

An Explainable Machine Learning Framework for AWS S3 CloudTrail-Like Log Analysis in Digital Forensics

Aseel Aljuhani¹, Asma A. Alhashmi², Abdulbasit A. Darem^{3*}

Department of Computer Science, College of Science, Northern Border University, Arar 73213, Saudi Arabia^{1,2}
Center for Scientific Research and Entrepreneurship, Northern Border University, Arar 73213, Saudi Arabia³

Abstract—The advent of cloud computing for data storage and management has led to the emergence of advanced cyber threats that call for the need for effective cloud log analysis for digital forensic investigations. While there has been considerable development of machine learning algorithms for anomaly detection, most of the current approaches focus on detection effectiveness without considering interpretability. This research introduces an explainable hybrid machine learning architecture for analyzing Amazon Web Services (AWS) CloudTrail-like logs in cloud digital forensics. The framework uses XGBoost for classification, Isolation Forest for anomaly detection, rule-based reasoning to detect risky behaviors, and SHapley Additive exPlanations (SHAP) to explain XGBoost predictions. The framework further includes temporal behavior analysis and sensitive cloud object interactions for enhanced cloud digital forensic investigation. The performance of the framework is tested on 169,986 synthetic CloudTrail-like events, with the main evaluation using 35,963 temporally held-out events. The PR-AUC score obtained in the experiments is 0.9324, and ROC-AUC is 0.9827. During monitoring mode, the framework achieves a precision of 0.9677 and recall of 0.7930, whereas forensic mode raises precision to 0.9822 and reduces recall to 0.7844, while keeping the F1-score around 0.87 in both modes. Additional experiments highlight the role of sensitive-object features and demonstrate consistent discrimination across chronological validation folds.

Keywords—Cloud forensics; cloud audit logs; AWS CloudTrail; Explainable AI; anomaly detection; SHAP

I. INTRODUCTION

The emergence of cloud computing has revolutionized the storage, processing, and management of digital data [1]. Nevertheless, limited access to physical storage media in cloud environments makes log analysis more critical [2]. In cloud object stores like Amazon Simple Storage Service (S3), audit telemetry becomes vital for investigations, especially CloudTrail logs, as they keep track of who performed which action and when [3]. Such logs are vital in determining the sequence of events that occurred, such as unauthorized access, misuse of privileges, destructive activities, and data theft [4].

For this purpose, the threat model focuses on the security risk of unauthorized access, data exfiltration, and destructive operations in cloud object storage. Potentially suspicious activity is flagged through behavior patterns, including frequent access patterns, interaction with sensitive objects, rapid deletion of data, and possible data exfiltration. Although

it is important for security professionals who must respond to these attacks, analysis becomes time-consuming and difficult because of the volume, speed, and diversity of cloud logs [5]. However, there are issues in event classification beyond efficiency. It is necessary to understand the process of classification, find out reasons for classifying an event in a particular way, and correlate events with specific behaviors [6]. Cloud forensic investigation involves more than speed. In fact, to conduct forensic investigations, it is required to be efficient, explainable, defensible, and reproducible, as well as align with forensic standards and practices in evidence collection, processing, and presentation [6], [7].

This study presents an explainable machine learning approach for the analysis of cloud object storage logs, which considers temporal behavior patterns such as bursts of activity, suspicious combinations of actions, unusual read-write activity, and suspicious use of sensitive objects that may be targeted for exfiltration. The proposed framework is implemented in the form of a unified pipeline accessible via an API-based system. This provides an easy-to-use interface for uploading and analyzing logs on demand while observing the detection results and rationale behind the detected suspicious activities.

The main contributions of this work are:

- Development of a forensic investigation pipeline based on features extracted from S3 event logs analogous to CloudTrail log information.
- Integration of supervised machine learning, anomaly detection, and rule-based validation to support the forensic analysis process.
- Provision of SHAP-based feature explanations for flagged XGBoost predictions to improve clarity.
- Evaluation of framework decision sensitivity under controlled noise and missing-field conditions.

Digital forensic standards, such as NIST SP 800-86 and ISO/IEC 27037 [7], [8], emphasize evidence traceability, preservation, and repeatability.

By combining explainable machine learning with reliable logging practices, cloud digital forensic investigations can benefit from interpretable triage and more transparent analysis.

*Corresponding author.

Based on this gap, the present study addresses the following research question: How can explainable machine learning support the analysis of cloud activity logs for digital forensic investigations?

II. BACKGROUND AND RELATED WORK

Cloud forensics differs from traditional digital forensics because data and systems are located in various cloud providers [9]. In general, physical devices can be directly accessed by investigators [10], but in cloud environments, the investigator is mainly dependent on audit logs and cloud-provider records [9]. Various problems have been discovered in cloud forensics research [11].

NIST has outlined such challenges and discussed a set of general techniques for cloud forensics investigations [12]. Other scientists noted the importance of evidence preservation, reproducibility, and forensics in their studies [13]. Certain standards and guidelines, such as NIST SP 800-86 and ISO/IEC 27037, focus on such aspects as accuracy, traceability, and reproducibility of digital evidence [7], [8]. The above aspects are even more important in the case of a cloud environment, where access to digital evidence can be challenging [14].

In such environments, audit logs become critical in forensic investigations, particularly in cloud storage services [4]. Actions such as GetObject, PutObject, and DeleteObject can be recorded by AWS CloudTrail for Amazon S3 when data event logging is enabled [15], [16]. Therefore, log analysis becomes crucial to understand the user activity and identify possible anomalies [17].

Currently, there is ample literature on forensic frameworks and evidence gathering in cloud forensics. FROST provides assistance in evidence gathering in an OpenStack environment [18], while another approach is based on using APIs to obtain information from cloud infrastructure [19]. However, much less work has been done on scalable and explainable analysis of user behavior in cloud forensic investigations.

A number of methods have been utilized in machine learning for anomaly detection and classification purposes. XGBoost has proven highly effective in working with structured data [20], while Isolation Forest helps to detect rare or uncommon cases [21]. Moreover, probability calibration methods can be applied to enhance prediction reliability and confidence estimation Table I.

