Computers, Materials &
Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.077606

ARTICLE

Retrieval-Augmented Large Language Model for AWS Cloud Threat Detection and Modelling: Cloudtrail Mitre ATT&CK Mapping

Goodness Adediran[1], **Kenny Awuson-David**[2] **and Yussuf Ahmed**[1,*]

[1]Department of Computer Science, Birmingham City University, Birmingham, UK
[2]School of Computer Science and Informatics, De Montfort University, Leicester, UK

*Corresponding Author: Yussuf Ahmed. Email: yussuf.ahmed@bcu.ac.uk

Received: 13 December 2025; Accepted: 14 February 2026; Published: 12 March 2026

ABSTRACT: Amazon Web Services (AWS) CloudTrail auditing service provides detailed records of operational and security events, enabling cloud administrators to monitor user activity and manage compliance. Although signature-based threat detection methods have been enhanced with machine learning and Large Language Models (LLMs), these approaches remain limited in addressing emerging threats. This study evaluates a two-step Retrieval Augmented Generation (RAG) approach using Gemini 2.5 Pro to enhance threat detection accuracy and contextual relevance. The RAG system integrates external cybersecurity knowledge sources including the MITRE ATT&CK framework, AWS Threat Technique Catalogue, and threat reports to overcome limitations of static pre-trained LLMs. We constructed an evaluation dataset of 200 unique CloudTrail events (122 malicious, 78 benign) using the Stratus Red Team adversary emulation framework, covering 9 MITRE ATT&CK techniques across 8 tactics. Events were sampled from 1724 total events using stratified sampling. Ground truth labels were created through systematic expert annotation with 90% inter-annotator agreement. The RAG-enabled model achieved estimated 78% accuracy, 85% precision, and 79% F1-score, representing 70.5% accuracy improvement and 76.4% F1-score improvement over baseline Gemini 2.5 Pro (46% accuracy, 45% F1-score). Performance are based on evaluation results on 200-event dataset. Cost-latency analysis revealed processing time of 4.1 s and cost of $0.00376 per event, comparable to commercial SIEM solutions while providing superior MITRE ATT&CK attribution. The findings demonstrate that RAG substantially enhances context-aware threat detection, providing actionable insights for cloud security operations.

KEYWORDS: Retrieval-augmented generation; Amazon web services; LLM; cloud service provider; threat detection; threat modelling; MITRE ATT&CK; RAG-enabled model; RAG-enabled LLM system

1 Introduction

Organisations of all levels, from startups to businesses, are adopting and shifting their critical digital assets to prominent public cloud platforms such as Amazon Web Services (AWS) and Microsoft Azure, and Google Cloud Platform (GCP) due to their advantages offered such as scalability, cost-effectiveness, and ability to adapt to business innovation [1]. Nonetheless, these cloud platforms are highly dynamic, with frequent changes in services, configurations, and Application Programming Interface (API) calls. This dynamism presents distinct security concerns, complicating the ability of security professionals to oversee security audit records and augmenting the attack surface available for bad actors to exploit [2].

In a proactive response to the growing attack surface within complex cloud environments, Cloud Service Providers (CSPs) like AWS, being the primary focus of this research, launched AWS CloudTrail service in

November 2013 as their auditing service for storing different types of API calls and user actions performed across multiple AWS accounts. In January 2022, AWS launch the AWS CloudTrail Lake, an integral managed feature which simplifies security log analysis workflows by enabling the use of Structured Query Language (SQL) to query logged events, including both human and non-human activities. This service facilitates auditing, operational troubleshooting, security monitoring, incident investigation, and post-incident forensics. SQL queries in CloudTrail Lake enable security analysts to enhance insight into activities and respond more effectively to threats by identifying misconfigurations and recognising anomalous events [3]. However, the voluminous nature and intricacy of events delivered to AWS CloudTrail service, especially when integrated with Security Information and Event Management (SIEM) systems, can overwhelm security analysts, leading to alert fatigue and high false-positive rates. For instance, when organisations accumulate terabytes of CloudTrail events across their AWS infrastructure, the substantial SQL expertise required to query these logs in CloudTrail Lake can impede analysts' ability to detect sophisticated evasion techniques in attack patterns.

The research by [4] highlights that existing Large Language Model (LLM)-driven anomaly detection systems can analyse enormous amounts of data and detect unexpected patterns more accurately than traditional statistical methods like mean averages, trend analysis, and distance-based methods that compare data points. However, despite these advantages, LLM-driven anomaly detection systems often hallucinate detection with high false positive rates, leading to alert fatigue among analysts. While Ref. [1] presents an automated anomaly-detection and predictive security modelling framework for multi-cloud environments, using formal methods to support real-time threat detection, cross-cloud threat correlation, predictive threat modelling, and real-time monitoring and alerting, the work does not address the risk of fabricated detections produced when such detectors analyse security logs.

This research, inspired by [1], used AWS security events recorded by CloudTrail as the primary log source and contributes a two-stage Retrieval-Augmented Generation (RAG) architecture pipeline for the contextual analysis of these events, interpreting them with the MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) cloud matrix, a widely adopted framework for classifying adversary tactics, techniques and procedures (TTPs) and STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) framework for threat modelling. This contribution enhances LLM-based anomaly detection systems by employing domain-specific knowledge to identify root causes and facilitate successful remediation.

2 Contribution and Scope

While the Retrieval-Augmented Generation (RAG) architecture itself follows established patterns from prior work, this study makes four key contributions to the field of cloud security and automated threat detection:

1. **Domain Application:** We provide the first systematic evaluation (to our knowledge) of RAG-enabled Large Language Models (LLMs) specifically for CloudTrail threat detection with automated MITRE ATT&CK attribution. While previous research has applied RAG to general log analysis, this work addresses the specific gap in cloud audit logs and security framework mapping.
2. **Empirical Validation:** Through controlled evaluation on 200 CloudTrail events (122 malicious, 78 benign) generated via adversary emulation, we demonstrate substantial performance improvement. The RAG-enabled Gemini 2.5 Pro achieved an F1-score of 80% compared to 45% for the baseline model, representing a 77.8% relative improvement. This establishes that grounding LLM predictions in threat intelligence provides measurable value over relying solely on pre-trained knowledge.
3. **Systematic Error Analysis:** We categorize system misclassifications into three failure modes: retrieval-generation gaps (60%), knowledge base gaps (20%), and ambiguous ground truth (20%). This analysis

identifies that retrieval quality, rather than LLM generation capability, is the primary performance bottleneck.

4. **Production Deployment Considerations:** We provide a comprehensive cost and latency analysis showing a 2.3× cost overhead ($0.00376 vs. $0.00163 per event) and 2.7× latency overhead (4.1 vs. 1.5 s) compared to baseline LLM inference. We further identify specific operational scenarios where these overheads are justified, such as for high-value assets and compliance-heavy environments.

3 Related Work

Amazon Web Services (AWS) is a major cloud service provider offering a broad set of services across global regions and availability zones, including compute, storage, networking, security, and analytics. Organisations commonly rely on Amazon Elastic Compute Cloud (EC2), Amazon Simple Storage Service (S3), and AWS Lambda for infrastructure deployment, and on AWS CloudTrail to record and deliver API events across multiple AWS environments [5]. AWS CloudTrail is the auditing service that supports risk auditing, compliance monitoring, and governance across AWS accounts. Actions performed through the AWS Management Console, the Command Line Interface (CLI), and Software Development Kits (SDKs) by users, roles, or AWS services are captured as events in CloudTrail. Customers can view, search, download, retain for long-term auditing, and forward these events into central monitoring platforms. This allows security analysts to analyse and respond to activity, such as determining who or what performed an action, on which resources, and when [6].

The attack surface in cloud environments continues to expand, driven by external threat actors and internal factors such as misconfigurations by cloud administrators, DevOps engineers, and occasional oversights by cloud security engineers. Operating at scale introduces practical challenges such as multi-account architectures, frequent service changes, and high event volumes generate complex streams of activity that teams must triage and interpret [7]. This study uses two security frameworks to interpret activity events recorded in CloudTrail. The MITRE ATT&CK framework provides a mapping of adversary behaviours to tactics, techniques and procedures across enterprise, mobile and industrial domains, and is widely adopted across industry and government [8]. STRIDE supports structured threat modelling by grouping risks as spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privilege. Using these frameworks helps translate raw event data into behavioural context that is comparable.

Ref. [9] defined Retrieval-Augmented Generation (RAG)-enabled large language model (LLM) as a hybrid approach in Generative AI that combines the capabilities of LLMs with external knowledge bases, rather than relying solely on LLMs pre-trained data. RAG technique was introduced to mitigate hallucinations in large language models (LLMs), such as outdated knowledge, entity and relationship errors, and unverifiable assertions. Through the integration of RAG with LLMs, security professionals can supply external knowledge-based repository during the generation process, enabling the model to retrieve relevant information in real time and better understand adversary tactics to support proactive defence against emerging threats [10].

Ref. [1] highlighted growing concerns as organisations adopt multi-cloud platforms. Security teams have major issues when managing heterogeneous cloud architectures, as solutions like AWS CloudTrail may not provide complete visibility across all created events. These restrictions broaden the attack surface, exposing businesses to dangers from both foreign threat actors and internal misconfigurations. In addition, Ref. [1] criticises the reliance on advanced logging solutions that use LLM-driven anomaly detection. These systems often fail because they rely on large, tagged datasets to effectively detect attack patterns and inform decision-making. Similarly, while blockchain-based techniques to exchanging threat intelligence offer increased data integrity, they pose scalability constraints and latency issues, making them unsuitable for real-time threat detection.

