challenge, especially in light of sophisticated poisoning and backdoor attacks. Future directions should focus on developing robust and adaptive security mechanisms that can detect and mitigate these threats while maintaining the integrity of the global model. The emphasis would be on improving the resilience of FL systems, enhancing verification processes, and developing scalable solutions that support high performance even in adversarial settings.

- **Enhanced Robustness Against Advanced Attacks:** As discussed in Section 3, FL environments face significant challenges from adversarial attacks, such as model poisoning and backdoor insertion, particularly in heterogeneous data settings [181].

  To counter model poisoning attacks, where compromised clients degrade global model performance, future work should explore adaptive aggregation techniques that dynamically adjust the contributions of client updates based on anomaly detection metrics. For example, methods like adaptive local aggregation [182] or sparsification-based defenses [183] could be extended to incorporate real-time monitoring of update trajectories [184]. Additionally, integrating client-side defenses like FL-WBC, which perturbs parameter spaces affected by attacks, could mitigate long-term attack impacts [185].

  For backdoor attacks, in which malicious clients insert triggers into models to induce targeted misclassifications, future defenses could leverage hybrid anomaly detection approaches combining statistical gradient analysis and cryptographic verification [186]. Techniques like ARIBA have shown promise in identifying distributional anomalies in model updates. Furthermore, incorporating multi-method adaptive aggregation algorithms (e.g., SAPAA-MMF) could enhance robustness by balancing contributions based on data quality and variance [187]. Exploring interdisciplinary methods inspired by biological immune systems could also provide novel insights for adaptive and self-healing mechanisms in FL systems.

- **Resilience to Emerging Threats in Dynamic Environments:** Deploying FL in dynamic settings such as autonomous vehicles and smart cities introduces unique vulnerabilities, including free-riding attacks, model extraction attacks, and jamming threats [188]. Addressing these challenges requires targeted strategies:

  Free-Riding Attacks: Free-riders exploit FL aggregation protocols by contributing no meaningful updates while benefiting from the global model [67]. Future work should explore advanced anomaly detection mechanisms such as high-dimensional clustering techniques (e.g., STD-DAGMM) to identify free-riders [70]. Integrating blockchain-based accountability frameworks could also enhance trust by recording client contributions transparently [189].

  Model Extraction Attacks: Malicious clients can reverse-engineer global models to steal intellectual property or compromise privacy [190]. Hybrid encryption techniques combining HE and MPC could be employed to counter this. Furthermore, gradient obfuscation methods that distort shared parameters without degrading model performance warrant investigation [191].

  emphCollaborative Jamming Attacks: Jamming attacks in FL-based 5G networks disrupt communication channels, degrading model performance. In 5G networks, jammers exploit public NR standards (e.g., PUCCH intra-slot hopping patterns and RACH protocols) [192] to disrupt synchronization signals with energy-efficient methods like reactive jamming, posing critical risks to public safety and military operations. Mitigation includes spread spectrum techniques (DSSS/FHSS) and ML-based detection (XGBoost ensembles achieving 99.72% accuracy). Concurrently, model extraction attacks—enabled via API query duplication (e.g., LLM ”leeching”) [193]—threaten proprietary models in finance and healthcare. Defenses like ModelGuard's information-theoretic perturbation maintain ¡3% utility loss while thwarting extraction. A promising direction involves implementing decentralized jamming detection frameworks using convolutional autoencoders for unsupervised anomaly detection and FedProx algorithms for supervised classification [194].

- **Fairness, Bias Mitigation, and Security Integration:** As FL models are increasingly deployed in sensitive applications like healthcare and finance, ensuring fairness under adversarial conditions remains a critical challenge. A particularly concerning threat is fairness attacks (poisoning), where attackers manipulate data or model updates to introduce or amplify bias [195]. These attacks disproportionately harm specific groups, such as racial minorities or underrepresented communities, making fairness a direct target of adversarial manipulation.

  Future research should focus on developing fairness-preserving aggregation methods that integrate anomaly detection with fairness constraints. For example, leveraging techniques like FairFed [196], which adaptively reweights client contributions based on fairness metrics, could counteract biased updates. Additionally, interdisciplinary approaches combining cryptographic tools with fairness-aware algorithms can enhance defenses against malicious clients [197]. Addressing indirect bias–where even non-malicious clients contribute biased data unintentionally–requires advanced techniques like fairness-aware pruning or incorporating domain adaptation methods to balance performance across heterogeneous client distributions [198].

- **Scalable, Efficient, and Verifiable Secure Aggregation:** As FL scales to larger and more complex systems, secure aggregation techniques face significant challenges, particularly in mitigating attacks such as model poisoning and Sybil attacks. A critical technical challenge is designing aggregation protocols that balance computational efficiency with robust security guarantees. For instance, lightweight encryption mechanisms like homomorphic hash functions or single-mask symmetric encryption could reduce overhead while maintaining privacy and verifiability [199]. Additionally, dynamic masking strategies, which adapt to threat levels in real-time, could enhance resilience against predictable attack patterns [200].

  To counter Sybil's attacks effectively, interdisciplinary approaches integrating DP with anomaly detection methods show promise. For example, combining DP with graph-based anomaly detection could identify malicious clients based on their interaction patterns [201]. Furthermore, verifiable aggregation protocols such as LightVeriFL can ensure the integrity of updates by leveraging homomorphic commitment schemes for lightweight verification [202]. Future work should explore these approaches in scenarios with high user dropout rates to ensure robustness.

  Blockchain technology offers a promising avenue for tamper-resistant and auditable aggregation. However, its scalability remains a concern due to high computational costs. A potential solution is hybrid architectures that combine blockchain with adversarial training techniques or verifiable delay functions to balance security and efficiency [203]. Another plausible solution is using dedicated off-chain servers to handle validation and aggregation (e.g., Fantastyc's proof generation), reducing on-chain operations by 70% [204]. Moreover, reinforcement learning-based adaptive Proof-of-Work (PoW) dynamically adjusts mining difficulty in response to real-time miner capabilities and network conditions, reducing energy waste by up to 45% and lowering computational overhead for honest clients [205]. Research should focus on optimizing these solutions for decentralized FL settings where central servers are absent.

7.2. Privacy Future Directions

Privacy preservation is a critical aspect of FL, mainly when dealing with sensitive data distributed across multiple clients. The future of privacy-enhancing techniques will focus on improving the efficiency and scalability of existing methods, making them suitable for a wide range of applications, from edge devices to large-scale cross-silo FL environments. The goal is to ensure data privacy without compromising model performance or significantly increasing computational and communication overhead.

- **Privacy-Enhancing Techniques for Non-IID Data:** Non-IID data presents one of the biggest challenges in FL, particularly in safeguarding privacy while ensuring robust model performance (see Section 2.3). In non-IID scenarios, privacy-preserving techniques like HE, MPC, and DP face limitations due to data heterogeneity, which increases susceptibility to targeted inference attacks [206]. Attackers can exploit discrepancies in data distributions across clients to perform data reconstruction or membership inference attacks. To address these vulnerabilities, future research should focus on:

  Dynamic Privacy Mechanisms: Developing adaptive DP mechanisms that adjust privacy budgets based on client-specific data heterogeneity. For instance, privacy budgets could be dynamically allocated using metrics such as the Hellinger, Jensen-Shannon, or Earth mover's distances to quantify inter-client distribution disparities [32].

  Scalable Encryption Protocols: Optimizing HE and MPC for non-IID settings by reducing computational overhead through techniques like hybrid encryption schemes or gradient compression [207].

  Robust Aggregation Methods: Designing aggregation techniques that mitigate the influence of skewed client updates, such as similarity-weighted aggregation or clustering-based approaches [206].

- **Integration of Advanced Cryptographic Protocols:** As FL continues to scale, particularly in sensitive domains like IoT and healthcare, privacy remains vulnerable to specific attacks such as inference and canary gradient attacks. Future research must focus on integrating advanced cryptographic protocols that enhance privacy while minimizing performance costs. Such future work includes the following technical challenges:

  Inference Attacks: Adversaries reconstruct sensitive data from aggregated model updates. This requires efficient HE schemes that support secure aggregation without significant computational overhead [147].

  Canary Gradient Attacks: Attackers inject small perturbations into gradients or weight updates. Existing cryptographic methods struggle to detect such subtle manipulations [59].

