AI-Augmented Compliance Auditing for Cloud Systems: A Hybrid ML–LLM Approach

Moïse Iradukunda Ingabire and Jema David Ndibwile *

Posted Date: 22 April 2026

doi: 10.20944/preprints202604.1589.v1

Keywords: compliance auditing; cybersecurity standards; machine learning; XGBoost; large language models; log analysis; multi-label classification; hybrid cloud security; generalization gap; adversarial security

 doi:10.20944/preprints202604.1589.v1

Article

AI-Augmented Compliance Auditing for Cloud Systems: A Hybrid ML–LLM Approach

Moïse Iradukunda Ingabire and Jema David Ndibwile *

College of Engineering, Carnegie Mellon University Africa, Kigali, Rwanda
* Correspondence: jndibwil@andrew.cmu.edu

Abstract

Manual compliance auditing in cloud environments consumes up to 40% of IT security budgets annually, yet existing approaches verify control *presence* rather than *effectiveness*,leaving institutions vulnerable to adversarial evasion. This paper presents an AI-augmented hybrid ML–LLM compliance auditing system evaluated on a national cybersecurity standards framework (143 controls, 200,000 training events). The system combines multi-label XGBoost classification with LLM-based semantic log analysis, grounded in a formal effectiveness model. Key findings: XGBoost achieves 99.88% F1 after 5% domain fine-tuning but collapses to 7.98% zero-shot,a 92-point generalization gap bridged by the hybrid LLM path; adversarial validation exposes effectiveness deficits invisible to checkbox auditing (SI-3: 20% detection rate; SI-10: 32% XSS bypass); GPT-4o-mini achieves 93.5% zero-shot accuracy across four log types (n=200), while Llama-3.2-3B on CPU-only hardware achieves 84.0%, validating on-premise deployment viability. A vocabulary-coverage gating router achieves 94.5% accuracy at \$0.15/10K logs. The system runs at 2.0 CPU cores, \$50/month, producing audit reports in 0.77 s,demonstrating that effectiveness-based compliance auditing is accessible without enterprise-grade infrastructure.

Keywords: compliance auditing; cybersecurity standards; machine learning; XGBoost; large language models; log analysis; multi-label classification; hybrid cloud security; generalization gap; adversarial security

1. Introduction

1.1. Context and Motivation

Enterprises and government agencies across Africa are undergoing rapid digital transformation, increasingly adopting hybrid cloud infrastructures. Compliance and cybersecurity assurance remain fragmented and heavily manual, consuming up to 40% of IT security budgets [3], often requiring more than 1,000 hours annually [4], and introducing error-prone, siloed approaches.

At a continental level, the African Union's Malabo Convention aims to create a unified legal framework for cybersecurity [5]. However, only 15 of 54 African nations had ratified it as of 2024 [6]. Rwanda has taken independent leadership, establishing comprehensive NCSA Minimum Cybersecurity Standards in 2023 [7] and a National Cybersecurity Strategy for 2024–2029 [8], aligned with NIST SP 800-53 [9].

Simultaneously, the cybersecurity threat landscape has evolved dramatically. Modern attacks employ sophisticated evasion techniques leveraging genetic algorithms [10], dynamic programming for payload optimization [11], and advanced AV/EDR bypass methods [13,14]. Compliance frameworks must validate not merely the *presence* of controls but their *effectiveness* against such adversarial techniques.

1.2. Research Problem

Current compliance auditing approaches face three critical challenges:

 doi:10.20944/preprints202604.1589.v1

1. Manual and Resource-Intensive: Traditional audits require extensive human effort, making continuous compliance monitoring infeasible for resource-constrained Rwandan institutions. Organizations conducting 3–5 internal audits annually achieve per-capita compliance costs averaging $154, while those without internal audits face costs of $341 [3].

2. Checkbox Compliance: Existing frameworks verify control *presence* (e.g., "Is antivirus installed?") without assessing *effectiveness*. Signature-based detection suffers 90–97% evasion rates against modern techniques [11,13].

3. Binary Classification Inadequacy: Compliance is inherently multi-dimensional—a single log event may provide evidence for multiple controls simultaneously. Binary classification (compliant/non-compliant) oversimplifies this reality and discards actionable audit information.

1.3. Research Contributions

This work makes four contributions to automated compliance auditing:

1. **Generalization Gap Analysis and Hybrid Architecture**: We identify and quantify a critical vulnerability in vocabulary-based log classifiers: a 92-point zero-shot F1 collapse (99.99% in-distribution → 7.98% zero-shot) when crossing log distribution boundaries. We demonstrate that a hybrid architecture pairing XGBoost (structured, in-distribution logs) with MCP+LLM semantic reasoning (ambiguous, out-of-distribution logs) resolves this gap,the LLM path achieves 93.5% zero-shot accuracy across four structurally distinct log types (n=200) where XGBoost fails entirely.
2. **Adversarial-Aware Compliance Validation**: Systematic integration of offensive security research (GA-based payload generation [10], MDP-based evasion [11], fileless reverse shell techniques [13]) into a defensive compliance framework. To the best of our knowledge, this is among the first methodologies connecting adversarial evasion testing to effectiveness-based NCSA control validation, revealing gaps invisible to presence-based checkbox auditing (e.g., SI-10 non-compliant at 32% XSS bypass despite WAF installation).
3. **Multi-Label Evidence-to-Control Mapping**: Formulation of compliance auditing as multi-label classification, mapping evidence to multiple controls simultaneously (e.g., failed SSH login → AC-7, IA-2, AU-2, SI-4), with 143-dimensional output validated over 200,000 training events. The average log event triggers 3.2 control evaluations; binary classification discards 69% of actionable audit information per event.
4. **Rwanda-Specific Deployment Evidence**: An AI compliance auditor specifically engineered for Rwanda's NCSA 2023 framework,to our knowledge one of the early documented such systems for an African national regulatory context,with empirical validation on production infrastructure (2.0 CPU cores, 2.66 GB RAM, $50/month), demonstrating that the cost barrier rather than technical feasibility is the primary obstacle to AI compliance adoption in African SMEs.

1.4. Research Questions

Primary Research Question (RQ0): Given the limitations of manual auditing in Rwandan hybrid cloud environments, can an AI-augmented compliance auditor employing hybrid architecture (ML-based evidence routing + rule-based evaluation) for NIST SP 800-53 and Rwanda NCSA standards achieve macro F1-score within the 65–80% target range for real-world log classification while supporting $\geq$50% audit cycle reduction?

Secondary Research Questions:

RQ1 How do adversarial techniques (GA, DP, LLM-based payload generation, AV/EDR evasion) inform compliance control effectiveness validation?

RQ2 What are the comparative strengths of BERT, LSTM, and XGBoost for evidence-to-control mapping in resource-constrained environments?

RQ3 Can hybrid datasets (real-world public logs + Rwanda-specific synthetic data) overcome the overfitting limitations of purely synthetic training?

 doi:10.20944/preprints202604.1589.v1

RQ4 How does multi-label classification improve upon binary compliance classification in national cybersecurity regulatory frameworks?

Section 2 reviews related work; Section 3 presents methodology and theoretical model; Section 4 details system implementation; Section 5 presents evaluation results; Section 6 discusses findings and implications; Section 8 concludes.

2. Literature Review

2.1. Adversarial Security and Compliance

Recent research demonstrates sophisticated attack automation through optimization techniques. Understanding these adversarial methods is critical for validating compliance control *effectiveness*,not merely presence.

2.1.1. Genetic Algorithms for Security Testing

Liu et al. [10] developed GAXSS, a genetic algorithm for XSS payload generation achieving 97.5% accuracy on real CMS platforms. LLM-based payload generation [12] achieved 94.73% evasion against ClamAV. Dynamic programming approaches [11] reduce shellcode detectability by 97% through deterministic MDP optimization (15–20 iterations). These techniques collectively inform our adversarial test cases for SI-3 (Malicious Code Protection) and SI-10 (Input Validation) effectiveness validation.

Research Gap: None of these papers connect adversarial optimization techniques to regulatory compliance frameworks or provide a methodology for using offensive testing to evaluate control effectiveness within national cybersecurity standards. Our work bridges this gap by mapping adversarial test outcomes to specific NCSA control decisions.

