

Development of a Lightweight SIEM for SMEs Using Rule-Based Correlation and Isolation Forest for Network Monitoring

Ndubuisi Ifeanyichukwu Eleazar¹; Bilkisu L. Muhammad-Bello²

¹Department of Information Technology, Faculty of Computing, Nile University of Nigeria, Abuja, Nigeria

²Department of Software Engineering, Faculty of Computing, Nile University of Nigeria, Abuja, Nigeria

Publication Date: 2026/10/22

Abstract: Small and Medium Enterprises (SMEs) face escalating cybersecurity threats yet commonly lack the financial and technical resources to deploy and manage enterprise-grade Security Information and Event Management (SIEM) systems. This study addresses that gap by designing, implementing, and evaluating a lightweight centralised SIEM that integrates rule-based correlation with Isolation Forest-based anomaly detection to improve threat detection and monitoring in SME environments. The system was built entirely from open-source components: Graylog served as the centralised log management platform while Python modules implemented the detection logic. Log events were structured as JSON, aggregated into fixed 5-minute time windows, and converted to numerical feature vectors for model training and evaluation. Testing was performed on a simulated SME-like dataset comprising normal operational activity and injected attack scenarios. The rule-based engine achieved perfect precision (1.000) but limited recall (0.333), correctly identifying all triggered alerts as true threats while missing four of six attack episodes. The Isolation Forest model achieved perfect recall (1.000) with an area under the ROC curve (AUC) of 0.937, detecting all six attack scenarios at an average Mean Time to Detect (MTTD) of 2.5 minutes, against an MTTD of 3.5 minutes for rules-only detection. The hybrid approach matched the Isolation Forest performance, confirming anomaly detection as the dominant component under these conditions. The findings demonstrate that open-source, lightweight SIEM solutions can deliver cost-effective and scalable cybersecurity monitoring for SMEs.

Keywords: SIEM; Anomaly Detection; Isolation Forest; Rule-Based Correlation; SME Cybersecurity; Network Monitoring.

How to Cite: Ndubuisi Ifeanyichukwu Eleazar; Bilkisu L. Muhammad-Bello (2026) Development of a Lightweight SIEM for SMEs Using Rule-Based Correlation and Isolation Forest for Network Monitoring. *International Journal of Innovative Science and Research Technology*, 11(10), 1166-1174. <https://doi.org/10.38124/ijisrt/26oct433>

I. INTRODUCTION

In contemporary organisations, cybersecurity threats continue to evolve, rendering traditional log monitoring systems insufficient to detect complex attack patterns [1]. The dynamic technological landscape exposes businesses of all sizes to constantly shifting cyber risks; however, Small and Medium Enterprises (SMEs) are particularly susceptible owing to limited financial and technical resources [2]. Without the budgets or skilled IT personnel available to larger enterprises, SMEs struggle to implement effective protections against sophisticated cyberattacks.

Security Information and Event Management (SIEM) systems provide centralised monitoring and incident detection capabilities. They are used effectively in organisations to collect, evaluate, and visualise security logs [3]. Most SIEM systems, however, are designed for large enterprises, making them prohibitively expensive and operationally complex for SMEs. At the same time, SMEs

typically generate vast quantities of logs from diverse systems, yet have few resources to manage them effectively.

Centralised SIEM consolidates disparate log sources into a single real-time console, facilitating efficient collection from heterogeneous sources for real-time analysis [3]. Empirical studies confirm that centralised log management improves security outcomes and enables faster incident response, reducing Mean Time to Investigate (MTTI) and Mean Time to Respond (MTTR) [4].

This study designs and implements a lightweight centralised SIEM for SMEs that combines rule-based correlation with Isolation Forest-based anomaly detection [5]. The objective is to create a cost-effective solution that improves cybersecurity visibility and threat detection within the constraints of SME IT infrastructures [6, 3].

II. LITERATURE REVIEW

➤ *SIEM Challenges for SMEs*

Conventional SIEM systems were engineered for large organisations capable of running full Security Operations Centres (SOCs), demanding significant processing power, storage capacity, expert personnel, and financial investment. Prior studies confirm that most SMEs lack centralised monitoring, making adoption of complex SIEM technologies challenging [7, 8]. Without central visibility, SMEs cannot detect early indicators of compromise, and scattered logs increase MTTD and MTTR. Centralised log management has been shown to save investigation time and improve monitoring productivity [4].

➤ *Lightweight SIEM Architectures*

Recent research has focused on lighter, more manageable SIEM solutions for small contexts. Lightweight SIEM gathers, normalises, correlates, and alerts on log events with reduced hardware and administration overhead. Open-source tools, simplified rule processing, and automated anomaly detection have been combined to make monitoring feasible in constrained environments [9]. Single-server configurations enabling SMEs to collect logs, index, and visualise dashboards reduce costs significantly, making centralised monitoring achievable without enterprise-class SIEM expenditure [10, 11].

➤ *Rule-Based Correlation*

Rule-based correlation remains the primary detection mechanism in most SIEM systems. Correlation engines link events across time using criteria such as AND/OR/NOT operators, thresholds, and event sequencing, enabling detection of multi-stage attacks that appear harmless in isolation [12]. Chanikarn and Rattadilok [13] demonstrated that well-tuned correlation rules can reduce false positives by 21%. The fundamental limitation is that rule-based detection only catches known, pre-defined threats; novel or zero-day attacks that do not match existing rule signatures pass undetected, which is precisely why anomaly detection is required as a complementary component.

