Auditing Privacy Claims in Federated Learning for IoT Intrusion Detection: A Systematic Review of Threat Models and Evidence

Shah Zaib, Khairullah Khan, Muhammad Javed
Department of Computer Science, University of Science and Technology, Bannu, Pakistan
*Correspondence: shahzebbiseb@gmail.com

Citation | Zaib. S, Khan. K, Javed. M, "Auditing Privacy Claims in Federated Learning for IoT Intrusion Detection: A Systematic Review of Threat Models and Evidence", IJIST, Vol. 8 Issue.9 pp 3619-3662, October 2026
DOI:
Received| Oct 01, 2026 **Revised|** Oct 17, 2026 **Accepted|** Oct 21, 2026 **Published|** Oct 25, 2026.

Federated Learning (FL) is increasingly used for intrusion detection in Internet of Things (IoT) networks because it enables collaborative model training without sharing raw telemetry. However, keeping data local does not, by itself, guarantee privacy. This review examines whether privacy claims in FL-based IoT intrusion detection are supported by clearly defined threat models and appropriate evidence. Peer-reviewed studies published between 2017 and September 2026 were reviewed. The search identified 1,164 records; after removal of 327 records, 837 records were screened, 382 were excluded, and 485 reports were sought for retrieval. Of these, 395 reports were assessed for eligibility and 259 were excluded, leaving 136 studies for the final review. The review also considered confidentiality attacks, poisoning and Byzantine behavior, non-independent and non-identically distributed (non-IID) data, continual adaptation, evaluation practices, and deployment constraints. Three recurring weaknesses emerged. First, many studies treated decentralized training as a privacy guarantee without clearly defining privacy or the threat model. Second, confidentiality was often assessed separately from integrity and availability. Third, strong detection results on centrally partitioned benchmark datasets were frequently presented as evidence of federated readiness despite unrealistic data partitioning. Based on these findings, we propose an evidence-based taxonomy, a privacy-claim auditing template, and a staged research roadmap. The findings indicate that credible privacy-preserving FL intrusion detection requires explicit identification of protected information and adversaries, threat-model-consistent attack evaluation, transparent reporting of privacy–utility–resource trade-offs, and realistic consideration of temporal change, malicious clients, data heterogeneity, and communication constraints. Meaningful privacy protection should therefore be demonstrated through threat-aware and deployment-realistic evaluation rather than inferred from the mere use of privacy mechanisms.

Keywords: Federated Learning; Intrusion Detection; Internet of Things; Privacy; Gradient Leakage; Differential Privacy; Secure Aggregation; Poisoning; Non-IID Data; Systematic Review

Introduction:

Improving intrusion detection across Internet of Things (IoT) environments often requires collaboration. A model trained within a single household, factory, hospital, vehicle fleet, or network segment sees only a limited range of normal behavior and malicious activity. Combining information from multiple locations may improve attack detection; however, the resulting network intelligence can contain sensitive and potentially identifying information. Packet traces, data flows, and device identifiers may reveal details on people's movements within an organization and their relationship with other organizations (businesses). Centralizing this information may also create an attractive target for attackers and increase the risk of unauthorized disclosure or violations of contractual and cross-border data-transfer requirements. Federated learning (FL) allows for training of a shared model while keeping raw observations with their owners [1]. However, storing information locally is not sufficient to address privacy concerns.

The information shared while doing a federated learning session may also contain confidential data. Gradients may reveal individual data points or attributes, while model parameters and outputs may enable membership inference and property inference attacks. An honest but curious or malicious coordinator might also manipulate the training process to extract more information, while changes in a client's participation over time may reveal behavioral changes [2][3][4][5]. However, inference attacks are only one aspect of this problem. Malicious actors could poison the global detector, inject backdoors, create sybils and disrupt learning through withholding information. However, mechanisms hiding clients' updates will be more difficult for servers to identify anomalous contribution and aggregate data effectively enough. Calling a detector "federated" is very vague as to what kind of security that provides. Privacy and security should be assessed against clearly defined threat actors. Detection effectiveness and the computational limitations of IoT devices should also be considered.

They are especially relevant to intrusion detection since accuracy is critical as it depends upon experiment design. Traffic on networks tends to be unbalanced; it changes with respect to time and varies greatly depending upon device type and environment. The preprocessing steps may also affect outcomes significantly enough. For instance, randomly shuffling an aggregated dataset might put data for identical devices/sessions together during training and testing phases so that models can recognize patterns which aren't present when deployed. Also, simulated clients generated through random assignment of row numbers are not necessarily representing devices, gateways, or organizations that have significantly varied traffic patterns. High overall accuracy may still be maintained if rare attack types are often overlooked and false positives aren't acceptable for operation. A defense which stops data recovery from an honest but curious server might offer some protection to adaptive poisoning. Conversely, strong aggregation might ignore odd yet valid notifications of uncommon gadgets and new products. Privacy, safety; detection effectiveness; equity and energy use should all be considered simultaneously instead of individually.

Previous reviews have been conducted on FL architectures, aggregation techniques, IoT security applications, and privacy technologies [6][7][8][9][10][11]. The research shows how far this area is from being an initial demonstration for federated intrusion detection systems. Nevertheless, most reviews focus on algorithm-based literature and specific techniques like DP, secure aggregation, homomorphic encryption, blockchain, personalization, and robust aggregation. They rarely begin by asking whether the available evidence actually supports the privacy claim being made. As a result, studies using different adversaries and threat assumptions may be grouped together even when their security properties are not directly comparable. A system may be described as privacy-preserving simply because it uses a recognized privacy mechanism, even when its parameters, observable

information, or residual leakage are not reported. Similarly, experimental results may appear comparable despite substantial differences in client construction, attack conditions, and communication or computational budgets. A recent survey [12] of collaborative intrusion detection (CID) in SD-WAN environments further highlights constraints in architecting, operating and benchmarking federated IDS, again stressing the importance of linking privacy claims to explicit assumptions regarding system and evidence.

This review addresses this gap by focusing on privacy-preserving federated learning for network- and behavior-based intrusion or anomaly detection in IoT and industrial IoT environments. Centralized intrusion detection studies are included only when they introduce datasets or evaluation practices that have substantially influenced this research area. We also consider general studies of FL privacy and robustness when they provide attacks, guarantees, or defensive methods needed to evaluate privacy-preserving FL intrusion detection systems (PPFL-IDSs). Purely cryptographic communication protocols without a learning-based detector, as well as privacy-preserving analytics that do not use federated training, are outside the scope of this review. The review covers research published from 2017, when modern FL was formally introduced, through September 2026.

The review makes four main contributions. First, it assesses the scope and maturity of PPFL-IDS research through an evidence-based taxonomy. Second, it introduces a framework for auditing privacy claims by linking the protected asset and privacy unit to the assumed adversary, observable information, selected mechanism, stated guarantee, supporting evidence, residual risk, and system cost. Third, it examines how confidentiality, integrity, availability, data heterogeneity, and continual adaptation interact, and how these relationships affect the credibility of experimental and deployment claims. Finally, it develops a dependency-aware research roadmap to support more comparable evaluations and more realistic deployment of PPFL-IDSs.

Accordingly, the review addresses four research questions.

RQ1: How has privacy-preserving FL-based intrusion detection for IoT and IIoT evolved, and what is the scope and evidential maturity of the field?

RQ2: How are privacy claims defined and supported, and which mechanisms protect which assets against which adversaries and residual risks?

RQ3: How do integrity and availability threats, non-IID heterogeneity, personalization, participation, and continual learning interact with privacy and detection performance?

RQ4: How credible, comparable, and deployment-relevant is the experimental evidence, and which research gaps should be prioritized?

Research Objectives:

The goals of this systematic review were explicitly written based on the four research questions and were devised to enable an evidence-based evaluation of privacy-preserving federated learning for IoT intrusion detection.

Objective 1: To describe and quantify the evolution, range and evidential maturity of privacy preserving federated learning-based intrusion detection in IoT and IIoT environments, such as application framework, learning scenarios, datasets and assessment of performance, related to RQ1.

Objective 2: To identify and critically audit privacy claims of PPFL-IDS studies based on the following: protection of the asset, privacy unit, assumed adversary, observable information, protection mechanism, stated guarantee, supporting evidence, and residual risk, corresponding to RQ2.

Objective 3: To investigate the interaction between integrity and availability threats, non-IID data, personalization, client participation and continuous learning with privacy protection and intrusion-detection performance, corresponding to RQ3.

Objective 4: To determine the credibility, comparability, reproducibility, and deployment relevance of available evidence; and to identify research gaps and priorities related to more realistic evaluation and deployment of PPFL-IDS, aligned with RQ4.

Novelty Statement:

The novelty of this review is not the offering of a new list of PPFL-IDS algorithms or privacy mechanisms, but first a systematic analysis of whether the evidence reported by PPFL-IDS studies adequately supports their privacy claims to support privacy claims.

Earlier reviews of federated-learning-based intrusion detection have primarily organized the literature according to architectures, learning models, aggregation methods, application domains, or broader security and privacy mechanisms. This review, on the other hand, presents a privacy-privacy claim auditing approach based on evidence that establishes a common privacy claim chain: protected asset → privacy unit → adversary → observable information → mechanism → stated guarantee → supporting evidence → residual risk and system cost.

This way, you capture the concept of federated data locality in contrast to more stringent requirements like confidentiality, formal privacy guarantees and inference resistance. It also assesses privacy evidence based on the adversary and the observable information used in asserting the claim, instead of seeing the existence of any named privacy mechanism as providing sufficient evidence. Additionally, alongside integrity, availability, non-IID data, participation, continual adaptation, and deployment constraints, privacy is analyzed.

The analytical focus is thus shifted from a question of "Which privacy mechanisms are used in PPFL-IDS?" to a question of, "What is actually being protected, by whom, by what mechanisms, with which assumptions, and what evidence?" The primary methodological novelty of this claim-to-evidence perspective is represented by this type of approach.

The remainder of the article proceeds from concepts to audit. Conceptual and analytical foundations define the system, adversaries, privacy claims, and evidence levels. Systematic review methodology describes the review protocol and appraisal scheme. Evidence landscape and integrated taxonomy synthesize the evidence through an integrated taxonomy. Privacy mechanisms and evidence audit, together with integrity and availability threats, examine confidentiality mechanisms and threats to integrity and availability. Experimental evidence and deployment realism evaluate the reliability of experimental findings and their relevance to real-world deployment. Research gaps and prioritized roadmap identify key research gaps and prioritize future research directions. Finally, the conclusion summarizes the main findings and contributions of the article. The abbreviations used throughout the survey are listed in Table 1.

Table 1. Abbreviation List

Abbreviation	Definition	Abbreviation	Definition
CPS	Cyber-Physical System	NIDS	Network Intrusion Detection System
DP	Differential Privacy	non-IID	Non-independent and non-identically distributed
FL	Federated Learning	PPFL	Privacy-Preserving Federated Learning
FL-IDS	Federated Learning–based Intrusion Detection System	PPFL-IDS	Privacy-Preserving Federated Learning–based Intrusion Detection System
IDS	Intrusion Detection System	PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
IID	Independent and Identically Distributed	PRISMA-S	PRISMA extension for reporting literature searches

IIoT	Industrial Internet of Things	RQ/RQs	Research Question/Research Questions
IoT	Internet of Things	TEE	Trusted Execution Environment
MPC	Multi-Party Computation	FedAvg	Federated Averaging

Conceptual and Analytical Foundations:

PPFL-IDS as a Socio-Technical System:

A PPFL-IDS works within an extended scope of pipelines not just for customers, aggregators and classifiers. It starts with a sensor stage where endpoints and gateways collect information about packet data; flow information; log files from systems; radio signals etc.; control of processes. Every customer translates this information to local training samples; then modifies their detectors; sends models' parameters, features and metrics or ciphertext. The coordination can be done by means of a side server; cloud computing service; consortium nodes and peer-to-peer protocols. Coordination is done by a group member who gathers input from others; this information is then combined with that of other members to create an updated version of the model. After deployment, the alert system sends notifications to users of devices; security officers; automated responses and regulators. Identity management, authentication, traceability of updates; software supply chain security and incident response are supported throughout this process.

Looking at the complete pipeline also broadens how privacy should be understood. Raw packets may remain on a device, yet its model updates can still reveal behavioral patterns. Secure aggregation can prevent the server from viewing an individual update, but the subsequently released global model may still be vulnerable to membership inference. Likewise, local differential privacy can reduce information leakage without hiding when a particular client participates. A trusted execution environment may protect one stage of the computation while introducing additional assumptions about hardware security and remote attestation. The appropriate privacy unit also depends on the application. It may be an individual packet or flow, a session, a user, a device, an organization, or a client's complete participation history. Therefore, a guarantee defined using record-level adjacency should not be described as providing device-level or client-level privacy unless that broader protection has been explicitly defined and supported by evidence. Figure 1 illustrates the PPFL-IDS system boundary and the principal points at which information may be observed, inferred, or manipulated across the federated training and deployment pipeline.

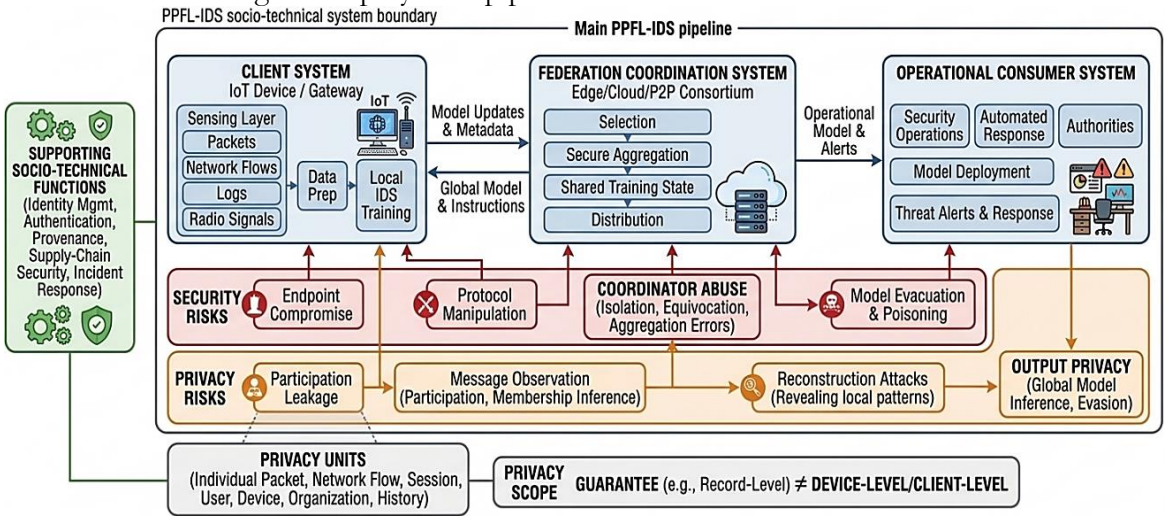

Figure 1. PPFL-IDS system boundary and principal attack surfaces

The objective of an IDS also influences its privacy risk. Supervised multiclass detectors learn class-specific decision boundaries from labeled attacks, while anomaly detectors build a profile of normal behaviour. Autoencoders and sequence models may capture fine-grained patterns in network traffic, and continual or personalized systems can retain local or temporal state. These differences affect the extent and type of information that may be exposed through model updates. Gradient leakage, for example, depends on factors such as model architecture, batch size, loss function, update frequency, initialization, and the attacker's knowledge.

The operational IDSs function within an evolving setting. Alerts may change networks' behavior; defenders might create new signature types; attackers could adjust to these changes, while label arrivals are delayed and ambiguous. Feedback impacts may alter how models learn things to know more about them through updates. Privacy claims which do not consider this training and operation scenario are incomplete.

