IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 1, January 2026

DOI: 10.17148/IJARCCE.2026.151120

AI DRIVEN SIEM

Prajwal B N[1], Prof. Vidya S[2]

PG Student, Dept. of MCA, Bangalore Institute of Technology, Bengaluru-560004, Karnataka, India[1]

Assistant Professor, Dept. of MCA, Bangalore Institute of Technology, Bengaluru-560004, Karnataka, India[2]

Abstract: In recent years, the rapid growth of interconnected digital systems has significantly increased the number and complexity of cyber attacks. Security Information and Event Management (SIEM) systems play an important role in monitoring and analyzing security events in modern organizations. However, traditional SIEM platforms rely mainly on static rule-based detection and manual analysis, which limits their ability to detect unknown threats and respond efficiently in real time. This paper proposes an AI-driven SIEM system for real-time cyber threat detection and security monitoring. The proposed system integrates network traffic analysis, system performance monitoring, and deep learning–based intrusion detection to provide intelligent and automated security analysis. A Convolutional Neural Network is used to classify network behavior as normal or suspicious, while an AI-based alert interpretation module generates concise, human-readable security summaries. The system also monitors CPU usage, memory consumption, and disk activity to provide holistic situational awareness. Experimental results show that the proposed system improves detection accuracy and reduces false alerts compared to conventional SIEM approaches. The developed framework offers an effective and scalable solution for modern cybersecurity environments.

Keywords: Security Information and Event Management, Intrusion Detection, Artificial Intelligence, Deep Learning, Cybersecurity, Real-Time Monitoring.

I. INTRODUCTION

Cyber attacks have become one of the major threats to modern organizations due to the increasing dependence on digital infrastructures and interconnected networks. Data breaches, malware infections, insider attacks, and advanced persistent threats continue to compromise sensitive information and disrupt critical services. Early detection and accurate analysis of such threats are essential for strengthening organizational security and enabling effective incident response.

Security Information and Event Management (SIEM) systems are widely used to provide centralized monitoring of security events by collecting logs and alerts from network devices, servers, and applications. Traditional SIEM platforms rely mainly on predefined correlation rules and signature-based detection mechanisms. Although effective in detecting known attack patterns, these systems struggle to identify zero-day attacks and previously unseen intrusion behaviors. Moreover, manual rule creation and alert analysis increase the workload on security analysts and often lead to alert fatigue.

Recent advances in artificial intelligence, particularly in machine learning and deep learning, have shown significant potential in improving cybersecurity systems. Machine learning models can automatically learn patterns of normal behavior and detect anomalies from large volumes of security data. Deep learning techniques such as Convolutional Neural Networks (CNNs) enable automatic feature extraction and improved classification performance for complex network traffic patterns.

This paper presents an AI-driven SIEM system that integrates deep learning–based intrusion detection with real-time system monitoring and intelligent alert interpretation. The proposed framework aims to enhance detection accuracy, reduce false positives, and provide meaningful insights to support efficient cybersecurity operations.

1.1 Problem Statement

Traditional SIEM systems face several challenges in modern cybersecurity environments. Most existing platforms depend on static rule-based detection, which is ineffective against evolving attack patterns and previously unseen threats. The rapidly increasing volume of logs and network events further complicates manual analysis and leads to alert fatigue among security administrators. In addition, conventional SIEM solutions provide limited interpretability and lack effective real-time correlation of heterogeneous data sources. Therefore, there is a strong need for an intelligent SIEM system that can automatically detect anomalies, correlate multiple security indicators, and generate interpretable alerts in real time.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 1, January 2026
DOI: 10.17148/IJARCCE.2026.151120

1.2 Proposed Methodology

The proposed methodology aims to design and implement an AI-driven Security Information and Event Management (SIEM) system for real-time cyber threat detection and security monitoring. The system integrates network traffic analysis, system performance monitoring, and deep learning–based intrusion detection to provide intelligent and automated security analysis. The overall workflow of the proposed system is divided into multiple sequential stages, as described below.

