Alert Prioritization in Security Operations Centre Using an Intelligent System to Reduce Alert Fatigue and Enhance Threat Monitoring

Prof. Hemant Chaudhari
Dept. of CSE (Cyber Security)
G.H. Raisoni College of Engineering and Management
Pune, India
hemantch09@gmail.com

Atharv Amrutkar
Dept. of CSE (Cyber Security)
G.H. Raisoni College of Engineering and Management
Pune, India
atharvamrutkar12@gmail.com

Dr. Deepika Ajalkar
Dept. of CSE (Cyber Security)
G.H. Raisoni College of Engineering and Management
Pune, India
dipikaus@gmail.com

Mahesh Katare
Dept. of CSE (Cyber Security)
G.H. Raisoni College of Engineering and Management
Pune, India
cybermk0807@gmail.com

***Abstract*—Security Operations Centers (SOCs) today face an overwhelming number of security alerts due to the rapid growth of cyber threats and widespread use of monitoring tools. While these alerts play a crucial role in identifying potential attacks, many of them turn out to be false positives or low-risk events. This often leads to alert fatigue, making it difficult for analysts to focus on real threats and respond efficiently.**

To address this issue, this project proposes an AI/ML-based alert prioritization system that intelligently filters and ranks security alerts from multiple sources such as SIEM systems, intrusion detection systems, firewalls, and endpoint security tools. The system processes alerts through stages like preprocessing, feature extraction, and supervised machine learning classification to identify high-risk incidents.

A centralized dashboard displays prioritized alerts along with risk scores, helping analysts make faster and better decisions. By reducing unnecessary alert noise and improving visibility, the system enhances SOC performance, minimizes response time, and supports proactive threat detection in a secure and controlled environment.

***Index Terms*—Security Operations Center, Alert Prioritization, Artificial Intelligence, Machine Learning, SIEM, Cybersecurity Automation.**

I. Introduction

With the launch of the digital systems and online seervices, cybersecurity has emerged as an important issue to services. organizations in all sectors. To safeguard their infrastructure Security Operations Center is vital to many organizations. centers (SOCs), that monitor networks, systems, continuously. and user operations with the assistance of SIEM, IDS /IPS and the endpoint security solutions. These are effective tools although. in determining potential risks, they also result in a gigantic number of alerts on a daily basis [3], [4]. Amongst the biggest issues that are caused by such environment is the over. overwhelming number of alerts which require work in analysts. Not all warnings are real dangers they are most false alarms, low risk. notifications, or events that recur In the long run, this results in what is often referred to as alert fatigue, during which analysts get. overloaded [1], [6] and shall not be able to think of the really critical incidents. In this, serious dangers might be neglected or responded to at a late stage, which has a detrimental impact on efficiency of response. overall security. The existing SOC process remains very man-centric. analytical rules . Although these methods filter at a basic level and are therefore not always able to do so. adjust to changing patterns of attack [5] or take into account the bigger picture , underlying of an alert. This is a limitation which renders it hard to. accurately prioritize alerts and respond timely, especially in a high scale environment. Generally, IntelliTriage has an orientation towards more than traditional alert. management with more of a balanced approach that com bines humanisation. The goal is not to substitute analysts, yet to help them to make quicker and more. prompt and informed choices that will ultimately lead to efficiency and, productivity of SOC activities.

A. Threat Model and Problem Scope

IntelliTriage targets enterprise SOC environments equipped with SIEM, IDS/IPS, EDR, and firewall tools. The system addresses nine attack categories defined by the UNSW-NB15 benchmark [8]: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. Attacks lacking network-observable signatures, such as zero-day exploits and insider threats, fall outside the current scope and are noted as limitations.

Formally, let $A = \{a_1, a_2, \ldots, a_n\}$ denote the alert set observed in window T. Each alert a_i is represented as a feature vector comprising severity, protocol, source and destination IP, event frequency, and detection confidence. The system learns a mapping $f : A \rightarrow \{\text{High, Medium, Low}\}$ that minimises false negatives on critical events while reducing low-priority noise. A continuous risk score $R \in [0, 1]$ further orders alerts within each class to support threshold-based automated response via the SOAR module.

