| Volume 02 | Issue 06 | November 2026 | ISSN: 3139-1737

International Journal of Advances and Applications in Convergent Studies & Transdisciplinary (IJ-AACST)

Bridging Disciplines, Bridging Knowledge

Aligned with the Vision of NEP 2020 and UGC Guidelines

Website: https://aacstij.org || Email: journaleditor@aacstij.org

Sentinel LLM: An Evidence-Grounded Hybrid AI Framework for Cloud Threat Detection and Risk-Aware Incident Response

DOI: https://doi.org/ 10.5281/zenodo.24103862

S. D. Gujar[1]*, Dr. V. S. Tondre[2]

*Corresponding Author: [1]Department of Computer Science & Application, Brijlal Biyani Science College, Amravati, Maharashtra, India; affiliated to Sant Gadge Baba Amravati University (SGBAU), Amravati

santooshgujar@gmail.com

ORCID iD: https://orcid.org/0009-0006-3203-6747

[2]Department of Computer Science & Application, Brijlal Biyani Science College, Amravati, Maharashtra, India; affiliated to Sant Gadge Baba Amravati University (SGBAU), Amravati

vstondre@gmail.com

ORCID iD: https://orcid.org/0009-0001-1357-4527

ABSTRACT

Cloud platforms produce large volumes of security information, including login records, API calls, network events, permission changes, and file-access logs. Rule-based systems may miss unusual behaviour or create too many alerts. Large language models (LLMs) can explain technical events in simple language, but they may also produce unsupported statements. This paper proposes SentinelLLM, a hybrid framework that combines machine-learning anomaly detection, retrieval-augmented generation (RAG), and an LLM for cloud threat analysis. The framework collects cloud events, prepares the data, identifies unusual activity, retrieves relevant security policies, and generates an evidence-linked explanation and response recommendation. High-impact actions require policy checks and human approval. An AWS prototype is proposed using CloudTrail, GuardDuty, CloudWatch, S3, Lambda, SQS, and Amazon Bedrock. The paper defines an evaluation plan based on detection quality, evidence support, response quality, latency, and cost. Numerical examples in the result tables are explicitly illustrative planning values; they are not measured experimental results.

KEYWORDS: Cloud security, large language models, anomaly detection, retrieval-augmented generation, AWS, threat detection, incident response, explainable artificial intelligence.

1. INTRODUCTION

Cloud computing is widely used by businesses, colleges, hospitals, government offices, and online services. It provides computing power, storage, and applications without requiring every organization to maintain its own hardware. However, cloud environments create security challenges because resources, users, and permissions can

© 2026 The Author(s). This article is published by the AACST in the IJ-AACST under the Creative Commons Attribution 4.0 International (CC BY 4.0) License. The authors retain copyright and grant AACST a non-exclusive license to publish, distribute, and archive the work.

change frequently. Attackers may misuse stolen credentials, gain excessive permissions, access files without approval, or exploit weak configurations. Cloud security services collect events and identify suspicious activity, but analysts may still receive many alerts and spend time deciding which ones need attention.

Machine learning can identify behaviour that differs from known patterns, while LLMs can summarize events and explain technical information in ordinary language. SentinelLLM combines these capabilities but keeps detection, explanation, and response authorization as separate steps.

A. Problem Statement

The problem is how to turn large volumes of cloud events into reliable, understandable, and actionable security information. The system should identify suspicious behaviour, show the evidence behind an alert, explain possible impact, and recommend a safe next step.

B. Objectives

- Design a framework combining anomaly detection, RAG, and LLM-based explanation.
- Identify suspicious cloud activities using machine-learning methods.
- Link explanations to source events and trusted security guidance.
- Prioritize incidents using anomaly level, estimated impact, and confidence.
- Recommend responses while keeping high-impact actions under human control.
- Evaluate detection, explanation, response quality, latency, and cost.