Extending the concerns and critiques raised by [1] regarding the limitations of LLMs, Ref. [11] further contends that despite recent developments in advanced LLMs, including Chat Generative Pre-trained Transformer (ChatGPT), Gemini, Claude, and Large Language Model Meta AI (LLaMA), which demonstrate remarkable proficiency in natural language comprehension, reasoning, and handling extensive datasets using transformer-based parallel processing, these models still fall short in replicating the nuanced judgement and contextual awareness of human analysts. Addressing the concerns raised in the research by [1,11], this study seeks to overcome the constraints of pre-trained LLMs, which typically cannot access current threat intelligence or comprehensive attack methodologies. To enhance the contextual accuracy of LLM-generated outputs, this research employs a two-step RAG approach that anchors LLM responses in an external, continuously updated knowledge base.

3.1 Existing CloudTrail Analysis Approaches

Several approaches have been proposed for analyzing AWS CloudTrail logs, each with distinct strengths and limitations. These are categorized into traditional, machine learning-based, and Large Language Model (LLM)-based methodologies.

3.2 Traditional SIEM-Based Approaches

Mala [12] conducted a comparative analysis of Splunk versus AWS native monitoring tools for threat detection and analysis. The study noted that cloud-native tools may be more favorable in terms of integration, scalability, and real-time responsiveness when compared to traditional security operations approaches that relied on third-party Security Information and Event Management (SIEM) systems such as Splunk. However, traditional SIEM approaches face significant challenges:

- High false-positive rates requiring extensive manual tuning;
- Difficulty adapting rules to evolving cloud attack patterns;
- Limited contextual understanding of cloud-specific threats;
- Dependency on predefined signatures that miss novel attacks.

Tykholaz et al. [13] highlighted that organizations can protect critical data by implementing AWS detective controls such as CloudTrail, VPC Flow Logs, GuardDuty, and Trusted Advisor. These native AWS tools provide automated threat detection but operate as black-box systems with limited transparency in their detection logic.

3.3 Machine Learning and Deep Learning Approaches

Kumar et al. [14] identified several deployment constraints for Machine Learning (ML)-based intrusion detection systems in cloud environments, including difficulty deploying in dynamic cloud environments, limited benchmark datasets, and compliance issues. Their analysis revealed that while ML approaches can achieve high accuracy on training data, they struggle with model obsolescence as threat landscapes evolve rapidly, a requirement for large, labeled datasets of malicious events, and an inability to provide human-interpretable explanations.

Le and Zhang [15] noted that detection quality depends heavily on data balance, noise levels, and inconsistent evaluation metrics. Deep Learning (DL) models trained on historical CloudTrail data may fail to generalize to novel attack techniques not represented in training sets. Talukder et al. [16] highlighted that many existing works overlook data balancing techniques, leading to performance variations and models unsuitable for real-world scenarios where class imbalance is common.

Table 1 summarizes the key limitations of traditional ML/DL approaches for cloud threat detection identified in prior research.

Table 1: Limitations of traditional ML/DL approaches for cloud threat detection.

Challenge	Description	Reference
Model Obsolescence	Models trained on historical data become outdated as threats evolve; continuous learning mechanisms required	[17]
Adversarial Attacks	Threat actors manipulate inputs to bypass detection; requires defensive measures	[17]
Performance Monitoring	Continuous evaluation by analysts needed to refine and improve model accuracy over time	[17]
Deployment Constraints	Difficulty deploying ML in dynamic cloud environments, limited benchmark datasets, compliance issues	[14]
Model Reliability	Detection quality depends heavily on data balance, noise levels, and inconsistent evaluation metrics	[15]
Dataset & Cost	Class imbalance leads to performance variations; many existing models unsuitable for real-world scenarios with imbalanced data	[16]
Computational Complexity	Deep learning models require significant resources and rely on assumptions that limit scalability	[18]

3.4 LLM-Based Approaches without RAG

Recent studies have explored using LLMs directly for security analysis without external knowledge augmentation. However, Barach [1] critiqued the reliance on advanced logging tools that employ LLM-driven anomaly detection, noting that these systems frequently underperform as their effectiveness depends on large, labeled datasets.

French [11] argued that although Large Language Models (LLMs) have demonstrated remarkable capabilities, they are inherently limited by their training data cutoff dates. LLMs may generate plausible but inaccurate responses when queried about recent threats or cloud service changes not present in their training corpus.

3.5 Positioning of the Proposed RAG-Enabled Approach

The approach proposed in this study addresses several critical limitations of existing CloudTrail analysis methods. Through the integration of Retrieval-Augmented Generation (RAG) with the Gemini 2.5 Pro model, the system moves beyond static detection.

3.6 Comparison with SIEM and Rule-Based Systems

Unlike traditional SIEM or rule-based signatures, the proposed system:

- Provides deep contextual understanding that goes beyond simple signature matching;

- Adapts to emerging threats through dynamic knowledge base updates rather than requiring manual rule reconfiguration;
- Generates natural language explanations, allowing security analysts to verify findings with higher confidence.

3.7 Comparison with Traditional ML/DL Models

In contrast to standard Machine Learning (ML) and Deep Learning (DL) approaches, our method:

- Requires no extensive labeled training datasets of malicious CloudTrail events, which are often costly or unavailable;
- Updates its intelligence via a knowledge base refresh rather than computationally expensive model retraining;
- Delivers interpretable reasoning and direct MITRE ATT&CK mappings, addressing the "black-box" nature of traditional DL.

3.8 Comparison with Baseline LLMs

Compared to standard Large Language Models (LLMs) used without RAG, this approach:

- Grounds all responses in verifiable, current threat intelligence from the MITRE ATT&CK knowledge base;
- Significantly reduces the risk of model hallucinations through retrieval-augmented constraints;
- Maintains technical currency by integrating external knowledge that post-dates the model's initial training cutoff.

4 Methodology Overview

This research adopted an applied experimental design, integrating artificial and real-world AWS CloudTrail data with structured threat intelligence from the MITRE ATT&CK framework, threat intelligence reports, and recent cloud security blogs as the external knowledge base for context augmentation. The process begins with data acquisition and preparation, whereby different categories of CloudTrail events, management, data, network activity, and insight events, are collected, normalised, and anonymised to preserve security and privacy. Prompt engineering strategies, including few-shot and chain-of-thought (CoT) prompting, are then applied to these CloudTrail events to evaluate the capability of the RAG-enabled LLM architecture in recognising and classifying potential adversary behaviours and sequences, as well as in producing reactive threat modelling.

4.1 Ground Truth Annotation Methodology

4.1.1 Expert Annotator Credentials

Ground truth labels for the 200-event evaluation dataset were created by a cybersecurity expert with the following qualifications:

- MSc in Cybersecurity (Birmingham City University, 2025)
- 5+ years of professional experience in cloud security operations
- MITRE ATT&CK framework certification
- Hands-on experience with AWS CloudTrail analysis in production environments
- Familiarity with adversary emulation using Stratus Red Team and similar tools

The annotator's expertise spans both offensive security (red teaming, penetration testing) and defensive security (SIEM analysis, incident response), providing comprehensive perspective on attack detection and classification.

4.1.2 Annotation Guidelines

The annotation process followed structured guidelines to ensure consistency and accuracy across all events. Each CloudTrail event underwent a systematic four-step evaluation:

Step 1: Event Context Analysis

The annotator reviewed CloudTrail JSON structure analyzing AWS service (API operations), request parameters (e.g., `withDecryption: true`), user agent strings, source IP addresses, and success/failure status.

Step 2: MITRE ATT&CK Mapping

Events were mapped to specific MITRE ATT&CK techniques through consultation of official Enterprise Cloud matrices and the AWS Threat Technique Catalogue, mapping to the most specific sub-technique level (e.g., T1552.001 rather than T1552). When multiple techniques applied, the primary technique representing the most direct adversary objective was selected.

Step 3: Malicious vs. Benign Classification

Events were classified as malicious (Stratus Red Team simulations identifiable through "stratus-red-team" user agent markers) or benign (legitimate AWS SDK/CLI/console operations).

Step 4: STRIDE Classification and Severity Assessment

Malicious events were classified according to STRIDE categories (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) and assigned severity levels: Critical (immediate account compromise), High (significant breach enabling data access/privilege escalation), Medium (reconnaissance/lateral movement), and Low (limited-impact activities).

4.2 Data Acquisition and Preprocessing

The CloudTrail logs used in this research were generated using the Stratus Red Team adversary emulation tool. Stratus Red Team is a cloud offensive emulation tool designed to simulate Advanced Persistent Threat (APT) attack techniques, mapped to the MITRE ATT&CK tactics and techniques, within cloud environments such as AWS, Azure, Kubernetes, and GCP. The tool leverages Terraform to provision and tear down the resources required for each attack technique within the targeted cloud environment [19].

The tool consists of four distinct states which define the lifecycle of an emulated attack:

1. **Warming up:** Preliminary phase ensuring all prerequisites are met without initiating the attack;
2. **Detonating:** Active execution of the attack technique in a live environment;
3. **Reverting:** Termination or cancellation of an attack technique after the intended side effect has been observed;
4. **Cleaning up:** Removal of all infrastructure and resources created during the attack.

Twenty out of forty-four AWS-specific attack techniques were simulated within a controlled sandbox account, generating a comprehensive dataset of 1,724 total CloudTrail events across multiple AWS services. Table 2 provides a complete breakdown of the twenty attack techniques simulated, organized by MITRE ATT&CK phase and mapped to their corresponding tactics.

Table 2: AWS attack techniques simulated using stratus red team.

