Journal of *Cybersecurity and Privacy*

MDPI

Review

Poisoning Attacks in Federated Learning: An Accountability- Oriented Survey with Centralized Learning as a Baseline

Safiia Mohammed, Dima Alhadidi and Alioune Ngom *

Department of Computer Science, University of Windsor, Windsor, ON N9B 3P4, Canada; mohamm7d@uwindsor.ca (S.M.); dima.alhadidi@uwindsor.ca (D.A.)
* Correspondence: angom@uwindsor.ca

Abstract

Artificial intelligence (AI) systems are increasingly deployed in high-stakes domains, where poisoning attacks can corrupt training data, manipulate model updates, or implant covert backdoors. This survey examines poisoning attacks in federated learning (FL), using centralized learning as a baseline to explain how distributed data, client heterogeneity, privacy-preserving aggregation, and untrusted coordination expand the threat surface. It positions prior surveys and synthesizes representative primary studies through an accountability-oriented lens focused on attribution, audit evidence, traceability, and forensic readiness. The review compares major attack classes, including data poisoning, model poisoning, backdoor insertion, server-side manipulation, Sybil behavior, collusion, and multi-round poisoning. It also evaluates countermeasures such as Byzantine-robust aggregation, anomaly detection, validation-based filtering, malicious-secure aggregation, authenticated update handling, provenance mechanisms, ledger-based evidence, and verifiable aggregation protocols. The analysis shows that robustness alone is insufficient for trustworthy FL unless defenses also preserve evidence that supports independent verification, post-incident reconstruction, and governance review. Persistent gaps remain in causal forensic attribution, privacy-preserving evidence governance, malicious-server threat modeling, scalable verifiability tooling, recovery after poisoning, and deployment-ready benchmarks. The survey concludes that accountable FL should be designed as an evidence-producing system, not merely as a privacy-preserving or attack-resistant training architecture, especially for regulated, cross-silo, and high-risk real-world deployments.

Keywords: poisoning attacks; federated learning; model poisoning; backdoor attacks; byzantine-robust aggregation; cryptographic verification; accountability; trustworthy AI

Academic Editors: Tao Zhang, Xiangyun Tang, Jiacheng Wang, Chuan Zhang and Jiqiang Liu

Received: 2 May 2026
Revised: 26 July 2026
Accepted: 28 July 2026
Published: 7 August 2026

1. Introduction

Artificial intelligence (AI), especially machine learning (ML), is transforming sectors ranging from healthcare and education to finance and public administration, but its expanding use has sharpened concerns about reliability, transparency, and safety [1]. Poisoning attacks are especially consequential because they corrupt the training process itself through malicious data, manipulated model updates, or hidden backdoors [2,3]. These risks become harder to analyze in federated learning (FL), where participants do not share raw data and where aggregation can itself become an attack surface.

Accountability is therefore central: a meaningful response to a poisoning incident requires responsible governance and justification [4,5], practical accountability mecha-

nisms [6], audit access and evidence [7], and traceability records showing who acted, what changed, and when [8]. Centralized learning can often rely on dataset provenance, training logs, and unified pipeline control. FL weakens these pathways because privacy-preserving mechanisms, client heterogeneity, and semi-trusted coordination reduce observability.

Existing surveys already cover ML poisoning countermeasures [3,9], broad attack taxonomies [10], general FL security [11,12], FL poisoning [13,14], and FL model-poisoning defenses [15]. Recent primary studies also examine network-focused data-poisoning vulnerabilities [16], auditable participant selection [17], zero-knowledge aggregation [18,19], public auditability [20], privacy-preserving auditability [21], and provenance-oriented transparency [22].

The remaining gap identified in the screened survey corpus is integrative: these strands are not compared in that corpus within a single FL-oriented accountability framework that treats attribution, audit evidence, and untrusted-server behavior alongside robustness. All gap claims in this manuscript are bounded by the searched databases, search period, inclusion criteria, screened corpus, and core-survey comparison described in Section 2. This manuscript addresses that gap through three contributions:

1. It identifies how poisoning changes when it moves from centralized training to FL: reduced observability, weaker attribution, new server-side manipulation and collusion threats, and a greater need for audit-ready evidence.
2. It develops an Accountability-Integrated Taxonomy (AIT) that extends conventional poisoning categories by adding observability, attribution granularity, audit evidence, and trust assumptions.
3. It synthesizes aggregation-based, detection-based, cryptographic, and governance-oriented defenses through a common accountability lens, emphasizing untrusted servers, verifiable aggregation, and standards-relevant evidence.

2. Survey Methodology

This manuscript follows a bounded scoping-review design. It uses structured screening, predefined coding fields, and claim calibration to support transparent synthesis rather than pooled effect estimation. The method documents explicit search strings, eligibility rules, deduplication procedures, a PRISMA-style flow summary, a survey-like screening log, and a core-survey comparison matrix so that readers can reconstruct the literature base and novelty claims.

The review addresses four questions. RQ1 asks how prior surveys delimit poisoning in FL. RQ2 examines underexplored threat assumptions, especially server-side manipulation and collusion. RQ3 considers how defense families perform under non-independent and non-identically distributed (non-IID) data, secure aggregation, and untrusted-server assumptions. RQ4 identifies the accountability artifacts that make defenses auditable rather than merely robust.

2.1. Reproducible Search Protocol

The search covered IEEE Xplore, the ACM Digital Library, Scopus, Web of Science Core Collection, Elsevier ScienceDirect, SpringerLink, arXiv, and Google Scholar. Database searches were conducted between 10 January 2026 and 13 January 2026. Backward and forward citation chasing, Google Scholar update checks, and targeted searches for verifiable aggregation, provenance, auditability, and accountability were completed on 20 March 2026. The literature snapshot was frozen on that date.

Query fields followed each source's supported syntax, including title, abstract, keyword, and full-record fields. Eligible records were written in English, available as peer-reviewed publications or open preprints, and published between 2011 and the freeze date.

Table 1 reports the exact database-specific query strings and the number of records exported before deduplication.

Table 1. Reproducible database-search protocol and pre-deduplication record counts.

Database or Source	Date	Exact Query or Reconstruction Rule	Records	Record Handling
IEEE Xplore	10 January 2026	((“federated learning” AND (poisoning OR backdoor OR Byzantine OR “model poisoning”) AND (survey OR review OR taxonomy OR defense OR attack)) OR (“machine learning” AND “data poisoning” AND (survey OR review))) in all metadata	24	Export title, authors, year, venue, DOI, abstract, and keywords.
ACM Digital Library	10 January 2026	[[All: “federated learning”] AND [All: (poisoning OR backdoor OR Byzantine OR “secure aggregation”)] AND [All: (survey OR review OR taxonomy OR defense OR attack OR auditability OR accountability)]] OR [[All: “data poisoning”] AND [All: “machine learning”] AND [All: survey]]	21	Export citation metadata and DOI where available.
Scopus	11 January 2026	TITLE-ABS-KEY(“federated learning” AND (poisoning OR backdoor OR byzantine OR “model poisoning” OR “secure aggregation”) AND (survey OR review OR taxonomy OR defense OR attack OR accountability OR auditability OR traceability))	28	Export CSV metadata; retain computer-science, engineering, and security venues.
Web of Science	11 January 2026	TS=((“federated learning”) AND (poisoning OR backdoor OR Byzantine OR “secure aggregation” OR auditability OR traceability OR provenance) AND (survey OR review OR taxonomy OR defense OR attack))	19	Export full record and cited-reference metadata.
Elsevier ScienceDirect	11 January 2026	Title, abstract, keywords: “federated learning” AND poisoning AND (attack OR defense OR survey OR review), plus “data poisoning” AND “machine learning” AND survey	16	Export research and review articles with metadata.
SpringerLink	12 January 2026	“federated learning” AND (poisoning OR backdoor OR security OR privacy) AND (survey OR review OR taxonomy OR defense)	15	Export computer-science and engineering records.
arXiv	13 January 2026	all: “federated learning” AND all: (poisoning OR backdoor OR auditable OR traceable OR provenance OR “secure aggregation”) AND all: (survey OR taxonomy OR defense OR attack)	16	Retain open preprints when no peer-reviewed version was available by the freeze date.
Google Scholar	13 January 2026; update 20 March 2026	Advanced search phrases: “federated learning poisoning survey”, “federated learning backdoor defense survey”, “auditable federated learning”, “verifiable aggregation federated learning”, and “federated learning provenance”. The first 100 results for each phrase were screened by title and snippet.	8	Add only records not already exported from indexed databases.
Backward and forward citation chasing	20 March 2026	Reference lists and citing papers of the survey-like records, plus targeted checks for zero-knowledge aggregation, blockchain FL audit trails, provenance, and traceability.	6	Add only records satisfying the FL, poisoning, or accountability relevance rules.
Total before deduplication			153	

The main takeaway from Table 1 is that the search was distributed across multiple indexed databases, while Google Scholar and citation chasing contributed only additional non-duplicate records. This combination provided broad coverage without allowing supplementary searches to dominate the corpus.

2.2. Deduplication, Eligibility, and Screening Flow

Records were deduplicated before substantive screening. Exact matches were removed by DOI, arXiv identifier, and normalized title, in that order. Possible near matches were then checked manually using the title, first author, author set, year, venue, and abstract.

When a preprint and a peer-reviewed version described the same study, the peer-reviewed version was retained unless the preprint contained a materially newer taxonomy or a more complete technical appendix. In that case, the published version remained the primary citation, and the preprint was retained only as a supplementary source. Conference-to-journal extensions were treated as one record when their threat models, experiments, and conclusions were substantially the same. They were treated as separate records only when the later version added a distinct survey, taxonomy, or accountability mechanism.

Eligibility criteria were defined before screening. Records were included when they met at least one analytical need of the review: (I1) survey, review, systematization, or taxonomy work on poisoning, backdoors, adversarial ML, or FL security; (I2) primary FL studies on poisoning, Byzantine behavior, robust aggregation, secure aggregation, verifiable aggregation, provenance, traceability, auditability, or forensic readiness; (I3) governance, accountability, or standards sources needed to interpret audit evidence; or (I4) centralized poisoning work needed as a baseline for explaining what changes in FL.

Records were excluded for any of seven reasons: no training-time poisoning relevance (E1); no substantive FL relevance where FL was required (E2); only inference-time adversarial-example or backdoor discussion without FL accountability relevance (E3); an overly broad application-only scope with no transferable taxonomy or mechanism (E4); duplicate or superseded content (E5); no comparative or analytical value for the research questions (E6); or inaccessible or non-substantive content, such as abstracts, slides, posters, or very short non-archival items (E7).

Table 2 summarizes the PRISMA-style record flow from the initial search pool to the survey-like and core-survey sets. Counts are reported as screening-log counts rather than estimates.

Table 2. PRISMA-style screening flow and study-selection logic.

Screening Step	Count	Action	Result Used in This Manuscript
Database and citation records identified	153	Export metadata listed in Table 1.	Pre-deduplication search pool.
Duplicates and superseded versions removed	33	Remove duplicates by DOI, arXiv identifier, normalized title, and version status using the rules above.	120 unique candidate records.
Title and abstract screening	120	Apply E1–E7 to titles, abstracts, venues, and snippets.	42 records excluded; 78 records advanced to full-text screening.
Full-text eligibility screening	78	Assess full texts or extended preprints for relevance to RQ1–RQ4.	10 additional records excluded; 68 records retained for coding.
Role assignment within retained records	68	Classify retained records by function.	17 survey-like works and 51 primary, framework, governance, standards, or evidence-supporting studies.
Core-survey selection	17	Apply the core-survey definition in the text and the documented screening decisions.	8 core surveys selected for direct novelty comparison; 9 survey-like works retained as supporting or exclusion-context records.
Final synthesis	68	Code threat model, defense family, privacy setting, trust assumption, evidence artifact, evaluation setting, and deployment constraint.	Evidence base for the subsequent survey-comparison, attack-taxonomy, accountability, and defense-scenario analyses.

Table 2 shows that 68 of the 153 initially identified records were retained for synthesis and that the survey-comparison set was narrowed further to eight core surveys, providing a transparent basis for the manuscript's bounded novelty claims.

Table 3 reports the exclusion-code counts used in the screening log. The counts sum to 85 excluded or removed records: 33 duplicate or superseded records removed during deduplication and 52 records excluded during title/abstract or full-text screening.

A survey-like work was defined as a survey, review, systematization of knowledge, taxonomy, tutorial-style synthesis, or systematic review that compared multiple papers or defense families, rather than proposing only a single method. A survey-like work was classified as a core survey when it met one or more of the following criteria:

- it directly surveyed FL poisoning, FL security, or adversarial FL;
- it provided a taxonomy needed to position this manuscript's novelty;
- it compared multiple defense families relevant to poisoning, robustness, privacy, or accountability;
- it substantially overlapped with this manuscript's scope;

- it had not been superseded by a more recent or more directly relevant survey;
- it helped establish the baseline for comparing this paper's accountability-oriented contribution.

Table 3. Exclusion-code counts for the screening flow.

Code	Exclusion Reason	Records Excluded or Removed
E1	No training-time poisoning relevance	12
E2	No substantive FL relevance	10
E3	Only inference-time adversarial-example or backdoor discussion, with no FL-accountability relevance	8
E4	Application-only scope with no transferable taxonomy or mechanism	9
E5	Duplicate or superseded content	33
E6	No comparative or analytical value for the research questions	8
E7	Inaccessible or non-substantive content, such as abstracts, slides, posters, or very short non-archival items	5
Total		85

Seventeen records met the survey-like definition. Eight were selected as core surveys because they were the closest comparators to this manuscript's scope under the criteria above. The remaining nine survey-like works were retained as supporting or background evidence, but they were not treated as core novelty comparators because they were too broad, too narrow, insufficiently FL-specific, or superseded by closer surveys.

2.3. Coding, Appraisal, and Claim Calibration

To avoid overstating what a scoping review can support, claims were calibrated through structured coding and appraisal. Each included survey and primary study was coded for its threat model, defense family, privacy setting, trust assumption, evidence artifact, evaluation setting, and deployment constraint. The appraisal did not use a numerical quality score or exclude or rank studies. Instead, it assessed how strongly each study could support a given synthesis claim. High support required an explicit threat model, evaluation setting, and defense or evidence artifact. Medium support required clear conceptual or empirical relevance but incomplete deployment detail. Low support was used only for background framing. Table 4 maps the main claim types to the evidence used and the limits placed on interpretation.

The key implication of Table 4 is that different claims require different evidence strengths: survey-positioning evidence supports bounded novelty claims, whereas practical recommendations are presented as design implications rather than universal conclusions.

Table 4. Coding, appraisal, and claim calibration scheme.

Claim Type	Coding Fields	Evidence Source	Permitted Interpretation
Novelty relative to surveys	Scope, taxonomy depth, FL specificity, accountability coverage	Survey-screening and core-survey comparison records	Bounded claim about the screened survey-like corpus, not all possible surveys.
Attack and defense synthesis	Threat model, attacker role, manipulation target, defense family	Representative primary studies and defense-family comparisons	Qualitative synthesis of recurring patterns, not pooled effectiveness estimates.
Accountability analysis	Observability, attribution, audit artifact, trust boundary, replayability	AIT indicator definitions and accountability-oriented studies	Comparison of accountability profiles under stated assumptions.
Practical design guidance	Deployment setting, overhead, privacy constraint, verification mechanism	Cryptographic-accountability studies and scenario-based defense comparisons	Design implications for future frameworks, stated as recommendations rather than universal requirements.

Screening and coding were performed using a predefined coding sheet based on the inclusion/exclusion criteria and coding fields reported in this section. Ambiguous records were revisited after full-text review and resolved by applying the stated criteria consistently. A second-pass consistency check ensured that similar records received consistent exclusion codes, role assignments, and claim-support labels.

Table 5 links the manuscript's main synthesis claims to the specific evidence tables used to support them. This table is intended to make the boundary of each claim explicit: the claims are traceable to the screened and coded corpus, not to an exhaustive census of all FL literature.

Table 5. Claim-to-evidence traceability.