  Key Management in HE: Current single-key HE schemes risk key leaks, necessitating multi-key or secret-sharing schemes for enhanced security [208].

  Therefore, some proposed solutions that can be explored in future research are: First, develop hybrid cryptographic frameworks combining HE with MPC to protect against both classical and quantum adversaries [147]. Second, implement adaptive gradient clipping techniques alongside DP to mitigate inference and canary attacks without degrading model accuracy [209]. Third, design decentralized key management systems using secret sharing or blockchain-based approaches to enhance security in HE implementations.

- **Enhanced Verification of Aggregated Models:** Ensuring the integrity of aggregated models while preserving privacy is critical, especially given the growing threat of GAN-based inference attacks. These attacks exploit GANs to infer sensitive information about training data in FL settings [210]. Future research must address specific challenges, such as reducing computational overhead and communication costs while maintaining robust privacy guarantees. Promising directions include:

  Lightweight Verifiable Aggregation Protocols: Techniques such as homomorphic hashing and bilinear aggregate signatures have shown potential for verifying aggregation results [211]. However, these methods often face scalability issues due to high-dimensional model gradients. Research should optimize these protocols by leveraging advanced cryptographic techniques like polynomial commitments or ProxyZKP frameworks [212].

  Combating GAN-Based Attacks: Defense mechanisms like Anti-GAN frameworks, which manipulate visual features to thwart GAN-based inference attacks, are promising [122]. Future work could explore integrating such frameworks with secure aggregation techniques to enhance privacy without compromising model accuracy.

  ZKPs for Privacy-Preserving Verification: ZKPs enable entities to prove the correctness of computations without revealing sensitive data [122]. While ZKPs hold great promise for FL, current implementations face scalability challenges.

  Optimizing ZKP Scalability: Techniques like zk-SNARKs and zk-STARKs provide efficient proof systems but require further optimization for large-scale FL applications. The ProxyZKP framework, which uses polynomial decomposition to reduce proof generation times, offers a viable path forward [212]. Another avenue is using Batch verification processes to verify multiple proofs simultaneously, cutting verification overhead by up to 70%, while recursive composition hierarchically aggregates proofs into compact representations, ideal for large-scale deployments [213]. For resource-constrained environments, collaborative zk-SNARKs distribute proof generation across parties, linearly reducing per-node complexity [214].

Quantitative Privacy-Performance Trade-off Models: While numerous studies have explored the qualitative trade-offs between privacy, security, and model performance in FL, a unified quantitative framework remains an open challenge. Existing research provides valuable insights into individual trade-offs, such as the impact of privacy budgets in DP on model utility, but lacks a standardized mathematical formulation that systematically captures these interdependencies. Future research should focus on developing mathematical models that integrate privacy loss, computational overhead, and model accuracy into a single framework. These models could incorporate utility functions that balance security guarantees with performance metrics, similar to approaches in economic game theory or optimization-based frameworks [143]. By addressing these directions, future research can bridge the gap between qualitative discussions and rigorous quantitative analysis, ensuring a more precise understanding of privacy-performance trade-offs in FL.

7.3. Joint Directions on Security–Privacy in FL

While this survey separately addresses challenges in security and privacy, real-world FL deployments often suffer from their combined vulnerabilities. A critical future direction lies in understanding how attacks on one axis may amplify risks on the other, and how certain defenses may inadvertently open new threat vectors. For instance, a poisoning attack can manipulate the model's sensitivity to benign client gradients, thereby increasing the effectiveness of gradient inversion techniques. Similarly, colluding Sybil clients can bias the global model toward a specific user's data distribution, enhancing the attacker's chances in subsequent membership inference. Conversely, privacy defenses like secure aggregation or heavy DP noise may hinder the detection of adversarial behavior, thereby weakening overall system security.

Future research should pursue co-designed mechanisms that bridge this gap:

- Integrated threat modeling that considers both privacy leakage and security compromise in unified scenarios.
- Privacy-aware robust aggregation techniques that maintain anomaly detection capabilities even under DP noise.
- Layer-wise defense strategies that protect sensitive layers with cryptographic tools while preserving transparency in others for anomaly auditing.
- Client-side collaborative monitoring, using lightweight trusted execution environments (TEEs) to audit gradients locally before encrypted aggregation.
- Benchmark frameworks that evaluate FL systems against compound attack scenarios rather than isolated vectors.

Based on the previous analysis, FL research must evolve from siloed views to holistic frameworks, ensuring that strengthening one defense front does not unintentionally weaken the other. Tackling this interplay remains a fundamental challenge and opportunity for building resilient and trustworthy federated systems.

An emerging dimension of this security–privacy interplay arises from the integration of FL with Generative AI (GenAI) systems, including large language models (LLMs). While these models offer powerful personalization and collaborative capabilities, they also amplify both axes of vulnerability. For example, GenAI systems are particularly prone to data memorization, making them susceptible to privacy leakage even under secure aggregation [215]. This problem becomes more pronounced in FL, where attackers may exploit intermediate gradients or personalized prompts to extract or reconstruct client data. This raises new privacy risks beyond what traditional FL defenses like secure aggregation or DP were designed to handle. Security threats also take new forms. For instance, poisoning attacks in generative models may bias completions toward specific ideologies or inject imperceptible toxic content. Detection and mitigation become more complex when the goal of an attack is to subtly influence output distributions rather than flip classification labels [216]. Additionally, verifying the integrity and alignment of decentralized GenAI systems becomes increasingly difficult without centralized auditing. Future research must thus explore new FL frameworks tailored for generative tasks. These may include federated instruction tuning pipelines with private prompt alignment, hybrid split-federated architectures to manage compute imbalance, and adaptive privacy controls that account for generative memorization risks, for example, considering *federated unlearning* notions [217]. Addressing these questions is essential for deploying GenAI responsibly in distributed environments, as well as for advancing the robustness and trustworthiness of FL systems more broadly.

8. Conclusion

This survey provides an in-depth analysis of the security and privacy challenges in FL. It reveals that despite FL's design to enhance data privacy, it is susceptible to various threats, such as data poisoning, model inversion, and backdoor attacks, underscoring the need for effective defense mechanisms. By categorizing these attacks and their impacts, we offer a structured understanding of FL systems' diverse threats. We also highlight the importance of balancing privacy, security, and model performance through techniques like cryptographic methods and DP. Recent research trends indicate a growing focus on addressing these issues, calling for scalable and adaptive solutions suitable for dynamic environments. Future research should develop innovative, energy-efficient solutions to address the identified challenges, paving the way for more secure and practical FL applications. Overall, this survey is a valuable resource for future work advancing secure, privacy-preserving collaborative learning systems.

9. Acknowledgments

Daniel Mauricio Jimenez G. was partially supported by PNRR351 TECHNOPOLE – NEXT GEN EU Roma Technopole – Digital Transition, FP2 – Energy transition and digital transition in urban regeneration and construction and Sapienza Ateneo Research grant "La disintermediazione della Pubblica Amministrazione: il ruolo della tecnologia blockchain e le sue implicazioni nei processi e nei ruoli della PA.". José Luis Hernández Ramos was supported by a Leonardo Grant 2023 to Researchers and Cultural Creators from the BBVA Foundation, a Consolidación Investigadora grant (CNS2023-145059) funded by MICIU/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR, the project ONOFRE-4 funded by MICIU/AEI/10.13039/501100011033 and FEDER, UE, and the project PCI2023-145989-2 (REMINDER) funded by MICIU/AEI/10.13039/501100011033 and the European Union. Aris Anagnostopoulos was supported by the ERC Advanced Grant 788893 AMDROMA, the EC H2020RIA project "SoBigData++" (871042), the PNRR MUR project PE0000013-FAIR, the PNRR MUR project IR0000013-SoBigData.it, and the MUR PRIN project 2022EKNE5K "Learning in Markets and Society.". Ioannis Chatzigiannakis and Yelizaveta Falkouskaya were supported by PE07-SERICS (Security and Rights in the Cyberspace) – European Union Next-Generation-EU-PE0000014 (Piano Nazionale di Ripresa e Resilienza – PNRR). Andrea Vitaletti was supported by PE11 - MICS (Made in Italy – Circular and Sustainable) – European Union Next-Generation-EU (Piano Nazionale di Ripresa e Resilienza – PNRR).