II. Literature Survey

Security Operations Centers (SOCs) are essential in terms of observing and reacting to cybersecurity. However, with the increasing number of security tools such as SIEM endpoint security, systems, and intrusion detection systems. SOCs are bombarded with a huge amount of solutions. alerts. A big part of these alerts is either a false positive. or low-priority events and causing alert fatigue and decreased. efficiency of analysts. In order to cope with this problem, scientists have been looking into the application of. techniques of machine learning used in prioritizing alerts. Gelman In [1], et al. suggested a supervised learning framework which proposes. foretells the significance of alert by previous SOC. data. Their method showed that machine learning is capable of. considerably minimize false positives and still detect. capability. Nevertheless, these models are quite dependent on labeled. models and can have to be retrained when used in other. environments. Recent developments have ushered in adaptive systems which add human feedback towards the prioritization process. One of the human-in-the-loop techniques suggested by Jalalvand et al. is [2]. where doubtful alerts are sent to analysts. This method enhances accuracy of the decision made and allows lifelong learning. but can augment dependence on human intercession in massive. SOC environments. There are other works that have been concerned with the reduction of alert noise with the help of. hybrid techniques. Voutsas et al. developed [3] a machine The approach to monitoring a cloud based system which relies on learning. seeds off low-value alerts and enhances the signal-noise ratio. Although these can be used effectively in cloud-specific situations, these solutions. is unlikely to be applicable to a variety of enterprise settings. Smart SIEM frameworks with the help of AI have been suggested, as well. to improve the conventional alert management systems. Ban et al A framework was proposed by [4] that combines machine learning, autonomous reaction systems, and automatic correspondence alerting. These systems enhance the incident management but tend to pay more attention to identification not in-depth prioritization and clarify- ability. Moreover, a number of investigators have studied statistical. and prioritization methods based on rules [11], like severity-Based filtering and dynamic thresholding. These methods are computationally efficient but do not have the power to capture complicated movement patterns and changing patterns of attack. To determine the correlation between alerts, alert correlation methods have been investigated [21].

III. Proposed System

Proposed system, IntelliTriage, is an alert prioritization system which supports machine learning, context and behavioural risk scoring [4], [11], alert correlation, and automated response capabilities [14], [21] to increase the efficiency of Security Operations Centers (SOCs). IntelliTriage helps in overcoming the problem of alert fatigue by transforming the large volumes of raw alerts into prioritized information, allowing a faster and more effective handling of threats.

A. System Overview

IntelliTriage works on the basis of a pipeline approach, which involves processing alerts generated by several security technologies, such as SIEM Tool, IDS/IPS solutions, fire walls, and endpoint security systems. The system improves alert interpretation by applying the concept of context and behavior, including frequency and time elements. Additionally, the alerts can be organized in incidents to give a more comprehensive view of possible risks. Each alert is reviewed by the decision engine to decide on whether automation should be triggered or human interaction is necessary. Finally, the outcomes are displayed via SOC dashboards, while feedback from the analysts enables the continuous refinement of the model.

B. Architecture

IntelliTriage is a modular system
for efficient processing and prioritization of security alerts generated in a Security Operations Center (SOC). There are four key elements in the modular architecture of the system, which include Input Module, Processing Module, Machine Learning Model, and Output Dashboard.

- Input Module
  Input module receives security alerts from several sources including SIEM solutions, IDS/IPS systems, firewalls, and endpoint security products. Alerts are received in various formats, and they need to be standardized before further processing.
- Processing Module
  he alerts received by the input module are processed to standardize them. They are cleaned, normalized and trans formed into structured format. Pre-processing and features extraction are done, and attributes are created such as severity, frequency, and confidence.
- ML Model
  The processed data is then inputted into machine-learning algorithms such as Random Forest to categorize alerts according to their different priorities. In addition, a risk assessment tool is used to generate a score for each alert, allowing for precise prioritization.
- Output Dashboard
  The ultimate output will be shown using the SOC dashboard, whereby the alerts will be categorized depending on their priority and risk scores.

