WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

AI-Powered Security Operations Center Analyst Using Machine Learning: A Proposed Framework for Threat Detection and Incident Analysis

Kripa Sonar[1], Shruti Deshmukh[1], Avian Raj[1], Shrutik Aher[1]

[1]Department of Artificial Intelligence and Machine Learning ISBM College of Engineering, Nande, Pune, Maharashtra, India
Author: sd0832491@gmail.com

ABSTRACT :

Security Operations Centers (SOCs) process large volumes of events generated by firewalls, servers, authentication systems, and network devices. Manual review of these events can be time-consuming and may delay the investigation of suspicious activity. This paper presents a proposed AI-powered SOC analyst framework that combines supervised machine learning, anomaly detection, threat-intelligence enrichment, and generative-AI-assisted incident reporting in a web-based platform. The planned pipeline uses Logstash to parse incoming logs and Elasticsearch to index them for search and analysis. A Python FastAPI backend coordinates processing, while Random Forest or XGBoost is proposed for classifying known attack patterns. A TensorFlow-based autoencoder is proposed to flag deviations from learned normal behaviour. Suspicious indicators may be enriched using services such as AbuseIPDB and Virus Total. Generative AI is intended to produce analyst-readable incident summaries and suggested remediation steps, while a React.js dashboard presents alerts and incident history. MongoDB is specified for storing incident records and audit information. The framework defines an evaluation plan based on precision, recall, F1-score, false-positive rate, detection latency, and response time. As the project report describes a system under development, this paper does not claim measured performance or validated experimental results. Instead, it documents the proposed architecture, workflow, evaluation criteria, and implementation limitations.

Keywords: Security Operations Center; cybersecurity; machine learning; intrusion detection; anomaly detection; generative AI; threat intelligence; FastAPI; Elasticsearch.

1 Introduction

Organizations generate security events through authentication systems, servers, firewalls, web services, and network devices. These records can provide evidence of suspicious activity, but their volume and variety make manual investigation difficult. A Security Operations Center (SOC) monitors such events, investigates alerts, and coordinates incident handling. Machine learning can assist analysts by classifying known patterns and highlighting activity that differs from expected behaviour. This paper describes a proposed AI-powered SOC analyst intended to bring log processing, machine-learning-based threat classification, anomaly detection, threat-intelligence context, and AI-assisted reporting into one platform. The system is designed to support—not replace—security analysts. Predictions and generated recommendations require review before high-impact response actions are taken.

The main contributions of this paper are: (i) a modular framework for processing security logs and presenting potential threats; (ii) a two-stage detection approach combining supervised classification with anomaly detection; and (iii) an evaluation plan that identifies metrics needed to assess detection quality and operational usefulness.

The current project documentation specifies these components but does not provide completed experimental measurements. Therefore, the framework is presented as a proposal whose effectiveness must be established through implementation and testing.

2 Related Work

The project synopsis identifies several research directions relevant to AI-assisted security monitoring: explainable AI (XAI), machine-learning and deep-learning intrusion detection, hybrid detection systems, and generative-AI-assisted threat classification. These directions provide motivation for making alerts both detectable and understandable. Prasad et al. discuss explainability and anomaly detection as ways to make cybersecurity decisions easier to interpret (Prasad et al., 2026). Such methods are relevant when an analyst needs to understand why an event was flagged rather than receiving only a label. Mondragon et al. describe an AI-enabled Zero Trust framework for cyber-vulnerability detection (Mondragon et al., 2025). This work is relevant to the broader objective of using AI to support security analysis.

Nugraha et al. focus on an explainable intrusion-detection framework (Nugraha et al., 2025). Mubeen and Gudla describe a hybrid deep-learning intrusion-detection approach (Mubeen and Gudla, 2026). Ganesh et al. discuss a two-stage generative-AI fusion approach for cloud-security intrusion detection (Ganesh et al., 2026). As summarized in the project report, these studies motivate combining detection with interpretation and contextual reporting. The cited works should be checked against their publishers' records before final journal submission to confirm complete bibliographic metadata.

3 Problem Definition and Objectives

3.1 Problem Definition

Security logs often arrive in different formats and contain a mixture of routine activity and potentially malicious events. Manual analysis can delay identification, and rule-based monitoring alone may not identify every unfamiliar pattern. Analysts also need context to prioritize alerts and communicate the meaning of an incident. The proposed system addresses these needs by structuring log data, classifying known attack patterns, flagging unusual events, enriching selected alerts, and presenting the resulting information in a centralized dashboard.

3.2 Objectives

The objectives are to:

1. collect and normalize security logs from supported sources;
2. classify known attack patterns using supervised machine learning;
3. identify unusual records using an anomaly-detection model;
4. assign alert severity levels of Low, Medium, High, or Critical;
5. enrich suspicious indicators using available threat-intelligence services;
6. generate readable incident summaries and suggested remediation steps;
7. store incident history and display findings through a web dashboard; and
8. evaluate the system using defined detection and operational metrics.