References

[1] T. R. MacNish, M. F. Danilevicz, P. E. Bayer, M. S. Bestry, D. Edwards, Application of machine learning and genomics for orphan crop improvement, Nature communications 16 (2025) 982.

[2] H. A. Ganatra, Machine learning in pediatric healthcare: Current trends, challenges, and future directions, Journal of Clinical Medicine 14 (2025) 807.

[3] A. K. V. N. Biju, A. S. Thomas, J. Thasneem, Examining the research taxonomy of artificial intelligence, deep learning & machine learning in the financial sphere—a bibliometric analysis, Quality & Quantity 58 (2024) 849–878.

[4] L. Jin, X. Zhai, K. Wang, K. Zhang, D. Wu, A. Nazir, J. Jiang, W.-H. Liao, Big data, machine learning, and digital twin assisted additive manufacturing: A review, Materials & Design (2024) 113086.

[5] R. R. A. Guste, A. K. S. Ong, Machine learning decision system on the empirical analysis of the actual usage of interactive entertainment: A perspective of sustainable innovative technology, Computers 13 (2024) 128.

[6] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial intelligence and statistics, PMLR, 2017, pp. 1273–1282.

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

[8] H. R. Kumbhar, S. S. Rao, Federated learning enabled multi-key homomorphic encryption, Expert Systems with Applications 268 (2025) 126197.

[9] D. M. Jimenez Gutierrez, H. M. Hassan, L. Landi, A. Vitaletti, I. Chatzigiannakis, Application of federated learning techniques for arrhythmia classification using 12-lead ecg signals, in: International Symposium on Algorithmic Aspects of Cloud Computing, Springer, 2023, pp. 38–65.

[10] Y. Kong, Z. Li, C. Jiang, Asia: A federated boosting tree model against sequence inference attacks in financial networks, IEEE Transactions on Information Forensics and Security (2024).

[11] S. Salim, N. Moustafa, A. Almorjan, Responsible deep federated learning-based threat detection for satellite communications, IEEE Internet of Things Journal (2025).

[12] K. Hu, S. Gong, Q. Zhang, C. Seng, M. Xia, S. Jiang, An overview of implementing security and privacy in federated learning, Artificial Intelligence Review 57 (2024) 1–66.

[13] E. Hallaji, R. Razavi-Far, M. Saif, B. Wang, Q. Yang, Decentralized federated learning: A survey on security and privacy, IEEE Transactions on Big Data (2024).

[14] A. K. Nair, E. D. Raj, J. Sahoo, A robust analysis of adversarial attacks on federated learning environments, Computer Standards and Interfaces 86 (2023) 103723. URL: `https://www.sciencedirect.com/science/article/pii/S0920548923000041`. doi:`https://doi.org/10.1016/j.csi.2023.103723`.

[15] H. N. C. Neto, J. Hribar, I. Dusparic, D. M. F. Mattos, N. C. Fernandes, A survey on securing federated learning: Analysis of applications, attacks, challenges, and trends, IEEE ACCESS 11 (2023) 41928–41953.

[16] H. Li, C. Li, J. Wang, A. Yang, Z. Ma, Z. Zhang, D. Hua, Review on security of federated learning and its application in healthcare, Future Generation Computer Systems 144 (2023) 271–290.

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

[18] X. Gong, Y. Chen, Q. Wang, W. Kong, Backdoor attacks and defenses in federated learning: State-of-the-art, taxonomy, and future directions, IEEE Wireless Communications 30 (2022) 114–121.

[19] A. Qammar, J. Ding, H. Ning, Federated learning attack surface: taxonomy, cyber defences, challenges, and future directions, Artificial Intelligence Review (2022) 1–38.

[20] P. Liu, X. Xu, W. Wang, Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives, Cybersecurity 5 (2022) 4.

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

[22] X. Yin, Y. Zhu, J. Hu, A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions, ACM Computing Surveys (CSUR) 54 (2021) 1–36.

[23] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, G. Srivastava, A survey on security and privacy of federated learning, Future Generation Computer Systems 115 (2021) 619–640.

[24] M. Alazab, S. P. RM, M. Parimala, P. K. R. Maddikunta, T. R. Gadekallu, Q.-V. Pham, Federated learning for cybersecurity: Concepts, challenges, and future directions, IEEE Transactions on Industrial Informatics 18 (2021) 3501–3509.

[25] N. Truong, K. Sun, S. Wang, F. Guitton, Y. Guo, Privacy preservation in federated learning: An insightful survey from the gdpr perspective, Computers & Security 110 (2021) 102402.

[26] I. Kholod, E. Yanaki, D. Fomichev, E. Shalugin, E. Novikova, E. Filippov, M. Nordlund, Open-source federated learning frameworks for iot: A comparative review and analysis, Sensors 21 (2020) 167.

[27] A. P. Siddaway, A. M. Wood, L. V. Hedges, How to do a systematic review: a best practice guide for conducting and reporting narrative reviews, meta-analyses, and meta-syntheses, Annual review of psychology 70 (2019) 747–770.

[28] H. Elayan, M. Aloqaily, M. Guizani, Deep federated learning for iot-based decentralized healthcare systems, in: 2021 International Wireless Communications and Mobile Computing (IWCMC), IEEE, 2021, pp. 105–109.

[29] M. Zhang, Y. Wang, T. Luo, Federated learning for arrhythmia detection

of non-iid ecg, in: 2020 IEEE 6th International Conference on Computer and Communications (ICCC), IEEE, 2020, pp. 1176–1180.

[30] S. Sakib, M. M. Fouda, Z. M. Fadlullah, K. Abualsaud, E. Yaacoub, M. Guizani, Asynchronous federated learning-based ecg analysis for arrhythmia detection, in: 2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom), IEEE, 2021, pp. 277–282.

[31] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, B. He, A survey on federated learning systems: Vision, Hype and Reality for Data Privacy and Protection (2019).

[32] D. M. Jimenez, A. Anagnostopoulos, I. Chatzigiannakis, A. Vitaletti, Fedartml: A tool to facilitate the generation of non-iid datasets in a controlled way to support federated learning research, IEEE Access 12 (2024) 81004–81016. doi:10.1109/ACCESS.2024.3410026.

[33] S. Saha, T. Ahmad, Federated transfer learning: Concept and applications, Intelligenza Artificiale 15 (2021) 35–44.

[34] K. Cheedella, S. Fathimabi, D. Chinamuttevi, Amazon product recommendation system using apache spark, in: 2024 14th International Conference on Cloud Computing, Data Science & Engineering (Confluence), IEEE, 2024, pp. 223–227.

[35] S. Wu, Z. Xu, Y. Zhang, Y. Zhang, D. Ramage, Prompt public large language models to synthesize data for private on-device applications, arXiv preprint arXiv:2404.04360 (2024).

[36] C. Thapa, P. C. M. Arachchige, S. Camtepe, L. Sun, Splitfed: When federated learning meets split learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, 2022, pp. 8485–8493.

[37] C. Thapa, M. A. P. Chamikara, S. A. Camtepe, Advancements of federated learning towards privacy preservation: from federated learning to split learning, Federated Learning Systems: Towards Next-Generation AI (2021) 79–109.

[38] A. Singh, P. Vepakomma, O. Gupta, R. Raskar, Detailed comparison of communication efficiency of split learning and federated learning, arXiv preprint arXiv:1909.09145 (2019).

[39] D. M. J. G., D. Solans, M. Heikkila, A. Vitaletti, N. Kourtellis, A. Anagnostopoulos, I. Chatzigiannakis, Non-iid data in federated learning: A survey with taxonomy, metrics, methods, frameworks and future directions, 2024. URL: https://arxiv.org/abs/2411.12377. arXiv:2411.12377.

[40] D. Yu, H. Zhang, Y. Huang, Z. Xie, Data distribution inference attack in federated learning via reinforcement learning support, High-Confidence Computing 5 (2025) 100235.

[41] K. Zhao, L. Wang, F. Yu, B. Zeng, Z. Pang, Fedmp: A multi-pronged defense algorithm against byzantine poisoning attacks in federated learning, Computer Networks 257 (2025) 110990.

[42] B. Dong, D. Chen, Y. Wu, S. Tang, Y. Zhuang, Fadngs: Federated learning for anomaly detection, IEEE Transactions on Neural Networks and Learning Systems (2024).

