Mitigating Alert Fatigue in Real-Time Cyber Threat Detection; A Self-Contextualizing Log Intelligence Framework

1st Sundas Fatima Shafique
Department of Information & Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
f23bince1m04055@iub.edu.pk

2nd Syeda Eeman Fatima
Department of Information & Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
f23bince1m04020@iub.edu.pk

3rd Aoun Muhammad
Department of Information & Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
aoun.muhammad@iub.edu.pk

4th Sana Tariq
Department of Information & Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
sana.tariq@iub.edu.pk

***Abstract*—Alert fatigue problems can have serious consequences for security. In modern cybersecurity environments, passive security log generation and increased cyber threats are the cause of massive log data. In this log flood, excessive security alerts and high false positive rates are very common. It hides real attacks. That is why there is a need for an intelligent log analysis system along with real-time threat detection, especially through a self-contextualising log intelligence framework. The proposed framework is a hybrid approach of machine learning-based Self-Contextualizing (SC)system design that cleans security logs, then collects and structures them, so logs are easily understood, which improves threat analysis and alert management. The designed framework evaluates data using standard cybersecurity metrics, which include accuracy, precision, recall, F1-score, and false positive rate. The proposed framework aims to reduce alert fatigue, improve detection accuracy for threats, and enhance cybersecurity monitoring in real-time environments. The framework integrates Zeek for network monitoring, Filebeat for log collection, Elasticsearch for data storage and indexing, and NetworkX for graph-based contextualization. *Index Terms*—Alert Fatigue, Cyber Threat Detection, Log Intelligence, Machine Learning, Anomaly Detection, SIEM, Intrusion Detection System, Real-Time Monitoring**

I. Introduction

In the proposed system, both implemented and conceptual frameworks are included. In the implemented environment, log collection is done through Zeek, preprocessing through Filebeat and data storage and indexing are done through Elasticsearch. Moreover, the contextualization process is simulated through graph-based modelling of NetworkX, through which the relationship between security systems can be understood. Machine learning models like Random Forest, SVM, LSTM, and Autoencoders are evaluated in a controlled environment. So, their comparative analysis can be done. Along with this, Large Language Models like GPT-4 are used for contextualization, but Large Language Model Meta AI(LLaMA) is considered for future extensions for intelligent log interpretation. Experimental evaluation is conducted by the combination of publicly available log datasets (network security logs, intrusion detection datasets, authentication logs datasets) and synthetically generated SIEM logs(fake or simulated logs generated that mimic a real SIEM environment).In these datasets, network traffic logs, authentication records, and intrusion detection alerts are included. The dataset is divided into a ratio of 80:20 for training and testing by splitting. In pre-processing, log normalization, duplicate removal, and feature extraction techniques are applied, which are important for machine learning models. The proposed framework is compared with Security Information and Event Management(SIEM) based alert generation systems and standard machine learning-based intrusion detection approaches. From this comparison, it is clear that the proposed system shows improvement in alert reduction, detection accuracy, and contextual understanding. The novelty of the proposed framework is that it integrates graph-based contextual analysis with machine learning-based alert classification, through which structured interpretation of raw security logs is possible and alert fatigue is significantly reduced in real-time environments. Traditional SIEM systems only use static rules or isolated ML models, but our system reduces alert fatigue and provides intelligent prioritisation by understanding dynamic log relationships.

A. Related Work

In the past, traditional IDS (intrusion detection systems) and SIEM tools were used for cybersecurity. They mainly work on signature-based detection. system was effective for detecting known attacks but failed when an unknown attack was performed on it. With the passage of time, as attacks

DOI: https://doi.org/10.67535/tsp.000003.007
Published under the ICACNC 2026 Proceedings