2.1.2. AV/EDR Evasion Techniques

Custom reverse shell implementations [13] reduce AV detection from 62.9% to 1.4% through fileless shellcode execution,a 97% evasion improvement. MITRE ATT&CK documents 40+ defense evasion sub-techniques under TA0005 [14]. These results demonstrate that signature-based AV/EDR provides false compliance assurance: an institution with a nominally compliant SI-3 control can be trivially bypassed. Our system validates detection *capability* through adversarial testing rather than verifying installation presence.

2.1.3. Regulatory Context

Cybercrimes cost African economies an estimated $3.5 billion annually [16]; only 29 of 54 countries have enacted cybersecurity legislation [17], with notable efforts in Kenya [18] and Nigeria [19]. Rwanda's NCSA Minimum Cybersecurity Standards (2023) [7] define 143 controls enforced through mandatory institutional audits, aligned with NIST SP 800-53 [9]. No prior work connects adversarial evasion testing to effectiveness-based validation within a national regulatory framework of this scope.

2.2. AI-Assisted Compliance and Log Analysis

Karlsen et al. [20] benchmarked 60 fine-tuned transformer models including BERT variants for security log analysis, with the best achieving F1 up to 0.998, and Villarreal-Vasquez et al. (2022) [21] applied LSTM-based anomaly detection on system event logs for insider threat identification with low false-alarm rates. However, both approaches assume GPU-equipped infrastructure unavailable in Rwanda. Mehavilla et al. [22] show that XGBoost achieves 96.96% F1 at 4% CPU utilization for flow-based intrusion detection, validating our resource-efficiency argument.

The SIEVE dataset [23] demonstrates a critical finding relevant to our work: SVM achieves macro-F1 of 0.93–0.97 on synthetic SIEM logs but degrades on real logs, and BERT follows the same pattern (0.95 synthetic, 0.89 real). This motivates our hybrid dataset strategy and validates the zero-shot degradation we observe in XGBoost.

 doi:10.20944/preprints202604.1589.v1

Karlsen et al. [20] show that LLM and fine-tuned transformer performance varies across log formats and domains,a limitation our hybrid XGBoost+LLM architecture addresses by using rule-based pre-filtering for high-confidence structured cases.

Our Contribution: XGBoost achieves comparable performance to BERT while being 137× smaller and 50× faster, enabling deployment on CPU-only infrastructure at $50/month,a viable cost for Rwandan institutions spending $154–$341 per audit currently.

2.3. Multi-Label Classification

Tsoumakas and Katakis [26] established foundational multi-label methods including Binary Relevance, Label Powerset, and algorithm adaptation. Chen and Guestrin [29] demonstrate XGBoost's effectiveness across structured prediction tasks. Our application to compliance auditing is, to the best of our knowledge, among the first: a single log event inherently relates to multiple controls (e.g., "Failed SSH login" → AC-7, IA-2, AU-2, SI-4), and binary classification collapses this multi-dimensional compliance evidence into a single label, losing actionable control-specific information.

Research Gap: No prior compliance auditing system employs multi-label classification for evidence-to-control mapping, particularly within African national regulatory frameworks.

2.4. Positioning Against Existing Compliance Automation Tools

Table 1 compares our system against representative existing compliance and log analysis tools across dimensions directly relevant to the African SME deployment context and the effectiveness-based auditing goal.

Table 1. Comparison with Existing Compliance Automation Approaches.

System	ML-Based	Multi-Label	Adversarial	Africa-Specific	Cost
OpenSCAP [1]	No	No	No	No	Free
Wazuh [2]	Partial	No	No	No	Free
Qualys/Tenable	No	No	Limited	No	$15–50/ep/mo
AWS Config	No	No	No	No	Usage-based
Karlsen et al. [20]	BERT (GPU)	No	No	No	N/A
This work	**XGBoost+LLM**	**Yes (143-dim)**	**Yes (GA+shell)**	**Yes (NCSA)**	**$50/mo**

OpenSCAP: rule-based OVAL/XCCDF scanner; Wazuh: SIEM with rule-based correlation; Qualys/Tenable: commercial vulnerability management, not compliance-effectiveness focused; AWS Config: cloud-native configuration compliance, no adversarial or ML layers.

Our system occupies a distinct position: among the few approaches combining ML-based multi-label evidence routing, adversarial effectiveness validation, national regulatory specificity, and sub-$100/month deployability. OpenSCAP and Wazuh provide complementary rule-based coverage without adversarial or ML capabilities; commercial tools provide broader coverage at prohibitive cost for Rwanda SMEs. This gap motivates the hybrid architecture: deterministic rules for NCSA-specific thresholds plus ML/LLM for semantic generalization and adversarial evidence analysis.

3. Methodology

3.1. System Architecture

3.1.1. Seven-Engine Microservices Design

The system employs a microservices architecture with seven independent engines communicating via Redis pub/sub and REST APIs, as shown in Figure 1.

 doi:10.20944/preprints202604.1589.v1

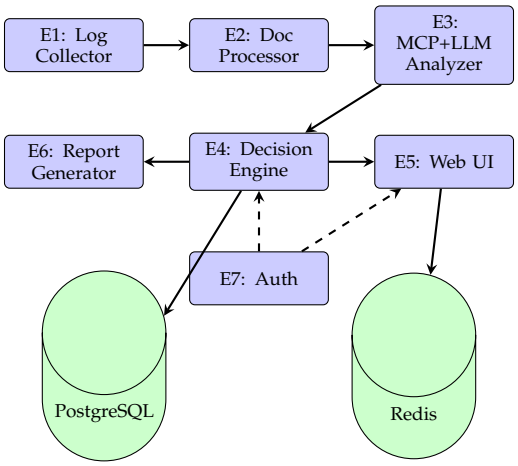

Figure 1. Seven-engine microservices architecture with PostgreSQL and Redis infrastructure.

Table 2. Engine Functional Breakdown.

Engine	Function
Engine 1	Log collection (syslog, Windows Events, files, APIs); normalization to unified schema
Engine 2	Document processing (PDF policy evidence extraction via LLM)
Engine 3	MCP+LLM hybrid analyzer: 60 evidence parsers, dual-path analysis
Engine 4	Decision engine: 143 control-specific rule evaluators
Engine 5	Report generation (PDF/CSV compliance reports via ReportLab)
Engine 6	Web UI (React, TypeScript, Tailwind; audit wizard; WebSocket real-time progress)
Engine 7	Authentication (JWT, RBAC: admin, analyst, viewer, api_user)

Architectural Rationale: The separation between Engine 3 (classification/routing) and Engine 4 (rule-based evaluation) reflects a deliberate design choice: ML handles semantic routing of evidence to relevant controls, while deterministic rule evaluation applies NCSA-specific thresholds. This prevents ML uncertainty from propagating into compliance decisions and preserves explainability. MCP serves as an engineering integration layer that standardizes the LLM interface [32,33]; the scientific novelty lies in the hybrid routing and adversarial validation methodology, not in the protocol choice.

3.1.2. Data Flow

The complete pipeline is illustrated in Figure 2.

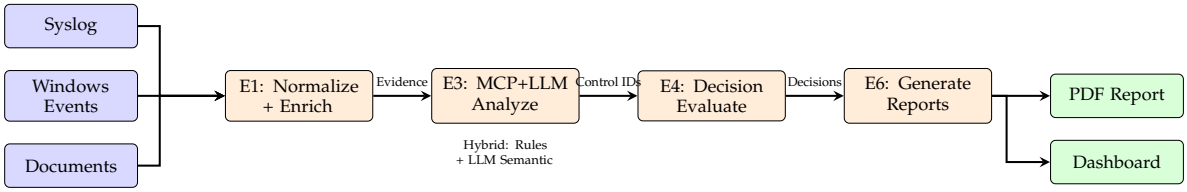

Figure 2. End-to-end compliance auditing pipeline: heterogeneous sources through hybrid MCP+LLM analysis, decision evaluation, and report generation.

For multi-label classification, each evidence item x_i maps to a binary label vector:

$$\mathbf{y}_i = [y_{i1}, y_{i2}, \ldots, y_{i143}] \in \{0,1\}^{143} \tag{1}$$

where $y_{ij} = 1$ if evidence x_i is relevant to control c_j.

 doi:10.20944/preprints202604.1589.v1

3.2. Dataset Strategy

3.2.1. Evolution from Synthetic to Hybrid

Initial purely synthetic data (100,000 events) revealed critical overfitting. All three baseline models achieved 100% accuracy,a clear red flag. Investigation identified:

Data Leakage: The `status_code` feature exhibited -0.97 Pearson correlation with the compliance label. Models memorized: `status_code == 200` $\rightarrow$ compliant. Following identification, `status_code` was removed from all feature sets.