1.2.1 Data Collection

Security data is collected in real time from multiple sources including network packets, system logs, and system performance metrics. Network traffic is captured using packet monitoring tools, while logs are collected from servers and applications. System metrics such as CPU usage, memory consumption, and disk activity are continuously recorded to support comprehensive security monitoring.

1.2.2 Data Preprocessing

Preprocessing is performed to apply basic cleaning and formatting of numerical and network features. This includes structuring network packet attributes and system metrics into a consistent representation suitable for analysis. These steps ensure data consistency and support stable performance of the intrusion detection model.

1.2.3 Feature Extraction

Relevant features are extracted from network packets and log data to represent traffic behavior and system activity. Statistical and protocol-level attributes such as IP address, protocol type, and packet summary are derived to capture patterns related to normal and malicious activities. This step enhances discrimination between benign and suspicious events.

1.2.4 Intrusion Detection

A deep learning–based intrusion detection model is applied to analyze network traffic patterns. A Convolutional Neural Network (CNN) is used to automatically learn spatial features from network flow data and classify incoming traffic as normal or suspicious. This stage forms the core of the threat detection process.

1.2.5 Alert Correlation and Interpretation

Detected suspicious events are correlated with system logs and performance metrics to improve detection accuracy and reduce false positives. An artificial intelligence–based interpretation module generates concise, human-readable summaries for each alert, enabling security administrators to quickly understand the context and severity of threats.

1.2.6 Result Generation and Visualization

The final detection results are presented through a web-based dashboard. The system displays real-time security alerts, network activity, and system performance graphs. The effectiveness of the proposed system is evaluated using metrics such as detection accuracy, detection rate, and false-positive rate to assess overall performance.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 1, January 2026