became more sophisticated, the limitations of the system were seen obviously on a large scale [1], and the system lacks adaptability with modern devices.[2] [3]. Later on, anomaly-based detection approaches were introduced, which tried to differentiate between normal and abnormal activities. Most of the time, benign activities were considered suspicious, which caused an increase in alert fatigue[4][5]. Machine learning techniques introduced new directions in cybersecurity, like clustering algorithms and classification methods like K-MAP and Support Vector Machine; decision trees helped in detection accuracy, but they were also only useful for labelled datasets. These techniques also fail because they can't adapt to a model's dynamic environment [6][7]. In deep learning approaches, Long Short-Term Memory and neural networks use temporal autoencoders so that sequential data and log patterns can be classified easily. They have proven beneficial in network traffic monitoring and computational analysis, but the issue occurs in real-time implementation and cost issues, which is quite challenging [8]. In this way, Security Information and Event Management systems are continuously improving, where AI techniques are integrated for the improvement of log correlation and event prioritisation. Some frameworks have introduced context-aware analysis, through which alerts are filtered more intelligently and are prioritised, but integration and scalability issues still exist. Moreover, recent research focuses on self-adaptive and automated systems, which improve their performance through feedback mechanisms and evolve detection strategies, which reduces false positives and increases overall system efficiency. Although present research has made a lot of progress, a big research gap exists, especially in alert fatigue, real-time log intelligence reduction, and contextual understanding. To address this gap, our Self-Contextualizing-Eye design framework is proposed, which will provide real-time context-aware analysis through adaptive learning so that detection performance will be more accurate and reliable [9]. At first, a lightweight Context-Aware Cybersecurity Framework is proposed that is designed for a pervasive computing environment. But there is a limitation in real-time alert prioritisation and adaptive learning, which is important for modern cybersecurity systems.[10].

Similarly,[11] introduced LogHub, a large-scale dataset platform that supports AI-based log analysis. [12]developed an AI-based automated log analysis tool that improves pattern recognition, but it also lacks real-time processing and adaptive response to attacks.

In addition, [13] a study was conducted on a systematic study on log file analysis for smart troubleshooting in Industry 4.0, emphasising the importance of log data, but it was not specifically targeting cybersecurity threats.[14]A system was proposed that provided a decentralised computing framework focusing on data privacy and control, but it does not address alert fatigue or threat detection. Furthermore,[15] the issue of alert fatigue in Security Operations Centres (SOCs) suggested technological improvements, although their approach lacks intelligent real-time prioritisation mechanisms.

[16]Integrated Security Information and Event Management and intrusion detection system with machine learning for real-time threat analysis, but the system also lacks deep contextual understanding of logs, which is also a very big limitation.[17]compared open-source intrusion detection tools such as Snort, Suricata, and Zeek, identifying Zeek as a powerful log-generating system. Moreover,[18]discussed adaptive learning using artificial intelligence, which enables systems to improve through feedback, but was not specifically applied to cybersecurity log analysis.[19]explored graph-based approaches for network monitoring and botnet detection, demonstrating strong contextual capabilities but with high computational complexity. [20]proposed graph-based neural network techniques for capturing relationships in data; however, their work is not focused on real-time alert systems or cybersecurity applications. This framework also extends beyond conventional adaptability to emphasising the necessity of dataset-aware Machine learning model selection to optimise the Security Information and Event Management system. However, the issue was that it was computationally expensive. In this project, there will be four different ML models used to evaluate their performances on different datasets but it was not specifically designed for cybersecuirty[21].

II. Methodology

This section outlines the overall methodology for the SCLI (self-contextualising log intelligence) system that is proposed. Its main aim is to reduce real-time cyber attack detection and alert fatigue. Existing systems like anomaly detection, signature-based and deep learning approaches have many limitations, like computational cost, high false positives, and lack of real-time adaptability. To solve these problems, our system is proposed. This system provides a context-aware, adaptive and lightweight pipeline. It collects logs, contextualises, processes and intelligently analyses them.

A. Proposed System Design

The project starts with the design phase, where the proposed SCLI (Self-Contextualising Log Intelligence) framework architecture was prepared. The main objective of this design was to connect log collection, preprocessing, contextual analysis, machine learning detection, and alert prioritisation in one complete pipeline.

The system was designed in such a way that logs coming from different sources are first collected and then converted into a clean and structured format. Raw logs usually contain noise, repeated entries, and unorganised values, so preprocessing is required before analysis. After that, a contextual engine adds more useful information to logs, such as time patterns, source behaviour, historical activities, and event relations. This helps the system understand the log events better than traditional rule-based systems.