2. RELATED WORK AND RESEARCH GAP

AI-based cybersecurity research includes intrusion detection, security-log analysis, vulnerability assessment, and incident response. Machine-learning models can classify known patterns or identify unusual activity, but their results depend on training data and may change when cloud behaviour changes. LLMs can summarize technical information, but they may produce statements that are not supported by the available evidence.RAG can provide an LLM with relevant documents, such as cloud security policies, incident playbooks, and attack descriptions. Retrieved documents do not guarantee correctness; the final explanation must still be checked against original event records.

This study investigates the combination of machine-learning detection, trusted knowledge retrieval, and evidence-checked LLM explanations. The table summarizes the research opportunities to be tested; it does not claim that no prior work has addressed them.

TABLE I. RESEARCH GAP AND PROPOSED APPROACH

Area	Issue	Proposed approach
Alert detection	Missed unusual events or excess alerts	Compare anomaly models with rule baseline
Alert explanation	Scores do not explain suspicious behaviour	Link explanations to event IDs
Security context	Model may lack local policies	Retrieve approved policies using RAG

Response safety	Unreviewed automation may disrupt services	Policy checks and human approval
Evaluation	Detection and explanation assessed separately	Measure detection, evidence, response, latency, cost

3. PROPOSED SENTINELLLM FRAMEWORK

SentinelLLM has six main stages: cloud event collection, data preparation, anomaly detection, knowledge retrieval, LLM explanation, and controlled response. The detector identifies unusual activity; retrieval supplies trusted context; the LLM prepares an explanation; and a policy layer controls response actions.

A. Event Collection and Preparation

For an AWS prototype, event sources include CloudTrail, GuardDuty, CloudWatch, VPC Flow Logs, and AWS Config. Events are normalized, timestamps standardized, duplicates removed, and useful features extracted. Possible features include identity, API operation, resource identifier, source address, authentication result, event time, access frequency, and permission changes. Passwords, access tokens, and unnecessary personal data must be removed.

B. Machine-Learning Detection

Isolation Forest can be tested for unsupervised anomaly detection. Random Forest or XGBoost can be tested when labelled examples are available. For event feature vector x, the detector returns A(x) between 0 and 1. A higher score indicates unusual behaviour, not proof of an attack.

C. Retrieval and LLM Explanation

The RAG component searches a controlled knowledge base containing AWS security guidance, approved policies, response playbooks, and MITRE ATT&CK descriptions. Retrieved material is provided with the relevant event fields. A retrieval expression is $Dq = TopK(R(q,K))$, where q is the event query, K is the knowledge base, R is the retrieval function, and Dq is the retrieved context.

The LLM produces a report containing the suspected activity, supporting event identifiers, possible impact, uncertainty, and recommended investigation steps. The system verifies that factual statements refer to available events or retrieved sources. Logs and retrieved documents are treated as untrusted input.

D. Risk Score and Response Control

The initial risk score is $Rs = \alpha A + \beta I + \gamma C$, where A is anomaly score, I is estimated impact, C is classification confidence, and $\alpha + \beta + \gamma = 1$. For an initial demonstration only, weights are set to $\alpha = 0.40$, $\beta = 0.35$, and $\gamma = 0.25$. These weights require validation against labelled data.

TABLE II. INITIAL RISK BANDS (DESIGN SETTINGS)

Risk score	Priority	Handling
0.00–0.49	Low	Log and monitor
0.50–0.79	Medium	Analyst review
0.80–1.00	High	Urgent review; approval before action

The thresholds are proposed design settings, not empirically optimized values. A high score triggers investigation, not automatic destructive action.

4. SYSTEM ARCHITECTURE AND IMPLEMENTATION

The prototype can use AWS managed services. The first version should collect events, store them securely, run a detector, and display evidence-backed findings. Advanced components can be added after the basic workflow is stable.

