

Privacy-Preserving Network Intrusion Detection Using Federated Learning with Explainable AI and an Autonomous SOC Agent Pipeline

Dr. Shashikumar D R

Computer Science and Engineering
Sai Vidya Institute of Technology
Bengaluru, India

Sahana G C

Computer Science and Engineering
Sai Vidya Institute of Technology
Bengaluru, India

K N Pallavi

Computer Science and Engineering
Sai Vidya Institute of Technology
Bengaluru, India

Mahalakshmi S

Computer Science and Engineering
Sai Vidya Institute of Technology
Bengaluru, India

Kavyanjali B R

Computer Science and Engineering
Sai Vidya Institute of Technology
Bengaluru, India

Pranathi M

Computer Science and Engineering
Sai Vidya Institute of Technology
Bengaluru, India

Abstract—Centralized intrusion detection systems (IDS) require pooling raw network traffic at a single location, which is a significant privacy liability in domains such as banking, healthcare, and telecommunications, and typically rely on static detection thresholds that produce high false-positive rates as traffic patterns drift over time. This paper presents a two-phase system that closes both gaps. Phase-I is a Federated Learning (FL) based IDS in which five simulated clients, each holding a non-identically-distributed partition of the CICIDS2017 dataset, train a per-client LSTM Autoencoder locally and share only model weights — never raw traffic — with a custom FedAvg aggregator implemented over Python threading. Differential Privacy (DP) noise is added to gradients before aggregation to bound information leakage from any single client's update, and an adaptive thresholding mechanism recalibrates the anomaly-detection boundary using a rolling window of reconstruction errors to address the false-positive gap. Per-prediction explanations are produced with SHAP and LIME. Phase-II extends detection into an autonomous, auditable Security Operations Centre (SOC) agent that consumes alerts from the Phase-I detector and executes a six-stage pipeline — Detect, Retrieve, Score, Draft, Verify, Decide — retrieving supporting context from a MITRE ATT&CK knowledge base through TF-IDF based retrieval-augmented generation, computing a rule-based severity score, drafting an incident report, passing every report sentence through a hallucination guardrail before any action is taken, and finally auto-closing, queuing, or escalating the incident. The system achieves a binary classification F1-score of 99.42% (precision 98.94%, recall 99.92%) and a multi-class accuracy of 91.57%, with only a 3.73% accuracy gap between the differentially private and non-private centralized baselines. We describe the system's architecture, workflow, evaluation methodology, and a live walkthrough, and discuss the engineering challenges — including severe class imbalance and framework incompatibility — encountered while building it.

Index Terms—Federated learning, intrusion detection, differential privacy, explainable AI, SHAP, LIME, retrieval-augmented generation, MITRE ATT&CK, autonomous SOC agent, CI-

CIDS2017.

I. INTRODUCTION

A. Motivation

Network intrusion detection sits at the intersection of two conflicting pressures. The first is scale and sensitivity: organisations in banking, healthcare, and telecom generate high-volume traffic that must be inspected continuously, yet that same traffic routinely carries information the organisation cannot legally or contractually centralise. The second is drift: attack patterns, user behaviour, and network load change across the day and across deployments, so a detector tuned once and left alone accumulates false positives until analysts stop trusting it. Our review of the literature shows that most existing IDS work resolves only one of these two pressures at a time: either it centralises data to get a strong model, or it preserves privacy but keeps the detection threshold static [1]–[7].

This paper addresses both pressures together with a Federated Learning (FL) based detector that never centralises raw traffic, combined with an adaptive thresholding mechanism that recalibrates itself against recent traffic. It then goes one step further: rather than stopping at a raw anomaly flag, the system extends detection into an autonomous, explainable, and auditable Security Operations Centre (SOC) agent that triages, contextualises, and drafts a response to every alert, escalating only what genuinely needs a human analyst.