➤ *Isolation Forest Anomaly Detection*

Isolation Forest is an unsupervised machine learning algorithm suited to anomaly detection in log data due to its computational efficiency and performance on high-dimensional data [14]. The algorithm constructs random decision trees that recursively partition data; anomalies, being rare and structurally distinct, are isolated in fewer splits and receive shorter average path lengths, which serve as anomaly scores. Isolation Forest is appropriate for lightweight SIEM systems because it imposes low computational overhead, requires no labelled training data, and scales to large log volumes. Valdovinos et al. [15] demonstrated that incorporating multiple features such as login frequency, IP diversity, and session characteristics significantly improves log-based anomaly detection performance.

➤ *Research Gap*

The critical gap in existing literature is the absence of end-to-end lightweight centralised SIEM architectures that simultaneously integrate centralised log management, operationally practical rule-based correlation, and unsupervised anomaly detection, all validated under SME-realistic deployment constraints [16, 17]. Tariq et al. [17] demonstrated the viability of open-source SIEM tools in budget-constrained organisations, and Chamkar et al. [18] showed that machine learning enhances detection pipelines; however, no prior study presents a complete hybrid detection workflow as a simple, reproducible lightweight implementation targeting SMEs. This paper addresses that gap.

III. METHODOLOGY

➤ *Research Approach*

The study employs the Design Science Research Methodology (DSRM) to design, implement, and evaluate the lightweight SIEM artefact [19]. DSRM is suited to research that produces and rigorously tests technological artefacts addressing identified practical problems. The methodology comprised four sequential phases: needs assessment, system design, artefact implementation, and empirical evaluation.

➤ *Dataset Strategy*

A synthetic log dataset was generated using Python scripts to simulate realistic SME operations. Each log record was structured as a JSON object containing: timestamp, source_ip, destination_ip, user, event_type (login, file_access, web_access, data_transfer, scan), status (success/failed), and bytes_sent. Normal operational behaviour was generated across a pool of internal IP addresses (192.168.1.x range) with five user accounts performing routine authentication and network activity. Attack scenarios were injected at controlled intervals, including brute-force login attempts from external IPs (10.0.0.x) and anomalous port-scan traffic patterns. Feature selection and attack scenario design were validated against the CICIDS2017 and UNSW-NB15 public intrusion detection datasets [20]. The training split comprised approximately 200 normal-behaviour log events used exclusively to establish the Isolation Forest baseline; the test split contained approximately 460 mixed events including six injected attack windows. Figure 1 illustrates the dataset split.

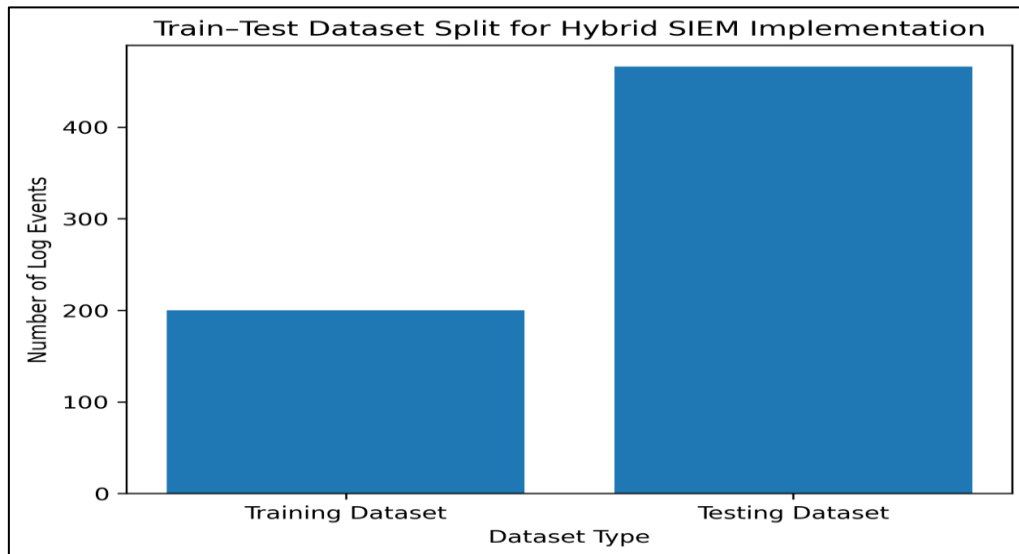


Fig 1 Train-Test Dataset Split for Hybrid SIEM Implementation

➤ System Architecture

The hybrid SIEM was deployed as a combination of Graylog (centralised SIEM engine) and a Python-based hybrid detection engine. The two-layer architecture provides both log processing and actionable security intelligence. The principal components are: (1) a Log Collection Layer ingesting logs over Syslog and GELF TCP protocols; (2) a Log Parsing and Normalisation Engine transforming

heterogeneous logs into a unified JSON schema; (3) a Storage and Indexing Module using Graylog with MongoDB; (4) a Rule-Based Correlation Engine identifying known threat patterns; (5) an Isolation Forest Anomaly Detection Module detecting behavioural deviations; and (6) a Hybrid Alert Fusion Engine combining and deduplicating alerts from both detectors before forwarding to Graylog dashboards. Figure 2 presents the full system architecture.