Manuscript Claim	Evidence Table or Source	Evidence Basis	Claim Boundary
Prior surveys provide limited explicit coverage of forensic accountability	Survey-screening log and core-survey coverage matrix	Eight core surveys coded for auditability, forensic accountability, compromised-server treatment, and custody/tamper-evidence coverage	Limited to the selected core surveys and the survey-like screening log.
Poisoning defenses are often evaluated using robustness metrics rather than audit evidence	FL-attack taxonomy, AIT application, and defense-scenario evidence	Primary defense studies coded by threat model, evidence artifact, accountability indicators, and scenario-dependent assumptions	Limited to the coded primary studies and representative defense families.
Compromised-server assumptions remain underdeveloped in the reviewed corpus	Core-survey coverage, threat-model coding, and defense-scenario evidence	Core surveys and accountability-oriented studies coded for server-side trust assumptions, verifiable aggregation, and audit evidence	Limited to the reviewed corpus and explicit trust-boundary coding.
This work contributes an accountability-oriented comparison lens	Coding-appraisal, claim-traceability, novelty-analysis, and AIT-indicator schemes	Coding and point-by-point novelty analysis show how this survey organizes poisoning defenses around evidence, attribution, auditability, and server trust	An analytical contribution, not an exhaustive coverage claim.

Throughout the manuscript, claims about what prior surveys in the screened set omit, what reviewed defenses lack, or what future frameworks should prioritize refer to the screened and coded corpus rather than to every possible publication in FL security. Broad field-level language is therefore avoided unless it is directly supported by the survey-positioning log, the coded primary-study tables, or the evidence-trace tables above.

2.4. Survey-like Screening Log and Core-Survey Comparison

Table 6 lists the 17 survey-like records used to derive the final core-survey set. The table records each work's title, authors, year, source, category, decision, and rationale. It is included in the main paper because this subset directly supports the novelty claim and allows reviewers to reconstruct the selection of eight core surveys from the 17 survey-like records.

The takeaway from Table 6 is that eight of the 17 survey-like records were sufficiently close to the manuscript's scope to serve as core novelty comparators; the remainder were retained as supporting context rather than discarded.

The same selection process was used to avoid overstating novelty. Table 7 positions this manuscript against the eight core surveys and shows that centralized ML poisoning surveys [3,9], broad adversarial-ML taxonomies [10], general FL security surveys [11,12], FL poisoning-focused reviews [13,14], and FL model-poisoning defense syntheses [15] are already well developed. The contribution of the present review is, therefore, not to duplicate those catalogs. Instead, it asks a different synthesis question: after a poisoning incident in FL, which defenses preserve enough evidence to reconstruct what happened, who acted, when, which trust boundary was crossed, and whether the server behaved honestly?

Table 8 compares topic coverage across the eight core surveys. "C" means that the topic is central to the survey, "P" means partial or secondary coverage, and "–" means that the topic is absent or only incidental. The matrix supports a bounded novelty claim: within the screened survey set, the closest surveys discuss poisoning, robustness, privacy,

or trustworthiness, but they do not jointly organize FL poisoning defenses by actor attribution, server-action verification, chain of custody, tamper evidence, and post-incident reconstruction value.

Table 6. Screening log for the 17 survey-like records used to derive the 8 core surveys.

Record Title	Authors/Year	Source Found	Type	Decision	Reason
A taxonomy and survey of attacks against machine learning [10]	Pitropakis et al., 2019	ScienceDirect; Scopus	Adversarial-ML survey	Core	Foundational adversarial-ML baseline for attack taxonomy and novelty positioning.
A comprehensive survey on poisoning attacks and countermeasures in machine learning [3]	Tian et al., 2022	ACM DL; Scopus	ML poisoning survey	Core	Centralized poisoning survey used to clarify what changes in FL.
Wild patterns reloaded: A survey of ML security against training data poisoning [9]	Cinà et al., 2023	ACM DL; Scopus	Training-data poisoning survey	Core	Detailed centralized training-data poisoning taxonomy for comparison.
A detailed survey on federated learning attacks and defenses [11]	Sikandar et al., 2023	Scopus; Google Scholar	FL security survey	Core	General FL attack-defense survey used to map threat surfaces.
A survey on federated learning poisoning attacks and defenses [13]	Liang et al., 2023	arXiv; Google Scholar	FL poisoning survey	Core	Direct reference point for FL poisoning attacks and defenses.
Poisoning attacks in federated learning: A survey [14]	Xia et al., 2023	IEEE Xplore; Scopus	FL poisoning survey	Core	Peer-reviewed FL poisoning survey retained for attack-coverage comparison.
Threats and defenses in the federated learning life cycle [12]	Li et al., 2025	Google Scholar; citation chasing	FL life-cycle survey	Core	Life-cycle reference for privacy, robustness, and security assumptions.
Defense strategies toward model poisoning attacks in federated learning: A survey [15]	Wang et al., 2022	IEEE Xplore; arXiv	FL model-poisoning defense survey	Core	Defense-focused reference on FL model poisoning.
Poisoning attacks and defenses on artificial intelligence: A survey [23]	Ramirez et al., 2022	arXiv; Google Scholar	ML poisoning survey	Non-core	Relevant ML poisoning survey that overlaps with the retained centralized poisoning baselines and is not FL-specific.
Data poisoning attacks in the training phase of ML models [24]	Srivastava et al., 2024	Google Scholar	ML poisoning review	Non-core	Covers training-time poisoning but is not substantively FL-focused.
Backdoor attacks and countermeasures on deep learning [25]	Gao et al., 2020	arXiv; Google Scholar	Backdoor review	Non-core	Relevant backdoor review, but it does not analyze FL accountability or aggregation.
Backdoor learning survey [26]	Li et al., 2022	Scopus; Google Scholar	Backdoor survey	Non-core	Wide-ranging backdoor survey with limited FL-specific accountability treatment.
Threats to training: poisoning attacks and defenses [27]	Wang et al., 2022	ACM DL; Scopus	ML poisoning survey	Non-core	Overlaps with centralized poisoning surveys already retained in [3,9].
Robust deep learning using hardware-software co-design [28]	Zhang et al., 2023	ACM DL	Robustness systematization	Non-core	Broader hardware-software robustness scope than FL poisoning accountability.
Trustworthy federated learning: privacy, security, and beyond [29]	Chen et al., 2025	SpringerLink; Scopus	Trustworthy-FL survey	Non-core	Trustworthy-FL synthesis in which poisoning is one topic among many.

Table 6. *Cont.*

Record Title	Authors/Year	Source Found	Type	Decision	Reason
Robust federated learning under adversarial attacks [30]	Xia and Cheng, 2023	ACM DL; Scopus	Robust-FL survey	Non-core	Robustness-oriented survey retained as supporting evidence rather than a core novelty baseline.
Statistical poisoning detection in FL [31]	Rahman and Debnath, 2024	ScienceDirect; Scopus	Detection survey	Non-core	Narrow detection focus, useful for evidence discussion but insufficient for cross-survey positioning.

Table 7. Positioning this survey relative to the eight core surveys.

Survey	Primary Scope	Main Strength	Gap Relative to This Work
Broad adversarial-ML taxonomy survey [10]	Adversarial ML taxonomy	Foundational attack taxonomy across ML settings	Provides little FL-specific poisoning detail and no accountability- or audit-evidence perspective.
Poisoning-in-ML survey [3]	Poisoning in ML	Strong overview of poisoning attacks and countermeasures in centralized ML	Does not foreground FL-specific trust boundaries, server-side threats, or forensic readiness.
Training-data poisoning survey [9]	Training-data poisoning in ML	Detailed synthesis of poisoning threat models and defenses	Focuses on poisoning taxonomy rather than FL accountability, untrusted aggregation, and evidence requirements.
FL attack-defense survey [11]	FL attacks and defenses broadly	Security overview of FL attack surfaces	Covers FL security as a whole, but not poisoning-specific accountability or auditability.
FL poisoning attack-defense survey [13]	FL poisoning attacks and defenses	FL-oriented catalog of attack and defense families	Places less emphasis on accountability under untrusted-server assumptions.
FL poisoning survey [14]	FL poisoning	Peer-reviewed FL-focused synthesis with updated attack examples	Gives limited attention to traceability, verifiable evidence, and malicious-server accountability.
FL life-cycle threat survey [12]	FL life-cycle threats and defenses	Life-cycle perspective on FL security and privacy	Treats poisoning as one life-cycle issue among many, with little centralized baseline comparison or accountability taxonomy.
FL model-poisoning defense survey [15]	FL model-poisoning defenses	Defense-focused classification of local-update evaluation and global aggregation methods	Emphasizes defenses more than traceability, audit evidence, or malicious-server scenarios.
This survey	Accountability-oriented FL poisoning	Shows how moving from centralized poisoning to FL reduces observability, weakens attribution, introduces server-side and collusion risks, and increases evidence requirements	Uses representative literature and explicit screening for accountability-oriented comparison rather than exhaustive or pooled evidence coverage.

Table 8. Topic coverage matrix for the eight core surveys used in the novelty claim. C = central coverage; P = partial or secondary coverage; – = absent or incidental.

Core Survey	FL Priv.	Poison/ Backdoor	Robust Agg.	Block./ Logs	Audit	Forensic Acc.	Comp. Server	Custody/ Tamper	Implication for Novelty
Ref. [10] Broad adversarial-ML taxonomy	–	P	–	–	–	–	–	–	Baseline taxonomy; not FL- or evidence-oriented.
Ref. [3] ML poisoning survey	–	C	P	–	–	–	–	–	Strong poisoning baseline without FL accountability boundaries.
Ref. [9] Training-data poisoning survey	–	C	–	–	–	–	–	–	Detailed poisoning taxonomy focused on centralized data pipelines.
Ref. [11] FL attack-defense survey	P	P	P	–	–	–	P	–	General FL security context with limited audit-evidence analysis.
Ref. [13] FL poisoning attack-defense survey	P	C	P	–	–	–	P	–	Direct poisoning reference with limited accountability or custody treatment.
Ref. [14] FL poisoning survey	P	C	P	–	–	–	P	–	Updated FL poisoning synthesis with little verifiable-evidence framing.
Ref. [12] FL life-cycle threat survey	C	P	P	P	P	–	P	–	Life-cycle breadth; poisoning accountability is not the organizing lens.
Ref. [15] FL model-poisoning defense survey	P	C	C	–	–	–	–	–	Defense-focused; model-update evaluation and robust aggregation remain separate from broader evidence requirements.
This survey	P	C	C	P	C	C	C	C	Combines poisoning, FL trust boundaries, audit evidence, and accountability indicators.

Table 9 states the novelty claim against the closest overlapping survey streams and mechanism-focused literature. This manuscript does not claim that secure aggregation, provenance records, verifiable computation, blockchain anchoring, or audit logs are individually new. The contribution is the accountability-oriented interpretation of these mechanisms: each defense is assessed by the evidence it preserves, the actor or server action it can help reconstruct, the trust boundary it assumes, and the forensic or audit claim it can support after a poisoning incident.

AIT's novelty is therefore not the introduction of a single new defense primitive. Rather, it provides a synthesis lens for comparing FL poisoning defenses by their accountability consequences. It connects threat-model assumptions, evidence artifacts, verification mechanisms, and post-incident audit needs in one comparison structure. This allows defenses that appear similar under robustness or privacy metrics to be distinguished by whether they support actor attribution, server-action verification, tamper-evident reconstruction, and chain-of-custody preservation.

Table 9. Granular novelty analysis relative to overlapping literature.

Closest Literature Stream	Established Contribution	Remaining Accountability Gap	AIT Interpretation in This Survey
Centralized poisoning and adversarial-ML surveys [3,9,10]	Taxonomies of poisoning mechanisms, poisoned samples, provenance, training logs, and centralized robustness.	Usually assume one controlled pipeline, with limited treatment of FL server behavior, collusion, or cross-round evidence.	Uses the centralized baseline to expose FL-specific losses in observability and attribution and the added need to verify server actions.
Trustworthy-FL and life-cycle surveys [12,29]	Broad treatment of privacy, security, robustness, fairness, transparency, and life-cycle controls.	Do not usually connect actor, round, server action, retained artifact, and audit claim in one poisoning-specific analysis.	Narrows trustworthiness to post-incident accountability: what remains observable, attributable, verifiable, and reviewable after poisoning.
General FL attack-defense and robust-FL surveys [11,30]	FL threat surfaces, Byzantine and Sybil behavior, non-IID data, and aggregation performance.	Robustness is commonly assessed as attack resistance rather than as evidence production for forensic reconstruction.	Separates defenses that reduce attack impact from those that preserve replayable evidence, custody, and server-action verification.
FL poisoning surveys [13,14]	Detailed catalogs of data poisoning, model poisoning, backdoors, malicious clients, and defense effectiveness.	Accountability artifacts and post-incident reconstruction remain secondary to accuracy, attack success, and detection metrics.	Reinterprets defenses through observability, attribution, aggregation verifiability, tamper evidence, replayability, and audit overhead.
FL model-poisoning defense survey [15]	Defenses based on evaluating local model updates and applying robust global aggregation.	Does not fully cover malicious servers, collusion, or evidence across trust boundaries.	Extends model-update integrity to evidence integrity by asking which client, round, server, proof, and provenance artifacts remain usable after an incident.
Secure and verifiable aggregation work [18–21,32–39]	Commitments, authenticated shares, proofs, receipts, and verifiable or malicious-secure aggregation.	Proves bounded computation or aggregation properties but does not independently identify poisoned data, triggers, or responsible actors.	Codes what each proof establishes, what it omits, which trust boundary it crosses, and which detection evidence must complement it.
Blockchain, provenance, and accountable-FL work [17,22,40–42]	Ledger anchors, signatures, participant selection, provenance records, and model-version histories.	May not address pre-anchor tampering, record completeness, or the poisoning-specific meaning of retained evidence.	Interprets these artifacts as chain-of-custody evidence for attribution, server-action reconstruction, tamper-evident replay, and audit timelines.
General AI auditability and accountability work [7,8,43–46]	Governance principles, audit access, traceability, responsibility, and standards alignment.	Lacks an FL-poisoning mapping from technical mechanisms to incident evidence and post-incident claims.	Translates accountability principles into coding fields for actor, round, server action, artifact, verification, replayability, and audit claim.

Table 10 separates broad trustworthy-FL framing from the narrower post-incident accountability question used in this manuscript. Recent trustworthy-FL work addresses important goals such as privacy, robustness, fairness, transparency, and security [12,29]. AIT does not replace those goals; it asks what can be proven or reconstructed after a poisoning incident.

Table 11 makes the threat-model contribution explicit. The point is not that malicious clients, Byzantine behavior, collusion, or malicious servers are unknown. Rather, this manuscript reorganizes these threats around the accountability question: which defenses preserve enough evidence to reconstruct what happened, who acted, when, and whether the server behaved honestly?

Table 10. Broad trustworthy-FL framing versus AIT accountability framing.

Trustworthy-FL Framing	AIT Accountability Framing
Asks whether the FL system is secure, private, robust, fair, transparent, or broadly trustworthy.	Asks what evidence remains after an attack and whether that evidence can support incident reconstruction.
Emphasizes prevention, resilience, compliance posture, or performance under attack.	Emphasizes attribution, auditability, reconstruction, verification, and chain-of-custody after a poisoning incident.
Measures defenses by accuracy, attack success rate (ASR), robustness, privacy cost, convergence, or communication overhead.	Measures defenses by evidence preservation, actor traceability, server-side verifiability, tamper evidence, forensic replayability, and audit overhead.
Often assumes a trusted or semi-trusted server when discussing coordination, aggregation, or life-cycle controls.	Explicitly codes server-side trust assumptions, malicious-aggregator risks, server filtering, delayed anchoring, pre-anchor tampering, and client–server collusion.
Treats logs, provenance, verification, or blockchain mechanisms as trustworthiness or transparency mechanisms.	Treats those mechanisms as forensic accountability artifacts and asks what audit claim each artifact can support.

Table 11. Threat-model coverage distinguishing AIT from adjacent survey streams. C = central coverage; P = partial or assumption-specific coverage; – = absent or incidental.