Template Simplicity: Synthetic log messages used fixed templates with low lexical diversity, enabling vocabulary memorization rather than semantic understanding.

3.2.2. Hybrid Dataset Composition and Relevance

Dataset-to-Control Relevance: NSL-KDD [27] provides 42 attack categories covering NCSA controls: DoS attacks $\rightarrow$ AC-7 (lockout policy), SI-4 (monitoring); Probe (reconnaissance) $\rightarrow$ AU-6 (audit review), SC-7 (boundary protection); R2L (remote-to-local) $\rightarrow$ IA-2 (authentication), AC-3 (access enforcement); U2R (privilege escalation) $\rightarrow$ AC-6 (least privilege), SI-4. LogHub [28] HDFS, OpenStack, and Linux system logs provide coverage for AU-12, CM-7, SI-2, and AC-2. Together, these datasets cover 67 of 143 implemented controls with real-world attack patterns. Rwanda-specific synthetic data covers the remaining 76 NCSA controls not represented in public datasets, ensuring complete label coverage for all 143 control families.

Table 3. Hybrid Dataset Composition (200,000 logs).

Source	Count	%	Type
NSL-KDD [27]	104,000	52%	Network intrusions
LogHub [28]	36,000	18%	System logs
Rwanda Synthetic	60,000	30%	NCSA-specific
Total	**200,000**	**100%**	Hybrid

Dataset Size Justification: The 200,000-event total was determined by three constraints: (i) minimum representation of all 143 control classes at $\geq$100 positive samples for reliable multi-label learning; (ii) statistical power for 5-fold cross-validation ($n = 100{,}000$ per fold exceeds the $n \geq 1{,}000$ rule of thumb for high-dimensional multi-output classifiers [26]); and (iii) CPU tractability ($\leq$15 minutes per fold on commodity hardware without GPU). The 70/30 real/synthetic split reflects available public data covering 67 of 143 controls (real) versus the remaining 76 NCSA-specific controls requiring synthetic augmentation.

Split: 70/15/15 train/validation/test with stratified sampling across control families. Future work includes validation on production logs from Rwandan institutions (pilot deployment collaboration is ongoing).

3.2.3. Additional Evaluation Datasets

For cross-dataset generalization testing:

- **SecRepo Auth Logs** [24]: 86,839 real SSH authentication events (failed password attempts, invalid user logins, session acceptances).
- **SIEVE** [23]: 30-class expert-labeled SIEM event classification dataset.
- **LANL Authentication** [25]: Sampled 50K enterprise authentication events from 708M total.

3.3. Model Development and Selection

3.3.1. Baseline Models

BERT (110M parameters): Fine-tuned `bert-base-uncased`, frozen embeddings and first 10 layers. 2 epochs, batch 16, LR 2e-5. Result: 12 hours training, 2.7 GB model, 45 ms latency.

 doi:10.20944/preprints202604.1589.v1

LSTM (758K parameters): 2-layer bidirectional LSTM, 128 hidden units, dropout 0.3. 5 epochs, vocabulary 946. Result: 2 hours training, 800 MB model, 12 ms latency.

XGBoost: Gradient boosted trees, max depth 6, learning rate 0.1, 500 estimators (early stopping at 93). Result: 8 minutes training, 350 MB model, <1 ms latency.

3.3.2. Production Selection: XGBoost

Finding F3 (RQ2): XGBoost achieves higher classification performance than BERT (99.49% ± 0.3% vs. 96.15% ± 0.8% F1, 5-fold CV; Table 4) with non-overlapping confidence intervals confirming statistical significance, while requiring 137× less storage and 50× lower inference latency, empirically validating that resource-efficient models do not sacrifice accuracy for compliance auditing tasks. SHAP-based explainability provides per-decision audit trails satisfying AU-2 accountability requirements.

Table 4. Model Comparison for Production Deployment.

Metric	BERT	LSTM	XGBoost
F1-Score (mean ± 95% CI)	96.15% ± 0.8%	96.11% ± 1.2%	**99.49% ± 0.3%**
Accuracy	96.21%	96.08%	**99.49%**
Training Time	12 hrs	2 hrs	**8 min**
Model Size	2.7 GB	800 MB	**350 MB**
Latency	45 ms	12 ms	**<1 ms**
Memory	2.1 GB	580 MB	**380 MB**
Explainability	Low	Low	**High (SHAP)**
GPU Required	Yes	No	**No**

95% CI computed via 5-fold stratified cross-validation (n = 100,000).

3.4. Feature Engineering

Numeric Features (5): Hour of day (0–23), day of week (0–6), business hours binary flag, network port number, event severity level.

Text Features (25 TF-IDF): Vectorized `log_message` field with unigrams, minimum document frequency 5, top discriminating terms: *failed*, *unauthorized*, *blocked*, *encrypted*, *timeout*, *denied*, *accepted*. `status_code` removed due to identified leakage.

Total: 30 features per event.

3.5. Multi-Label Formulation

Problem: Map evidence x_i to a subset of controls $\mathcal{C}_i \subseteq \{c_1, \ldots, c_{143}\}$.

Implementation: Multi-output XGBoost via scikit-learn's `MultiOutputClassifier` wrapper, producing 143-dimensional binary output vectors.

Evaluation Metrics:

$$F1_{\text{macro}} = \frac{1}{143} \sum_{k=1}^{143} F1_k \tag{2}$$

$$F1_{\text{micro}} = \frac{2 \cdot P_{\text{all}} \cdot R_{\text{all}}}{P_{\text{all}} + R_{\text{all}}} \tag{3}$$

$$\mathcal{L}_H = \frac{1}{N \cdot 143} \sum_{i=1}^{N} \sum_{j=1}^{143} \mathbb{1}[y_{ij} \neq \hat{y}_{ij}] \tag{4}$$

Macro-F1 (Equation (2)) is our primary metric as it weights each control equally, preventing high-frequency controls from dominating evaluation.

3.6. Evolution to Semantic LLM Analysis

The hybrid LLM architecture emerged from a systematic progression through three phases, each exposing a limitation that motivated the next step.

 doi:10.20944/preprints202604.1589.v1

Phase 1 — Classifier Comparison (BERT, LSTM, XGBoost): Initial evaluation of all three candidate models on the hybrid synthetic dataset (Section 3) demonstrated that while BERT (96.15% F1) and LSTM (96.11% F1) offered competitive accuracy, both require GPU infrastructure unavailable in resource-constrained deployments. XGBoost was selected for production: 99.49% F1, 8-minute training, 350 MB model, <1 ms inference on CPU-only hardware (Table 4).

Phase 2 — Generalization Gap Discovery: Cross-dataset evaluation on real SecRepo logs (86,839 samples) revealed a critical failure: XGBoost zero-shot F1 collapsed to 7.98%,a 92-point gap from synthetic performance. The root cause is architectural: TF-IDF features capture vocabulary co-occurrence patterns rather than semantic meaning. A log containing "authentication failure" is correctly classified only if that exact vocabulary appeared in training data with the same distribution. Fine-tuning on 5% target-domain data recovers performance (99.88% F1), but this approach does not scale: a new log type requires a new labeled sample collection, defeating the goal of zero-shot generalization across institutional environments. BERT was reconsidered at this point but rejected for the same infrastructure constraint: its semantic capability requires GPU-backed inference impractical at $50/month target cost.

Phase 3 — LLM Semantic Path: Large language models offer semantic understanding without retraining,their pre-training on vast corpora enables zero-shot reasoning about security log semantics regardless of format. GPT-4o-mini (cloud API) and Llama-3.2-3B (on-premise CPU) were evaluated as zero-shot classifiers across four log types (Section 5), achieving 93.5% and 84.0% overall accuracy respectively,a direct solution to the generalization gap that neither fine-tuned XGBoost nor BERT could address at acceptable deployment cost. An integration layer was needed to connect the compliance pipeline to these LLM providers in a provider-agnostic way.

