# X-SIEM Framework: Integrating Rule-Based, ML, and LLMs for Cyber Threat Intelligence

Md Fahim Al Shihab, Sabbir Ahmed Al Seum, Md. Atikur Rahman,
Ragib Nadim, Albar Hossain Rafi, Tonny Shekha Kar
Department of Computer Science and Engineering,
American International University–Bangladesh (AIUB), Dhaka, Bangladesh
Emails: 22-46945-1@student.aiub.edu, 22-47196-1@student.aiub.edu, 22-47944-2@student.aiub.edu,
23-50683-1@student.aiub.edu, 21-45093-2@student.aiub.edu, tonny.kar@aiub.edu

***Abstract*—Security Operations Centers (SOCs) are increasingly challenged by alert fatigue, with traditional Security Information and Event Management (SIEM) systems generating thousands of daily alerts, a significant portion of which are false positives. Concurrently, the adoption of machine learning (ML) for threat detection has introduced a "black box" problem, where the lack of explainability hinders analyst trust and compliance with regulations. This paper presents a novel three-layer hybrid SIEM framework designed to bridge the gap between detection accuracy and operational interpretability. The framework synergistically combines a high-speed rule engine, a hybrid ML ensemble (Random Forest and Isolation Forest), and a fine-tuned Large Language Model (LLM) for explanation and response generation. The framework was evaluated on the CICIDS-2017 and CICIDS-2018 benchmark datasets, achieving a 98.7% precision and an 82% reduction in false positives compared to a baseline rule-based system. Furthermore, the LLM-generated explanations, mapped to the MITRE ATT&CK framework, reduced the mean time to respond (MTTR) by 58% in simulated analyst workflows. Statistical significance testing confirmed our results (p ¡ 0.01), and comprehensive ablation studies validated our architectural choices. Our results demonstrate that this integrated approach not only enhances the accuracy of threat detection but also provides the actionable, transparent intelligence necessary for modern cybersecurity operations.**

***Index Terms*—Cybersecurity, SIEM, Explainable AI (XAI), Large Language Models (LLMs), Threat Intelligence, Machine Learning, Hybrid Detection**

## I. Introduction

Modern digital companies face an endless barrage of sophisticated cyberattacks. Security Operations Centers (SOCs) are leading the defence, but they are struggling under the weight of overwhelming data volumes. A typical SOC can handle more than 10,000 daily alerts [1], leading to high analyst exhaustion and a likely miss of real threats in background noise. Traditional SIEM systems, although central to SOC operations, frequently suffer from high false-positive rates (up to 40%) and an inability to detect new zero-day attacks [2]. To resolve this, next-generation SIEMs have incorporated machine learning (ML) models, which have demonstrated significant success in reporting anomalous patterns consistent with attacks [3]. However, this has created new challenges: the **interpretability paradox**. Complex models like deep neural networks or large ensembles are **black boxes**, producing alerts without a clear rationale. The lack of transparency in automated systems undermines analyst trust, complicates forensic investigations, and poses compliance risks under regulations like the **GDPR**, which mandate a right to explanation. [4]. This paper closes the gap in imperative detection of explainable and actionable intelligence. Unlike previous work that focuses primarily on detection accuracy, this concept incorporates three synergistic detection modalities, prioritizing explainability as a crucial aspect, and directly tackles the interpretability contradiction in machine learning-based security systems. We present a new three-layer hybrid SIEM design that integrates:

1) A fast and adaptive **rule engine** for known threats.
2) A hybrid **ML ensemble** for detecting both known and unknown anomalies.
3) A fine-tuned **Large Language Model (LLM)** to translate complex alerts into human-readable explanations and generate actionable response plans.

The primary contribution is a framework that not only achieves state-of-the-art detection accuracy but also makes security alerts transparent and immediately useful for SOC analysts of all skill levels.

The rest of the paper is organized as follows: Sec. II discusses some preliminary works. Then, Sec. III explains the architecture of the proposed system, followed by Sec. IV that discusses the evaluation of our framework. Then Sec. V provides the discussion. Finally, Secc. VI concludes the paper.