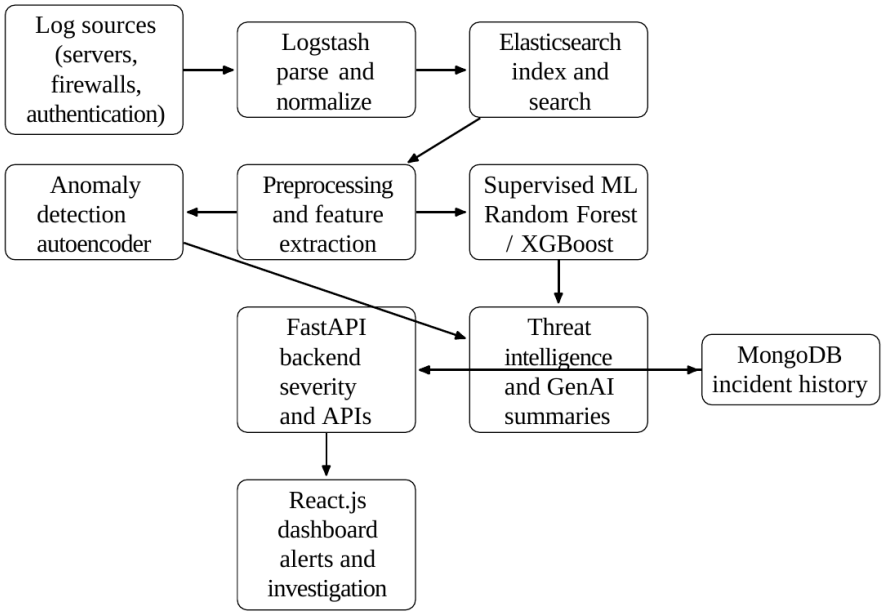


Figure 1: Proposed architecture of the AI-powered SOC analyst. The diagram represents the planned component flow, not a claim that every integration has been completed.

4 Proposed System Architecture

The framework separates ingestion, analysis, enrichment, persistence, and presentation into modules. This modular design is intended to make individual components easier to test and replace.

4.1 Log Collection and Processing

The input layer is intended to accept logs from sources such as Linux authentication services, Windows event logs, firewalls, and web servers. Logstash is specified to parse and normalize incoming records, while Elasticsearch is used for indexing and efficient searching. The actual set of connected log sources will depend on the implemented ingestion configuration.

4.2 Preprocessing and Feature Extraction

Raw records must be converted into consistent structured data before model inference. Candidate features described in the project report include source IP address, destination port, protocol, failed-login counts, and traffic volume. Preprocessing may include validation, duplicate removal, missing-value handling, normalization, and selection of relevant fields. Feature definitions must be aligned with the dataset and model training pipeline.

4.3 Supervised Attack Classification

Random Forest and XGBoost are proposed for classifying known attack patterns, including distributed denial-of-service (DDoS) activity, port scans, SSH brute-force attempts, and web-injection attempts. A supervised model learns associations between labeled examples and their attack classes. For a new event, the model produces a predicted class and, where supported by the implementation, a confidence or probability estimate. The model's usefulness depends on representative labeled data and evaluation on data not used during training.

4.4 Anomaly Detection

A TensorFlow-based autoencoder is proposed to identify records that differ from learned normal patterns. An autoencoder learns to reconstruct its input. The reconstruction error can be calculated as

$$e(x) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2, \quad (1)$$

where x is the input feature vector, $\hat{x}$ is the reconstructed vector, and n is the number of features. Those reconstruction error exceeds a threshold may be flagged for review. The threshold must be selected and validated using suitable data; an anomaly is not, by itself, proof of an attack.

4.5 Threat Intelligence and Generative AI

The proposed framework may query threat-intelligence services such as AbuseIPDB and VirusTotal to obtain additional context for suspicious IP addresses, domains, or file hashes. Availability, access permissions, rate limits, and API policies can affect enrichment. Generative AI is intended to turn structured alert information into readable summaries and possible remediation recommendations. Generated text should be grounded in the available evidence, clearly identified as AI-generated, and reviewed by an analyst before consequential action.

4.6 Backend, Storage, and Dashboard

FastAPI is specified as the backend layer that coordinates processing and exposes endpoints to the frontend. MongoDB is intended to store incident records, analyst notes, and audit history. The React.js dashboard is designed to show alerts, severity, incident details, and investigation status. Authentication, role-based access, input validation, protected API communication, and secure handling of service credentials are important implementation requirements.

5 Methodology

The planned processing workflow consists of the following stages:

1. **Ingestion:** receive supported log records and preserve the original event where appropriate.
2. **Validation and preprocessing:** validate fields, normalize formats, handle missing values, and extract model features.
3. **Detection:** apply the supervised classifier to identify known attack categories and the autoencoder to flag unusual patterns.
4. **Alert assessment:** combine relevant detection outputs and contextual indicators to assign a severity category. The exact scoring logic and thresholds must be defined during implementation and documented for reproducibility.
5. **Enrichment:** retrieve external threat context for eligible indicators when the service is available.
6. **Incident explanation:** prepare a concise incident summary and suggested next steps using the available event evidence.
7. **Persistence and display:** store incident details and expose them through the dashboard for analyst review.
8. **Feedback and audit:** record analyst decisions and relevant actions to support traceability and later evaluation.