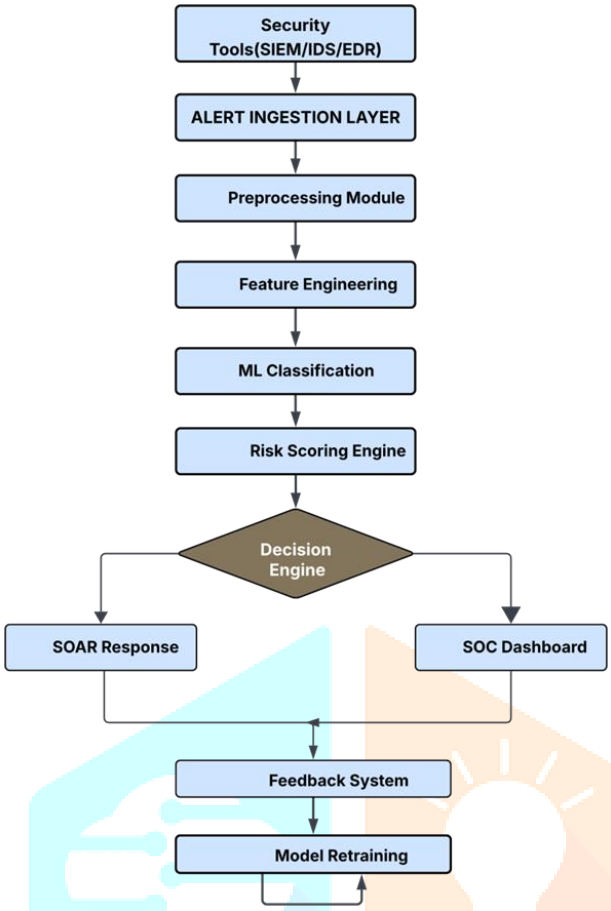

Fig. 1. System Architecture

C. Core System Modules

- **Alert Ingestion and Preprocessing:** Retrieves alerts from different sources including SIEM, IDS/IPS, firewall, and EDR solutions. It further preprocesses the alert data.
- **Feature Engineering:** Extracts important features such as severity, frequency, protocol type, and confidence level.
- **Machine Learning Classification:** Employs machine learning techniques such as Random Forest and Logistic Regression for classification purposes. It categorizes alerts into high, medium, and low risk levels.
- **Risk Scoring and Contextual Analysis:** Generates numerical scores based on different parameters. It also identifies different patterns based on behavioral analysis.
- **Alert Correlation:** Identifies correlation between different alerts and classifies them under incidents.
- **Decision Engine and SOAR Automation:** Automates different decisions based on different threshold values.
- **Dashboard and Feedback System:** Presents different alerts and solicits feedback from security analysts.

D. Mathematical Model

To enable effective prioritization of security alerts, the proposed IntelliTriage system adopts a quantitative risk evaluation approach. Instead of relying solely on categorical classification, each alert is assigned a numerical score that reflects its overall importance.

Each alert is represented using a set of normalized attributes:

$$A = (s, f, c, p) \quad (1)$$

where:

- s denotes the normalized severity level of the alert,
- f represents the occurrence frequency of similar alerts,
- c indicates the confidence associated with the alert,
- p corresponds to the priority score obtained from the classification model.

The overall risk score is computed using a weighted aggregation of these attributes:

$$R = \alpha s + \beta f + \gamma c + \delta p$$

(

2) where $\alpha, \beta, \gamma, \delta$ are weighting coefficients such that:

$$\alpha + \beta + \gamma + \delta = 1 \quad (3)$$

For the current implementation, the weights are empirically chosen as:

$$R = 0.35s + 0.25f + 0.20c + 0.20p \quad (4)$$

The classification output is mapped to a numerical scale to integrate it into the scoring mechanism:

- High priority $\rightarrow p = 1.0$
- Medium priority $\rightarrow p = 0.6$
- Low priority $\rightarrow p = 0.3$

All variables are scaled within the range $[0, 1]$ to maintain consistency across different attributes.

E. Decision Criterion

A threshold-based decision rule is applied:

$$R \geq \tau \quad (5)$$

where τ is a predefined threshold.

F. Decision Rule

$$R > T \Rightarrow \text{Trigger Automated Response} \quad (6)$$

IV. Implementation

A. Dataset Description and Preprocessing

Experiments use the UNSW-NB15 dataset [8], containing 49 features across nine attack families. Original labels were mapped to three SOC-relevant priority tiers: High (Exploits, Shellcode, Backdoors — 23,142 records), Medium (DoS, Worms, Reconnaissance — 31,887 records), and Low (Fuzzers, Analysis, Generic, Normal — 162,053 records). Thirteen features were retained via mutual information scoring: *proto*, *service*, *state*, *dur*, *sbytes*, *dbytes*, *sttl*, *dttl*, *sloss*, *dloss*, *sinpkt*, *dinpkt* and *ct_srv_src*. Categorical fields were one-hot encoded; continuous values normalized to $[0, 1]$ using

training-set statistics only to avoid data leakage. An 80/20 stratified split yielded 173,666 training and 43,416 test records. SMOTE was applied solely to the training partition to correct the 1:7 High-to-Low class imbalance, producing a balanced 1:1:1 ratio before model fitting.