Literature Stream	Mal. Clients	Poison/ Backdoor	Byz.	Collus.	Mal./ Semi-Trusted Server	Server Filtering	Delayed Anchor/ Pre-Log Tamper	Post-Incident Audit	AIT Accountability Interpretation
Trustworthy-FL and FL life-cycle surveys [12,29]	P	P	P	P	P	P	–	P	Broad trust goals are reframed as incident-evidence questions: what artifacts remain, and who can verify them?
General FL attack-defense and robust-FL surveys [11,30]	C	P	C	P	P	P	–	–	Robustness assumptions are linked to reconstruction or challenge of aggregation decisions.
FL poisoning surveys [13,14]	C	C	P	P	P	P	–	P	Poisoning mechanisms are mapped to attribution granularity, replayability, and server-action evidence.
FL model-poisoning defense survey [15]	C	C	C	P	–	P	–	–	Model-update defenses are extended to server-side and custody-sensitive evidence needs.
Secure/verifiable aggregation studies [18–21,32–39]	P	P	P	P	C	P	–	P	Proofs and receipts are treated as bounded evidence: they verify aggregation properties but not all poisoning semantics.
Blockchain, provenance, and accountable-FL work [17,22,40–42]	P	P	–	P	P	P	P	C	Ledger and provenance artifacts are evaluated for chain of custody, delayed-anchoring risk, and audit replay value.
This survey / AIT	C	C	C	C	C	C	C	C	Jointly codes attacker role, server trust, evidence artifact, tamper evidence, and post-incident audit claim.

Table 12 makes the evidence-artifact contribution explicit. AIT treats these artifacts not merely as system logs or implementation details, but as forensic accountability artifacts whose value depends on what they can prove, preserve, or replay after a poisoning incident.

Unlike prior poisoning or trustworthy-FL surveys that primarily compare defenses by robustness, privacy, accuracy, attack success rate, convergence, or communication overhead, this manuscript also asks whether each defense preserves usable artifacts for attribution, audit, replay, and post-incident verification. Table 13 summarizes the resulting accountability trade-offs.

The central takeaway from Table 13 is that stronger privacy or robustness does not automatically yield stronger accountability; evidence retention, verifiability, and replayability must be evaluated separately from attack resistance.

Table 12. Evidence artifacts organized by AIT for forensic accountability.

Evidence Artifact	Typical Prior Interpretation	AIT Forensic-Accountability Interpretation	Post-Incident Question Supported
Client update hashes and signed client updates	Integrity checks or authenticated submission metadata	Binds an update to an actor, round, and submitted artifact, if identity management is reliable	Which client submitted the update, and was it changed after submission?
Server-signed global model hashes and model-version history	Model release or reproducibility metadata	Records the global model broadcast, enabling checks for model forking or inconsistent server behavior	Did the server distribute the same model and preserve a verifiable model lineage?
Round-level logs, timestamped receipts, and chain-of-custody records	Operational logging or compliance records	Establishes a timeline linking participants, update receipt, aggregation, filtering, and release decisions	When did the suspicious event occur, and which artifacts were in custody at each stage?
Aggregation transcripts, secure aggregation proofs, and zero-knowledge proofs	Privacy-preserving verification or correctness proof	Provides bounded verification showing that a stated aggregation or admissibility computation was performed	Was the aggregation correct under the stated protocol, and what does the proof not reveal about poisoned data or triggers?
Witness committee records and validation or filtering decisions	Distributed validation, committee approval, or filtering metadata	Shows which external or client-side actors approved, rejected, or challenged updates and under which rule	Who participated in verification, and can the filtering decision be independently audited?
Anomaly scores and poisoning indicators	Detection outputs or defense metrics	Become interpretable only when linked to clients, rounds, thresholds, and retained update/model artifacts	Why was an update flagged, and can the explanation be replayed or challenged?
Ledger anchors and delayed anchoring records	Tamper-evident logging or blockchain transparency	Provides custody evidence only after anchoring; AIT separately codes pre-anchor tampering and delayed-write risk	Was the record anchored before modification was possible, and what remains unverifiable before anchoring?
Provenance records and replayable audit bundles	Reproducibility, transparency, or data/model lineage	Packages hashes, signatures, logs, proofs, model versions, and detection outputs for auditor use	Can an independent auditor reconstruct the poisoning event without raw client data?

Table 13. Defense trade-offs interpreted through the AIT accountability lens.

Defense Trade-Off	Common Technical Interpretation	AIT Accountability Interpretation
Robustness vs. auditability	Robust aggregation can reduce attack impact and improve accuracy under adversarial updates.	A robust decision is more auditable when discarded, down-weighted, or selected updates remain traceable and replayable.
Privacy vs. attribution	Secure aggregation and privacy controls protect clients from update inspection.	Privacy can obscure per-client visibility; AIT asks which privacy-preserving artifacts still support attribution or bounded verification.
Secure aggregation vs. observability	Secure aggregation improves confidentiality but hides individual gradients.	Hidden updates weaken anomaly forensics unless accompanied by commitments, admissibility proofs, receipts, or committee records.
Differential privacy vs. evidence usefulness	Differential privacy limits leakage and changes update statistics.	Added noise can reduce forensic signal quality, so AIT separates privacy budget from evidence value and replayability.
Blockchain anchoring vs. latency/overhead	Ledgers provide tamper evidence with storage, consensus, and communication costs.	AIT asks whether anchoring is timely enough to prevent pre-anchor tampering and whether ledger entries support a complete incident timeline.
Anomaly detection vs. explainability	Detectors are evaluated by detection accuracy, false positives, and ASR reduction.	AIT asks whether anomaly scores are interpretable, linked to retained artifacts, and useful as evidence rather than only as filtering signals.
Server trust vs. verifiability	Semi-trusted servers simplify coordination and aggregation.	AIT codes whether server sampling, filtering, aggregation, and broadcast actions are externally verifiable or merely assumed honest.
Tamper evidence vs. deployment complexity	Signatures, logs, proofs, and ledgers increase overhead and engineering burden.	AIT treats this overhead as an accountability cost and compares it with the audit value of chain-of-custody preservation.

3. Poisoning Attacks and Accountability Taxonomy

3.1. Baseline Poisoning Taxonomy

As ML models become more prevalent across application domains, attacks on the training phase have intensified [3,47]. In poisoning attacks, adversaries inject malicious data into training datasets or manipulate model updates to slow convergence, degrade accuracy, or implant targeted behaviors [24,48]. These threats were first studied in centralized learning settings, where data and optimization are managed within a single training pipeline.

FL preserves many of the same technical attack goals, but it changes the operational context: participants train locally, the server aggregates remote updates, and raw data remain hidden [49]. As a result, FL inherits classical poisoning mechanisms while introducing additional trust boundaries, concealment opportunities, and attribution problems [50]. Most importantly, FL expands the threat surface beyond malicious clients alone. An un-

trusted or compromised server can manipulate aggregation, alter client weights, fork models, or collude with clients to implant backdoors.

These server-side possibilities change what evidence is available, who can verify it, and how responsibility can be assigned after an incident. Figure 1 summarizes the general poisoning process within the ML training pipeline. Building on prior centralized poisoning [3], adversarial-ML [10], FL-oriented [25], and backdoor [51] taxonomies, this section organizes poisoning attacks along three dimensions—attacker knowledge, manipulation target, and intent—that later support the accountability-oriented extension.

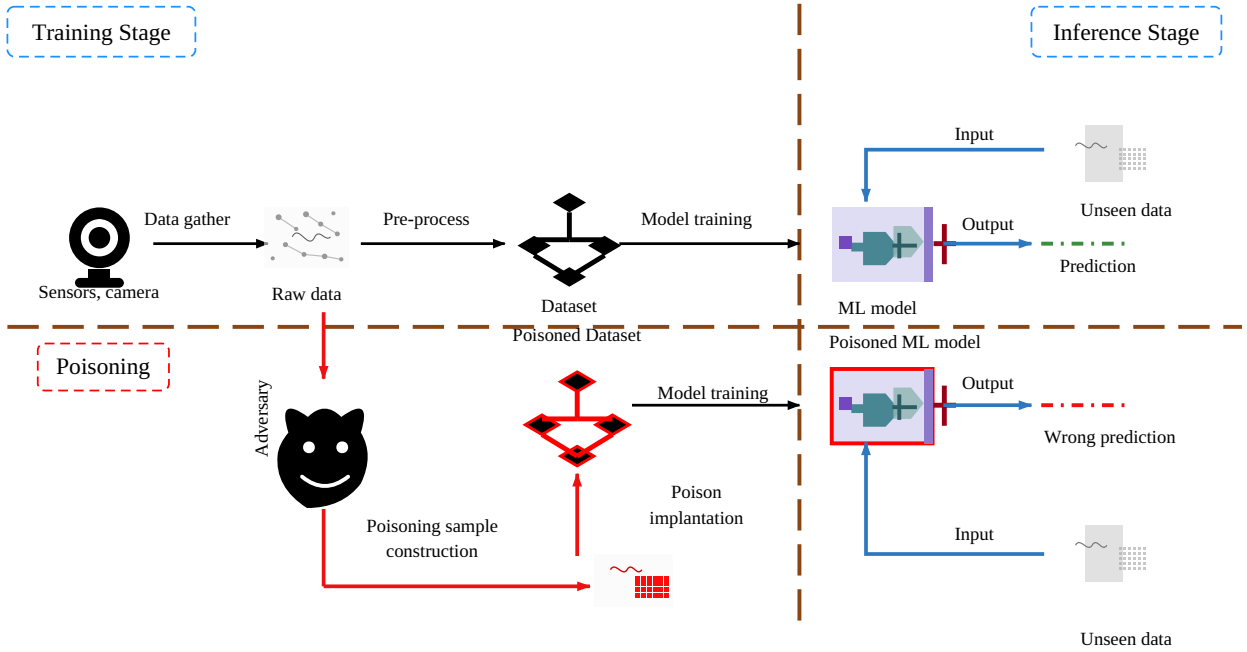

Figure 1. General poisoning attack in the ML training pipeline, adapted from [3].

3.1.1. Taxonomy Overview

Poisoning attacks can be categorized along three orthogonal dimensions, each capturing a distinct aspect of adversarial capability:

1. Attacker knowledge: the degree of visibility into model internals or the training pipeline (white-box, gray-box, or black-box) [26].
2. Manipulation target: the component the adversary modifies (training data, model parameters, or training dynamics).
3. Attack intent: whether the goal is the untargeted degradation of overall performance or the targeted manipulation of specific outputs or behaviors.

Table 14 summarizes the taxonomy by attacker access, manipulation target, attack intent, representative mechanism, and supporting references. White-box adversaries exert maximal control but may also be more detectable through forensic logging or gradient fingerprints, whereas gray-box and black-box adversaries rely on transferability or surrogate modeling to achieve stealth at the cost of fine-grained control. These distinctions matter when designing accountability-oriented defenses, such as verifiable aggregation and provenance-based auditing.

Table 14. Unified typology of poisoning attacks.

Access	Target	Intent	Example Mechanism	Key Refs.
White-box	Data/model	Untargeted or targeted	Gradient manipulation, label flipping, and model replacement	[2,11,16,52]
Gray-box	Data/model	Untargeted or targeted	Limited-knowledge poisoning and adaptive gradient updates (e.g., AgrEvader)	[53,54]
Black-box	Data	Untargeted or targeted	Query-based or transfer poisoning and shadow-model training	[27,55]
White-box	Backdoor	Targeted	Trigger embedding and model-level implants	[13,28,56,57]

3.1.2. Data Poisoning Attacks

Data poisoning introduces malicious or mislabeled samples into the training set to corrupt the learned decision boundary. Two primary variants are commonly distinguished:

- Clean-label poisoning: the attacker does not modify labels but crafts adversarial inputs that collide with target-class features [56,58]. GAN-generated poisons can operate under non-IID conditions and without full knowledge of the victim model [14].
- Dirty-label poisoning: both inputs and labels are modified to implant malicious behavior or enable targeted misclassification. Generative poisoning shows how such behavior can be synthesized in FL [59], while robustness and detection studies show why these attacks remain difficult to measure and filter consistently [60,61]. Label flipping is a simple example: static label flipping (SLF) directly changes labels, whereas dynamic label flipping (DLF) targets feature-overlapping samples to increase stealth [58].

Recent data-poisoning defenses continue to refine how these variants are detected and measured in neural-network and FL settings [62,63].

3.1.3. Model Poisoning Attacks

Model poisoning manipulates gradients or parameters to distort global learning or implant persistent malicious behavior. Recent work frames poisoning as a risk-management problem [52], shows how model replacement can backdoor FL [64], and demonstrates distributed backdoor attacks across coordinated clients [65]. This threat is especially important in FL, where compromised clients can submit arbitrary updates to the server. Common mechanisms include the following:

- Model replacement: adversaries scale updates to overwrite the global model in a single round [52].
- Gradient ascent: adversaries send updates in the opposite optimization direction to amplify divergence [66,67].
- Stealth poisoning: adversaries craft malicious updates that mimic benign statistics to evade anomaly detection [66].

Together, data- and model-level poisoning show why provenance and audit trails matter: the server often sees only an update, not the local decision process that produced it.

3.1.4. Backdoor Attacks

Backdoor poisoning implants hidden triggers that activate during inference only when a specific pattern or condition is present [13,56]. Backdoors may operate at two levels:

- Data level: imperceptible patches, pixel patterns, or text perturbations embedded into poisoned samples [68,69].

- Model level: direct manipulation of model weights or aggregation procedures to implant latent malicious behavior [28,57]. Effective backdoors are both stealthy and semantically valid, making them particularly dangerous in FL, where decentralized updates hinder attribution and forensic reconstruction.

3.1.5. Key Takeaways

Data, model, and backdoor poisoning form a continuum: adversaries may corrupt inputs, manipulate gradients, or implant persistent triggers. In centralized learning, these threats often arise from contaminated datasets or insider access [70]. In FL, the same mechanisms are compounded by untrusted servers, heterogeneous clients, and cross-round collusion [50,52].

From an accountability standpoint, the common weakness is limited evidence. Few defenses provide update traceability, cryptographic validation of model lineage, or reliable attribution to specific clients or servers. Recent surveys [52,71] likewise show that robustness and privacy are often discussed separately from accountability. The following sections therefore use this taxonomy to compare centralized and federated poisoning and then evaluate defenses by their ability to produce audit-ready evidence.

3.2. Poisoning Attacks in Centralized Learning

Table 15 summarizes dominant poisoning mechanisms in centralized learning, organized by attack strategy and intended impact. Although these approaches differ in subtlety and computational sophistication, they share the assumption that attackers can inspect or directly modify the training dataset or model state.

Observations and Auditability Differences

Centralized poisoning attacks reveal structural properties that distinguish them from attacks in FL. Recent work from 2022 to 2025 reinforces these observations:

- Full data visibility enables high-impact, optimization-driven poisoning. In centralized settings, adversaries can directly inspect or manipulate the entire training dataset. This access supports large-scale integrity attacks discussed in FL-oriented surveys [12], adjacent model-inversion risks [24], regression-focused poisoning attacks [70], and adaptive manipulations that are difficult to realize when only partial data visibility is available.
- The stealth-versus-impact trade-off is more pronounced. Recent clean-label and influence-based attacks achieve high stealth but smaller global impact [14,28], whereas aggressive gradient-based and model-update poisons cause substantial degradation but are easier to detect [15,23]. Centralized settings allow adversaries to manage this trade-off more effectively because they retain full control over the data pipeline.
- Centralized logging supports stronger auditability and traceability. Centralized workflows benefit from reproducible pipelines, dataset versioning, and unified audit logs [3,70]. Forensic tools such as gradient fingerprinting and lineage tracing allow investigators to reconstruct poisoning events [25,71]. In contrast, FL lacks global visibility, and poisoned updates often blend with benign ones [50,52].
- Adversarial diversity is more limited than in FL. Centralized systems involve a single training pipeline, making multi-party collusion and multi-round adaptive poisoning relatively rare. Recent FL attack studies demonstrate broader attack surfaces in general FL settings [11], poisoning-specific FL settings [13,14], and provenance-sensitive or transparency-oriented FL workflows [22].

Table 15. Common poisoning attacks in centralized ML.