Federated Configurations and Heterogeneity:

In horizontal federated learning, participants share a common feature space but hold distinct datasets. It's also a popular area for PPFL-IDS studies since gateway and organization can easily get similar flows out of different traffic streams. Vertical FL is used when participants have different attributes for the same or overlapping entities. for instance, network, host, identity and applications might be viewed differently by them regarding an incident. This partnership needs to be aligned with a company's interests; it also involves proper attribution for features that may raise more security concerns. Federated transfer learning is more flexible since there are no strict requirements for data similarity nor feature space overlap between models trained on different datasets. It may be able to facilitate inter-vendor and cross-functional cooperation; however, it is currently not applicable for IoT intrusion detection. Other types vary more on their organizational structure for coordinating activities together. Hierarchical FL employs intermediate edge aggregators for reducing wide area communications and maintaining some level of locality; decentralized FL distributes aggregation among peers. While decentralized FL reduces dependence upon one central server, it is harder to manage trust and models consistency.

However, it's more than that to build a federation. Localization differences also affect learning significantly enough to be important for training purposes. Quantity skew gives more power to bigger data sets; label skew is when some users see most harmless traffic and very little attack activity. Distributions may differ due to types of devices used; protocol selection; software versioning; location where data is collected and processed priorly. The meanings and distributions might vary with respect to changes that occur later on. A concept change alters relationship between feature observations and attacks; while a covariate shift affects types of network flows. With time as they build up models might become less accurate; leaks would not be as good as before when tested initially.

They have been especially significant for federated intrusion detection. Since FedAvg weights local contributions according to sample size, big sites may overpower the global model despite having fewer users operating safety critical devices. Personalization, client clustering, proximal optimization, knowledge distillation, and incremental learning may be helpful to tackle them. Nevertheless, every method alters how data gets shared while learning; it might be more vulnerable to breaches of confidentiality and safety issues. They must then also be assessed by their precision/accuracy and security level rather than just whether it converges properly enough to protect against attacks successfully.

From Privacy Intuition to an Auditable Claim:

Privacy statements must first define what information will be safeguarded and to whom, a database, individual person, a computer system. Then it will have to know who is playing an adversarial role; what data they have at their disposal; and what information needs to be kept confidential. If "protection" refers to limits for data leaks then it must also include

definitions of what constitutes privacy along with protocols and trust assumptions. The evidence is relevant to evaluate security as per that particular condition.

Making them clear prevents various types of assurances being reduced to an "privacy preserving" label. Locality of data is when a record stays with you where it came from during learning process. Confidentiality is concerned with whether an individual has access to their own confidential information that they have been permitted by law and policy for viewing. Privacy formalizes this concept through something called "differential privacy". Also, low attack success rates are only valid for those specific circumstances that were tested and do not show resilience against other types of attacks nor more powerful ones. Compliance with regulations is an independent issue that cannot be deduced solely based upon architecture of software systems.

FL mainly affects how information gets processed during learning; therefore it supports locality of data. Confidentiality is not guaranteed by default; it also doesn't guarantee formal privacy nor resistance against inference/attack/reconstruction attacks and regulations. Every claim needs to be outlined with definition; assumption; and proof. Table 2 summarizes the minimum elements required to formulate an auditable privacy claim in a PPFL-IDS, including the protected asset, privacy unit, adversary, observable information, guarantee, evidence, and residual risk.

Table 2. Minimum Structure of an Auditable Privacy Claim

Field	Question to answer	Typical PPFL-IDS options	Failure if omitted
Protected asset	What information is sensitive?	Traffic record, label, identity, attribute, update, model, metadata	Privacy becomes undefined
Privacy unit	Whose or which contribution changes?	Record, event, session, user, device, site, client history	Guarantee may protect the wrong unit
Adversary	Who attacks, and with what capability?	Eavesdropper, curious/malicious server, client, coalition, model querier	Evidence cannot be scoped
Observables	What can the adversary inspect or manipulate?	Gradient, update, aggregate, checkpoint, output, timing, participant set	Attack surface is understated
Mechanism	What removes, transforms, hides, or bounds information?	Minimization, DP, secure aggregation, encryption, TEE, representation sharing	A label substitutes for a design
Guarantee	What property follows under which assumptions?	Formal bound, cryptographic confidentiality, empirical resistance, locality only	Strength is overstated
Evidence	What theorem, attack, baseline, and metric test the claim?	Accounting, reconstruction, membership, ablation, systems measurement	Claim is not falsifiable
Residual risk/cost	What remains exposed and what is paid?	Metadata, collusion, final-model leakage, utility, latency, energy, availability	Protection is presented as cost-free

The information that needs protection extends well beyond raw network traffic. Labels, device identities, sensitive attributes, client membership, local model states, update values, and operational metadata may also be confidential. The relevant adversary depends on the deployment and may include an honest-but-curious coordinator, a malicious server, a passive eavesdropper, colluding clients, an external user querying the model, or an attacker controlling an endpoint. These actors may not have access to the same information.

Depending on the threat model, they may observe individual gradients, updates collected across multiple training rounds, aggregated values, model checkpoints, prediction outputs, participation schedules, or side-channel signals. A privacy mechanism is meaningful only when its protection is aligned with the asset, adversary, and observable information defined in the threat model.

For example, encrypted transport protects updates from interception while they are being transmitted, but it does not prevent the receiving server from examining those updates. Secure aggregation can hide an individual client's contribution from the coordinator when its participation threshold and collusion assumptions are satisfied. However, it does not necessarily limit what can be inferred from the resulting aggregate or from a released model. Differential privacy (DP) can provide a formal bound on information leakage, but the presence of DP alone does not establish the strength or meaning of that guarantee. Its interpretation depends on factors such as the adjacency definition, clipping rule, sampling procedure, privacy accountant, composition across training rounds, and the reported values of ε and δ [13][14].

Confidentiality, Integrity, and Availability are Coupled:

Classical separation of security properties helps with analyses; however, they tend to be too narrow from an operational standpoint. Confidentiality mechanisms may help to decrease information required by auditors during testing processes. Aggregation could be based upon distance, update sign, normality and trusted verification information that might not always be available due to encryption of updates and exposure of confidential details by them. Clips help to restrict both differential-privacy sensitivity and the size of attack updates, but too much clipping may also stifle legitimate input from minority customers. Also, noise may help to prevent attack reconstruction attacks better by reducing detection rates for rare events and hiding indicators that are used during models' poisoning tests. Client selection may decrease power usage by excluding unreliable customers; however, it might also be used to identify activities through participation rates along with repeated exclusions that are biased towards some groups of users only.

Threat interactions also matter much more. Malicious servers can transmit various models on various client machines; they might monitor these changes themselves, or alter group members for more data leaks/attacks success rates. Colluding customers might try isolating someone else's contribution within an anonymous pool if there are few privacy sets available. A Sybil attacker might be able to evade defenses assuming every participant is a separate entity. A denial-of-service attack and client dropping can impact not just availability but also lower thresholds for security, and introduce biases to privacy mechanisms through selection.

Therefore, for evaluation of one defense system with respect to another where every participant is assumed to act truthfully gives you information on that particular system but not necessarily how well it performs generally as an ensemble. This does not guarantee safety nor confidentiality of an entire architecture. The Meaningful PPFL-IDS assessment must also take account of various defense/threats interactions with real-world scenarios for deployments.

Evidence Levels:

The evidence for privacy and security is classified at four levels. At E0, support is limited to an architectural statement like stating that raw data are local or identifying a protection mechanism. E1 is that the mechanism is implemented; it describes parameters/protocols for this system of protection which has not been tested yet. E2 needs to be evaluated empirically with respect to some given adversarial attack, including what we know about it as an attacker. E3 applies if there is an official/formal/compositional assurance provided by assumption with respect to parameter values utilized during implementation. The empirical tests are still relevant for E3 since there is no assurance that they will work as expected from theory alone.

The deployment evidence has been evaluated individually since none of them shows if an approach satisfies real-world needs like response time (latency), power usage, network traffic load, ease-of-use/maintenance and sensitivity to threats.

Separating them helps to avoid conflating fundamentally dissimilar kinds of evidence that are not interchangeable. A research paper might offer an excellent theoretical assurance of data security but provide no practical proof it could work well with low-power gadgets. On the other hand, it might also be assessed by means of practicality and yet remain susceptible for honest but curious coordinators. Thus, our synthesis captures those strengths and weaknesses separately instead of aggregating them under an aggregate rating system.

Systematic Review Methodology:

Protocol and Reporting Standard:

The literature review was conducted in accordance with the principles of PRISMA 2020 and PRISMA-S [15][16]. The search included peer-reviewed articles from 1 January 2017 to September 2026. The review process involved database search, duplicate search, title and abstract screening, eligibility judgement of the full-text, backward and forward checks of citations, data extraction and evidence appraisal. The study-selection process, including identification, screening, eligibility assessment, and final inclusion, is presented in the PRISMA 2020 flow diagram in Figure 2. A total of 1164 records were found of which 327 records were removed prior to screening. The remaining 837 records were screened, 382 were excluded, and 485 reports were sought for retrieval. Of these, 90 could not be retrieved, and 395 were assessed for eligibility. Full-text assessment excluded 259 reports; leaving 136 in for the systematic review.

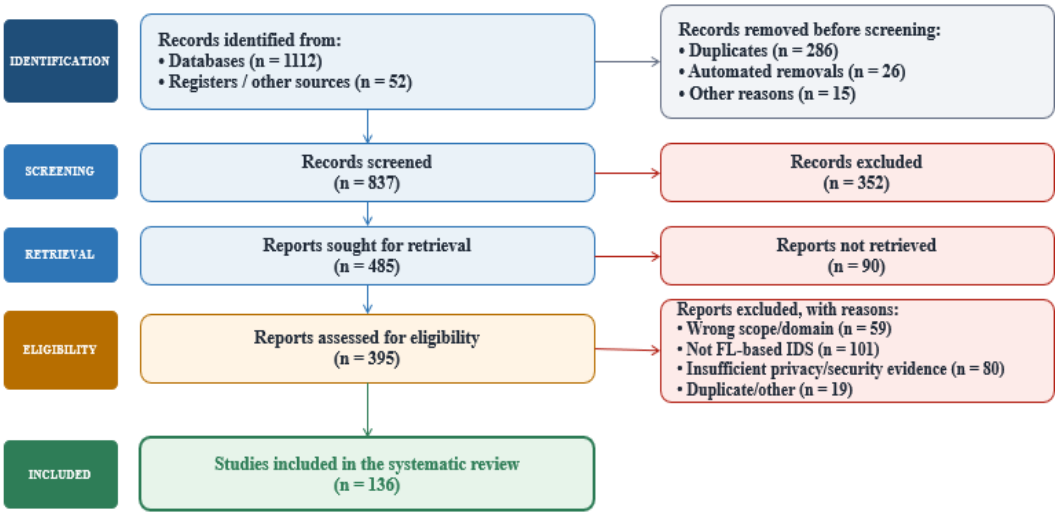

Figure 2. PRISMA 2020 flow diagram for study selection

The review protocol was not prospectively registered in a public systematic-review registry. The review was conducted using predefined research questions, eligibility criteria, search concepts, data-extraction variables, and quality-appraisal criteria, with reporting guided by PRISMA 2020 and PRISMA-S.

Information sources and search strategy:

The main search included six databases Scopus, the Web of Science Core Collection, IEEE Xplore, the ACM Digital Library, ScienceDirect, and SpringerLink. Google Scholar is used solely to search backwards through citations and forwards from them; it also helps locate available copies of articles which have been found previously due to limitations with reproducibility. The search strategy was based on four key aspects of the review question federated learning; intrusion or anomaly detection; IoT and related deployment environments; and privacy or adversarial security. Search term modifications were made according to their

respective databases' syntax such as field code, wildcard and proximity operator usage but maintaining same conceptual framework for every query.

This is an example of a Boolean expression used for searching:

("federated learning" OR "federated deep learning" OR "collaborative federated learning") AND ("intrusion detection" OR "anomaly detection" OR "attack detection" OR "botnet detection" OR "network security") AND ("Internet of Things" OR IoT OR "industrial IoT" OR IIoT OR "cyber-physical system" OR "edge device*" OR "smart device" OR vehicular) AND (privacy OR "privacy-preserving" OR confidentiality OR leakage OR inference OR "secure aggregation" OR "differential privacy" OR encryption OR poisoning OR Byzantine OR robust)

The final search group deliberately includes terms related to both confidentiality and model integrity. This reflects the review's focus on whether privacy mechanisms remain effective when the federated training process itself may be exposed to adversarial behavior. Foundational FL studies outside the IoT intrusion-detection literature were also considered through targeted backward and forward citation searching when they introduced an attack, formal guarantee, or defensive method that was necessary for evaluating claims made in PPFL-IDS studies.

The search strategy was structured around four concept groups: federated and collaborative learning; intrusion and anomaly detection; IoT and related network environments; and privacy and security mechanisms or risks. Database-specific search expressions for Scopus, Web of Science Core Collection, IEEE Xplore, ACM Digital Library, ScienceDirect, and SpringerLink are provided in Supplementary Material S1.

Eligibility and Study Selection:

We included primary studies that developed or evaluated an FL-based intrusion or anomaly detector using data from IoT, IIoT, cyber-physical, edge, smart-network, or closely related network-security environments. To support meaningful comparison, each study had to report sufficient methodological information to identify its federated-learning design, data source, and evaluation outcomes. Privacy and security focused studies are acceptable if you are evaluating an attack/defense based on network intrusion information or analyzing a technique used for implementing PPFL-IDS technology. Peer-reviewed journal articles and full conference papers were prioritized as primary sources for this review. Foundational method papers from outside the application domain were kept when they introduced an attack, guarantee, or technique required for the analysis, but these were considered as conceptual references instead of PPFL-IDS deployments.

We excluded editorials, presentations of papers; non-archival abstracts; and records that are unavailable due to lack of complete text/metadata. Publications duplicates have been merged together; all complete copies are kept as per their priority. The study of centralized IDS without any machine learning component was not included; nor did we include other types of IoT security research that are not related to ML based intrusion detection systems. The papers which used "federated" only for identity management and cloud federation were not eligible. The review articles helped identify research gaps and find other sources; however, these are not included with our main study's data set for count mechanism and evidence purposes. Preprints are considered for providing information about new research but they are not regarded as conclusive evidence if there exists an article on this topic that has already been published by other researchers. Inclusion, exclusion, and appraisal criteria used for selecting studies are described in Table 3 and included as supporting data for evaluating consistency of evidence.

Table 3. Eligibility and Appraisal Rules

Stage	Include	Exclude or downgrade	Required record

Population / domain	IoT, IIoT, CPS, edge, smart-network or close network-security settings	Unrelated domains without transferable privacy/security evidence	Domain and client interpretation
Intervention	Federated or decentralized collaborative learning for IDS/anomaly detection	Centralized IDS; non-learning federation	Topology, learning task, aggregation
Evidence type	Peer-reviewed primary study; seminal method; review for gap comparison	Editorial, abstract-only item, duplicate, inaccessible metadata	Publication type and version
Privacy / security	Explicit claim, mechanism, attack, robustness or deployment evidence	Marketing terminology without extractable design	Claim and supporting passage
Quality appraisal	Threat model, mechanism, partition, attack/baseline, statistics/resources, reproducibility	Unreported dimensions retained but judged unclear	Domain-level judgment and rationale

Data extraction and Coding:

Four groups of variables were used to extract data from each eligible primary study. System-level variables included the application domain, federation topology, client type, participation mode, learning task, model family, aggregation method, personalization strategy, and update frequency. The protected asset, the privacy unit, the adversary, the observed information, the defense mechanism and the guarantee stated were captured in the case-related variables. The characteristics of the datasets were evaluated, which encompassed non-IID partitioning, attacks, uncertainty estimates, bases, ablations, temporal separation, dataset provenance, and client construction. Operational variables such as communication volume, number of training rounds, latency, computational factor, memory consumption, energy consumption, client drop-out, scalability and deployment environment were considered.