An intelligent analysis layer was also included in the framework. In this layer, AI and machine learning models such as GPT-4 are used to identify suspicious activities, unknown attacks, and abnormal patterns. These models read the log content and try to understand if the activity is normal or

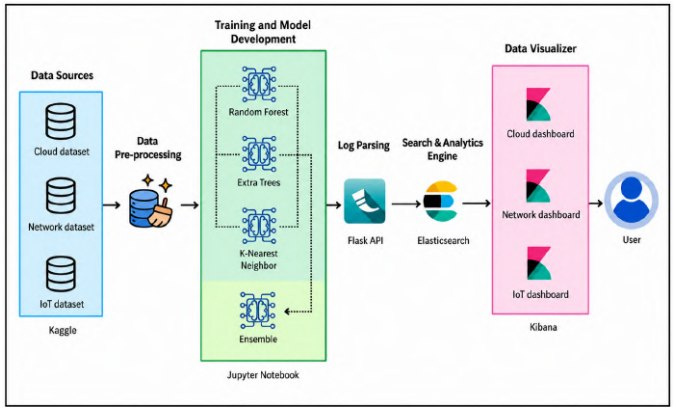

Fig. 1: main architecture

malicious. This makes the system more adaptive and capable of handling modern cyber threats. After threat detection, the system performs alert prioritisation. Alerts are ranked according to their severity and importance. High-risk alerts are forwarded immediately, while repeated or low-priority alerts are filtered or suppressed. This process helps to reduce alert fatigue and allows security analysts to focus only on important incidents.

The proposed framework was also designed for scalability and near real-time performance. It can process a large amount of logs continuously and provide faster detection results. Because of this, the system is suitable for enterprise networks and Security Operations Centre (SOC) environments where thousands of alerts are generated daily. Overall, the proposed system design provides an integrated, smart, and efficient solution for cyber threat detection. It combines log intelligence, contextual understanding, AI-based analysis, and smart alert management in one framework to improve detection accuracy and reduce unnecessary workload.

B. Baseline Anaysis

In the previous systems, the following problems were found: they were only signature-based systems, meaning they only detect known attacks and were vulnerable to zero-day threats. Anomaly detection systems give highly false-positive alerts. They were computationally expensive and rarely deployed in real-time. Hybrid models were introduced; there was partial improvement, but scalability and log integrity issues still exist. For example, there is no real-time alert prioritisation, no adaptive learning, no intelligent filtering and no contextual log analysis. All these limitations verify that there is a need for a context-aware framework.

C. Data Collection & Log Handling

In this research, we are collecting log data from various sources like network traffic logs, system event logs, user activity logs and public datasets (available on Google) and some we have created in our tested environment. The data can also be collected from simulated enterprise environment tools. Used for log handling are Filebeat (a lightweight log shipper, real-time log forwarding) and Elasticsearch for data storage and indexing(fast for searching and querying). For log analysis, advanced AI models are used; GPT-4, the main purpose is to understand the log meaning. We are using GPT-4 as a semantic log interpreter. Its work is to understand the logs and convert them into structured and meaningful text. It just provides a summary in natural language of log events and identifies abnormal patterns. The output of GPT-4 is then used with machine learning models for final detection and alert prioritization. This provides better analysis than traditional machine learning models.

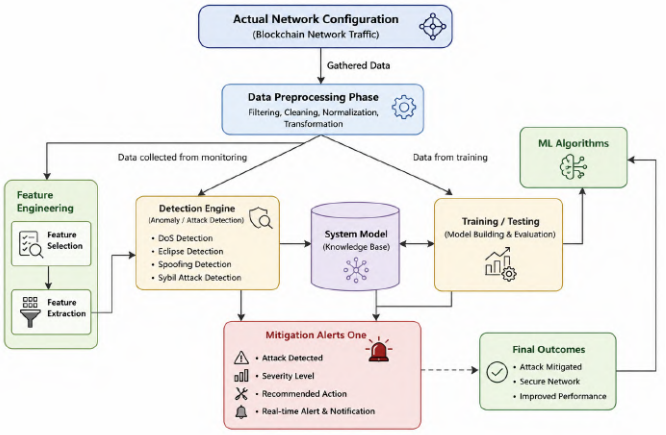

Fig. 2: Actual Network Configration

D. Feature Extraction

Raw log data is noisy and inconsistent. So pre-processing is almost compulsory because it helps with handling missing values, data normalisation, log parsing(converts into a structured format)and noise removal. So we first collected the data, cleaned the noise from it, and arranged it accordingly. This step increases the model accuracy and reduces the false positive rate. In the next step, the features of a system are extracted, such as IP address patterns, login frequency, session duration, and time-based behaviour. In statistical feature engineering, it is done so that the model can understand the context better.