Attack Type	Description/Mechanism	Impact/Observations
Gradient-based poisoning	Crafts poisoned samples using bilevel optimization to distort gradient updates [70].	Can degrade accuracy by up to 60% and enable precise targeted misclassification or divergence.
Label flipping	Randomly or selectively alters labels to mislead training [23,72].	Reduces accuracy by 20–40%, with effects visible in class-confusion patterns.
Clean-label poisoning	Inserts benign-looking samples that collide with target-class features [56].	Hard to detect because it preserves overall accuracy while enabling targeted misclassification.
Backdoor poisoning	Embeds imperceptible triggers into poisoned inputs [73].	Produces near-100% targeted misclassification when the trigger is present.
Availability attacks	Corrupts data or features to prevent convergence [74].	Leads to accuracy collapse or unstable training failure.
Influence-function attacks	Identifies and poisons influential training points using Hessian-vector analysis [75].	Enables stealthy manipulation with a small poisoning budget.
Regression poisoning	Manipulates continuous-valued regression data to maximize prediction error [70].	Doubles or triples MSE and affects downstream risk models.
Domain-specific poisoning	Uses contextualized attacks targeting healthcare or ICS systems [76].	Causes misdiagnosis or anomaly suppression in 85–90% of industrial monitoring cases.

- Provenance and accountability pathways are clearer. Centralized ML benefits from complete ownership over data collection, preprocessing, and training workflows. Accountability tools—dataset fingerprinting, secure provenance tracking, and auditable pipelines—are easier to enforce [77,78]. In FL, the absence of unified provenance and verifiable aggregation complicates attribution of malicious behavior [12,28]. These observations show that centralized poisoning attacks remain highly damaging, yet comparatively easier to audit and attribute than their FL counterparts. This motivates a separate analysis of poisoning in FL, where untrusted participants, opaque aggregation, and cross-round interactions create a broader and more complex threat surface.

3.3. Poisoning Attacks in FL

FL introduces structural vulnerabilities that do not arise in centralized ML. Adversaries can exploit client–server communication and heterogeneous local data in general FL attack settings [11,12], poisoning-focused FL settings [13,16], privacy-preserving aggregation [50], and provenance-sensitive distributed pipelines [79].

Aggregation is a particular point of failure. Malicious gradients can bias the global model while blending into natural non-IID variation, and secure aggregation or differential privacy can hide update-level anomalies from inspection [49,80]. Temporal attacks add another layer of difficulty because adversaries can distribute small perturbations across rounds, reinforcing backdoors or slowly drifting the model without triggering abrupt statistical alarms [28,52]. These structural limits motivate accountability mechanisms such as auditable participant selection [18], verifiable aggregation proofs [19–21], and federated provenance frameworks [22].

3.3.1. Client-Side Poisoning

Client-side poisoning is among the most prevalent and well-studied classes of adversarial threats in FL. Because FL delegates local training to individual participants, a compromised or malicious client can manipulate its data, training procedure, or gradient updates before transmission to the server. This autonomy, combined with limited visibility into local computation, allows adversaries to inject harmful behavior without accessing

a centralized training pipeline [28,52]. The attack surface therefore spans local dataset curation, preprocessing, model optimization, and gradient packaging.

Data poisoning in FL follows the clean-label, dirty-label, label-flipping, and backdoor patterns introduced above, but the audit problem is sharper. Because data remain on-device, the server cannot directly inspect whether a client has altered labels, inserted triggers, or corrupted feature distributions. Even substantial corruption may therefore appear only indirectly through gradients. Under non-IID data, label inconsistencies can resemble natural client heterogeneity, while clean-label attacks may preserve normal training-loss profiles [14,23].

Model and gradient poisoning bypass the dataset and directly manipulate a client's computed update. The main mechanisms are model replacement, in which a scaled update overwrites the global model in one round [52]; gradient ascent, in which the update intentionally moves optimization in the wrong direction [67]; and stealth poisoning, in which malicious updates mimic benign statistics to evade anomaly detectors [66].

These attacks are powerful in FL because heterogeneous client data naturally produce high-variance gradients. That variance makes it harder to distinguish malicious updates from honest but unusual local training behavior.

3.3.2. Server-Side and Aggregation Attacks

Although FL is often framed around malicious clients, the coordinating server remains a powerful control point because it selects participants, distributes model states, and defines how updates are aggregated. A compromised server can bias aggregation or alter client weights in FL life-cycle threat models [12,22], fork or manipulate model versions in provenance-sensitive settings [79], and inject poisoned parameters or amplify adversarial updates in model-poisoning scenarios [28,81].

These attacks are difficult to detect because clients usually cannot inspect aggregation internals or verify whether the broadcast model is a faithful aggregate of submitted updates [49,80]. Secure aggregation can intensify this blind spot: it protects client confidentiality but does not, by itself, prove that the server aggregated honestly or distributed the same model to all clients. Server-side poisoning therefore exposes a central accountability limitation in FL: robust aggregation and majority voting help only when the server is trusted, whereas untrusted-server settings require verifiable aggregation, signed transcripts, or external audit evidence.

3.3.3. Collusion and Multi-Round Poisoning

The iterative, multi-round structure of FL introduces threats that exploit temporal dynamics and distributed control. Unlike one-shot poisoning in centralized learning, FL adversaries can coordinate across clients, rounds, or even with the server itself. These persistent strategies create long-term poisoning trajectories that gradually reshape the global model while evading round-by-round anomaly detection.

Client collusion is one of the most effective forms of distributed poisoning. When multiple malicious participants synchronize their updates, they can influence the global model while maintaining statistical similarity to one another. This behavior undermines robust aggregation methods such as trimmed mean, median, and Krum, which rely on malicious updates appearing as isolated outliers [82,83]. By distributing small portions of a perturbation across multiple clients, adversaries can create a significant aggregate effect while each individual update remains deceptively benign.

Collusion may also occur between clients and the central server. In this scenario, the server amplifies or prioritizes malicious updates, distributes tailored model versions to attacking clients, or subtly adjusts aggregation rules to support the adversaries' objec-

tives [13]. Such hybrid collusion is especially challenging to detect because it invalidates the foundational trust assumptions of FL: the server is assumed honest, and clients are assumed independent. When these assumptions fail, many FL defenses lose their theoretical guarantees.

Multi-round poisoning offers another avenue for stealth. Instead of injecting a large malicious update at once, adversaries gradually drift the model so that each update appears statistically consistent with natural fluctuations caused by non-IID client data. Slow-drift poisoning, adaptive gradient manipulation [54], and periodic trigger reinforcement enable adversaries to accumulate harmful influence without triggering anomaly detectors that rely on abrupt deviations. The temporal dimension also helps preserve backdoors even if some malicious clients drop out or are intermittently inactive.

Sybil attacks further amplify poisoning by creating multiple fake client identities. An adversary can inflate its share of participating clients in each round, overwhelm robust aggregators such as Krum or FoolsGold, and effectively take control of the global model [82,84]. These attacks exploit weak identity verification, especially in cross-device settings with many ephemeral participants.

Altogether, collusive and multi-round poisoning transform poisoning from a one-time event into an evolving adversarial process. Non-IID data, decentralized control, and repeated training rounds make these attacks difficult to detect or attribute without specialized accountability and provenance mechanisms.

Table 16 summarizes the major families of poisoning attacks in FL by comparing their mechanisms, stealth characteristics, detectability, and representative countermeasures. It also shows how non-IID data, decentralized control, and multi-round optimization amplify certain attacks while weakening traditional defenses.

Table 16. Comparative analysis of major poisoning attacks in FL.

Attack Type	Mechanism	Stealth Level	Detectability	Primary Defense (Examples)
Label flipping	Clients alter labels randomly or target specific classes [23,72]	Low (large gradient shift)	High under distance metrics	Local/global validation, confusion-matrix checks
Gradient poisoning	Adversarial gradient crafting via bilevel optimization [2,70]	Medium	Medium (norm and angle metrics)	Krum [85], trimmed mean [83]
Model replacement	Single-round overwrite of global model via scaled malicious updates [52]	Low (extreme deviation)	High (Euclidean distance, cosine)	FLTrust [86], Multi-Krum [85]
Backdoor (visible)	Large, high-frequency or patterned triggers embedded in inputs [73]	Medium	Medium (spectral activation analysis)	Spectral signatures [87] and pruning
Backdoor (stealthy)	Clean-label triggers or imperceptible perturbations [14,56]	Very high	Very low under non-IID noise	Neural Cleanse [88], STRIP [89]

3.3.4. Why FL Is Harder to Audit

Compared to centralized learning, FL poses major structural barriers to forensic analysis, accountability, and post-incident investigation [71,77].

- Lack of global visibility. The server never sees client data, intermediate states, or, when secure aggregation is used, local gradients, which makes attribution extremely difficult [28].
- Opaque aggregation. Robust aggregation techniques discard statistical information needed for forensics, while secure aggregation fully hides client updates [80].
- Untrusted participation. Cross-device FL allows open enrollment, enabling adversaries to create multiple fake clients and amplify their influence through Sybil-style coordination [79,82].
- No unified provenance trail. FL often lacks unified dataset lineage, versioning, and update histories, unlike centralized pipelines in which provenance is easier to trace [78]. These structural challenges make FL intrinsically harder to secure and audit and

motivate the need for accountability-integrated taxonomies and forensic-ready FL architectures, as explored in later sections.

3.4. Accountability-Integrated Taxonomy (AIT)

The Accountability-Integrated Taxonomy (AIT) extends traditional poisoning classifications by showing how accountability, attribution, and auditability differ between centralized learning and FL. AIT is intended as an analytical coding instrument rather than a descriptive checklist. A method can be robust against poisoning while still scoring low on accountability if it does not preserve evidence needed for post-incident reconstruction, attribution, aggregation verification, tamper detection, or audit. Table 17 defines the indicators used to code each method consistently across attack and defense families.

The purpose of AIT is not to replace existing poisoning taxonomies, but to make them more useful for FL design and evaluation. Conventional schemes usually identify what is attacked, such as data, labels, gradients, aggregation, or model behavior. AIT adds a second layer that asks how the attack or defense can be observed, which actor can be linked to the event, what evidence is preserved, and which trust assumption applies.

For example, anomaly detection can provide strong statistical visibility but weak support under secure aggregation. Byzantine-robust aggregation can resist isolated malicious clients while remaining weak under server distrust or collusion. Verifiable aggregation can offer high auditability but impose communication and computation costs. This lens distinguishes methods that merely improve robustness from those that also produce evidence for diagnosis, remediation, and external audit.

AIT is therefore distinct from simply relabeling secure aggregation, provenance, verifiability, or audit logging. Secure aggregation is coded as a privacy mechanism when it hides updates, as an accountability mechanism when it emits commitments or verification transcripts, and as a forensic limitation when it prevents client-level replay. Provenance is coded according to whether it links a dataset, client update, communication round, or server-side aggregation decision.

Verifiability is separated into aggregation correctness, participant authenticity, and tamper evidence. Audit logging is treated as useful only when the log is identity-bound, tamper-evident, retained, and interpretable after a poisoning incident. This decomposition converts known mechanisms into comparable accountability roles under explicit threat models and deployment trade-offs.

In centralized learning, the entire pipeline—including data ingestion, preprocessing, model updates, and training logs—is observable within a single controlled environment. This visibility helps trace poisoned samples, mislabeled data, or anomalous gradients to dataset sources or training iterations, a property emphasized by early poisoning work [2], regression-poisoning studies [70], and malware-oriented adversarial-learning analyses [75]. FL weakens this accountability baseline because evidence is distributed across clients, rounds, privacy mechanisms, and aggregation decisions. AIT therefore scores both the technical attack or defense mechanism and the evidence pathway available for audit.

AIT can be applied as a repeatable coding procedure. First, identify the poisoning mechanism using the baseline taxonomy: data poisoning, label flipping, model poisoning, gradient manipulation, backdoor insertion, server-side manipulation, or collusion. Second, specify the deployment setting, including update visibility, secure aggregation or differential privacy, server trust, and possible client collusion. Third, score the five accountability-benefit indicators in Table 18 and the audit-overhead level in Table 19. Fourth, record limiting assumptions or ambiguities in a notes column rather than averaging the indicators into a single score. The resulting profile shows where a method is practically useful, where it fails, and what additional evidence mechanisms are needed.

Table 17. Operational definitions of AIT indicators for FL poisoning defenses.

AIT Indicator	Meaning in FL Poisoning	Evidence That Counts	What Does Not Count
Observability	Extent to which poisoning-relevant events can be inspected during or after training, including client updates, validation effects, aggregation choices, and model-release events.	Per-client update statistics, validation traces, anomaly scores, rejected-update records, visible aggregation inputs, committee observations, or privacy-preserving proofs that expose verification facts.	General claims of robustness, final accuracy alone, encrypted aggregates without auxiliary evidence, or logs that only record that a round occurred without poisoning-relevant details.
Attribution specificity	Granularity with which suspicious behavior can be linked to an accountable entity, such as a sample source, client, cohort, round, committee, server action, or model version.	Signed client updates, authenticated client identifiers, sampling receipts, committee decisions, round identifiers, server-signed model hashes, and provenance links between actor, artifact, and round.	Anonymous aggregate metrics, unbound anomaly scores, unverifiable client labels, or post hoc suspicion that cannot be connected to an actor or control point.
Aggregation verifiability	Ability of clients, auditors, or committees to check that the server computed, filtered, and released the claimed aggregate under the stated protocol.	Commitments, authenticated transcripts, secure aggregation receipts, zero-knowledge proofs, verifiable computation outputs, witness signatures, and server-signed global model hashes.	Trusting the server by assumption, reporting clean accuracy, or using a robust aggregator without evidence that the selected, trimmed, or broadcast updates match the claimed computation.
Tamper evidence	Ability to detect later modification, deletion, reordering, or substitution of evidence relevant to a poisoning incident.	Hash chains, append-only ledgers, trusted timestamps, digital signatures, signed receipts, immutable model-version records, and independently retained audit bundles.	Ordinary mutable logs, local files without signatures or timestamps, undocumented retention policies, or dashboards whose records can be rewritten by the same party under investigation.
Forensic replayability	Ability to reconstruct the incident after the fact without exposing raw client data by re-checking submitted updates, filtering decisions, aggregation outputs, model versions, and detection outcomes.	Versioned models, retained update hashes or summaries, round metadata, validation traces, anomaly thresholds, rejection logs, proof transcripts, provenance records, and replayable audit bundles.	Final model accuracy, unverifiable summaries, one-time alerts with no retained evidence, or privacy mechanisms that permanently hide all client- or round-level evidence without substitute proofs.
Audit overhead	Additional storage, computation, communication, latency, governance, or infrastructure burden required to preserve accountability evidence; this is a cost indicator, not an accountability benefit.	Extra proof-generation time, ledger storage, retained logs, signatures, key management, committee operations, auditor access control, and communication expansion.	Accuracy loss alone, normal FL training cost, or implementation effort unrelated to evidence preservation or auditability.

Table 18. AIT accountability-benefit scoring scale.

Score	Meaning	Coding Interpretation
0	Not supported or not discussed	No explicit evidence supports the indicator, or the required evidence is hidden, unavailable, mutable, or outside the method's stated assumptions.
1	Weak or implicit support	Indirect evidence may exist, but it is not role-bound, retained, independently verifiable, or free from reliance on an unstated trusted component.
2	Partial or design-level support	Evidence artifacts or protocol steps support the indicator, but coverage is incomplete, assumption-specific, probabilistic, or not fully operationalized.
3	Strong operational support	Direct, role-bound, retained, and reviewable evidence supports independent verification or post-incident analysis under the stated threat model.

The coding rules are applied conservatively:

1. Code only what the method explicitly supports under the stated deployment and threat model.
2. Do not assign high accountability scores merely because a method improves robustness, clean accuracy, convergence, or attack success rate.
3. Do not treat privacy mechanisms as accountability mechanisms unless they preserve verifiable evidence such as commitments, receipts, proof transcripts, or retained audit metadata; if secure aggregation hides client updates, lower observability and attribution unless substitute evidence makes the hidden step verifiable.

4. If a method has no logs, signatures, hashes, commitments, audit transcript, retained metadata, or replay mechanism, it should not receive high tamper-evidence or replayability scores.
5. If a server can alter sampling, filtering, aggregation, logging, or model release without external verification, lower aggregation verifiability and tamper evidence.
6. When uncertain, assign the lower score and explain the ambiguity in the notes column.

Table 19. AIT audit-overhead cost scale.