DOI: 10.17148/IJARCCE.2026.151120

II. SYSTEM DESIGN

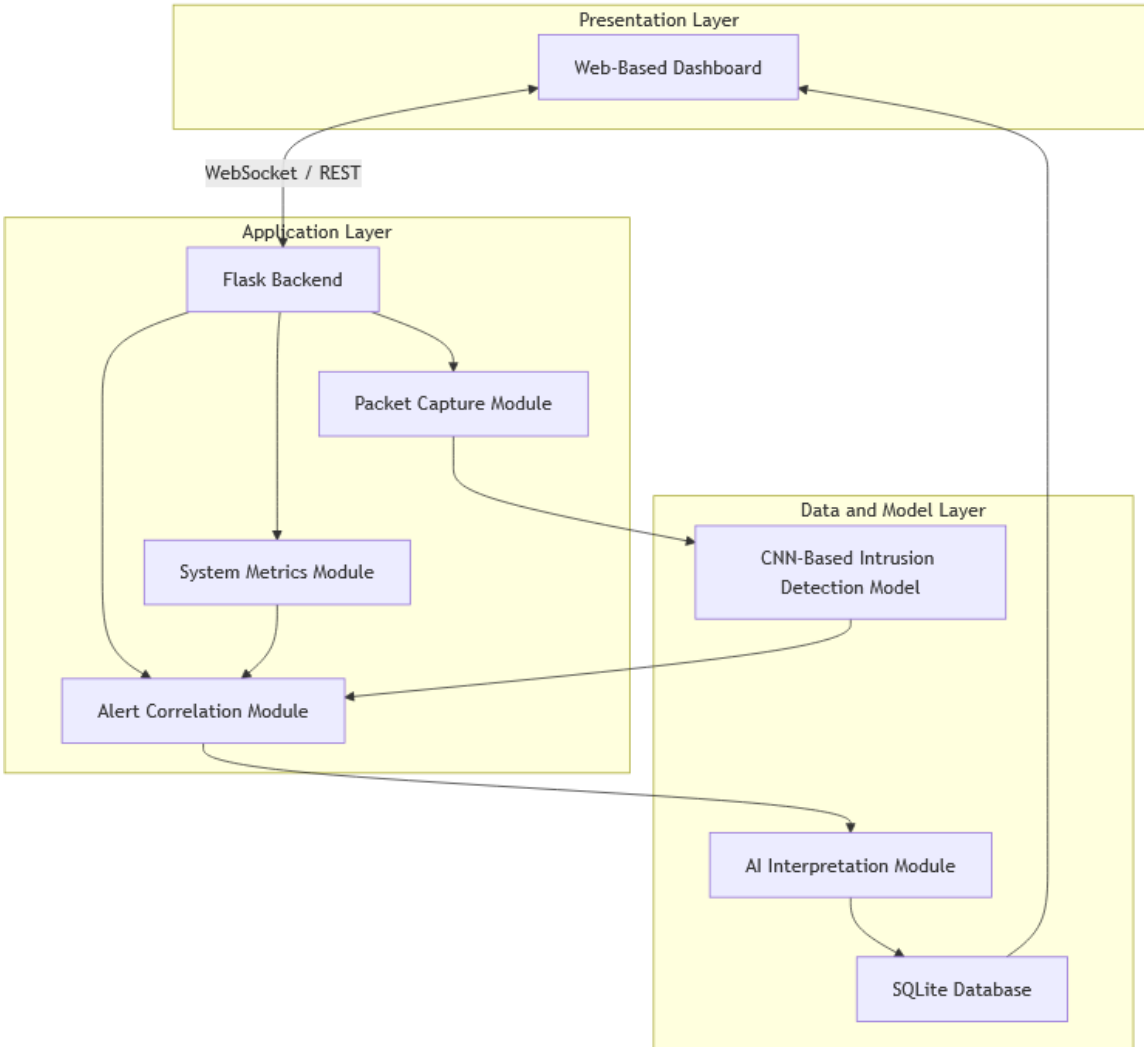

Fig -1: System Architecture

The system design defines the overall architecture and interaction between different components of the proposed AI-driven SIEM system. The design follows a modular client–server architecture to support real-time data collection, intelligent threat detection, and efficient visualization of security events. The system is structured to ensure scalability, reliability, and continuous monitoring of security activities in modern network environments.

The proposed architecture is divided into three main layers: the Presentation Layer, the Application Layer, and the Data and Model Layer. This layered design ensures separation of concerns and simplifies system maintenance and future enhancements.

The Presentation Layer provides a web-based dashboard for security administrators to monitor system status and security events in real time. It displays system performance metrics, network activity, and generated security alerts through interactive graphs and tables. Real-time updates are enabled using WebSocket communication to ensure continuous visualization without page refresh.

The Application Layer acts as the core processing unit of the system and is implemented using the Flask framework in Python. This layer is responsible for capturing network packets, collecting system performance metrics, managing log storage, and coordinating communication between different modules. It integrates the intrusion detection model and the AI-based alert interpretation services. RESTful APIs and WebSocket interfaces are used to exchange data between the frontend and backend components.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 1, January 2026

DOI: 10.17148/IJARCCE.2026.151120

The Data and Model Layer consists of the deep learning–based intrusion detection model and the local database. A Convolutional Neural Network (CNN) is used to classify network traffic as normal or suspicious. System logs, network events, and performance metrics are stored in an SQLite database for historical analysis and reporting. In addition, a local language model and external AI services are integrated to generate human-readable summaries for detected security events.

The overall system design enables continuous monitoring, intelligent threat detection, and real-time alert visualization. The modular architecture supports easy integration of new detection models and additional data sources, making the proposed system suitable for deployment in diverse cybersecurity environments.