B. Problem Statement

Four recurring weaknesses in current practice motivate this work:

- **Centralized training is a privacy liability.** Pooling raw traffic from multiple network segments or organisations at one location for model training creates a single high-value target and can itself violate data-protection obligations.
- **Static thresholds age badly.** A detection boundary fixed at deployment time does not track legitimate shifts in traffic volume or composition, so false-positive rates climb over time and analysts begin to ignore alerts.
- **Detections rarely explain themselves.** A bare anomaly score gives an analyst little to act on; without a feature-level explanation, triage still depends on manual log inspection.
- **Detection and response are disconnected.** Even a well-tuned detector still hands off to a manual SOC workflow (reading the alert, looking up the relevant technique, judging severity, and deciding what to do) that does not scale with alert volume.

To address these weaknesses, we propose a Federated Learning-based IDS that keeps raw traffic on each client, applies Differential Privacy (DP) to limit information leakage from client updates, adds SHAP/LIME explainability for analyst trust, and uses adaptive thresholding to reduce false positives. We then extend the pipeline into an autonomous SOC agent that retrieves relevant threat context, scores severity, drafts an incident report, and verifies that report before any automated action is taken.

C. Contributions

This paper makes the following contributions:

- 1) A federated LSTM-Autoencoder intrusion detector trained across five non-IID CICIDS2017 clients without centralising raw traffic, achieving a binary F1-score of 99.42% and a multi-class accuracy of 91.57%.
- 2) Gaussian differential-privacy noise applied to client updates, with the privacy–utility trade-off evaluated against a non-private centralized baseline, showing a 3.73% difference in the reported accuracy metric.
- 3) An adaptive thresholding mechanism that recalibrates the anomaly boundary from a rolling window of reconstruction errors.
- 4) Per-prediction SHAP and LIME explanations that give analysts a feature-level rationale for each detection.
- 5) A six-stage SOC agent that retrieves MITRE ATT&CK context, computes a rule-based severity score, drafts an incident report, verifies every sentence against retrieved evidence, and then auto-closes, queues, or escalates the incident, with a traceable record of each step.

II. RELATED WORK

Recent literature on AI-assisted network defence clusters into four themes: federated learning for intrusion detection, differential privacy in federated settings, explainable AI for IDS, and retrieval-augmented generation (RAG) for cyber-threat intelligence and incident response. This section reviews representative work in each theme and identifies the gap this project closes.

A. Federated Learning for Intrusion Detection

Hernandez-Ramos et al. [1] systematically review FL-based intrusion detection and report that most proposals validate on IID partitions and stop at model accuracy, without addressing the response side of the workflow. Belarbi et al. [2] apply federated deep learning to IoT intrusion detection and show competitive accuracy against centralized baselines, but do not evaluate an adaptive or drifting threshold. Vyas et al. [3] survey privacy-preserving FL for IoT IDS and note that differential-privacy guarantees and detection accuracy are usually reported independently rather than as a measured trade-off. Wang et al. [4] propose a semi-supervised FL method for IoT intrusion detection that reduces the labelled-data requirement, and Ocampo et al. [5] extend privacy-preserving FL intrusion detection toward 6G network slices; both target the detection model itself rather than what happens once an alert fires. Olanrewaju-George and Pranggono [7] compare unsupervised and supervised deep FL-IDS models for IoT and confirm that unsupervised (autoencoder-style) formulations generalise well to unseen attack types, which is the detection paradigm adopted here.

B. Differential Privacy in Federated Settings

Saidi and Khouri [6] propose a federated intrusion-detection framework that adds differential privacy to client updates and report the resulting accuracy trade-off, providing the closest precedent to the DP mechanism used in this project. Their work, like most in this space, evaluates the DP/accuracy trade-off on a single detection task and does not connect the private model to a downstream response pipeline — a gap this project closes by carrying the DP-trained model through to an auditable SOC agent.

C. Explainable AI for Intrusion Detection

Gaspar et al. [8] evaluate LIME and SHAP on a multi-layer perceptron IDS and find both useful for analyst-facing explanations but computationally non-trivial at scale. Arreche et al.'s E-XAI [9] benchmarks several black-box explainability frameworks specifically for network intrusion detection and finds no single method dominates across metrics, motivating the combined SHAP-and-LIME approach used here. Wali et al. [10] pair Random-Forest based detection with explainability for a reliable IDS, and Fatema et al.'s FedXAI-IDS [11] is the closest related system, combining federated learning with SHAP-based explanation for IoT IDS; it stops at explanation, however, and does not attach a severity-scoring or incident-response stage to what SHAP surfaces.