Coding followed a conservative evidence rule: information was recorded as stated by the authors, and missing information was coded as "not reported" rather than inferred. A study that claimed “privacy-preserving FL” was not classified as being differential privacy unless a mechanism and certain parameters were explicitly provided. Likewise, encrypted transport was not counted as secure aggregation and a data split was deemed not to be non-IID unless the authors explained how to create such a data split. Security and privacy were associated to the experimental conditions used for their evaluation. Detection results from one setting were therefore not aggregated with more privacy claims from another setting. This procedure minimized the risk of bias arising from selective and cross-experimental reporting of evidence.

Distinction Between Privacy and Security Concepts:

The four concepts of privacy, confidentiality, integrity, and security are differentiated from one another in this review since they each address a different type of protection. Privacy refers to the inference or release of sensitive information about people, devices, or data records, either during training or in model updates, or through released models. Confidentiality will involve ensuring that unauthorized access is not granted to the information, while integrity will involve ensuring that the training updates, model parameters, or detection outputs aren't allowed or detected to be altered by unauthorized individuals. Security is a broader word that refers to protecting against threats to confidentiality, integrity, and availability. These goals can be interdependent, and a claim that is supported by evidence for the first goal doesn't necessarily provide evidence for the second one. For instance, encrypted communication can secure the transmission of data without omitting the ability to leak data through model

changes, and powerful aggregation can limit some poisoning attacks without obscuring the contributions of the clients. This review does not assume privacy protection based exclusively on a broadly stated security label or mechanism, but rather validates each claim of privacy and/or security by testing it under the specified threat model and mechanism, along with accompanying evidence.

Quality and Confidence Appraisal:

The methodological aspects were evaluated on six levels, clarity of the threat model, specification of the privacy, robustness mechanism, validity of the federated data partitioning, adequacy of attack types and comparison baselines; completeness of statistics/reports/resources availability. Every site got an "adequately reported", "partially reported" and "not available / unclear". Research studies are also not excluded due to poor reports. However, this evaluation was utilized for determining what level of credibility should have been given by these results; also, how strongly they were represented through synthesis.

Appraisals were customized according to what was provided for evaluation purposes. To provide formal assurances about this research project, we asked if it was clear how 'adjacent' and 'unit-of-protection' were defined; what they assumed by doing so; what kind of information on privacy data was provided during this process; etc. The empirical privacy evaluation should have described the attacker's ability to access data prior to an event; their capability of optimizing attacks for given events; and comparison between unprotected baseline and protected defense settings. Also, they had been asked about impacts on confidentiality and sensitivity of information, and accuracy for detecting attacks / intrusions etc.

The robustness test can be evaluated by considering number and type of attacks, malicious, benign, targetability of attack targets along with their capability for attacking system performance while given clean inputs vs attacked input set. The claims for deployability have been evaluated much more rigorously by means of testing them through real hardware/software/networks during an assigned timeframe; communications, computing power and electricity consumption are also tracked directly instead of being projected.

Studies in each of the 6 appraisal domains were given a domain-level judgment based on a three-category coding scheme: "2 = adequately reported, 1 = partially reported or not available, and 0 = not available/unclear". It was judged to be an adequately reported (2) study when the information contained in the study was sufficient so that the connection between the characteristic and the reported claim or result can be determined independently. When characteristic described, but significant methodological details, assumptions, parameters or evaluation conditions missing, a partially reported (1) judgment has been made. A judgment of not available/unclear (0) was given when the information was unavailable, ambiguous, or could not be found within the study.

The six domains were assessed individually; threat-model clarity, privacy specification, robustness mechanism, validity of federated data partitioning, attack type, and comparison baselines, and statistical, reporting, and resource information. Studies were not excluded because of low appraisal ratings. Rather, domain-level judgments were employed for determining evidential confidence of a synthesis. The six domain scores were not aggregated into an overall score due to the fact that weaknesses in different dimensions of the evidence are not necessarily substitutable.

Synthesis Method and Limitations:

Studies were so heterogeneous that they could not be combined for an effective summary statistic of effects. Different data sets were utilized; clients partitioned models are used for attacks and evaluations were made on them respectively along with privacy parameters set during this process. Some did not include a confidence interval and another measure for variability. There were no analyses conducted here either. However, results are

summarized by means of narrative synthesis and evidence matrix analysis. Results are quantitative if they can be directly compared between two sets of data with identical parameters (partitioning), threats and measurement types. The detection score for an experiment that could not be compared with another's was reported as 'not available' instead of being estimated/imputed.

There are some constraints on this study's coverage. Studies might not have been reported due to lack of access and change on terms used by researchers; also there's a tendency for positive outcomes being published more readily compared against failures/negatives. Starting the search in 2017 excludes previous research on distributed intrusion detection which didn't use current FL terms. Restricting the primary data source to English-language and peer-reviewed articles might exclude industrial applications and reports from local media sources. Thus, this paper focuses on verifiability and reliability of each claim not comprehensiveness alone. Figure 3 describes the systematic review process, including literature selection/screening steps that relate to claims audits and synthesis of evidence.

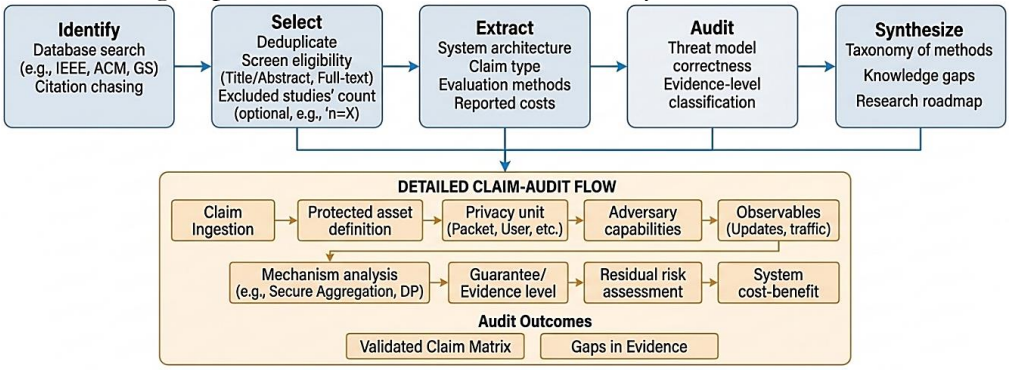

Figure 3. Systematic-review workflow and claim-audit logic

To improve traceability between the research questions and the review findings, Table 4 maps each research question to the corresponding extracted variables, synthesis methods, and principal findings. This mapping clarifies how the evidence collected during the review was used to address each research question and provides a transparent link between the review methodology and the subsequent synthesis.

Table 4. Mapping of Research Questions to Extracted Variables, Synthesis Methods, and Findings

Research question	Extracted variables	Synthesis method	Main finding
RQ1	Publication year, application domain, federation topology, client type, participation mode, learning task, model family, aggregation, personalization, update frequency, dataset	Descriptive and narrative synthesis; evidence-level classification	PPFL-IDS research has evolved from feasibility-focused FL-IDS toward heterogeneous, adversarial, privacy-aware, and resource-conscious systems, but evidence maturity remains uneven.
RQ2	Protected asset, privacy unit, adversary, observed information, defense mechanism, stated guarantee, privacy parameters, leakage attacks, residual risks	Claim–mechanism–evidence audit and taxonomy	Privacy claims vary substantially in scope; data locality is frequently treated as privacy although stronger confidentiality or inference-resistance claims require additional evidence.

RQ3	Non-IID partitioning, personalization, client participation, poisoning, Byzantine behavior, client dropout, continual learning, detection performance	Cross-theme narrative synthesis and evidence matrix	Privacy, robustness, heterogeneity, participation, and detection performance are interdependent and may involve important trade-offs.
RQ4	Attack coverage, baselines, statistics, uncertainty, reproducibility, communication, latency, computation, memory, energy, scalability, deployment environment	Quality appraisal, evidence-level assessment, and gap analysis	Major evidence gaps concern unrealistic client construction, limited adversarial testing, incomplete privacy guarantees, insufficient resource reporting, and limited deployment validation.

Evidence Landscape and Integrated Taxonomy:

Development of the Field:

The earliest FL-based intrusion-detection studies focused primarily on feasibility. They demonstrated that devices or sites could learn a shared anomaly-detection model from local network observations without transferring the underlying traffic records. DÏoT, for example, introduced federated self-learning for device-type-specific IoT behaviour. Subsequent studies explored recurrent models, segmented federations, hierarchical coordination, and industrial applications [17][18][19][20]. Together, this early work showed that collaborative training could broaden detection coverage across distributed traffic sources. Privacy, however, was generally understood as keeping raw data local rather than protecting the information exchanged during federated training.

As the field matured, researchers began paying greater attention to heterogeneous client data and malicious participation. Proposed approaches included multi-view ensembles, client clustering, personalization, knowledge transfer, proximal optimization, incremental learning, and selective client participation [21][22][23]. At the same time, poisoning research showed that the update channel enabling collaboration could also become a pathway for compromising the detector. Studies examining attacks against FL-based IoT intrusion detection, along with systems such as SecFedNIDS and dependable aggregation frameworks, increasingly treated participant integrity as a property that needed to be evaluated rather than assumed [14][24][25].

More recent research has focused more directly on privacy. [19] examined privacy-preserving collaboration in industrial IoT environments. FedDef investigated gradient reconstruction in network intrusion detection, while differential-privacy-based approaches reported the resulting trade-offs between privacy and detection performance [26][27][28]. Other recent frameworks have combined encrypted aggregation with poisoning detection, communication reduction, or personalized learning [29][30]. In several useful ways, the literature extends this direction up to the 2026 literature. FedPrIDS is a collaborative network intrusion detection system with adaptive differential privacy, and it is tested based on the CIC-IDS-2017 dataset, UNSW-NB15 and Bot-IoT [31]. In a hierarchical IIoT architecture, HFL-DP NIDS handles both client-to-edge and edge-to-cloud interfaces by incorporating differential privacy, it reports privacy accounting at every round, and provides membership-inference evaluation in addition to formal privacy budget [32]. FSL-IDS reports efficiency-oriented measurements, and it strives to solve the resource binding problem in IoT deployment by implementing hierarchical federated learning and sparse representation learning as well as quantizing models [33]. The combined system of DP-FAL involves hierarchical federated learning, differential privacy, and adversarial training for V2X intrusion prevention system, and reports detection, privacy-utility and communication metrics [34].

These studies serve to reinforce the evidence, but they also serve to show that privacy mechanisms, adversarial conditions, threat assumptions, data partitioning and the costs of the systems need to be audited together when relying on federated training only. FedFusion-IDS, an integrated federated CNN-BiLSTM-Attention learning approach in combination with multi-objective feature selection and explainability on the TON_IoT dataset in 10 clients for 20 communication rounds [35]. The evaluation provides additional evidence of detection, evidence of zero day and evidence of interpretability, but the privacy claim is of an architecture-based nature, rather than backed by a formal privacy guarantee or a specific leakage-attack evaluation.

Review articles have followed a similar progression. Earlier surveys categorized FL-based intrusion detection by architecture, learning model, aggregation method, or application area, while later reviews introduced broader privacy and security perspectives [8][10][11]. More recent systematic reviews have expanded the literature coverage and developed more detailed taxonomies [6][7][9]. However, an important gap remains: the traceability of individual protection claims. In particular, it is often unclear whether a claimed protection is precisely defined, correctly implemented, tested against an appropriate adversary, and evaluated in terms of its operational cost.

An Evidence-Centered Taxonomy:

PPFL-IDS are organized according to five parallel dimensions by taxonomy. Federations are defined by their nature of being cross device/silo/hierarchical/decentralised client type, endpoint, gateway, organization, vehicle, synchronous/training mode and frequency. Privacy claim defines what is at stake; which privacy system applies to it; who might be attacking them; what they have knowledge about them so far; when this will happen again.

A dimension of mechanisms categorizes strategies based upon their reliance upon data minimization, perturbation; secure computation; trustworthy execution; compression/ representation sharing; decentralization and mixes thereof. Learning dynamics deals with aspects influencing training behavior such as diversity of data types; individualized approaches; ongoing education processes; clients' choice and delivery changes over time. Finally, the evidence dimension states whether a study gives you an official assurance, experimental testing on attacks; comparative effectiveness between detection tools; costs associated with implementing those methods sufficiently so that they can be reproduced by others.

Together, these dimensions provide a way to distinguish not only what a PPFL-IDS does, but also what it claims to protect, how that protection is achieved, how the learning environment affects it, and how convincingly the claim has been supported. These five dimensions are integrated into the evidence-centered taxonomy shown in Table 5, which provides a common structure for examining PPFL-IDS architecture, privacy claims, mechanisms, learning dynamics, and supporting evidence.

Table 5. Integrated Evidence-Centered Taxonomy

Dimension	Core categories	Audit question	Frequent confounder
System / deployment	Cross-device, cross-silo, hierarchical, decentralized; synchronous/asynchronous	Who computes, sees, aggregates, and acts?	A simulated client is treated as a real device/site
Claim/threat	Asset, privacy unit, adversary, observables, failure event	What is protected from whom?	Locality is treated as confidentiality
Mechanism	Minimization, perturbation, secure computation, TEE, transformation, governance	Which observation or influence is changed?	Mechanism name substitutes for parameters

Learning dynamics	Non-IID data, personalization, selection, drift, continual learning	How does change alter utility and leakage?	Benign outliers are confused with attackers
Evidence	Formal guarantee, empirical attack, utility, resource cost, reproducibility	What has actually been demonstrated?	Best benchmark score becomes system assurance

Separating these dimensions helps avoid treating similar labels as if they represented the same level of protection. Systems described as using differential privacy, for example, may protect different privacy units, use different clipping thresholds and privacy-accounting methods, or expose different forms of operational metadata. Similarly, two secure-aggregation protocols may rely on broadly similar cryptographic techniques while making different assumptions about client collusion, participation, or dropout.

The same issue arises when comparing studies that use the same dataset. Results may not be directly comparable if one study creates clients by randomly assigning rows, while another constructs client from traffic generated by distinct physical devices. These choices produce different levels of data heterogeneity and can influence both privacy exposure and detection performance. The taxonomy therefore places each study in context by considering its design, assumptions, and supporting evidence. It is intended to clarify differences among approaches, not to rank mechanisms on a single scale or imply that any one approach is universally stronger than the others.

Claim–Mechanism–Evidence Patterns:

There are four types of claims mechanism evidence pattern found among these studies. First of all, there are some papers where they say "privacy" means "data location": raw data stays on clients' computers and not much security around update files, intermediary state information, published model details or usage history. Localization of data helps reduce direct data transfers and also provides some level of protection against inferential attacks on private information.

Also, there are more ways of keeping data private such as differential privacy, secure aggregation, encryption, blockchain-based coordination, perturbation and trust executions now used for PPFL-IDS design purposes. Nevertheless, its effectiveness as an argument Parallel Formal Information Security Model relies upon parameters chosen by us for our system design; also against attackers' knowledge about this model and available data from them. Empirical evaluation usually studies just one or few instances of reconstruction attack, membership inference and poisoning attack. These findings are useful for testing an attack model evaluation; however, we shouldn't generalize them on more powerful, adaptable or collaborative threats without sufficient proof of their existence. Composite systems which are composed of multiple layers can offer more comprehensive security but there might be some compromises with respect to usability, visibility, delay, and reliability due to interaction between these components.

Thirdly, there are more studies on detection accuracy compared with other things like privacy protection strength against adversarial examples of non-IID datasets and resistance towards specific attack vectors. While many studies showor, as shown by research papers indicate effective learning of FL based on partitions for intrusion detection systems (FL), there are few results showing client level confidentiality protection against attacks such as adaptation or collusion among attackers; also no study has demonstrated reliability with respect to privacy accounting over several iterations during retraining cycles. Similarly, security measures for detecting malicious update detection are usually separate from integrity protection methods.

Therefore, good prediction ability does not mean secure data and safe systems. The detection tool, privacy protection, security of data, accessibility to users' devices and network

operations costs must be considered independently when describing an effective PPFL-ID system.