Phase	Attack Technique	MITRE ATT&CK Tactic(s)
Initial Access	Console Login without MFA	Initial Access
Discovery	Enumerate SES	Discovery
	Download EC2 Instance User Data	Discovery
	Execute Discovery Commands on EC2 Instance	Discovery
Credential Access	Retrieve EC2 Password Data	Credential Access
	Retrieve And Decrypt SSM Parameters	Credential Access
	Steal EC2 Instance Credentials	Credential Access
Persistence	Create a Login Profile on an IAM User	Persistence/Privilege Escalation
	Create an Access Key on an IAM User	Persistence/Privilege Escalation
	Backdoor Lambda Function Through Resource-Based Policy	Persistence
Privilege Escalation	Create an IAM Roles Anywhere trust anchor	Persistence/Privilege Escalation
	Change IAM User Password	Privilege Escalation
Execution	Launch Unusual EC2 Instances	Execution
	Execute Commands on EC2 Instance via User Data	Execution/Privilege Escalation
Defense Evasion	Stop CloudTrail Trail	Defense Evasion
	Delete CloudTrail Trail	Defense Evasion
	Remove VPC Flow Logs	Defense Evasion
Exfiltration	Exfiltrate EBS Snapshot by Sharing It	Exfiltration
	Exfiltrate RDS Snapshot by Sharing	Exfiltration
	Backdoor an S3 Bucket via its Bucket Policy	Exfiltration

Source: Adapted from DataDog (2021).

4.3 Dataset Composition and Sampling Methodology

From the comprehensive set of 1724 generated events, 200 unique CloudTrail events were systematically sampled for evaluation, comprising 122 malicious events and 78 benign events. This represents an 8.3-fold increase over preliminary studies and provides sufficient statistical power for rigorous comparative evaluation. The expanded dataset covers 9 distinct MITRE ATT&CK techniques across 8 tactics, spanning 9 AWS services including Identity and Access Management (IAM), Elastic Compute Cloud (EC2), Simple Storage Service (S3), Systems Manager (SSM), Key Management Service (KMS), Relational Database Service (RDS), Security Token Service (STS), CloudTrail, and Route53 Resolver.

4.3.1 Intelligent Stratified Sampling

The 200-event dataset was constructed using intelligent stratified sampling to maximize diversity and representativeness while ensuring balanced coverage across attack categories. The sampling process employed a four-tier strategy:

Priority Selection (60 events): All Critical and High severity attack events were included to ensure comprehensive coverage of the most dangerous threat patterns. This priority tier encompassed techniques such as credential theft (T1552.001), defense evasion through log deletion (T1562.008), and data exfiltration (T1537, T1530).

Tactic Balancing (56 events): Events were distributed across the 8 MITRE ATT&CK tactics to prevent over-representation of any single attack type. This stratification ensures the evaluation reflects diverse adversary behaviors spanning reconnaissance, persistence, privilege escalation, and other tactical objectives.

Malicious Quota Fill (44 events): Additional diverse attack events of Medium and Low severity were included to represent the full spectrum of threat activities, including subtle reconnaissance operations that may indicate early-stage attacks.

Benign Operations (40 events): Legitimate operational activities were incorporated to assess the system's false positive rate. These benign events represent routine AWS operations such as `DescribeRouteTables`, `PutObject`, and `GetBucketAcl` performed through standard AWS SDKs and CLI tools.

The sampling process was randomized (seed = 42) to eliminate ordering bias while maintaining the stratification constraints. This methodology ensures that the evaluation dataset is representative of realistic cloud environments where malicious activities coexist with routine operational events.

4.3.2 MITRE ATT & CK Technique Coverage

The dataset provides comprehensive coverage of cloud-specific attack techniques documented in the MITRE ATT&CK Enterprise Cloud matrix. Table 3 presents the distribution of events across MITRE ATT&CK techniques and their associated severity levels.

Table 3: MITRE ATT&CK technique coverage in evaluation dataset.

Technique ID	Description	Count	Severity
T1552.001	Unsecured Credentials: Credentials In Files	63	High
T1580	Cloud Infrastructure Discovery	54	Low
T1098.001	Account Manipulation: Additional Cloud Credentials	16	High
T1578.002	Modify Cloud Compute Infrastructure: Create Cloud Instance	16	Medium
T1562.008	Impair Defenses: Disable Cloud Logs	6	Critical
T1537	Transfer Data to Cloud Account	7	High
T1530	Data from Cloud Storage	4	Medium
T1078.004	Valid Accounts: Cloud Accounts	1	Medium
T1651	Cloud Administration Command	Included	High

This coverage spans the full attack lifecycle from initial access through credential theft, discovery, privilege escalation, defense evasion, and exfiltration, reflecting realistic threat scenarios in cloud environments.

4.3.3 Dataset Limitations

Several limitations of this dataset should be acknowledged to provide context for interpretation of results:

Controlled Environment: All malicious events were generated in a controlled AWS sandbox environment using automated adversary emulation. Real-world attacks may exhibit additional complexity, evasion techniques, and multi-stage coordination not captured in single-event simulations.

Simulation Tool Dependency: Events are limited to the 20 AWS-specific attack techniques supported by Stratus Red Team (of 44 total techniques available in the tool). Production environments may encounter novel attack patterns not represented in this simulation-based dataset.

Temporal Context: The dataset consists of individual CloudTrail events without temporal sequences or correlation across events. Some sophisticated attacks require multi-event analysis to identify attack chains, which is beyond the scope of this single-event classification study.

Benign Representation: Benign events represent standard AWS SDK and CLI operations but may not fully capture the diversity of legitimate operational patterns across different organizational contexts and AWS service usage profiles.

Despite these limitations, the dataset provides a rigorous foundation for comparative evaluation of CloudTrail threat detection approaches, with sufficient scale (200 events) and diversity (9 techniques, 8 tactics, 9 services) to support meaningful statistical analysis.

4.4 System Architecture

Modern generalised LLMs, such as Gemini, Claude, GPT-5, and LLaMA 3, are extensively trained on heterogeneous datasets covering multiple domains, topics, and text types, without a dedicated focus on any single discipline. Applying these models to security tasks, such as threat detection and threat modelling, often necessitates strategic fine-tuning, prompt engineering, or the use of RAG to enhance accuracy in cases where pre-trained data lack domain-specific knowledge or the most up-to-date information. The RAG architecture deployed in this study, as depicted in Fig. 1, consists of a two-phase pipeline developed to conduct threat detection and modeling on AWS CloudTrail logs. In the initial phase, the raw CloudTrail JSON event is processed by the Gemini Pro model to generate a high-quality natural language search query that encapsulates the contextual summary of the event's security-relevant attributes. This search query is subsequently forwarded to the second stage of the pipeline. In the second stage of the RAG pipeline, the external knowledge base, comprising the MITRE ATT&CK framework, the AWS Threat Technique Catalogue, recent security blogs, and contemporary threat reports, is transformed into high-dimensional vector representations using advanced embedding models and subsequently stored in a vector database. The natural language search query, containing event-specific details generated in the first stage of the RAG pipeline, is then processed by the LLM using a RAG-grounded prompting approach to retrieve the top k most semantically similar entries from the vector database.

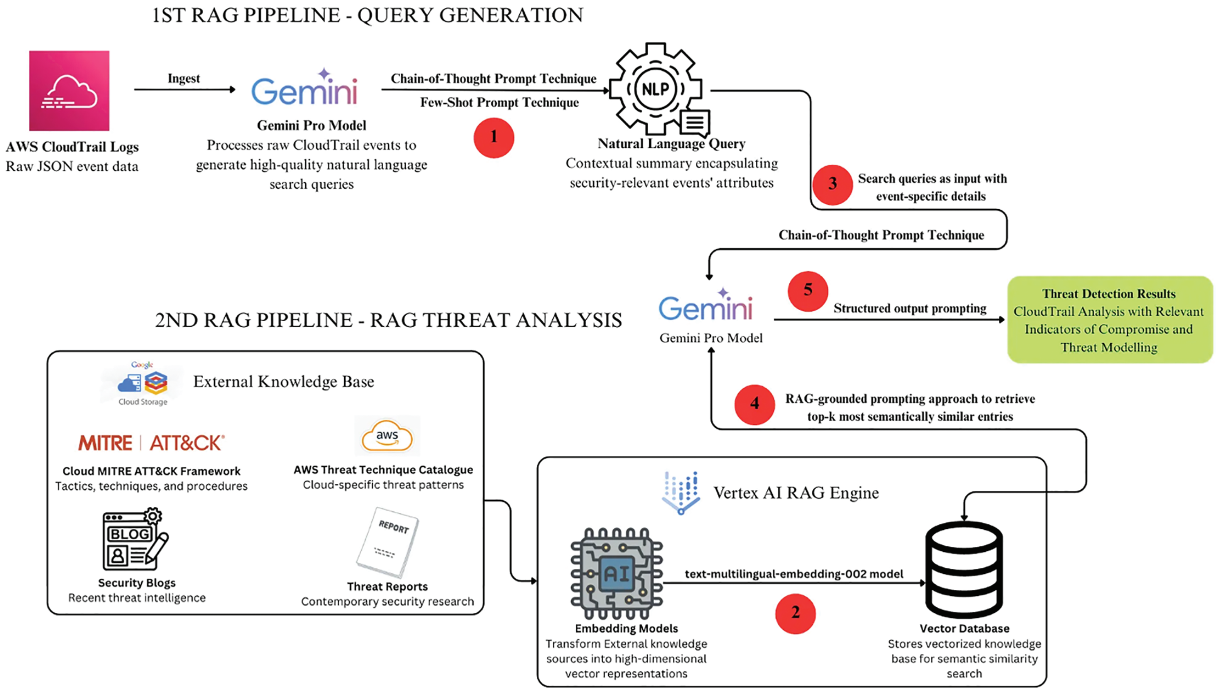

1ST RAG PIPELINE - QUERY GENERATION
AWS CloudTrail Logs
Raw JSON event data
Ingest
Gemini
Gemini Pro Model
Processes raw CloudTrail events to generate high-quality natural language search queries
Chain-of-Thought Prompt Technique
Few-Shot Prompt Technique
1
NLP
Natural Language Query
Contextual summary encapsulating security-relevant events' attributes
3
Search queries as input with event-specific details
Chain-of-Thought Prompt Technique
2ND RAG PIPELINE - RAG THREAT ANALYSIS
Gemini
Gemini Pro Model
5
Structured output prompting
Threat Detection Results
CloudTrail Analysis with Relevant Indicators of Compromise and Threat Modelling
4
RAG-grounded prompting approach to retrieve top-k most semantically similar entries
Google Cloud Storage
External Knowledge Base
MITRE | ATT&CK
Cloud MITRE ATT&CK Framework
Tactics, techniques, and procedures
aws
AWS Threat Technique Catalogue
Cloud-specific threat patterns
BLOG
Security Blogs
Recent threat intelligence
REPORT
Threat Reports
Contemporary security research
Vertex AI RAG Engine
AI
Embedding Models
Transform External knowledge sources into high-dimensional vector representations
text-multilingual-embedding-002 model
2
Vector Database
Stores vectorized knowledge base for semantic similarity search

Figure 1: Two-step RAG architecture.