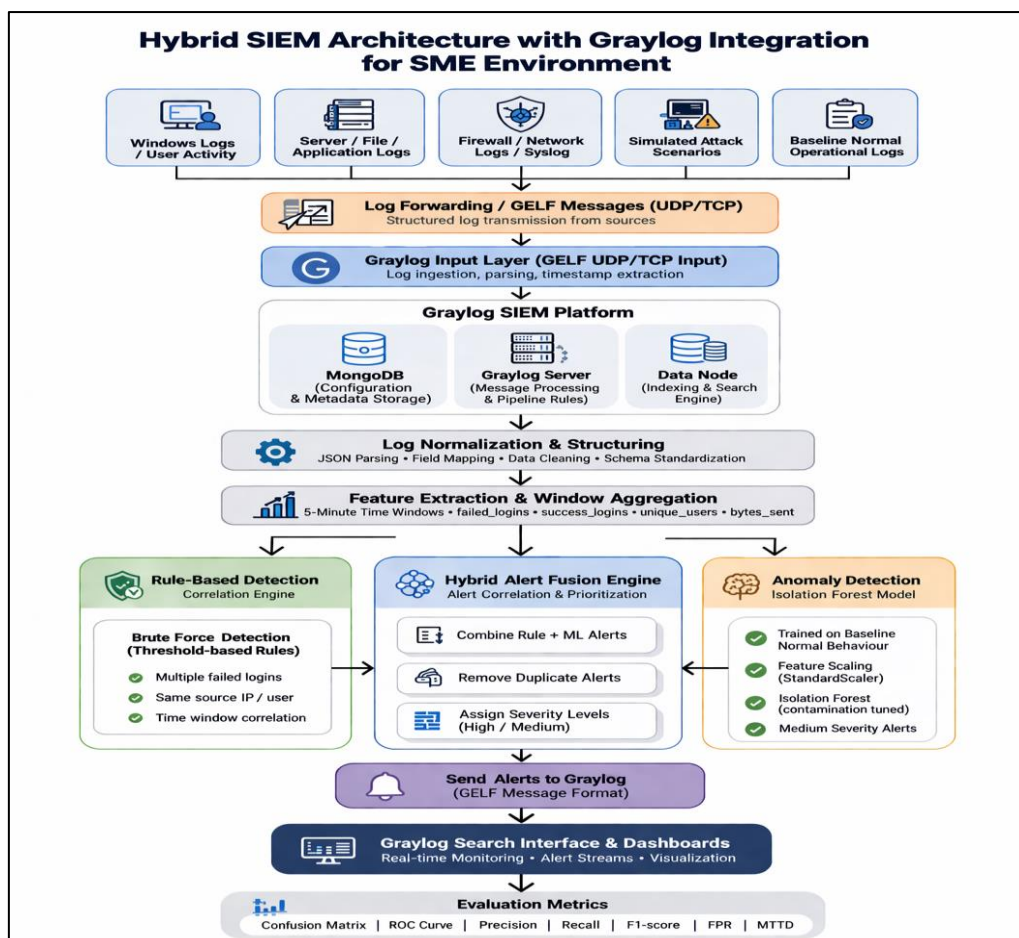


Fig 2 Hybrid SIEM Architecture with Graylog Integration for SME Environment

➤ Implementation

This section describes the implementation in sufficient detail to enable full reproducibility. All source code and

configuration files are structured within a Docker Compose project directory with the following layout:

```
graylog-siem/
├── docker-compose.yml      (Graylog + MongoDB + DataNode)
├── generate_logs.py        (synthetic log generator)
├── hybrid_detector.py      (feature extraction + IF model + rule engine)
└── logs/sample_logs.json   (generated log data)
```

• Step 1 – Graylog Deployment Via Docker Compose

Graylog 7.0 was deployed using a three-container Docker Compose stack: the Graylog server (message processing and pipeline rules), a Graylog Data Node

(indexing and search engine), and MongoDB 7 (configuration and metadata storage). The stack is started with `docker compose up -d`. Figure 3 shows all three containers confirmed healthy via `docker compose ps`.

```
PS C:\Users\user\OneDrive\Desktop\graylog-siem> docker compose ps
NAME                IMAGE                                COMMAND                                SERVICE    CREATED        STATUS
graylog             graylog/graylog:7.0               "/usr/bin/tini -- wa..."          graylog    2 days ago    Up 2 days (healthy)
graylog-datanode    graylog/graylog-datanode:7.0      "tini -- /entrypoint..."          datanode   2 days ago    Up 2 days
graylog-mongodb     mongo:7                           "docker-entrypoint.s..."          mongodb    2 days ago    Up 2 days
PS C:\Users\user\OneDrive\Desktop\graylog-siem> |
```

Fig 3 Running Graylog Services (docker compose ps) — All Containers Healthy

• Step 2 – GELF TCP Input Configuration

After Graylog initialised (accessible at <http://localhost:9000>), a GELF TCP input was created via System → Inputs → Launch new input. The input was bound to 0.0.0.0 on port 12202 with null-delimiter framing and

UTF-8 encoding. This endpoint receives structured JSON alert messages from the Python detection engine. Figure 4 shows the configured input in its running state with throughput metrics confirming 161 total active connections processed.