Datasets as Part of the Threat Model:

Dataset selection directly shapes the questions that an intrusion-detection experiment can meaningfully answer. N-BaIoT, for example, contains traffic from nine IoT devices infected with Mirai and BASHLITE, making device-based partitioning possible [36]. Bot-IoT, ToN_IoT, Edge-IIoTset, and CICIoT2023 broaden the range of devices, protocols, and attack types represented in the literature. Edge-IIoTset was specifically designed to support both centralized and federated experiments, while CICIoT2023 provides traffic collected from a large real-time IoT testbed [37][38][39][40]. Despite their usefulness, these datasets are still snapshots of particular testbeds, devices, attack scripts, and feature-extraction pipelines. Reusing them cannot fully reproduce the governance challenges of cross-organizational collaboration, adaptation by live adversaries, diversity in firmware and configurations, or the sensitive identities present in operational networks.

Leakage of information from data set of training set, to test set will affect results for centralized and federated approaches equally well. Random splitting could group data of identical devices and sessions together, and, or attacks, so that we can train our model on those specific sources instead of just all attacks together. By creating a client through random assignment of data points there is an issue that locality-aware datasets might still be too close and thus not really test diversity well enough with respect to FedAvg. Balancing artificially may hide a false positives load which is likely to occur if attacks are common enough.

Addresses, identifiers and time stamps are also a problem here. Privacy attacks may exploit fields that would normally be removed during deployment, while aggressive preprocessing may remove the very sensitive structure that a proposed privacy mechanism is intended to protect. These choices can therefore affect both the apparent privacy risk and the validity of the evaluation.

For this reason, the taxonomy records dataset provenance, the unit used to construct clients, temporal ordering, controls for data overlap, label distribution, the location and nature of preprocessing, and whether clients represent actual devices or sites. Arbitrary groups of randomly selected rows do not provide the stable client identity needed to support a client-level privacy claim. Likewise, withholding random samples from attack classes that were already represented during training does not demonstrate zero-day detection. In each case, the experimental unit and evaluation split should correspond to the scope of the claim being made.

Synthesis Rather Than Algorithm Ranking:

The available evidence does not support a universal ranking of PPFL-IDS mechanisms. Secure aggregation is well suited to settings where the coordinator should not be able to inspect individual client updates. However, hiding those updates can also reduce the information available for detecting anomalous contributions. Central differential privacy generally preserves more detection utility than local differential privacy, but it requires trust in the aggregator or in a secure computation mechanism that can add noise correctly. Local differential privacy lowers that need for trust, but more noise could have a big impact on performance especially with rare attacks.

The homomorphic encryption enables direct computation of data without decryption; however, there are high costs for computation time, keys' security issues that need not be addressed by low-power IoT devices with limited resources available today. Personalization might help be better at detecting minority clients but there are also risks associated with using localized models and common representations that could lead to data leakage issues. Aggregation is robust to some types of attack on poisonings however it might refuse an odd but valid update coming from a client that has no IID data at all.

The trade-offering is to evaluate these options according to how they are meant for use instead of ranking them together as an overall measure. Design should be credible if it's appropriate to choose a security measure that applies directly onto what you're trying to protect; realistic assumptions about how things work within this particular context; also consider whether there are additional threats coming from other parts of your infrastructure. These are applied here according to what is known about privacy and trustworthiness.

What the Taxonomy Reveals About Review Novelty:

Reviewing studies that focus on specific mechanisms can show what methods have been investigated by scientists; however, this does not necessarily indicate how well those conclusions were supported by evidence. Three common ways to exaggerate strength of evidence are highlighted by this taxonomy. Terminological inflation happens if "local" data are considered enough for privacy protection; also when "noise" (ordinary) is called differential privacy without an explicit assurance that it's true. Experimental inflation happens if there's some form of defense that's credited for protecting something despite no evidence being provided to support this claim; it's backed up by formal analysis and comparison to an appropriate baseline. Validity of external validation refers to presenting findings from an easy-to-access subset of data as proof that this technique can be used on other devices/websites.

The categories don't imply weakness of study results either. Some of these prototypes were developed for more specific technical problems; they might have been valid within those particular limits. It is a problem to extend these findings further than they were established by earlier research studies themselves.

The separation of taxonomic features is also revealing interrelationships which are often overlooked by algorithms-based classification systems. For instance, data heterogeneity impacts optimization; it could provide insight on customers' needs; malicious update is harder to detect if there are any similarities between them with good ones. An odd discovery could be an unusual gadget; it may also include helpful information about a new threat; it might even come out of intentional poisoning. A distance-based aggregation function might not differentiate between those options without more information. The communication reduction strategies are not just about saving on resources but also changing what information is accessible for attackers. Selection of clients impacts on convergence rate, accessibility and behavioral patterns are observed during this process. As such because this choice affects many parts within an architecture so it should be analyzed from various angles instead of being confined by one particular class.

Evidence is another distinction that separates non-evaluations from bad results. "Not reported" means there is no report/testing while "tested and failed" shows that something has been tested but not worked out successfully enough to prove its limitations. Negative results are especially useful if you find out how well your defense fails on small clients; high levels of collusion among customers and labels; low amounts of available power for defending against them. These results will assist to determine when an IDSS is reliable for use as a PPFL-IDS. These differences are reflected in the comparative tables presented under Privacy Mechanisms and Evidence Audit, Integrity and Availability Threats, and Experimental Evidence and Deployment Realism, as well as in Research Gaps and Prioritized Roadmap.

Privacy Mechanisms and Evidence Audit:

Data Minimization and Architectural Locality:

One of the biggest privacy benefits for FL are its ability to minimize unnecessary data transfers. If a company does not require full packet trace information nor operation logs then local processing and training is better for reducing access to raw data, simplifying its management process significantly. DÏoT and subsequent FL-based intrusion-detection systems show how devices and gateways can contribute to a shared model while keeping their

traffic locally [18]. Nevertheless, however please define what is protected here by "federated" protocols do not centralize any particular kind of data directly. No, it doesn't safeguard data from being stored within customer update messages; models released for use and user engagement statistics.

Information transfer could also be minimized through restricting feature set size; sharing summary data for a given region; exchanging logits/prototypes between clients; decreasing frequency of updates and maintaining individualization at client side. However, these decisions may impact on detection effectiveness. Coarse statistics might conceal packet-level details by removing those of small attack volumes. Shared representations might reveal characteristics for some types of attacks; however, reduced interaction would decrease this knowledge and slow learning about new threats. Therefore, evaluations should concentrate more upon what amount of data could have been provided without compromising on relevant operationally critical memory instead of just comparing between federated and centralized learning models.

Leakage From Gradients, Updates, and Models:

Studies of gradient inversions show that communication between models while they are being trained could be informative for recovering some features from their original datasets. Deep Leakage from Gradients showed that shared gradients could be used to reconstruct training examples, while later approaches improved inversion over various models and previous assumptions [2][5]. Studies related to collaborative learning have found out that model's update may show some hidden characteristics about local datasets. White-box membership inference attacks exploit intermediate models' states and active interaction with the training process [3][4]. While there are many more differences between network traffic data and image data, there are still some privacy concerns, such as that if you have enough information, clients' updates might contain numbers/categories/classes, and device behaviors.

Effectiveness is highly dependent upon experiments conducted here. Reconstruction tends to be simpler if you have access to an instance/ batch of samples for generating gradient information; knowing about parameters like initialization and updates that aren't aggregated together during this process. Training steps may be multiple; batch size is big; compression, secure aggregation (sag) and heterogeneity of data could complicate inversions significantly even if they are successful with some setups alone. Therefore, a good privacy audit would state what models are able to see by attackers, and, or their help. Also, it must include things like batch sizes, number of local moves; features are scaled; categorical variable is handled by means (or whatever); cost for optimization is considered here along with criteria on how attacks succeed.

Metrics of images to measure similarities may not work well with tabular networks if you have no explanation about what's important there. Reconstruction of a sequence might reveal confidential data despite being far away from an existing data point on a Euclidean scale. Similarly, partially recovered features might also be sufficient to allow an attacker to create a traffic pattern that is not detectable by this system.

FedDef is especially important as it focuses on gradient leakage within an intrusion detection system framework where reconstruction relates more closely with attack traffic rather than just being visually/quantitatively similar [27]. Broadly speaking, privacy assessment needs to be concerned with possible harms that may arise during implementation of this project. Meaningful outcomes could be better membership inference; recovery of sensitive attribute/valid protocol field, device re-identification, or construction of traffic that avoids detection.

Differential Privacy:

Differential privacy (DP) gives an upper limit on how much the distribution of a released output can change when one protected unit is added to or removed from a dataset

[13]. But for FL, we need to clarify what "protected" means at all times. A record level DP restricts how much one data point can affect another when it comes to being protected either by localizing them and aggregating them together as part of some larger system. Client level DP is concerned with answering this more important issue - can you detect if all clients are participating together. Usually, it involves taking samples from all clients' updates then adding some amount of 'noise' that is based upon how much we are sampling them individually together with their privacy costs cumulatively over time. Reporting just an amplification factor for noises and "random" perturbations as DP would not suffice for interpreting this claim about security.

The reproducibility report for a DP test must include information on adjacencies; clipping norms and locations where they are used; client sampling models; noise distributions and multipliers; number of training iterations, privacy accountants and resulting epsilon-delta guarantees. Also, it should mention if this is a local/central guarantee and who will be able to monitor unclipped/unnoised information. Aggregation security may allow for centralized/ decentralized noise injection and privacy of users' data; however, this is dependent upon trustworthiness of noise generation, client abandonment rate and collusion. Privacy accounting should also account for privacy loss during a training cycle, and, or release, and while doing hyperparameters search/searching for another detector if they use identical datasets again and then on.

Data privacy has a significant effect on intrusion detection system. Unusual attack types and small customer bases might result in reports which are quite different to what most people are seeing today. Clipping might block some of that good information and adding more noise could lead to more missed attack/alarm rates. Studies that vary privacy budgets and client configurations help make these trade-offs visible, but their conclusions are specific to the stated privacy unit, privacy accountant, sampling procedure, and dataset [19][26][28]. Accuracy by itself also isn't sufficient. The utilities must also include information about attacks (attack classes), clients, timestamps; false positives rate and predictive accuracy of predictions. Evaluations should also compare multiple privacy budgets and include a clipped but non-private baseline so that the effects of clipping and noise can be distinguished. Recent HFL-DP evidence likewise shows that clipping, noise, and participation assumptions materially affect the privacy–utility trade-off and should be reported explicitly [32].

A correctly specified DP guarantee provides stronger assurance than simply showing that a particular reconstruction attack failed. Even so, DP does not protect model integrity, hide every form of metadata, or establish regulatory compliance. When the privacy accountant or adjacency relation is not adequately defined, the approach should be described as a perturbation technique rather than presented as providing formal differential privacy.

Secure Aggregation and Encrypted Computation:

Secure aggregation allows a coordinator to compute an aggregate of client updates without seeing each contribution individually, provided that the protocol's participation threshold, dropout tolerance, and cryptographic assumptions are satisfied [41]. It is primarily designed to protect against an honest-but-curious coordinator and may also reduce the information exposed if the server is compromised. However, this protection has clear limits. An aggregate can still reveal sensitive information when only a small number of clients participate, when the server adaptively selects participant groups, or when colluding parties already know most of the individual contributions. Information may also leak through the released global model, while repeated aggregates can allow an observer to isolate changes over time. Secure aggregation therefore limits what the coordinator can directly observe, but it does not provide a numerical bound on privacy loss.

The encryption based PPFL-IDS also have to describe what is done for privacy-preserving computations. Key information includes what types of operation occur to

ciphertexts; who holds key management rights; whether keys get rotated periodically; how do you handle clients dropping off their services altogether; and which models and precisions are supported by this system. Also, it must indicate if there are still unencrypted update versions that can be used to detect poison attacks. Homomorphic encryption supports aggregation/limited inference by not decrypting data; while multi-party computation distributes trust among multiple participants. However, it is feasible to them depending upon workloads. Ciphertext expansion, modular arithmetic, extra communication rounds, and key management may be too much for low-power clients despite their performance on an emulator of a desktop computer system. Therefore, runtime evaluation must differentiate between computations done on end points/ gateways versus those related to data aggregation, transmission and decryption expenses.

Observability is an issue at its core. An inspector is not allowed to look at each update independently to see if there's any sign that someone has been poisoned; they also can't see these updates themselves. Some possible compromises are secure computation of strong statistics; client-side attestation / trusted execution; verification of computation; small trusted validation set; privacy preserving reputation mechanism. Every choice adds more credibility, equipment and protocols.

Secure aggregation and an effective decryption mechanism should be described by this research paper as to what happens during encryption/decryption process along with who gets access upon receiving those encrypted data points. Without these details, they might seem like complementary systems but cannot have both safeguards at once.

Trusted Execution, Blockchain, and Decentralization:

A trusted execution environment (TEE) could be used to separate aggregations from stronger security measures that are not just based upon encryption alone. Nevertheless, this is due to assumptions regarding physical components; remote attestation; side-channel security; patches; control over a system. Blockchains and distributed ledger systems are capable of providing immutable update commitment, coordination, incentive mechanisms and traceability; however, these technologies do not ensure that models are kept secret nor guarantee accuracy. Hashing publishes might improve transparency while posting changes increases vulnerability to surveillance. Latency to consensus along with persistent storage of information may prove troublesome while doing iot collaborations at scale.

Peer-to-peer decentralized FL reduces dependence upon one person as a leader; however, this increases connectivity of nodes within their network which could potentially be exposed through notifications from other members who are nearby. Also, it alters possible toxins' routes of action; trust-building needs; and accountability processes. Thus, these strategies should be more accurately described by choice for government management and security issues instead of just privacy concerns due to their use with some particular technology.

It's still about who can't see what else they're not allowed to know; how much of that would be collusive between parties, and, or, along with costs involved.

Compression, Sparsification, and Representation Sharing:

Quantization, sign-based update, sparsification and knowledge distillation is mainly used for reducing communication cost. Lowering accuracy/quantity for data transfer while learning could also increase difficulty to certain types of attacks through reconstruction methods. Nevertheless, compressed update messages may also contain information about memberships and privacy features. Communication reduction should not be treated as evidence of privacy protection. SAFE-IDS, for instance, combines sign-based aggregation with an optimizer that addresses non-IID learning while also limiting privacy-related exposure [42]. Avoiding transmission of precise gradient data alone does not suffice for privacy claims either. The evaluation must determine if a hacker is able to deduce private data based on what

they receive through communication channels, signs, forms, and whether this process results in additional types of leaks.

Knowledge distillation alters how information gets passed around; it's about transferring knowledge rather than just passing through models' parameter-wise like weights/logits, etc. It will help to collaborate between users with various models; however, output from this exchange might contain details regarding their training set/example/inputs utilized for generating these results. Public reference datasets might not be sufficient for representing an organization's IoT behavior; synthetic references may raise further queries regarding their generation process and if they contain any details of other organizations' data. Therefore, compression and sharing strategies are about altering what is accessible for attackers not just privacy tools by default. Privacy preserving, they must also not have any other definitions for being secure unless specified by attack types that are considered relevant here.

Operational Criteria for Evidence Levels:

The E0-E3 framework classifies the strength of evidence supporting a specific privacy or security claim in federated learning for IoT intrusion detection. It is classified, not by the terms used in the proposed mechanism, but by the results of the study itself. It is especially important to note that having local copies of raw data, using a transformation that is privacy-sensitive, or using a denoting a cryptographic mechanism does not set a formal privacy guarantee without the other components of the package. Implementation details, threat model, evaluation and assurance actually reported should justify the evidence level.

E0 - Architectural or assertion-level evidence. This level applies when a study introduces federated learning, decentralized storage of data, or a mechanism described as a basis for privacy or security, but does not give details on how it was implemented and/or claim specific validation of the privacy or security aspect. Level is used to show that protections are asserted or motivated, not that the attack has been ruled out.