At the same time, explainability has been gaining importance in security. The SHAP method is used extensively to explain model predictions by quantifying the impact of individual features [22]. Recent cybersecurity papers apply SHAP to improve the interpretability of intrusion detection systems [23], [24]. Previous works have stressed the importance of explainable AI in cybersecurity systems in terms of trust and accountability [25], [26]. Still, very little research has been done on explainable AI in cloud forensic log analysis.

A number of recent studies have shown very high accuracy in detecting anomalies in cloud environments and logs using machine learning algorithms. For instance, Girish and Rao found that LSTM-based models in OpenStack environments

could attain up to 94% test accuracy [27], and Shahzad et al. achieved 97.06% accuracy for binary detection using ensemble learning and 99.91% for multiclass classification using CNN-LSTM [28]. However, such studies pay little attention to the explainability of their detection results.

More recent studies have also investigated using explainable AI (XAI) and machine learning methods for detecting anomalies within cloud computing. Rajak [29] utilized the Isolation Forest and Autoencoders models on cloud logs, and managed to get approximately 0.92 precision, 0.95 recall, 0.94 F1-score, and an AUC greater than 0.95. In another paper, Hermosilla, Berrios, and Allende-Cid [30] applied SHAP and LIME to XGBoost and TabNet models and found that XGBoost achieved 97.8% validation accuracy and showed higher interpretability than TabNet. Furthermore, Vane et al. [31] implemented SHAP and LIME on a cloud-native intrusion detection system, resulting in a 97.4% accuracy in detecting the anomaly while reducing the Mean Time to Respond (MTTR) by 40%. Papavasileiou and Oubibi [32] employed a CNN-LSTM model with SHAP and LIME for explainable cloud anomaly detection, achieving 97.8% accuracy on UNSW-NB15 and 98.2% on the proprietary cloud audit log dataset, while providing explanations for the model's decisions. AI technology has also been used in digital forensic surveillance via the VigilEye framework that provides object detection and evidence preservation capabilities [33].

In fact, several other studies point out that high detection accuracy is not sufficient in the absence of forensic interpretability and analyst trust. According to Ma et al. [5], high-performing anomaly detection systems often lack sufficient explainability. Al-Ghuwairi et al. [34] found that time-series anomaly detection models can be both efficient and accurate, and Studiawan and Sohel [35] reported that deep autoencoders can achieve around 96.7% accuracy and 94% F1-score. In addition, Alharthi and Garcia [36] proposed the CIAF framework for automated forensic log analysis that provided precision and recall values of about 93%.

Furthermore, other papers emphasize the increasing role of explainability in cloud-based machine learning systems. Selvam and Selvy [37] reported 94.5% accuracy and a 27% decrease in false positives using hybrid XAI methods. Idamakanti [38] used federated learning to improve cloud network anomaly detection performance. Talati [39], Goyal [40], and Devarajan et al. [41] pointed out the significance of transparency and trust in cloud machine learning systems.

For an operational comparison, the Elasticsearch, Logstash, and Kibana (ELK) stack supports log ingestion, indexing, search, and visualization. The proposed framework instead evaluates CloudTrail-like events using calibrated machine-learning scores, an additional anomaly component, bounded behavioral rules, and XGBoost-specific SHAP explanations. These capabilities are complementary, and two native Elasticsearch rule-query baselines were evaluated on the same held-out TEST events, as reported in Section IV-G. This evaluates two specified detection policies, not the full ELK stack.

The current study concentrates explicitly on the use of synthetic CloudTrail-like log analysis in digital forensics based

on a hybrid approach that incorporates XGBoost, Isolation Forest, and rule-based detection mechanisms with SHAP-based explanations of XGBoost predictions. Therefore, the current study seeks to bridge the gap between detection accuracy, explainability, and alignment with digital forensic principles (NIST SP 800-86 and ISO/IEC 27037).

TABLE I. COMPARISON OF PREVIOUS STUDIES AND THE PROPOSED FRAMEWORK

Study	Dataset / Environment	Method	Explainability
Girish & Rao [27]	OpenStack cloud logs	Stacked and Bidirectional LSTM	No
Shahzad et al. [28]	Cloud anomaly dataset	Ensemble Learning / CNN-LSTM	No
Rajak [29]	Cloud logs	Isolation Forest + Autoencoder	No
Hermosilla et al. [30]	Intrusion detection datasets	XGBoost, TabNet	SHAP, LIME
Vane et al. [31]	Cloud intrusion detection	CNN + SHAP / LIME	Yes
Papavasileiou & Oubibi [32]	UNSW-NB15 + PCAL-10k cloud audit logs	CNN-LSTM + SHAP + LIME	Yes
Alharthi & Garcia [36]	Microsoft Azure forensic logs	CIAF Framework	Limited
Proposed Framework	Synthetic AWS CloudTrail-like logs	XGBoost + Isolation Forest + Rule-based Detection	SHAP (XGBoost)

III. MATERIALS AND METHODS

Fig. 1 presents the overall architecture of the proposed framework.

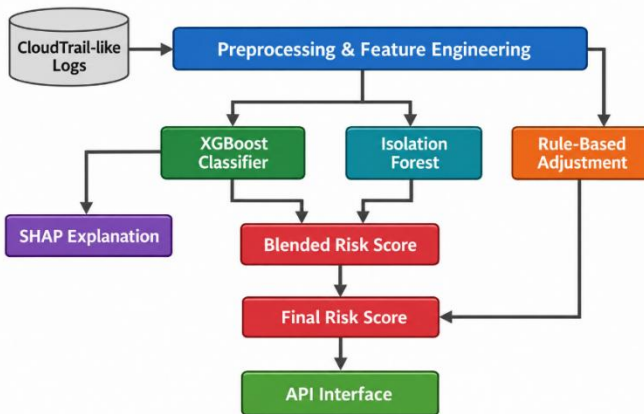


Fig. 1. Overall architecture of the proposed framework.