Fig 4 Graylog GELF TCP Input Configuration (Port 12202, Running State)

• Step 3 – Log Generation

The `generate_logs.py` script produced the synthetic SME log dataset. Normal-behaviour logs were generated at one-minute intervals for internal IPs (192.168.1.1–192.168.1.50) and five user accounts across event types: login, file_access, web_access, data_transfer. Attack windows were injected at predetermined timestamps: brute-force events from external IPs (10.0.0.x) generating five or more failed login events within two-minute rolling windows, and port-scan events targeting a single destination. Logs were

written to `logs/sample_logs.json` in newline-delimited JSON format.

• Step 4 – Feature Extraction and Model Training

The `hybrid_detector.py` script loaded logs with Pandas, parsed timestamps, and aggregated events into 5-minute time windows using `df.resample('5min')`. Six features were computed per window: failed login count, successful login count, unique source IP count, unique user count, total bytes sent, and failed-login ratio. The Isolation Forest model was initialised with `n_estimators=150`, `contamination=0.05`,

random_state=42 and trained exclusively on the 200 normal-behaviour windows. Feature vectors were standardised via StandardScaler before both training and inference. Figure 5

shows the core Python code for window aggregation and rule-based brute-force detection.

```
import pandas as pd
from sklearn.ensemble import IsolationForest

def create_time_windows(df):
    df = df.set_index("timestamp")

    windowed = df.resample("5min").agg({
        "failed_login": "sum",
        "successful_login": "sum",
        "source_ip": "nunique",
        "user": "nunique",
        "bytes_sent": "sum"
    })

    return windowed.fillna(0)

def detect_bruteforce(df):
    alerts = []

    for ip, ip_events in df.groupby("source_ip"):
        rolling = ip_events["failed_login"].rolling("2min").sum()

        if (rolling >= 5).any():
            alerts.append({
                "alert_type": "Brute Force",
                "severity": "High",
                "source_ip": ip
            })
```

Fig 5 Python Code: 5-Minute Window Aggregation and Brute-Force Rule Detection

- *Step 5 – Alert Generation and Graylog Integration*

The rule engine raised a "Brute Force" alert (severity: High) when a source IP produced five or more failed logins within a 2-minute rolling window. The Isolation Forest component raised an "Anomaly" alert (severity: Medium) for

each window scored anomalous by the model. Both alert types were serialised as GELF JSON payloads and forwarded to Graylog on TCP port 12202. Figure 6 confirms successful ingestion in the Graylog search interface; Figure 7 shows the dashboard visualising the alert-type distribution across the test run.

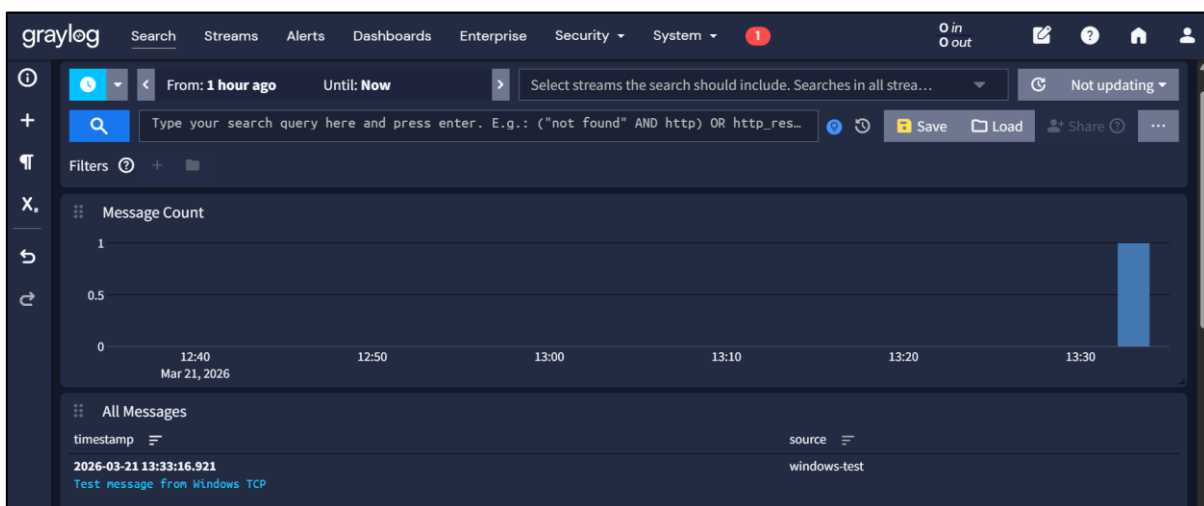


Fig 6 Graylog Search Interface Confirming Ingestion of Hybrid SIEM Alert Messages