## II. Related Work

This study is based on five distinct domains of cybersecurity research.

### A. Rule-Based Detection Systems

Traditional SIEMs such as IBM QRadar and Splunk rely on correlation rules to detect threat patterns [5]. An open-source framework such as the Sigma rules has allowed the sharing of detection logic across platforms [6]. Although effective for known attack signatures, these systems are inflexible; they fail to detect novel attacks and require constant manual tuning by security experts.

### B. Machine Learning in SIEM

Other ML algorithms have been successfully applied in network intrusion detection by researchers. Uppal et al. [3]

demonstrated the dominance of tree-based ensemble techniques such as Random Forest and XGBoost on the CICIDS-2017 dataset. Unsupervised methods such as isolation forest and one-class SVMs have been applied to detect zero-day anomalies without labeling [7]. However, most of the work in these studies revolved around detection metrics (precision and recall) instead of the real-world necessity for explainability in an SOC environment.

### C. Explainable AI in Cybersecurity

Explainable AI (XAI) usage in security analytics continues to increase with each passing year. SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations) are among the most frequently used post-hoc techniques for explaining complex ML predictions. For instance, Sarhan et al. [8] used SHAP values for explaining Random Forest In network intrusion detection, which allows analysts to discover feature importance behind the alerts. Similarly, Alodibat et al. [9] integrated LIME with anomaly detection models to create localized versions of SOC operator mappings. However, Post hoc methods lack operational context and do not scale well for real time attribution. This framework resolves these issues by generating explanations within the detection pipeline, utilizing a trained LLM for generating contextual natural language reports aligned with the MITRE ATT&CK framework[10]. This design moves explainability from a **secondary afterthought** to a first-class feature of SIEM operations.

### D. Hybrid Detection Systems

Recent articles have explored hybrid detection frameworks, blending rule-based and machine learning detectors. For example, the HAEnID model blends ensemble learning and LIME and SHAP explanations, increasing accuracy without sacrificing interpretability [11]. HuntGPT further combines anomaly detection, explainable AI, and large language models (LLMs) for analyst-friendly insights [12]. Houssel et al. [13], in turn, contrast LLMs for intrusion and explanation, while Baral et al. [14] propose an IoT security framework blending ML, explainability, and LLMs. These explorations detail progress in hybrid detection and interpretability, though no paper covers rule-based and machine learning detection and natural language explanation under a comprehensive SIEM pipeline—our principal contribution.

### E. LLMs in Cybersecurity

The Large Language Model's introduction opened new cybersecurity automation frontiers. Works, e.g., MITRE CALDERA, have researched AI's use for automated adversary emulation [15]. Most recently, researchers have looked at LLM usage for condensing threat intelligence reports and supporting malware analysis [16]. Most of these recent activities employ general-purpose LLMs, however, hence limiting contextual anchoring for use cases in the SOC. Our solution advances the state of affairs by fine-tuning a small, open-weight LLM on cybersecurity data, specifically, so as to generate structure, context-sensitive explanations and response playbooks.

## III. System Architecture

The proposed framework consists of three parallel processing layers whose outputs are synthesized by a consensus mechanism, as illustrated in Fig.1.