D. RAG and LLM-Based Threat Intelligence

Rajapaksha et al. [12] build a RAG-based question-answering system for cyber-attack investigation, and Alhuzali [13] similarly applies retrieval-augmented generation to threat-intelligence report generation; both demonstrate that grounding an LLM's output in a retrieved knowledge base improves factual reliability over an ungrounded prompt, which is the principle this project's TF-IDF retrieval stage relies on. Blefari

et al.'s CyberRAG [14] proposes an agentic RAG pipeline for attack classification and reporting that is architecturally close to this project's SOC agent, but does not report a hallucination-verification step gating automated action. Loumachi et al. [15] apply RAG and LLMs to incident-timeline reconstruction, and Kurniawan et al.'s CyKG-RAG [16] combines a knowledge graph with RAG for cybersecurity question answering; both confirm that structured threat knowledge (such as MITRE ATT&CK, used here) measurably improves retrieval quality over free-text corpora alone.

E. Synthesis

Read together, the surveyed work demonstrates federated training, differential privacy, explainability, and RAG-grounded threat intelligence as four separately validated capabilities, but no reviewed system combines all four into one pipeline that (i) trains without centralising raw traffic, (ii) bounds privacy leakage with DP, (iii) explains every detection, (iv) adaptively recalibrates its own threshold, and (v) hands off to an autonomous, hallucination-checked incident-response agent rather than a bare alert. Table I compares this work against the four study clusters across a broader set of capability terminologies; closing this integration gap is the specific contribution of the architecture described in Section IV.

III. LIMITATIONS OF EXISTING TOOLS

A. Centralized IDS Pipelines

Conventional IDS deployments — whether signature-based or deep-learning based — require raw traffic (or heavily derived features) to be pooled at a central training location. This is workable inside a single organisation but breaks down whenever the goal is to pool detection knowledge across tenants, branches, or organisations that cannot share raw data, which is precisely the setting banking, healthcare, and telecom operators are in.

B. Static-Threshold Anomaly Detectors

Autoencoder- and statistics-based anomaly detectors are usually deployed with a fixed reconstruction-error threshold chosen at validation time. As traffic composition drifts, that fixed boundary either misses genuine anomalies or floods analysts with false positives; neither failure mode self-corrects without manual retuning.

C. Standalone Explainability Tools

SHAP and LIME are typically applied as a post-hoc, offline analysis step run by a data scientist rather than surfaced per-alert to an on-duty analyst, and neither tool by itself decides what action an explained alert warrants.

D. Manual SOC Triage and SIEM Correlation Rules

Security Information and Event Management (SIEM) platforms correlate and surface alerts but leave root-cause lookup, technique attribution, severity judgement, and the close/queue/escalate decision to a human analyst for every event, which does not scale with alert volume and is where most of the delay between detection and response accumulates.

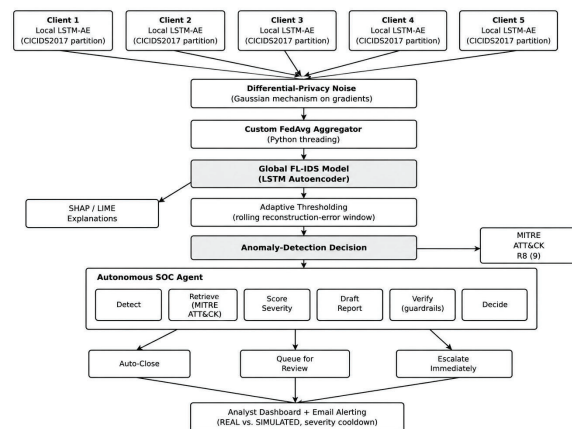


Fig. 1. System architecture: federated LSTM-Autoencoder training with differential privacy and adaptive thresholding, handing off to the six-stage autonomous SOC agent.

IV. PROPOSED SYSTEM ARCHITECTURE

The system is organised into two cooperating phases around a shared detection artefact — the global FL-IDS model — as summarised in Fig. 1: five clients train locally, their updates are DP-noised and combined by a custom FedAvg aggregator, the resulting global model feeds an adaptively-thresholded anomaly decision, and every flagged anomaly is handed to the autonomous SOC agent, whose outcome drives the analyst dashboard and alerting.