A. Laboratory Setting and Data

The experimental dataset was constructed on an AWS-like laboratory setup that simulated CloudTrail-style S3 activity. The input data consisted of 169,986 JSONL records representing object store activities, user identity information, source IPs, timestamps, request parameters, and both normal and suspicious actions. Supplementary data were gathered for robustness evaluation, scenario exploration, and reporting purposes. Since the experiment was carried out in a synthetic environment, there were no actual user data involved in the

research, while preserving selected S3 operation semantics. In order for behavioral characteristics useful for forensic investigation to remain, the dataset was designed to reflect typical user behavior and several types of cloud-forensics attacks, including aspects such as resource usage, data transfer, and data deletion. The dataset represents selected simulated behaviors and does not establish population-level representativeness. The synthetic dataset was prepared following common practices in cybersecurity dataset development, including activity simulation, event labeling, and feature extraction [42], [43].

B. Parsing and Semantic Normalization of Events

Raw request parameters were parsed and normalized into canonical operations (LIST, PUT, GET, DELETE, and RM). This step transformed the heterogeneous Application Programming Interface (API) and command patterns into behaviorally significant categories. Bucket names, object keys, and resource-specific attributes were also extracted by the parser to enable downstream contextualization.

C. Feature Engineering

Feature engineering was guided by forensic and behavioral considerations. Flags were employed to identify whether it was a read operation, write operation, list operation, delete operation, or removal operation. A sensitive-keyword detector identifies object keys associated with secrets, credentials, backups, financial documents, payroll content, tokens, or other private information. The exfiltration-bucket flag identifies writes to the defined exfiltration bucket. Time was encoded using hour-of-day and day-of-week sine/cosine transformations. In order to circumvent single-event inspection, the framework produced temporal contextual clusters along with rolling counts in 1, 5, 10, 15, and 30-minute time windows. These windows sampled the repeating patterns of listing, writing, reading, deletion, and access to sensitive objects. The high-level features calculated the burst intensity, list-to-write transition, delete pressure, write-vs-read ratio, destructive combination, and window intensity. The final feature set had 63 features. The rolling counts are calculated in chronological order for each user, summarizing actions in the preceding 1-, 5-, 10-, 15-, and 30-minute intervals (including the current event). They supply investigators with a compact description of repeated activity and potential behavioral bursts.

D. Labels and Feature-Set Variants

The main experiments were based on the existing label structure and heuristics of the project in the training pipeline. Rule-based heuristics were used to generate labels that reflected the known suspicious behaviors of deletion bursts, access of sensitive objects, and the possible data exfiltration patterns. Four controlled configurations were evaluated: FULL (63 features), NO_BURST_STRICT (45), NO_SENSITIVE_STRICT (55), and NO_WINDOWS_STRICT (13). The configurations were retrained independently; relevant rule adjustments were disabled when their required features were removed. Because the labels are heuristic-derived, their overlap with some engineered indicators was considered when interpreting the results.

E. Model Training and Calibration

XGBoost was selected because it works well with structured data and captures nonlinear feature relationships. Isolation Forest was included to provide a complementary anomaly score. To assess the performance of the framework, a chronological FIT/CAL/TUNE/TEST partition was applied, with 30-minute temporal embargoes and session checks to limit leakage. XGBoost and Isolation Forest were fitted using FIT; sigmoid calibration was used in CAL; decision thresholds were selected on TUNE and evaluated once on the held-out TEST partition Table II. Median imputation was fitted using training data. Two thresholds were defined: a monitoring threshold and a stricter forensic threshold. Three expanding-window temporal cross-validation folds were additionally trained inside FIT only.

TABLE II. EXPERIMENTAL SETUP AND MODEL TRAINING CONFIGURATION

Item	Setting
Split	Chronological FIT 101,156 / CAL 15,038 / TUNE 16,012 / TEST 35,963
Excluded	1,817 events (30-minute temporal embargo)
Calibration	Sigmoid calibration
Model	XGBoost + Isolation Forest with rule-based adjustment
Features	63 features
Thresholds	Monitoring = 0.6624, Forensic = 0.7003

The chronological separation and session checks were used to reduce temporal and session-related leakage between training and held-out evaluation. Experiments were executed on Windows 11, Python 3.12.10, XGBoost 3.1.3, and scikit-learn 1.8.0.

F. Hybrid Inference Logic

Inferences were not made using the output of the classifier alone. The probability output from the calibrated model was used along with the normalized anomaly score from Isolation Forest to produce a blended hybrid score. This score was then enhanced by the final bounded rule-boosting layer for dangerous behavioral combinations like mass writes with delete bursts, highly destructive intensity, and abnormally write-heavy behavior. The rule boost was intentionally bounded to preserve the primary role of the learned model while incorporating investigator intuition about dangerous behaviors.

The final prediction score is computed as:

- 1) $P_{blend}(x) = 0.8 P_{model}(x) + 0.2 A(x)$
- 2) $P_{final}(x) = \min(1, P_{blend}(x) + R(x))$

where, $P_{model}(x)$ is the calibrated XGBoost probability, $A(x)$ is the normalized Isolation Forest anomaly score, and $R(x)$ is the bounded rule-based term (maximum 0.12). The IF score normalization used the fifth percentile of FIT decision scores and the 99th percentile of FIT positive anomaly strengths for scaling, with clipping to [0,1]. The coefficient $\alpha = 0.8$ was fixed for this implementation, and the final score was clipped to [0,1].

G. Explainability

In order to increase the transparency of the model, SHAP analysis was applied to identify how each feature affected XGBoost predictions (Fig. 2). For each flagged event, the framework identifies the most influential features that increased or decreased the classifier prediction. This provides feature-level context for reviewing alerts. Event-level sensitivity and stability checks were also performed on XGBoost explanations.

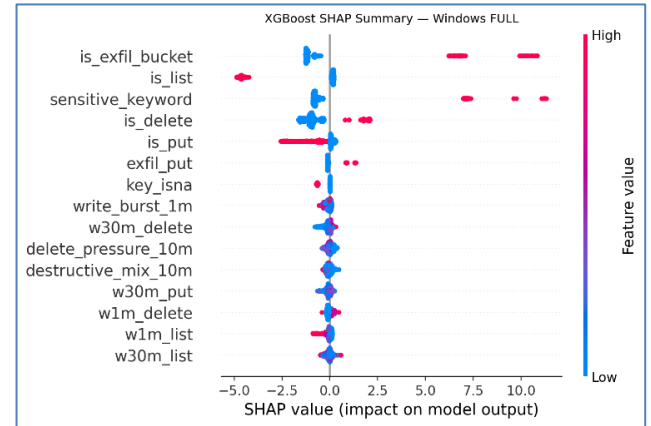


Fig. 2. SHAP summary plot of influential XGBoost features on the held-out sample.