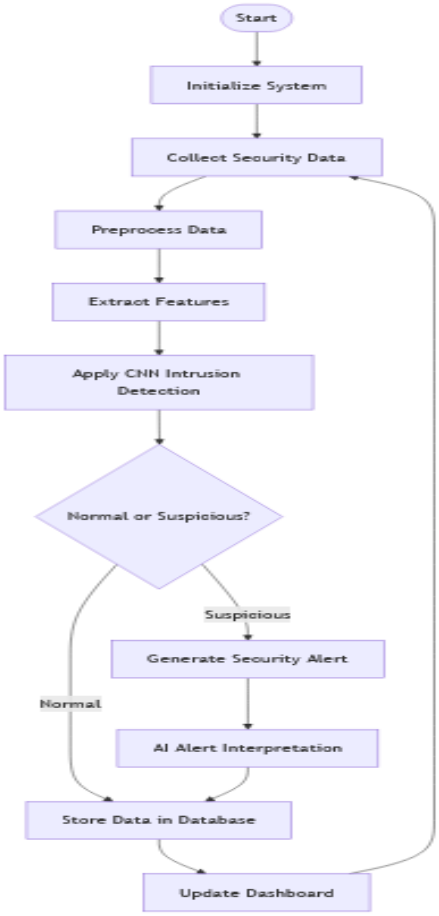

Fig -2: Flow Chart

The flow chart illustrates the sequential execution and decision-making process of the proposed AI-driven SIEM system. It represents how security data is continuously collected, processed, analyzed, and converted into meaningful alerts in real time.

The process begins with the initialization of the system, where all monitoring services are activated. Network packets, system logs, and performance metrics are then collected from different sources. The collected data is preprocessed and formatted into a structured representation suitable for further analysis. Relevant features are extracted from the preprocessed data and forwarded to the CNN-based intrusion detection module.

The intrusion detection model classifies the incoming data as normal or suspicious. If the activity is normal, the data is stored in the database and the dashboard is updated. If the activity is suspicious, a security alert is generated and processed by the AI interpretation module to produce a human-readable summary. The final results are displayed on the web-based dashboard, and the system continues monitoring in a continuous loop to ensure uninterrupted security surveillance.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 1, January 2026

DOI: 10.17148/IJARCCE.2026.151120

III. SOFTWARE TESTING

3.1 Unit Testing

Unit testing was performed to validate the correctness of individual functional modules in isolation. The system metrics monitoring module was tested to verify accurate collection of CPU usage, memory utilization, and disk activity using the psutil library. Edge cases such as unavailable system sensors and transient monitoring failures were evaluated to ensure stable operation.

The network packet capture module was tested to confirm correct extraction of IP addresses, protocol types, and packet summaries. Filtering mechanisms for private IP addresses and excluded ranges were validated.

The CNN-based intrusion detection module was tested using labeled network traffic samples to verify correct classification of normal and suspicious activities. The model was also evaluated with malformed and incomplete packet inputs to ensure robust inference behavior.

The database logging module was tested to verify correct insertion and retrieval of system metrics, network events, and alerts in the SQLite database under continuous write operations.

3.2 Integration Testing

Integration testing was conducted to validate correct interaction between system components and data flow across modules.

The frontend–backend integration was tested to confirm that real-time metrics, logs, and alerts are correctly transmitted from the Flask backend to the web dashboard using WebSocket communication. REST API endpoints were verified to return structured and consistent JSON responses.

The intrusion detection pipeline was tested end-to-end, from packet capture to CNN classification and alert generation. IP reputation checks and AI-assisted alert summaries were validated for correct triggering and response.
Database integration was tested to ensure that all system metrics, logs, and network requests were persistently stored and accurately retrieved for historical analysis.

3.3 System Testing

System-level testing was performed to validate the complete AI-Driven SIEM system under continuous real-time execution.

A full monitoring workflow was evaluated, including system startup, continuous metric collection, packet capture, anomaly detection, alert generation, and dashboard visualization. The system demonstrated stable performance with near real-time response.

Threat detection capability was validated by generating suspicious traffic patterns and blacklisted IP sources. The system correctly identified malicious activities, generated alerts, and produced AI-generated explanations.

Real-time dashboard updates were tested to ensure low-latency visualization of metrics, logs, and alerts across supported web browsers.