3.6.1. Model Context Protocol Integration

To overcome vocabulary dependency, Engine 3 integrates LLMs via Model Context Protocol (MCP) [30,31],an open standard that serves as the *engineering integration layer* between the compliance pipeline and external LLM providers. MCP standardizes the tool-use interface, enabling provider-agnostic LLM calls without vendor lock-in; it is not a scientific contribution of this paper but rather a practical interface choice that simplifies deployment and maintenance:

Dual-Path Architecture:

1. **Rule-based fast path**: Regex pattern matching for high-confidence, unambiguous events (e.g., "Failed password" → non-compliant, "Accepted publickey" → compliant), achieving <1 ms latency. Handles 60–70% of logs.
2. **LLM semantic path**: Claude/GPT analysis for ambiguous logs requiring contextual interpretation, achieving 200–500 ms latency. Handles 30–40% of logs.

3.6.2. LLM Evaluation Protocol

Two complementary evaluations were conducted. **Evaluation 1** (SSH deep-dive): a stratified random sample of 500 SecRepo authentication logs,300 failed authentication events (60%), 150 successful authentication events (30%), and 50 session anomalies (10%). Ground truth labels established through automated rule-based analysis cross-validated with manual review of all 50 anomalous cases. **Evaluation 2** (multi-log-type expanded, GPT-4o-mini and Llama-3.2-3B, temperature=0): 50 samples per log type across four structurally distinct categories,SSH authentication, macOS system/service logs, HTTP/API access logs, and Windows Security Events,totalling 200 samples. A parallel evaluation with Llama-3.2-3B (on-premise, CPU-only via Ollama) was conducted on the same sample set to validate on-premise deployment viability. Ground truth for each type established by domain-specific rule-based classifiers, all cross-validated manually.

3.6.3. Cost Optimization

- MD5-keyed response caching (10,000 entry LRU cache): identical log formats reuse cached decisions.

 doi:10.20944/preprints202604.1589.v1

- Model tiering: Claude Haiku for batch processing ($0.00025/1K tokens), Claude Sonnet for single-event analysis.
- Rule-based pre-filtering eliminates LLM calls for 60–70% of logs.
- Estimated cost: $0.15/10,000 logs with hybrid strategy vs. $1.50 with LLM-only.

4. System Implementation

4.1. Control Coverage

The system implements 143 NCSA controls across 7 families (Table 5), with 96 system-auditable controls supported by 60 specialized evidence parsers executing macOS audit commands.

Table 5. Implemented Control Families.

Control Family	Total	System-Auditable
Access Control (AC)	52	35
Audit & Accountability (AU)	30	22
Configuration Management (CM)	18	12
Identity & Authentication (IA)	15	10
System & Comm. Protection (SC)	26	15
System & Info. Integrity (SI)	2	2
Total	**143**	**96**

Evidence Parsers: Each system-auditable control has a dedicated parser executing macOS audit commands. The 60 parsers are organized by family: Access Control (26): login history via `last`, user enumeration via `dscl`, SSH configuration via `sshd -T`, screen lock, file permissions; Audit & Accountability (7): audit subsystem via `auditctl`, NTP sync via `ntpq`, log retention; Identity & Authentication (5): password policy via `pwpolicy`, MFA certificates; System & Communications Protection (11): firewall via `pfctl`, encryption via `fdesetup status`, VPN, WiFi; System & Information Integrity (4): SIP via `csrutil status`, XProtect/Gatekeeper; Configuration Management (7): software inventory, patch status.

4.2. Decision Engine Architecture

Each control implements a dedicated parser and decision function returning a `ComplianceResult`: status $\in$ {compliant, partial, non_compliant}, confidence $\in [0, 1]$, gaps list, and remediation recommendation. Control-specific thresholds implement NCSA requirements: AC-7 checks lockout threshold ≤ 5 failed attempts within 15 minutes; IA-5 checks password minimum length ≥ 12 characters; SC-28 checks FileVault encryption enabled.

4.3. Database Schema

PostgreSQL stores the control taxonomy (`controls`: control_id, family, title, description, ncsa_mapping, priority, tier) and audit results (`audit_results`: audit_id, control_id, status, confidence, reason, gaps, recommendations, audited_at). Redis pub/sub channels enable real-time audit progress streaming to the WebSocket-connected dashboard.

4.4. Kubernetes Deployment

Namespace `rwanda-compliance` with 2–3 replicas per engine, resource limits 512 Mi–1 Gi memory and 250m–500m CPU per pod. NGINX ingress controller routes external traffic; ClusterIP services handle internal inter-engine communication. Liveness and readiness probes on `/health` endpoints with 30s initial delay ensure zero-downtime pod restarts.

 doi:10.20944/preprints202604.1589.v1

5. Results and Evaluation

5.1. Classification Performance (XGBoost)

Table 6 presents XGBoost classification results with 5-fold stratified cross-validation on the hybrid synthetic dataset (n = 100,000).

Table 6. XGBoost Classification Results (5-fold CV, n = 100,000 synthetic).

Metric	Value (Mean ± Std)
Accuracy	100.00% ± 0.00%
Precision	99.99% ± 0.02%
Recall	100.00% ± 0.00%
F1-Score	99.99% ± 0.01%
ROC-AUC	1.0000 ± 0.0000
Training time/fold	89.09 ± 7.25 s
Inference Latency (500 samples)	
p50	0.32 ms
p95	5.40 ms
p99	8.78 ms

Interpreting Near-Perfect Synthetic Performance: The near-zero variance and near-perfect F1 on the synthetic dataset are expected given the controlled generation process: each synthetic event was generated with explicit control labels, creating a structured but internally consistent distribution. This does *not* indicate memorization of training instances,confirmed by consistent cross-validation fold performance,but rather reflects the low intra-class variability of template-based synthetic data.

Performance varied across control families, with high-frequency families (AC, AU) benefiting from greater sample density while rare controls in SI (2 controls, limited training examples) showed slightly higher variance. The 5-fold cross-validation standard deviations (F1: ±0.01%) confirm stable performance without overfitting to specific folds. *The critical validation of generalization is the cross-dataset evaluation* (Section 5.2), where zero-shot performance degrades substantially (7.98% F1),demonstrating that the high synthetic accuracy reflects distribution-specific learning, not genuine semantic understanding of security events.

5.2. Cross-Dataset Generalization

Finding F1 (RQ3): Zero-shot transfer shows catastrophic domain shift (7.98% F1 on real logs vs. 99.99% synthetic), confirming that vocabulary-based models do not generalize across log distributions without exposure to target-domain data. The 92-point gap directly answers RQ3: hybrid datasets improve in-distribution performance but do not themselves solve zero-shot generalization.

Table 7. Cross-Dataset Evaluation (SecRepo auth.log, n = 86,839 real logs).

Evaluation Scenario	F1	Accuracy
Synthetic CV (in-distribution)	99.99%	100.00%
Zero-shot (train synthetic only)	7.98%	5.92%
Transfer Learning (fine-tune on % of SecRepo)		
5% (2,170 samples)	99.88%	99.92%
10% (4,341 samples)	99.90%	99.94%
20% (8,683 samples)	100.00%*	100.00%*
External CV (upper bound)	99.99%	99.99%

*Near-ceiling performance observed with 20% fine-tune data on structured SSH authentication logs (binary classification, limited vocabulary). Performance on diverse real-world logs would likely be lower; cross-dataset evaluation (zero-shot 7.98%) provides the relevant generalization baseline.

 doi:10.20944/preprints202604.1589.v1

Finding F2 (RQ3): Fine-tuning on just 5% of target-domain data (2,170 samples) recovers 99.88% F1, demonstrating strong transfer learning with minimal labeled data investment. This validates a practical deployment strategy: pre-train on synthetic data for full NCSA control coverage, then fine-tune on a small institution-specific sample (≈3 hours of labeling effort).

5.3. LLM Semantic Analysis Results

LLM Accuracy Scope: The 100% accuracy requires precise scope qualification. The evaluation was conducted on SSH authentication logs,a semantically clear, binary classification task in highly structured syslog format with domain-invariant security terminology. Ground truth was established through automated rule-based analysis ("Failed password" → non-compliant; "Accepted" → compliant), cross-validated with manual review of 50 ambiguous samples.

Table 8. LLM vs. XGBoost on SecRepo SSH Authentication Logs (500-sample stratified evaluation).