H. Evaluation Design

The evaluation was carried out through multiple experiments to investigate discrimination, interpretability, and practical alert behavior. A chronological held-out test set was used to calculate PR-AUC, ROC-AUC, precision, recall, and F1-score. The framework was also applied to the full dataset for descriptive alert counts, without presenting full-data classification statistics as independent test performance. Ablation examined burst, sensitive-object, and temporal-window configurations. Three chronological cross-validation folds evaluated within-training stability, while 50,000 paired clean and perturbed records measured alert decision sensitivity. A mathematical prevalence analysis explored expected alert behavior at different assumed attack rates. Matched component and classical baselines were evaluated on the same held-out TEST partition, and event-level SHAP perturbations were used to assess explanation behavior. Two exploratory rule-query baselines were also executed through Elasticsearch on the same 35,963 held-out TEST event identifiers: static object/bucket indicators, and those indicators combined with five-minute per-user/IP activity counts.

IV. RESULTS

A. Main Model Performance

The key results of discrimination for this case study section are obtained on a held-out test set. The trained and calibrated model showed good discrimination; PR-AUC was 0.9324, and ROC-AUC was 0.9827 on 35,963 temporally held-out events.

Under the monitoring mode configuration, the model demonstrated precision of 0.9677, recall of 0.7930, and F1-

score of 0.8717. Under the forensic mode configuration, when a stricter threshold was used, precision was improved to 0.9822, recall declined to 0.7844, and the F1-score became 0.8722 (Table III). Mean PR-AUC and ROC-AUC were also high in three expanding-window chronological cross-validation folds, at 0.9168 and 0.9790, respectively. The held-out test set was not used in these folds (Fig. 3 and 4).

TABLE III. MAIN RESULTS FOR THE PROPOSED FRAMEWORK

Mode	Threshold	Precision	Recall	F1	PR-AUC	ROC-AUC
Monitoring	0.6624	0.9677	0.7930	0.8717	0.9324	0.9827
Forensic	0.7003	0.9822	0.7844	0.8722	0.9324	0.9827

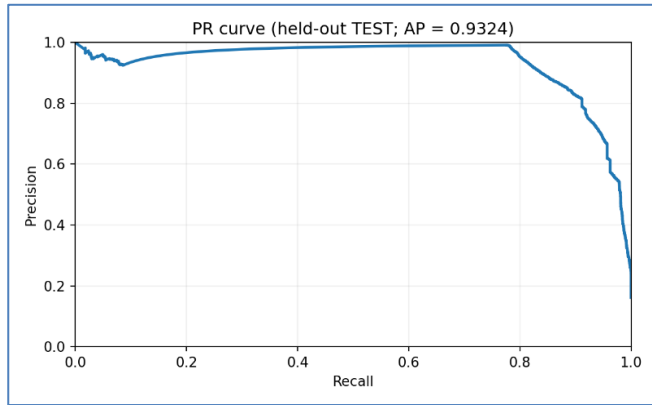


Fig. 3. Precision-recall curve for the held-out test set.

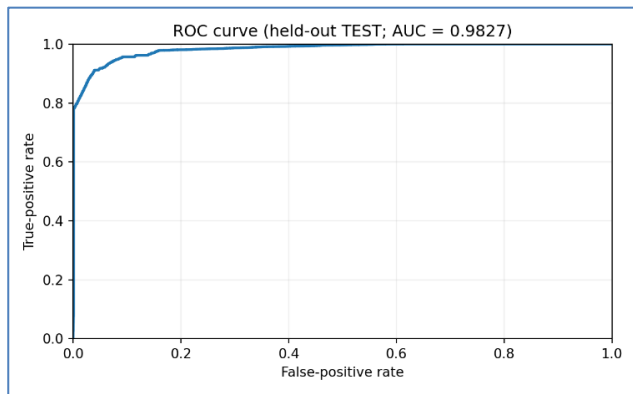


Fig. 4. ROC curve for the held-out test set.

B. Full Data Reporting Behavior

Of the 169,986 events, 27,107 (15.95%) were labeled as suspicious and 142,879 (84.05%) as normal based on heuristic rules. The monitoring configuration generated 23,281 alerts after applying the trained pipeline to the complete dataset (169,986 events), corresponding to an alert rate of 13.70%. The forensic configuration generated 22,219 alerts, corresponding to 13.07% (Table IV). These figures describe overall alert volumes, including training events, and are not used as an independent performance estimate.

This is operationally meaningful: the monitoring mode produces broader alert coverage, while the forensic mode uses a more conservative decision threshold.

TABLE IV. FULL-DATASET DESCRIPTIVE ALERT VOLUMES

Mode	Rows	Alerts	Alert rate
Monitoring	169,986	23,281	0.1370
Forensic	169,986	22,219	0.1307

C. Ablation Analysis

There is an interesting pattern in the ablation results. Removing burst features produced a similar monitoring F1-score to FULL (0.8725 versus 0.8717), while removing sensitive-object features resulted in a substantial performance decrease (monitoring F1 = 0.6108; forensic F1 = 0.2436). Removing window-based temporal features led to a modest monitoring F1 change (0.8693 versus 0.8717). These findings emphasize the importance of sensitivity-related signals for the synthetic labeling task, while the predictive contribution of burst and rolling-window groups was limited under the tested configurations (Table V).

TABLE V. ABLATION STUDY SUMMARY

Feature set	Features	Mon. F1	For. F1	PR-AUC	ROC-AUC
FULL	63	0.8717	0.8722	0.9324	0.9827
NO_BURST_STRICT	45	0.8725	0.8725	0.9567	0.9868
NO_SENSITIVE_STRICT	55	0.6108	0.2436	0.6461	0.8873
NO_WINDOWS_STRICT	13	0.8693	0.8639	0.9539	0.9855

The reported ablations used independently retrained models and tuning-derived thresholds. The reduced configurations can also modify their relevant rule adjustments; therefore, the comparison describes complete configuration sensitivity rather than a single-feature causal effect. No statistical significance is claimed for the small F1 differences.