A. Client Setup and Data Partitioning

Five simulated clients each hold a non-identically-distributed (non-IID) partition of the CICIDS2017 dataset [17], reflecting the reality that different network segments see different traffic mixes. No client's partition, nor any raw record within it, ever leaves that client during training.

B. Federated Learning IDS

Each client trains an LSTM Autoencoder locally; the model learns to reconstruct benign traffic sequences, and reconstruction error becomes the anomaly signal. Only model weights — never raw traffic or gradients over raw records in the clear — are sent to the aggregator. Because the Flower federated-learning framework was found to be incompatible with the Python 3.13 runtime used for this project, a custom FedAvg aggregator was built directly over Python's threading module to simulate concurrent client updates and weighted averaging. Before aggregation, Gaussian differential-privacy noise is added to each client's update, bounding how much any single client's contribution can be inferred from the aggregated model. The aggregated global model then feeds an adaptive thresholding stage that recalibrates the anomaly-decision boundary using a rolling window of recent reconstruction errors, rather than a value fixed at training time, directly addressing the false-positive gap identified in Section I-B. Per-prediction SHAP and LIME attributions are computed

TABLE I
 CAPABILITY COMPARISON AGAINST SURVEYED RELATED WORK

Capability	FL-IDS studies [1]–[5], [7]	DP studies [6]	XAI studies [8]–[11]	RAG/LLM studies [12]–[16]	This work
Federated (non-centralized) training	✓	✓	partial	–	✓
Differential-privacy guarantee	–	✓	–	–	✓
Adaptive / self-recalibrating detection threshold	–	–	–	–	✓
Per-prediction explainability (SHAP/LIME)	partial	–	✓	–	✓
Root-cause / technique attribution	partial	–	partial	✓	✓
Knowledge-grounded retrieval (RAG)	–	–	–	✓	✓
Rule-based, auditable severity scoring	–	–	–	partial	✓
Hallucination / groundedness verification	–	–	–	partial	✓
Autonomous incident-report drafting	–	–	–	✓	✓
Human-in-the-loop escalation path	–	–	–	–	✓
Alert-fatigue mitigation (cooldown logic)	–	–	–	–	✓
Immutable, end-to-end audit trail	–	–	–	partial	✓
Single detect-to-response pipeline	–	–	–	partial	✓

alongside every classification to give analysts a feature-level explanation.

SOC agent consumes alerts produced by the federated detector and executes six stages in sequence:

- 1) **Detect** — the global FL-IDS model flags an anomaly and passes its record and confidence to the agent.
- 2) **Retrieve** — a TF-IDF based retrieval-augmented generation (RAG) step searches a MITRE ATT&CK knowledge base of nine techniques [18], preferring an exact label match and falling back to semantic-similarity retrieval for inputs that do not match a known label exactly.
- 3) **Score severity** — a rule-based formula (Section VI) combines the retrieved technique’s severity, the detector’s confidence, and traffic context into a single severity score, deliberately avoiding an LLM-based judgement at this step.
- 4) **Draft report** — an LLM/template stage drafts a human-readable incident report grounded in the retrieved technique and the detection evidence.
- 5) **Verify** — a hallucination guardrail checks every sentence of the drafted report against the source evidence before the report is allowed to trigger any action.
- 6) **Decide** — the incident is auto-closed, queued for review, or escalated immediately, depending on the severity score and the outcome of verification.

C. Alerting Subsystem

A severity-based cooldown governs outbound email alerts so that repeated notifications for the same ongoing attack do not cause alert fatigue, while high-severity events still reach the analyst promptly rather than being suppressed by the cooldown.

D. Analyst Dashboard

A Streamlit dashboard presents live pipeline activity and explicitly distinguishes a REAL alert sourced from the Phase-I SQLite attack log from a SIMULATED (fallback) alert, so that evaluators and analysts can tell genuine detections from demonstration data at a glance.