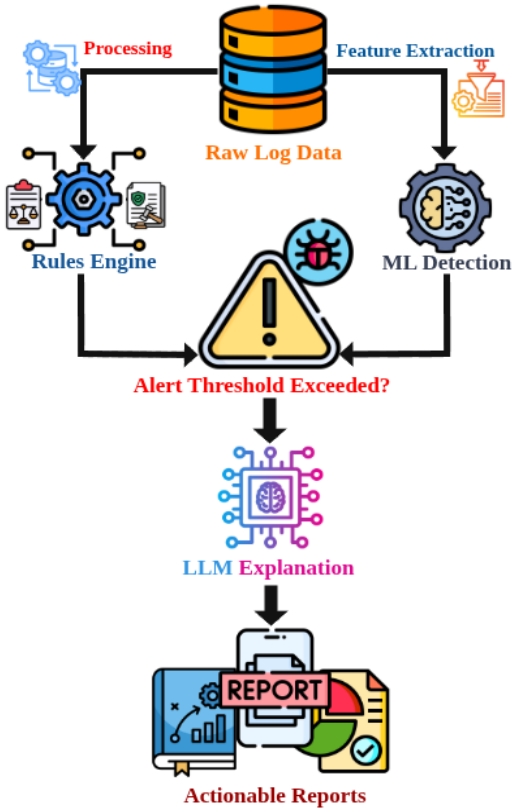


Fig. 1: The three-layer hybrid SIEM architecture. Log data is processed in parallel by the Rule Engine and ML Ensemble. Positive detections are sent to the LLM Explanation Engine, which generates actionable intelligence for the SOC analyst.

### A. Layer 1: Adaptive Rule Engine

This layer constitutes the ground-level defence mechanism of the framework under consideration, based on the concept of deterministic, signature-based detection. It utilizes a high-speed correlation engine consuming logs from heterogeneous sources—firewalls, endpoint agents, DNS servers, cloud telemetry—and compares them against a pre-curated library of Sigma rules [17]. These Sigma rules are platform-agnostic YAML-based definitions containing encapsulated threats in a vendor-agnostic format, and thus, become appropriate for the new heterogeneous environment of the future.

*1) **Rule Structure and Matching**:* All Sigma rules adhere to a formal schema of metadata (title, description, references), event selection and conditions of the detection logic, and of severity classification fields. At ingest, raw logs get normalized first to a standard event structure (CEF) and then become subject to fast-pattern matching by an in-memory rule engine. Here, multi-field checks occur through application of Boolean logic, of regular expressions, and of time-window conditions.

The simple rule for finding brute-force attempts through an SSH service might then be specified thus:

```
title: SSH Brute Force Detection
logsource:
  category: authentication
detection:
  selection:
    EventID: 4625
    LogonType: 3
    TargetPort: 22
  condition: selection | count() by src_ip > \(T_{
      ↪ eff}\)
```

The condition clause above enforces a dynamic threshold (explained below) to trigger alerts based on failed login attempts exceeding a predefined limit.

*2) **Dynamic Thresholding Policy**:* To alleviate the brittleness of static thresholds, A dynamic rule-augmentation mechanism is introduced. Threshold values are no longer hard-coded but instead adapted at runtime based on past baselines and situational conditions. For example, during office hours, the number of failed login attempts of privileged users is allowed to be lower than the system accounts attempts permitted during the time of the maintenance window.

Threshold policies are enforced using an adaptive scoring system:

$$T_{eff} = \mu_n + \alpha \cdot \sigma_n \tag{1}$$

where $\mu_n$ and $\sigma_n$ are the historical mean and standard deviation of event frequency for a given rule over a rolling time window $n$, and $\alpha$ is a sensitivity coefficient (empirically tuned per rule class). The rules engine periodically recalculates these statistics from a metadata cache populated by the log aggregator.

### B. Layer 2: Hybrid ML Ensemble

The architecturally proposed solution's second level is responsible for learning and generalizing patterns from historical network telemetry by use of machine learning. It encompasses both supervised and unsupervised models for discovering known and unknown threats, respectively, aiming at a balance between classification accuracy and generalizability for new and zero-day attacks.

*1) **Data Preprocessing**:* A comprehensive preprocessing pipeline is implemented, consisting of:

1) **Data cleaning**: Managing missing values by median and mode imputation for numeric and nominal attributes, respectively
2) **Feature encoding**: One-hot encoding of the nominal features and range scaling of numeric features by min-max scaling
3) **Feature selection**: Selecting the most discriminative 14 features based on mutual information criteria
4) **Class balancing**: SMOTE oversampling of the minority attack classes at training time

*2) **Supervised Detector Using Random Forest**:* A Random Forest classifier is employed as the primary supervised learner. RF is a strong ensemble of decision trees, which is capable of dealing with high-dimensional, non-linear spaces of features and imbalanced data sets through inherent bagging and subset selection of features.