E1 - Specified or implemented mechanism. This level is appropriate for a protection mechanism whose function has been sufficiently described, implemented, or configured in detail to evaluate their usefulness against a specified adversary and where a privacy or security claim is not sufficiently tested. Implementing a mechanism does not make resistance to attacks absolute or constitute a formal guarantee.

E2 - Empirical adversarial evidence. This level results in evaluations of relevant privacy or security claims for a clearly outlined attack or adversary, and the results and conclusions are interpretable. The attack setting, observed information, the procedure for evaluating, and the limitations that apply to the procedure should be adequately defined. Routine intrusion detection accuracy or a simple comparison of the utility of the model without empirical evidence does not constitute evidence for a privacy claim.

E3 - Formal or compositional assurance. This level refers to studies that represent a formal guarantee, protocol-security, or compositional assurance guaranteeing a clearly bounded claim, within the context of specified time and other parameters and assumptions. The assurance ought to feature the appropriate safeguarded asset or privacy system, threats presume, and constraints. If no corresponding analytical or formal support is provided, merely adding differential privacy, encryption, or secure aggregation to the components is not enough to classify as E3.

These grades indicate the amount of evidence supporting such a claim and do not necessarily indicate the general quality of a particular study. It is possible that just one study would contribute E2 evidence for intrusion detection performance while contributing E0 or E1 evidence for privacy. When there is evidence for more than one level, that classification should reflect the claim that is being evaluated and that range should be retained only if the

evidence truly covers the range of the classification. Any missing information is interpreted as "not reported," and not assumed evidence of protection or vulnerability.

The unit of analysis needs to be explicitly defined when designing a quantitative report. If studies are included, each study should be assigned to one study level in accordance with a pre-defined criterion, such as the highest study level supported by the relevant privacy claim. Alternatively, if you count claims, then you can have a study with multiple observations. These two methods should not be compared at the same time and their total should not be offered as an alternative to each other as evidence maturity measures. To ensure consistent interpretation of privacy and security claims across the included studies, this review classifies the supporting evidence into four levels (E0–E3), as summarized in Table 6.

Table 6. Operational Criteria for Classifying Evidence Supporting Privacy and Security Claims

Level	Operational criterion	Evidence required	What does not qualify by itself
E0	Architectural or assertion-level claim	A stated rationale or architectural description	Keeping raw data local or calling a method privacy-preserving
E1	Specified or implemented mechanism	Sufficient implementation, configuration, or protocol detail	Naming a mechanism without relevant validation
E2	Empirical adversarial evaluation	A specified attack/adversary, evaluation procedure, and reported outcomes	Ordinary IDS accuracy or utility results alone
E3	Formal or compositional assurance	A formal guarantee or security analysis with explicit assumptions and parameters	Merely using differential privacy, encryption, or secure aggregation

Mechanism Evidence Matrix:

Table 7 applies the claim-audit framework to representative PPFL-IDS mechanisms, showing the evidence supplied by each study alongside the principal residual risk and the corresponding evidence level.

Table 7. Representative Mechanism–Claim–Evidence Audit

Study / author(s) / year	Mechanism, target, and data/setting	Evidence actually supplied	Main residual issue	Level
DÏoT — [18]	Federated device-type anomaly profiles and raw-data minimization; evaluated on more than 30 off-the-shelf IoT devices with a real-world smart-home deployment.	Distributed IDS feasibility and operational prototype evidence	No general guarantee against update/model inference	E0–E1
[19]	Differential privacy for IIoT federated intrusion detection; non-IID ToN_IoT data with FedAvg and Fed+ aggregation.	Mechanism-enabled FL-IDS experiments	Claim depends on implementation and attacker assumptions	E1–E2
2DF-IDS — [28]	Decentralized FL with R-LWE key exchange and differential privacy; evaluated on a real-world IoT/IIoT intrusion dataset.	Privacy mechanism plus IDS utility evaluation	Budget, unit, composition and systems cost govern meaning	E1–E3
FedDef — [27]	Input perturbation against gradient leakage; evaluated across KDD99, Mirai, CICIDS2017, and UNSW-NB15.	Task-specific reconstruction/evasion evaluation and theoretical analysis	Coverage remains attack- and configuration-dependent	E2–E3
Secure aggregation — [41]	Cryptographic hiding of individual client inputs from the coordinator; protocol/system evaluation rather than an IDS-specific dataset.	Protocol security and practical systems analysis	Aggregate/model leakage, threshold, collusion, dropout	E3
SecFedNIDS — [14]	Poisoned-model and poisoned-data detection for FL-based NIDS; evaluated on UNSW-NB15 and CICIDS2018.	Label-flipping and clean-label poisoning experiments with model-level and data-level defenses under heterogeneous traffic.	Coverage depends on the evaluated poisoning models, detector thresholds, and transfer to adaptive or colluding attackers.	E2
SAFE-IDS — [42]	Sign-based sharing/aggregation under non-IID data; CIC-IDS2017, CIC-IDS2018, CIC-DDoS2019, KDDCup99, and NSL-KDD.	IDS utility and privacy-motivated transformed updates	Transformed observables still require matched attacks	E1–E2
ROCHE — [30]	Homomorphic encryption, adaptive quantization, and dropout tolerance;	Integrated mechanism and IDS evaluation	Composition, trust, and edge cost require scenario-specific audit	E1–E3

	evaluated with ToN_IoT, Edge-IIoTset, and X-IIoTID.			
SecureDyn-FL — [29]	Temporal gradient auditing, encrypted aggregation, and personalization; N-BaIoT under IID/non-IID settings, including up to 50% adversarial clients.	Integrated poisoning, privacy, non-IID, and systems evaluation	Assurance depends on audit placement, trust, key management, and transfer beyond one dataset	E1–E3
FedPrIDS — [31]	Adaptive differential privacy for collaborative NIDS; CIC-IDS-2017, UNSW-NB15, and Bot-IoT across five synthesized organizational settings.	Formal epsilon-delta privacy design plus multi-dataset IDS utility and communication evaluation	Guarantee depends on the stated privacy unit, accountant, threat model, and repeated-release composition	E2-E3
HFL-DP NIDS — [32]	Hierarchical FL with clipping and Gaussian noise at client-edge and edge-cloud interfaces; CIC-IDS2017 and Edge-IIoTset under IID, non-IID, and class-imbalanced settings.	Round-level privacy accounting, privacy-utility analysis, and membership-inference evaluation	Simulated hierarchy; deployment cost and broader attack coverage remain scenario-dependent	E2-E3
FSL-IDS — [33]	Hierarchical sparse learning and quantized deployment for constrained IoT; IoTID20 with resource evaluation including Raspberry Pi/ESP32-class devices.	Detection, ablation, sensitivity, and resource-efficiency evaluation	Privacy rests mainly on federated locality and robustness controls rather than a formal confidentiality guarantee	E1-E2
DP-FAL — [34]	Differential privacy plus adversarial learning for V2X federated intrusion prevention; CICIDS2017 and VeReMi.	Detection, adversarial robustness, privacy-utility, and communication evaluation	Privacy and robustness depend on attack coverage, DP settings, and transfer beyond the evaluated V2X conditions	E2-E3
FedFusion-IDS — [35]	Federated CNN-BiLSTM-Attention with NSGA-II/SHAP feature selection and explainability; TON_IoT, 10 clients, 20 communication rounds.	Detection, zero-day, and explainability evaluation; reported 66.26% global accuracy, F1 = 0.6524, with ROC-AUC above 0.99 for most primary attacks.	Privacy is mainly inferred from federated data locality; no formal privacy budget or dedicated leakage-attack evaluation is reported.	E1–E2

Note. E0 = architectural assertion; E1 = specified mechanism; E2 = empirical adversarial evidence; E3 = formal or compositional evidence.

Three major findings are made by this study. First, strong assertions tend to be very detailed. Secure aggregation may conceal personal contribution to an identified leader; differential privacy may limit impact on a known privacy entity; and an empirical defense may mitigate an identified attack given those circumstances. Such statements have value since they define what is being said clearly enough to warrant further discussion.

Secondly, various mechanisms safeguard distinct aspects of a system; therefore, they may work together efficiently if there is compatibility between these bases. Confidentiality mechanisms like this could be interfering with information needed to defend against integrity attacks.

Third, there is also a residual risk and cost to systems that are covered by this insurance policy. An algorithm which can't finish during this cycle of learning; consumes too much power and results to an unacceptable degree on rare attack types isn't really useful for security purposes regardless how good it might be mathematically speaking.

A Threat-Driven Mechanism Selection Logic:

The mechanism choice ought to start from an intelligent person who is likely to be a good observer instead of one that's currently popular with tech enthusiasts. In case of an outside spying party then you need to use secure communications with encryption on top of this; however, this will not protect endpoints nor servers from being compromised by hackers. If the coordinator is honest but curious and individual client updates are sensitive, secure aggregation can directly reduce what the coordinator can observe, while DP can give an additional bound on what is revealed through aggregates or released models. Nevertheless, if there is a possibility that a coordinating party might be acting maliciously then it should include issues like models' inconsistency, adaptability to participants' behavior selection along with proper implementation for clipping/aggregation and noise generation. Aggregating securely is not enough to stop an orchestrator from sending corrupted models back through their clients' systems.

The collusion of customers alters these standards for protection. Honesty count and freedom to contribute are critical for security. Cross-device system might have bigger participants with less secure identities while cross-silo system has more contracts and fewer companies involved. Threshold protocol and clients' sampling would help them; however, they are still susceptible of being attacked by a sybil attack and dropping out. Latterly they might have greater reliance upon management controls, audits and better computing power; however, frequent data aggregation by many companies could lead to higher disclosure risks whereas customer level DP may be costly to implement. Thus, its utility is also varied depending upon a given network topology type.

The protected asset should determine the privacy unit. Packet- or flow-level protection may be appropriate for isolated records, but IoT traffic is often correlated across sessions, devices, households, and sites. Record-level DP may therefore leave group behavior inferable. Client-level DP may better match the privacy needs of a device or organization, but it treats what may be a large local dataset as a single protected unit and can consequently increase sensitivity. Event-level definitions may be more appropriate for time-bounded incidents, while user-level definitions may need to cover activity across multiple devices. Privacy claims should therefore justify their adjacency definitions in relation to the specific harm they are intended to prevent.

Mechanism selection should also reflect the detector's operational role. A gateway-based anomaly detector that generates local alerts may need to share only compact updates describing normal behavior. A consortium-wide multiclass detector may require collaboration around class-specific information. In a hierarchical industrial system, raw features might remain within the plant, local aggregation could occur at a regional edge, and only a global backbone could be distributed more widely. In many cases, the least-exposing design is

achieved by changing the task decomposition or sharing frequency rather than by adding another cryptographic layer. Privacy-oriented minimization and protocol design should therefore come before adding noise or encryption, while recognizing that minimization alone does not constitute a formal privacy guarantee.

The final step is falsification. For every proposed mechanism, researchers should identify an observation or attack that could disprove the claim, a baseline that demonstrates the mechanism's incremental effect, and an operational constraint that could make the approach impractical. A claim that cannot be meaningfully falsified is too vague to audit. Likewise, a design that withstands only a weak attack should clearly state that limitation. This approach shifts privacy engineering away from simply accumulating protective features and toward reducing clearly defined threats through measurable evidence.

Integrity and Availability Threats:

Poisoning and Backdoor Attacks:

An FL-based IDS gives participants a direct role in influencing a security control, which creates an important integrity risk. A compromised client may alter labels or training examples, optimize its local model toward an adversarial objective, scale or replace its update, or implant a backdoor that is triggered by a specific traffic pattern. Untargeted poisoning aims to degrade overall detection performance, while targeted poisoning seeks to suppress or misclassify a particular attack. Backdoors are especially concerning because they can preserve normal performance on clean and common-attack samples while introducing a hidden decision rule that is activated only by a specific trigger. Standard validation may therefore fail to expose them.

General FL research has shown that model-replacement and distributed backdoor attacks can exploit aggregation and persistence even when the resulting global model maintains high accuracy [43]. Domain-specific studies likewise show that FL-based IoT IDSs can be compromised through malicious local training [24]. SecFedNIDS and other dependable FL-IDS frameworks have subsequently proposed mechanisms for detecting malicious contributions or performing more robust aggregation [14][25]. The key question, however, is whether these defences address realistic attack capabilities. A compromised gateway may manipulate all traffic associated with a particular client; a botnet operator may control multiple devices; an insider at a participating site may know the model and training protocol; and a legitimate device compromised after enrollment may gradually contribute poisoned "normal" behavior.

Attack evaluations should therefore vary the proportion of malicious clients, their persistence, data access, knowledge of the defence, and participation patterns. Studies should distinguish data poisoning from direct model manipulation and report both clean detection utility and attack success. Random label flipping can provide a useful baseline stress test, but it is a weak representation of an adaptive adversary. For anomaly detection, contaminating the learned normal profile may be more realistic than manipulating class labels. In continual-learning settings, slow poisoning that follows the natural pattern of concept drift may also evade defences designed to detect sudden changes.

Robust Aggregation and Update Auditing:

Byzantine-robust aggregation aims to estimate a useful global update even when some clients send arbitrary or malicious messages. Approaches such as coordinate-wise median, trimmed mean, geometric median, Krum-type rules, norm bounds, clustering, trust bootstrapping, and learned anomaly scores rely on different assumptions about the model dimension, proportion of malicious clients, concentration of benign updates, or availability of trusted data [44][45]. These assumptions can be particularly difficult to maintain in IoT environments, where legitimate client updates may vary considerably. An update from a rare device, newly deployed firmware, regional attack burst, or minority protocol may appear

anomalous even when it is legitimate. A defence can therefore improve aggregate accuracy by rejecting precisely those clients that provide evidence of new or uncommon threats.

Robustness should consequently be evaluated alongside benign non-IID behavior rather than only after attackers are introduced. The evaluation should first establish a baseline with honest but heterogeneous clients. Reporting only global accuracy can hide client-level exclusion and harm to rare attack classes. More informative measures include per-client recall, worst-client utility, acceptance rates for benign and malicious updates, backdoor success, convergence behavior, and changes in prediction calibration. Recent theoretical and empirical research points to an inherent tension between performance without attacks and robustness to Byzantine behavior, suggesting that no fixed aggregation rule is likely to dominate across all data distributions [46].

Update auditing can rely on validation loss, update similarity, temporal consistency, influence, reputation, or provenance. Validation-based methods require a trusted and representative dataset, which can be difficult to maintain in an open-world IDS and may itself become a target. Similarity-based methods assume that benign clients form sufficiently coherent groups, an assumption that feature and label skew can easily violate. Reputation mechanisms can make use of historical behavior, but they may still be exploited by clients that behave honestly during an initial period before launching a strategic attack. They can also disadvantage legitimate clients that participate intermittently.

For these reasons, auditing should make the basis for trust explicit and allow uncertainty, quarantine, or further verification where appropriate. Treating every unusual update as definitively malicious can be as harmful as failing to detect a genuine attack, particularly in heterogeneous and continually changing IoT environments.

Sybil, Collusion, and Availability Attacks:

A sybil attack compromises security by counting number of users irrespective to whether they are administrators and hardware owners separately. One bad actor could be able to spread out their updates on several accounts; they might have more power under majority rule systems; they may also manipulate ratings for others' accounts while reducing how anonymous it is when aggregating data together through secure aggregation techniques. FoolsGold-style defenses use similarity between clients' history to identify coordinated behavior, but attackers can intentionally vary their updates, while honest clients using similar tools might generate correlated updates [47]. Identity enrollment, device attestation, organizational quotas, rate limits, and governance should also be included as part of our robustness strategy since they are typically not part of an ML study.

Risks are associated with collusion on privacy and security issues respectively. An agent of a client could try isolating someone else's work; clients might share key information and updates. A hacker might even collaborate on poisoning while still adhering to acceptable limits for individual updates. Although secure aggregation protocol usually assumes a certain threshold for success, PPFL-IDS do not evaluate this relationship between thresholds and clients' dropouts, participants' selection processes and organization's credibility levels. Small or highly intermittent federations are especially susceptible since every participation event provides more information on an individual client, whereas dropouts contribute less truthfully collectively.