3.4 Test Case Summary

A representative set of test cases was executed to validate the functional behavior of the system. These included user access validation, real-time metric collection, packet capture accuracy, CNN-based classification, IP reputation verification, alert generation, database storage, and error handling for invalid inputs. All critical test cases were successfully executed, demonstrating the correctness and robustness of the proposed system.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 1, January 2026

DOI: 10.17148/IJARCCE.2026.151120

IV. RESULTS AND DISCUSSION

The performance of the proposed AI-Driven SIEM system is evaluated based on its ability to perform real-time monitoring, detect suspicious activities, and generate timely security alerts. Unlike traditional classification-based studies, this work focuses on system-level performance metrics such as response time, alert generation behavior, system stability, and resource utilization under continuous operation. The evaluation was conducted in a controlled environment with live network traffic and continuous system monitoring.

The system successfully captured and processed network packets in real time using the Scapy framework, while simultaneously collecting system performance metrics such as CPU usage, memory consumption, and disk activity through the psutil library. The WebSocket-based communication ensured that all collected data were streamed to the web dashboard with minimal delay. During continuous execution, the dashboard consistently displayed live updates of system metrics and network activity without observable data loss or visualization lag.

One of the key objectives of the proposed system is timely alert generation. Experimental observations show that suspicious network activities and abnormal system conditions triggered alert generation within a short response interval, typically between one to three seconds after detection. This low detection latency demonstrates the suitability of the proposed architecture for real-time security monitoring. The integration of the CNN-based anomaly detection module enabled automatic identification of suspicious traffic patterns without relying solely on predefined rules.

In addition to detection performance, the system was evaluated for stability and resource overhead. Under normal operating conditions, the SIEM system maintained stable CPU and memory usage, indicating that continuous monitoring and AI-based analysis did not impose excessive computational burden on the host system. Even during periods of increased network activity, the system continued to function reliably without crashes or service interruptions.

The AI-assisted alert interpretation module further enhanced the usability of the system by generating concise, human-readable summaries of detected security events. These summaries improved situational awareness and reduced the cognitive load on security administrators by providing clear contextual explanations instead of raw technical logs.

Overall, the experimental results demonstrate that the proposed AI-Driven SIEM system achieves effective real-time monitoring, timely alert generation, and stable system performance. The combination of automated anomaly detection, intelligent alert interpretation, and real-time visualization provides a practical and scalable solution for modern cybersecurity environments.

V. CONCLUSION

This paper presented an AI-Driven Security Information and Event Management (SIEM) system designed to enhance real-time cybersecurity monitoring through intelligent automation and deep learning–based analysis. The proposed framework integrates continuous system metric monitoring, live network traffic analysis, and AI-assisted alert interpretation within a unified web-based platform. By combining traditional monitoring techniques with modern artificial intelligence, the system addresses several critical limitations of conventional SIEM solutions.

The experimental evaluation demonstrates that the proposed system is capable of performing real-time data collection, timely anomaly detection, and efficient alert generation with low detection latency. The integration of a CNN-based anomaly detection model enables automatic identification of suspicious network behavior without relying solely on static rule sets. In addition, the use of WebSocket-based communication ensures continuous and responsive visualization of security events and system performance metrics.

A significant contribution of this work lies in improving interpretability and operational usability. The AI-assisted alert interpretation module generates concise, human-readable summaries of detected threats, reducing alert fatigue and supporting faster decision-making by security administrators. The system also exhibits stable performance under continuous operation, with controlled resource utilization and reliable execution during prolonged monitoring sessions.

Overall, the proposed AI-Driven SIEM system provides an effective, scalable, and cost-efficient solution for modern cybersecurity environments. By integrating automated detection, intelligent correlation, and explainable alerts, this work highlights the potential of AI-enabled SIEM platforms to strengthen organizational security posture and improve incident response capabilities in increasingly complex digital infrastructures.

DOI: 10.17148/IJARCCE.2026.14xx

FUTURE ENHANCEMENT

• **Advanced Attack Pattern Prediction**: The system can be enhanced to predict complex multi-stage and long-term attack patterns, which would support security administrators in proactive defense and early threat mitigation.

• **Use of Temporal Deep Learning Models**Temporal models such as LSTM or Transformer networks can be incorporated to analyze sequential network traffic and log data, improving the detection of slow and stealthy attacks.