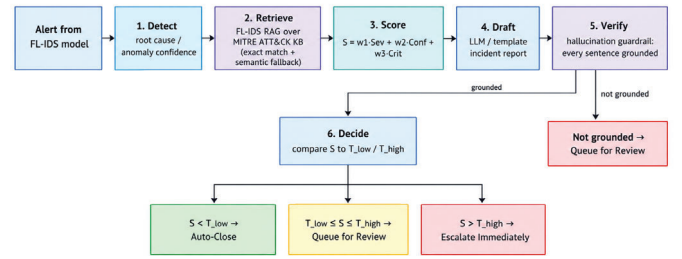


Fig. 2. Methodology: per-alert SOC agent workflow, from detection through MITRE ATT&CK-grounded retrieval, rule-based severity scoring, report drafting, hallucination verification, and the final auto-close / queue / escalate decision.

V. SYSTEM WORKFLOW

The end-to-end pipeline runs in two linked stages, illustrated in Algorithm 1: (1) the federated training loop produces and continually refines the global FL-IDS model; (2) every anomaly the global model flags is handed to the six-stage SOC agent, which auto-closes the incident, queues it for analyst review, or escalates it immediately.

Fig. 2 illustrates this same workflow for a single alert as it moves through the SOC agent’s six stages and is routed to one of the three terminal outcomes.

VI. ANOMALY DETECTION AND SEVERITY SCORING MODEL

Anomaly detection rests on the reconstruction error of the LSTM Autoencoder: a record whose reconstruction error e exceeds the adaptive threshold τ is flagged. Rather than a value fixed at training time, τ is recomputed from a rolling window of recent reconstruction errors, so the boundary tracks legitimate drift in traffic volume and composition instead of aging into either excess false positives or missed detections.

Once a record is flagged, the SOC agent computes a severity score S that determines routing:

$$S = w_1 \cdot T_{sev} + w_2 \cdot Conf + w_3 \cdot Crit \quad (1)$$

Algorithm 1 Federated detection and autonomous response

Require: client partitions $D_1, \dots, D_5$, technique KB K

```

1: for each round  $r$  do
2:   for each client  $i$  in parallel do
3:      $\theta_i \leftarrow \text{TRAINLOCAL}(D_i)$  {LSTM Autoencoder}
4:      $\theta_i \leftarrow \theta_i + \text{DPNOISE}(\theta_i)$  {Gaussian mechanism}
5:   end for
6:    $\theta_{\text{global}} \leftarrow \text{FEDAVG}(\theta_1, \dots, \theta_5)$  {custom threading aggregator}
7:    $\tau \leftarrow \text{ADAPTIVETHRESHOLD}(\text{recent reconstruction errors})$ 
8: end for
9: for each incoming record  $x$  do
10:   $e \leftarrow \text{RECONERROR}(\theta_{\text{global}}, x)$ 
11:  if  $e > \tau$  then
12:     $d \leftarrow \text{DETECT}(x, e)$  {Agent stage 1}
13:     $c \leftarrow \text{RETRIEVE}(d, K)$  {Agent stage 2, TF-IDF RAG}
14:     $s \leftarrow \text{SCORESEVERITY}(d, c)$  {Agent stage 3}
15:     $rpt \leftarrow \text{DRAFTREPORT}(d, c, s)$  {Agent stage 4}
16:    if  $\text{VERIFY}(rpt) = \text{false}$  then
17:      {Agent stage 5}
18:       $\text{QUEUEFORREVIEW}(rpt)$ 
19:    else if  $s > T_{\text{high}}$  then
20:       $\text{ESCALATE}(rpt)$  {Agent stage 6}
21:    else if  $s < T_{\text{low}}$  then
22:       $\text{AUTOCLOSE}(rpt)$ 
23:    else
24:       $\text{QUEUEFORREVIEW}(rpt)$ 
25:    end if
26:  end if
27: end for
    
```

where T_{sev} is the severity associated with the retrieved MITRE ATT&CK technique, Conf is the detector's confidence (derived from the reconstruction error margin above τ), and Crit is the criticality of the affected asset or traffic segment; w_1, w_2, w_3 are configurable weights that sum to 1. The score is deliberately computed by a rule-based formula rather than an LLM, keeping the routing decision auditable and reproducible rather than exposed to a hallucinated judgement — the same design principle noted for the risk-routing decision in comparable multi-agent remediation systems. An incident is auto-closed only when S falls below a low threshold τ_{low} and the hallucination guardrail confirms every sentence of the drafted report is grounded in retrieved evidence; it is escalated immediately whenever S exceeds a high threshold τ_{high} , regardless of how the report reads, so that a single strong severity signal cannot be diluted by a well-written but lower-confidence report.