[43] L. Lamport, R. Shostak, M. Pease, The byzantine generals problem, in: Concurrency: the works of leslie lamport, 2019, pp. 203–226.

[44] M. Fang, X. Cao, J. Jia, N. Gong, Local model poisoning attacks to {Byzantine-Robust} federated learning, in: 29th USENIX security symposium (USENIX Security 20), 2020, pp. 1605–1622.

[45] V. Shejwalkar, A. Houmansadr, Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning, in: NDSS, 2021, p. 19.

[46] Q. H. Nguyen, N. Ngoc-Hieu, T.-A. Ta, T. Nguyen-Tang, K.-S. Wong, H. Thanh-Tung, K. D. Doan, Wicked oddities: Selectively poisoning for effective clean-label backdoor attacks, 2024. URL: https://arxiv.org/abs/2407.10825. arXiv:2407.10825.

[47] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, T. Goldstein, Poison frogs! targeted clean-label poisoning attacks on neural networks, 2018. URL: https://arxiv.org/abs/1804.00792. arXiv:1804.00792.

[48] J. Zhang, B. Chen, X. Cheng, H. T. T. Binh, S. Yu, Poisongan: Generative poisoning attacks against federated learning in edge computing systems, IEEE Internet of Things Journal 8 (2021) 3310–3322. doi:10.1109/JIOT.2020.3023126.

[49] P. Gupta, K. Yadav, B. B. Gupta, M. Alazab, T. R. Gadekallu, A novel data poisoning attack in federated learning based on inverted loss function, Computers and Security 130 (2023) 103270. URL: https://www.sciencedirect.com/science/article/pii/S0167404823001803. doi:https://doi.org/10.1016/j.cose.2023.103270.

[50] V. Tolpegin, S. Truex, M. E. Gursoy, L. Liu, Data poisoning attacks against federated learning systems, 2020. URL: https://arxiv.org/abs/2007.08432. arXiv:2007.08432.

[51] Y. Sun, H. Ochiai, J. Sakuma, Attacking-distance-aware attack: Semi-targeted model poisoning on federated learning, IEEE Transactions on Artificial Intelligence 5 (2024) 925–939. doi:10.1109/TAI.2023.3280155.

[52] L. Shi, Z. Chen, Y. Shi, G. Zhao, L. Wei, Y. Tao, Y. Gao, Data poisoning attacks on federated learning by using adversarial samples, in: 2022 International Conference on Computer Engineering and Artificial Intelligence (ICCEAI), 2022, pp. 158–162. doi:10.1109/ICCEAI55464.2022.00041.

[53] A. R. Shahid, A. Imteaj, P. Y. Wu, D. A. Igoche, T. Alam, Label flipping data poisoning attack against wearable human activity recognition system, 2022. URL: https://arxiv.org/abs/2208.08433. arXiv:2208.08433.

[54] M. Baruch, G. Baruch, Y. Goldberg, A little is enough: Circumventing defenses for distributed learning, 2019. URL: https://arxiv.org/abs/1902.06156. arXiv:1902.06156.

[55] S. Wang, G. Zuccon, An analysis of untargeted poisoning attack and defense methods for federated online learning to rank systems, in: Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval, 2023, pp. 215–224.

[56] F. Qiao, Z. Li, Y. Kong, A privacy-aware and incremental defense method against gan-based poisoning attack, IEEE Transactions on Computational Social Systems (2023).

[57] J. Zhang, J. Chen, D. Wu, B. Chen, S. Yu, Poisoning attack in federated learning using generative adversarial nets, in: 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), 2019, pp. 374–380. doi:10.1109/TrustCom/BigDataSE.2019.00057.

[58] W. Sun, B. Gao, K. Xiong, Y. Wang, A gan-based data poisoning attack against federated learning systems and its countermeasure, 2024. URL: https://arxiv.org/abs/2405.11440. arXiv:2405.11440.

[59] X. Cao, N. Z. Gong, Mpaf: Model poisoning attacks to federated learning based on fake clients, 2022. URL: https://arxiv.org/abs/2203.08669. arXiv:2203.08669.

[60] C. Fung, C. J. M. Yoon, I. Beschastnikh, Mitigating sybils in federated learning poisoning, 2020. URL: https://arxiv.org/abs/1808.04866. arXiv:1808.04866.

[61] X. Xiao, Z. Tang, C. Li, B. Jiang, K. Li, Sbpa: Sybil-based backdoor poisoning attacks for distributed big data in aiot-based federated learning system, IEEE Transactions on Big Data (2022) 1–12. doi:10.1109/TBDATA.2022.3224392.

[62] T. Liu, X. Hu, T. Shu, Facilitating early-stage backdoor attacks in federated learning with whole population distribution inference, IEEE Internet of Things Journal 10 (2023) 10385–10399. doi:10.1109/JIOT.2023.3237806.

[63] Y. Dai, S. Li, Chameleon: Adapting to peer images for planting durable backdoors in federated learning, 2023. URL: https://arxiv.org/abs/2304.12961. arXiv:2304.12961.

[64] H. Zhang, J. Jia, J. Chen, L. Lin, D. Wu, A3fl: Adversarially adaptive backdoor attacks to federated learning, in: A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine (Eds.), Advances in Neural Information Processing Systems, volume 36, Curran Associates, Inc., 2023, pp. 61213–61233. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/c07d71ff0bc042e4b9acd626a79597fa-Paper-Conference.pdf.

[65] T. D. Nguyen, P. Rieger, R. De Viti, H. Chen, B. B. Brandenburg, H. Yalame, H. Möllering, H. Fereidooni, S. Marchal, M. Miettinen, et al., {FLAME}: Taming backdoors in federated learning, in: 31st USENIX Security Symposium (USENIX Security 22), 2022, pp. 1415–1432.

[66] J. Wang, X. Chang, J. Mišić, V. B. Mišić, Y. Wang, Pass: A parameter audit-based secure and fair federated learning scheme against free-rider attack, IEEE Internet of Things Journal 11 (2024) 1374–1384. URL: http://dx.doi.org/10.1109/JIOT.2023.3288936. doi:10.1109/jiot.2023.3288936.

[67] Y. Fraboni, R. Vidal, M. Lorenzi, Free-rider attacks on model aggregation in federated learning, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 1846–1854.

[68] J. Lin, M. Du, J. Liu, Free-riders in federated learning: Attacks and defenses, 2019. URL: https://arxiv.org/abs/1911.12560. arXiv:1911.12560.

[69] J. Domingo-Ferrer, A. Blanco-Justicia, J. Manjón, D. Sánchez, Secure and privacy-preserving federated learning via co-utility, IEEE Internet of Things Journal 9 (2022) 3988–4000. doi:10.1109/JIOT.2021.3102155.

[70] Z. Zhu, J. Shu, X. Zou, X. Jia, Advanced free-rider attacks in federated learning, in: the 1st NeurIPS Workshop on New Frontiers in Federated Learning Privacy, Fairness, Robustness, Personalization and Data Ownership, 2021, p. 10.

[71] E. T. M. Beltrán, M. Q. Pérez, P. M. S. Sánchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez, A. H. Celdrán, Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges, IEEE Communications Surveys & Tutorials (2023).

[72] Y. Shi, Y. E. Sagduyu, T. Erpek, Jamming attacks on decentralized federated learning in general multi-hop wireless networks, 2023. URL: https://arxiv.org/abs/2301.05250. arXiv:2301.05250.

[73] G. Kim, Y. Kim, The threat of disruptive jamming to blockchain-based decentralized federated learning in wireless networks, Sensors 24 (2024). URL: https://www.mdpi.com/1424-8220/24/2/535. doi:10.3390/s24020535.

[74] R. Schlegel, S. Kumar, E. Rosnes, A. G. i Amat, Codedpaddedfl and codedsecagg: Straggler mitigation and secure aggregation in federated learning, 2022. URL: https://arxiv.org/abs/2112.08909. arXiv:2112.08909.

[75] K. N. Kumar, R. Mitra, C. K. Mohan, Revamping federated learning security from a defender's perspective: A unified defense with homomorphic encrypted data space, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 24387–24397.

[76] Y. Li, Z. Guo, N. Yang, H. Chen, D. Yuan, W. Ding, Threats and defenses in federated learning life cycle: A comprehensive survey and challenges, 2024. URL: https://arxiv.org/abs/2407.06754. arXiv:2407.06754.