Level	Meaning
Low	Minimal extra cost beyond ordinary FL training, such as lightweight metadata retention or local scoring already produced by the defense.
Medium	Moderate storage, computation, communication, or governance cost, such as retained round logs, signed messages, validation traces, or periodic auditor access.
High	Significant technical or governance burden, such as cryptographic proof generation, witness committees, large communication expansion, persistent ledger storage, or complex key management.
Very high	Difficult to deploy without special infrastructure, such as cross-organization governance, high-throughput ledger operation, specialized cryptographic services, trusted timestamping, or continuous independent audit capacity.

Tables 20 and 21 illustrate the application of these coding rules. Table 20 compares representative FL defense families by accountability benefits and audit overhead, while Table 21 summarizes the main conflicts that arise when privacy or robustness mechanisms limit attribution, replayability, or evidence preservation and specifies how AIT resolves each conflict.

Table 21 modifies the AIT procedure through a conflict-resolution protocol rather than through unrestricted exceptions. First, conflicting objectives remain separate dimensions. A privacy or robustness benefit cannot compensate numerically for missing attribution, tamper evidence, or replayability, and the indicators are not averaged into one score. Second, direct access to raw updates is not mandatory when substitute evidence supports the same audit claim. This bounded exception applies only when the substitute artifact is role-bound, retained, independently verifiable, and generated under the stated threat model.

Third, conditional disclosure mechanisms, such as threshold-authorized opening or auditor access to encrypted evidence, receive credit only when activation conditions, key holders, access logs, and oversight responsibilities are specified. Finally, uncertainty is preserved explicitly. Probabilistic reconstruction, cohort-level attribution, or uncorroborated anomaly detection is coded at the supported level rather than promoted to exact client-level attribution.

These rules allow AIT to accommodate privacy-preserving and resource-constrained deployments without weakening its evidentiary standard. They also encourage practical designs based on data minimization, encrypted or access-controlled retention, batched commitments, selective disclosure, and deployment-specific audit profiles.

The scoring scheme improves transparency and consistency, but does not eliminate interpretive judgment. Some methods provide incomplete protocol details, and accountability properties may depend on deployment choices, such as log retention, key management, trusted timestamping, auditor availability, or whether secure aggregation is paired with verifiable evidence. For future methods, the same indicators operate as design requirements. Researchers should report not only accuracy under attack, but also what evidence

is generated, who can verify it, whether it can be replayed, and whether the overhead is realistic for deployment.

Table 20. Worked application of AIT scoring to representative FL defense families.

Method/Family	Obs.	Attrib.	Agg. Verif.	Tamper	Replay	Audit Overhead	Notes/Ambiguity
FedAvg baseline	1	1	0	0	0	Low	Server may see client updates in plain FL, but baseline FedAvg provides no built-in signatures, tamper evidence, verification transcript, or replay bundle.
Krum or Byzantine-robust aggregation	2	1	1	0	1	Low–Medium	Improves robustness against outliers when assumptions hold; accountability depends on retaining selected/rejected update metadata and is weak under server tampering.
Trimmed mean or median aggregation	2	1	1	0	1	Low–Medium	Can reduce poisoned-update impact but may discard useful evidence unless trimming thresholds, discarded coordinates, and update hashes are logged.
Anomaly-detection defenses	2	2	0	1	2	Medium	Logged scores, thresholds, and update hashes improve observability and attribution; false positives under non-IID data and mutable logs limit stronger scores.
Secure aggregation alone	0	0	0	0	0	Medium	Strong privacy but hides client updates; accountability scores rise only if the protocol also emits commitments, receipts, or verifiable transcripts.
Differential privacy	1	0	0	0	0	Low–Medium	Protects privacy but injected noise can weaken forensic reconstruction and attribution; does not verify aggregation by itself.
Blockchain or ledger anchoring	2	2	1	3	2	High	Strong tamper evidence after anchoring; delayed anchoring, pre-anchor tampering, key management, and ledger overhead remain limiting assumptions.
Verifiable aggregation or zero-knowledge proof	2	1	3	2	2	High	Provides strong evidence of aggregation correctness, but may not identify which client supplied poisoned data or whether a semantic backdoor exists.
Witness committee or external auditor	2	2	2	2	2	High–Very high	Independent witnesses improve verification and replay if they retain signed records; cost and governance burden depend on committee design and auditor access.

Table 21. Common AIT coding conflicts among privacy, robustness, and accountability.

Conflict	Effect on AIT Coding	Resolution or Bounded Exception
Secure aggregation vs. attribution	Hidden client updates reduce direct observability and may prevent individual attribution.	Apply a substitute-evidence exception only when role-bound commitments, signed receipts, admissibility proofs, or threshold-authorized opening provide independently verifiable evidence. Otherwise, retain the lower attribution score and report the supported granularity, such as cohort rather than client.
Differential privacy vs. forensic replay	Noise can reduce reconstruction accuracy and weaken explanations based on update statistics.	Score replayability using the evidence that remains under the declared privacy budget. Preserve encrypted hashes, model versions, thresholds, and proof transcripts; do not claim exact replay when only probabilistic reconstruction is possible.
Robust aggregation vs. evidence preservation	Median, trimmed mean, Krum, or similar methods may discard or transform evidence while reducing attack impact.	Require a minimal evidence bundle containing update commitments, selection or rejection decisions, thresholds, software versions, and aggregate hashes. Retention may be encrypted or access-controlled, but unavailable discarded evidence cannot receive replay credit.
Blockchain logging vs. audit overhead	Anchoring improves tamper evidence after recording but increases cost, storage, latency, governance burden, and key-management complexity.	Use deployment-specific profiles rather than a universal requirement: high-stakes cross-silo settings may justify full anchoring, whereas cross-device settings may use batched hashes, off-chain encrypted logs, and periodic anchoring. AIT reports the resulting overhead separately.
Anomaly detection vs. false attribution	Non-IID behavior can resemble poisoning, so a detector alert may incorrectly implicate a benign client.	Treat anomaly scores as investigation triggers rather than conclusive attribution. Scores above 1 require interpretable features, calibrated thresholds, retained evidence, and corroboration by validation, repeated-round behavior, or independent review.

4. Defenses and Countermeasures in FL

The reviewed evidence supports multilayered defenses for poisoning threats arising from decentralized training, untrusted participants, and limited observability. This section integrates category-based defenses, life-cycle-aligned mechanisms, and accountability-centered strategies from the reviewed corpus to synthesize how defenses address the technical, temporal, and governance dimensions of FL.

4.1. Threat-Model Boundaries and Defense Applicability

Before comparing defenses, it is necessary to separate the threat model from the evidence model. FL poisoning defenses are often evaluated under different assumptions about client visibility, aggregation trust, privacy mechanisms, and adversary coordination. Secure aggregation weakens a detector that requires visible per-client gradients. A robust aggregator that assumes an honest server does not address server tampering. Similarly, a protocol that verifies aggregation correctness does not by itself establish data integrity.

Table 22 therefore separates the main threat types and identifies the evidence and defenses relevant to each boundary.

Table 22. Threat-model matrix separating FL poisoning, privacy, server, collusion, and audit-evidence failures.

Threat Type	Definition	Attacker Capability	Evidence Needed	Defenses That Apply	Defenses That May Not Apply
Data poisoning	Malicious client modifies local samples, labels, class balance, or trigger patterns before training [12–14].	Controls local data but may submit apparently normal updates.	Client behavior, data-distribution signals, validation traces, trigger tests, update effects, and provenance metadata.	Anomaly detection, robust aggregation, validation-based checks, backdoor testing, provenance logging [40,41,88–99].	Aggregation verification alone; it can prove the server computed correctly but cannot prove local data integrity.
Model poisoning	Malicious client manipulates the model update, gradient, scaling factor, or local training objective directly [16,28,81].	Controls local training or submitted update.	Update norms, gradients or update summaries, hashes, clipping records, anomaly scores, selected/rejected update metadata.	Robust aggregation, update filtering, clipping, anomaly detection, malicious-secure admissibility checks [32–36,84–87,100,101].	Data-only defenses and validation-only checks that do not inspect or constrain submitted updates.
Server tampering	Aggregator changes, drops, delays, reorders, forks, or selectively reports updates, logs, or global models [49,80].	Controls aggregation, filtering, broadcast, or logging.	Aggregation transcript, signed updates, server-signed model hashes, commitments, timestamps, ledger anchors, witness records.	Verifiable aggregation, audit logs, witness committees, signed transcripts, zero-knowledge proofs, ledger anchoring [19–21,36–39,50,82,102].	Client-only poisoning defenses and robust aggregation rules that assume the server applies the rule honestly.
Collusion and Sybil behavior	Multiple clients coordinate poisoning, imitate one another, or create fake identities to amplify influence [30,82,83].	Controls several clients, identities, or client–server relationships.	Correlation evidence, identity checks, enrollment records, update similarity, sampling receipts, Sybil-detection traces.	Graph/correlation defenses, robust aggregation under bounded collusion, identity controls, participant-selection audit [18,30,82–85].	Simple single-update outlier detection and defenses assuming independent malicious clients.
Privacy leakage	Server, client, auditor, or observer infers private client data from updates, gradients, validation data, or metadata [24,48–50,80].	Observes updates, gradients, aggregates, logs, or side-channel metadata.	Privacy-risk analysis, leakage tests, exposed metadata inventory, and definition of what evidence remains visible.	Secure aggregation, differential privacy, encryption, secure multiparty computation (MPC), homomorphic encryption, privacy-preserving audit protocols [22,33–40,50,80].	Visible-update forensic methods that require raw gradients or identifiable per-client update details.
Audit-evidence failure	System cannot reconstruct what happened after an attack because logs are absent, mutable, incomplete, or not identity-bound [7,8,42,44].	Missing evidence, tampered evidence, or insufficient retention rather than a single poisoning action.	Signed logs, timestamps, hashes, model-version history, provenance records, proof transcripts, replayable audit bundles.	Ledger anchoring, provenance, audit bundles, signed receipts, governance mapping, external auditors [21,22,36,41,42,44–46,82,102–104].	Pure robustness methods that reduce attack impact but preserve no signed, timestamped, replayable, or tamper-evident evidence.

This manuscript distinguishes four server assumptions. A trusted serverfollows the protocol honestly and applies sampling, filtering, aggregation, logging, and model-release rules as specified. An honest-but-curious server follows the protocol but may try to infer private information from updates, aggregates, metadata, or audit logs. A malicious aggregator may drop, alter, reorder, delay, fork, selectively report, or suppress updates, logs, or global models. An audited server may be honest or malicious, but its actions are checked through signed logs, witnesses, cryptographic proofs, ledger anchors, or external auditors.

Robust aggregation and anomaly detection usually require a trusted or audited server. Secure aggregation mainly addresses an honest-but-curious server unless it is paired with verifiable aggregation. Verifiable aggregation targets audited or malicious-aggregator settings by checking aggregation correctness, not the semantic cleanliness of client data.

Client adversaries are also separated. Benign clients follow the protocol but may have non-IID data. Byzantine clients may send arbitrary or malformed updates. Data-poisoning clients alter local samples or labels, while model-poisoning clients directly manipulate updates, including scaled or replacement updates. Backdoor attackers aim to preserve clean accuracy while implanting trigger-specific behavior. Sybil clients create multiple identities, and colluding clients coordinate updates or identities across rounds.

Robust aggregation mainly targets Byzantine or model-poisoning clients under bounded malicious-client fractions. Validation and backdoor tests target data and backdoor poisoning when validation evidence is representative. Graph or correlation methods target

Sybil and collusive behavior. Identity-bound logging supports attribution but does not itself remove malicious updates.

Table 23 consolidates these assumptions and shows which defense families remain applicable under secure aggregation, malicious-server behavior, collusion, and audit-evidence requirements.

Table 23. Defense assumptions and incompatibilities used to interpret scenario-based defense comparisons and AIT scores.

Defense Category	Requires Visible Client Updates?	Compatible with Secure Aggregation?	Handles Malicious Clients?	Handles Malicious Server?	Handles Collusion?	Supports Audit Evidence?	Key Limitation
Robust aggregation	Usually yes	Often no unless modified	Partially, under bounded fractions	No, unless aggregation is externally verified	Limited	Weak unless selection/rejection logs are retained	Assumes the server applies the rule honestly and may discard forensic evidence.
Anomaly detection	Usually yes	Limited; needs summaries, committees, or proofs	Yes for visible anomalous updates	No, unless scoring and logs are audited	Limited–Medium	Medium if scores, thresholds, and hashes are retained	Sensitive to non-IID data and false attribution.
Secure aggregation	No	Yes	Not directly	Honest-but-curious only unless verifiable	No	Low unless extra proofs or logs are added	Protects privacy but reduces observability and attribution.
Verifiable aggregation	Not always	Sometimes	No for semantic data poisoning	Yes for aggregation correctness	No, except bounded protocol adversaries	High for server actions	Proves the server aggregated correctly; does not certify local training data.
Blockchain and logging	Depends on what is logged	Can be compatible	No direct defense	Helps detect tampering after logging	No direct defense	High if signed, timestamped, and retained	Protects records after anchoring but not the truthfulness of what was logged.
Differential privacy	No	Yes	No direct poisoning defense	No	No	Low	Reduces privacy leakage but weakens forensic detail and replayability.
Witness committee/auditor	Depends on audit design	Possible	Partial through validation or challenge checks	Yes, if committee verifies server actions	Partial	High if records are signed and retained	Adds governance, availability, and committee-trust assumptions.

Taken together, these boundaries create several incompatibilities. Secure aggregation reduces privacy leakage but limits client-level observability and attribution unless commitments, receipts, or proof artifacts replace direct visibility. Verifiable aggregation shows that a server followed a stated computation, but it does not rule out poisoned local data or hidden triggers. Robust aggregation may reduce attack impact while preserving little forensic evidence unless update hashes, selected-update metadata, and rejection decisions are retained. Blockchain anchoring can show that records were not altered after anchoring, but not that the original log was complete or truthful. Differential privacy reduces leakage but may also weaken forensic replayability and attribution by perturbing update signals.

These assumptions constrain both the scenario-based defense comparisons and the AIT scores: methods are coded according to the evidence they preserve under their stated visibility, privacy, server-trust, and collusion assumptions.

4.2. Category-Based Defenses in FL

FL employs a broad spectrum of defense mechanisms, but each applies only under specific visibility, client-adversary, privacy, and server-trust assumptions. Because FL decentralizes training and limits global visibility, no single defense family in the reviewed corpus is sufficient across the evaluated scenarios. Layered approaches are therefore better supported than single-mechanism defenses [11,13]. This subsection groups defenses into anomaly and statistical detection, performance-based filtering, Byzantine-robust aggregation, cryptographic defenses, and accountability-enabling frameworks.

4.2.1. Anomaly and Statistical Detection

Anomaly-detection methods identify suspicious client updates by analyzing statistical deviations in gradient norms, cosine similarity, weight directions, or parameter trajectories when client- or round-level update features are visible. These methods assume that malicious updates differ significantly from honest ones in magnitude or orientation [86,92]. Classical metrics include Euclidean distance, cosine similarity, and Pearson correlation, often combined with clustering to separate benign and adversarial groups [25,93]. More advanced approaches employ deep autoencoders, SVM-based detectors, and GAN-inspired models to learn representations of benign update distributions [94,95].

Although effective in homogeneous settings, these defenses degrade under high non-IID heterogeneity, where benign updates naturally diverge and poisoned updates can blend into statistical noise [12,79]. Secure aggregation further reduces their applicability by obscuring individual gradients [80].

4.2.2. Performance-Based Filtering

Performance-based defenses evaluate client updates through validation accuracy, loss behavior, or prediction consistency on a trusted validation set or audited validation process [86,96]. Poorly performing updates are rejected or down-weighted before aggregation. Collaborative validation schemes use client-side validation or client grouping and voting to assess update credibility and model performance [97,98]. Although intuitive and easy to implement, these defenses rely on high-quality, representative validation data. Poisoned or biased validation sets can cause honest clients to be misclassified as adversaries, and well-crafted backdoors can preserve clean accuracy while remaining undetected [99].