Threats to availability go beyond statistical attacks such as dropping clients; delaying/stale information delivery; spam messages; bad encryption; overloading resources; tampering with voting processes (manipulation). An attack-proof statistics-based system does not mean it's also a denial-of-service at the protocols level. Cryptography may also raise costs for these types of attacks by requiring more expensive verifications and decryptions prior to rejecting bad input data. Asynchronous training will help to minimize delays from laggards;

however, this approach is prone to stale updates which hackers might take advantage of easily enough.

The full PPFL-IDS specification must include details about timeout management; retry mechanisms for malformed update messages; minimum aggregation threshold values; coordinator failures; and recovery procedures. Also, it needs to indicate that this mechanism affects privacy if there are few customers left to aggregate them together.

Adversary–Defense Evidence Matrix:

The principal threats, attacker capabilities, relevant controls, and residual or displaced risks identified across these scenarios are consolidated in the Adversary–Defense Evidence Matrix in Table 8.

Table 8. Adversary–Defense Evidence Matrix

Threat	Attacker capability	Relevant controls	What may be reduced	Residual or displaced risk
Gradient / update inference	Observe client messages or isolate cohorts	Secure aggregation, DP, perturbation, minimization	Direct reconstruction, membership or attributes	Aggregate/final model, metadata, collusion
Model poisoning	Manipulate local data, labels, optimizer or update	Clipping, robust aggregation, validation, influence audit	Untargeted degradation	Adaptive in-bound attacks; benign heterogeneity
Backdoor	Optimize targeted trigger while preserving clean utility	Robust aggregation, trigger testing, pruning/inspection, rollback	Known-trigger success or persistence	Unknown triggers; secure aggregation visibility tension
Sybil / collusion	Create identities or coordinate parties	Enrollment, attestation, quotas, similarity/reputation, thresholds	Vote/weight capture and isolation	Identity privacy, false linkage, adaptive diversity
Availability	Drop, delay, flood, send malformed or stale updates	Threshold protocols, verification, timeouts, asynchronous recovery	Round failure and resource exhaustion	Smaller anonymity sets and biased participation
Malicious coordinator	Send tailored models, alter groups or protocol execution	Verifiable computation, consistency checks, TEE/MPC, client audit	Equivocation and incorrect aggregation	Hardware/trust assumptions; side channels

The matrix makes a central limitation visible: most defenses cover one stage of one attack path. Clipping limits magnitude but not a carefully optimized in-bound update. Robust aggregation reduces some poisoning effects but does not hide updates. Secure aggregation hides updates but restricts inspection. DP may bound individual influence yet does not prevent a client controlling many records or identities. Provenance and attestation establish software or identity claims but do not guarantee benign data. The required research objective is composable assurance: a design should explain how mechanisms interact, which assumptions remain, and which failures are detected versus merely tolerated.

Composite and Adaptive Evaluation:

Modular defense strategies are often preferred by single threat tests since these test attacks individually rather than collectively. It would be better if we tested full attack chain

scenarios that reflect what an attacker might do with current defenses. One possible attack chain might start by compromising identities; use several sybils for influencing participation; introduce an underpowered backdoor that is within clipping limits; finally leverage secure aggregation to mask individual contributions. Another may be an attacker coordinating with some customers to send them personalized model instances that isolate them from others while leaking information over several iterations. Another case would be that there is a delay on networks and outdated information so that customers who are not fraudulent will look suspicious then get deleted whereas fraudsters' data will stay current. These situations are more realistic because hackers will adjust according on what defense system is being used by an organization's defenders.

A composite test is not a requirement for all studies to run a full-scale red team exercise. A tiered testing approach is more practical to use instead of this one. Tier 1 may be able to perform an adaptive, non-adaptive attack and also have no-attack as their baseline. Tier 2 might provide attackers with information about deployed defenses so they can adjust strategies based upon those limits. Tier 3 could have at least two attack capabilities or stages, such as Sybil participation with poisoning, or client dropout with collusion. Reporting the testing tier would be easier to compare between studies and less likely to treat a fixed attack implementation as the upper limit of adversarial capabilities.

Integrity assessment also includes recovery which is frequently overlooked. A intrusion detection system might detect an attack on training systems later when there is substantial harm done. Thus, it is necessary for research to include information about checkpoints; rollbacks; models comparison; clients quarantined keys revoked and returned. Also check if there are back doors left on after removing attackers' code; do not forget about forensics that may cause more problems with your security. A system that eventually recovers model performance but is not able to identify a certain type of attack for several hours might also be technically unacceptable. Time of detection, time of recovery, and cumulative missed alert rates are some examples that could be used for translating models' resilience to actual risks on operations level.

Governance of humans also matters a lot especially for a cross-silo federation model. A company might be divided on if this strange act was really evil; due to their own procedures; because something happened locally. Thus, effective procedures should include appeals process; evidence preservation and accountability without compromising confidentiality issues. Reputation rating is not allowed for taking permanent decisions of credibility with proper supervision. The above mechanisms are not substitutes for strong security measures; however, these play a vital part when it comes to managing uncertainties, false alarms and unjustified allegations within an actual government system.

Experimental Evidence and Deployment Realism:

Dataset Validity and Generalization:

Some examples of PPFL-IDS can be obtained through replaying attack samples that are publicly available. It's understandable since real world attack attacks are hard to replicate and operational data is usually confidential whereas datasets allow for comparison of results from various research projects. Nevertheless, though good results are usually indicative of effective functioning for a given dataset and processing workflow rather than being able to generalize effectively across other deployments such as those involving iot devices. The model might also pick up on test-bed specific id's, artefacts generated through attacks tooling and persistent devices' fingerprints. While cross-dataset evaluations may show significant variations between datasets; however, they do not necessarily indicate good generalization since features' definitions and labels are usually defined priorly for comparison purposes.

Therefore, an improved testing strategy would be to use both group aware and temporal split of information sets together. A device level holdout tests if models transfer

from current devices to new ones whereas sites are held out for evaluating cross-organization performance. Temporal holdout can show the impact of data drift, and attack family holdout gives an incomplete picture to evaluate performance on unseen attacks. Testing across datasets may also reveal disparities between data gathering methods, features. None of these tests by themselves is an example for true zero-day security. However, taken as a whole these reduce the chances of reporting results being due to overlap between tests/training sets rather than actual differences between them. The preprocessing needs to be applied solely for training set. Normalization, features selection etc. should not depend upon statistics derived from global test data during federated learning process.

Data sets are public and must have a good history of their origins. It must describe how this was captured; what version of data set it came from; which filters were applied; duplicates removed; classes mapped onto features sampled randomly for clients' purposes. This is especially relevant for research studies where they refer back to an original version of data rather than a modified one. The results of N-BaIoT, Bot-IoT, ToN_IoT, Edge-IIoTset, and CICIoT2023 are not comparable to each other since they are all about IoT traffic. They are different by type of devices, attack scenarios, features set, and data collection conditions [36][37][38][39][40]. Another example is that performance was poor due to testing on various platforms whereas federated learning performed better with diverse datasets included during training process. This means that good results on an individual test set do not necessarily mean that your model can generalize well enough for practical use [48].

Federated Client Construction:

Federated learning client architecture has been an important choice for designing models but rarely reported properly enough. Randomly partitioning a row to an independent and identical distribution (IID) client is usually used for evaluating distributed optimizations with little similarity among them. It is not a good representation of what an actual federation looks like. Label partitioning based on dirichlets is an effective method for controlling levels of labels' skewness; however, this approach does not account for variations due to device types and protocol differences over time and organizational boundaries. Also, shard-based partitioning might lead to unrealistic large disparities among customers; it could also inadvertently associate variability with random orderings of rows. If possible then partition by device type, location of data collection point; network segment and organization and period would give better picture for federation model.

An effective assessment must include a minimum of three construction project cases for clients. Firstly, it is an IID diagnostic split that will help to detect implementation/ optimization issues. Secondly it's a controlled synthetic non-IID split, with the partition parameter and random seed clearly stated to ensure reproducibility of this study. Third, however, needs to be an organic, application focused division which reflects what customers will really be doing. Each study needs to be summarized by summarizing a clients' sample size; classes distribution features schedule of participation. Results should also distinguish between cross-device federations, which may involve many resource-constrained clients, and cross-silo federations, where only a small number of organizations participate. They vary greatly regarding privacy unit, trust assumption, client churn rate and feasibility for secure aggregation.

The non-IID architecture impacts attack modeling too. Malicious clients must then have been chosen based upon reasonable distribution of customers instead of being randomly sampled at all times. For instance, an attacker with access to a rare type of device might naturally generate an update that seems strange, whereas a large organization may have more influence over sample-weighted aggregation. Also, a botnet might produce lots of small-scale sybils instead of just a few big ones. Therefore, defense claims must not necessarily mean that they have been made for balanced client sizes only then it is advisable to interpret these

statements carefully based upon their respective clientele structures on which these evaluations are conducted.

Metrics, Baselines, and Statistical Evidence:

Accuracy by itself cannot be a good measure of imbalanced intrusion detection system. Therefore, study reports must include class-wise accuracy, precision and F1 score; macro average of these scores; false positives rates for each class; precision-recall curves (PRC) respectively. Also include rates of noise production for each period of measurement if relevant. It's especially crucial for calibrating models that use score-based responses to activate various levels of reaction. For open set/anomaly detection problems detection delays and capability of detecting new attacks not seen before would be better indicators compared with closed-set classifiers' accuracies scores.

Privacy and security must also be assessed with respect to its own measure of success. To evaluate for a reconstruction attack then it is important to focus more upon semantic correctness along with recovering specific attributes instead of just distances between them. Attack advantage or some other similar non-threshold-based metric should be reported by membership inference study. The poisoning experiment must measure attack efficacy; accuracy with respect to unpolluted samples; and client-specific effects. Differential privacy (DP) researchers must state what is their privacy budget, how they account for it. The system level evaluation must take these factors under consideration such as amount of information transferred over time; number of round trips needed for reaching end-user; wall clock time; computation at endpoints; power usage (if applicable) along with other things like encryption costs.

Baseline selection must enable isolation for every treatment's impact on outcome measures. PPFL-IDS experiments are likely to be better compared to local-only, centralised, and private FedAvg baselines (and possibly also heterogeneous). A test attack should also be conducted on an unprotected system with some countermeasures available to defend it. A clipping test with no noise is an important control for DP experiments. Also, compare encryption-based implementations to those that use identical algorithms for teaching but have different levels of protection on their heads. Aggregation techniques are also worth evaluating on benign scenarios for measuring efficiency and cost of computation without attacks occurring at all times. Ablation tests must eliminate every meaningful element within this model instead of comparing it with unrelated classes alone.

A result from an individual trial may not accurately reflect results of many trials together. The random seed must influence every relevant source for variation such as data splits; models' initializations; clients' choices; attacks' optimizations. Then results must also report from several trials with appropriate statistics for mean, variance. A statistical test has value if you define what "experimental unit" means properly enough. Repeated epochs are not to be considered as separate replicates; also, each flow obtained during one sampling event cannot count towards another's deployment. Often, a confidence interval derived from a client's data set and another partition of their own dataset is more informative than an individual test conducted for each record.

Operational Evidence and Reproducibility:

Most PPFL-IDS testing is done with a simulation of workstation/server machines. Simulations are helpful to conduct controlled experimentation and comparative studies; however, they don't prove how well your method would work with limited resources at hand. Hardware based testing with devices like a Raspberry Pi client and a Jetson Xavier server provides better proof of feasibility, although it does not guarantee privacy nor readiness to deploy on an actual system [49]. For instance, a raspberry pi gateway is limited to certain types of sensors/devices whereas an encryption measurement taken from a computer does not include power consumption nor communications costs for batteries and wifi modems. Thus,

deployments must specify what is being deployed at every level, endpoints, gateways, edge servers, clouds, along with their architecture, hardware, software stack, model size, time-to-market, bandwidth availability, network loss rate, number of clients participating during testing period.

The communication cost must also be expressed as a byte rate instead of just numbers for models' parameters and communications round numbers. Updates to models; masks for data; keys exchange; proof of authenticity, retry attempts and metadata are some examples that may significantly increase communication costs. Also, computation cost must include features extraction and encryption processes along with models' local learning process. The latency measurement must differentiate between local processing time; inter-network communication time; data aggregation times and inference times to identify sources for delays. If conserving resources is your main goal for this study, then measure actual consumption rather than estimate it through proxy data. Also, longer duration trials need to take care of operational problems like bad models update failures alerts rollbacks drift etc along with getting more/less accurate data points.

Reproducibility is about giving enough details so that another researcher can reproduce this study. Source codes; configurations file; datasets version/retrieval method; preprocessing steps; random seed value; client partitioning strategy for attacks; privacy accounting parameters and hardware/software environment information. Privacy and security sensitive parts might need to be released gradually; however not releasing every vulnerability related implementation would be hard for testing independently. Containerized environments and machine-readable experiments may also help to reduce ambiguity, increase reproducibility of results between experiments. The parameter selection for a cryptographic system must also include documentation of its implementation details. Similarly, DP tests must also contain enough details regarding an accounting system's implementation and sampling assumptions so that their privacy claims can be verified by others independently of them.

Evidence-Quality Checklist:

The reporting requirements discussed above are consolidated into the evidence-quality checklist in Table 9, which distinguishes minimum credible reporting from stronger forms of empirical and methodological evidence.

Table 9. Evidence-Quality Checklist for PPFL-IDS Studies

Domain	**Minimum credible report**	**Stronger evidence**	**Common warning sign**
Threat model	Asset, unit, adversary, observables and assumptions	Adaptive / colluding adversaries and failure analysis	"FL ensures privacy"
Federation	Client meaning, counts, sizes, participation and partition	Natural device/site plus controlled sensitivity splits	Random rows called non-IID
Detection	Macro/per-class metrics and false-positive behavior	Temporal, device/site, attack-family and cross-dataset holdout	Accuracy-only ranking
Privacy	Parameters and matched attack or formal definition	Several attacks, composition and task-specific harm	Noise/encryption named without settings
Integrity	Clean and attacked baselines; attacker fraction/capability	Adaptive/composite attacks and recovery	Only random label flipping

Statistics	Seeds, repeated runs and dispersion	Correct experimental unit and uncertainty across clients/partitions	Single best run
Systems	Bytes, time, hardware and component placement	Energy, loss/dropout, long-running testbed or field trial	Parameter count used as communication
Reproducibility	Code/configuration, data version, preprocessing and partitions	Containers, manifests, attack code and claim traceability	Unspecified derived dataset

The checklist is designed for both reviewers and authors. It does not demand a field trial from every algorithm paper. It demands alignment between the strength of the claim and the strength of the evidence. A simulation can support an algorithmic statement; it should not be presented as proof of real-time, energy-efficient, privacy-compliant deployment. A formal theorem can establish a property under assumptions; it should not obscure absent attack tests or systems cost. Transparent boundaries increase, rather than diminish, the value of a study.

A Minimum Reporting Protocol for Comparable Studies:

An abbreviated basic reporting requirement might significantly enhance comparability among PPFL-IDSs research projects. Then, researchers must precisely define the unit of federation by specifying which clients are members; how many there are; how much information they have to offer; how clients contribute towards their education. Secondly, please provide access to the data partitioning process via an api endpoint (partition manifest) and by providing codes along with random seed values needed for reproducing this experiment. Thirdly, please document all details of your training process such as epoch number; batch length; optimization algorithm (optimizer); learning rates; clipping strategy; aggregated weight calculation method for clients' selection procedure stops condition and initialization parameters. Fourth, the threat model and privacy unit must also exist prior to describing the privacy mechanism. Finally, there must also exist enough information about parameter settings for replication of this security promise by other researchers and assessment of their respective attack vectors' scope.