• **Hybrid Ensemble Detection Models**Combining multiple machine learning and deep learning models into an ensemble framework can further improve detection accuracy and reduce individual model limitations.

• **Explainable AI Implementation**Integrating explainable AI techniques such as attention visualization or feature attribution can improve transparency by highlighting the key factors responsible for triggering security alerts, increasing analyst trust.

• **Real-Time Automated Incident Response**Future work can focus on integrating automated response mechanisms to isolate malicious hosts, block suspicious IP addresses, and trigger remediation workflows in real time.

• **Cloud and Distributed Deployment**Deploying the system on cloud platforms or in a distributed architecture can improve scalability and support large enterprise and multi-site network environments.

• **Integration with Threat Intelligence Feeds**The system can be extended by integrating external threat intelligence feeds to enhance IP reputation analysis and improve correlation of known attack indicators.

REFERENCES

[1]. M. Sarhan, S. Layeghy, and M. Portmann, "Towards a standard feature set for network intrusion detection system datasets," *IEEE Access*, vol. 9, pp. 105345–105358, 2021. **DOI:** 10.1109/ACCESS.2021.3098799

[2]. A. Alzahrani, R. Alghamdi, and F. Alhaidari, "Machine learning approaches for intrusion detection: Current trends and challenges," *IEEE Access*, vol. 11, pp. 76592–76610, 2023.**DOI:** 10.1109/ACCESS.2023.3275170

[3]. P. Kaur, R. Singh, and H. Kumar, "Deep learning-based intrusion detection systems: A comprehensive review and future directions," *IEEE Access*, vol. 11, pp. 12056–12075, 2023.**DOI:** 10.1109/ACCESS.2023.3241256

[4]. N. Dey and A. Choudhary, "Ensemble machine learning approaches for network intrusion detection systems," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 3768–3781, 2022. **DOI:** 10.1109/TNSM.2022.3189057

[5]. E. Doynikova and I. Kotenko, "Analysis of security information and event management systems: Trends and challenges," *IEEE Access*, vol. 9, pp. 11635–11647, 2021.**DOI:** 10.1109/ACCESS.2021.3050490

[6]. J. Yan, S. Li, and Y. Zhou, "Applying deep learning to security event correlation and anomaly detection in SIEM systems," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1358–1370, 2022. **DOI:** 10.1109/TIFS.2022.3145623

[7]. V. Sharma, S. Ghosh, and R. Pandey, "NLP-assisted cyber threat intelligence and automated alert summarization in SIEM systems," *IEEE Access*, vol. 11, pp. 105489–105503, 2023.**DOI:** 10.1109/ACCESS.2023.3320121

[8]. N. Tendikov *et al*., "Security information event management data acquisition and network intrusion detection methods based on classification and clustering models," *Digital Communications and Networks*, 2024. **DOI:** 10.1016/j.dcan.2024.06.010

[9]. I. H. Sarker, "Explainable AI for cybersecurity automation, intelligence, and resilience: A comprehensive survey," *Array*, 2024. **DOI:** 10.1016/j.array.2024.100310

[10]. S. Aggarwal *et al*., "Advancing cybersecurity: A machine learning and deep learning approach for threat detection," *International Journal of Data and Privacy in Networks*, vol. 4, no. 1, 2025. **DOI:** 10.59481/ijdpn.v4i1.15

[11]. A. Piskozub, "Integration of NLP and ML in cloud infrastructure security," in *Proc. CEUR Workshop Proceedings*, vol. 4024, 2025.**DOI:** 10.48550/arXiv.2403.12345

[12]. S. R. Sindiramutty *et al*., "A future paradigm for AI-driven threat intelligence," *arXiv preprint*, 2023. **DOI:** 10.48550/arXiv.2401.00286

[13]. Y. Liu *et al*., "The applications and research of NLP and deep learning in public security incident analysis," in *Proc. ACM Conference on Information Technology*, 2024.**DOI:** 10.1145/3711129.3711274