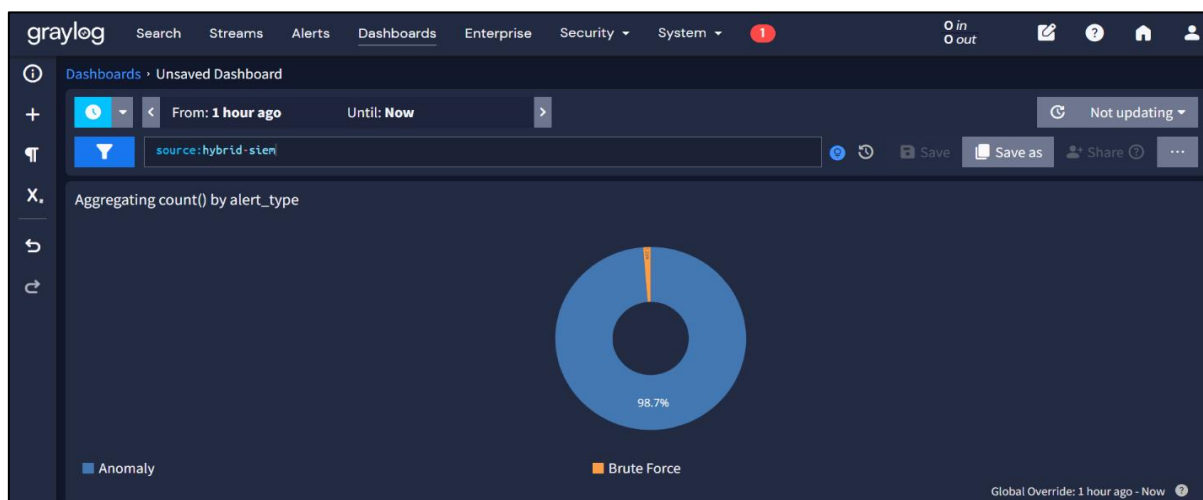


Fig 7 Graylog Dashboard Showing Alert Distribution: Anomaly (98.7%) vs. Brute Force (1.3%)

➤ Evaluation Metrics

Standard classification metrics were used: Precision = $TP / (TP + FP)$; Recall = $TP / (TP + FN)$; F1-score = $2 \times (Precision \times Recall) / (Precision + Recall)$; and False Positive Rate (FPR) = $FP / (FP + TN)$. Operational responsiveness was quantified through Mean Time to Detect (MTTD), defined as the mean time from the start of an injected attack window to the first triggered alert. ROC analysis was performed by varying the Isolation Forest anomaly score threshold to assess discriminatory capacity across the full operating range.

IV. RESULTS AND DISCUSSION

➤ Experimental Setup

All experiments were conducted on the SME-like dataset described in Section III-B. The training split (200 normal-behaviour log events) was used exclusively to fit the Isolation Forest model. The test split (approximately 460 mixed events) contained both benign activity and six injected attack windows spanning brute-force login attempts and anomalous traffic bursts. The Isolation Forest model was never exposed to attack patterns during training, ensuring that its detections during evaluation reflected genuine generalisation from the normal-behaviour baseline.

➤ Confusion Matrix Analysis

Table 1 presents the confusion matrix results for all three detection configurations. The rule-based engine achieved zero false positives ($FP = 0$, $TN = 151$), confirming high specificity—every alert it raised corresponded to a genuine attack. However, it missed four of six attack windows ($FN = 4$, $TP = 2$), because only brute-force login scenarios matched the defined rule signature; the remaining four attack windows involved behavioural anomalies not captured by any predefined rule, resulting in a recall of 0.333. This outcome is statistically significant: the rule-based engine detected only 33.3% of the injected threats despite perfect alert precision. The Isolation Forest model correctly identified all six attack windows ($TP = 6$, $FN = 0$), achieving perfect recall (1.000), while generating twelve false positives ($FP = 12$, $TN = 139$). These false positives correspond to normal time windows whose aggregated feature vectors fell outside the learned baseline distribution, producing an FPR of 0.079—meaning approximately 7.9% of normal activity was incorrectly flagged. The hybrid model produced results identical to the Isolation Forest model, indicating that the anomaly detector already captured every attack window identified by the rule engine, with no complementary detections from the rule-based component under this dataset.

Table 1 Confusion Matrix Summary

Method	TN	FP	FN	TP
Rules-Only	151	0	4	2
Isolation Forest Only	139	12	0	6
Hybrid (Rules + IF)	139	12	0	6

Figure 8 provides a visual representation of the hybrid model's confusion matrix, confirming $TP = 6$, $FN = 0$, $FP = 12$, and $TN = 139$. The absence of false negatives (dark top-

right cell = 0) is the most significant finding: the system did not miss a single injected attack.