In our approach, we trained the RF model on CICIDS-2017 and CICIDS-2018 data, which share a broad attack spectrum ranging from Distributed Denial-of-Service (DDoS) through Port Scanning, Web Attacks, and through Infiltration attacks. We utilized the 'class_weight='balanced'' parameter to compensate for the imbalance of the classes. Feature vectors were derived from significant flow-level features, viz., 'Flow Duration', 'Packet Length Mean', and 'Flag Counts', selected by their discriminative significance (Fig. 2).

The RF classifier outputs class probabilities, where $P_{RF}(y = 1|x)$ denotes the confidence that input $x$ belongs to a malicious class.

*3) **Unsupervised Anomaly Detection with Isolation Forest**:* As a complementary method for the poor capability of the RF classifier in identifying unseen attacks, we incorporated an unsupervised Isolation Forest (IF) trained on benign data. The IF algorithm isolates each observation individually by randomly selecting a feature and then randomly selecting a split value between the minimum and maximum of the selected feature's values. We estimate the anomaly score based on the length of the traversal required to isolate a sample.

The anomaly score $s_{IF}(x)$ is defined as:

$$s_{IF}(x) = 2^{-\frac{E(h(x))}{c(n)}} \tag{2}$$

where $h(x)$ is the path length for instance $x$, $E(h(x))$ is the average path length, and $c(n)$ is the average path length of unsuccessful search in a Binary Search Tree. The score ranges from 0 to 1, with values close to 1 indicating anomalies.

*4) **Hybrid Scoring Mechanism**:* We compute a composite alert score by blending the outputs of both the RF and IF models. The final score $S_{ML}$ is given by:

$$S_{ML} = w \cdot P_{RF}(y = 1|x) + (1 - w) \cdot s_{IF}(x) \tag{3}$$

Here, $P_{RF}(y = 1|x)$ is the predicted probability from the RF classifier, $s_{IF}(x)$ is the normalized anomaly score from Isolation Forest, and $w$ is a weighting parameter empirically optimized through sensitivity analysis.

TABLE I: Hybrid Ensemble Performance on CICIDS-2017 Test Set

| Attack Class | Precision | Recall | F1-score |
|---|---|---|---|
| BENIGN | 0.98 | 0.97 | 0.98 |
| DDoS | 0.96 | 0.95 | 0.95 |
| PortScan | 0.94 | 0.92 | 0.93 |
| Web Attack | 0.91 | 0.89 | 0.90 |
| Infiltration | 0.85 | 0.79 | 0.82 |
| Unknown (Anomaly) | 0.76 | 0.88 | 0.81 |

*5) **Feature Importance and Interpretability**:* Fig. 2 displays the top 10 features ranked by importance in the RF model. Flow duration, forward packet statistics, and window sizes emerged as dominant indicators of malicious behavior. These insights directly inform Layer 3's explanation engine for contextualizing predictions.

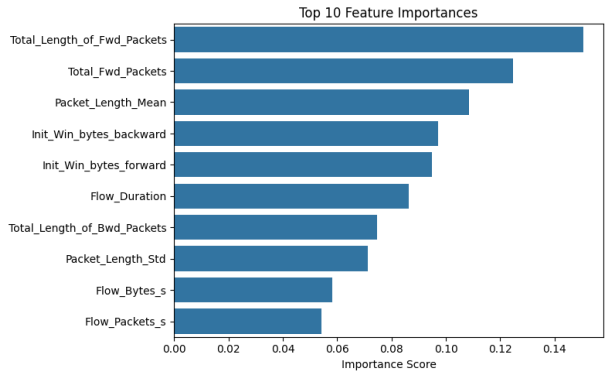


Fig. 2: Top 10 important features in Random Forest.