4.2.3. Byzantine-Robust Aggregation

Byzantine-robust aggregation (BRA) algorithms mitigate anomalous or adversarial client updates when the server honestly applies the aggregation rule or when that application is externally verified [100,105]. Popular BRA methods include Krum and Multi-Krum, which select updates that are closest to the majority [85]; trimmed mean and median, which remove extreme values before averaging [83]; and Bulyan, which combines selection and trimming to improve robustness [67]. Although these methods are effective against isolated malicious clients, they rely on honest-majority assumptions and degrade under collusive or Sybil attacks [82]. Non-IID data further weakens BRA performance by making honest updates appear inconsistent or adversarial [12,105].

4.2.4. Cryptographic and Encryption-Based Defenses

Cryptographic and encryption-based defenses make selected properties of client participation, update admissibility, or aggregation correctness verifiable under explicit trust assumptions. These mechanisms can be grouped by control point. Participant authenticity and identity binding reduce impersonation and Sybil-style entry points [18,22]. Signatures, commitments, and provenance metadata bind updates to rounds and evidence trails [22]. Clipping or admissibility controls constrain malicious updates before aggregation [32].

Malicious-secure aggregation protocols address adversarial participants through ELSA [33], multi-round Flamingo aggregation [34], and RoFL robustness checks [35]. Zero-knowledge and public-audit mechanisms produce proof transcripts through ZKFL-style aggregation [19], low-cost zero-knowledge-proof collaboration [20], and publicly auditable FL [21]. Newer verifiable aggregation protocols extend this proof-oriented design space [36,37].

Plain secure aggregation mainly protects confidentiality. By itself, it neither detects poisoning nor identifies malicious clients. It becomes accountability-relevant when com-

bined with authenticated shares, commitments, MPC-based admissibility checks, or zero-knowledge proofs. Homomorphic encryption and MPC can also aggregate encrypted updates, but they introduce additional computational and communication overhead [100].

Overall, these mechanisms complement robust aggregation and anomaly detection by supplying privacy-preserving aggregation controls [40], tamper-evident audit evidence [41], and proof transcripts for third-party verification, post-training audit, and compliance review [36,37]. Table 24 groups the recent mechanisms by their main accountability role. Its key takeaway is that cryptographic approaches verify specific properties—such as participant authenticity, update admissibility, provenance, or aggregation correctness—but none independently establishes that client data are semantically clean.

Table 24. Recent (2023–2025) cryptographic, integrity-preserving, and verifiable FL research supporting accountability and poisoning resilience.

Theme	Representative Work (2023–2025)	Relevance to Accountability, Auditability, and Traceability
Verifiable aggregation/integrity	[36,37]	Enables cryptographic checks, such as zero-knowledge or verifiable computation techniques, allowing clients or external auditors to verify aggregation correctness without revealing individual client updates. This strengthens auditability for server actions, but does not by itself detect poisoned client data.
Malicious-secure aggregation (preventive poisoning resilience)	[32–35]	Provides cryptographic and MPC-based aggregation protocols that tolerate malicious clients and untrusted servers by enforcing update integrity and admissibility constraints before aggregation, thereby limiting adaptive poisoning attempts and strengthening accountability guarantees.
Client authenticity/ participant integrity	[18,22]	Uses auditable participant selection, identity-binding signatures, or traceable communication channels to reduce impersonation, repudiation, and Sybil-style poisoning entry points before model aggregation begins.
Update provenance/ round traceability	[22]	Preserves round-level lineage, commitments, or provenance metadata so that suspicious updates can be linked to a participant, a communication round, and an evidence trail for later audits or incident responses.
Secure aggregation with accountability support	[38,39]	Integrates secure aggregation with verification artifacts such as commitments, authenticated logs, or proofs that can serve as forensic evidence, thereby enabling traceability and post-incident audits while preserving client privacy.

These mechanisms also introduce trade-offs. Protocol complexity, computational overhead, and communication costs can limit scalability, especially in resource-constrained settings. Encryption can also hinder fine-grained anomaly detection and statistical inspection. Cryptographic mechanisms should therefore be treated as complements to robust aggregation and detection-based defenses, not as stand-alone solutions.

4.2.5. Accountability-Oriented Frameworks

Accountability-driven defenses introduce verifiable mechanisms for provenance tracking, tamper-evident aggregation, and forensic readiness. BVDFed combines Byzantine-resilient aggregation with verifiability, whereas CAFCOR combines robust gradient aggregation with correlated noise under an untrusted-participant and untrusted-server threat model [90,91]. Neither mechanism alone supplies hash-chained logs or audit-ledger integration. Blockchain-based FL systems can record model updates and coordination decisions in append-only ledgers, enabling traceability after logging without proving that the original content was complete or truthful. Digital signatures and cryptographic identity binding prevent impersonation and support attribution during malicious-activity analysis. These mechanisms enhance accountability when logs are complete and identity-bound, but they also introduce computational, storage, and governance overhead. They are evidence-enabling complements to robust aggregation and validation rather than complete defenses on their own.

Table 25 defines each dimension used in the ordinal, evidence-weighted scenario ratings reported in Table 26. These ratings are judgments rather than measured effect

sizes or universal rankings. The labels "Low", "Medium", and "High" have dimension-specific meanings. For example, "High" under server distrust indicates explicit support for untrusted or verifiable aggregation, whereas "High" under non-IID tolerance indicates repeated empirical or survey evidence of acceptable operation under heterogeneous client data.

Evidence labels are E = empirical evidence, D = design or protocol evidence, I = author interpretation, and M = mixed or conflicting evidence. When sources conflict, empirical results are prioritized over design claims, and design claims are prioritized over author interpretation. Ranges are reported where needed, and weak evidence is rated conservatively.

Table 25. Criteria used to derive the ordinal ratings in Table 26.

Dimension	Low	Medium	High
Non-IID tolerance	Evidence reports degradation under heterogeneity, or the method assumes IID-like update similarity.	Partial tolerance under moderate heterogeneity, selected datasets, additional calibration, or trusted validation data.	Empirical or repeated survey evidence shows operation under heterogeneous clients when stated assumptions hold.
Secure aggregation compatibility	Requires visible per-client updates or gradients and is weakened when secure aggregation hides them.	Compatible only with added metadata, committees, validation traces, commitments, or post-aggregation checks.	Designed to operate with encrypted/hidden updates or to emit privacy-preserving verification artifacts.
Server-distrust support	Assumes an honest or semi-honest server and gives clients/auditors no independent verification path.	Provides partial evidence, logs, committees, or trusted components, but does not fully remove server trust.	Explicitly supports malicious or untrusted-server settings through proofs, signatures, transcripts, ledgers, or external verification.
Collusive-attacker robustness	Mainly addresses isolated malicious clients and is weak against Sybil or coordinated attackers.	Some support through similarity analysis, identity binding, committees, or robust assumptions that hold only for limited collusion.	Directly models or verifies against coordinated clients, Sybils, dishonest majorities, or client–server collusion under stated bounds.

Table 26. Evidence-qualified comparison of representative defense-family tendencies under challenging FL scenarios. Ratings are ordinal summaries, not pooled performance estimates. Evidence labels: E = empirical, D = design/protocol, I = author interpretation, M = mixed or conflicting evidence.

Defense Family/Subfamily	Non-IID Tolerance	Secure Aggregation Compatibility	Server-Distrust Support	Collusive-Attacker Robustness	Evidence Type	Key Supporting Studies	Notes/Limitations
Median/trimmed-mean aggregation	Low–Medium (E,M)	Low (I)	Low (I)	Low (E,M)	E,M,I	[83,105]	Robustness depends on honest-majority and distributional assumptions; coordinate trimming can reduce attack impact but may discard evidence needed for replay or attribution.
Krum/Multi-Krum/Bulyan	Low–Medium (E,M)	Low (I)	Low (I)	Low (E,M)	E,M,I	[66,67,85,105]	Useful against isolated Byzantine updates, but optimized model-poisoning and collusion can exploit selection rules; server honesty is usually assumed.
Trust-bootstrapped aggregation/clipping	Medium (E)	Low–Medium (E,I)	Low (I)	Low–Medium (E,I)	E,I	[86,100,101]	Root datasets, clipping rules, or trusted anchors improve robustness under some heterogeneity, but do not by themselves verify server-side aggregation or preserve full audit evidence.
Anomaly/statistical detection	Low–Medium (E,M)	Low (I)	Low (I)	Low–Medium (E,M)	E,M,I	[12,79,92–96]	Requires visible update features, norms, reconstructions, or validation traces; non-IID data can create false positives, and secure aggregation removes many inspection signals.
Graph/correlation-based collusion or Sybil detection	Medium (E,I)	Low (I)	Low–Medium (I)	Medium (E,M)	E,M,I	[82,84]	More relevant to coordinated behavior than generic anomaly detection, but evidence is strongest when client identities and cross-round similarity traces are visible and retained.

Table 26. *Cont.*

Defense Family/Subfamily	Non-IID Tolerance	Secure Aggregation Compatibility	Server-Distrust Support	Collusive-Attacker Robustness	Evidence Type	Key Supporting Studies	Notes/Limitations
Performance-based filtering/validation voting	Medium (E,I)	Low–Medium (I)	Low (I)	Low–Medium (E,I)	E,I	[86,96–99]	Practical when representative trusted validation data or committees exist; clean accuracy can miss backdoors, and poisoned or biased validation data can invert decisions.
Secure aggregation alone	Low (I)	High (D)	Low (D,I)	Low (I)	D,I	[32,33,49,80]	Strong privacy compatibility but weak accountability: hidden per-client updates reduce observability, attribution, and many anomaly defenses unless separate evidence artifacts are emitted.
Malicious-secure aggregation/MPC admissibility	Medium (D)	High (D)	Medium–High (D)	Medium (D)	D	[32–35]	Protocols can enforce admissibility or tolerate malicious participants under stated bounds, but they may not identify semantic poisoning or provide full client-level forensic replay.
Verifiable aggregation/zero-knowledge proofs	Medium–High (D)	High (D)	High (D)	Medium (D,I)	D,I	[19–21,36–39]	Strong support for aggregation correctness and server accountability; not equivalent to poisoning detection or actor attribution unless combined with detection and identity evidence.
Blockchain or ledger logging	Medium (D,I)	Medium (D,I)	Medium–High (D,I)	Medium (D,I)	D,I	[17,40–42,102]	Improves tamper evidence after anchoring and supports audit trails, but delayed anchoring, pre-log tampering, consensus latency, and storage overhead limit deployment.
Provenance/traceability frameworks	Medium (D,I)	Medium (D,I)	Medium (D,I)	Low–Medium (I)	D,I	[22,103]	Valuable for evidence preservation and replay, but usually complements rather than replaces poisoning detection, robust aggregation, or verifiable server checks.
Witness committee/external auditor	Medium (D,I)	Medium (D,I)	Medium–High (D,I)	Medium (D,I)	D,I	[18,21,98]	Independent witnesses can improve verification and replay if they retain signed records; governance cost, validator trust, and collusion among witnesses remain open issues.

Table 26 is intended as a decision-support summary, not a universal ranking of defense families. Ratings are bounded by the cited studies and should be interpreted as evidence-weighted judgments. Because methods within a defense family vary substantially, broad ratings are reported as ranges where appropriate. The table also separates robustness from accountability: a defense can be strong against poisoning while still weak in forensic replayability, attribution, tamper evidence, or server-distrust support. Under the AIT lens, secure aggregation, robust aggregation, and blockchain logging are therefore not scored as interchangeable defenses; they preserve different kinds of evidence and fail under different trust assumptions.

4.3. Structured Trade-Off Analysis

The defense landscape reviewed in Section 4.2 suggests three recurring trade-offs for future FL design.

The first is accuracy/robustness versus privacy. Detection-based and validation-based defenses benefit from visibility into per-client behavior, but secure aggregation and strong privacy controls reduce this visibility. As a result, privacy-preserving deployments frequently lose the very signals that help identify poisoning attempts.

The second is auditability versus scalability. Evidence-rich designs such as authenticated logging, verifiable aggregation, and proof-carrying updates improve accountability, but they increase communication, storage, and verification overhead. This makes them attractive for cross-silo or high-stakes settings, yet potentially burdensome in large-scale cross-device FL.

The third is security coverage versus deployment realism. Many defenses are evaluated against isolated malicious clients, whereas real deployments may face heterogeneous data, adaptive collusion, or partially trusted infrastructure. A defense that is theoretically strong under one threat model may therefore provide limited assurance under more realistic assumptions.

These trade-offs imply that FL defenses should not be compared only by reductions in attack success rate. They should also be compared by the evidence they preserve, the trust assumptions they require, and the deployment constraints they impose.

4.4. Persistent Gaps

Although substantial progress has been made in understanding and mitigating poisoning attacks in both centralized learning and FL, the reviewed defenses remain fragmented across robustness, privacy, and accountability objectives. Many mechanisms address robustness or privacy in isolation, with less attention to how adversaries adapt across rounds, persist through multi-step strategies, or exploit the limited auditability of decentralized training. FL also introduces structural blind spots—such as untrusted aggregation, severe non-IID heterogeneity, and restricted visibility into client behavior—that make attribution especially challenging [16,50].

As coded in this survey, many reviewed defenses do not provide verifiable provenance, tamper-evident aggregation, or post-incident forensic reconstruction, leaving systems exposed to opaque, stealthy, and collusive attacks that may evade detection even under robust aggregation [13,66]. These recurring gaps motivate the framework requirements summarized next.

A further limitation is that family-level ratings can obscure domain-specific adversary behavior. Adjacent security work illustrates why narrow evaluation assumptions can be misleading. Liveness-only biometric defenses may fail against puppet attacks even when they defeat conventional spoofing [104], and LLM-based smart-contract detectors must be evaluated against unseen fraud semantics and real-world deployment data rather than only static benchmark labels [106]. These studies are not evidence for the FL ratings in Table 26. Instead, they motivate a conservative interpretation of those ratings and reinforce the need for scenario-specific FL benchmarks.

4.5. Opportunities for Unified Frameworks

The evolution of FL demands architectural frameworks that jointly optimize privacy, robustness, and accountability rather than treating these guarantees as independent objectives. Current research still separates these goals. AI-enabled threat detection motivates automated monitoring [107], broad FL surveys emphasize differential privacy and secure aggregation [108], and hybrid privacy-preserving FL shows how confidentiality controls can be integrated into training [109].

Privacy mechanisms protect confidentiality but provide limited support for verifiable training or forensic traceability. Conversely, robust aggregation strategies such as CAFCOR and BVDFed improve resilience to Byzantine and poisoning attacks [90,91]; CAFCOR explicitly considers an untrusted server, while neither approach by itself provides a complete transparent audit trail. Accountability-enhancing tools—including privacy-preserving aggregation controls [40], blockchain-secured logging [41], and verifiable secure

aggregation [102]—offer traceability but remain largely decoupled from privacy and robustness objectives.

For that reason, we do not claim a fully specified new FL architecture here. Rather, we distill a conceptual design space for systems that are simultaneously private, robust, and auditable. This space has four requirements. Privacy-preserving observability enables the inspection of poisoning-relevant events without exposing raw client data. Evidence-carrying aggregation requires aggregation to emit commitments, signatures, or proofs rather than only a model output. Attribution across trust boundaries distinguishes client misconduct, server misconduct, and collusion. Standards-mappable assurance allows technical artifacts to support internal audits, external assurance, or compliance review.

AIT can guide future framework design by converting these requirements into implementation checkpoints. A practical accountable-FL method should specify threat-role assumptions and identify the minimum evidence needed in each training round. It should bind that evidence to clients, committees, or servers through verifiable identifiers and define how auditors can replay or verify events without accessing raw data.

To remain deployable, the method should report accountability overhead separately from model accuracy. Relevant costs include proof generation, communication expansion, ledger or log storage, and aggregation latency. These checkpoints translate a conceptual taxonomy into measurable design obligations that can be benchmarked, optimized, and compared across deployment settings.

The interaction between these requirements is illustrated in Figure 2. The three layers should be read as a design decomposition rather than a claim of implementation completeness. At the privacy-preserving layer, techniques such as differential privacy, homomorphic encryption, and MPC protect client updates from inference and leakage.