TABLE III. PROPOSED IMPLEMENTATION COMPONENTS

Layer	Technology	Purpose
Collection	CloudTrail, GuardDuty, CloudWatch	Audit events, findings, logs
Storage	Amazon S3	Event and experiment data
Queue/processing	SQS, Lambda	Asynchronous event processing
ML	Python, Scikit-learn, XGBoost	Anomaly score/classification
LLM	Amazon Bedrock or evaluated open model	Evidence-based explanation
Retrieval	FAISS or OpenSearch	Policy and threat knowledge
Workflow	Step Functions	Analysis and approval
Dashboard	React, FastAPI	Alerts, evidence, risk, status

A. Processing Steps

- Collect cloud events and security findings from approved sources.
- Normalize events and remove unnecessary sensitive fields.
- Calculate anomaly scores and correlate related events.
- Retrieve relevant security policies and threat information.
- Generate an evidence-based explanation using the LLM.
- Validate event identifiers and factual claims.
- Calculate risk priority and display the result.
- Request approval before high-impact response actions.
- Record analyst decisions and response outcomes.

5. RESEARCH METHODOLOGY

The evaluation should use public cybersecurity datasets and controlled cloud events. Public network intrusion datasets support general anomaly-detection experiments but do not directly represent every AWS audit event. AWS-specific claims should be tested with sanitized CloudTrail records or controlled events generated in a test

account.

TABLE IV. DATASET PLAN (PROPOSED EXPERIMENT DESIGN)

Dataset/Source	Planned Records	Use
CIC-IDS2017	2,000 sample flows	Network intrusion baseline
CSE-CIC-IDS2018	2,000 sample flows	Additional attack/benign evaluation
UNSW-NB15	2,000 sample records	General intrusion baseline
Controlled AWS events	1,000 events	IAM, API, S3, configuration scenarios

The sample counts above are proposed experimental targets for planning, not claims about downloaded dataset sizes or completed experiments.

TABLE V. CONTROLLED SECURITY SCENARIOS

ID	Scenario	Planned test cases
S1	Repeated failed sign-ins	100
S2	Unusual access pattern	100
S3	IAM policy change	100
S4	Unexpected privilege increase	100
S5	Unusual object access	100
S6	Security-group modification	100
S7	Unusual API activity	100

These are target test-case counts. The final paper must report actual counts and labeling procedures after the test runs.

A. Baselines and Metrics

Compare rule-based detection, Isolation Forest, a supervised classifier, LLM-only analysis, LLM with RAG, and SentinelLLM. Use precision, recall, F1-score, false-positive rate, detection latency, evidence support, response quality, and operational cost. Use a held-out test set and repeat runs where appropriate.

6. RESULTS AND ANALYSIS

The uploaded manuscript describes a proposed framework and does not contain completed experimental measurements. To satisfy the request for populated tables without presenting invented results as real, the numerical values below are clearly marked as illustrative demonstration values. They show how a results section can be formatted and interpreted. They must be replaced by actual measurements before academic or IEEE submission.

TABLE VI. ILLUSTRATIVE DETECTION RESULTS (NOT EXPERIMENTAL)

Method	Precision	Recall	F1-score	FPR
Rule-based	0.82	0.70	0.76	0.12
Isolation Forest	0.85	0.78	0.81	0.10
Random Forest	0.90	0.86	0.88	0.07
LLM-only	0.84	0.80	0.82	0.11
SentinelLLM hybrid	0.92	0.89	0.90	0.05

Illustrative interpretation: in this example only, the hybrid row has the highest F1-score (0.90) and the lowest false-positive rate (0.05). These values are not results from a running system and cannot be used as evidence of achieved performance.

TABLE VII. ILLUSTRATIVE EXPLANATION AND RESPONSE SCORES (NOT EXPERIMENTAL)

Approach	Evidence support	Explanation quality / 5	Safe recommendation rate
LLM without retrieval	0.72	3.4	0.78
LLM with RAG	0.86	4.0	0.86
SentinelLLM + validation	0.94	4.4	0.95

The scores above are hypothetical examples. For a real study, use a predefined rubric, at least two independent reviewers where possible, and report agreement between reviewers.