E. Contextualization Using Network X

Instead of understanding logs in raw form, we analyse them by making graph-based relationships, in which a relationship is established between the users, IP addresses, and events. tool used for this is NetworkX; it's a Python library that provides a better context approach to the system.

F. Anomaly Detection

This is a core stage where the model applies artificial intelligence and machine learning (AI/ML). For machine learning, random forest and SVM are used. For deep learning Long short-term memory (time-based patterns) and autoencoders(anomaly detection) are used mainly. These models detect normal vs abnormal behaviours. Contextual analysis is the main novelty part; every alert must be analysed with

context. User behaviour and historical logs must be combined. so that the attack patterns can be understood more clearly. This step significantly reduces false positives, and it has increased detection reliability.

TABLE I: Class Distribution Before and After Data Balancing for IDS Model

Traffic Class	Original Dataset	Processed Dataset
Benign	1,020,450	185,320
Malicious	3,785,210	210,875

G. Alert Prioritization & Alert Fatigue Reduction

Alert prioritisation is important because Security Operations Centre analysts receive thousands of attacks, and the system deals with them. Our system ranks the alerts according to their severity; low-risk alerts are filtered, and critical alerts are highlighted. If we consider the attack ratio of the attacks we get daily, the alerts that the system deals with on a daily basis are almost 10,000+, of which false positives are 70-80% of the time, and almost 40-60% of the time is wasted. This step plays a crucial role in reducing alert fatigue.

TABLE II: Distribution of Attack Types in the Dataset

Attack type	Number of instances
SMURF	2,640,500
NEPTUNE	1,120,300
Back	3,450
POD	265
DOS	979
Buffer overflow	33
Load module	8.5
PERL	4
Rootkit	10
FTP write	9
Guess password	53
IMAP	12
MultiHop	7
PHF	5
SPY	2
Warez client	102
Warez master	20
IPSWEEP	13,481
NMAP	2,316
PORTSWEEP	11,413
SATAN	15,892
Normal	972,781

H. Adaptive Learning Mechanism& Performance Evaluation

Feedback loop is added into the system. So, the analysts can collect feedback to retain the model and the system is improved. This feature is not available in traditional systems; this makes our framework smart. We used metrics to evaluate the system's performance, like accuracy, precision, recall, F1-score, and false positive rate. This section explains the system specifications that are used for building an implemented SIEM and IDS for industrial use. The mentioned research uses open-source tools to build Security Information and Event Management for live analysis based on machine learning in an intrusion detection system. The open source system can help IT technicians in industry and companies to implement and use this research on their network to monitor the cyber-attack. The general overview of our proposed system can be seen in Fig. 4.

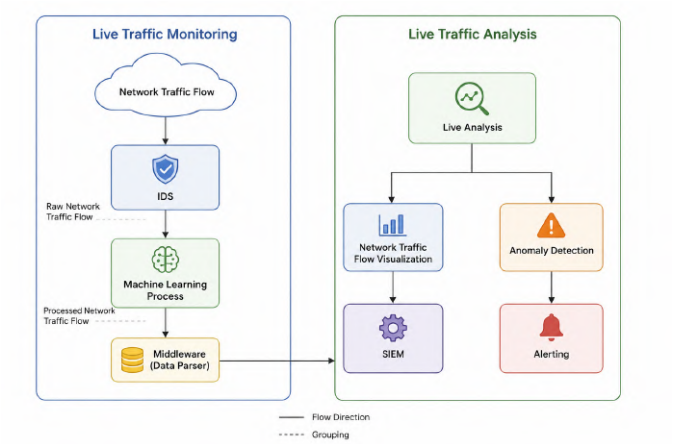

Fig. 3: Live traffic monitoring and analysis flow

I. Zeek As Intrusion Detection System

Zeek can perform multi-layer analysis of network activities. Zeek is an IDS that can be used with other third-party applications, which makes it easier to analyse network flow traffic to look for anomalies over the internet because logs are divided into different types of logs, as shown in Table 3. Zeek

TABLE III: Log And Description For Data-analysis

Log Name	Description
Connection log	Transmission Control Protocol (TCP) / User Datagram Protocol (UDP) / Internet Control Message Protocol (ICMP) connection
DHCP Lease log	Dynamic Host Configuration Protocol (DHCP) leases
DNS Query log	Domain Name System (DNS) activity
HTTP Access log	Hypertext Transfer Protocol (HTTP) request and replies
FTP Session log	File Transfer Protocol (FTP) activity

system generates many logs, among which the Connection log is the most important. It is used for the detection of anomalies because the connection log is the foundation of other logs generated from it. The connection logs are generated from layer 3- 4 on the Open Systems Interconnection(OSI) model. It is the input for machine learning tools used to analyse whether the network traffic flow shows suspicious behaviour or can be called an anomaly over the network.