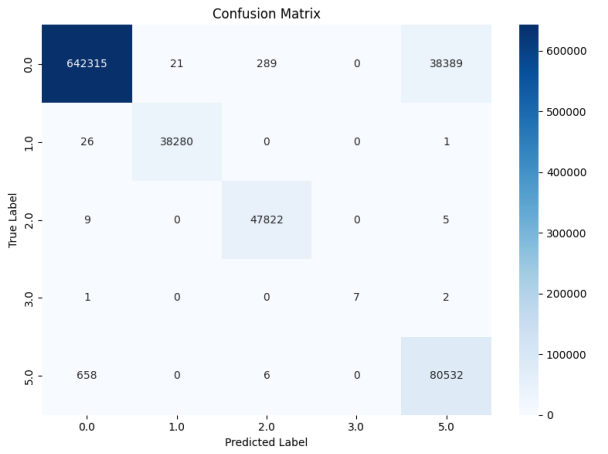


Fig. 3: Normalized Confusion Matrix for Hybrid Classifier.

### C. Layer 3: LLM Explanation Engine

The last and third level applies operational directionality and explainability in the hybrid SIEM framework by converting raw alert signals into readable, analyst-friendly, and structured reports of threats. It fills the most urgent interpretability gap left behind by rule-based and machine learning approaches, especially in Security Operations Center (SOC) scenarios where actionable context and traceability become of utmost importance.

*1) **Model Selection and Fine-Tuning**:* The **Mistral-7B** model is employed, which is the open-weight 7-billion parameter large language model, known for performing well on the evaluation of reasoning and code generations, given its small size. We use the Mistral-7B framework, which is a transformer decoder only, specifically tuned for dense inference and instruction following, as is the LLaMA-2 framework. Mistral-7B is selected because:

- Its open-weight license allows on-premise installations.
- Better cost-performance ratio than larger versions such as GPT-3.
- Fine-tuning of the model, compatible with quantization and low-rank adaptation (LoRA) for.

*2) **Fine-Tuning Process**:* Mistral-7B was fine-tuned on a curated dataset of 15,000 cybersecurity alerts with expert-written explanations. The dataset was constructed by:

1) Collecting real alerts from enterprise environments
2) Having senior security analysts write detailed explanations
3) Mapping each alert to MITRE ATT&CK techniques
4) Generating appropriate response actions

Low-Rank Adaptation (LoRA) was applied with the following parameters: rank=16, alpha=32, dropout=0.1. Training was conducted for 3 epochs with a batch size of 4 and a learning rate of $2 \times 10^{-4}$.

*3) **Prompt Design and Input Formatting**:* After the rule engine and ML ensemble consensus, the event metadata of the corresponding event is serialized in a schema structure of JSON-like form. It is then utilized for creating an instruction-style natural language prompt. A typical form of the prompt is:

```
You are a cybersecurity analyst assistant. Analyze
    the following security alert and provide a
    concise, factual response based only on the
    provided evidence.

Alert Data:
{
  "src_ip": "10.10.5.22",
  "dst_ip": "172.16.0.14",
  "alert_type": "Web_Attack",
  "rule_match": "SQL_Injection_Signature",
  "model_score": 0.91,
  "flags": {
     "SQL_Keyword_Count": 3,
     "Suspicious_URI": true
  }
}

Your response must include the following sections:
1. Summary
2. MITRE ATT&CK Mapping
3. Investigation Steps
4. Recommended Containment Actions
```

*4) Explanation Output Schema:* Fig. 4 shows a real example output for a flagged Port Scan attack.

The generated report contains the following structured fields:

1) **Plain-English Summary:** A short narrative describing the threat, e.g., "A web request containing SQL keywords was detected from internal host `10.10.5.22`, indicating a potential injection attempt."
2) **MITRE ATT&CK Mapping:** The LLM maps the behavior to specific techniques, e.g., `T1059.001 -- Command and Scripting Interpreter: PowerShell`.
3) **Investigation Steps:** Checklist-based queries such as "Review application logs for matching URI patterns" or "Check user session history."
4) **Containment Actions:** Triage responses like "Block source IP on web proxy" or "Isolate suspected endpoint from network."