Table 1: Planned evaluation metrics

Metric	Purpose
Accuracy	Proportion of predictions that are correct across all evaluated examples.
Precision	Proportion of predicted threat events that are actual threats.
Recall	Proportion of actual threat events that are correctly detected.
F1-score	Harmonic mean of precision and recall, useful when both missed threats and false alerts matter.
False-positive rate (FPR)	Proportion of normal events incorrectly classified as threats.
Mean time to detect (MTTD)	Average time between the relevant event and its detection, using a defined measurement procedure.
Mean time to re-spond (MTTR)	Average time required to respond to an incident, where response timestamps are available and consistently defined.
Inference latency	Time taken by the model to produce a prediction for an input event.

6 Evaluation Plan

The project report identifies standard classification and operational metrics for evaluating the proposed system. These metrics should be calculated on a clearly described test set and reported only after testing is completed. For binary classification, precision and recall can be defined as

$$Precision = \frac{TP}{TP+FP}, \quad Recall = \frac{TP}{TP+FN}, \tag{2}$$

where TP, FP, and FN denote true positives, false positives, and false negatives. The F1-score is

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \tag{3}$$

The evaluation should document the dataset source, class distribution, preprocessing, train/validation/test split, model configuration, hardware, and threshold-selection procedure. Data leakage must be avoided by ensuring that closely related records do not unintentionally appear across training and test partitions. If the system is evaluated in an operational setting, the measurement window and definitions for detection and response times should also be stated.

7 Security, Limitations, and Risk Considerations

The framework has several limitations that require attention. First, model quality depends on the diversity, freshness, and labeling quality of the security data. New attack patterns may not resemble training examples, and anomaly detection can flag legitimate but unusual activity. Second, external threat-intelligence and generative-AI services can be affected by network availability, API limits, service changes, or incomplete context. Third, integrating log pipelines, models, APIs, storage, and a dashboard introduces interface and reliability risks.

Security logs may contain sensitive identifiers and authentication details. The implementation should minimize unnecessary data collection, restrict access, protect credentials, and apply suitable retention controls. False-positive and false-negative alerts must be expected and measured. AI-generated summaries should not be treated as authoritative evidence, and suggested remediation should be checked by a qualified analyst. Any future automated response capability should use least privilege, explicit approval for high-impact actions, and auditable records.

8 Conclusion and Future Work

This paper has presented a proposed AI-powered SOC analyst framework that combines log processing, supervised attack classification, anomaly detection, threat-intelligence enrichment, generative-AI-assisted reporting, and a web dashboard. The architecture is intended to help analysts organize security events, identify suspicious behaviour, and review incident context in one interface. The use of both supervised classification and anomaly detection may support complementary views of known and unusual activity, but the actual benefit must be established experimentally.

The project documentation does not provide completed benchmark results; therefore, no detection accuracy or performance improvement is claimed here. Future work should complete and validate the ingestion and analysis pipeline, document the selected datasets and model configurations, compare candidate classifiers, calibrate anomaly thresholds, evaluate the metrics in Table 1, and test the system under representative workloads. Additional work may include improved explainability, more log-source integrations, robust incident correlation, and carefully governed response capabilities.

Acknowledgements

The authors may acknowledge their project guide, department, institution, or other contributors here. Please confirm the correct names and wording before submission.

REFERENCES :

1. P. W. C. Prasad, M. S. Sayeed, D.-M. Nguyen, D. P. Hutabarat, and G. M. Mohiuddin, “Explainable AI: Enhancing Decision-Making in the Detection of Cyber Threats,” *Frontiers in Computer Science*, vol. 8, Article 1762332, 2026. doi: https://doi.org/10.3389/fcomp.2026.1762332.

2. J. C. Mondragon, P. Branco, G.-V. Jourdan, A. E. Gutiérrez-Rodríguez, and R. R. Biswal, "A Secure Artificial Intelligence Enabled Zero Trust Framework for Cyber Vulnerability Detection in Cybersecurity Systems," *Scientific Reports*, 2025. *Note: complete volume, article number, and DOI should be verified before submission.*
3. B. Nugraha, A. V. Jnanashree, and T. Bauschert, "A Versatile XAI-Based Framework for Efficient and Explainable Intrusion Detection Systems," *Annals of Telecommunications*, vol. 80, pp. 1095–1120, 2025. doi: https://doi.org/10.1007/s12243-025-01118-9.
4. S. Mubeen and B. Gudla, "Deep Shield IDS: An AI-Powered Intrusion Detection System Leveraging Hybrid IDS Net for Robust Network Security," *Scientific Reports*, 2026. *Note: complete volume, article number, and DOI should be verified before submission.*
5. V. Ganesh, N. Khade, H. Taiwade, et al., "A Two-Stage Generative-AI Fusion Intrusion Detection System for Calibrated and Reliable Cloud Security," *Scientific Reports*, 2026. *Note: complete author list, volume, article number, and DOI should be verified before submission.*