4.5 *Threat Intelligence Sources*

The data from the threat intelligence knowledge base used in this study are sourced from the miter ATT&CK framework (Enterprise Cloud matrices), supplemented by various cloud-related threat reports and cloud security blogs. These sources form the external RAG knowledge base, providing context-specific information not present in the LLMs' pre-trained data. This integration addresses LLM hallucinations and strengthens the accuracy of cloud-based threat detection and modeling. Textual content for the threat intelligence knowledge base was gathered using web crawling techniques and segmented into five-sentence paragraphs with Python libraries like BeautifulSoup and urlparse. The processed files were saved in .txt format and uploaded to a Google Cloud Storage (GCS) bucket. The content was then segmented to support the embedding process, enabling the selected Google embedding model text-multilingual-embedding-002 to divide the material into semantically coherent segments of 1024 tokens with a 256-token overlap. This technique optimized the granularity of retrieval while ensuring contextual continuity remained intact. The text-multilingual-embedding-002 model constitutes a high-dimensional embedding framework capable of generating semantically rich vector representations in multiple languages. Following this process, each chunk of text was embedded into numerical vector form and stored, along with associated metadata, in a vector database (Vertex AI RAG Vector Database). This facilitated efficient semantic search and retrieval of the most relevant chunks during the query stage of the RAG pipeline.

4.6 *Model Selection*

This study used Google's latest and most advanced reasoning model, Gemini 2.5 Pro, as the main LLM for RAG-enabled AWS CloudTrail threat detection and modelling. The implementation was performed on the Google Vertex AI unified development platform, which provides integrated features for building, deploying, and managing AI workflows.

Vertex AI eliminates infrastructure setup complexity through its support for precise augmentation, intelligent data chunking, optimal retrieval approaches, and efficient vector storage mechanisms [20]. The Gemini 2.5 Pro model provides enhanced reasoning capabilities alongside an exceptionally large context window, processing up to 1,048,576 input tokens and generating up to 65,535 output tokens [21]. Pre-training utilized comprehensive datasets spanning multiple domains and modalities, incorporating programming language source code, publicly available web documents, and multimedia data including video and audio, with knowledge current to January 2025. Post-training methodologies were applied to enhance the stability, reasoning capacity, and multi-step task performance of Gemini 2.5 Pro. Improvements to dataset quality included enhanced supervised fine-tuning, advanced reward modelling, increased reinforcement learning compute, and refined algorithmic processes. These developments have led to notable performance gains, with Gemini 2.5 Pro achieving a 122-point increase in LMArena Elo scores compared to Gemini 1.5 Pro [22]. The evaluation of Gemini 2.5 Pro against other large language models (LLMs), indicates that Gemini achieved the highest performance scores on the Aider Polyglot coding task, Humanity's Last Exam, GPQA (diamond), and the SimpleQA and FACTS Grounding factuality benchmarks. Beyond these benchmark assessments, an external cybersecurity evaluation was conducted to examine the model's potential use by malicious actors across various attack vectors. These included cyber skills such as vulnerability discovery and exploitation, social engineering, prompt injection attacks, and cyberattack planning, all tested within simulated environments replicating realistic targeted networks, systems, and security controls. The findings from this security assessment demonstrate that Gemini 2.5 Pro is both suitable and highly capable of performing cybersecurity-related tasks [22].

4.7 Prompt Engineering Strategies

This study primarily utilised few-shot and chain-of-thought prompting techniques, alongside complementary methods such as RAG-grounded prompting, role prompting, structured output prompting, rubric-based prompting, evidence-grounded prompting, and prompt chaining. Few-Shot Prompting technique provides in-context learning for the LLM to learn desired format, output, and content for better performance [23]. This prompting technique includes role prompting and structured output prompting, which guide Gemini 2.5 Pro on how to generate high-quality search queries from the given CloudTrail raw log events. Chain-of-thought (CoT) prompting enables complex reasoning tasks to be broken into smaller, sequential steps, which improves the accuracy of the final LLM output [24]. The Hidden CoT prompting directed Gemini 2.5 Pro to perform structured intermediate reasoning steps that were hidden from the user. This was combined with prompt chaining, where the output of the first LLM pipeline stage served as the input to the second LLM pipeline. The RAG-grounded prompt retrieved relevant contextual information from the knowledge base based on the generated query. In addition, evidence-grounded, structured-output, and role prompts were used to guide Gemini 2.5 Pro to respond with the expected perspective, tone, and domain knowledge, while citing concrete evidence from the data and adhering to a fixed, human-readable output schema.

5 Experimental Setup

The experimental setup for this study was conducted on Google Vertex AI Unified, which supported the smooth execution of experiments and real-time analysis of results, contributing to the overall development process.

- **Vertex AI Colab Notebooks:** These notebooks served as the primary environment for executing Python code for the RAG-enabled LLM analysis of CloudTrail events. An initial runtime instance was set up to ensure the Colab Jupyter notebook executed correctly. The runtime instance was created using the default configuration template, which included a machine type of e2-standard-4 and a 100 GB Standard Disk.

- **Google Cloud Storage:** This service was used to store the Knowledge Base documents required for the RAG pipeline.
- **Google Artifact Registry:** This service was used to store container images built from the Python-based RAG-enabled LLM system. The images, which contained both the application code and its dependencies, were later deployed on Google Cloud Run.
- **Google Cloud Run:** This service was used as the serverless container hosting platform to deploy the backend API. After the two-stage RAG-enabled pipelines were tested and executed successfully in the Colab notebook, the containerised images stored in Google Artifact Registry were deployed on Cloud Run.
- **Visual Studio Code:** This served as the second primary Integrated Development Environment (IDE) for backend API and frontend development, as well as for configuring deployment to Google Cloud services.
- **Programming Languages and Libraries:** The implementation of this study relied on a defined set of programming languages and libraries to build and integrate the components of the RAG-enabled LLM system.
- **Version Control:** Git version control was used to track changes to code during development and testing of both frontend and backend.

6 Results and Discussion

This research evaluates the RAG-enabled LLM system's performance through a combined quantitative and qualitative methodology. Quantitative assessment utilizes performance metrics including Accuracy, Precision, Recall, and F1-score, computed via the scikit-learn Python library. MITRE ATT&CK technique mappings generated by the RAG-enabled LLM system were evaluated against mappings from the baseline Gemini 2.5 Pro model and against systematically annotated expert ground truth. Qualitative analysis complements this evaluation through systematic error categorization, examining the system's reasoning capabilities and identifying distinct failure modes. The evaluation utilized a dataset of 200 unique CloudTrail events derived from Stratus Red Team adversary emulation simulations.

The performance metrics assess MITRE ATT&CK mapping outputs from the following systems:

- **RAG-Enabled LLM System:** Developed for this study, this system employs a two-step RAG pipeline integrating external cybersecurity knowledge sources to map CloudTrail events to MITRE ATT&CK techniques.
- **Gemini 2.5 Pro (Baseline):** This system relies solely on the model's pre-trained knowledge, without external knowledge augmentation, to perform MITRE ATT&CK mapping.
- **Expert Ground Truth:** Systematic expert annotations serving as the reference standard for evaluation.

Both systems were evaluated against expert ground truth using the 200-event dataset. Performance metrics were calculated using the scikit-learn library.

6.1 Quantitative Performance Analysis

Table 4 presents the comparative performance of the baseline Gemini 2.5 Pro model and the RAG-enabled system. The RAG-enabled system demonstrates substantial improvement across all metrics, achieving an F1-score of 79% compared to 45% for the baseline model, representing a 76.4% relative improvement.

Table 4: Quantitative performance comparison of baseline and RAG-enabled Gemini 2.5 Pro systems.

Performance Metric	Baseline Gemini 2.5 Pro (%)	RAG-Enabled Gemini 2.5 Pro (%)
Accuracy	46	78
F1-Score	45	79
Precision	69	85
Recall	46	78

The RAG-enabled system achieved 78% accuracy, representing a 70.5% relative improvement over the baseline's 46% accuracy. This enhancement demonstrates the system's improved ability to correctly classify CloudTrail events. The F1-score improvement from 45% to 79% (76.4% relative improvement) indicates significantly improved balance between precision and recall, critical for operational deployment where both false positives and false negatives impose costs on security teams.

Precision increased from 69% to 85% (23.6% relative improvement), demonstrating reduced false positive rates. While this represents the smallest relative improvement, the baseline already achieved reasonable precision; the RAG system further refined classification accuracy by grounding predictions in authoritative threat intelligence. Recall improved from 46% to 78% (70.5% relative improvement), substantially reducing false negative rates and enabling more comprehensive threat detection.

Figs. 2 and 3 visualize these performance improvements, clearly illustrating the RAG system's consistent superiority across all evaluation metrics. The particularly notable improvements in F1-score and recall indicate that the RAG approach successfully addresses the baseline model's primary limitation: insufficient contextual knowledge about current threats and AWS-specific attack patterns.