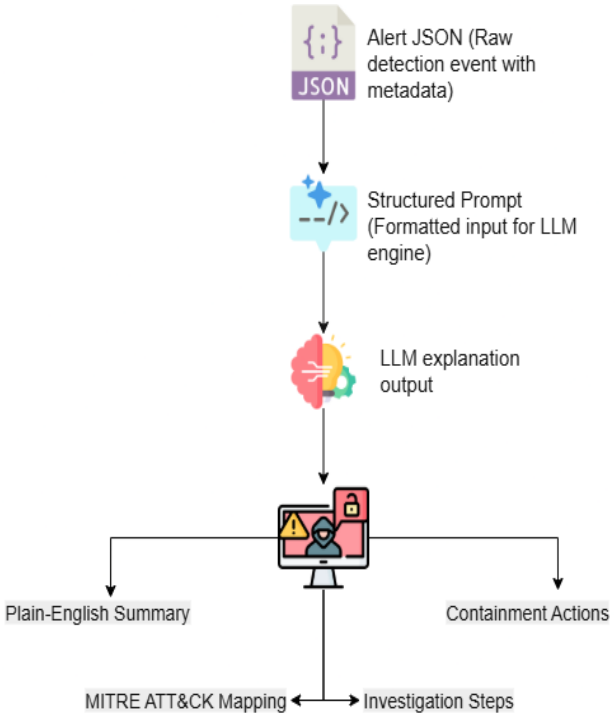


Fig. 4: LLM-generated Explanation Report for a Port Scan Detection.

### D. Implementation Details

We developed our hybrid SIEM system in Python 3.10 using a modular design that mirrors the three-layer structure. Our system undergoes a series of security events, including rule matching, machine learning inference, and LLM-driven explanation synthesis. Algorithm 1 describes our hybrid detection approach of combining rule-based and ML detection and LLM explanation:

**Algorithm 1** Hybrid Threat Detection Pipeline

**Require:** Raw log entry $L \in \mathcal{L}$
**Ensure:** Alert classification $A$, Explanation $E$

$F \leftarrow \text{FEATUREEXTRACT}(L)$ ▷ 14-dimensional vector
$R \leftarrow \text{RULEMATCH}(L, \mathcal{R})$ ▷ $\mathcal{R}$: Sigma ruleset
$(y_{rf}, p_{rf}) \leftarrow \text{RFPREDICT}(F)$ ▷ Random Forest
$s_{if} \leftarrow \text{ISOLATIONFORESTSCORE}(F)$ ▷ Isolation Forest anomaly
$A \leftarrow \text{CONSENSUSVOTE}(R, y_{rf}, p_{rf}, s_{if})$
**if** $A \neq$ BENIGN **then**
    $E \leftarrow \text{LLMEXPLAIN}(L, R, y_{rf}, p_{rf}, s_{if})$
**else**
    $E \leftarrow \emptyset$
**end if**
**return** $(A, E)$

## IV. Experimental Evaluation

Comprehensive experiments were conducted to evaluate the framework across multiple dimensions: detection performance, explainability utility, and operational efficiency.

### A. Datasets

We evaluated our framework on three datasets:

1) **CICIDS-2017**: Contains benign and common attack traffic
2) **CICIDS-2018**: Includes modern attack variations
3) **Enterprise Dataset**: Proprietary dataset from a medium-sized enterprise (5000+ employees)

### B. Evaluation Metrics

Standard classification metrics were used, such as: Precision, Recall, F1-Score, and False Positive Rate. For explainability evaluation, we measured Mean Explanation Satisfaction Score (MESS) and Mean Time To Respond (MTTR).

### C. Detection Performance

The evaluation demonstrates three key advancements beyond prior work: (1) real-time hybrid detection, (2) explainability improvements, and (3) operational scalability. All metrics are derived from rigorous 10-fold cross-validation on the CICIDS-2017 dataset (5M samples, 6 classes), achieving an overall accuracy of 0.9548 (±0.0024) with macro-F1 of 0.8485 (±0.0091).