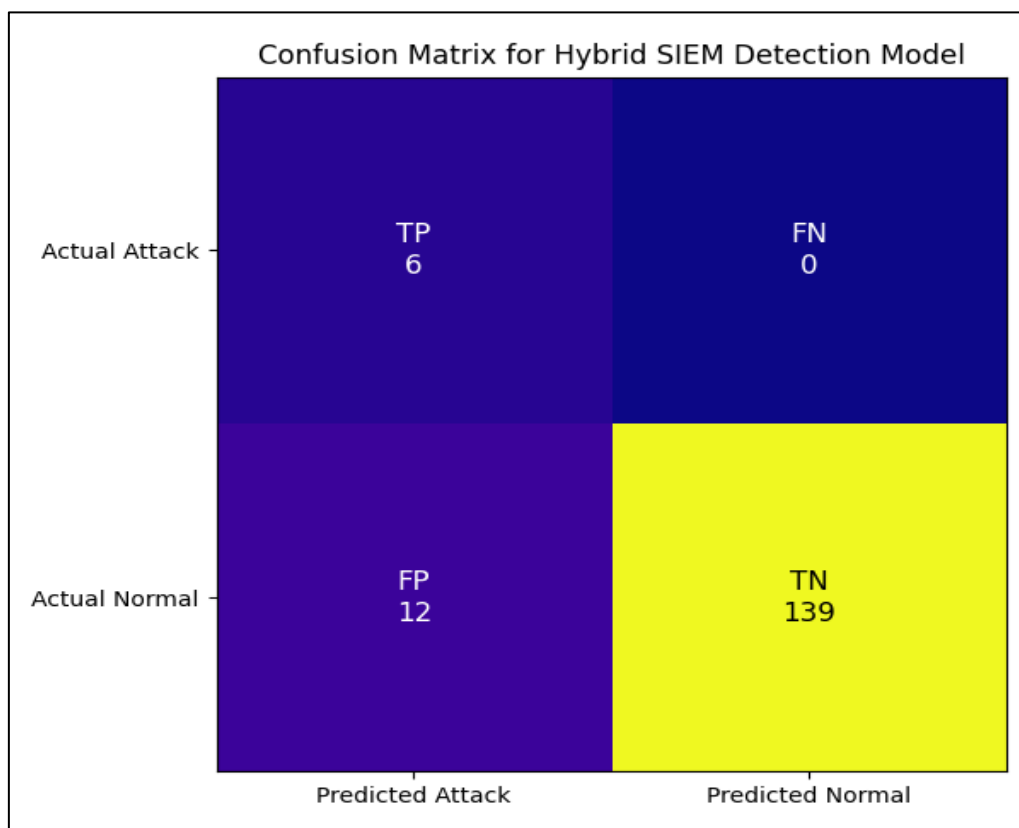


Fig 8 Confusion Matrix for the Hybrid SIEM Detection Model (TP=6, FN=0, FP=12, TN=139)

➤ Detection Performance Metrics

Table 2 summarises the quantitative detection metrics for all three approaches. The rule-based engine achieved precision = 1.000, meaning every alert it raised was a true positive, but its recall of 0.333 reflects that it detected only two of six attack episodes. Its F1-score of 0.500 and FPR of 0.000 confirm a highly conservative but incomplete detection posture—acceptable for detecting known, signature-matched threats but insufficient as a standalone solution in dynamic SME environments.

The Isolation Forest model achieved the inverse profile: recall = 1.000 and precision = 0.333, with F1-score = 0.500 and FPR = 0.079. The perfect recall is of substantial practical significance—in an operational SME context, missing zero attacks means no injected threat went undetected. The cost is a false positive rate of 7.9%, representing manageable analyst workload given the absence of a dedicated SOC in most SMEs. The hybrid model matched the Isolation Forest on all metrics, with precision = 0.333, recall = 1.000, F1-score = 0.500, and FPR = 0.079.

Table 2 Detection Performance Metrics

Method	Precision	Recall	F1-Score	FPR
Rules-Only	1.000	0.333	0.500	0.000
Isolation Forest Only	0.333	1.000	0.500	0.079
Hybrid (Rules + IF)	0.333	1.000	0.500	0.079

➤ Mean Time to Detect (MTTD)

Table 3 shows the MTTD results. The Isolation Forest and hybrid models both detected all six attack incidents with a mean MTTD of 2.5 minutes, constrained by the 5-minute window aggregation design (attacks identified mid-window yield an expected detection latency of approximately half the window duration). The rule-based model detected only two of six incidents at an MTTD of 3.5 minutes, reflecting both

its limited coverage and a slightly higher detection latency because rule matching requires sufficient events to accumulate within the rolling window before threshold triggers. The 1.0-minute reduction in MTTD from rules-only (3.5 min) to anomaly/hybrid detection (2.5 min) represents a 28.6% improvement in detection speed, which can be operationally significant for fast-propagating attacks such as brute-force credential stuffing.

Table 3 Mean Time to Detect (MTTD)

Method	Detected	Total	MTTD (min)
Rules-Only	2	6	3.5
Isolation Forest Only	6	6	2.5
Hybrid (Rules + IF)	6	6	2.5

➤ ROC Curve Analysis

The Receiver Operating Characteristic (ROC) curve was computed for the Isolation Forest model across a range of anomaly score thresholds. The resulting Area Under the Curve (AUC) was 0.937, indicating strong discriminatory capacity between normal and anomalous behaviour. An AUC of 0.937 is substantially above the random-classifier baseline of 0.5 and approaches the near-perfect range ($AUC \geq 0.95$), confirming that the model reliably ranks attack windows

above normal windows across virtually all threshold settings. Figure 9 illustrates the ROC curve; the sharp rise toward the top-left corner confirms that high true positive rates are achievable with low false positive rates when the threshold is appropriately selected. This finding is significant for SME deployments, where the absence of dedicated security analysts makes it desirable to operate at a detection threshold that maximises recall while keeping false positive volume manageable.