VII. ILLUSTRATIVE WALKTHROUGH

A live demonstration run clarifies how the stages connect. A genuine alert already present in the SQLite attack log is picked up by the agent's Detect stage. The Retrieve stage matches the alert's label against the MITRE ATT&CK knowledge base and

TABLE II
TECHNOLOGY STACK BY PIPELINE COMPONENT

Component	Technology
Detection model	PyTorch LSTM Autoencoder, scikit-learn
Federated aggregation	Custom FedAvg over Python threading (Python 3.13)
Differential privacy	Gaussian noise mechanism on client gradients
Explainability	SHAP, LIME
Knowledge retrieval	TF-IDF RAG over MITRE ATT&CK KB (9 techniques)
Agent services / API	FastAPI
Alert store	SQLite
Analyst dashboard	Streamlit
Alerting	Email, severity-based cooldown

returns the corresponding technique. The Score stage combines the technique's severity with the detector's confidence to compute S . The Draft stage produces an incident report describing the detected behaviour, the matched technique, and the recommended response. The Verify stage checks every sentence of that report against the retrieved evidence and the original alert record and confirms all are grounded. Because the computed severity is high, the final Decide stage routes the incident to ESCALATE IMMEDIATELY, and the dashboard marks the alert as a REAL DB alert rather than a simulated one.

VIII. IMPLEMENTATION TECHNOLOGY STACK

The prototype targets a Python-centric stack, reflecting the languages and frameworks most compatible with the LSTM/FedAvg training pipeline and the SOC-agent services. The detection model is implemented in PyTorch [19] and the analyst dashboard in Streamlit [20]. Table II summarises the mapping between each pipeline component and the technology used to implement it.

A key implementation challenge was severe class imbalance in CICIDS2017: roughly 82% of records are BENIGN, which initially collapsed the model to predicting only the majority class (approximately 37% raw accuracy). This was resolved through systematic rebalancing and loss adjustment during training rather than by discarding minority-class records, so that rare attack types remain represented in the trained model.

IX. TESTING AND VALIDATION

The global model was evaluated on an 80/20 stratified train-test split spanning all 15 CICIDS2017 attack classes. Binary classification (benign vs. attack) achieved an F1-score of 99.42%, precision of 98.94%, and recall of 99.92%; multi-class classification across all 15 labels achieved 91.57% accuracy. To validate the cost of privacy, the differentially private model was benchmarked against a non-private centralized baseline and showed only a 3.73% accuracy gap, and both were further benchmarked against a centralized Random Forest baseline and a local-only (non-federated, single-client) baseline to confirm that federated aggregation recovers most of the accuracy a fully centralized approach would achieve.

The RAG retrieval stage was validated with a dedicated smoke test: every genuine CICIDS2017 label resolved to its

TABLE III
VALIDATION METRICS SUMMARY

Metric	Result
Binary F1-score	99.42%
Binary precision	98.94%
Binary recall	99.92%
Multi-class accuracy (15 classes)	91.57%
DP vs. non-private accuracy gap	3.73%
RAG exact-match retrieval score	1.0

matching MITRE ATT&CK technique via an exact match (similarity score 1.0), while inputs with no exact label correctly fell back to weaker semantic-similarity matching rather than failing silently. The hallucination guardrail was validated end-to-end by confirming that every sentence of a drafted incident report is checked as grounded in source evidence before any auto-action — close, queue, or escalate — is permitted.

X. SECURITY CONSIDERATIONS

Because the agent is authorised to draft reports and, for low-severity incidents, auto-close them without a human in the loop, its own safeguards matter as much as the vulnerabilities it is designed to catch. Three safeguards are built into the design. First, no raw client traffic ever leaves the client that generated it — only DP-noised model weights are shared with the aggregator, following the least-exposure principle established in Phase-I. Second, the hallucination guardrail gates every automated action: a drafted report that cannot be verified as grounded in retrieved evidence is queued for human review rather than acted on. Third, the severity-based email cooldown is itself a safeguard against both under- and over-alerting — it prevents alert fatigue from repeated notifications for the same ongoing attack while still guaranteeing that high-severity events reach the analyst promptly rather than being silently suppressed.