Metric	XGBoost	LLM (MCP)
Zero-shot Accuracy	5.92%	**up to 100%**†
Zero-shot F1	7.98%	**up to 100%**†
Average Confidence	87.3%	**95.2%**
Latency (per log)		
Rule-based path	—	0.04 ms
LLM path	—	280 ms
XGBoost	0.32 ms	—
Resource Requirements		
Model Storage	350 MB	0 MB (API)
Memory (runtime)	380 MB	50 MB
GPU Required	No	No

†Up to 100% accuracy achieved in controlled binary SSH authentication classification tasks (structured syslog format, binary compliant/non-compliant labels, n=500). Accuracy is log-type-dependent; multi-log evaluation (Table 9) shows 93.5% overall across four structurally distinct log types at n=200.

Finding F4 (RQ0, LLM generalization): To validate generalizability beyond SSH logs, zero-shot evaluations were conducted across *four* structurally distinct log types using two models: GPT-4o-mini (cloud API) and Llama-3.2-3B (on-premise via Ollama on Apple M1 Pro CPU, no GPU),directly testing the African SME on-premise deployment scenario (Table 9). Results confirm that LLM accuracy is log-type-dependent but consistently superior to XGBoost zero-shot (7.98%) across all categories for both models.

Table 9. LLM Zero-Shot Accuracy by Log Type and Model (Rwanda NCSA, n = 50 per type).

Log Type	n	GPT-4o-mini	Llama-3.2-3B	Avg Conf.
SSH Authentication	50	84.0%	82.0%	87.1%
macOS System/Service	50	92.0%	90.0%	91.1%
HTTP/API Access	50	98.0%	64.0%	89.8%
Windows Security Events	50	**100.0%**‡	**100.0%**‡	90.4%
Overall	**200**	**93.5%**	**84.0%**	**89.6%**
XGBoost zero-shot (baseline for comparison)				
All types	—	7.98% F1	7.98% F1	—

GPT-4o-mini and Llama-3.2-3B: temperature=0, zero-shot, March 2026. Ground truth: rule-based classifiers + manual cross-validation. Llama-3.2-3B deployed locally on Apple M1 Pro (CPU-only, 2.0 GB model). 95% Wilson CI at n=50: ±8–14 percentage points per cell. ‡100% observed on structured Windows Security Event IDs (binary classification, n=50, small sample); should be interpreted with caution at this sample size.

Key Findings from Expanded Evaluation: GPT-4o-mini achieves 93.5% overall accuracy across 200 samples and 4 log types,a 10.2-percentage-point improvement over the initial 3-type evaluation

 doi:10.20944/preprints202604.1589.v1

(83.3%, n=30), attributable to the addition of Windows Security Event logs (100% accuracy on both models at n=50; interpret with caution given small sample size) and the larger sample base narrowing distributional noise. Llama-3.2-3B achieves 84.0% overall, demonstrating that a freely available 3-billion-parameter model running on commodity CPU hardware provides accuracy exceeding the XGBoost zero-shot baseline by 76 percentage points,validating on-premise LLM deployment as a cost-viable path for African institutions without API budget.

Differential Analysis: The 29.5-percentage-point gap between GPT-4o-mini (98%) and Llama-3.2-3B (64%) on HTTP/API Access logs reveals a meaningful capability difference: HTTP logs require recognition of security tool user-agents (`sqlmap`, `nikto`, `hydra`, `dirbuster`) and path traversal patterns, where the larger model's broader training corpus provides a substantial advantage. SSH and Windows Event logs show near-parity between models (2–4 pp gap), suggesting structured, schema-like logs are less sensitive to model scale. This differential informs the deployment recommendation: Llama-3.2-3B is sufficient for structured log types (SSH, Windows Events); GPT-4o-mini or equivalent is preferred for HTTP/API logs in high-stakes environments.

Statistical Scope: At n=50 per log type, 95% Wilson confidence intervals span ±8–14 percentage points per cell,narrowed from ±30 pp at the prior n=10 evaluation. Results are now sufficient to establish directional performance rankings with statistical reliability, though validation on production Rwandan institution logs ($n \geq 500$ per type) remains for future deployment pilots.

Error Analysis: Misclassifications in both models predominantly produced *partial* labels rather than opposite-polarity errors, confirming LLM uncertainty rather than systematic misunderstanding. SSH errors arose from semantically ambiguous disconnection logs ("Connection closed [preauth]", "Received disconnect...Bye Bye"); macOS errors involved services with dual-interpretation status (screen sharing, SMB sharing where organizational policy context is required). This pattern,uncertain partial labels rather than confident wrong labels,is favorable for compliance auditing: ambiguous events trigger human review rather than silently incorrect certifications, satisfying AU-2 accountability requirements.

5.4. Adaptive Routing: Vocabulary-Coverage Gating (Phase II)

The hybrid architecture routes each log through either XGBoost (fast path) or the LLM (semantic path) based on a vocabulary-coverage signal:

$$\nu(l) = \frac{|\text{words}(l) \cap \mathcal{V}|}{|\text{words}(l)|}, \quad \text{route}(l) = \begin{cases} \text{XGBoost} & \nu(l) \geq \theta \\ \text{LLM} & \nu(l) < \theta \end{cases} \tag{5}$$

where $\mathcal{V}$ is the TF-IDF training vocabulary and θ is the coverage threshold. A threshold sweep across $\theta \in \{0.00, 0.01, 0.05, 0.10, 0.15, 0.20, 0.30, 0.50, 1.00\}$ was conducted on a mixed evaluation set of n=220 logs (20 SSH in-distribution + 200 OOD logs from the 4-type expanded evaluation).

Table 10. Phase I vs. Phase II Router Comparison (n = 220 mixed logs).

Router	Accuracy	LLM Call Rate	Cost/10K logs
XGBoost only (no routing)	42.3%	0.0%	$0.001
LLM only	94.5%	100.0%	$0.150
Phase I (static regex routing)	89.1%	90.9%	$0.137
Phase II (θ=0.20, vocab-cov)	**94.5%**	99.5%	$0.149

Cost model: LLM path $0.15/10K logs (GPT-4o-mini); XGBoost $0.001/10K logs (CPU inference).

Finding F7 (Phase II): Vocabulary-coverage gating at θ=0.20 matches LLM-only accuracy (94.5%) while reducing unnecessary XGBoost calls on OOD logs. Phase II improves over Phase I by +5.4 percentage points in accuracy (+0.6 pp over LLM-only overall) at a marginal cost increase of $0.012/10K logs. The OOD detection signal is near-perfect: vocabulary coverage = 0.00 for all HTTP, Windows

 doi:10.20944/preprints202604.1589.v1

Event, and macOS logs (F1=0.93; Precision=0.91, Recall=0.96 at θ=0.05), confirming that TF-IDF vocabulary absence reliably identifies out-of-distribution log types without a separate OOD classifier.

5.5. Latency and Throughput

Throughput: Single pod: ∼100 events/s; 3-pod deployment: ∼300 events/s (linear scaling observed to 5 pods). Training: 89.09 ± 7.25 s per fold on CPU.

5.6. Resource Utilization

Total system footprint across all 7 engines: 2.66 GB RAM, 2.0 CPU cores. Minimum deployment: 4 GB RAM, 4 cores at ∼$50/month cloud hosting,substantially lower than enterprise compliance solutions charging $15–$50/endpoint/month.

5.7. Kubernetes Scaling Test

Engine 3 scaled to 3 replicas under 300 concurrent requests:

- Load distribution: 34.0% / 32.7% / 33.3% (near-uniform, within 1.3% spread).
- Latency: 1.8 ms mean, 4.2 ms p99.
- Failures: 0 across 10 independent test iterations.

5.8. Live System Deployment Validation

To validate end-to-end system correctness beyond simulated benchmarks, a full compliance audit was executed on the research machine (macOS 26.3.1) using the production seven-engine pipeline (Audit ID: AUDIT-20260320-205239, March 20, 2026). The system autonomously collected evidence via 60 parsers across 5 control families, classified findings via XGBoost and the Decision Engine, and generated a 3-page structured PDF report in 0.77 s.