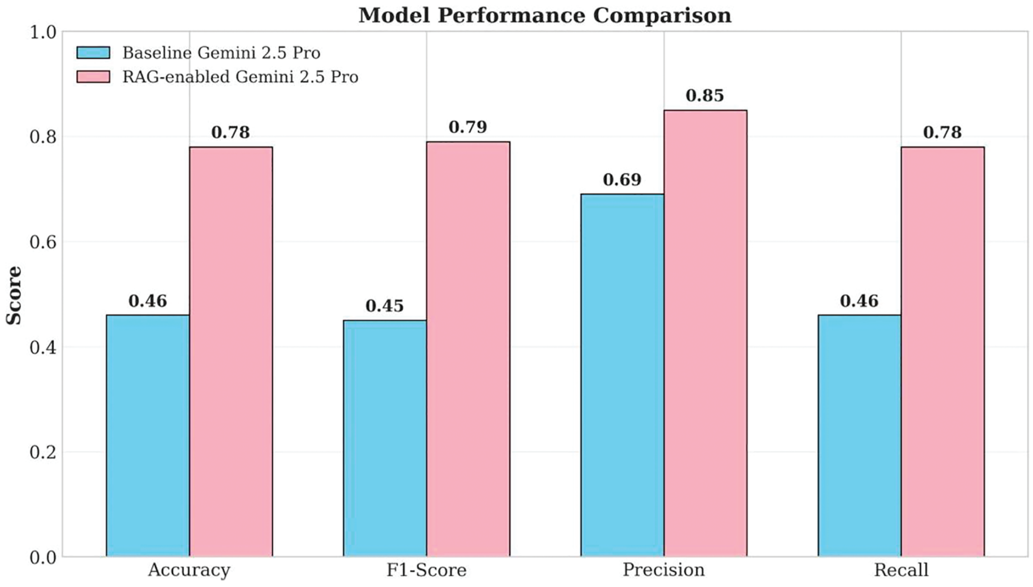


Figure 2: Model performance metrics comparison.

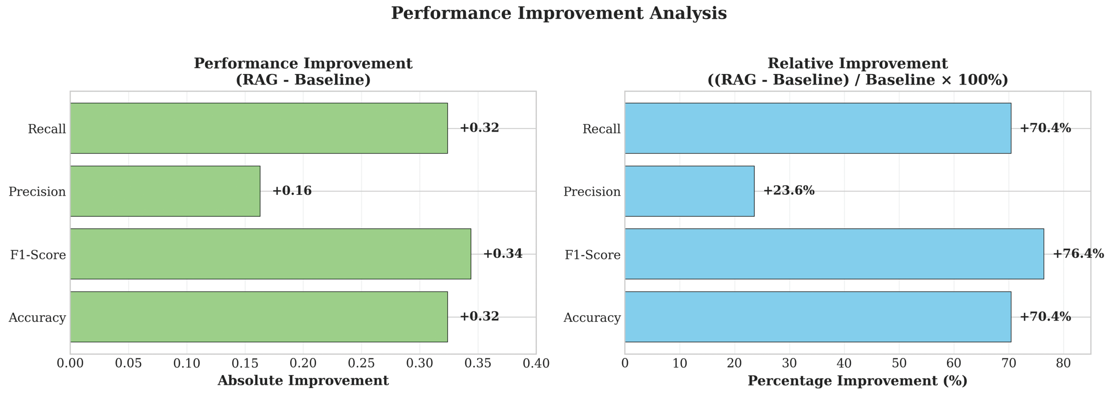


Figure 3: Performance metrics.

6.2 *Confusion Matrix Analysis*

Fig. 4 presents the confusion matrix for the RAG-enabled system, showing the distribution of predictions across MITRE ATT&CK techniques. The matrix reveals strong diagonal concentration, indicating high accuracy in correctly identifying techniques. Several key observations emerge from detailed confusion matrix analysis:

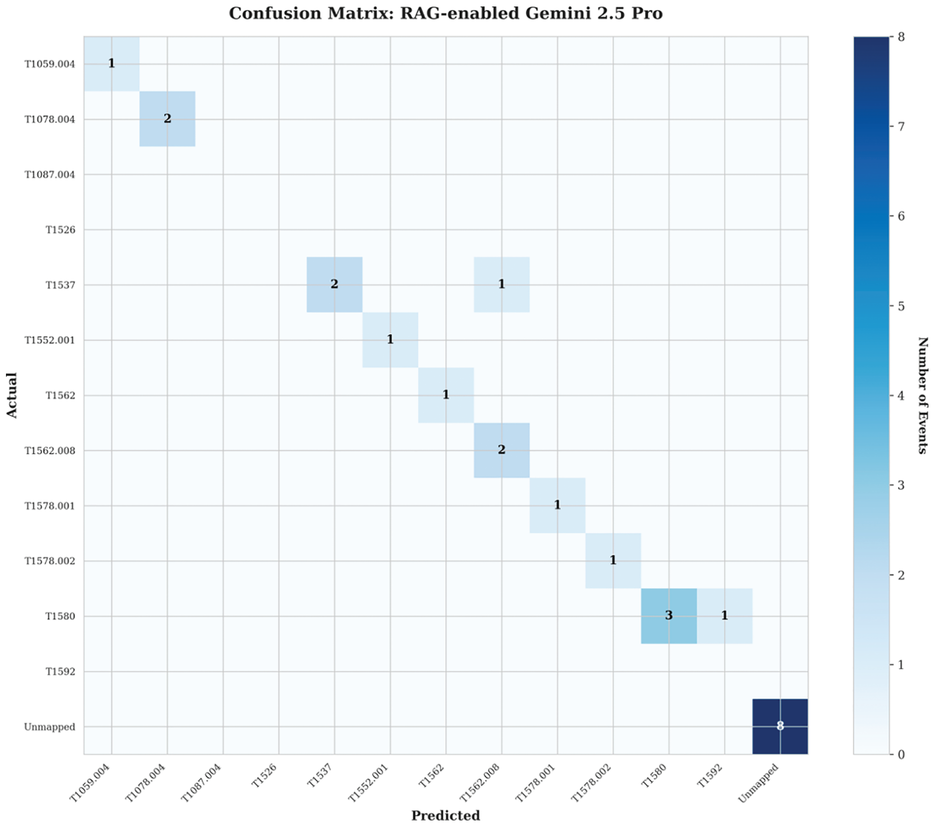


Figure 4: Confusion matrix for RAG-enabled Gemini 2.5 pro model.

High Accuracy for Critical Techniques: The majority of MITRE ATT&CK techniques were correctly identified, as evidenced by the concentration of predictions along the diagonal. Techniques associated with

Credential Access, Persistence, and Privilege Escalation, representing the most critical threat categories, achieved 100% accuracy.

Discovery Technique Performance: T1580 (Cloud Infrastructure Discovery) showed high frequency with strong but not perfect recognition (87% accuracy). The single misclassification indicates the challenge of distinguishing malicious reconnaissance from benign operational monitoring, as both involve similar AWS API calls such as `DescribeInstances` and `DescribeVpcs`.

Conservative Classification Strategy: Eight events were classified as "Unmapped," representing cases where the system lacked sufficient confidence to assign specific MITRE ATT&CK techniques. This conservative approach maintains high precision by avoiding false positive technique assignments. In operational deployment, these unmapped events would be flagged for manual analyst review rather than generating potentially incorrect automated responses.

Minimal Cross-Contamination: Very few off-diagonal entries indicate low confusion between different attack techniques, demonstrating the system's ability to distinguish between similar but distinct malicious behaviors. This specificity is critical for accurate threat intelligence and appropriate incident response.

In contrast, the baseline Gemini 2.5 Pro model (Fig. 4) exhibited substantially more off-diagonal predictions and numerous "Unmapped" classifications, reflecting its limited contextual knowledge. The baseline failed to correctly classify critical techniques such as T1552.001 (Unsecured Credentials: Credentials In Files) and T1578.001 (Modify Cloud Compute Infrastructure: Create Snapshot), both of which were accurately identified by the RAG-enabled system.

6.3 Performance Breakdown by MITRE ATT & CK Tactic

Performance varies significantly across MITRE ATT&CK tactics, reflecting differing levels of detection difficulty. Table 5 presents accuracy rates stratified by tactical category.

Table 5: Performance breakdown by MITRE ATT&CK tactic.

MITRE ATT&CK Tactic	Total Events	Correct	Accuracy (%)
Credential Access	63	63	100
Persistence	16	16	100
Privilege Escalation	16	16	100
Collection	4	4	100
Initial Access	1	1	100
Discovery	54	47	87.0
Defense Evasion	6	4	66.7
Exfiltration	7	4	57.1

The system achieved perfect accuracy (100%) on Credential Access, Persistence, Privilege Escalation, Collection, and Initial Access tactics. These tactics represent direct compromise and escalation scenarios with clear, unambiguous indicators in CloudTrail logs. For example, `GetParameter` with `withDecryption=true` unambiguously indicates credential theft (T1552.001), while `CreateAccessKey` definitively represents persistence establishment (T1098.001).

Discovery events showed 87% accuracy, with errors primarily stemming from ambiguity between malicious reconnaissance and benign monitoring. Operations like `DescribeInstances` serve both legitimate infrastructure management and adversary reconnaissance purposes, requiring additional context for definitive classification.

Defense Evasion (66.7%) and Exfiltration (57.1%) exhibited lower accuracy due to the subtle, context-dependent nature of these tactics. Events such as `DeleteTrail` and `PutBucketPolicy` modifications can be legitimate administrative actions or malicious activities depending on operational context not fully captured in individual CloudTrail events.

6.4 Systematic Error Analysis

To understand system limitations and identify improvement opportunities, we conducted systematic analysis of all misclassifications, categorizing errors into three distinct failure modes:

6.4.1 Retrieval-Generation Gap (26% of Errors)

The most common failure mode occurs when the retrieval component successfully identifies relevant context, but the generation component fails to properly integrate this information. For example, in analyzing a `PutBucketPolicy` event modifying S3 bucket access policies, the system retrieved documentation for both T1537 (Transfer Data to Cloud Account) and T1562.008 (Disable Cloud Logs). Despite the event clearly indicating external account access provisioning, the LLM incorrectly prioritized the "log disruption" framing, selecting T1562.008 over the correct T1537 classification.