TABLE VIII. ILLUSTRATIVE LATENCY AND COST (NOT EXPERIMENTAL)

Stage	Latency (ms)	Example cost / 1,000 events (USD)
Preprocessing	35	0.02
ML detection	18	0.01
Knowledge retrieval	120	0.08
LLM inference	1,450	1.20
End-to-end	1,623	1.31

The cost figures are hypothetical placeholders only. Actual costs depend on AWS region, selected model, token volume, compute configuration, storage, and data transfer. Measure latency over repeated trials and report median and 95th percentile, not only a single average.

A. Statistical Reporting Standard

For final experiments, report the number of test records, class distribution, train/validation/test split, random seed, model configuration, and repeated-run count. Report mean ± standard deviation for repeated trials and include confidence intervals where suitable. Do not compare models using different test partitions.

7. SECURITY, PRIVACY, AND ETHICAL CONSIDERATIONS

Cloud logs may contain personal information and sensitive operational details. Use a test account and sanitized data wherever possible. Never send credentials or secrets to an LLM. Access to logs and response tools should follow least-privilege principles.

Logs and retrieved documents must be treated as untrusted input because malicious instructions may be placed inside event fields. The model must not execute arbitrary commands. Response tools should use allow-listed actions, strict input validation, separate IAM permissions, approval gates, and audit records. The system should fail safely when evidence is missing or the model is unavailable.

8. LIMITATIONS AND FUTURE WORK

The design depends on the quality of cloud logs, labels, and security knowledge. A model trained in one environment may not perform equally well in another. Public network datasets do not fully represent cloud control-plane events. LLM outputs may vary between models and versions, and inference can add latency and cost. Risk-score weights and thresholds require validation.

Future work can extend the prototype to multiple cloud providers, test additional LLMs, use graph-based event correlation, and study privacy-preserving methods. Agentic investigation may be explored after detection and approval stages have been evaluated.

9. CONCLUSION

This paper presents SentinelLLM, a proposed framework that combines machine learning and large language models to support cloud security operations. The design brings together cloud-event collection, anomaly detection, retrieval of trusted security information, evidence-based explanation, risk prioritization, and controlled incident response. Machine learning identifies unusual patterns, retrieval supplies relevant context, and the LLM explains the available evidence in language that an analyst can understand.

The framework does not treat an LLM as the final authority for security decisions. Explanations must be checked against source records, and high-impact actions must pass policy checks and human approval. The populated numerical tables in this draft are illustrative examples only; they are not measured findings. The next research step is to implement the AWS prototype, run repeatable experiments, replace every illustrative value with measured results, and compare the framework with rule-based, machine-learning, and LLM-based baselines. Only reproducible measurements can establish whether SentinelLLM improves detection and response without

weakening security, privacy, or operational control.

10. REFERENCES

1. C. Autio et al., “Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile,” NIST AI 600-1, National Institute of Standards and Technology, 2024, doi: 10.6028/NIST.AI.600-1.
2. MITRE, “MITRE ATT&CK,” [Online]. Available: https://attack.mitre.org/. Accessed: Oct. 30, 2026.
3. Amazon Web Services, “AWS CloudTrail User Guide,” [Online]. Available: https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-user-guide.html. Accessed: Oct. 30, 2026.
4. Amazon Web Services, “Amazon GuardDuty User Guide,” [Online]. Available: https://docs.aws.amazon.com/guardduty/latest/ug/what-is-guardduty.html. Accessed: Oct. 30, 2026.
5. Amazon Web Services, “Amazon Bedrock User Guide,” [Online]. Available: https://docs.aws.amazon.com/bedrock/latest/userguide/what-is-bedrock.html. Accessed: Oct. 30, 2026.
6. M. Ring et al., “A survey of network-based intrusion detection data sets,” Computers & Security, vol. 86, pp. 147–167, 2019.
7. N. Moustafa and J. Slay, “UNSW-NB15: A comprehensive data set for network intrusion detection systems,” in Proc. MilCIS, 2015.
8. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in Proc. ICISSP, 2018, pp. 108–116.