D. Scenario-Based Behaviour

Evaluation under different assumed attack prevalences provides an analytical view of alert behavior. Using the held-out monitoring sensitivity and false-positive rate, projected precision rises as the assumed proportion of suspicious events increases. These are mathematical projections for 20,000 hypothetical events at each prevalence, not newly generated attack datasets or independently observed detection results (Table VI).

TABLE VI. ANALYTICAL MONITORING-MODE PREVALENCE SENSITIVITY (20,000 HYPOTHETICAL EVENTS)

Assumed attack rate	Projected alerts	Projected precision	Assumed recall	Projected F1
0%	102	N/A	N/A	N/A
20%	3,254	0.975	0.793	0.875
50%	7,981	0.994	0.793	0.882
80%	12,708	0.998	0.793	0.884
100%	15,859	1.000	0.793	0.885

E. Robustness to Noisy Data

On 50,000 paired clean and perturbed synthetic events, monitoring generated 6,516 alerts (13.03%) before perturbation and 5,977 alerts (11.95%) afterward; forensic mode generated 6,360 alerts (12.72%) and 5,834 alerts (11.67%), respectively. The perturbations changed 937 monitoring decisions (1.87%) and 754 forensic decisions (1.51%). These findings characterize decision sensitivity under controlled field changes, rather than independently verified robustness to real attacks (Table VII).

TABLE VII. PAIRED 50,000-EVENT PERTURBATION EXPERIMENT

Mode	Clean alerts	Perturbed alerts	Changed decisions	Perturbed alert rate
Monitoring	6,516	5,977	937	0.1195
Forensic	6,360	5,834	754	0.1167

F. Explainability Evaluation

To evaluate explanations beyond the global feature summary, event-level SHAP sensitivity and stability were tested on the trained XGBoost component (Fig. 5). In 100 cases, perturbing the three most influential features caused a larger margin change than perturbing the three least influential features in 86% of cases. The median absolute margin shift was 4.6143 for the former and 0.0 for the latter.

Across 300 small-jitter comparison pairs, top-ranked feature agreement was 97.33%, mean top-three overlap was 99.22%, and exact top-three set agreement was 97.67%. These figures indicate stable feature rankings under the specified perturbation protocol.

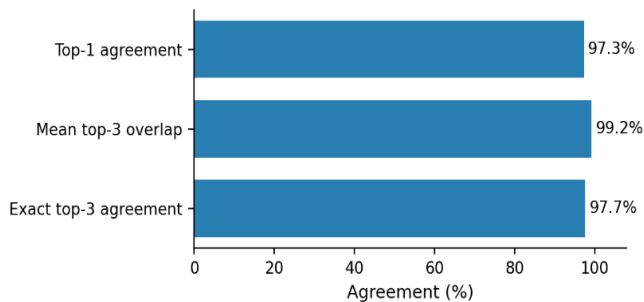


Fig. 5. Stability of event-level XGBoost SHAP feature rankings under small timestamp perturbations.

SHAP explains the XGBoost component rather than the entire hybrid score. Its benefit to investigators remains a topic for future user-centered evaluation.

As an illustrative flagged TEST event (source_line 134136), the saved Windows model recorded calibrated XGBoost probability 0.9474, normalized Isolation Forest score 0.4742, and rule adjustment 0.1000, producing final risk 0.9527 according to (1)–(2). This score decomposition makes the three components auditable for an individual event; SHAP attribution itself remains specific to the XGBoost component (Table VIII).

TABLE VIII. EVENT-LEVEL SHAP VERIFICATION

Test	Result
Top-three feature perturbation more influential	86.0% of 100 cases
Top-ranked feature stability	97.33%
Mean top-three feature overlap	99.22%
Exact top-three feature set agreement	97.67%

G. Baseline Comparison

As a baseline comparison, Random Forest and Logistic Regression were evaluated alongside XGBoost-only, Isolation Forest-only, rule-only, and component-removal variants on the same chronological held-out test set. Supervised operating thresholds were selected using TUNE, while the rule-only result is a generator-concordance diagnostic rather than independently verified malicious-event detection.

Random Forest achieved 0.9860 ROC-AUC and 0.9566 PR-AUC; Logistic Regression achieved 0.9868 ROC-AUC and 0.9578 PR-AUC. XGBoost-only reached 0.9878 ROC-AUC and 0.9589 PR-AUC, with a monitoring F1-score of 0.8723. The hybrid framework achieved an F1-score of 0.8717, showing comparable classification performance under the synthetic benchmark.

Although several baselines achieved comparable performance, the proposed framework combines calibrated classification with anomaly scoring, bounded rule-based reasoning, and SHAP explanations of XGBoost predictions in a unified forensic-oriented interface. The component-removal rows show closely matched F1-scores, while the rule-only result requires cautious interpretation because the same behavioral indicators informed synthetic labelling (Table IX).

Overall, the hybrid framework offers a structured, interpretable approach to cloud forensic alert triage. Its value in this study lies in the integrated workflow rather than a demonstrated improvement in predictive accuracy over XGBoost alone.

TABLE IX. MATCHED HELD-OUT BASELINE COMPARISON

Model	Monitoring F1	ROC-AUC	PR-AUC
Hybrid framework	0.8717	0.9827	0.9324
XGBoost-only	0.8723	0.9878	0.9589
Random Forest	0.8697	0.9860	0.9566
Logistic Regression	0.8688	0.9868	0.9578
IF-only (fixed normalization)	0.2788	0.5213	0.1711
Rule-only (label-concordance diagnostic)	0.8904	0.9012	0.8344
XGBoost + IF (no rule boost)	0.8720	0.9858	0.9501
XGBoost + rules (no IF)	0.8718	0.9844	0.9417

The two component-removal variants retain the FULL monitoring cutoff; their scores are diagnostic rather than independently optimized operating-point comparisons. All rows use the same held-out TEST split.