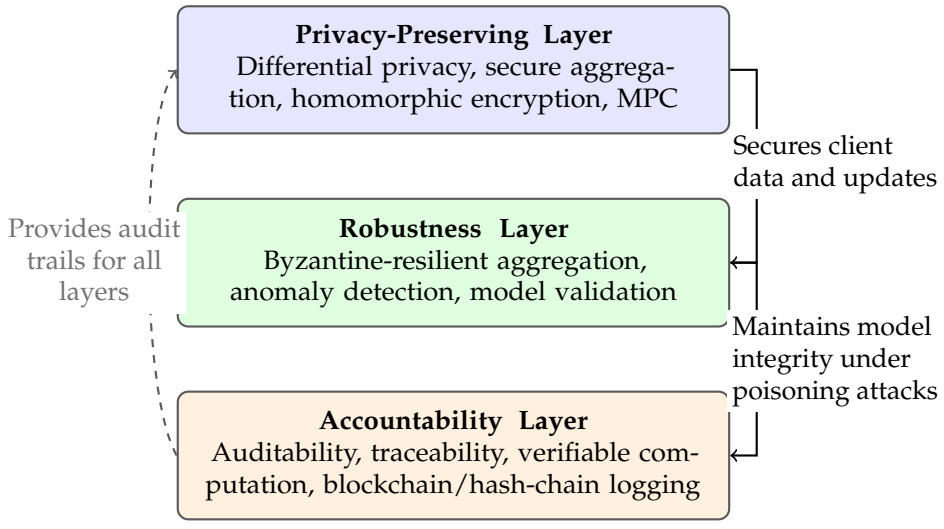

Figure 2. Conceptual architecture for a unified trustworthy FL framework integrating privacy, robustness, and accountability layers.

The robustness layer incorporates Byzantine-resilient aggregation, anomaly detection, and validation-based filtering. Blockchain-enabled FL supports decentralized coordination [102], Sybil mitigation addresses identity-based poisoning [82], and robust-FL surveys synthesize adversarial aggregation defenses [30]. The accountability layer adds tamper-evident audit trails through commitments, hash-chained logs, identity-binding signatures, and zero-knowledge proofs. Statistical poisoning detection can support audit triggers [31], auditable AI foundations clarify evidence requirements [43], and AI accountability work connects those artifacts to responsibility and oversight [44,45].

This layered view also deepens the paper's brief mapping to ISO/IEC 42001. At the governance level, FL systems need explicit role definitions, risk ownership, and incident-handling rules. At the operational level, they need evidence about sampling, aggregation, validation, and model release decisions. At the assurance level, they need artifacts that an independent auditor could inspect without re-running the entire training process. In other words, standards alignment is meaningful only if technical defenses emit reviewable evidence, not merely if they improve accuracy under attack.

5. Implications for Future Research and Practice

The implications of this study are twofold. For research, accountability should be evaluated as an explicit outcome rather than inferred from robustness or privacy metrics. For practice, retained FL artifacts should be tied to concrete governance duties, including incident investigation, risk ownership, and evidence review.

Future Directions

Future research should address three less-developed questions rather than simply restating the need for stronger logging, verification, and benchmarks.

First, accountable FL needs causal forensic models that distinguish correlation from responsibility. An audit trail should help estimate whether a client, server decision, validation rule, or collusive pattern caused a poisoned outcome. It should also report uncertainty rather than merely assign blame.

Second, evidence governance should become an explicit design problem. Researchers should determine which artifacts are sufficient for investigation, which are too privacy-sensitive to retain, how long evidence should persist, and how audit records interact with deletion, unlearning, or regulatory retention duties after an incident.

Third, the field should move from attack detection to recovery science. Relevant topics include selective rollback, client quarantine, targeted unlearning, model patching, and documentation of residual risk after attribution. Together, these directions shift the focus from building defenses in isolation to understanding how accountable FL systems investigate, recover from, and justify decisions after failures.

Table 27 synthesizes key research on accountability, auditability, and traceability in FL. These works span blockchain-backed audit trails, zero-knowledge-based verification, provenance tracking, explainability-driven debugging, and regulatory audit frameworks.

Taken together, Table 27 shows that the reviewed accountability research provides complementary pieces rather than a complete solution. Ledgers strengthen tamper evidence, proofs verify computations, and provenance supports tracing, but effective incident reconstruction requires these artifacts to be integrated.

Table 27. Recent research in accountability and traceability in FL.

Title	Focus Area
ISO/IEC 42001: Artificial Intelligence Management Systems [46]	Life-cycle governance, auditability, accountability
Blockchain-Based Federated Learning: A Survey and New Perspectives [42]	Auditing, traceability, taxonomy in BCFL
Blockchain-Enabled Accountable FL for Edge AI Systems [40]	Blockchain-backed accountability and edge audit trails
Verifiable and Accountable FL via Permissioned Blockchain [41]	Permissioned ledger evidence and accountability controls
Auditable FL with Blockchain-Based Participant Selection [18]	Auditable sampling, ledgered evidence

Table 27. *Cont.*

Title	Focus Area
zkFL: ZKP-based Gradient Aggregation for FL [19]	Verifiable aggregation, privacy
Trusted Model Aggregation with Zero-Knowledge Proofs [36]	Zero-knowledge proofs for trusted aggregation
RiseFL: Secure and Verifiable Data Collaboration with Low-Cost ZKPs [20]	Low-cost zero-knowledge proofs, verifiable training
Publicly Auditable and Privacy-Preserving FL [21]	Public auditability with robust aggregation
TraceFL: Interpretability-Driven Debugging in FL [103]	Neuron-level provenance, debugging
Enhancing Data Provenance & Transparency in FL [22]	Provenance tracking, reproducibility
PPTFL: Privacy-Preserving and Traceable FL [22]	Traceability with privacy preservation

6. Conclusions

This survey examined poisoning attacks in FL through an accountability-centered lens and supports three bounded conclusions within the screened corpus. First, privacy-preserving training, non-IID data, secure aggregation, client heterogeneity, and untrusted-server behavior reduce visibility and thereby change the poisoning problem in FL. Second, robustness alone is insufficient. Defenses that improve accuracy under attack may still fail to preserve evidence about which participant, server, round, or artifact caused an incident. Third, accountable FL requires defenses that are not only attack-resistant but also traceable, verifiable, and usable for post-incident diagnosis, audit, and governance review.

These conclusions should be interpreted as evidence from the reviewed and coded corpus rather than as exhaustive claims about all FL literature.

Overall, the central finding is that future FL systems should be designed as evidence-producing systems, not only as privacy-preserving or attack-resistant systems. This requires defenses that preserve enough trustworthy evidence for diagnosis, recovery, and governance review without collapsing the privacy guarantees that motivate FL in the first place.

Author Contributions: S.M. conceived the study, conducted the literature review, and drafted the manuscript. D.A. contributed to the design of the research framework, supervised the methodological development, and provided critical revisions. A.N. contributed to the conceptualization, supervised the overall research direction, and reviewed the manuscript for intellectual content. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: During the preparation of this manuscript, the authors used generative AI tools solely for language refinement and proofreading. The authors reviewed and edited all AI-assisted output and take full responsibility for the content of the publication.

Conflicts of Interest: The authors declare that the research was conducted in the absence of commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Jobin, A.; Ienca, M.; Vayena, E. The global landscape of AI ethics guidelines. *Nat. Mach. Intell.* **2019**, *1*, 389–399. [CrossRef]
2. Biggio, B.; Nelson, B.; Laskov, P. Poisoning attacks against support vector machines. In Proceedings of the 29th International Conference on Machine Learning (ICML), Edinburgh, UK, 26 June–1 July 2012.
3. Tian, Z.; Cui, L.; Liang, J.; Yu, S. A comprehensive survey on poisoning attacks and countermeasures in machine learning. *ACM Comput. Surv.* **2022**, *55*, 1–35. [CrossRef]
4. Mittelstadt, B.D.; Allo, P.; Taddeo, M.; Wachter, S.; Floridi, L. The ethics of algorithms: Mapping the debate. *Big Data Soc.* **2016**, *3*, 2053951716679679. [CrossRef]

5. Novelli, C.; Taddeo, M.; Floridi, L. Accountability in artificial intelligence: What it is and how it works. *AI Soc.* **2024**, *39*, 1871–1882. [CrossRef]
6. San Miguel, B.; Naseer, A.; Inakoshi, H. Putting accountability of AI systems into practice. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, Online, 7–15 January 2021; pp. 5276–5278. [CrossRef] [PubMed]
7. Cen, S.; Alur, R. From transparency to accountability and back: A discussion of access and evidence in AI auditing. In Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency, Rio de Janeiro, Brazil, 3–6 June 2024. [CrossRef]
8. Kroll, J.A. Outlining traceability: A principle for operationalizing accountability in computing systems. In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (ACM), Virtual Event, 3–10 March 2021; pp. 21–31. [CrossRef]
9. Cinà, A.E.; Demontis, A.; Biggio, B.; Pelillo, M.; Roli, F. Wild patterns reloaded: A survey of machine learning security against training data poisoning. *ACM Comput. Surv.* **2023**, *55*, 1–39. [CrossRef]
10. Pitropakis, N.; Panaousis, E.; Giannetsos, T.; Anastasiadis, E.; Loukas, G. A taxonomy and survey of attacks against machine learning. *Comput. Sci. Rev.* **2019**, *34*, 100199. [CrossRef]
11. Sikandar, H.; Waheed, H.; Tahir, S.; Malik, S. A detailed survey on federated learning attacks and defenses. *Electronics* **2023**, *12*, 260. [CrossRef]
12. Li, Y.; Guo, Z.; Yang, N.; Chen, H.; Yuan, D.; Ding, W. Threats and defenses in the federated learning life cycle: A comprehensive survey and challenges. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 15643–15663. [CrossRef] [PubMed]
13. Liang, J.; Wang, R.; Feng, C.; Chang, C.-C. A survey on federated learning poisoning attacks and defenses. *arXiv* **2023**, arXiv:2306.03397.
14. Xia, G.; Chen, J.; Yu, C.; Ma, J. Poisoning attacks in federated learning: A survey. *IEEE Access* **2023**, *11*, 10708–10722. [CrossRef]
15. Wang, Z.; Kang, Q.; Zhang, X.; Hu, Q. Defense strategies toward model poisoning attacks in federated learning: A survey. In Proceedings of the 2022 IEEE Wireless Communications and Networking Conference (WCNC), Austin, TX, USA, 10–13 April 2022; pp. 548–553. [CrossRef]
16. Nowroozi, E.; Haider, I.; Taheri, R.; Conti, M. Federated learning under attack: Exposing vulnerabilities through data poisoning attacks in computer networks. *IEEE Trans. Netw. Serv. Manag.* **2025**, *22*, 822–831. [CrossRef]
17. Zeng, H.; Zhang, M.; Liu, T.; Yang, A. A federated learning framework with blockchain-based auditable participant selection. *Comput. Mater. Contin.* **2024**, *79*, 5125–5142. [CrossRef]
18. Wang, Z.; Dong, N.; Sun, J.; Knottenbelt, W.; Guo, Y. ZKFL: Zero-knowledge proof-based gradient aggregation for federated learning. *arXiv* **2023**, arXiv:2310.02554.
19. Zhu, Y.; Wu, Y.; Luo, Z.; Ooi, B.C.; Xiao, X. Secure and verifiable data collaboration with low-cost zero-knowledge proofs. *Proc. VLDB Endow.* **2024**, *17*, 2321–2334. [CrossRef]
20. Zeng, H.; Yang, A.; Weng, J.; Chen, M.-R.; Xiao, F.; Liu, Y.; Yao, Y. Enabling privacy-preserving and publicly auditable federated learning. *arXiv* **2024**, arXiv:2405.04029.
21. Chen, J.; Xue, J.; Wang, Y.; Huang, L.; Baker, T.; Zhou, Z. Privacy-preserving and traceable federated learning for data sharing in industrial IoT. *Expert Syst. Appl.* **2023**, *213*, 119036. [CrossRef]
22. Gu, M.; Naraparaju, R.; Zhao, D. Enhancing data provenance and model transparency in federated learning systems: A database approach. *arXiv* **2024**, arXiv:2403.01451.
23. Ramirez, M.A.; Kim, S.-K.; Al Hamadi, H.; Damiani, E.; Byon, Y.-J.; Kim, T.-Y.; Cho, C.-S.; Yeun, C.Y. Poisoning attacks and defenses on artificial intelligence: A survey. *arXiv* **2022**, arXiv:2202.10276. [CrossRef]
24. Srivastava, M.; Kaushik, A.; Loughran, R.; McDaid, K. Data poisoning attacks in the training phase of machine learning models: A review. In Proceedings of the 32nd Irish Conference on Artificial Intelligence and Cognitive Science, Dublin, Ireland, 9–10 December 2024; pp. 64–75.
25. Gao, Y.; Doan, B.G.; Zhang, Z.; Ma, S.; Zhang, J.; Fu, A.; Nepal, S.; Kim, H. Backdoor attacks and countermeasures on deep learning: A comprehensive review. *arXiv* **2020**, arXiv:2007.10760.
26. Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z.B.; Swami, A. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*; ACM Computing Surveys: New York, NY, USA, 2017; pp. 506–519.
27. Wang, Z.; Ma, J.; Wang, X.; Hu, J.; Qin, Z.; Ren, K. *Threats to Training: A Survey of Poisoning Attacks and Defenses on Machine Learning Systems*; ACM Computing Surveys: New York, NY, USA, 2022. [CrossRef]
28. Zhang, R.; Hussain, S.; Chen, H.; Javaheripi, M. Systemization of Knowledge: Robust Deep Learning Using Hardware-software Co-design in Centralized and Federated Settings. *ACM Trans. Des. Autom. Electron. Syst.* **2023**, *28*, 1–32. [CrossRef]
29. Chen, C.; Liu, J.; Tan, H.; Li, X.; Wang, K.I.-K.; Li, P.; Sakurai, K.; Dou, D. Trustworthy federated learning: Privacy, security, and beyond. *Knowl. Inf. Syst.* **2025**, *67*, 2321–2356. [CrossRef]
30. Xia, Y.; Cheng, L. Robust federated learning under adversarial attacks: Survey and outlook. *ACM Comput. Surv.* **2023**, *55*, 1–39. [CrossRef]