Local-only, centralized, and unprotected FL baselines should be included; macro-averaged and per-class detection results; client-level performance variation; multiple random seeds; at least one data split that is realistic for deployment. A privacy claim must have an associated threat assessment and an explicit certification of security assurance. Strength claims must account for cleanliness and attack ability along with variability of benign types. Also, efficiency of resources must have been justified through cost analysis for communications and operations with defined equipment. The authors must differentiate between value chosen by means of validation set tests and those obtained from test sets at end of experimentation phase. Attack evaluation must state if attacks are optimized for defending what is being tested here.

Tables results must not focus on one high-performing score only. Instead of this, an "assurance panel" is more concise and can be used to describe effectiveness of detection; confidentiality guarantees / attacks results; integrity performance; client behavior (worst case); communication cost/latency/evidence strength. If there is no evaluation of this dimension then report it as 'not evaluated' instead of its omission would be wrong. It encourages researchers to specify what is being measured rather than creating an aggregate score that's made up out of random weights. Also, it enables consumers' judgment about their relevance for an inter-device consumer network; cross-silo industrial setting; and controlled test lab scenario.

A claims document is needed for reproducibility of this work, as it should link every significant result with an experiment; theorems; sources used to prove them. This kind of

traceability is more frequently used for safety and system assurance studies but not so much with ml research. Similarly, this approach is also applicable for reviewing papers where every research findings are linked with their respective articles along with reviews by reviewers. An extractable database plus a list of exclusions will also help to update systematic review information more easily without repeating interpretations again.

Also, study should clearly mention what is not studied. It could also be related to collusion between clients; server activity; time-variation of power usage and new hardware that was never tested before. An accurate limitation statement regarding limitations should not be as general as that which refers to future research since it clearly defines what we have to work with currently. Advance progress of similar PPFL-IDS studies would be less dependent upon employing similar model architectures but rather consistent reporting regarding fundamental assumptions; partitioning of datasets used for attacks and their associated expenses.

Research Gaps and Prioritized Roadmap:

From Privacy Labels to Auditable Privacy Claims:

Semantic is probably the biggest issue right now. Privacy preserving refers to techniques like local storage of information; encryption of messages; noise injection during machine learning models' update process and secure aggregation methods where neither privacy nor identity are specified for attackers targeting them respectively. Lack of specificity prevents comparisons between research efforts to be made easily; it also hinders cumulative development within this area of study. Therefore, a standard privacy claim card needs to be reported on each PPFL-IDSs research project. At least it must specify what is being protected (asset), privacy unit; adversary model; observable information; protection mechanism; formal guarantee; parameters; supporting evidence; residual risk and cost. This data may be useful for automating audit processes; it can help with review systems and benchmarks filtering. Journals and conferences might promote that approach through considering an unnamed threat scenario a reporting gap.

The evaluation must be closely related to that particular issue of privacy being claimed here. For instance, a defense strategy to prevent gradient leakage must also be tested on various reconstructions of data and threats that are pertinent for detecting them correctly. Also, an inference about membership should be stated explicitly as to what is assumed for models' access and attackers' prior knowledge. The claims of secure aggregation need to address issues related to client dropping out and collusion; meanwhile, client level differential privacy must consider all forms of information leakage during a full cycle from data collection through model deployment. The negative outcome, case study of failure for an effective safeguard is also significant as it helps define its limits to be protected by that particular system/product/service.

Composing Confidentiality, Integrity, and Availability:

Robustness without compromising privacy is a key issue here. Current secure aggregations are limited to provide enough information for some strong aggregation techniques; auditing changes directly on data is risky as it may reveal private details. Thus future studies should concentrate more upon secure computation of robust statistics, verifiable clipping, privacy-preserving anomaly detection, and threshold based protocol which are still functional even if some clients leave. The design should be able to resist adaptive poisons and sybils that would unfairly exclude lawful minorities from a non-IID setting.

Formalization is another aspect that needs to be done for effective writing. Privacy-preserving technologies like Differential Privacy (DP), encryption, trusted hardware, attestation, and reputation have varying levels of trust assumptions to protect them from various types of attacks. Therefore, their combination needs to be depicted with an explicit trust graph and a systematic failure analysis. It's more about identifying potential threats,

compromised parties, failures, that would compromise security/confidentiality/availability rather than creating a foolproof solution for detecting/catching/recovering these incidents.

Privacy-Preserving Detection Under Open-World Change:

Most PPFL-IDS studies are conducted with an assumption that they're classified as stationary, closed set problems. However, in practice it's dynamic due to more sensors being deployed; newer software versions being installed; changing loads on systems; and evolving attacks by hackers. Learning by itself increases more models' release rate that will increase privacy budgets and attack surfaces. While personalization may increase local relevance, it is harder to audit and coordinate changes for a common model. Therefore, future studies may want to investigate drift detection, rehearsal/memory mechanisms, update scheduling, privacy accounting and backdoor persistence over time.

The open world test must be based on this scenario with time-stamped information; new types of attacks that have not been encountered before; late/low-quality label updates; devices being replaced by others; and recovery after a bad adaptation. The privacy mechanism needs to be assessed over its entire life cycle as an instrument for detecting signals instead of just during tests only. Selective updates, privacy budget renewals; machine learning unlearning and auditable retirement of old clients' data could be interesting avenues to explore further though these are still quite immature when it comes to IoT security solutions.

From Simulated Federation to Operational Evidence:

There is no standard problem set for this area which includes realistic clients' partitioning; attack script generation; privacy attack simulation; network disruption testing and resource measurement. The evaluation framework should preserve the diversity of IoT environments rather than ranking all systems on a single leaderboard. Rather than that, it must be defined as separate categories like home/office; hospital/campus; industry/healthcare and city/smart city. Reporting must be evaluated to ensure its relevance for each claim being made. Containerized client models, reproducible transformation of collected information and hardware-in-the-loop profiles might be able to close this divide between simulations and test runs.

Validation of operations must be done gradually by tracing simulations; emulations on realistic network types and devices; controlled tests on live traffic/attacks; shadow mode deployments then actual field testing with human alerts. With more emphasis on implementation of evaluations for safety issues; management problems and privacy concerns are becoming critical now-a-days. Reporting on failure of a test cycle, false alarms, operators' workloads and maintenance cost will be much more useful to others than just telling them about another single high point for precision.

Prioritized Research Agenda:

The resulting research priorities, their minimum deliverables, dependencies, and proposed success criteria are summarized in Table 10.

Table 10. Prioritized Research Agenda

Priority	Research program	Minimum deliverable	Dependency	Success criterion
1	Standardized privacy-claim card	Human- and machine-readable claim, parameters, evidence and residual risk	None	Claims become comparable and falsifiable
1	Shared PPFL-IDS challenge protocols	Natural partitions, attack tiers, privacy attacks, resource profiles	Claim schema	Reproducible scenario-level comparison

1	Private Byzantine-robust aggregation	Secure/verified robust statistics under non-IID data and dropout	Explicit trust graph	Integrity without exposing individual updates
2	Lifetime privacy and continual IDS	Composition across drift, updates, unlearning and backdoor recovery	Claim/accounting standards	Bounded lifetime leakage with retained detection value
2	Cross-domain and open-world evaluation	Temporal, unseen-device/site and unseen-attack scenarios	Challenge suite	Generalization reported without overlap inflation
3	Resource-aware field assurance	Hardware-in-loop, shadow mode and monitored field trials	Stable mechanisms/ protocols	Measured utility, privacy, robustness and operating cost

Note. Priority 1 establishes shared semantics and technical foundations; later programs depend on those foundations rather than simply on calendar time.

The roadmap is dependency-aware. Standardized claims and shared protocols are near-term infrastructure because they make later comparisons possible. Private robust aggregation and lifetime privacy accounting are high-priority technical foundations. Continual, cross-domain, and edge-constrained trials should build on those foundations rather than treating a sophisticated prototype as inherently deployable. Success should be judged by assurance coverage and reproducibility as well as predictive performance. The dependencies among the proposed research priorities are illustrated in Figure 4, linking standardized claims and evaluation protocols to technical foundations and, ultimately, operational assurance.

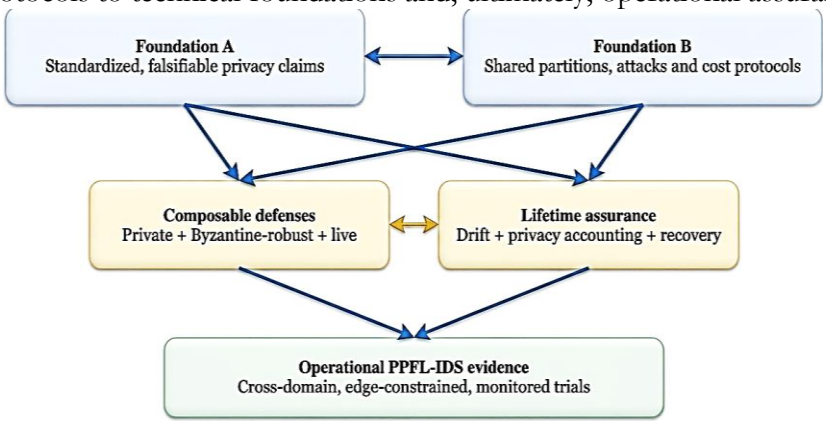

Figure 4. Dependency-aware roadmap from auditable claims to operational assurance

Practical Implications and Recommendations:

The findings of this review provide practical guidance for researchers, reviewers, and editors evaluating federated learning (FL)-based IoT intrusion detection systems. A localized approach is not a panacea for privacy protections, and views on privacy and security should be accompanied by well-specified threat models and sufficient empirical or formal evidence to support them.

Researchers should explicitly state the proposed mechanisms, parameters involved, conflicting assumptions and risks left. Detection performance should not be considered when evaluating privacy protection and computational overhead, communication costs, and constraints on IoT resources should be reported.

The reviewer and editor may use the following E0–E3 classification system suggested in this review to evaluate the level of support for each claim. The following questions should be taken into account when conducting a practical assessment:

Is the threat model and the information exposed during training clearly described?

Are privacy and security mechanisms sufficiently defined and analyzed against relevant attacks?
Are detection performance, residual risks and resource requirements reported?
Are the conclusions in proportion to the evidence presented?
This checklist is designed to be a helpful tool for assessing in a structured way rather than a guarantee of security or privacy. Statements of claim should only be restricted to protections shown under the assumption and evaluation conditions provided.

Conclusion:

Federated Learning (FL) changes where IoT intrusion detection data is processed, but this alone does not guarantee privacy. Updates to models; summary data; model releases; metadata; collaborative learning processes are all potential targets of observation/ manipulation that should be clearly stated here. Literature shows that decentralized IDS training is possible and offers some good defense options such as Differential Privacy (DP), secure aggregation, encryption of computation, perturbation, robust aggregation, personalization, and trusted execution. Nevertheless, there is still a lot of information scattered among various threats model types; clients' settings; test results; and deployment scenarios.

Thus, this report goes further than just tallying up measures for privacy protection; it is more concerned with evaluating what these approaches claim to be doing. The architecture links a protected assets/privacy unit to an attacker; observable data; defense strategy; security assurance; test results; residual risks and implementation costs. Confidentiality, integrity, availability, data heterogeneity, and resource constraints are interrelated as per this study's findings. For instance, secure aggregation may be more difficult to analyze changes on poisoning; clipping might hide information about legitimate rare clients' behavior; robustness measures could confuse normal variations with attacks; and constant updates would use up privacy budgets while giving chances at slow, persistent poisons. They're not reasons for giving up on FL; they are trade-offs that need to be made.

A journal quality PPFL-IDS report must pass these four tests – clearly define what you are protecting and by whom; propose an approach that's suitable for this particular protection claim; evaluate whether your solution covers those boundaries with respect to relevant data partitioning scenarios; report on both value and cost effectiveness. Progress is meaningful if we have standardised claims card; reproducible challenges; privacy-preserving strong aggregations of data over time; lifetime privacy accounting; staged operational evaluations. Until such practices become routine, the term "privacy-preserving" should be treated as a claim that requires evidence, rather than as a property automatically provided by using federated learning.

References:

[1] B. McMahan, E. Moore, … D. R.-A. intelligence, and undefined 2017, "Communication-efficient learning of deep networks from decentralized data," *proceedings.mlr.pressB McMahan, E Moore, D Ramage, S Hampson, BA y ArcasArtificial Intell. Stat. 2017•proceedings.mlr.press*, Accessed: Oct. 04, 2026. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a?ref=https://githubhelp.com

[2] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting Gradients -- How easy is it to break privacy in federated learning?," *Adv. Neural Inf. Process. Syst.*, vol. 2020-December, Mar. 2020, Accessed: Oct. 21, 2026. [Online]. Available: https://arxiv.org/pdf/2003.14053

[3] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting Unintended Feature Leakage in Collaborative Learning," *Proc. - IEEE Symp. Secur. Priv.*, vol. 2019-May, pp. 691–706, May 2018, doi: 10.1109/SP.2019.00029.

[4] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and

Federated Learning," *Proc. - IEEE Symp. Secur. Priv.*, vol. 2019-May, pp. 739–753, Jun. 2020, doi: 10.1109/SP.2019.00065.

[5] L. Zhu and S. Han, "Deep Leakage from Gradients," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12500 LNCS, pp. 17–31, Jun. 2019, doi: 10.1007/978-3-030-63076-8_2.

[6] A. Belenguer, J. A. Pascual, and J. Navaridas, "A Review of Federated Learning Applications in Intrusion Detection Systems," *Comput. Networks*, vol. 258, p. 111023, Feb. 2025, doi: 10.1016/J.COMNET.2024.111023.

[7] T. Bunko, M. N. Johnstone, W. Yang, and B. A. Scott, "A survey of privacy-preserving federated learning for intrusion detection systems," *Artif. Intell. Rev. 2026 595*, vol. 59, no. 5, pp. 125-, Feb. 2026, doi: 10.1007/S10462-026-11519-4.

[8] E. M. Campos *et al.*, "Evaluating Federated Learning for intrusion detection in Internet of Things: Review and challenges," *Comput. Networks*, vol. 203, p. 108661, Feb. 2022, doi: 10.1016/J.COMNET.2021.108661.

[9] J. L. Hernandez-Ramos *et al.*, "Intrusion Detection based on Federated Learning: a systematic review," *ACM Comput. Surv.*, vol. 57, no. 12, Aug. 2023, doi: 10.1145/3731596.

[10] L. Lavaur, M. O. Pahl, Y. Busnel, and F. Autrel, "The Evolution of Federated Learning-Based Intrusion Detection and Mitigation: A Survey," *IEEE Trans. Netw. Serv. Manag.*, vol. 19, no. 3, pp. 2309–2332, Sep. 2022, doi: 10.1109/TNSM.2022.3177512.

[11] A. Vyas, P. C. Lin, R. H. Hwang, and M. Tripathi, "Privacy-Preserving Federated Learning for Intrusion Detection in IoT Environments: A Survey," *IEEE Access*, vol. 12, pp. 127018–127050, 2024, doi: 10.1109/ACCESS.2024.3454211.

[12] W. Almuseelem, "Collaborative intrusion detection systems in SD-WAN enterprise networks with federated learning: a comprehensive survey," *J. King Saud Univ. Comput. Inf. Sci. 2026 386*, vol. 38, no. 6, pp. 379-, May 2026, doi: 10.1007/S44443-026-00625-6.

[13] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3–4, pp. 211–487, 2013, doi: 10.1561/0400000042

[14] Z. Zhang, Y. Zhang, D. Guo, L. Yao, and Z. Li, "SecFedNIDS: Robust defense for poisoning attack against federated learning-based network intrusion detection system," *Futur. Gener. Comput. Syst.*, vol. 134, pp. 154–169, Sep. 2022, doi: 10.1016/J.FUTURE.2022.04.010.

[15] M. J. Page *et al.*, "The PRISMA 2020 statement: an updated guideline for reporting systematic reviews," *Syst. Rev.*, vol. 10, no. 1, Dec. 2021, doi: 10.1186/S13643-021-01626-4.