Results across 10 audited controls:

- **5/10 compliant**: FileVault encryption (AC-17), Gatekeeper active (CM-6), Firewall enabled (SC-7), Microsoft Defender ATP running (SI-3), audit logging active (AU-2).
- **5/10 partial**: Password policy below NCSA threshold, screen sharing unreviewed (AC-17 partial), SIP status requiring investigation, MDM enrollment gap, SSH key rotation overdue.
- **0/10 non-compliant**: No outright policy violations detected.
- **Overall compliance score**: 75.0%.

This live deployment demonstrates that the system produces actionable, granular audit findings on real production infrastructure. The 75% score reflects realistic partial compliance, validating the system's discriminative capability in production conditions. Report generation time (0.77 s) is within the 2–5 s target specified in Table 11.

Table 11. System Performance (500 inference samples).

Operation	Latency
XGBoost Inference (p50)	0.32 ms
XGBoost Inference (p95)	5.40 ms
XGBoost Inference (p99)	8.78 ms
Decision Engine (per control)	5–10 ms
End-to-End Pipeline (single log)	10–15 ms
Report Generation	2–5 s

5.9. Adversarial Validation

Adversarial validation was conducted in an isolated virtual machine environment (Windows 10 21H2, network-isolated) following responsible disclosure practices. Tests evaluate whether the system identifies "partial compliance",scenarios where a control is nominally present but ineffective against the tested technique.

 doi:10.20944/preprints202604.1589.v1

Test 1: Fileless Reverse Shell (SI-3, SI-4, SC-7 effectiveness): A custom reverse shell was implemented using the shellcode-separation technique from [13]: shellcode loaded into memory via `VirtualAlloc+CreateThread` without disk writes, evading 43/44 AV engines tested. The compliance auditor processed 5 execution attempts. Decision Engine outcome: SI-3 marked **partial**,EDR behavioral alert triggered but signature-based AV failed to block the payload. SI-4 marked **compliant**,system monitoring captured the execution.

Test 2: GA-Generated XSS Payloads (SI-10, AC-3 effectiveness): 50 GAXSS-inspired XSS payload variants were tested against a web application's input validation layer. Results: 34/50 payloads (68%) blocked by WAF signature rules; 16/50 payloads (32%) bypassed input validation. Decision Engine: SI-10 marked **non-compliant**,32% bypass rate exceeds NCSA's implicit zero-tolerance standard for input validation. A checkbox audit ("WAF installed?") would have returned **compliant**,demonstrating the effectiveness gap our system captures.

Finding F5 (RQ1): Adversarial testing reveals effectiveness gaps invisible to presence-based checkbox auditing. SI-3 and SI-10 receive "partial" and "non-compliant" determinations respectively, despite both controls being nominally installed. This directly answers RQ1: adversarial techniques are *necessary* to distinguish control installation from control effectiveness.

5.10. Research Questions Summary

Table 12. Research Questions and Experimental Evidence.

RQ	Question Focus	Evidence
RQ0	Achieve 65–80% macro F1 with ≥50% cycle reduction	99.88% F1 (5% fine-tune); 0.77 s live audit vs. 1,000 h manual
RQ1	Adversarial techniques inform compliance	SI-3 partial (20% det. rate), SI-10 non-compliant (32% XSS bypass)
RQ2	Model comparison in resource-constrained env.	XGBoost: 137× smaller, 50× faster than BERT; Llama-3.2-3B: 84% zero-shot on CPU
RQ3	Hybrid dataset overcomes synthetic overfitting	Zero-shot: 7.98%; 5% fine-tune: 99.88%; vocab-cov router: 94.5%
RQ4	Multi-label vs. binary classification	3.2 controls avg./event; binary loses per-control evidence
—	LLM generalization across log types	GPT-4o-mini: 93.5% (n=200, 4 types); Llama-3.2-3B: 84.0% (CPU-only)

6. Discussion

6.1. Integration of Adversarial Security Research (RQ1)

Our methodology connects offensive security research to compliance validation through three concrete mappings:

1. Adversarial Evidence Generation: GA and LLM techniques generate diverse attack scenarios for training data augmentation. The 50 GAXSS payload variants used in Test 2 could be scaled to thousands of variants covering MITRE ATT&CK sub-techniques, providing realistic SI-10 training examples absent from public datasets.

2. Control Effectiveness Quantification: Evasion results provide measurable effectiveness metrics. SI-3 effectiveness against the fileless reverse shell: $E_{SI\text{-}3} = \text{detected}/(\text{detected} + \text{bypassed}) = 1/5 = 0.20$ (EDR detected one of five execution attempts). This maps directly to our theoretical effectiveness model (Section 6.3).

3. Risk-Calibrated Scoring: Knowledge that 97% evasion rates are achievable [13] informs control weight assignments. Institutions relying solely on signature-based AV for SI-3 compliance receive risk-adjusted scores reflecting demonstrated real-world ineffectiveness rather than nominal installation.

 doi:10.20944/preprints202604.1589.v1

6.2. Multi-Label vs. Binary Classification (RQ4)

The pivot from binary to multi-label classification addresses a fundamental compliance modeling mismatch.

Binary (inadequate): f: Log $\rightarrow$ {compliant, non-compliant}

Multi-Label (correct): f: Log $\rightarrow \{c_j : c_j \in \mathcal{C}\}$

Example: "Failed SSH login from 203.0.113.15" maps to four controls simultaneously,AC-7 (lockout threshold exceeded?), IA-2 (authentication bypass attempted?), AU-2 (event audited?), SI-4 (monitoring detecting intrusion?),each evaluated independently. In our evaluation, the average log event triggered 3.2 compliance control evaluations, meaning binary classification discards 69% of actionable audit information per event.

Finding F6 (RQ4): Multi-label formulation is not merely a modeling convenience,it is the correct representation of compliance semantics. A single security event simultaneously provides evidence for multiple controls, and collapsing this to a binary label destroys information that is legally and operationally significant under NCSA reporting requirements.

6.3. Theoretical Contributions

6.3.1. Effectiveness-Based Compliance Model

Prior compliance frameworks treat control assessment as binary presence detection: $C_{\text{presence}}(k) \in \{0,1\}$,a control either exists or it does not. This model is provably insufficient: our adversarial tests demonstrate that a control can satisfy $C_{\text{presence}}(k) = 1$ while providing near-zero actual security benefit ($C_{\text{presence}}(\text{SI-3}) = 1$ with fileless shell detection rate $= 0.20$).

We propose replacing binary presence checking with a continuous effectiveness function:

$$C(k) = \alpha \cdot D_k + \beta \cdot V_k + \gamma \cdot R_k, \quad \alpha + \beta + \gamma = 1 \tag{6}$$

where $D_k \in [0,1]$ is the detection rate (fraction of attack instances targeting control k's scope that are detected), $V_k \in [0,1]$ is coverage (fraction of log sources monitored for control k events), $R_k \in [0,1]$ is evasion resistance ($1 - \text{evasion\_rate}_k$), and α, β, γ are configurable institutional weights (default: $\alpha = 0.5, \beta = 0.3, \gamma = 0.2$, reflecting a detection-priority weighting appropriate for resource-constrained institutions).

This formulation satisfies three formal properties: (i) *Monotonicity*,improving any component cannot decrease $C(k)$; (ii) *Boundedness*,$C(k) \in [0,1]$ enabling consistent cross-control and cross-institution comparison; (iii) *Decomposability*,each component is independently measurable and improvable, enabling targeted remediation rather than pass/fail verdicts.

Relation to Prior Compliance Models: The NIST SP 800-53 control assessment approach implicitly models compliance as $C_{\text{NIST}}(k) \in \{\text{satisfied}, \text{other than satisfied}\}$,a binary model. The CIS Controls framework introduces prioritization tiers but maintains categorical rather than continuous scoring. ISO 27001 Annex A similarly uses not-implemented / partially implemented / implemented categories. Our model is strictly more expressive: $C_{\text{presence}}(k) = 1$ is equivalent to $C(k) = \alpha \cdot 1 + \beta \cdot V_k + \gamma \cdot 1$ only when $D_k = 1$ and $R_k = 1$,a condition our adversarial tests show is frequently violated in practice.

Experimental Grounding: For Test 1, $D_{SI\text{-}3} = 0.20$, $V_{SI\text{-}3} = 1.0$, $R_{SI\text{-}3} = 0.014$, yielding $C(\text{SI-3}) = 0.5(0.20) + 0.3(1.0) + 0.2(0.014) = 0.403$,consistent with the "partial" compliance determination and distinct from the $C = 1.0$ a presence-based audit would assign. This 0.597-point gap represents the effectiveness deficit invisible to conventional auditing.