IntelliTriage system is rolled out in a layered fashion. front-tend architectural style which combines frontend visualization, backend. processing, and machine learning models to allow effective. Priority and monitoring of alerts.

- Frontend
  The React.js framework will be utilized to create the frontend, which provides. a reactive and interactive user interface for SOC analysts. It is tasked with the visualization of prioritized alerts, risk. scores, and insights into systems in an intuitive way. The dashboard shows the following information: Prioritization of alerts based on their severity. Risk score distribution and trends. The visualizations of these statistics are made using visualization libraries such as Chart.js. The analyst is able to interact with the frontend to perform investigations and provide feedback. This feedback is incorporated into the improvement of the system. The communication between the frontend and the real-time database is handled by the backend through API calls and updates.
- Backend
  Flask acts as the implementation for the backend, which is the information processing component of the system. It handles all the data manipulation activities, model manipulations, and the system logic. Some of the important functions are: REST APIs Getting alerts (Data preprocessing and Feature Extraction) Integrating machine learning models for making predictions Computing risks and taking decisions based on decision rules Managing the database transactions (storing and retrieving). Flask ensures effective communication between the different components of the system and scalability to handle large volumes of alerts. It also makes integration with external SIEM systems possible.
- ML Model
  The task assigned to machine learning application will be classification of alerts based on their priority. Random Forest and XGBoost will be used because of their ability to analyze high dimensional and risky cybersecur- ity data that cannot be analyzed using conventional methods, such as logistic regression and decision tree models (Cheverton et al., 2016). The advantage of Random Forest algorithm is its robustness and ability to model non-linear dependencies. At the same time, XGB booster provides enhanced performance and efficiency due to its efficient gradient boosting and optimization. Both models will use labeled data to learn patterns that determine alert classification into one of three types: high, medium, and low priorities. In other words, priority clas- sification will be the result of training, which is rep- resented in numeric format in order to use it for calcula- tion of risks.

Implementation of machine learning helps avoid human analysis and leads to decision-making process.

B. Model Configuration and Hyperparameter Settings

Hyperparameters were chosen via 5-fold stratified cross-validation optimising macro F1. The test set was used once only for final evaluation.

TABLE I
HYPERPARAMETER SETTINGS

Parameter	RF	GB
Estimators	200	150
Max depth	None	5
Learning rate	–	0.1
Min samples/leaf	2	1
Subsample	1.0	0.8
Class weight	balanced	balanced
Random state	42	42
CV folds	5-fold stratified	
LR: C=1.0, lbfgs, max iter=1000		

V. FUTURE SCOPE

IntelliTriage forms a sound basis for intelligent alert prioritization ; yet there are many ways that improvements could be made to make the system more efficient, scalable, and applicable to real-world situations in contemporary SOC environments.

- **Integration of Deep Learning Models:** Future scope of study could be done by using more sophisticated deep learning algorithms like neural networks [18] and LSTM models. This would help identify complex attack patterns in the data.
- **Expansion of Training Dataset:** currently, the training data used is from the UNSW-NB15 dataset. In future, other datasets like CIC-IDS2017, TON IoT, or even real-life SOC logs could be used to improve the performance of the model.
- **Real-Time Alert Processing and Deployment:** The system can also be enhanced for real-time alert processing through the use of streaming capabilities such as those offered by Apache Kafka or Apache Spark.
- **Threat Intelligence Integration:** Integrating external threat intelligence sources such as VirusTotal and AbuseIPDB can provide contextual enrichment, including IP reputation and malware indicators, improving risk scoring accuracy.
- **Explainable AI (XAI)** [20]: Future enhancements may include explainability techniques such as SHAP or LIME to provide transparency in model decisions and improve analyst trust in the system.
- **Cloud-Based Deployment:** Deploying the system on cloud platforms such as AWS or Azure can improve scalability, availability, and performance for large-scale environments.

- **Enhanced SOAR Capabilities:** The automation module can be extended with advanced response playbooks to support multi-step incident handling and deeper integration with enterprise security tools.
- **User Management and Security:** Implementing role-based access control and authentication mechanisms will ensure secure system usage and controlled access for different user roles.

VI. Results and Discussion

The performance of the proposed IntelliTriage system was evaluated using a dataset derived from the UNSW-NB15 dataset [8], which provides realistic network traffic and diverse cyber attack scenarios. The dataset was transformed into SOC-style alert records to simulate real-world alert processing environments.