[77] T. Kim, S. Singh, N. Madaan, C. Joe-Wong, pfeddef: Characterizing evasion attack transferability in federated learning, Software Impacts 15 (2023) 100469. URL: https://www.sciencedirect.com/science/article/pii/S2665963823000064. doi:https://doi.org/10.1016/j.simpa.2023.100469.

[78] Q. Pang, Y. Yuan, S. Wang, W. Zheng, Adi: Adversarial dominating inputs in vertical federated learning systems, arXiv preprint arXiv:2201.02775 (2022).

[79] S. Li, D. Yao, J. Liu, Fedvs: Straggler-resilient and privacy-preserving vertical federated learning for split models, 2023. URL: https://arxiv.org/abs/2304.13407. arXiv:2304.13407.

[80] J. Park, D.-J. Han, M. Choi, J. Moon, Sageflow: Robust federated learning against both stragglers and adversaries, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan (Eds.), Advances in Neural Information Processing Systems, volume 34, Curran Associates, Inc., 2021, pp. 840–851. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/076a8133735eb5d7552dc195b125a454-Paper.pdf.

[81] S. Chowdhury, A. Mukherjee, R. Halder, fedrlchain: Secure federated deep reinforcement learning with blockchain, IEEE Transactions on Services Computing 16 (2023) 3865–3878. doi:10.1109/TSC.2023.3294063.

[82] B. Deressa, M. A. Hasan, Trustbandit: Optimizing client selection for robust federated learning against poisoning attacks, in: IEEE INFOCOM 2024 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2024, pp. 1–8. doi:10.1109/INFOCOMWKSHPS61880.2024.10620802.

[83] P. Blanchard, E. M. E. Mhamdi, R. Guerraoui, J. Stainer, Byzantine-tolerant machine learning, 2017. URL: https://arxiv.org/abs/1703.02757. arXiv:1703.02757.

[84] C. Xie, O. Koyejo, I. Gupta, Generalized byzantine-tolerant sgd, 2018. URL: https://arxiv.org/abs/1802.10116. arXiv:1802.10116.

[85] S. Li, E. C.-H. Ngai, T. Voigt, An experimental study of byzantine-robust aggregation schemes in federated learning, IEEE Transactions on Big Data (2024) 1–13. URL: http://dx.doi.org/10.1109/TBDATA.2023.3237397. doi:10.1109/tbdata.2023.3237397.

[86] K. Pillutla, S. M. Kakade, Z. Harchaoui, Robust aggregation for federated learning, IEEE Transactions on Signal Processing 70 (2022) 1142–1154. URL: http://dx.doi.org/10.1109/TSP.2022.3153135. doi:10.1109/tsp.2022.3153135.

[87] S. Li, E. Ngai, T. Voigt, Byzantine-robust aggregation in federated learning empowered industrial iot, IEEE Transactions on Industrial Informatics 19 (2023) 1165–1175. doi:10.1109/TII.2021.3128164.

[88] S. Han, B. Buyukates, Z. Hu, H. Jin, W. Jin, L. Sun, X. Wang, W. Wu, C. Xie, Y. Yao, K. Zhang, Q. Zhang, Y. Zhang, C. Joe-Wong, S. Avestimehr, C. He, Fedsecurity: Benchmarking attacks and defenses in federated learning and federated llms, 2024. URL: https://arxiv.org/abs/2306.04959. arXiv:2306.04959.

[89] E. M. E. Mhamdi, R. Guerraoui, S. Rouault, The hidden vulnerability of distributed learning in byzantium, 2018. URL: https://arxiv.org/abs/1802.07927. arXiv:1802.07927.

[90] C. Xie, O. Koyejo, I. Gupta, Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance, 2019. URL: https://arxiv.org/abs/1805.10032. arXiv:1805.10032.

[91] C. Xie, S. Koyejo, I. Gupta, Zeno++: Robust fully asynchronous sgd, 2021. URL: https://arxiv.org/abs/1903.07020. arXiv:1903.07020.

[92] Y. Jiang, Y. Li, Y. Zhou, X. Zheng, Sybil attacks and defense on differential privacy based federated learning, in: 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2021, pp. 355–362. doi:10.1109/TrustCom53373.2021.00062.

[93] Z. Pan, Z. Ying, Y. Wang, C. Zhang, C. Li, L. Zhu, One-shot backdoor removal for federated learning, IEEE Internet of Things Journal (2024) 1–1. doi:10.1109/JIOT.2024.3438150.

[94] R. Lu, W. Zhang, Q. Li, H. He, X. Zhong, H. Yang, D. Wang, Z. Xu, M. Alazab, Adaptive asynchronous federated learning, Future Generation Computer Systems 152 (2024) 193–206. URL: https://www.sciencedirect.com/science/article/pii/S0167739X23004004. doi:https://doi.org/10.1016/j.future.2023.11.001.

[95] X. Li, Z. Song, J. Yang, Federated adversarial learning: A framework with convergence analysis, in: A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, J. Scarlett (Eds.), Proceedings of the 40th International Conference on Machine Learning, volume 202 of *Proceedings of Machine Learning Research*, PMLR, 2023, pp. 19932–19959. URL: https://proceedings.mlr.press/v202/li23z.html.

[96] C. Fung, C. J. M. Yoon, I. Beschastnikh, The limitations of federated learning in sybil settings, in: 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020), USENIX Association, San Sebastian, 2020, pp. 301–316. URL: https://www.usenix.org/conference/raid2020/presentation/fung.

[97] N. Jebreel, J. Domingo-Ferrer, Fl-defender: Combating targeted attacks in federated learning, 2022. URL: https://arxiv.org/abs/2207.00872. arXiv:2207.00872.

[98] C.-W. Ching, T.-C. Lin, K.-H. Chang, C.-C. Yao, J.-J. Kuo, Model partition defense against gan attacks on collaborative learning via mobile edge computing, in: GLOBECOM 2020 - 2020 IEEE Global Communications Conference, 2020, pp. 1–6. doi:10.1109/GLOBECOM42002.2020.9322591.

[99] G. Xu, H. Li, Y. Zhang, S. Xu, J. Ning, R. H. Deng, Privacy-preserving federated deep learning with irregular users, IEEE Transactions on Dependable and Secure Computing 19 (2022) 1364–1381. doi:10.1109/TDSC.2020.3005909.

[100] C. Zhu, S. Roos, L. Y. Chen, LeadFL: Client self-defense against model poisoning in federated learning, in: A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, J. Scarlett (Eds.), Proceedings of the 40th International Conference on Machine Learning, volume 202 of *Proceedings of Machine Learning Research*, PMLR, 2023, pp. 43158–43180. URL: https://proceedings.mlr.press/v202/zhu23j.html.

[101] S. Li, E. C.-H. Ngai, T. Voigt, An experimental study of byzantine-robust aggregation schemes in federated learning, IEEE Transactions on Big Data (2023).

[102] H. Zhang, Y. Liu, X. He, J. Wu, T. Cong, X. Huang, Sok: Benchmarking

poisoning attacks and defenses in federated learning, arXiv preprint arXiv:2502.03801 (2025).

[103] C. Wu, X. Yang, S. Zhu, P. Mitra, Mitigating backdoor attacks in federated learning, arXiv preprint arXiv:2011.01767 (2020).

[104] M. Rigaki, S. Garcia, A survey of privacy attacks in machine learning, ACM Computing Surveys (2020).

[105] S. Kariyappa, C. Guo, K. Maeng, W. Xiong, G. E. Suh, M. K. Qureshi, H.-H. S. Lee, Cocktail party attack: Breaking aggregation-based privacy in federated learning using independent component analysis, in: International Conference on Machine Learning, PMLR, 2023, pp. 15884–15899.

[106] Z. Li, L. Wang, G. Chen, Z. Zhang, M. Shafiq, Z. Gu, E2egi: End-to-end gradient inversion in federated learning, IEEE Journal of Biomedical and Health Informatics 27 (2022) 756–767.

[107] D. Pasquini, M. Raynal, C. Troncoso, On the (in) security of peer-to-peer decentralized machine learning, in: 2023 IEEE Symposium on Security and Privacy (SP), IEEE Computer Society, 2023, pp. 418–436.

[108] A. Hatamizadeh, H. Yin, P. Molchanov, A. Myronenko, W. Li, P. Dogra, A. Feng, M. G. Flores, J. Kautz, D. Xu, et al., Do gradient inversion attacks make federated learning unsafe?, IEEE Transactions on Medical Imaging (2023).