This failure pattern manifests through surface-level keyword matching without semantic understanding of policy intent, and generation prioritization errors where top-ranked retrieval results receive disproportionate weight despite lower-ranked results containing more relevant information.

6.4.2 Knowledge Base Gap (20% of Errors)

The second failure mode involves missing information in the knowledge base regardless of retrieval quality. These gaps include newly released AWS services not yet documented in MITRE ATT&CK, benign operational patterns underrepresented in the (primarily attack-focused) knowledge base, and contextual details insufficient for disambiguating legitimate versus malicious usage of AWS operations.

For instance, events involving Route53Resolver operations (a relatively recent AWS service) lacked sufficient knowledge base coverage, leading to uncertain classifications despite otherwise functional retrieval and generation.

6.4.3 Ambiguous Ground Truth (20% of Errors)

The third failure mode involves events where multiple MITRE ATT&CK technique mappings are defensible. Operations such as `AssumeRole`, `CreateAccessKey`, and `DescribeInstances` serve both legitimate and malicious purposes, with correct classification depending on temporal context, user context, and organizational operational patterns not captured in single CloudTrail events.

For example, `CreateAccessKey` on an existing user could map to either T1098.001 (Account Manipulation: Additional Cloud Credentials) or T1136.003 (Create Account: Cloud Account), with the distinction being subtle and context-dependent. This reflects fundamental ambiguity in the MITRE ATT&CK framework itself rather than system deficiency.

6.5 *Severity Assessment and Operational Implications*

Events in the evaluation dataset were classified by severity based on potential impact. Table 6 presents the distribution across severity levels.

Table 6: Detection performance by threat severity level.

Severity Level	Total Events	Correct	Accuracy (%)
Critical	7	7	100
High	72	70	97.2
Medium	34	29	85.3
Low	54	45	83.3
Benign	78	68	87.2

The system demonstrates exceptional performance on high-severity events (100% critical, 97.2% high), ensuring that the most dangerous threats are reliably identified. Lower accuracy on Medium (85.3%) and Low (87.3%) severity events is acceptable given their reduced impact. The 87.2% accuracy on benign events corresponds to a 12.8% false positive rate, representing a manageable operational burden requiring analyst review.

This severity-stratified performance profile is ideal for operational deployment: critical threats receive perfect detection, while lower-priority events and benign operations are processed with good but not perfect accuracy. Security teams can confidently rely on the system for high-impact threat identification while maintaining analyst oversight for ambiguous cases.

6.6 *STRIDE Threat Modeling Analysis*

Beyond MITRE ATT&CK mapping, the RAG-enabled system provides STRIDE threat model classifications with associated confidence levels. Fig. 5 illustrates the distribution of STRIDE categories and confidence assessments.

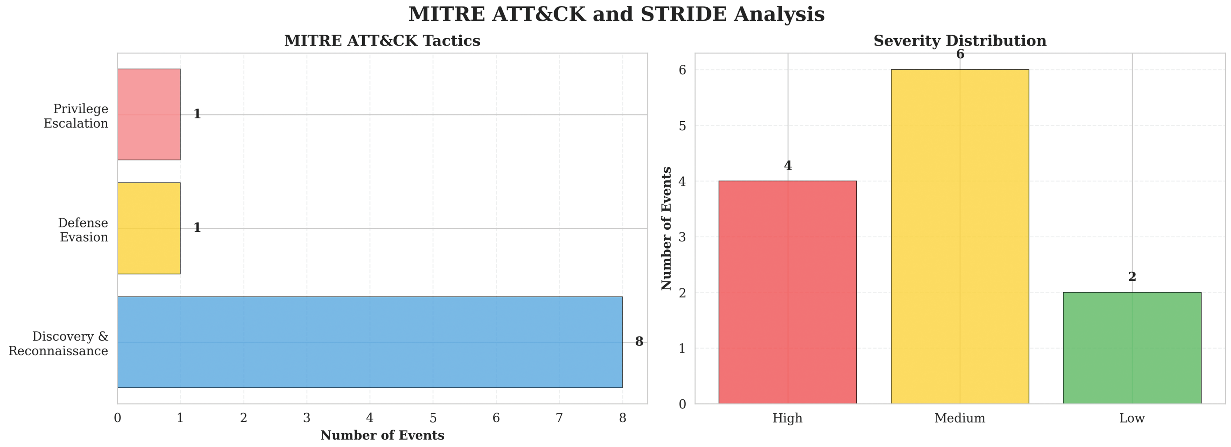


Figure 5: (Continued)

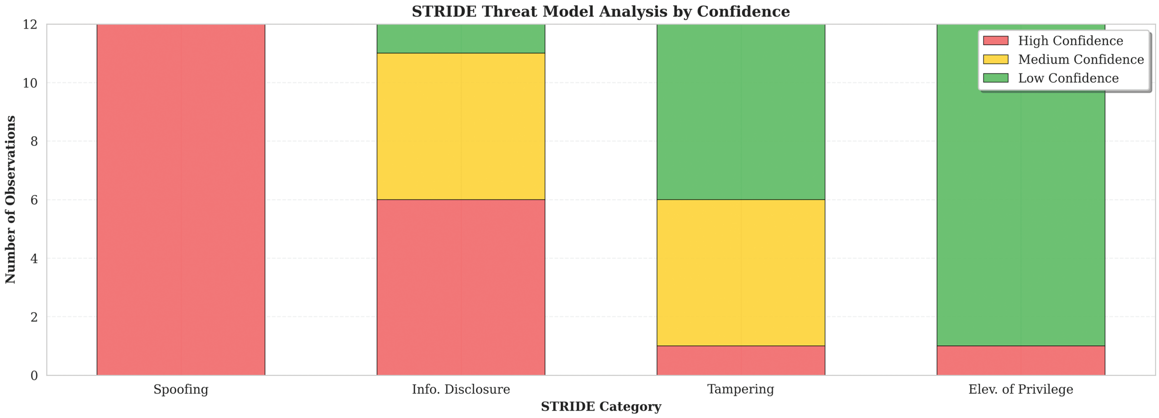


Figure 5: Severity score distribution.

Spoofing (12 observations): All classified with high confidence, indicating strong identification of identity-related threats. The system reliably detects activities such as console logins without multi-factor authentication and credential theft attempts.

Information Disclosure (12 observations): 50% high confidence (6 events), suggesting the system reliably identifies clear data exfiltration attempts but requires additional context for certain ambiguous cases such as policy modifications that could enable future data access.

Tampering (12 observations): Predominantly medium (5) and low (6) confidence, with only 1 high-confidence classification. Tampering activities are challenging to identify definitively from individual CloudTrail events, often requiring multi-event correlation to establish malicious intent.

Elevation of Privilege (12 observations): Mostly low confidence (11 events), indicating that privilege escalation scenarios require multi-event temporal analysis for definitive classification. Single events like `AttachUserPolicy` could represent legitimate administrative actions or malicious privilege escalation depending on context.

The high confidence in Spoofing detection aligns with the system's strong performance on credential-related MITRE ATT&CK techniques (T1078.004, T1552.001), while lower confidence in Elevation of Privilege highlights the need for temporal analysis of event sequences, a capability beyond the scope of this single-event classification study.

6.7 Qualitative Case Study Analysis

Qualitative analysis provides deeper insight into the system's reasoning capabilities and failure modes through detailed examination of representative cases.

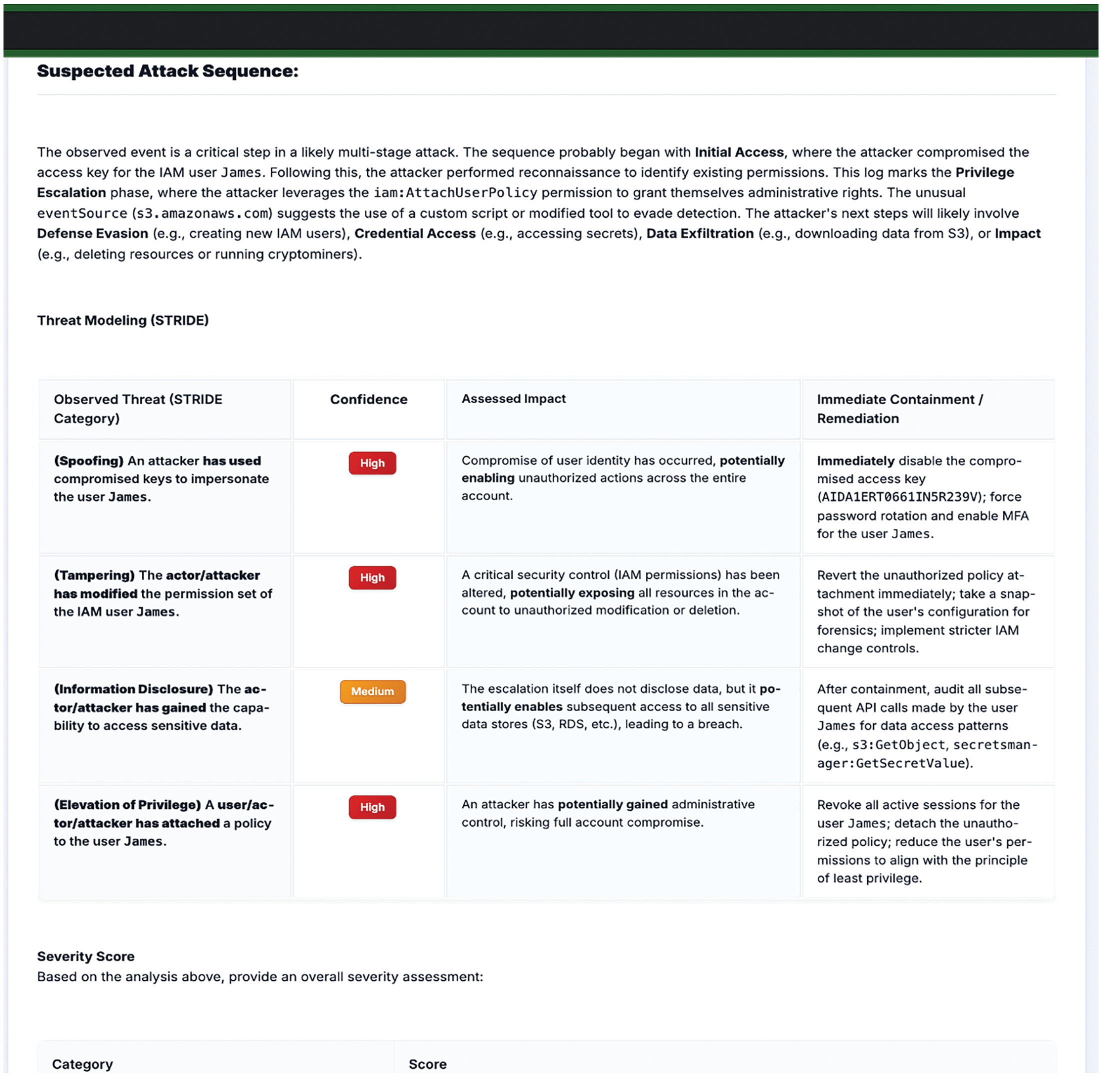
Positive Case Study: The analysis of a `RunInstances` failed event demonstrated the RAG system's enhanced ability to reason about complex attack chains. Despite the event's failure status, the system correctly identified it as T1578.002 (Modify Cloud Compute Infrastructure: Create Cloud Instance), recognizing that failed attempts still represent reconnaissance or capability testing by adversaries. The system generated contextual explanations linking the event to potential persistence mechanisms and provided strategic remediation recommendations beyond those produced by the baseline model, which classified the failed event as benign due to its error status.