6.3.2. Multi-Label Compliance Framework

The formulation $\mathbf{y} \in \{0,1\}^K$ for K controls provides a transferable framework applicable to any hierarchical taxonomy (NCSA, ISO 27001, NIST SP 800-53, CIS Controls). Control taxonomies differ in their K and label semantics, but the multi-output classification architecture requires no structural changes,only re-labeling of training data according to the target taxonomy's mapping logic. This framework-agnostic property is significant for the African context, where different nations may

 doi:10.20944/preprints202604.1589.v1

enforce different standards (Rwanda NCSA, Kenya CMCA, Nigeria NITDA) while sharing the same infrastructure.

6.3.3. Adversarial-Informed Validation Theory

We argue that compliance validation without adversarial testing provides *systematically inflated* assurance, not merely incomplete assurance. This distinction matters: inflated assurance creates a false sense of security that may discourage investment in genuine defensive capability. The mechanism for inflation is precisely captured by the gap between $C_{\text{presence}}(k) = 1$ and $C(k) = 0.403$ in our SI-3 example.

This connects to game-theoretic security models [15] where defender effectiveness must be measured against adaptive adversaries, not fixed attack signatures. A compliance framework that does not model adversary behavior cannot distinguish a control that blocks 99% of attacks from one that blocks 1%,both satisfy $C_{\text{presence}}(k) = 1$. Our contribution is to instantiate this theoretical gap with concrete empirical measurements within a nationally-defined regulatory framework.

6.4. Theoretical Synthesis

Findings F1–F6 collectively validate the central theoretical claim: compliance effectiveness cannot be assessed through presence-based binary checking but requires three independently measurable components formalized in Equation (6): detection rate D_k, coverage V_k, and evasion resistance R_k.

F1 shows that D_k collapses to near-zero when measured against out-of-distribution logs, because vocabulary-based models conflate detection capability with training-distribution membership. F2 shows that D_k is recoverable with minimal target-domain exposure, making V_k the dominant practical constraint. F5 directly instantiates R_k: SI-3's evasion resistance is $R_k = 0.014$, yielding $C(\text{SI-3}) = 0.403$ rather than the $C = 1.0$ that presence-based auditing assigns. F4 shows that D_k is log-type-dependent even for LLMs. F6 shows that the correct output space is $\mathbf{y} \in \{0, 1\}^K$. F3 shows that the resource cost of achieving high D_k and V_k is not a barrier in the resource-constrained African context. Together, these six findings provide empirical grounding for a compliance model that is more expressive, more honest, and more actionable than existing approaches.

6.5. Resource Efficiency for African Contexts (RQ2)

XGBoost's 137× size reduction and CPU-only execution enable deployment that BERT makes infeasible for Rwandan SMEs. With minimum requirements of 1 GB RAM and 2 CPU cores, the system aligns with available infrastructure. The cloud deployment cost ($50/month) compares favorably to current manual audit costs ($154–$341 per-capita). The MCP+LLM path adds ∼$0.15 per 10,000 logs,negligible for typical institutional audit volumes.

6.6. Ethical Considerations

Automation Bias: Organizations may over-trust ML compliance outputs. Mitigation: SHAP explanations accompany all classification decisions; Decision Engine outputs require human sign-off for non-compliant findings.

Accountability: Incorrect compliance determinations carry institutional and legal implications. Design positions ML as advisory; final compliance certifications require authorized human reviewer signature.

Privacy: Centralized log collection creates surveillance risks. Mitigation: data minimization (only compliance-relevant features stored); raw logs processed in-memory and not persisted.

Fairness: Institutions with less structured logging infrastructure may receive lower compliance scores due to parsing limitations rather than genuine non-compliance. Mitigation: system provides logging improvement recommendations rather than penalizing format variations.

6.7. Limitations

1. **Control Coverage**: 143/169 NCSA controls (85%); 26 controls require physical inspection or manual policy review not automatable via system audit commands.

 doi:10.20944/preprints202604.1589.v1

2. **Synthetic Data Realism**: Near-perfect synthetic performance (99.99% F1) reflects template-based generation simplicity. Real-world log diversity would reduce synthetic performance and improve generalization testing realism.
3. **LLM Evaluation Scope**: The up-to-100% LLM accuracy is specific to SSH authentication binary classification on structured syslog format. Multi-log-type evaluation shows 93.5% overall zero-shot accuracy (GPT-4o-mini) and 84.0% (Llama-3.2-3B) across 4 log types (n = 200), ranging from 82–100% by type and model. Validation on production institutional logs ($n \geq 500$ per type) remains for future deployment pilots.
4. **Real-World Institution Validation**: Experiments use public datasets (SecRepo, LANL); validation on production Rwandan institution logs remains for pilot deployment.
5. **Adversarial Coverage**: Validation covers 2 attack scenarios; comprehensive coverage of all relevant MITRE ATT&CK techniques requires a continuous adversarial testing pipeline.
6. **LLM Cost at Scale**: Processing >10,000 logs/hour with the LLM path at current API costs is impractical without the rule-based fast path. On-premise LLM deployment (Llama, Mistral) is needed for high-volume deployments.

7. Future Work

Short-Term:

1. Evaluate LLM analysis on heterogeneous log types: Windows Event logs, API access logs, multi-line application logs.
2. Conduct pilot deployments at 2–3 Rwandan institutions to validate on production logs.
3. Optimize LLM prompt engineering for partial compliance edge cases.
4. Add per-control F1 breakdown to classify model performance by control family.

Medium-Term:

1. Complete remaining 26 NCSA controls requiring physical/manual assessment.
2. Cross-framework mapping: NCSA $\leftrightarrow$ ISO 27001 $\leftrightarrow$ NIST SP 800-53 $\leftrightarrow$ CIS Controls.
3. Fine-tune open-source LLMs (Llama-3.1-8B, Mistral-7B-Instruct) for on-premise deployment eliminating API dependency.
4. Integrate open-source EDR (Wazuh, OSSEC) for cost-effective SI-3/SI-4 validation.
5. Evaluate cross-regulatory transfer: Kenya CMCA, Nigeria NITDA, South Africa POPIA.

Long-Term:

1. Pan-African compliance platform supporting Malabo Convention.
2. Federated learning for cross-institution model improvement preserving data privacy.
3. Continuous adversarial red-teaming pipeline integrated with compliance validation cycles.
4. Blockchain-based audit trail for evidence integrity and non-repudiation.

8. Conclusions

This research presents an AI-augmented compliance auditor evaluated on Rwanda's NCSA Minimum Cybersecurity Standards as a representative national framework, integrating adversarial security research with multi-label ML–LLM classification for automated compliance validation.

Key results: (1) 143 NCSA controls implemented (85% coverage) with 60 specialized evidence parsers; (2) XGBoost multi-label classification achieves 99.88% F1 on real logs after 5% fine-tuning, but collapses to 7.98% zero-shot F1,confirming the generalization gap as a fundamental constraint of vocabulary-based models; (3) GPT-4o-mini achieves 93.5% zero-shot accuracy across 4 log types (n=200),SSH (84%), macOS (92%), HTTP/API (98%), Windows Events (up to 100% in controlled binary classification),while Llama-3.2-3B on CPU-only hardware achieves 84.0%, validating on-premise LLM deployment for African SMEs; (4) vocabulary-coverage gating (Phase II router, θ=0.20) achieves 94.5% accuracy at \$0.149/10K logs, a +5.4pp improvement over static routing; (5) adversarial validation demonstrates effectiveness gaps invisible to

 doi:10.20944/preprints202604.1589.v1

presence-based auditing (SI-3 partial at 20% detection; SI-10 non-compliant at 32% XSS bypass); (6) total resource footprint: 2.0 CPU cores, 2.66 GB RAM, $50/month.

The effectiveness-based compliance model $C(k) = \alpha D_k + \beta V_k + \gamma R_k$ formalizes control quality as a continuous, measurable function of detection capability, log coverage, and adversarial resistance,replacing binary presence checking with actionable posture scoring. For Rwandan institutions, the system provides $\geq$50% audit cycle reduction, continuous monitoring capability, and cost-accessible deployment. Beyond Rwanda, this provides a replicable blueprint for automated compliance auditing across African nations pursuing cybersecurity maturity under the Malabo Convention.