[109] F. Boenisch, A. Dziedzic, R. Schuster, A. S. Shamsabadi, I. Shumailov, N. Papernot, When the curious abandon honesty: Federated learning is not private, in: 2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P), IEEE, 2023, pp. 175–199.

[110] D. Pasquini, D. Francati, G. Ateniese, Eluding secure aggregation in federated learning via model inconsistency, in: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, 2022, pp. 2429–2443.

[111] J. Chen, Y. Zhao, Q. Li, X. Feng, K. Xu, Feddef: Defense against gradient leakage in federated learning-based network intrusion detection systems, IEEE Transactions on Information Forensics and Security (2023).

[112] M. E. Dahlgaard, M. W. Jørgensen, N. A. Fuglsang, H. Nassar, Analysing the influence of attack configurations on the reconstruction of medical images in federated learning, arXiv preprint arXiv:2204.13808 (2022).

[113] S. Maddock, A. Sablayrolles, P. Stock, Canife: Crafting canaries for empirical privacy measurement in federated learning, arXiv preprint arXiv:2210.02912 (2022).

[114] Y. Zhang, G. Bai, M. A. P. Chamikara, M. Ma, L. Shen, J. Wang, S. Nepal, M. Xue, L. Wang, J. Liu, Agrevader: Poisoning membership inference against byzantine-robust federated learning, in: Proceedings of the ACM Web Conference 2023, 2023, pp. 2371–2382.

[115] J. Jia, A. Salem, M. Backes, Y. Zhang, N. Z. Gong, Memguard: Defending against black-box membership inference attacks via adversarial examples, in: Proceedings of the 2019 ACM SIGSAC conference on computer and communications security, 2019, pp. 259–274.

[116] Z. Zhang, F. Zhou, C. Zhang, C. Wen, X. Hu, T. Wang, A personalized federated learning-based fault diagnosis method for data suffering from network attacks, Applied Intelligence (2023) 1–16.

[117] H. Huang, X. Lei, T. Xiang, Mitigating cross-client gans-based attack in federated learning, Multimedia Tools and Applications (2023) 1–25.

[118] C. Xu, G. Neglia, What else is leaked when eavesdropping federated learning?, in: CCS workshop Privacy Preserving Machine Learning (PPML), 2021, pp. 1–12.

[119] S. Guo, X. Wang, S. Long, H. Liu, L. Hai, T. H. Sam, A federated learning scheme meets dynamic differential privacy, CAAI Transactions on Intelligence Technology (2023).

[120] A. K. Nair, J. Sahoo, E. D. Raj, Privacy preserving federated learning framework for iomt based big data analysis using edge computing, Computer Standards & Interfaces 86 (2023) 103720.

[121] X. Jin, P.-Y. Chen, C.-Y. Hsu, C.-M. Yu, T. Chen, Cafe: Catastrophic data leakage in vertical federated learning, Advances in neural information processing systems 34 (2021) 994–1006.

[122] Z. Xing, Z. Zhang, M. Li, J. Liu, L. Zhu, G. Russello, M. R. Asghar, Zero-knowledge proof-based practical federated learning on blockchain, arXiv preprint arXiv:2304.05590 (2023).

[123] L. Bangalore, M. H. F. Sereshgi, C. Hazay, M. Venkitasubramaniam, Flag: A framework for lightweight robust secure aggregation, in: Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security, Association for Computing Machinery, New York, NY, USA, 2023, pp. 14–28.

[124] M. Rathee, C. Shen, S. Wagh, R. A. Popa, Elsa: Secure aggregation for federated learning with malicious actors, in: 2023 IEEE Symposium on Security and Privacy (SP), IEEE, 2023, pp. 1961–1979.

[125] S. Singh, S. Rathore, O. Alfarraj, A. Tolba, B. Yoon, A framework for privacy-preservation of iot healthcare data using federated learning and blockchain technology, Future Generation Computer Systems 129 (2022) 380–388.

[126] M. Mansouri, M. Onen, W. B. Jaballah, M. Conti, Sok: Secure aggregation based on cryptographic schemes for federated learning, Proc. Priv. Enhancing Technol (2023) 140–157.

[127] W. Du, M. Li, L. Wu, Y. Han, T. Zhou, X. Yang, A efficient and robust privacy-preserving framework for cross-device federated learning, Complex & Intelligent Systems (2023) 1–15.

[128] B. Nagy, I. Hegedűs, N. Sándor, B. Egedi, H. Mehmood, K. Saravanan, G. Lóki, Á. Kiss, Privacy-preserving federated learning and its application to natural language processing, Knowledge-Based Systems 268 (2023) 110475.

[129] Z. Li, J. Zhang, L. Liu, J. Liu, Auditing privacy defenses in federated learning via generative gradient leakage, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10132–10142.

[130] B. Zhao, X. Liu, W.-N. Chen, R. Deng, Crowdfl: privacy-preserving mobile crowdsensing system via federated learning, IEEE Transactions on Mobile Computing (2022).

[131] B. Zhao, X. Liu, W.-n. Chen, When crowdsensing meets federated learning: Privacy-preserving mobile crowdsensing system, arXiv preprint arXiv:2102.10109 (2021).

[132] J. Sun, A. Li, B. Wang, H. Yang, H. Li, Y. Chen, Soteria: Provable defense against privacy leakage in federated learning from representation perspective, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 9311–9319.

[133] X. Cao, M. Fang, J. Liu, N. Z. Gong, Fltrust: Byzantine-robust federated learning via trust bootstrapping, arXiv preprint arXiv:2012.13995 (2020).

[134] L. Burkhalter, H. Lycklama, A. Viand, N. Küchler, A. Hithnawi, Rofl: Attestable robustness for secure federated learning, arXiv e-prints (2021) arXiv–2107.

[135] H. Lycklama, L. Burkhalter, A. Viand, N. Küchler, A. Hithnawi, Rofl: Robustness of secure federated learning, in: 2023 IEEE Symposium on Security and Privacy (SP), IEEE, 2023, pp. 453–476.

[136] H. Corrigan-Gibbs, D. Boneh, Prio: Private, robust, and scalable computation of aggregate statistics, in: 14th USENIX symposium on networked systems design and implementation (NSDI 17), 2017, pp. 259–282.

[137] S. Zahri, H. Bennouri, A. M. Abdelmoniem, An empirical study of efficiency and privacy of federated learning algorithms, arXiv preprint arXiv:2312.15375 (2023).

[138] Y. Zhu, Y. Wu, Z. Luo, B. C. Ooi, X. Xiao, Secure and verifiable data collaboration with low-cost zero-knowledge proofs, arXiv preprint arXiv:2311.15310 (2023).

[139] S. Das, S. R. Chowdhury, N. Chandran, D. Gupta, S. Lokam, R. Sharma, Communication efficient secure and private multi-party deep learning, Proceedings on Privacy Enhancing Technologies (2025).

[140] B. Zhang, G. Lu, P. Qiu, X. Gui, Y. Shi, Advancing federated learning through verifiable computations and homomorphic encryption, Entropy 25 (2023) 1550.

[141] M. Nasr, R. Shokri, A. Houmansadr, Comprehensive privacy analysis of deep learning: Stand-alone and federated learning under passive and active white-box inference attacks, ArXiv abs/1812.00910 (2018). URL: https://api.semanticscholar.org/CorpusID:54444175.

[142] F. Tramer, D. Boneh, Differentially private learning needs better features (or much more data), arXiv preprint arXiv:2011.11660 (2020).

[143] S. Mohammadi, A. Balador, S. Sinaei, F. Flammini, Balancing privacy and performance in federated learning: a systematic literature review on methods and metrics, Journal of Parallel and Distributed Computing 192 (2024) 104918. doi:10.1016/j.jpdc.2024.104918.

[144] B. Knott, S. Venkataraman, A. Hannun, S. Sengupta, M. Ibrahim, L. van der Maaten, Crypten: Secure multi-party computation meets machine learning, Advances in Neural Information Processing Systems 34 (2021) 4961–4973.

[145] Y. Liu, T. Fan, T. Chen, Q. Xu, Q. Yang, Fate: An industrial grade platform for collaborative learning with data protection, The Journal of Machine Learning Research 22 (2021) 10320–10325.