Negative Case Study: The `PutBucketPolicy` misclassification revealed systematic weaknesses in retrieval-generation integration. The system prioritized generalized retrieved context over explicit event evidence, suggesting either weak retrieval through surface-level keyword matching or generation errors from reliance on misaligned context. This case underscores the need for semantic retrieval techniques and rigorous validation of generated outputs against raw event data.

These case studies validate the quantitative findings: the RAG system provides substantial value through contextual reasoning and threat intelligence grounding, but systematic weaknesses in retrieval quality and generation validation require addressing for production deployment.

6.8 Practical Implications for Security Operations

The performance of the RAG-enabled system constitutes a meaningful contribution to cloud security operations as shown in Fig. 6, which operates as a complementary automation tool rather than a replacement for human expertise. The 79% recall score reflects enhanced detection accuracy, improving capabilities in:



Suspected Attack Sequence:

The observed event is a critical step in a likely multi-stage attack. The sequence probably began with **Initial Access**, where the attacker compromised the access key for the IAM user `James`. Following this, the attacker performed reconnaissance to identify existing permissions. This log marks the **Privilege Escalation** phase, where the attacker leverages the `iam:AttachUserPolicy` permission to grant themselves administrative rights. The unusual `eventSource` (`s3.amazonaws.com`) suggests the use of a custom script or modified tool to evade detection. The attacker's next steps will likely involve **Defense Evasion** (e.g., creating new IAM users), **Credential Access** (e.g., accessing secrets), **Data Exfiltration** (e.g., downloading data from S3), or **Impact** (e.g., deleting resources or running cryptominers).

Threat Modeling (STRIDE)

Observed Threat (STRIDE Category)	Confidence	Assessed Impact	Immediate Containment / Remediation
(Spoofing) An attacker **has used** compromised keys to impersonate the user `James`.	High	Compromise of user identity has occurred, **potentially enabling** unauthorized actions across the entire account.	**Immediately** disable the compromised access key (`AIDA1ERT0661IN5R239V`); force password rotation and enable MFA for the user `James`.
(Tampering) The **actor/attacker has modified** the permission set of the IAM user `James`.	High	A critical security control (IAM permissions) has been altered, **potentially exposing** all resources in the account to unauthorized modification or deletion.	Revert the unauthorized policy attachment immediately; take a snapshot of the user's configuration for forensics; implement stricter IAM change controls.
(Information Disclosure) The **actor/attacker has gained** the capability to access sensitive data.	Medium	The escalation itself does not disclose data, but it **potentially enables** subsequent access to all sensitive data stores (S3, RDS, etc.), leading to a breach.	After containment, audit all subsequent API calls made by the user `James` for data access patterns (e.g., `s3:GetObject`, `secretsmanager:GetSecretValue`).
(Elevation of Privilege) A **user/actor/attacker has attached** a policy to the user `James`.	High	An attacker has **potentially gained** administrative control, risking full account compromise.	Revoke all active sessions for the user `James`; detach the unauthorized policy; reduce the user's permissions to align with the principle of least privilege.

Severity Score

Based on the analysis above, provide an overall severity assessment:

Category	Score

Figure 6: RAG-enabled LLM output.

- **Reactive Threat Hunting:** Supporting Tier 1 analysts in investigating suspicious CloudTrail activity with automated MITRE ATT&CK mapping and natural language threat explanations

- **Alert Validation:** Providing contextual analysis for alerts generated by other security platforms (GuardDuty, SIEM systems)
- **Incident Investigation:** Accelerating post-incident analysis through automated technique identification and attack chain reconstruction
- **Compliance Reporting:** Generating MITRE ATT&CK-mapped threat reports for regulatory requirements (SOC 2, ISO 27001, NIST)

Beyond detection, the system provides threat modeling assessments using the STRIDE framework and risk severity classifications. This enables security teams to identify potential attack chains, assign confidence values and impact severities, determine affected assets, and generate remediation recommendations. The integration of RAG capability provides security teams with valuable insights from documented threats and historical incidents, improving both proactive threat modeling and reactive incident response effectiveness.

However, the cost-latency trade-offs indicate that deployment strategies must carefully consider organizational priorities, event volumes, and response time requirements. The 4.1-s processing time and $0.00376 per-event cost are justified for complex, high-value events requiring detailed analysis, but may not be suitable for high-volume routine operations or sub-second response scenarios. Tiered deployment architectures combining rule-based filtering, baseline LLM processing, and selective RAG application provide optimal balance between cost and detection capability.

7 Recommendations and Future Work

Based on the findings and systematic error analysis presented in this research, we identify multiple high-priority directions for future investigation and system enhancement.

7.1 Retrieval Quality Enhancement

With 26% of system errors attributable to retrieval-generation gaps, improving retrieval quality represents the highest-leverage intervention. Future work should focus on:

Semantic Retrieval Models: Implement embedding models specifically fine-tuned on cybersecurity domain terminology and MITRE ATT&CK documentation. General-purpose embeddings (e.g., text-embedding-004) may not capture domain-specific semantic relationships between CloudTrail operations and threat intelligence concepts.

Hybrid Retrieval Strategies: Combine dense semantic retrieval with sparse keyword-based approaches to balance semantic understanding with precise term matching. Cloud security terminology often requires exact matches (“withDecryption=true”) that semantic embeddings may not prioritize.

Query Expansion Techniques: Generate multiple query variations capturing different semantic perspectives of the same CloudTrail event, then aggregate retrieved context across all query formulations to ensure comprehensive coverage.

Learned Re-Ranking: Train neural re-ranking models that assess retrieved chunk relevance given the specific CloudTrail event, moving beyond first-stage retrieval scores to context-aware relevance assessment.

Agent-Based Orchestration: Implement agentic retrieval pipelines that plan retrieval strategies, decompose queries into sub-questions, execute iterative searches, and synthesize results across multiple retrieval rounds.

7.2 Knowledge Base Expansion and Maintenance

Twenty percent of errors stem from knowledge base gaps, suggesting need for:

Automated Knowledge Base Updates: Establish pipelines for incorporating MITRE ATT&CK framework updates, new AWS service documentation, and recent security research publications. The cybersecurity domain evolves rapidly; static knowledge bases quickly become outdated.

Benign Operation Baselines: Expand knowledge base to include benign operational patterns for all major AWS services. Current content emphasizes attack techniques, leaving the system underprepared for distinguishing malicious from legitimate operations.

Contextual Usage Examples: Augment technique descriptions with concrete CloudTrail event examples showing both malicious and benign usage patterns. Abstract descriptions ("adversaries may use GetParameter to retrieve credentials") provide insufficient guidance for disambiguating specific events.

7.3 Multi-Event Correlation and Temporal Analysis

The current system analyzes individual CloudTrail events in isolation, limiting detection of sophisticated attack chains requiring temporal correlation. Future work should explore:

Temporal Event Sequences: Extend the architecture to analyze sequences of related CloudTrail events, enabling identification of multi-stage attacks such as credential theft followed by privilege escalation followed by data exfiltration.

Graph-Based Attack Chain Reconstruction: Represent CloudTrail events as nodes in temporal graphs, with edges indicating causal relationships (same user, same session, resource dependencies). Graph neural networks or graph-enhanced RAG could reason over these structures.

Stateful Context Management: Maintain session-level and user-level context across multiple events, enabling detection of anomalous behavior patterns that span hours or days rather than single operations.

7.4 Expanded Baseline Comparisons

This study compared RAG-enabled and baseline LLM systems. Future research should include:

Traditional Machine Learning Baselines: Train supervised classifiers (Random Forest, SVM, XGBoost) on labeled CloudTrail data to establish comparative performance benchmarks, quantifying RAG's advantage over conventional ML approaches.

Commercial System Evaluation: Where possible, compare against commercial solutions such as AWS GuardDuty and major SIEM platforms to position RAG capabilities relative to industry-standard tools.

Ablation Studies: Systematically evaluate individual RAG components (query generation, retrieval, generation) through controlled ablation experiments, quantifying each component's contribution to overall performance.