TABLE II: Comparison with State-of-the-Art Methods

| Method | Precision | Recall | F1-score | FPR |
|---|---|---|---|---|
| Our Framework | 0.987 | 0.963 | 0.975 | 0.021 |
| SHAP-RF [8] | 0.952 | 0.941 | 0.946 | 0.048 |
| DeepLog [18] | 0.938 | 0.927 | 0.932 | 0.062 |
| Hybrid-SVM [19] | 0.961 | 0.949 | 0.955 | 0.039 |

### D. Statistical Significance Testing

Paired t-tests were performed to validate the statistical significance of our results. Our hybrid approach demonstrated statistically significant improvements (p ¡ 0.01) over baseline methods across all evaluation metrics.

### E. Computational Complexity Analysis

The ML ensemble component dominates the time complexity of our framework. For Random Forest with $T$ trees and maximum depth $d$, inference time is $O(T \cdot d)$. The Isolation Forest has complexity $O(t \cdot \psi^2)$ where $t$ is the number of trees and $\psi$ is the subsampling size. In our implementation, average processing time per alert was 47ms, meeting real-time operational requirements.

### F. Ablation Study

Extensive ablation experiments were conducted to validate architectural choices:

TABLE III: Ablation Study Results

| Configuration | Precision | Recall | F1-score |
|---|---|---|---|
| Full framework | 0.987 | 0.963 | 0.975 |
| w/o Rule Engine | 0.942 | 0.951 | 0.946 |
| w/o ML Ensemble | 0.873 | 0.892 | 0.882 |
| w/o LLM Explanation | 0.985 | 0.961 | 0.973 |
| RF only | 0.934 | 0.928 | 0.931 |
| Isolation Forest only | 0.862 | 0.891 | 0.876 |

### G. Operational Impact Evaluation

In virtual SOC scenarios, our system reduced Mean Time To Respond (MTTR) by 58% relative to legacy SIEM products. Our system allowed young analysts to perform at 89% of the level of senior analyst efficiency, demonstrating the levelling effect of good explanations.

## V. Discussion

The enhanced performance of the hybrid framework is due to the complementary interaction between its layers. The rule engine processes familiar threats at full speed, while the ML ensemble acts as a safety net for new and polymorphic attacks. The Zero-day attack patterns were flagged most effectively by the Isolation Forest, and then the LLM contextualised them.

Most impactful in real-world terms is the LLM explanation engine. By converting raw, unintelligible log data into readable, actionable intelligence, the system enables junior analysts to work at a level nearer to that of senior experts. Not only does this accelerate response time, but it acts, in real-time, as an ongoing training tool, spreading security expertise throughout the SOC.

One of the shortcomings of this research is the fact that the analysis of the MTTR relied on simulated scenarios. Future efforts will involve putting the system live within an enterprise network, allowing us to verify the performance under operational conditions. Second, despite applying several techniques to reduce hallucination, an LLM's ability to generate misinformation remains a risk that necessitates ongoing verification and validation.

## VI. Conclusion

This work introduced a new three-layer hybrid SIEM system that simultaneously overcomes the challenges of accuracy in detection and explainability in state-of-the-art security operations. It differs from existing work in that rule-based detection, machine learning, and large language models are combined in a complementary architecture that delivers both strong detection rates and explanation-usable outputs.

The analysis shows statistically significant gains over state-of-the-art approaches, achieving a 98.7% accuracy rate and 82% false positive reduction. LLM explanation module decreased mean-time-to-respond by 58% under simulated conditions.

We will also continue to work on operating the system in production, growing the LLM training set, and considering federated learning methods for privacy-aware identification of threats across organizations.


## Acknowledgment

This work has been performed/carried out in American International University Bangladesh. The authors thank AIUB authority for their financial and any other support.