For a platform-oriented baseline, the same 35,963 TEST events (5,825 heuristic-labeled suspicious events) were evaluated using two Elasticsearch rule-query policies. Static indicators included writes to an exfiltration-designated bucket and sensitive-object operations; a second policy also used five-minute LIST, PUT, and destructive-operation burst counts. Table X compares these query results with the Windows-trained FULL framework on the identical event identifiers and labels.

TABLE X. ELASTICSEARCH SEARCH-RULE BENCHMARK ON MATCHED HELD-OUT TEST

<i>Method</i>	<i>Alerts</i>	<i>Precision</i>	<i>Recall</i>	<i>F1</i>
FULL (Monitoring)	4,773	0.9677	0.7930	0.8717
FULL (Forensic)	4,652	0.9822	0.7844	0.8722
ES static indicators	4,498	1.0000	0.7722	0.8715
ES static + 5-min	4,637	0.9791	0.7794	0.8679

Note. Elasticsearch precision/recall/F1 measures the two specified search-rule policies, not the performance of the entire ELK stack. The rules were evaluated on synthetic heuristic-derived labels that overlap some indicator definitions; results are exploratory, not independent incident validation. The FULL rows use the final Windows-trained estimator and its original TUNE-selected thresholds.

V. DISCUSSION

The same-TEST Elasticsearch query comparison also showed competitive results: monitoring F1 was 0.8717 for FULL, 0.8715 for the static-query baseline, and 0.8679 for the query policy with five-minute bursts. Beyond binary alerts, the framework provides calibrated risk scores, two decision modes, and XGBoost SHAP explanations.

The experimental results support several observations about cloud digital forensics. Firstly, the proposed system indicates that semantically meaningful features can be informative for the evaluated synthetic forensic detection task. Semantically meaningful features related to sensitive objects, credentials, financial files, backups, and possible data exfiltration were influential for the classifier in this setting. The ablation study also supports this conclusion, as removing sensitive-object features caused a substantial performance loss, highlighting the contribution of semantic indicators to this benchmark.

Secondly, temporal behavior analysis provides context for reviewing sequences of cloud events. In fact, although individual events may not appear suspicious, aggregating activities across multiple time windows can reveal behavioral patterns like burst operations, repeated access attempts, and destructive sequences, which are difficult to reveal by analyzing events separately. This motivation coincides with the basic principle of conducting forensic investigations: event context can usefully complement individual event interpretation [7], [8].

There are also some practical benefits to the hybrid approach. First, XGBoost can deal well with complex nonlinear relations in structured cloud logs, whereas Isolation

Forest provides an additional unsupervised anomaly score. The additional rule-based layer introduces forensic knowledge into the model by focusing on combinations of risky behaviors, but does not dominate the learning process. Thus, the presented hybrid framework balances statistical learning with expert-driven analysis and supports a flexible approach to forensic-oriented alert analysis.

Furthermore, the use of SHAP for explainability is another significant contribution in the area. In addition to the classification scores, the framework identifies the features that most strongly influenced XGBoost predictions. Such explanations may support analyst review and evidence interpretation and meet the current needs for explainable security systems [6], [25], [26].

Lastly, there are two operating modes of the framework, which allow it to be used in different forensic scenarios. In particular, the monitoring mode focuses on a high recall rate to support continuous security monitoring and early detection, whereas the forensic mode uses a high precision rate in order to reduce the number of false-positive detections. Use of two thresholds enables the model to adjust to varying operational needs.

LIMITATIONS

There are several limitations that should be taken into account while analyzing the findings of the research. First, the framework was evaluated on a simulated AWS CloudTrail-like dataset that was designed to mimic actual cloud activity and attacks. While this dataset retains many forensic characteristics, it cannot fully reflect the complexity, scale, and diversity of real cloud infrastructure. Thus, the use of large-scale real-world CloudTrail datasets is needed to validate the generalizability of the proposed framework.

Second, the present study mainly considers object-storage activities without including other sources of cloud data, such as IAM-related activity, AWS VPC Flow Logs, CloudWatch Logs, and other cloud service telemetry. The use of multiple cloud telemetry sources would enable a more thorough understanding of attacker behavior patterns and improve forensic investigations.

Third, the synthetic labels were derived in part from predefined behavioral heuristics. Some of those indicators overlap the engineered features, meaning that held-out classification scores may reflect learned labeling conventions and do not constitute independent evidence of attack identification. Matched component tests did not establish a predictive gain from the additional anomaly component. Independent, forensically annotated cloud traces would strengthen the generalization assessment.

Lastly, while SHAP offers feature-level explanations of the XGBoost classifier, there was no formal user study or usability testing with digital forensic investigators within this research project. Further investigation is needed to determine how well these explanations support real-world forensic decision-making and incident investigation workflows.

VI. CONCLUSION AND FUTURE WORK

This study proposed a framework for explainable machine learning for conducting cloud forensics of object storage audit logs of the AWS-like infrastructure. This framework, relying on semantically motivated feature engineering, calibration of the gradient-boosted classifier, scoring anomalies, bounded rules, and explaining every alert, showed strong held-out discrimination on a controlled synthetic benchmark, with ROC-AUC = 0.9827 and PR-AUC = 0.9324. The results indicate that sensitive-object indicators were particularly influential in the controlled classification task, while the two operating thresholds supported different monitoring and forensic precision levels. Integrated approaches can combine behavioral signals with interpretable classifier outputs. In general, the study supports the claim that explainable analytics may be useful for cloud investigation triage, while further real-world validation remains necessary.

Based on the outcomes of this study, it is possible to consider explainable AI a promising component of cloud forensic investigation support. Additionally, for the purpose of the baseline comparison, Random Forest and Logistic Regression models have been implemented as well. The results showed that the proposed framework provides competitive synthetic-benchmark performance, as well as an integrated workflow in terms of explainability and forensic-oriented characteristics.