[146] C. He, S. Li, J. So, X. Zeng, M. Zhang, H. Wang, X. Wang, P. Vepakomma, A. Singh, H. Qiu, et al., Fedml: A research library and benchmark for federated machine learning, arXiv preprint arXiv:2007.13518 0 (2020) 0.

[147] S. Han, B. Buyukates, Z. Hu, H. Jin, W. Jin, L. Sun, X. Wang, C. Xie, K. Zhang, Q. Zhang, et al., Fedmlsecurity: A benchmark for attacks and defenses in federated learning and llms, arXiv preprint arXiv:2306.04959 0 (2023) 0.

[148] W. Jin, Y. Yao, S. Han, C. Joe-Wong, S. Ravi, S. Avestimehr, C. He, Fedml-he: An efficient homomorphic-encryption-based privacy-preserving federated learning system, arXiv preprint arXiv:2303.10837 0 (2023) 0.

[149] F. Lai, Y. Dai, S. Singapuram, J. Liu, X. Zhu, H. Madhyastha, M. Chowdhury, Fedscale: Benchmarking model and system performance of federated learning at scale, in: International Conference on Machine Learning, PMLR, 2022, pp. 11814–11827.

[150] N. Rodríguez-Barroso, G. Stipcich, D. Jiménez-López, J. A. Ruiz-Millán, E. Martínez-Cámara, G. González-Seco, M. V. Luzón, M. A. Veganzones, F. Herrera, Federated learning and differential privacy: Software tools analysis, the sherpa. ai fl framework and methodological guidelines for preserving data privacy, Information Fusion 64 (2020) 270–292.

[151] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. H. Li, T. Parcollet, P. P. B. de Gusmão, et al., Flower: A friendly federated learning research framework, arXiv preprint arXiv:2007.14390 (2020).

[152] K. H. Li, P. P. B. de Gusmão, D. J. Beutel, N. D. Lane, Secure aggregation for federated learning in flower, in: Proceedings of the 2nd ACM International Workshop on Distributed Machine Learning, 2021, pp. 8–14.

[153] M. Hipolito Garcia, A. Manoel, D. Madrigal Diaz, F. Mireshghallah, R. Sim, D. Dimitriadis, Flute: A scalable, extensible framework for high-performance federated learning simulations, arXiv e-prints (2022) arXiv–2203.

[154] H. R. Roth, Y. Cheng, Y. Wen, I. Yang, Z. Xu, Y.-T. Hsieh, K. Kersten, A. Harouni, C. Zhao, K. Lu, et al., Nvidia flare: Federated learning from simulation to real-world, arXiv preprint arXiv:2210.13291 (2022).

[155] P. Foley, M. J. Sheller, B. Edwards, S. Pati, W. Riviera, M. Sharma, P. N. Moorthy, S.-h. Wang, J. Martin, P. Mirhaji, et al., Openfl: the open federated learning library, Physics in Medicine & Biology 67 (2022) 214001.

[156] PaddlePaddle, Paddlefl, 2019. URL: https://github.com/PaddlePaddle/PaddleFL.

[157] A. Ziller, A. Trask, A. Lopardo, B. Szymkow, B. Wagner, E. Bluemke, J.-M. Nounahon, J. Passerat-Palmbach, K. Prakash, N. Rose, et al., Pysyft: A library for easy federated learning, Federated Learning Systems: Towards Next-Generation AI (2021) 111–139.

[158] Google, Tff, 2015. URL: https://www.tensorflow.org/federated.

[159] Google, Tff, 2015. URL: https://github.com/tensorflow/federated/tree/6477a3dba6e7d852191bfd733f651fad84b82eab/tensorflow_federated/python/research/targeted_attack.

[160] H. Wang, Y. Zhou, C. Zhang, C. Peng, M. Huang, Y. Liu, L. Zhang, Xfl: A high performace, lightweighted federated learning framework, arXiv preprint arXiv:2302.05076 (2023).

[161] H. Batool, A. Anjum, A. Khan, S. Izzo, C. Mazzocca, G. Jeon, A secure and privacy preserved infrastructure for vanets based on federated learning with local differential privacy, Information Sciences 652 (2024) 119717.

[162] B. Birchman, G. Thamilarasu, Securing federated learning: Enhancing defense mechanisms against poisoning attacks, in: 2024 33rd International Conference on Computer Communications and Networks (ICCCN), IEEE, 2024, pp. 1–6.

[163] D. Wu, L. Hao, B. Wei, K. Hao, T. Han, L. He, Backdoor attack based on privacy inference against federated learning, in: 2024 7th International Symposium on Autonomous Systems (ISAS), IEEE, 2024, pp. 1–6.

[164] C. Liu, J. Wang, D. Yu, Raf-gi: Towards robust, accurate and fast-convergent gradient inversion attack in federated learning, arXiv preprint arXiv:2403.08383 (2024).

[165] L. Peng, G. Luo, S. Zhou, J. Chen, Z. Xu, J. Sun, R. Zhang, An in-depth evaluation of federated learning on biomedical natural language processing for information extraction, NPJ Digital Medicine 7 (2024) 127.

[166] T. Qi, F. Wu, C. Wu, L. He, Y. Huang, X. Xie, Differentially private knowledge transfer for federated learning, Nature Communications 14 (2023) 3785.

[167] E. A. Mantey, C. Zhou, J. H. Anajemba, J. K. Arthur, Y. Hamid, A. Chowhan, O. O. Otuu, Federated learning approach for secured medical recommendation in internet of medical things using homomorphic encryption, IEEE Journal of Biomedical and Health Informatics (2024).

[168] X. Lessage, L. Collier, C.-H. B. Van Ouytsel, A. Legay, S. Mahmoudi, P. Massonet, Secure federated learning applied to medical imaging with fully homomorphic encryption, in: 2024 IEEE 3rd International Conference on AI in Cybersecurity (ICAIC), IEEE, 2024, pp. 1–12.

[169] B. Mali, P. K. Singh, N. Mazumdar, Safe-health: Guarding federated learning-driven smart healthcare with federated defense averaging against data poisoning, Security and Privacy (2024) e403.

[170] L. Sun, J. Tian, G. Muhammad, Fedkc: Personalized federated learning with robustness against model poisoning attacks in the metaverse for consumer health, IEEE Transactions on Consumer Electronics (2024).

[171] A. H. Omran, S. Y. Mohammed, M. Aljanabi, Detecting data poisoning attacks in federated learning for healthcare applications using deep learning, Iraqi Journal for Computer Science and Mathematics 4 (2023) 225–237.

[172] T.-N. Dao, T. P. Nguyen, Performance analysis of gradient inversion attack in federated learning with healthcare systems, REV Journal on Electronics and Communications 13 (2024).

[173] H. Sui, X. Sun, J. Zhang, B. Chen, W. Li, Multi-level membership inference attacks in federated learning based on active gan, Neural Computing and Applications 35 (2023) 17013–17027.

[174] T. Muazu, Y. Mao, A. U. Muhammad, M. Ibrahim, U. M. M. Kumshe, O. Samuel, A federated learning system with data fusion for healthcare using multi-party computation and additive secret sharing, Computer Communications 216 (2024) 168–182.

[175] H. Wu, Z. Zhao, L. Y. Chen, A. Van Moorsel, Federated learning for tabular data: Exploring potential risk to privacy, in: 2022 IEEE 33rd International Symposium on Software Reliability Engineering (ISSRE), IEEE, 2022, pp. 193–204.

[176] Z. Yang, H. Qu, Y. Hua, X. Zhang, X. Lin, Adversarial attacks on network intrusion detection systems based on federated learning, in: International Conference on Intelligent Computing, Springer, 2024, pp. 146–157.

[177] L. Lavaur, Y. Busnel, F. Autrel, Systematic analysis of label-flipping attacks against federated learning in collaborative intrusion detection systems, in: Proceedings of the 19th International Conference on Availability, Reliability and Security, 2024, pp. 1–12.

[178] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei, et al., Oscar: Object-semantics aligned pre-training for vision-language tasks, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16, Springer, 2020, pp. 121–137.

[179] S. Li, E. Ngai, T. Voigt, Byzantine-robust aggregation in federated learning empowered industrial iot, IEEE Transactions on Industrial Informatics 19 (2021) 1165–1175.