J. Adaptive Learning & Feedback

System continuously improves itself. The feedback from the security analyst is collected, the dynamic candidate set is updated, and then the model is retrained for it, and the system is updated.

K. Perfromance Evaluation

Evaluation metrics are maintained and checked regularly. The following are the evaluation metrics for the system. Accuracy(the alerts obtained should be authentic), precision(the false positive rate must be reduced), recall(alerts generated should not be fake alarms), F1-score (a balance between recall and precision should be achieved), and false positive rate(it should be minimal, almost none). In Fig 4 comparative performance of machine learning models is shown which are trained and tested under same dataset and same experimental conditions. In this fig a visual comparision of accuracy, precision, recall and F1-score is given.GPT-4 was used only as a semantic log contextualization component and was not evaluated as a standalone classification model. Therefore, its performance metrics are not included in Table IV.

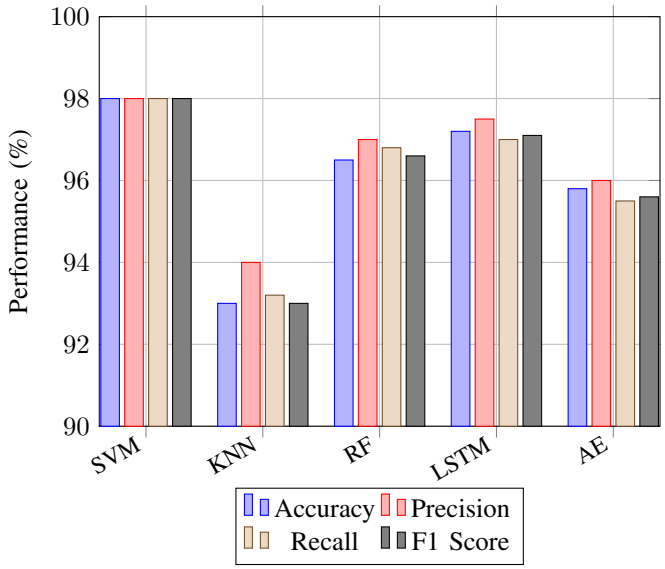

Fig. 4: Performance evaluation of machine learning models in IDS

L. Implementation Enviornment

The tools we used are Filebeat and Elasticsearch. Filebeat is a very lightweight log shipper and real-time log forwarding tool, and it collects logs from various sources. Elasticsearch helps in data storage along with indexing. It helps in fast searching and querying; it handles large datasets. The difference is that Filebeat just collects data and forwards it, while Elasticsearch stores data and also analyses it. To analyse logs in raw form, we contextualised them using networkX, in which graph-based relationships are made. relationship is built between users' IPs and events. This system approach helps better context understanding.

M. System Workflow

It collects the logs from the system, cleans those logs and gives them a properly structured format; then context is added to the logs(meaning of the logs is added to it), the AI model GPT-4 analyses the logs, and then alerts are generated. The alerts are prioritised according to their severity and based on their importance. Then the system again reviews the logs to confirm that they are authentic. Lastly, feedback is taken from the security analysts they improve and update the system accordingly.

III. Result & comparison

To evaluate the proposed SCLI (Self-Contextualising Log Intelligence) framework, multiple cybersecurity log datasets and simulated enterprise environments are used. Experimental results have shown that this system has performed significantly better than traditional rule-based and simple analogical station methods. In the previous system, the false positive rate was observed up to 70-80% whereas our proposed system has reduced it to 20-30%Ḋue to this reduction, the Security Operations Centre analyst receives only relevant and high-priority alerts, which has reduced the workload to a noticeable level. It also gave strong results relating to accuracy and efficiency in detection. Traditional IDS and Security Information and Event Management-based methods had an average detection rate of around 65-75% while our framework has achieved an accuracy of 85-99%Ṡimilarly, the threat detection rate is observed up to 91-95%, which is significantly better than previous systems. Due to Real-time log analysis and contextual validation, hidden attacks and suspicious behaviour were identified early. Every machine learning model is evaluated 5 times(5 independent runs). This ensures that results can be stable and unbiased. Final results are reported in the form of average values. Through this, the effect of randomness and dataset variation is reduced, and the model shows better reliability. Moreover, a 95% confidence interval is applied so that the statistical significance of the result can be verified. The observed variance was low, which proved that the model's performance is stable under the same data conditions.