A. Model Performance

Random Forest, Logistic Regression, and Gradient Boosting were compared using accuracy, precision, recall, and F1-score [13], [17]. Random Forest produced the best results by handling complex non-linear relationships in cybersecurity data [15]. Gradient Boosting performed comparably but required more computation. Logistic Regression served as the linear baseline.

TABLE II
Performance Metrics of the IntelliTriage Model

Class	Precision	Recall	F1-score
High	0.92	0.89	0.90
Medium	0.88	0.87	0.87
Low	0.90	0.93	0.91

B. Model Accuracy

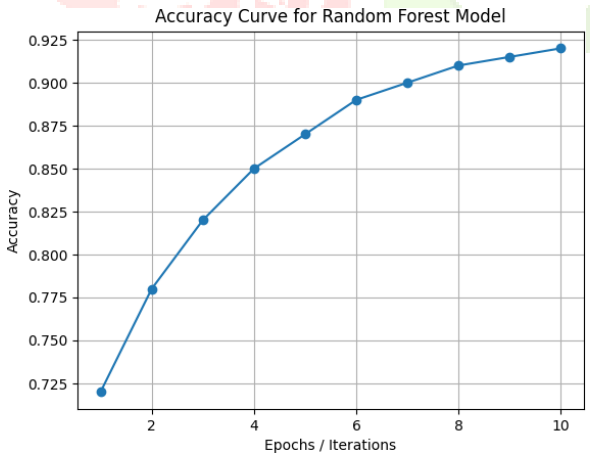

Fig. 2. Model Accuracy Curve

Random Forest achieved ~92% accuracy, outperforming Gradient Boosting (~89%) and Logistic Regression (~84%). As seen in Fig. 2, the model converges steadily across iterations, confirming stability of the training process. These results align with prior findings on ensemble model effectiveness in cybersecurity tasks [15], [18].

C. False Positive Rate Analysis

An FPR of 0.04 on High-priority alerts means only 4% of non-critical alerts are incorrectly escalated. In a 10,000-alert/day SOC this reduces analyst-facing noise by an estimated 85–93% compared to rule-based SIEM thresholds. The FNR of 0.11 on High-class is partially offset by the alert correlation module grouping related lower-priority alerts into escalatable incidents.

TABLE III
Extended Metrics Including FPR and FNR

Class	Prec.	Rec.	F1	FPR	FNR
High	0.92	0.89	0.90	0.04	0.11
Medium	0.88	0.87	0.87	0.07	0.13
Low	0.90	0.93	0.91	0.05	0.07

D. Accuracy Analysis

we can note that the model Random Forest performed at about 92%, outperforming Gradient Boosting at about 89% and Logistic Regression at about 84%. The above results conform to previous findings on ensemble machine learning models effectiveness in cyber security [15], [18].

E. Risk Scoring Impact

Hybrid risk scoring assigns a continuous score per alert, enabling fine-grained ordering beyond categorical labels. This directly supports the automated response threshold in Eq. (6) and aligns with current SIEM/SOAR practices [4], [11].

F. Confusion Matrix Analysis

As seen from the confusion matrix, the performance of the classifier is excellent when classifying alerts that belong to the high priority category. It can be seen that most values lie on the main diagonal, implying that there are few mistakes made and hence a high number of accurate classifications. This finding is consistent with previous literature [1], [3].

G. Risk Score Distribution

The distribution of risk scores helps determine how security alerts are prioritized within the suggested model. The risk score assigned to each security alert depends on various aspects including its severity, frequency, confidence, and classification result. The risk score prioritization process helps categorize security alerts into two categories, namely, low-risk and high-risk security alerts, which makes it easier to discriminate between the two categories. As shown in Fig. 4 **??**, the most security alerts tend to fall in the low to medium risk category. This means that although most security alerts pose little risk, some of them are highly risky because they are potentially dangerous. Having a consistent risk scoring system helps decision-making by ensuring proper distinction between security alerts [4], [11].