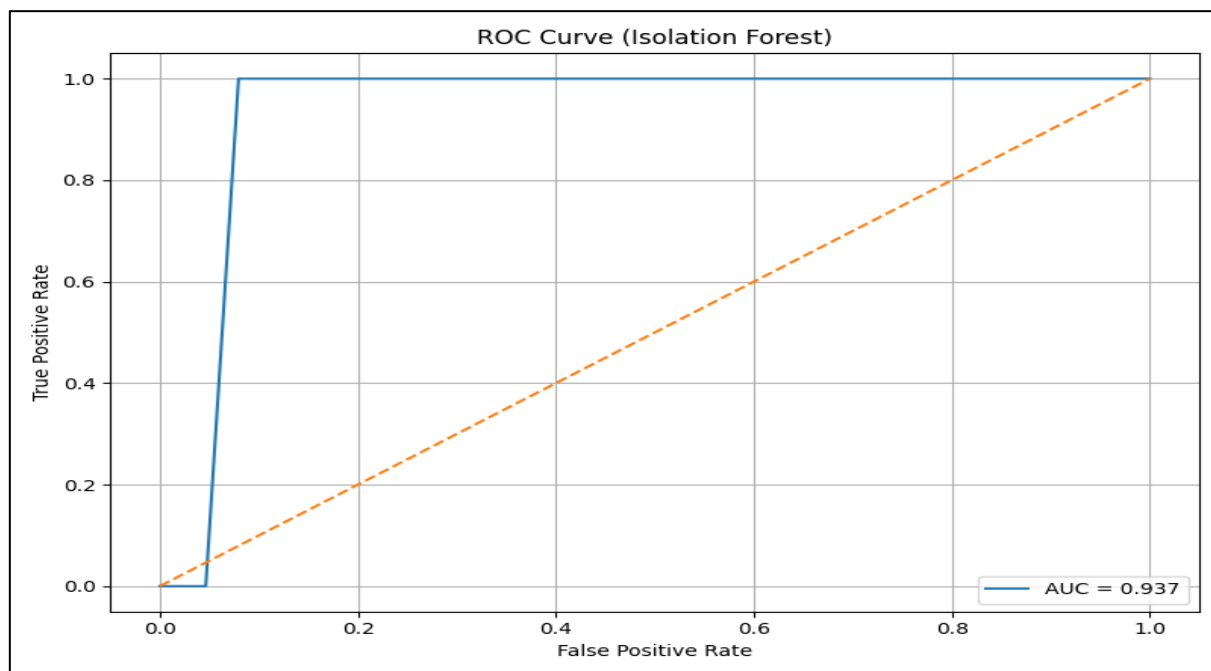


Fig 9 ROC Curve for Isolation Forest Model (AUC = 0.937)

➤ Discussion

The experimental results demonstrate that the proposed hybrid SIEM system successfully detects security threats in an SME-like environment. The most important finding is that the Isolation Forest model achieved perfect recall (1.000) and an AUC of 0.937, detecting all six injected attack scenarios at a mean MTTD of 2.5 minutes—metrics that establish the anomaly detection component as the primary driver of detection effectiveness in this system.

The rule-based engine's perfect precision (1.000) is equally significant: in production, this translates to zero false alarms from the rule component, which reduces alert fatigue for resource-constrained SME staff. The engine's low recall (0.333) is expected and by design—rules detect only known, explicitly coded patterns. In this experiment, only brute-force login behaviour matched the single correlation rule; the remaining attack windows were behavioural anomalies invisible to rule-based logic. This confirms the well-established principle that rule-based detection requires anomaly detection as a complement, not a substitute.

The identical performance of the hybrid and Isolation Forest models under this dataset reflects the controlled and relatively simple nature of the attack scenarios: the anomaly detector captured every attack that the rule engine could also detect, leaving no room for complementary improvement. In

real-world environments, where attack behaviours are more diverse, stealthy, and continuously evolving, rule-based systems play a critical role in detecting known threats with zero false positives, while anomaly detection provides coverage against unknown and zero-day attacks. The hybrid architecture therefore remains the recommended design, as it offers broader coverage across a wider threat landscape than either component alone.

From an implementation perspective, the results confirm that a lightweight Python–Graylog stack can provide effective security monitoring for SMEs without commercial SIEM licensing costs. The entire system runs on commodity hardware within a Docker Compose environment, making it deployable and maintainable by SMEs with limited IT staff. The open-source stack (Graylog, Python, Scikit-learn, Pandas, Docker) reduces total cost of ownership to near zero, addressing the primary barrier to SIEM adoption in SMEs identified in the literature.

The 7.9% FPR of the Isolation Forest model represents the main operational trade-off. In a dataset of 460 test events aggregated into approximately 151 normal windows, 12 normal windows were flagged as anomalous. This volume is manageable in an SME context, but underscores the importance of careful contamination parameter tuning and the potential value of incorporating additional contextual features

(e.g., day-of-week patterns, user role, device type) to reduce false positives in production deployments.