TABLE IV: Performance Evaluation of Machine Learning Models in IDS

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
SVM	98.0	98.0	98.0	98.0
KNN	93.0	94.0	93.2	93.0
Random Forest	96.5	97.0	96.8	96.6
LSTM	97.2	97.5	97.0	97.1
Autoencoder	95.8	96.0	95.5	95.6

Table 4 shows a performance comparison for different machine learning models. All models are evaluated under the same dataset and the same preprocessing pipeline. This ensures a fair comparison can take place. Results clearly show that advanced models of support vector machine (SVM) and ensemble methods provide better performance.Regarding alert management perspective, the system has shown remarkable improvement. In our dataset, we worked with 500+ alerts by generating them, but in a real-life situation, 10,000+ alerts were generated on a daily basis in enterprise environments; the proposed model reduced that to 40-60 % through intelligent filtering and

prioritization, through which only actionable alerts are shown to the analysts and repetitive or low-risk notifications are automatically suppressed. Detection-to-response time, which was up to hours before, is now in seconds, near real-time. From the overall comparison, it is concluded that the proposed SCLI framework not only improves the detection and accuracy but also enhances the operational efficiency of the system. Traditional systems were static and rigid, while the proposed system continuously improves through adaptive learning and analyst feedback. That is why, for a modern Security Operations Centre environment, this system provides more practical and scalable solutions.

IV. FUTURE WORK

As we have performed on a small scale as an experiment, the proposed framework can be validated in the future by deploying it in real-world enterprise and Security Operations Centre environments. The system is mainly tested through simulated datasets and controlled environments; that's why large-scale practical deployment will give better analysis of scalability and robustness. Evaluating its behaviour in different industries like banking, healthcare and cloud infrastructure will be useful. It will increase the trust level, and the decision-making process will be better; moreover, multilingual log understanding and natural language summarisation features can be added. In future versions, a blockchain-based log integrity mechanism can be integrated so that collected logs can be made tamper-proof and secure.

V. CONCLUSION

In this research paper, an ML-based Self-Contextualising Log Intelligence (SCLI) framework has been proposed that is designed for real-time cyber threat detection and alert fatigue mitigation. To address common limitations in existing systems like rule-based dependencies, high false positives and slow response time, an intelligent and adaptive pipeline is developed. This framework collects logs, cleans them, gives them structure, and contextualises them. Then it detects suspicious patterns and malicious activities through AI and ML models. It prioritises the alerts on the basis of their risk value. Through alert prioritisation and alert filtering, it sends only important alerts to analysts, which reduces the unnecessary workload.
Experimental evaluation has shown that the system has reduced false positives, detection accuracy is improved, and response time has been made significantly faster. This framework continuously improves itself through feedback from analysts and adaptive learning.In conclusion, the SCLI framework is a robust, scalable and intelligent solution for modern cybersecurity operations. If implemented with proper deployment and continuous monitoring, it can enhance the SOC team's productivity and also provide strong protection to organisations against evolving cyber threats.

REFERENCES

[1] Shahroz Tariq, Mohan Baruwal Chhetri, Surya Nepal, and Cecile Paris. Alert fatigue in security operations centres: Research challenges and opportunities. *ACM Computing Surveys*, 57(9):1–38, 2025.

[2] Xiaoyu Wang, Xiaobo Yang, Xueping orLiang, Xiu Zhang, Wei Zhang, and Xiaorui Gong. Combating alert fatigue with alertpro: Context-aware alert prioritization using reinforcement learning for multi-step attack detection. *Computers & Security*, 137:103583, 2024.

[3] Mohamed Salah Mohamed and Abdullahi Arabo. A siem-integrated cybersecurity prototype for insider threat anomaly detection using enterprise logs and behavioural biometrics. *Electronics*, 15(1):248, 2026.