However, the implementation of the framework in real cloud environments through its API, with real-time monitoring and continuous log collection, and validation using larger and more varied real-world datasets remain future research directions. Another future research direction will be the implementation of some additional sources of data, such as IAM and network logs, into the framework to expand the scope of analysis. Besides, future research may focus on adaptive learning techniques to adapt to emerging threats, as well as testing the system with forensic investigators to evaluate its practical usability and extending the comparison to additional baseline models.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA, for funding this research work through the project number “NBU-FFRGS-2026”.

DECLARATIONS

The authors declare that they have no conflicts of interest regarding the present study.

DATA AND CODE AVAILABILITY

The synthetic dataset and code used in this study are available from the corresponding author upon reasonable request.

REFERENCES

- [1] A. K. R. Avula, “Understanding cloud computing: How data storage works in the cloud,” *International Journal for Multidisciplinary Research*, vol. 6, no. 6, pp. 1–7, 2024, doi: 10.36948/ijfmr.2024.v06i06.33472.
- [2] B. Yankson and A. Davis, “Analysis of the current state of cloud forensics: The evolving nature of digital forensics,” in *Proc. IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, Abu Dhabi, UAE, 2019, pp. 1–8, doi: 10.1109/AICCSA47632.2019.9035336.
- [3] V. Ganapathy, “Audit trail and transparency in cloud-based auditing,” *Edumania—An International Multidisciplinary Journal*, vol. 2, no. 2, pp. 118–145, 2024, doi: 10.59231/edumania/9042. [Online]. Available: <https://doi.org/10.59231/edumania/9042>
- [4] Z. Chen, H. Tian, J. Lu, Y. Cai, T. Wang, and Y. Chen, “Operational-Behavior Auditing in Cloud Storage,” in *Cloud Computing, Security, Privacy in New Computing Environments*, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 197. Cham, Switzerland: Springer, 2018, pp. 162–172, doi: 10.1007/978-3-319-69605-8_15.
- [5] X. Ma, Y. Li, J. Keung, X. Yu, H. Zou, Z. Yang, F. Sarro, and E. T. Barr, “Practitioners’ expectations on log anomaly detection,” *IEEE Transactions on Software Engineering*, vol. 51, no. 9, pp. 2455–2471, Sep. 2025, doi: 10.1109/TSE.2025.3586700.
- [6] S. W. Hall, A. Sakzad, and K.-K. R. Choo, “Explainable artificial intelligence for digital forensics,” *WIREs Forensic Science*, vol. 4, no. 2, Art. no. e1434, 2022, doi: 10.1002/wfs2.1434. [Online]. Available: <https://doi.org/10.1002/wfs2.1434>
- [7] K. Kent, S. Chevalier, T. Grance, and H. Dang, *Guide to Integrating Forensic Techniques into Incident Response*, NIST Special Publication 800-86. Gaithersburg, MD, USA: National Institute of Standards and Technology, Aug. 2006, doi: 10.6028/NIST.SP.800-86. [Online]. Available: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-86.pdf>
- [8] International Organization for Standardization and International Electrotechnical Commission, *Information technology—Security techniques—Guidelines for identification, collection, acquisition and preservation of digital evidence*, ISO/IEC 27037:2012, Oct. 2012. [Online]. Available: <https://www.iso.org/standard/44381.html>
- [9] M. Akbar, M. Suaib, M. S. Husain, and S. Shukla, “A Compendium of Cloud Forensics,” in *Critical Concepts, Standards, and Techniques in Cyber Forensics*, M. S. Husain and M. Z. Khan, Eds. Hershey, PA, USA: IGI Global, 2020, pp. 215–227, doi: 10.4018/978-1-7998-1558-7.ch012.
- [10] S. S. Finta, “Digital forensics. The challenges of searching computer systems and authorized access to a computer system,” *Romanian Journal of Forensic Science*, vol. 26, no. 143, pp. 306–314, 2025.
- [11] M. Dwivedi, P. Pali, J. Choubey, A. Mishra, and H. Hedau, “Investigating challenges and future directions in cloud computing forensics: A comprehensive review,” *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 12, no. 5, pp. 8044–8049, May 2023, doi: 10.15680/IJIRSET.2023.1205495.
- [12] M. Herman et al., “NIST cloud computing forensic science challenges,” *NISTIR 8006*, National Institute of Standards and Technology, Aug. 2020, doi: 10.6028/NIST.IR.8006.
- [13] F. H. Deandra, Sherly, and I. Muda, “Advancing digital forensic investigations: Addressing challenges and enhancing cybercrime solutions,” *World Journal of Information Technology*, vol. 3, no. 1, pp. 10–15, Jan. 2025, doi: 10.61784/wjit3018.
- [14] A. M. Alenezi, “Digital and cloud forensic challenges,” *arXiv:2305.03059*, May 2023, doi: 10.48550/arXiv.2305.03059. [Online]. Available: <https://arxiv.org/abs/2305.03059>
- [15] A. O. Ojo and M. Benmubarak, “Investigating advanced persistent threat tactics in cloud environments: A forensic study of AWS CloudTrail log data,” *International Journal of Innovative Science and Research Technology*, vol. 10, no. 7, pp. 3170–3176, 2025, doi: 10.38124/ijisrt/25jul1786.
- [16] Amazon Web Services, “Enabling CloudTrail event logging for S3 buckets and objects,” *Amazon Simple Storage Service User Guide*. [Online]. Available: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/enable-cloudtrail-logging-for-s3.html>. [Accessed: Jun. 30, 2026].