XI. DATA HANDLING CONSIDERATIONS

Network traffic and derived features routinely contain information an organisation cannot let leave its own infrastructure. Because the federated design keeps each client's raw partition local by construction, only model weights — with DP noise already applied — ever cross a client boundary. The alert store (SQLite) and the incident reports drafted from it are treated as sensitive by default, since a decision log doubles as a record of exactly where a network's weaknesses were found; the dashboard's REAL/SIMULATED distinction exists partly so that demonstration traffic is never mistaken for, or mixed into, that sensitive record.

XII. DISCUSSION AND LIMITATIONS

Three limitations of the current design are worth stating plainly. First, the five clients are simulated partitions of a single public dataset rather than genuinely distributed deployments across separate organisations; the privacy mechanism is architecturally sound, but its operational behaviour under real

cross-organisation network conditions (latency, client dropout, adversarial clients) remains to be validated. Second, severe class imbalance in CICIDS2017 required deliberate rebalancing during training, and rare attack classes may still be under-represented relative to common ones despite this correction. Third, the MITRE ATT&CK knowledge base currently indexes only nine techniques; retrieval correctly falls back to semantic-similarity matching for inputs outside that set, but coverage for genuinely novel attack techniques is inherently bounded by the size of the indexed knowledge base until it is expanded.

A design commitment worth stating explicitly alongside these limitations: the severity-scoring step is deliberately rule-based rather than LLM-based, and the hallucination guardrail sits between every drafted report and any automated action. This is a trade-off against maximum automation, made because the cost of an incorrect automated decision in a security-incident context is disproportionate to the time an unchecked LLM judgement would save.

XIII. SCOPE AND FUTURE WORK

The current system covers federated LSTM-Autoencoder training with differential privacy and adaptive thresholding, SHAP/LIME explainability, and the six-stage autonomous SOC agent operating over a nine-technique MITRE ATT&CK knowledge base. The current pipeline uses a template-based drafting stage; a live LLM for report drafting is left as future work. The system also does not yet cover deployment across genuinely separate organisational networks. Planned extensions include integrating a live LLM for richer report generation while maintaining the same hallucination guardrail, and expanding the MITRE ATT&CK knowledge base beyond nine techniques.

XIV. CONCLUSION

This paper presented a two-phase, privacy-preserving intrusion detection system that trains a federated LSTM Autoencoder across five non-IID clients with differential-privacy guarantees and an adaptively recalibrated detection threshold, achieving a 99.42% binary F1-score and 91.57% multi-class accuracy without centralising raw traffic. It then extended detection into an autonomous, six-stage SOC agent that retrieves MITRE ATT&CK context through TF-IDF RAG, computes an auditable rule-based severity score, drafts an incident report, verifies every sentence of that report against retrieved evidence before permitting any automated action, and finally auto-closes, queues, or escalates the incident accordingly. Real engineering obstacles — severe class imbalance and a framework/runtime incompatibility that ruled out an off-the-shelf federated-learning library — were solved through systematic debugging rather than workarounds. Future work centres on integrating a live LLM for report drafting once available and broadening the technique knowledge base, ahead of submitting the associated research paper to MDPI Electronics.

ACKNOWLEDGMENT

The authors thank Dr. Shashikumar D R, Professor and Head, and Sahana G C, Assistant Professor, Department of

Computer Science and Engineering, Sai Vidya Institute of Technology, Bengaluru, for their guidance throughout the design and evaluation of this system.