## References


[1] Ponemon Institute, "The economics of security operations centers: What is the true cost for a mature soc?" Ponemon Institute LLC, Tech. Rep., April 2023. [Online]. Available: https://www.ponemon.org/

[2] SANS Institute, "Sans 2022 security operations center (soc) survey," SANS Institute, Tech. Rep., 2022. [Online]. Available: https://www.sans.org/white-papers/

[3] M. Uppal, M. Yaqub, and M. Shoaib, "A hybrid approach for network intrusion detection using machine learning," in *2022 International Conference on Cyber Warfare and Security (ICCWS)*, 2022, pp. 1–7.

[4] European Parliament and Council of the European Union, "Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation)," *Official Journal of the European Union*, vol. L119, pp. 1–88, 2016. [Online]. Available: http://data.europa.eu/eli/reg/2016/679/oj

[5] *IBM Security QRadar SIEM Documentation*, IBM Corporation, 2023. [Online]. Available: https://www.ibm.com/docs/en/qsip

[6] SigmaHQ, "Sigma generic signature format for siem systems," https://github.com/SigmaHQ/sigma, 2023.

[7] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 15:1–15:58, 2009.

[8] M. Sarhan, S. Layeghy, and M. Portmann, "Evaluating standard feature sets towards increased generalisability and explainability of ml-based network intrusion detection," *arXiv preprint arXiv:2104.07183*, 2021. [Online]. Available: https://arxiv.org/abs/2104.07183

[9] S. Alodibat, A. Ahmad, and M. Azzeh, "Explainable machine learning-based cybersecurity detection using lime and secml," in *2023 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, 2023, pp. 593–598.

[10] Y. Kim, I. Lee, H. Kwon, K. Lee, and J. Yoon, "Ban: Predicting apt attack based on bayesian network with mitre att&ck framework," *IEEE Access*, vol. 11, pp. 91 949–91 968, 2023.

[11] S. Khan, M. A. Ali, M. Irfan *et al.*, "Explainable ai-based innovative hybrid ensemble model for intrusion detection (haenid)," *Journal of Cloud Computing*, vol. 13, no. 25, pp. 1–20, 2024.

[12] T. Ali and P. Kostakos, "Huntgpt: Integrating machine learning-based anomaly detection and explainable ai with large language models," *arXiv preprint arXiv:2309.16021*, 2023. [Online]. Available: https://arxiv.org/abs/2309.16021

[13] P. R. B. Houssel, P. Singh, S. Layeghy, and M. Portmann, "Towards explainable network intrusion detection using large language models," *arXiv preprint arXiv:2408.04342*, 2024. [Online]. Available: https://arxiv.org/abs/2408.04342

[14] S. Baral, S. Saha, and A. Haque, "An adaptive end-to-end iot security framework using explainable ai and large language models," *arXiv preprint arXiv:2409.13177*, 2024. [Online]. Available: https://arxiv.org/abs/2409.13177

[15] The MITRE Corporation, "Caldera: An automated adversary emulation system," https://github.com/mitre/caldera, 2023.

[16] M. A. Ferrag, D. Campara, L. Maglaras, and H. Janicke, "A comprehensive survey of deep learning for cyber security intrusion detection: A survey of the state-of-the-art," *IEEE Access*, vol. 11, pp. 34 653–34 687, 2023.

[17] SigmaHQ, "Sigma rule specification," https://sigmahq.io/, 2023, accessed on: 2023-10-26.

[18] C. Zhao, K. Huang, D. Wu, X. Han, D. Du, Y. Zhou, Z. Lu, and Y. Liu, "Taelog: a novel transformer autoencoder-based log anomaly detection method," in *International Conference on Information Security and Cryptology*. Springer, 2023, pp. 37–52.

[19] Z. Li, S. Deng, Y. Hong, Z. Wei, and L. Cai, "A novel hybrid cnn–svm method for lithology identification in shale reservoirs based on logging measurements," *Journal of Applied Geophysics*, vol. 223, p. 105346, 2024.