[4] Noora Zidan Khalaf, Al Barazanchi, Israa Ibraheem, AD Radhi, Pritesh Shah, and Ravi Sekhar. Development of real-time threat detection systems with ai-driven cybersecurity in critical infrastructure. *Mesopotamian Journal of CyberSecurity*, 5(2):501–513, 2025.

[5] Paul Kearney, Mohammed Abdelsamea, Xavier Schmoor, Fayyaz Shah, and Ian Vickers. Combating alert fatigue in the security operations centre. *Available at SSRN 4633965*, 2023.

[6] Jasmin Bharadiya. Machine learning in cybersecurity: Techniques and challenges. *European Journal of Technology*, 7(2):1–14, 2023.

[7] Tao Ban, Takeshi Takahashi, Samuel Ndichu, and Daisuke Inoue. Breaking alert fatigue: Ai-assisted siem framework for effective incident response. *Applied Sciences*, 13(11):6610, 2023.

[8] Kingsley David Onyewuchi Ofoegbu, Olajide Soji Osundare, Chidiebere Somadina Ike, Ololade Gilbert Fakeyede, and Adebimpe Bolatito Ige. Real-time cybersecurity threat detection using machine learning and big data analytics: A comprehensive approach. *Computer Science & IT Research Journal*, 4(3):478–501, 2024.

[9] Hugo Plácido da Silva and Pietro Cipresso. *Computer-Human Interaction Research and Applications: 8th International Conference, CHIRA 2024, Porto, Portugal, November 21–22, 2024, Proceedings, Part I*. Springer Nature, 2025.

[10] Jalal Al-Muhtadi, Kashif Saleem, Sumayah Al-Rabiaah, Muhammad Imran, Amjad Gawanmeh, and Joel JPC Rodrigues. A lightweight cyber security framework with context-awareness for pervasive computing environments. *Sustainable Cities and Society*, 66:102610, 2021.

[11] Jieming Zhu, Shilin He, Pinjia He, Jinyang Liu, and Michael R Lyu. Loghub: A large collection of system log datasets for ai-driven log analytics. In *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*, pages 355–366. IEEE, 2023.

[12] Prerna Lohar and Trupti Baraskar. Automated ai tool for log file analysis. In *2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)*, pages 1762–1766. IEEE, 2025.

[13] Sania Partovian, Alessio Bucaioni, Francesco Flammini, and Johan Thornadtsson. Analysis of log files to enable smart-troubleshooting in industry 4.0: a systematic mapping study. *IEEE Access*, 12:147640–147658, 2023.

[14] Yitong Cheng, Yang Yu, and Zhichao Hua. Dpcapsule: A decentralized private computing system with self-controlled data. In *Proceedings of the 16th International Conference on Internetware*, pages 233–244, 2025.

[15] Sruthi Soman and Zoheir Ezziane. Leveraging tech towards keeping ahead of cyber threats and alleviating security analysts' alert fatigue. In *Proceedings of Eighth International Conference on Information System Design and Intelligent Applications*, pages 1–19. Springer, 2024.

[16] Adabi Raihan Muhammad, Parman Sukarno, and Aulia Arif Wardana. Integrated security information and event management (siem) with intrusion detection system (ids) for live analysis based on machine learning. *Procedia Computer Science*, 217:1406–1415, 2023.

[17] Abdul Waleed, Abdul Fareed Jamali, and Ammar Masood. Which open-source ids? snort, suricata or zeek. *Computer Networks*, 213:109116, 2022.

[18] Meet Ashokkumar Joshi. Adaptive learning through artificial intelligence. *Joshi, MA*, pages 41–43, 2024.

[19] Sofiane Lagraa, Martin Husák, Hamida Seba, Satyanarayana Vuppala, Moussa Ouedraogo, et al. A review on graph-based approaches for network security monitoring and botnet detection. *International Journal of Information Security*, 23(1), 2024.

[20] Tianyu Zhu, Leilei Sun, and Guoqing Chen. Embedding disentanglement in graph convolutional networks for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):431–442, 2021.

[21] Anis Masyitah Abd Bahrim, Mohsen Mohamad Hata, Noormadinah Allias, Zulaikha Hanan Bolhan, Mohd Yusof Darus, and Saidatul Izyanie Binti Kamarudin. Mitigating alert fatigue in security information and event management (siem): Ensemble machine learning integration with elasticsearch, logstash and kibana (elk). *JOIV: International Journal on Informatics Visualization*, 10(2), 2026.