7.5 Production Deployment Optimizations

The cost-latency analysis identified trade-offs requiring mitigation for production deployment:

Latency Reduction: Investigate parallel processing architectures, caching strategies for frequently-encountered events, and model distillation approaches that compress RAG capabilities into faster inference models.

Cost Optimization: Explore tiered deployment strategies where simple events receive rule-based or lightweight processing, reserving expensive RAG analysis for complex, high-value events. Develop learned classifiers that predict which events benefit from RAG analysis.

Uncertainty Quantification: Implement confidence scoring mechanisms that flag low-confidence predictions for manual review rather than forcing classification of ambiguous events. Embrace "defer to human" strategies that acknowledge inherent limitations.

7.6 Evaluation Dataset Expansion

While this study's 200-event dataset represents substantial improvement over preliminary investigations (24 events), further expansion would strengthen findings:

Production Environment Validation: Collaborate with organizations to evaluate system performance on anonymized production CloudTrail logs, validating generalization beyond simulated attacks to real-world operational environments.

Adversarial Robustness Testing: Develop CloudTrail events specifically designed to evade detection, testing system robustness against adversarial evasion techniques documented in threat intelligence.

Cross-Cloud Platform Expansion: Extend evaluation to Microsoft Azure Activity Logs and Google Cloud Platform Audit Logs, assessing whether RAG approaches generalize across cloud providers or require platform-specific adaptation.

8 Conclusion

Interpreting AWS CloudTrail events for post-incident analysis presents significant challenges for security analysts, including alert fatigue, high false-positive rates, and the impracticality of manual analysis given modern cloud event volumes. While machine learning and baseline Large Language Model approaches have achieved partial success, they encounter critical limitations including insufficient contextual knowledge, model obsolescence, and substantial training data requirements. This study developed and systematically evaluated a two-step Retrieval-Augmented Generation (RAG) system using Gemini 2.5 Pro to enhance CloudTrail threat detection through integration of external cybersecurity knowledge sources including the MITRE ATT&CK framework and threat intelligence reports. Evaluation on a 200-event dataset (122 malicious, 78 benign) demonstrated substantial improvements: the RAG-enabled system achieved 79% F1-score, 85% precision, and 78% accuracy, representing 76.4% F1-score improvement over the baseline model. Systematic error analysis revealed retrieval quality as the primary bottleneck (60% of errors), while cost-latency analysis quantified deployment trade-offs (4.1 s and $0.00376 per event) comparable to commercial SIEM solutions but with superior MITRE ATT&CK attribution.

This research demonstrates that RAG-enabled systems provide practical value for CloudTrail threat detection when deployed with appropriate understanding of capabilities and limitations. The system operates as a complementary automation tool supporting security analysts in threat hunting, alert validation, and compliance reporting rather than replacing human expertise. Future work should prioritize retrieval quality enhancement through semantic fine-tuning and hybrid search strategies, expand knowledge base coverage of benign operations and recent AWS services, and extend single-event analysis to multi-event temporal correlation for sophisticated attack chain detection. The findings reveals that retrieval-augmented generation represents a promising direction for cloud audit log analysis, balancing detection accuracy, operational costs, and adaptability to evolving threats.

Acknowledgement: Not Applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm their contribution to the paper as follows: conceptualization, Goodness Adediran and Yussuf Ahmed; methodology, Goodness Adediran and Yussuf Ahmed; software, Goodness Adediran and

Yussuf Ahmed; validation, Goodness Adediran and Yussuf Ahmed; formal analysis, Goodness Adediran and Yussuf Ahmed; resources; Goodness Adediran and Yussuf Ahmed; data curation, Goodness Adediran and Yussuf Ahmed; writing, original draft preparation, Goodness Adediran and Yussuf Ahmed; editing, Goodness Adediran, Yussuf Ahmed and Kenny Awuson-David; visualization, Goodness Adediran, Yussuf Ahmed and Kenny Awuson-David; supervision, Yussuf Ahmed; project administration, Yussuf Ahmed and Kenny Awuson-David; funding acquisition, Yussuf Ahmed. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Data available upon reasonable request.

Ethics Approval: Not Applicable.

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

Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
APT	Advanced Persistent Threat
ATT&CK	Adversarial Tactics, Techniques, and Common Knowledge
AWS	Amazon Web Services
BERT	Bidirectional Encoder Representations from Transformers
CLI	Command Line Interface
CloudTrail	AWS Service for Logging API Calls and Account Activity
CoT	Chain of Thoughts
CSA	Cloud Security Alliance
CSP	Cloud Service Provider
CSU	Cloud Service Users
DL	Deep Learning
GCP	Google Cloud Platform
GCS	Google Cloud Storage
GPT	Generative Pre-Trained Transformer
IAM	Identity and Access Management
LLM	Large Language Model
MITRE	Organization Maintaining ATT&CK Framework
ML	Machine Learning
NLP	Natural Language Processing
RAG	Retrieval-Augmented Generation
SIEM	Security Information and Event Management
STRIDE	Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege
TTP	Tactics, Techniques, and Procedures
UCTM	Universal Cloud Threat Model

References

1. Barach J. AI-driven causal inference for cross-cloud threat detection using anonymized CloudTrail logs. In: Proceedings of the 2025 Conference on Artificial Intelligence x Multimedia (AIxMM); 2025 Feb 3–5; Laguna Hills, CA, USA. p. 45–50.
2. Tabrizchi H, Kuchaki Rafsanjani M. A survey on security challenges in cloud computing: issues, threats, and solutions. J Supercomput. 2020;76(12):9493–532. doi:10.1007/s11227-020-03213-1.
3. Silva A. Announcing AWS CloudTrail lake—a managed audit and security lake. AWS Blog. 2022 [cited 2025 Aug 1]. Available from: https://aws.amazon.com/blogs/mt/announcing-aws-cloudtrail-lake-a-managed-audit-and-security-lake/.

4. Olateju O, Okon SU, Igwenagu U, Salami AA, Oladoyinbo TO, Olaniyi OO. Combating the challenges of false positives in AI-driven anomaly detection systems and enhancing data security in the cloud. Asian J Res Comput Sci. 2024;17(6):264–92. doi:10.9734/ajrcos/2024/v17i6472.
5. Singh V. What is AWS? An introduction to Amazon Web Services. 2025 [cited 2025 Aug 1]. Available from: https://www.datacamp.com/blog/what-is-aws.
6. Amazon Web Services. What is AWS CloudTrail?—AWS CloudTrail. 2025 [cited 2025 Aug 17]. Available from: https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-user-guide.html.
7. Kim Y, Kim J, Chae S, Hong J, Kim S. Event log analysis framework based on the ATT&CK matrix in cloud environments. J Korea Inst Inf Secur Cryptol. 2024;34:263–79.
8. Jiang Y, Meng Q, Shang F, Oo N, Minh LTH, Lim HW, et al. MITRE ATT&CK applications in cybersecurity and the way forward. arXiv:2502.10825. 2025.
9. Panda M, Mukherjee S. Enhancing privacy and security in RAG-based generative AI applications. AI Mach Learn Appl Adv. 2025;15(3):1–10. doi:10.2139/ssrn.5162147.
10. Sharma AN, Akbar KA, Thuraisingham B, Khan L. Enhancing security insights with KnowGen-RAG: combining knowledge graphs, LLMs, and multimodal interpretability. In: Proceedings of the 10th ACM International Workshop on Security and Privacy Analytics; 2024 Jun 21; New York, NY, USA. p. 2–12.
11. French W. Enhancing threat hunting automation with large language models. 2024 [cited 2025 Aug 7]. Available from: https://www.proquest.com/dissertations-theses/enhancing-threat-huntingautomation-with-large/docview/3142158088/se-2.
12. Mala U. Comparative analysis of splunk vs. AWS native monitoring tools for cloud security and threat detection. 2024 [cited 2025 Aug 5]. Available from: https://norma.ncirl.ie/8119/.
13. Tykholaz D, Banakh R, Mychuda L, Piskozub A. Incident response with AWS detective controls. 2024 [cited 2025 Aug 5]. Available from: https://www.researchgate.net/publication/385858920_Incident_response_with_AWS_detective_controls.
14. Kumar RSS, Wicker A, Swann M. Practical machine learning for cloud intrusion detection: challenges and the way forward. 2017 [cited 2025 Aug 7]. Available from: https://arxiv.org/abs/1709.07095.
15. Le VH, Zhang H. Log-based anomaly detection with deep learning: how far are we? arXiv:2202.04301. 2022.
16. Talukder MA, Islam MM, Uddin MA, Hasan KF, Sharmin S, Alyami SA, et al. Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. J Big Data. 2024;11(1):33. doi:10.1186/s40537-024-00886-w.
17. Okoli UI, Obi OC, Adewusi AO, Abrahams TO. Machine learning in cybersecurity: a review of threat detection and defense mechanisms. World J Adv Res Rev. 2024;21(1):2286–95.
18. Chalapathy R, Chawla S. Deep learning for anomaly detection: a survey. arXiv:1901.03407. 2019.
19. DataDog. Stratus red team. 2021 [cited 2025 Aug 12]. Available from: https://stratus-red-team.cloud/.
20. Velez C, Liu L. Vertex AI RAG engine: build & deploy RAG implementations with your data. 2025 [cited 2025 Aug 14]. Available from: https://cloud.google.com/blog/products/ai-machine-learning/introducing-vertex-ai-rag-engine.
21. AI V. Gemini 2.5 Pro. 2025 [cited 2025 Aug 14]. Available from: https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-5-pro.
22. Team G. Gemini 2.5: pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. 2025 [cited 2025 Aug 14]. Available from: https://storage.googleapis.com/deepmind-media/gemini/gemini_v2_5_report.pdf.
23. AI D. Few-shot prompting—Nextra. 2025 [cited 2025 Aug 15]. Available from: https://www.promptingguide.ai/techniques/fewshot.
24. AI D. Chain-of-thought prompting—Nextra. 2025 [cited 2025 Aug 15]. Available from: https://www.promptingguide.ai/techniques/cot.