- [17] X. Wu, "Review of anomaly detection based on log analysis," *International Journal of Advanced Network, Monitoring and Controls*, vol. 5, no. 4, pp. 40–49, 2020, doi: 10.21307/ijanmc-2020-036.
- [18] J. Dykstra and A. T. Sherman, "Design and implementation of FROST: Digital forensic tools for the OpenStack cloud computing platform," *Digital Investigation*, vol. 10, pp. S87–S95, 2013, doi: 10.1016/j.diin.2013.06.010.
- [19] D. Pawlaszczyk, M. Bochmann, P. Engler, C. Klaver, and C. Hummert, "API-based evidence acquisition in the cloud: A survey," *Open Research Europe*, vol. 2, Art. no. 69, 2022, doi: 10.12688/openreseurope.14784.1.
- [20] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.
- [21] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in *Proc. 2008 Eighth IEEE Int. Conf. Data Mining (ICDM)*, 2008, pp. 413–422, doi: 10.1109/ICDM.2008.17.
- [22] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, vol. 30, pp. 4765–4774, 2017.
- [23] K. Fatema, S. K. Dey, M. Anannya, R. T. Khan, M. M. Rashid, C. Su, and R. Mazumder, "Federated XAI IDS: An Explainable and Safeguarding Privacy Approach to Detect Intrusion Combining Federated Learning and SHAP," *Future Internet*, vol. 17, no. 6, Art. no. 234, 2025, doi: 10.3390/fi17060234.
- [24] T. B. Ogunseyi and G. Thiyagarajan, "An Explainable LSTM-Based Intrusion Detection System Optimized by Firefly Algorithm for IoT Networks," *Sensors*, vol. 25, no. 7, Art. no. 2288, 2025, doi: 10.3390/s25072288.
- [25] S.-U.-R. Sahito et al., "Explainable AI (XAI) for security decisions to mitigate cybersecurity attacks," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 55, no. 2, pp. 214–223, 2025, doi: 10.37934/araset.55.2.214223.
- [26] A. Bhatia and A. Kumar, "AI explainability and trust in cybersecurity operations," in *Deep Learning Innovations for Securing Critical Infrastructures*, R. Kumar, S.-L. Peng, P. Jain, and A. A. Elngar, Eds. Hershey, PA, USA: IGI Global Scientific Publishing, 2025, pp. 57–74, doi: 10.4018/979-8-3373-0563-9.ch004.
- [27] L. Girish and S. K. N. Rao, "Anomaly detection in cloud environment using artificial intelligence techniques," *Computing*, vol. 105, no. 3, pp. 675–688, 2023, doi: 10.1007/s00607-021-00941-x.
- [28] F. Shahzad, A. Mannan, A. R. Javed, A. S. Almadhor, T. Baker, and D. Al-Jumeily OBE, "Cloud-based multiclass anomaly detection and categorization using ensemble learning," *Journal of Cloud Computing*, vol. 11, Art. no. 74, 2022, doi: 10.1186/s13677-022-00329-y.
- [29] S. Rajak, "AI-based anomaly detection for cloud logs and distributed traces," *International Journal of Emerging Research in Engineering and Technology*, vol. 7, no. 2, pp. 19–26, Apr. 2026, doi: 10.63282/3050-922X.IJERET-V7I2P103.
- [30] P. Hermosilla, S. Berrios, and H. Allende-Cid, "Explainable AI for forensic analysis: A comparative study of SHAP and LIME in intrusion detection models," *Applied Sciences*, vol. 15, no. 13, Art. no. 7329, 2025, doi: 10.3390/app15137329. [Online]. Available: <https://doi.org/10.3390/app15137329>
- [31] J. H. Vane, A. Thorne, and L. M. Sterling, "Explainable AI (XAI) in cloud intrusion detection systems," *Institute of Applied Cybersecurity and Cloud Computing*, Apr. 15, 2026.
- [32] E. F. Papavasileiou and M. Oubibi, "Explainable AI for cloud anomaly detection: Making neural network-based threat recognition transparent and auditable," Nov. 2025. [Online]. Available: ResearchGate.
- [33] A. A. Aljuhani and F. H. Aljuhani, "Cloud-Based Intelligent Surveillance for Digital Forensics: AI-Enhanced Criminal Investigations," *International Journal of Advanced Computer Science and Applications*, vol. 17, no. 5, pp. 116–123, 2026, doi: 10.14569/IJACSA.2026.0170513.
- [34] A.-R. Al-Ghuwairi, Y. Sharrab, D. Al-Fraihat, M. AlElaimat, A. Alsarhan, and A. Algarni, "Intrusion detection in cloud computing based on time series anomalies utilizing machine learning," *Journal of Cloud Computing*, vol. 12, no. 1, Art. no. 127, 2023, doi: 10.1186/s13677-023-00491-x.
- [35] H. Studiawan and F. Sohel, "Anomaly detection in a forensic timeline with deep autoencoders," *Journal of Information Security and Applications*, vol. 63, Art. no. 103002, Dec. 2021, doi: 10.1016/j.jisa.2021.103002.
- [36] D. Alharthi and I. R. K. Garcia, "Cloud investigation automation framework (CIAF): An AI-driven approach to cloud forensics," *arXiv:2510.00452*, Oct. 2025, doi: 10.48550/arXiv.2510.00452.
- [37] A. S. P. and T. S. P., "Explainable AI (XAI) for insider threat detection: Balancing security and transparency in cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 37, no. 25–26, Art. no. e70390, 2025, doi: 10.1002/cpe.70390.
- [38] P. K. R. Idamakanti, "Cloud network anomaly detection using federated learning and explainable AI," *International Journal on Science and Technology*, vol. 16, no. 3, pp. 1–17, 2025, doi: 10.71097/IJSAT.v16.i3.7336.
- [39] D. V. Talati, "Transparency and interpretability in cloud-based machine learning with explainable AI," *International Journal of Multidisciplinary Research in Science, Engineering and Technology*, vol. 7, no. 7, pp. 11823–11831, Jul. 2024, doi: 10.15680/IJMRSET.2024.0707002.
- [40] B. Goyal, "Explainable AI (XAI) for cloud-based enterprise applications: Building trust and transparency in AI-driven decisions," *Journal of International Crisis and Risk Communication Research*, vol. 8, no. S9, pp. 63–71, Sep. 2025, doi: 10.63278/jicrcr.vi.3231.
- [41] H. R. Devarajan, S. B. Singh, and E. Howard, "Explainable AI for cloud-based machine learning: Interpretable models and transparency in decision making," *Tuijin Jishu/Journal of Propulsion Technology*, vol. 45, no. 2, pp. 2886–2894, Apr. 2024, doi: 10.52783/tjpt.v45.i02.6376.
- [42] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Computers & Security*, vol. 86, pp. 147–167, Sep. 2019, doi: 10.1016/j.cose.2019.06.005.
- [43] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP)*, 2018, pp. 108–116, doi: 10.5220/0006639801080116.