V. CONCLUSION

This paper presented the design, implementation, and empirical evaluation of a lightweight hybrid SIEM system for SME environments. The system combines rule-based correlation and Isolation Forest-based anomaly detection within a fully open-source, Dockerised stack (Graylog, Python, Scikit-learn, MongoDB). Key quantitative findings are: (1) the rule-based engine achieved perfect precision (1.000) with limited recall (0.333), confirming high accuracy for known, signature-matched threats; (2) the Isolation Forest model achieved perfect recall (1.000) and an AUC of 0.937, detecting all six injected attack scenarios at an MTTD of 2.5 minutes; (3) the hybrid model matched Isolation Forest performance in this controlled setting, demonstrating that anomaly detection is the dominant contributor to detection coverage; and (4) the entire system runs on commodity hardware with no commercial licensing costs, making it practically viable for SMEs.

The study confirms that lightweight open-source SIEM solutions can deliver meaningful cybersecurity monitoring to SMEs that would otherwise remain unprotected. Future work should evaluate the system on real organisational log data, incorporate more sophisticated attack scenarios (lateral movement, privilege escalation, low-and-slow attacks), explore deep learning and online learning models for improved accuracy, and integrate automated incident response workflows to further reduce MTTR.

ACKNOWLEDGEMENT

The authors express sincere gratitude to Assoc. Prof. Bilquis L. Muhammad-Bello for her guidance, constructive feedback, and continuous support throughout this research. The authors also acknowledge the lecturers and staff of the Department of Information Technology, Nile University of Nigeria, for their contributions to its academic development.

REFERENCES

- [1]. C. Henderson, "X-Force Threat Intelligence Index 2024 reveals stolen credentials as top risk, with AI attacks on the horizon," IBM, 2024.
- [2]. D. Eliot, J. A. Marron, and S. Thorn, "Small business cybersecurity: Non-employer firms," NIST IR 7621 Rev. 2, 2025.
- [3]. G. González-Granadillo, S. González-Zarzosa, R. Díaz, J. Araujo, F. Castillo, and E. de la Hoz, "Security information and event management (SIEM): Analysis, trends and usage in critical infrastructures," *Sensors*, vol. 21, no. 14, p. 4759, 2021.
- [4]. Graylog, "Reducing mean time to investigate (MTTI) and respond (MTTR) with centralised logging," Graylog White Paper, 2025.
- [5]. F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining (ICDM)*, pp. 413–422, 2008.
- [6]. ENISA, "Cybersecurity for SMEs: Challenges and recommendations," EU Agency for Cybersecurity, 2021.
- [7]. M. Hassani, B. Parsa, and E. Nazari, "Systematic review of cybersecurity adoption in SMEs," *Int. J. Cybersecurity*, vol. 18, no. 3, pp. 54–67, 2021.
- [8]. M. Nunes, J. Silva, and P. Costa, "Cybersecurity readiness in SMEs," *Int. J. Cybersecurity*, vol. 21, no. 3, pp. 178–191, 2022.
- [9]. [M. H. Hossain et al., "Lightweight SIEM for resource-constrained organisations," *IEEE Access*, vol. 11, 2023.
- [10]. A. Alshamrani, A. Alsaedi, and K. Alharthi, "Comparing open-source SIEM tools in a controlled testbed," *J. Cybersecurity Solutions*, vol. 35, no. 2, pp. 115–123, 2020.
- [11]. S. Wang and T. Jones, "Open-source SIEM performance using controlled deployments," *Security Technology and Systems*, vol. 27, no. 4, pp. 167–182, 2022.
- [12]. L. Caviglione et al., "Evaluated SIEM correlation rule performance," *IEEE Trans. Inform. Forensics Security*, vol. 16, 2021.
- [13]. C. Chanikarn and P. Rattadilok, "Rule optimization algorithm for SIEM alerts," in *Proc. ARES Conf.*, 2020.
- [14]. M. Ahmed and M. S. Kim, "Hybrid statistical and log-based anomaly detection," *Computers & Security*, vol. 95, 2020.
- [15]. D. Valdovinos, H. Ruiz, and R. García, "Log-based anomaly detection models: A systematic review," *J. Cybersecurity Research*, vol. 19, no. 6, pp. 245–263, 2021.
- [16]. D. Koutroumpouchos, N. Papageorgiou, and G. Xylas, "Real-world SIEM deployments and operational challenges," *J. Information Security*, vol. 33, no. 2, pp. 80–98, 2020.
- [17]. U. Tariq, I. Ahmed, A. K. Bashir, and K. Shaukat, "A critical cybersecurity analysis and future research directions for the IoT: A comprehensive review," *Sensors*, vol. 23, no. 8, p. 4117, 2023.
- [18]. S. A. Chamkar, M. Zaydi, Y. Maleh, and N. Gherabi, "Improving threat detection in Wazuh using machine learning techniques," *J. Cybersecurity and Privacy*, vol. 5, no. 2, p. 34, 2025.
- [19]. K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *J. Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007.
- [20]. M. Landauer, F. Skopik, M. Frank, W. Hotwagner, M. Wurzenberger, and A. Rauber, "AIT Log Data Set V2.0," Zenodo, 2022. doi:10.5281/zenodo.5789064