REFERENCES

- [1] J. L. Hernandez-Ramos et al., "Intrusion Detection based on Federated Learning: A Systematic Review," *ACM Computing Surveys*, vol. 57, no. 12, pp. 1–65, 2025.
- [2] O. Belarbi et al., "Federated Deep Learning for Intrusion Detection in IoT Networks," in *Proc. IEEE Global Communications Conf. (GLOBE-COM)*, 2023.
- [3] A. Vyas et al., "Privacy-Preserving Federated Learning for Intrusion Detection in IoT Environments: A Survey," *IEEE Access*, 2024.
- [4] Y. Wang, Z. Xue, T. Ohtsuki, B. Adebisi, and G. Gui, "Semisupervised Federated-Learning-Based Intrusion Detection Method for Internet of Things," *IEEE Internet of Things Journal*, vol. 10, no. 10, pp. 8645–8657, 2023.
- [5] J. A. Ocampo, L. Islami, and M. Alishahi, "Privacy-Preserving Federated Learning for IoT Intrusion Detection in 6G Networks," in *Proc. 18th Int. Symp. Foundations and Practice of Security (FPS)*, Springer, 2026, pp. 39–55.
- [6] A. Saidi and A. O. Khouri, "Enhancing Data Privacy in Intrusion Detection: A Federated Learning Framework With Differential Privacy," *Concurrency and Computation: Practice and Experience*, vol. 37, e70186, 2025.
- [7] B. Olanrewaju-George and B. Pranggono, "Federated Learning-Based Intrusion Detection System for the Internet of Things Using Unsupervised and Supervised Deep Learning Models," *Computers & Security: Applications*, vol. 3, p. 100068, 2025.
- [8] D. Gaspar, P. Silva, and C. Silva, "Explainable AI for Intrusion Detection Systems: LIME and SHAP Applicability on Multi-Layer Perceptron," *IEEE Access*, vol. 12, pp. 30164–30175, 2024.
- [9] O. Arreche, T. R. Guntur, J. W. Roberts, and M. Abdallah, "E-XAI: Evaluating Black-Box Explainable AI Frameworks for Network Intrusion Detection," *IEEE Access*, vol. 12, pp. 23954–23988, 2024.
- [10] S. Wali, Y. A. Farrukh, and I. Khan, "Explainable AI and Random Forest based Reliable Intrusion Detection System," *Computers & Security*, vol. 157, p. 104542, 2025.
- [11] K. Fatema, S. K. Dey, M. Anannya, R. T. Khan, M. M. Rashid, C. Su, and R. Mazumder, "FedXAI-IDS: A Federated Learning Approach with SHAP-Based Explainable Intrusion Detection for IoT Networks," *Future Internet*, vol. 17, p. 234, 2025.
- [12] S. Rajapaksha, R. Rani, and E. Karafili, "A RAG-Based Question-Answering Solution for Cyber-Attack Investigation and Attribution," in *Proc. European Symp. Research in Computer Security (ESORICS)*, Springer, 2024, pp. 238–256.
- [13] A. Alhuzali, "LLM-Powered Threat Intelligence: A Retrieval-Augmented Generation Approach for Cyber Attack Investigation," *PeerJ Computer Science*, 2025.
- [14] F. Blefari, C. Cosentino, F. A. Pironti, A. Furfaro, and F. Marozzo, "CyberRAG: An Agentic RAG Cyber Attack Classification and Reporting Tool," *arXiv preprint arXiv:2507.02424*, 2025.
- [15] F. Y. Loumachi, M. C. Ghanem, and M. A. Ferrag, "Advancing Cyber Incident Timeline Analysis through Retrieval-Augmented Generation and Large Language Models," *Computers*, vol. 14, no. 67, pp. 1–42, 2025.
- [16] K. Kurniawan, E. Kiesling, and A. Ekelhart, "CyKG-RAG: Towards Knowledge-Graph Enhanced Retrieval Augmented Generation for Cybersecurity," 2024.
- [17] Canadian Institute for Cybersecurity, "CICIDS2017 Dataset," University of New Brunswick, 2017. [Online]. Available: <https://www.unb.ca/cic/datasets/ids-2017.html>
- [18] MITRE Corporation, "MITRE ATT&CK Framework," 2024. [Online]. Available: <https://attack.mitre.org/>
- [19] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Proc. NeurIPS*, 2019. [Online]. Available: <https://pytorch.org/docs/>
- [20] Streamlit Inc., "Streamlit Documentation," 2025. [Online]. Available: <https://docs.streamlit.io/>