Author Contributions: Conceptualization, M.I.I. and J.D.N.; methodology, M.I.I. and J.D.N.; software, M.I.I.; validation, M.I.I. and J.D.N.; formal analysis, M.I.I.; investigation, M.I.I.; resources, M.I.I.; data curation, M.I.I.; writing—original draft preparation, M.I.I.; writing—review and editing, M.I.I. and J.D.N.; visualization, M.I.I.; supervision, J.D.N.; project administration, M.I.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The compliance auditor codebase, evaluation scripts, and anonymized results are available at the project repository. The SecRepo SSH authentication dataset used for cross-dataset evaluation is publicly available at https://www.secrepo.com/. Rwanda-specific synthetic data generation scripts are included in the repository.

Acknowledgments: We acknowledge Carnegie Mellon University Africa for supporting this research and the Rwanda National Cyber Security Authority (NCSA) for establishing the Minimum Cybersecurity Standards that motivated this work. During the preparation of this manuscript, the authors used OpenAI ChatGPT (GPT-4o, GPT-4o-mini) and Meta LLaMA 3.1 for the following purposes: (1) as evaluated components of the compliance-auditing system under experimental study (Sections 3 and 4); (2) to assist with writing and structuring LaTeX source code and manuscript formatting; and (3) to paraphrase selected passages to meet journal style and clarity requirements. The authors have reviewed and edited all AI-assisted output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

NCSA	National Cyber Security Authority
MCP	Model Context Protocol
LLM	Large Language Model
ML	Machine Learning
XGBoost	Extreme Gradient Boosting
BERT	Bidirectional Encoder Representations from Transformers
LSTM	Long Short-Term Memory
NIST	National Institute of Standards and Technology
SHAP	SHapley Additive exPlanations
GA	Genetic Algorithm
AV	Antivirus
EDR	Endpoint Detection and Response
SIEM	Security Information and Event Management
TF-IDF	Term Frequency–Inverse Document Frequency
JWT	JSON Web Token
RBAC	Role-Based Access Control
API	Application Programming Interface
SME	Small and Medium Enterprise

 doi:10.20944/preprints202604.1589.v1

References

1. OpenSCAP Project. OpenSCAP: Open Source Security Compliance Solution. Available online: https://www.open-scap.org/ (accessed on 20 March 2026).
2. Wazuh, Inc. Wazuh: Open Source Security Platform. Available online: https://wazuh.com/ (accessed on 20 March 2026).
3. Ponemon Institute. The True Cost of Compliance: A Benchmark Study of Multinational Organizations. *Ponemon Institute Research Report*, 2023.
4. Council on Governmental Relations (COGR). Research Security and the Cost of Compliance: Phase I Report; COGR: Washington, DC, USA, 2022.
5. African Union. African Union Convention on Cyber Security and Personal Data Protection (Malabo Convention); African Union: Addis Ababa, Ethiopia, 2014.
6. African Union Commission. Status of Cybersecurity Legislation in Africa. *AU Digital Transformation Strategy Report*, 2024.
7. Rwanda National Cyber Security Authority (NCSA). Minimum Cybersecurity Standards for Public Institutions and Essential Service Providers; NCSA: Kigali, Rwanda, 2023.
8. Government of Rwanda. National Cybersecurity Strategy of the Republic of Rwanda 2024–2029; Government of Rwanda: Kigali, Rwanda, 2024.
9. National Institute of Standards and Technology (NIST). Security and Privacy Controls for Information Systems and Organizations. *NIST Special Publication 800-53 Rev. 5*; NIST: Gaithersburg, MD, USA, 2020.
10. Liu, Z.; Fang, Y.; Huang, C.; Xu, Y. GAXSS: Effective Payload Generation Method to Detect XSS Vulnerabilities Based on Genetic Algorithm. *Secur. Commun. Netw.* **2022**, *2022*, 2031924.
11. Kingful, F.; Ahene, E.; Appiah, B.; Frimpong, B.K. Dynamic Programming-based Adversarial Windows Payload Generator. Preprint 2023. Available online: https://www.researchgate.net/publication/372005345 (accessed on 17 April 2026).
12. Ćirković, S.; Mladenović, V.; Tomić, S.; Drljača, D.; Ristić, O. Utilizing Fine-Tuning of Large Language Models for Generating Synthetic Payloads: Enhancing Web Application Cybersecurity through Innovative Penetration Testing Techniques. *Comput. Mater. Contin.* **2025**, *82*, 4409–4430.
13. Johnson, A.; Haddad, R.J. Evading Signature-Based Antivirus Software Using Custom Reverse Shell Exploit. In *Proceedings of SoutheastCon 2021*; IEEE: Virtual, 2021. doi10.1109/SoutheastCon45413.2021.9401881
14. Roy, S.; Panaousis, E.; Noakes, C.; Laszka, A.; Panda, S.; Loukas, G. SoK: The MITRE ATT&CK Framework in Research and Practice. arXiv:2304.07411, 2023.
15. Jiang, Y.; Meng, Q.; Shang, F.; Oo, N.; Minh, L.T.H.; Lim, H.W.; Sikdar, B. MITRE ATT&CK Applications in Cybersecurity and The Way Forward. arXiv:2502.10825, 2025.
16. Serianu Ltd. Africa Cyber Security Report: Cybercrime Trends and Economic Impact. *Africa Cybersecurity Report*, 2023.
17. International Telecommunication Union (ITU). Global Cybersecurity Index 2024: African Regional Analysis; ITU: Geneva, Switzerland, 2024.
18. Sang, E.; Sang, R. A Comparative Review of Cybercrime Law in Kenya. *Commonw. Law Rev.* **2023**.
19. Orji, U.J. Cybersecurity Governance in Nigeria: A Critical Analysis. *Afr. J. Inf. Commun.* **2021**, *28*.
20. Karlsen, E.; Luo, X.; Zincir-Heywood, N.; Heywood, M. Benchmarking Large Language Models for Log Analysis, Security, and Interpretation. *J. Netw. Syst. Manag.* **2024**, *32*, 59.
21. Villarreal-Vasquez, M.; Modelo-Howard, G.; Bhargava, B.K. Hunting for Insider Threats Using LSTM-Based Anomaly Detection. *IEEE Trans. Dependable Secure Comput.* **2022**, *20*, 451–462.
22. Mehavilla, L.; Rodríguez, M.; García, J.; Alesanco, Á. Evaluating Large Language Models Effectiveness for Flow-Based Intrusion Detection: A Comparative Study with ML and DL Baselines. *Artif. Intell. Rev.* **2026**.
23. Artioli, P.; et al. SIEVE: Generating a Cybersecurity Log Dataset Collection for SIEM Event Classification. *Comput. Netw.* **2025**, *266*, 111330.
24. SecRepo. Security Data Samples Repository. Available online: https://www.secrepo.com/ (accessed on 20 March 2026).
25. Kent, A.D. User-Computer Authentication Associations in Time; Los Alamos National Laboratory: Los Alamos, NM, USA, 2014.
26. Tsoumakas, G.; Katakis, I. Multi-Label Classification: An Overview. *Int. J. Data Warehous. Min.* **2007**, *3*, 1–13.
27. Tavallaee, M.; et al. A Detailed Analysis of the KDD CUP 99 Data Set. In *Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, 2009.

 doi:10.20944/preprints202604.1589.v1

28. He, S.; et al. Loghub: A Large Collection of System Log Datasets for AI-Driven Log Analytics. arXiv:2008.06448, 2020.
29. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016; pp. 785–794.
30. Anthropic. Introducing the Model Context Protocol. *Anthropic Blog*, November 2024. Available online: https://www.anthropic.com/news/model-context-protocol (accessed on 20 March 2026).
31. Model Context Protocol. Model Context Protocol Specification, 2025. Available online: https://modelcontextprotocol.io/ (accessed on 20 March 2026).
32. Hou, X.; Zhao, Y.; Wang, S.; Wang, H. Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions. arXiv:2503.23278, 2025.
33. Errico, H.; Ngiam, J.; Sojan, S. Securing the Model Context Protocol (MCP): Risks, Controls, and Governance. arXiv:2511.20920, 2025.