[180] A. Mangal, N. Kumar, Using big data to enhance the bosch production line performance: A kaggle challenge, in: 2016 IEEE international conference on big data (big data), IEEE, 2016, pp. 2029–2035.

[181] H. Yang, D. Gu, J. He, A robust and efficient federated learning algorithm against adaptive model poisoning attacks, IEEE Internet of Things Journal (2024).

[182] J. Zhang, Y. Hua, H. Wang, T. Song, Z. Xue, R. Ma, H. Guan, Fedala: Adaptive local aggregation for personalized federated learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, 2023, pp. 11237–11244.

[183] A. Panda, S. Mahloujifar, A. N. Bhagoji, S. Chakraborty, P. Mittal, Sparsefed: Mitigating model poisoning attacks in federated learning with sparsification, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2022, pp. 7587–7624.

[184] X. Ma, C. Wen, T. Wen, An asynchronous and real-time update paradigm of federated learning for fault diagnosis, IEEE Transactions on Industrial Informatics 17 (2021) 8531–8540.

[185] J. Sun, A. Li, L. DiValentin, A. Hassanzadeh, Y. Chen, H. Li, Fl-wbc: Enhancing robustness against model poisoning attacks in federated learning from a client perspective, Advances in neural information processing systems 34 (2021) 12613–12624.

[186] Y. Mi, Y. Sun, J. Guan, S. Zhou, Identifying backdoor attacks in federated learning via anomaly detection, in: Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data, Springer, 2023, pp. 111–126.

[187] L. Zhang, B. Bian, L. Luo, S. Li, H. Wang, Federated learning with multi-method adaptive aggregation for enhanced defect detection in power systems, Big Data and Cognitive Computing 8 (2024) 102.

[188] Z. Ma, H. Gao, S. Li, P. Wang, A stability-enhanced dynamic backdoor defense in federated learning for iiot, IEEE Transactions on Industrial Informatics (2024).

[189] R. Asif, S. R. Hassan, G. Parr, Integrating a blockchain-based governance framework for responsible ai, Future Internet 15 (2023) 97.

[190] I. A. Khan, I. Razzak, D. Pi, N. Khan, Y. Hussain, B. Li, T. Kousar, Fed-inforce-fusion: A federated reinforcement-based fusion model for security and privacy protection of iomt networks against cyber-attacks, Information Fusion 101 (2024) 102002.

[191] K. Yue, R. Jin, C.-W. Wong, D. Baron, H. Dai, Gradient obfuscation gives a false sense of security in federated learning, in: 32nd USENIX Security Symposium (USENIX Security 23), 2023, pp. 6381–6398.

[192] Y. Arjoune, S. Faruque, Smart jamming attacks in 5g new radio: A review, in: 2020 10th annual computing and communication workshop and conference (CCWC), IEEE, 2020, pp. 1010–1015.

[193] S. Hou, S. Li, B. Buyukates, Privacy-preserving prompt personalization in federated learning for multimodal large language models, arXiv preprint arXiv:2505.22447 (2025).

[194] S. Kuili, M. Amini, B. Kantarci, A two-stage cae-based federated learning framework for efficient jamming detection in 5g networks, arXiv preprint arXiv:2501.15288 (2025).

[195] J. Wang, X. Chang, J. Mišić, V. B. Mišić, Y. Wang, Pass: A parameter audit-based secure and fair federated learning scheme against free-rider attack, IEEE Internet of Things Journal 11 (2023) 1374–1384.

[196] Y. H. Ezzeldin, S. Yan, C. He, E. Ferrara, A. S. Avestimehr, Fairfed: Enabling group fairness in federated learning, in: Proceedings of the AAAI conference on artificial intelligence, volume 37, 2023, pp. 7494–7502.

[197] W. Du, D. Xu, X. Wu, H. Tong, Fairness-aware agnostic federated learning, in: Proceedings of the 2021 SIAM International Conference on Data Mining (SDM), SIAM, 2021, pp. 181–189.

[198] Y. Yang, M. Hu, Y. Zhou, X. Liu, D. Wu, Csra: Robust incentive mechanism design for differentially private federated learning, IEEE Transactions on Information Forensics and Security (2023).

[199] M. Zhang, E. Wei, R. Berry, Faithful edge federated learning: Scalability and privacy, IEEE Journal on Selected Areas in Communications 39 (2021) 3790–3804.

[200] J. Xiong, H. Zhu, Privmaskfl: A private masking approach for heterogeneous federated learning in iot, Computer Communications 214 (2024) 100–112.

[201] X. Kong, W. Zhang, H. Wang, M. Hou, X. Chen, X. Yan, S. K. Das, Federated graph anomaly detection via contrastive self-supervised learning, IEEE Transactions on Neural Networks and Learning Systems (2024).

[202] B. Buyukates, J. So, H. Mahdavifar, S. Avestimehr, Lightverifl: A lightweight and verifiable secure aggregation for federated learning, IEEE Journal on Selected Areas in Information Theory (2024).

[203] J. Cai, W. Shen, J. Qin, Esvfl: Efficient and secure verifiable federated learning with privacy-preserving, Information Fusion 109 (2024) 102420.

[204] W. Boitier, A. Del Pozzo, Á. García-Pérez, S. Gazut, P. Jobic, A. Lemaire, E. Mahe, A. Mayoue, M. Perion, T. F. Rezende, et al., Fantastyc: Blockchain-based federated learning made secure and practical, in: 2024 43rd International Symposium on Reliable Distributed Systems (SRDS), IEEE, 2024, pp. 260–270.

[205] P. Sethi, Reinforcement Learning assisted Adaptive difficulty of Proof of Work (PoW) in Blockchain-enabled Federated Learning, Ph.D. thesis, Virginia Tech, 2023.

[206] Z. He, F. Zhang, Y. Li, Y. Cao, Z. Cai, Privacy-enhanced personalized federated learning with layer-wise gradient shielding on heterogeneous iot data, IEEE Internet of Things Journal (2024).

[207] A. R. Elkordy, S. Prakash, S. Avestimehr, Basil: A fast and byzantine-resilient approach for decentralized training, IEEE Journal on Selected Areas in Communications 40 (2022) 2694–2716.

[208] J. Wang, R. Xin, O. Alfarraj, A. Tolba, Q. Tang, Privacy preserving security using multi-key homomorphic encryption for face recognition, Expert Systems 42 (2025) e13645.

[209] A. Muñoz, R. Rios, R. Román, J. López, A survey on the (in) security of trusted execution environments, Computers & Security 129 (2023) 103180.

[210] P. Morettin, A. Passerini, R. Sebastiani, A unified framework for probabilistic verification of ai systems via weighted model integration, arXiv preprint arXiv:2402.04892 (2024).

[211] X. Guo, Z. Liu, J. Li, J. Gao, B. Hou, C. Dong, T. Baker, V eri fl: Communication-efficient and fast verifiable aggregation for federated learning, IEEE Transactions on Information Forensics and Security 16 (2020) 1736–1751.

[212] T. Li, S. Cheng, T. L. Chan, H. Hu, A polynomial proxy model approach to verifiable decentralized federated learning, Scientific Reports 14 (2024) 28786.

[213] D. He, S. Chan, M. Guizani, An accountable, privacy-preserving, and efficient authentication framework for wireless access networks, IEEE Transactions on Vehicular Technology 65 (2015) 1605–1614.

[214] X. Liu, Z. Zhou, Y. Wang, J. He, B. Zhang, X. Yang, J. Zhang, Scalable collaborative zk-snark and its application to efficient proof outsourcing, Cryptology ePrint Archive (2024).

[215] B. Yan, K. Li, M. Xu, Y. Dong, Y. Zhang, Z. Ren, X. Cheng, On protecting the data privacy of large language models (llms): A survey, arXiv preprint arXiv:2403.05156 (2024).

[216] D. A. Alber, Z. Yang, A. Alyakin, E. Yang, S. Rai, A. A. Valliani, J. Zhang, G. R. Rosenbaum, A. K. Amend-Thomas, D. B. Kurland, et al., Medical large language models are vulnerable to data-poisoning attacks, Nature Medicine 31 (2025) 618–626.

[217] Z. Liu, Y. Jiang, J. Shen, M. Peng, K.-Y. Lam, X. Yuan, X. Liu, A survey on federated unlearning: Challenges, methods, and future directions, ACM Computing Surveys 57 (2024) 1–38.