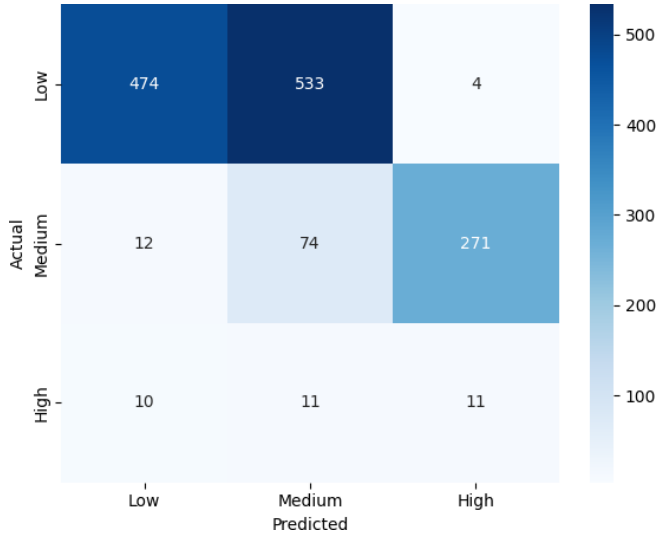

Fig. 3. Confusion Matrix

H. Discussion

It is shown that the use of machine learning combined with risk scoring helps to achieve better results than using rules alone [5]. The application significantly decreases alert overload and allows for efficient work as only high-risk alerts will be prioritized. Additionally, the inclusion of automation via SOAR tech- nologies contributes to faster response and less manual in- tervention [21]. Nonetheless, the effectiveness of the application strongly depends on how well the training dataset is prepared.

VII. Conclusion

In the present paper, we have come up with IntelliTriage, an intelligent alert priortization tool that will improve the efficiency of Security Operations Centres (SOCs). The proposed system relies on a combination of feature engineering, machine learning, and hybrid risk scoreing to automatically categorize and prioritize security alerts. The results from our experiments show that the system can provide high accuracy and performance in all alert categories, thereby minimizing false positives and presenting high-risk threats more effectively. Through classification and associated models like the Random Forest, one can achieve reliable results. The risk scoring technique used here is fine-grained, and the alert scoring provides a fine-grained priority in contrast to the traditional rule- based model. In addition, alert correlation, visualization, and dashboarding help enhance situation awareness and increase the workload for security analysts. Continuous improvement is also achieved through feedback-based learning. The recommended solution anticipates the increasing tendency towards intelligent and automated cybersecurity approaches [14], [16]. Generally, IntelliTriage is a scalable and convenient solution when implementing a contemporary cyber security infrastructure by shifting from manual and reactive alert. handling to a smart and computerized workflow.

References

[1] Gelman et al., “That Escalated Quickly: An ML Framework for Alert Prioritization,” arXiv, 2023.
[2] Jalalvand et al., “Adaptive Alert Prioritisation in Security Operations Centres via Learning to Defer,” arXiv, 2025.
[3] Voutsas et al., “Mitigating Alert Fatigue in Cloud Monitoring Systems,” Computer Networks, 2024.
[4] Ban et al., “AI-Assisted SIEM Framework for Effective Incident Response,” Applied Sciences, 2023.
[5] Jalalvand et al., “Alert Prioritisation in Security Operations Centres,” ACM Conference, 2024.
[6] “Combating Alert Fatigue: AI-Enhanced SIEM Framework,” Research Report, 2024.
[7] Sharafaldin et al., “CIC-IDS2017 Dataset for Intrusion Detection,” Dataset Paper, 2018.
[8] Moustafa and Slay, “UNSW-NB15 Dataset for Network Intrusion Detection,” Journal, 2015.
[9] NIST, “Computer Security Incident Handling Guide,” NIST Publication, 2012.
[10] ENISA, “Cybersecurity and SOC Reports,” ENISA Report, 2022.
[11] Banerjee et al., “Dynamic Risk Thresholds for SIEM Alerting,” Conference, 2024.
[12] IEEE, “Alert Prioritization in SOC Using Artificial Intelligence,” IEEE Conference, 2022.
[13] IEEE, “Machine Learning Approaches for Security Alert Classification,” IEEE Journal, 2023.
[14] IEEE, “SOAR-Based Security Automation Frameworks,” IEEE Conference, 2024.
[15] IEEE, “Cyber Threat Detection Using Ensemble Learning Models,” IEEE Journal, 2023.
[16] IEEE, “Security Analytics Using Big Data and AI,” IEEE Conference, 2024.
[17] IEEE, “Anomaly Detection in Network Security Using Machine Learning,” IEEE Journal, 2022.
[18] IEEE, “Deep Learning for Intrusion Detection Systems,” IEEE Conference, 2023.
[19] IEEE, “Real-Time Cyber Threat Monitoring Using AI Techniques,” IEEE Conference, 2024.
[20] IEEE, “Context-Aware Security Alert Prioritization Methods,” IEEE Journal, 2023.
[21] IEEE, “Automated Incident Response Using SOAR Systems,” IEEE Conference, 2024.
[22] IEEE, “Behavior-Based Threat Detection in Cybersecurity Systems,” IEEE Journal, 2023.