[16] M. L. Rethlefsen *et al.*, "PRISMA-S: an extension to the PRISMA Statement for Reporting Literature Searches in Systematic Reviews," *Syst. Rev.*, vol. 10, no. 1, Dec. 2021, doi: 10.1186/S13643-020-01542-Z.

[17] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated-Learning-Based Anomaly Detection for IoT Security Attacks," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2545–2554, Feb. 2022, doi: 10.1109/JIOT.2021.3077803.

[18] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A. R. Sadeghi, "DÏoT: A federated self-learning anomaly detection system for IoT," *Proc. - Int. Conf. Distrib. Comput. Syst.*, vol. 2019-July, pp. 756–767, Jul. 2019, doi: 10.1109/ICDCS.2019.00080.

[19] P. Ruzafa-Alcázar et al, "Intrusion Detection Based on Privacy-Preserving Federated

Learning for the Industrial IoT," *IEEE Trans. Ind. Informatics*, vol. 19, no. 2, pp. 1145–1154, 2023, doi: 10.1109/TII.2021.3126728.

[20] Y. Sun, H. Esaki, and H. Ochiai, "Adaptive Intrusion Detection in the Networking of Large-Scale LANs with Segmented Federated Learning," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 102–112, 2021, doi: 10.1109/OJCOMS.2020.3044323.

[21] R. M. P. and S. P. D. C. Attota, V. Mothukuri, "An Ensemble Multi-View Federated Learning Intrusion Detection for IoT," *IEEE Access*, vol. 9, pp. 117734–117745, 2021, doi: 10.1109/ACCESS.2021.3107337.

[22] Z. Jin, J. Zhou, B. Li, X. Wu, and C. Duan, "FL-IIDS: A novel federated learning-based incremental intrusion detection system," *Futur. Gener. Comput. Syst.*, vol. 151, pp. 57–70, Feb. 2024, doi: 10.1016/J.FUTURE.2023.09.019.

[23] S. Sun, L. Zhou, Z. Wang, and L. Han, "Robust intrusion detection based on personalized federated learning for IoT environment," *Comput. Secur.*, vol. 154, p. 104442, Jul. 2025, doi: 10.1016/J.COSE.2025.104442.

[24] "Poisoning Attacks on Federated Learning-based IoT Intrusion Detection System - NDSS Symposium." Accessed: Oct. 20, 2026. [Online]. Available: https://www.ndss-symposium.org/ndss-paper/auto-draft-51/

[25] R. Yang, H. He, Y. Wang, Y. Qu, and W. Zhang, "Dependable federated learning for IoT intrusion detection against poisoning attacks," *Comput. Secur.*, vol. 132, p. 103381, Sep. 2023, doi: 10.1016/J.COSE.2023.103381.

[26] S. S. Anwar, Asaduzzaman, and I. H. Sarker, "A differential privacy aided DeepFed intrusion detection system for IoT applications," *Secur. Priv.*, vol. 7, no. 6, p. e445, Nov. 2024, doi: 10.1002/SPY2.445.

[27] J. Chen, Y. Zhao, Q. Li, X. Feng, and K. Xu, "FedDef: Defense Against Gradient Leakage in Federated Learning-based Network Intrusion Detection Systems," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 4561–4576, Aug. 2023, doi: 10.1109/TIFS.2023.3297369.

[28] O. Friha, M. A. Ferrag, M. Benbouzid, T. Berghout, B. Kantarci, and K. K. R. Choo, "2DF-IDS: Decentralized and differentially private federated learning-based intrusion detection system for industrial IoT," *Comput. Secur.*, vol. 127, p. 103097, Apr. 2023, doi: 10.1016/J.COSE.2023.103097.

[29] I. Ali Soomro, H. Ur Rehman Khan, S. Jawad Hussain, A. Iqbal, W. Khalid, and H. Yu, "SecureDyn-FL: A Robust Privacy-Preserving Federated Learning Framework for Intrusion Detection in IoT Networks," *IEEE Trans. Netw. Serv. Manag.*, vol. 23, pp. 1742–1765, 2026, doi: 10.1109/TNSM.2025.3647642.

[30] I. Ali Soomro *et al.*, "ROCHE: A Robust and End-to-End Privacy-Preserving Federated Learning Framework for Intrusion Detection in Industrial Internet of Things," *IEEE Internet Things J.*, vol. 12, no. 24, pp. 52357–52377, Dec. 2025, doi: 10.1109/JIOT.2025.3612944.

[31] S. Mankotia, D. Conte de Leon, and B. P. Rimal, "FedPrIDS: Privacy-Preserving Federated Learning for Collaborative Network Intrusion Detection in IoT," *J. Cybersecurity Priv. 2026, Vol. 6, Page 10*, vol. 6, no. 1, p. 10, Jan. 2026, doi: 10.3390/JCP6010010.

[32] K. Yang *et al.*, "HFL-DP NIDS: A hierarchical federated learning-based network intrusion detection method with differential privacy," *Internet of Things*, vol. 38, p. 101967, Jul. 2026, doi: 10.1016/J.IOT.2026.101967.

[33] R. Anandhi and G. U. Srikanth, "Federated sparse-learning framework for detecting low-rate stealth attacks in resource-constrained IoT devices," *Sci. Reports 2026 161*, vol. 16, no. 1, pp. 21298-, May 2026, doi: 10.1038/s41598-026-46531-9.

[34] S. K. Sonker, V. K. Raina, B. B. Sagar, and R. C. Bansal, "Differentially private

federated adversarial learning–based intrusion prevention for V2X communications," *Discov. Comput. 2026 291*, vol. 29, no. 1, pp. 480-, Jul. 2026, doi: 10.1007/S10791-026-10398-1.

[35] S. Reddy and S. V. N. S. Kumar, "Fed fusion-IDS: a privacy-preserving federated intrusion detection system with multi-objective feature optimization and explainable deep learning," *Sci. Reports 2026*, Sep. 2026, doi: 10.1038/S41598-026-70386-9.

[36] Y. Meidan *et al.*, "N-BaIoT-Network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Comput.*, vol. 17, no. 3, pp. 12–22, Jul. 2018, doi: 10.1109/MPRV.2018.03367731.

[37] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning," *IEEE Access*, vol. 10, pp. 40281–40306, 2022, doi: 10.1109/ACCESS.2022.3165809.

[38] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Futur. Gener. Comput. Syst.*, vol. 100, pp. 779–796, Nov. 2019, doi: 10.1016/J.FUTURE.2019.05.041.

[39] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and Adna N Anwar, "TON-IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165130–165150, 2020, doi: 10.1109/ACCESS.2020.3022862.

[40] A. A. G. Euclides Carlos Pinto Neto, Sajjad Dadkhah, Raphael Ferreira, Alireza Zohourian, Rongxing Lu, "CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment," *Sensors*, vol. 23, no. 13, p. 5941, 2023, doi: https://doi.org/10.3390/s23135941.

[41] K. Bonawitz *et al.*, "Practical secure aggregation for privacy-preserving machine learning," *Proc. ACM Conf. Comput. Commun. Secur.*, pp. 1175–1191, Oct. 2017, doi: 10.1145/3133956.3133982

[42] A. A. A. ugli and J. S. Shin, "SAFE-IDS: A privacy-preserving framework for overcoming non-IID challenges in federated intrusion detection," *Comput. Secur.*, vol. 155, p. 104492, Aug. 2025, doi: 10.1016/J.COSE.2025.104492.

[43] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How To Backdoor Federated Learning," *Proc. Mach. Learn. Res.*, vol. 108, pp. 2938–2948, Jul. 2018, Accessed: Oct. 20, 2026. [Online]. Available: https://arxiv.org/pdf/1807.00459

[44] P. Blanchard, E. Mahdi El Mhamdi, R. Guerraoui, and J. Stainer, "Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[45] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping," *28th Annu. Netw. Distrib. Syst. Secur. Symp. NDSS 2021*, Dec. 2020, doi: 10.14722/ndss.2021.24434.

[46] Y.-R. Yang, C.-W. Shi, and W.-J. Li, "On the Tension between Byzantine Robustness and No-Attack Accuracy in Distributed Learning," Oct. 06, 2025, *PMLR*. Accessed: Oct. 20, 2026. [Online]. Available: https://proceedings.mlr.press/v267/yang25aa.html

[47] C. Fung, C. J. M. Yoon, and I. Beschastnikh, *The Limitations of Federated Learning in Sybil Settings*. 2020.

[48] Muhammad Ahmad Bilal, Ihtesham Ul Islam, Sarmad Idrees, "Dataset-centric evaluation of federated intrusion detection models in IoT networks," *Sci. Rep.*, vol. 16, 2026, [Online]. Available: http://nature.com/articles/s41598-025-32567-w

[49] M. H. Bhavsar, Y. B. Bekele, K. Roy, J. C. Kelly, and D. Limbrick, "FL-IDS:

Federated Learning-Based Intrusion Detection System Using Edge Devices for Transportation IoT," *IEEE Access*, vol. 12, pp. 52215–52226, 2024, doi: 10.1109/ACCESS.2024.3386631.

Supplementary Material S1. Database-Specific Search Strategies:

Search strategy:

The search strategy aimed to provide a foundation for identifying studies that focus on federated learning, intrusion detection or anomaly detection, and Internet of Things (IoT) and related network environments, and privacy or security. The conceptual strategy comprised four groups of terms: (1) federated and collaborative learning; (2) intrusion detection, anomaly detection, and network security; (3) IoT and related environments; and (4) privacy, information leakage, inference, cryptographic protection, and robustness against malicious participants.

Scopus:

Search fields: Title, abstract, and keywords.

("federated learning" OR "federated machine learning" OR "federated deep learning" OR "federated averaging" OR "collaborative learning" OR "distributed learning" OR "privacy-preserving federated learning") AND ("intrusion detection" OR "intrusion detection system" OR "anomaly detection" OR "network attack detection" OR "cyber attack detection" OR "malicious traffic detection" OR "network security") AND ("Internet of Things" OR IoT OR IIoT OR "industrial Internet of Things" OR "cyber-physical system*" OR CPS OR "edge computing" OR "edge network" OR "smart network" OR "smart home" OR "vehicular network" OR "connected vehicle") AND (privacy OR "data confidentiality" OR "information leakage" OR "gradient leakage" OR "model inversion" OR "membership inference" OR "differential privacy" OR "secure aggregation" OR encryption OR "homomorphic encryption" OR "privacy-preserving" OR poisoning OR Byzantine OR robustness OR "model leakage")

Web of Science Core Collection:

Search fields: Topic (title, abstract, author keywords, and Keywords Plus, as supported by the database)

("federated learning" OR "federated machine learning" OR "federated deep learning" OR "federated averaging" OR "collaborative learning" OR "distributed learning" OR "privacy-preserving federated learning") AND ("intrusion detection" OR "intrusion detection system" OR "anomaly detection" OR "network attack detection" OR "cyber attack detection" OR "malicious traffic detection" OR "network security") AND ("Internet of Things" OR IoT OR IIoT OR "industrial Internet of Things" OR "cyber-physical system*" OR CPS OR "edge computing" OR "edge network" OR "smart network" OR "smart home" OR "vehicular network" OR "connected vehicle") AND (privacy OR "data confidentiality" OR "information leakage" OR "gradient leakage" OR "model inversion" OR "membership inference" OR "differential privacy" OR "secure aggregation" OR encryption OR "homomorphic encryption" OR "privacy-preserving" OR poisoning OR Byzantine OR robustness OR "model leakage")

IEEE Xplore:

Search fields: All Metadata, subject to the fields and Boolean operators supported by the Advanced Search interface.

("federated learning" OR "federated machine learning" OR "federated deep learning" OR "federated averaging" OR "collaborative learning" OR "distributed learning" OR "privacy-preserving federated learning") AND ("intrusion detection" OR "intrusion detection system" OR "anomaly detection" OR "network attack detection" OR "cyber attack detection" OR "malicious traffic detection" OR "network security") AND ("Internet of Things" OR IoT OR IIoT OR "industrial Internet of Things" OR "cyber-physical systems" OR CPS OR "edge

computing" OR "edge networks" OR "smart networks" OR "smart homes" OR "vehicular networks" OR "connected vehicles") AND (privacy OR "data confidentiality" OR "information leakage" OR "gradient leakage" OR "model inversion" OR "membership inference" OR "differential privacy" OR "secure aggregation" OR encryption OR "homomorphic encryption" OR "privacy-preserving" OR poisoning OR Byzantine OR robustness OR "model leakage")

ACM Digital Library:

Search fields: Title, abstract, and author keywords, where supported by the selected ACM Digital Library search interface.

("federated learning" OR "federated machine learning" OR "federated deep learning" OR "federated averaging" OR "collaborative learning" OR "distributed learning" OR "privacy-preserving federated learning") AND ("intrusion detection" OR "intrusion detection system" OR "anomaly detection" OR "network attack detection" OR "cyber attack detection" OR "malicious traffic detection" OR "network security") AND ("Internet of Things" OR IoT OR IIoT OR "industrial Internet of Things" OR "cyber-physical systems" OR CPS OR "edge computing" OR "edge networks" OR "smart networks" OR "smart homes" OR "vehicular networks" OR "connected vehicles") AND (privacy OR "data confidentiality" OR "information leakage" OR "gradient leakage" OR "model inversion" OR "membership inference" OR "differential privacy" OR "secure aggregation" OR encryption OR "homomorphic encryption" OR "privacy-preserving" OR poisoning OR Byzantine OR robustness OR "model leakage")

ScienceDirect:

Search fields: Title, abstract, and keywords, using the platform's supported advanced-search syntax.

("federated learning" OR "federated machine learning" OR "federated deep learning" OR "federated averaging" OR "collaborative learning" OR "distributed learning" OR "privacy-preserving federated learning") AND ("intrusion detection" OR "intrusion detection system" OR "anomaly detection" OR "network attack detection" OR "cyber attack detection" OR "malicious traffic detection" OR "network security") AND ("Internet of Things" OR IoT OR IIoT OR "industrial Internet of Things" OR "cyber-physical systems" OR CPS OR "edge computing" OR "edge networks" OR "smart networks" OR "smart homes" OR "vehicular networks" OR "connected vehicles") AND (privacy OR "data confidentiality" OR "information leakage" OR "gradient leakage" OR "model inversion" OR "membership inference" OR "differential privacy" OR "secure aggregation" OR encryption OR "homomorphic encryption" OR "privacy-preserving" OR poisoning OR Byzantine OR robustness OR "model leakage")

SpringerLink:

Search fields: Title, abstract, and keywords where supported; otherwise, the platform's available search scope.

("federated learning" OR "federated machine learning" OR "federated deep learning" OR "federated averaging" OR "collaborative learning" OR "distributed learning" OR "privacy-preserving federated learning") AND ("intrusion detection" OR "intrusion detection system" OR "anomaly detection" OR "network attack detection" OR "cyber attack detection" OR "malicious traffic detection" OR "network security") AND ("Internet of Things" OR IoT OR IIoT OR "industrial Internet of Things" OR "cyber-physical systems" OR CPS OR "edge computing" OR "edge networks" OR "smart networks" OR "smart homes" OR "vehicular networks" OR "connected vehicles") AND (privacy OR "data confidentiality" OR "information leakage" OR "gradient leakage" OR "model inversion" OR "membership inference" OR "differential privacy" OR "secure aggregation" OR encryption OR

"homomorphic encryption" OR "privacy-preserving" OR poisoning OR Byzantine OR robustness OR "model leakage")

Supplementary Citation Searching:

Google Scholar was used for backward and forward citation searching and to help locate relevant records. Citation searching supplemented the database searches rather than serving as a seventh database-specific query equivalent. The final report should document the seed papers or reference lists examined, the citation-search procedure, the date(s) of searching, and the eligibility criteria applied to additional records.



Copyright © by authors and 50Sea. This work is licensed under Creative Commons Attribution 4.0 International License.