31. Rahman, M.; Debnath, B. A survey on statistical poisoning detection in federated learning. *J. Netw. Comput. Appl.* **2024**, *238*, 104890.
32. Rathee, M.; Shen, C.; Wagh, S.; Popa, R.A. ELSA: Secure aggregation for federated learning with malicious actors. In *Proceedings of the 2023 IEEE Symposium on Security and Privacy (SP)*; IEEE: Piscataway, NJ, USA, 2023.
33. Ma, Y.; Woods, J.; Angel, S.; Polychroniadou, A.; Rabin, T. Flamingo: Multi-round single-server secure aggregation with applications to private federated learning. In *Proceedings of the 2023 IEEE Symposium on Security and Privacy (SP)*; IEEE: Piscataway, NJ, USA, 2023.
34. Lycklama, H.; Burkhalter, L.; Viand, A.; Küchler, N.; Hithnawi, A. RoFL: Robustness of secure federated learning. In *Proceedings of the 2023 IEEE Symposium on Security and Privacy (SP)*; IEEE: Piscataway, NJ, USA, 2023. [CrossRef]
35. Jiang, Y.; Zarezadeh, M.; Dai, T.; Köpsell, S. AlphaFL: Secure aggregation with malicious security for federated learning against dishonest majority. In Proceedings of the Privacy Enhancing Technologies, Washington, DC, USA, 14–19 July 2025; pp. 348–368. [CrossRef]
36. Ma, R.; Hwang, K.; Li, M.; Miao, Y. Trusted model aggregation with zero-knowledge proofs in federated learning. *IEEE Trans. Parallel Distrib. Syst.* **2024**, *35*, 2284–2296. [CrossRef]
37. Xu, B.; Wang, S.; Tian, Y. Efficient verifiable secure aggregation protocols for federated learning. *J. Inf. Secur. Appl.* **2025**, *80*, 104161. [CrossRef]
38. Bouamama, J.; Benkaouz, Y.; Ouzzif, M. Vesafl: Verifiable secure aggregation for privacy-preserving federated learning. *IEEE Trans. Dependable Secur. Comput.* **2025**, *22*, 6592–6609. [CrossRef]
39. Bottoni, S.; Zizzo, G.; Braghin, S.; Trombetta, A. Verifiability and privacy in federated learning through context-hiding multi-key homomorphic authenticators. In *Proceedings of the IEEE/ACM 12th International Conference on Big Data Computing, Applications and Technologies (BDCAT '25), Nantes, France, 1–4 December 2025*; ACM: New York, NY, USA, 2025; pp. 1–6. [CrossRef]
40. Ning, Z.; Li, C.; Xu, Y. Blockchain-enabled accountable federated learning for edge AI systems. *IEEE Internet Things J.* **2024**, *11*, 10325–10337.
41. Liu, Z.; Wang, Y.; Tang, H. Verifiable and accountable federated learning via permissioned blockchain. *Future Gener. Comput. Syst.* **2024**, *155*, 521–535. [CrossRef]
42. Ning, W.; Zhu, Y.; Song, C.; Li, H.; Zhu, L.; Xie, J.; Chen, T.; Xu, T.; Xu, X.; Gao, J. Blockchain-based federated learning: A survey and new perspectives. *Appl. Sci.* **2024**, *14*, 9459. [CrossRef]
43. Cen, X.; Alur, R. Auditable AI systems: Foundations and techniques. *Commun. ACM* **2024**, *67*, 76–88.
44. Miguel, E.; Sanchez, J.; Ortega, P. Accountability in artificial intelligence: From principles to practice. *AI Ethics* **2021**, *1*, 43–59.
45. Malgieri, G.; Pasquale, F. The emerging principle of accountability in AI regulation. *Comput. Law. Secur. Rev.* **2022**, *45*, 105701.
46. *ISO/IEC 42001*; Artificial Intelligence Management System (AIMS)—Requirements. International Organization for Standardization: Vernier, Switzerland, 2023.
47. Zhang, Y.; Jia, R.; Pei, H.; Wang, W.; Li, B.; Song, D. The secret revealer: Generative model-inversion attacks against deep neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 253–261.
48. Biggio, B.; Nelson, B.; Laskov, P. Support vector machines under adversarial label noise. In Proceedings of the Asian Conference on Machine Learning (PMLR), Taoyuan, Taiwan, 14–15 November 2011; pp. 97–112.
49. Bonawitz, K.; Kairouz, P.; McMahan, B.; Ramage, D. Federated learning and privacy: Building privacy-preserving systems for machine learning and data science on decentralized data. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, Online, 15–19 November 2021. [CrossRef]
50. Bhagoji, A.N.; Chakraborty, S.; Mittal, P.; Calo, S. Analyzing federated learning through an adversarial lens. In Proceedings of the International Conference on Machine Learning (PMLR), Long Beach, CA, USA, 9–15 June 2019; pp. 634–643.
51. Li, Y.; Jiang, Y.; Li, Z.; Xia, S.-T. Backdoor learning: A survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *35*, 5–22. [CrossRef] [PubMed]
52. Bagdasaryan, E.; Veit, A.; Hua, Y.; Estrin, D.; Shmatikov, V. How to backdoor federated learning. In Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics (AISTATS), Sicily, Italy, 26–28 August 2020; Volume 108, pp. 2938–2948.
53. Mazzone, F.; Badawi, A.A.; Polyakov, Y.; Everts, M. Investigating privacy attacks in the gray-box setting to enhance collaborative learning schemes. *arXiv* **2024**, arXiv:2409.17283.
54. Zhang, Y.; Bai, G.; Chamikara, M.; Ma, M. Agrevader: Poisoning membership inference against byzantine-robust federated learning. In Proceedings of the ACM Asia Conference on Computer and Communications Security, Melbourne, Australia, 10–14 July 2023. [CrossRef]
55. Sakhnovych, Y. Black-Box Model Watermarking in Federated Learning. Ph.D. Thesis, TU Wien, Vienna, Austria, 2024.

56. Shafahi, A.; Huang, W.R.; Najibi, M.; Suciu, O.; Studer, C.; Dumitras, T.; Goldstein, T. Poison frogs! targeted clean-label poisoning attacks on neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 3–8 December 2018.
57. Shah, A.; Ahmad, A.; Ali, B.; Anwer, S. *Guarding the Gates: A Comprehensive Survey of Backdoor Attacks on Neural Networks*; SSRN: Rochester, NY, USA, 2024. [CrossRef]
58. Zhang, J.; Chen, B.; Cheng, X.; Binh, H.T.T.; Yu, S. PoisonGAN: Generative poisoning attacks against federated learning in edge computing systems. *IEEE Internet Things J.* **2020**, *8*, 3310–3322. [CrossRef]
59. Weng, C.-H.; Lee, Y.-T.; Wu, S.-H. On the trade-off between adversarial and backdoor robustness. In Proceedings of the Advances in Neural Information Processing Systems 33 (NeurIPS), Online, 6–12 December 2020.
60. Pan, M.; Zeng, Y.; Lyu, L.; Lin, X.; Jia, R. Asset: Robust backdoor data detection across a multiplicity of deep learning paradigms. In Proceedings of the 32nd USENIX Security Symposium, Anaheim, CA, USA, 9–11 August 2023.
61. Xu, C.; Liu, W.; Zheng, Y.; Wang, S.; Chang, C.-H. An imperceptible data augmentation based blackbox clean-label backdoor attack on deep neural networks. *IEEE Trans. Circuits Syst. Regul. Pap.* **2024**, *70*, 5011–5024. [CrossRef]
62. De Gaspari, F.; Hitaj, D.; Mancini, L.V. Have you poisoned my data? Defending neural networks against data poisoning. In *Computer Security—ESORICS 2024*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 85–104. [CrossRef]
63. Bena, N.; Anisetti, M.; Damiani, E.; Yeun, C.Y.; Ardagna, C.A. Protecting machine learning from poisoning attacks: A risk-based approach. *Comput. Secur.* **2025**, *155*, 104468. [CrossRef]
64. Xie, C.; Huang, K.; Chen, P.-Y.; Li, B. DBA: Distributed backdoor attacks against federated learning. In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.
65. Wan, Y.; Qu, Y.; Ni, W.; Xiang, Y.; Gao, L. Data and model poisoning backdoor attacks on wireless federated learning, and the defense mechanisms: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2024**, *26*, 1861–1897. [CrossRef]
66. Shejwalkar, V.; Houmansadr, A. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In Proceedings of the 28th Annual Network and Distributed System Security Symposium (NDSS), Online, 21–25 February 2021. [CrossRef]
67. El Mhamdi, E.M.; Guerraoui, R.; Rouault, S. The hidden vulnerability of distributed learning in byzantium. In Proceedings of the 35th International Conference on Machine Learning (ICML) (PMLR), Stockholm, Sweden, 10–15 July 2018; pp. 3521–3530.
68. Chen, L.; Liu, X.; Wang, A.; Zhai, W.; Cheng, X. FLSAD: Defending backdoor attacks in federated learning via self-attention distillation. *Symmetry* **2024**, *16*, 1497. [CrossRef]
69. Rocha, A.; Conti, M. Weidetect: Weibull distribution-based defense against poisoning attacks in federated learning for network intrusion detection systems. *arXiv* **2025**, arXiv:2504.04367.
70. Jagielski, M.; Oprea, A.; Biggio, B.; Liu, C.; Nita-Rotaru, C.; Li, B. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In Proceedings of the IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 21–23 May 2018; pp. 19–35. [CrossRef]
71. Sun, Z.; Liu, C.; Yang, Q.; Qi, Y. Data poisoning attacks on federated machine learning. *ACM Trans. Knowl. Discov. Data* **2021**, *15*, 1–25. [CrossRef]
72. Paudice, A.; Muñoz-González, L.; Lupu, E.C. Label sanitization against label flipping poisoning attacks. *IEEE Access* **2018**, *6*, 5423–5431.
73. Gu, T.; Dolan-Gavitt, B.; Garg, S. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv* **2017**, arXiv:1708.06733.
74. Chen, L.; Ye, Y.; Bourlai, T. Adversarial machine learning in malware detection: Arms race between evasion attack and defense. In Proceedings of the 2017 European Intelligence and Security Informatics Conference (EISIC), Athens, Greece, 11–13 September 2017; pp. 99–106. [CrossRef]
75. Koh, P.W.; Liang, P. Understanding black-box predictions via influence functions. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017.
76. Mozaffari Kermani, M.; Sur-Kolay, S.; Raghunathan, A.; Jha, N.K. Systematic poisoning attacks on and defenses for machine learning in healthcare. *IEEE J. Biomed. Health Inform.* **2015**, *19*, 1893–1905. [CrossRef] [PubMed]
77. Kroll, J.A.; Huey, J.; Barocas, S.; Felten, E.W.; Reidenberg, J.R.; Robinson, D.G.; Yu, H. Accountable algorithms. *Univ. Pa. Law. Rev.* **2017**, *165*, 633–705.
78. Miguel, J.; Chen, T. Machine learning provenance for accountability. In Proceedings of the USENIX Symposium on Operating Systems Design and Implementation, Virtual Event, 14–16 July 2021.
79. Cao, D.; Chang, S.; Lin, Z.; Liu, G. Understanding distributed poisoning attack in federated learning. In Proceedings of the IEEE Conference on Communications and Network Security (CNS), Washington, DC, USA, 10–12 June 2019.
80. Ma, Z.; Ma, J.; Miao, Y.; Li, Y. Shieldfl: Mitigating model poisoning attacks in privacy-preserving federated learning. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 1639–1654. [CrossRef]

81. ElZemity, A.; Arief, B. Privacy threats and countermeasures in federated learning for Internet of Things: A systematic review. In *Proceedings of the 2024 IEEE Conference on Communications and Network Security (CNS)*; IEEE: Piscataway, NJ, USA, 2024.
82. Fung, C.; Yoon, C.J.M.; Beschastnikh, I. Mitigating sybils in federated learning poisoning. *arXiv* **2018**, arXiv:1808.04866.
83. Yin, D.; Chen, Y.; Kannan, R.; Bartlett, P. Byzantine-robust distributed learning: Towards optimal statistical rates. In Proceedings of the 35th International Conference on Machine Learning (ICML), Stockholm, Sweden, 10–15 July 2018; pp. 5650–5659.
84. Awan, S.; Luo, B.; Li, F. CONTRA: Defending against poisoning attacks in federated learning. In Proceedings of the 26th European Symposium on Research in Computer Security (ESORICS 2021), Darmstadt, Germany, 4–8 October 2021; pp. 455–475. [CrossRef]
85. Blanchard, P.; El Mhamdi, E.M.; Guerraoui, R.; Stainer, J. Machine learning with adversaries: Byzantine tolerant gradient descent. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017.
86. Cao, X.; Fang, M.; Liu, J.; Gong, N.Z. FLTrust: Byzantine-robust federated learning via trust bootstrapping. *arXiv* **2020**, arXiv:2012.13995.
87. Tran, B.; Li, J.; Madry, A. Spectral signatures in backdoor attacks. In Proceedings of the NeurIPS, Montréal, QC, Canada, 3–8 December 2018.
88. Wang, B.; Yao, Y.; Shan, S.; Li, H.; Viswanath, B.; Zheng, H.; Zhao, B.Y. Neural Cleanse: Identifying and mitigating backdoor attacks in neural networks. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 20–22 May 2019.
89. Gao, Y.; Xu, C.; Wang, D.; Chen, S.; Ranasinghe, D.C.; Nepal, S. STRIP: A defence against trojan attacks on deep neural networks. In Proceedings of the ACSAC '19: Proceedings of the 35th Annual Computer Security Applications Conference, San Juan, PR, USA, 9–13 December 2019. [CrossRef]
90. Gao, X.; Fu, S.; Liu, L.; Luo, Y. BVDFed: Byzantine-resilient and verifiable aggregation for differentially private federated learning. *Front. Comput. Sci.* **2024**, *18*, 185810. [CrossRef]
91. Allouah, Y.; Guerraoui, R.; Stephan, J. Towards trustworthy federated learning with untrusted participants. In Proceedings of the 42nd International Conference on Machine Learning (ICML), Vancouver, BC, Canada, 13–19 July 2025; Volume 267, pp. 1184–1227.
92. Tounsi, A.; Salem, O.; Mehaoua, A. Anomaly detection in federated learning: A comprehensive study on data poisoning and energy consumption patterns in IoT devices. In Proceedings of the 2024 7th Conference on Cloud and Internet of Things (CIoT), Montreal, QC, Canada, 29–31 October 2024; pp. 1–8. [CrossRef]
93. Li, D.; Wong, W.E.; Wang, W.; Yao, Y.; Chan, M.C. Detection and mitigation of label-flipping attacks in federated learning systems with KPCA and k-means. In Proceedings of the 8th International Conference on Dependable Systems and Their Applications (DSA), Yinchuan, China, 5–6 August 2021; pp. 551–559.
94. Chen, X.; Zan, D.; Li, W.; Guan, B. A gan-based data poisoning framework against anomaly detection in vertical federated learning. In Proceedings of the ICC 2024 - IEEE International Conference on Communications, Denver, CO, USA, 9–13 June 2024. [CrossRef]
95. Alsulaimawi, Z. Federated learning with anomaly detection via gradient and reconstruction analysis. *arXiv* **2024**, arXiv:2403.10000.
96. Khraisat, A.; Alazab, A.; Alazab, M.; Jan, T.; Singh, S. Securing federated learning: A defense strategy against targeted data poisoning attack. *Discov. Internet Things* **2025**, *5*, 16. [CrossRef]
97. Zhu, W.; Liu, Z.; Chen, Z.; Shi, C.; Zhang, X.; Guo, S. FedValidate: A robust federated learning framework based on client-side validation. In Proceedings of the 2023 8th International Conference on Data Science in Cyberspace (DSC), Hefei, China, 18–20 August 2023; pp. 337–344. [CrossRef]
98. Zhang, X.; Kim, M.; Kumar, R. FLCert: Federated certification via client grouping and voting. *arXiv* **2022**, arXiv:2210.00584.
99. Hakeem, S.; Kim, H. Advancing intrusion detection in V2X networks: A comprehensive survey on machine learning, federated learning, and edge AI for V2X security. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 11137–11205. [CrossRef]
100. Liu, X.; Li, H.; Xu, G.; Chen, Z.; Huang, X.; Lu, R. Privacy-enhanced federated learning against poisoning adversaries. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 4574–4588. [CrossRef]
101. Panda, A.; Mahloujifar, S.; Nitin Bhagoji, A.; Chakraborty, S.; Mittal, P. SparseFed: Mitigating model poisoning attacks in federated learning with sparsification. In Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS), Virtual, 28–30 March 2022; pp. 7587–7624.
102. Zeng, Q.; Zhou, L.; Li, X. Decentralized and auditable federated learning using blockchain and smart contracts. *Inf. Sci.* **2024**, *656*, 119982. [CrossRef]
103. Gill, W.; Anwar, A.; Gulzar, M.A. Tracefl: Interpretability-driven debugging in federated learning. *arXiv* **2023**, arXiv:2312.13632.
104. Wu, C.; He, K.; Chen, J.; Zhao, Z.; Du, R. Liveness is not enough: Enhancing fingerprint authentication with behavioral biometrics to defeat puppet attacks. In Proceedings of the 29th USENIX Security Symposium (USENIX Security 20), Boston, MA, USA, 12–14 August 2020; pp. 2219–2236.
105. Li, S.; Ngai, E.C.H.; Voigt, T. An experimental study of Byzantine-robust aggregation schemes in federated learning. *IEEE Trans. Big Data* **2023**, *10*, 975–988. [CrossRef]

106. Wu, C.; Chen, J.; Wang, Z.; Liang, R.; Du, R. Semantic Sleuth: Identifying Ponzi contracts via large language models. In Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering (ASE '24), Sacramento, CA, USA, 27 October–1 November 2024; pp. 582–593. [CrossRef]
107. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and open problems in federated learning. *Found. Trends Mach. Learn.* **2021**, *14*, 1–210. [CrossRef]
108. Truex, S.; Liu, L.; Gursoy, M.E.; Yu, L.; Wei, W. A hybrid approach to privacy-preserving federated learning. In Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security 1 (AISec), London, UK, 15 November 2019; pp. 1–11. [CrossRef]
109. Yao, J.; Shen, J.; Wu, Y.; Zhang, R. AERO: Efficient and verifiable secure aggregation in federated learning. In Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS), Hong Kong, China, 18–21 July 2023; pp. 1012–1023. [CrossRef]