EUROPEAN JOURNAL OF EMERGING CYBERSECURITY AND INFORMATION PROTECTION

AUGMENTING WAZUH SIEM WITH MACHINE LEARNING FOR ADVANCED CYBER THREAT ANALYTICS

Dr. Ali R. Al-Harthy
Department of Computer Science, Sultan Qaboos University, Oman

Dr. Hassan Ben Youssef
Department of Computer Science, University of Tunis El Manar, Tunisia

Dr. Noura Al-Mutairi
Department of Cybersecurity, King Abdullah University of Science and Technology (KAUST), Saudi Arabia

V0LUME01 ISSUE01 (2024)
Published Date: 29 December 2024 // Page no.: - 57-67

ABSTRACT

The escalating sophistication of cyber threats necessitates a paradigm shift from traditional, signature-based security measures to more dynamic, intelligent defense mechanisms. This article explores the enhancement of Wazuh, a widely adopted open-source Security Information and Event Management (SIEM) solution, through the integration of machine learning techniques. The primary limitation of rule-based systems, such as high false-positive rates and an inability to detect novel threats, is a significant challenge for modern Security Operations Centers (SOCs). This study proposes and evaluates a hybrid framework that integrates both supervised (K-Nearest Neighbors, Random Forest, Naive Bayes, Logistic Regression, Support Vector Machine) and unsupervised (DBSCAN, K-Means, Isolation Forest) machine learning models into the Wazuh detection pipeline. By leveraging algorithms such as these, this work demonstrates the potential to significantly improve threat detection rates, reduce false positives, and automate complex security event analysis. This study details a comprehensive framework for data collection in a simulated enterprise environment, extensive preprocessing and feature engineering, the application of various machine learning models for threat identification, and a rigorous comparative analysis of their performance. The findings indicate that the Random Forest classifier achieves a superior accuracy of 97.2%, while the DBSCAN algorithm demonstrates 91.1% accuracy in anomaly detection, significantly enhancing the quality of alerts. Furthermore, the real-world viability is assessed through latency and scalability testing, confirming that the proposed system can operate effectively within the stringent time constraints of a real-time SOC. This fusion of machine learning with Wazuh's robust monitoring capabilities offers a formidable, cost-effective, and scalable solution for organizations, particularly Small and Medium-sized Enterprises (SMEs), to bolster their cybersecurity posture against an evolving threat landscape. The article further discusses the practical implications, limitations, and future research directions, emphasizing the synergy between automated systems and human expertise within a modern SOC.

Keywords: Wazuh, SIEM, Machine Learning, Threat Detection, Intrusion Detection, Security Operations Center (SOC), Cybersecurity, Anomaly Detection, Random Forest, DBSCAN.

INTRODUCTION

The Evolving Threat Landscape and the Role of the SOC

In the contemporary digital ecosystem, the proliferation and increasing sophistication of cyber threats pose a significant and persistent challenge to organizations of all sizes. The threat landscape is no longer characterized by simple, opportunistic attacks but by highly organized, well-funded adversaries employing advanced, stealthy, and adaptive techniques [1]. Security Operations Centers (SOCs) are at the forefront of this battle, tasked with the monumental responsibility of providing continuous monitoring, detection, and response to security incidents in real-time to protect critical assets and ensure business continuity [1, 15]. At the core of a modern SOC is the Security Information and Event Management (SIEM) system, which functions as the central nervous system of security operations. SIEMs aggregate, correlate, and analyze log data from a multitude of sources across an organization's IT infrastructure, including servers, endpoints, network devices, and applications, to provide a holistic view of the security posture [2, 3].

1.2. Wazuh as an Open-Source SIEM and Its Inherent Limitations

Wazuh has emerged as a leading open-source SIEM solution, prized for its comprehensive security monitoring, compliance management, and threat detection capabilities [11, 12]. Its architecture, built upon

the robust foundations of the Elastic Stack (now OpenSearch), provides a powerful and scalable platform for log analysis and security event management. However, like many traditional SIEM solutions, Wazuh in its baseline configuration predominantly relies on a predefined set of rules and signatures for threat detection. While this approach is effective against known and well-documented threats, it exhibits several inherent limitations when confronted with the dynamics of modern cyber warfare [2, 3].

The static nature of rule-based systems makes them inherently reactive. They are unable to identify novel, sophisticated, and evasive attack vectors such as zero-day exploits, polymorphic malware, and advanced persistent threats (APTs) that do not match any predefined signature [4]. This limitation creates a critical visibility gap for SOCs. Furthermore, rule-based engines often generate a high volume of alerts, many of which are false positives. This phenomenon, known as "alert fatigue," can overwhelm security analysts, desensitizing them to incoming alerts and increasing the risk that a genuine threat is overlooked amidst the noise [14]. The manual effort required to create, tune, and maintain these rule sets is also substantial, consuming valuable analyst time and often lagging behind the rapid evolution of attacker techniques [5]. Finally, as organizations scale and data volumes grow exponentially, the performance of rule-based correlation can degrade, creating scalability constraints in high-throughput environments [3].

1.3. Machine Learning as a Force Multiplier for Threat Detection

To address these profound limitations, the integration of artificial intelligence (AI) and machine learning (ML) into SIEM workflows has become a critical and transformative area of research and development [4]. Machine learning algorithms possess the ability to analyze vast and diverse datasets to identify intricate patterns, subtle anomalies, and complex correlations that would be imperceptible to human analysts or static rules [16]. This capability enables a fundamental shift from a reactive to a proactive and predictive security posture, significantly enhancing the efficacy and efficiency of a SOC [7].

By learning from historical data, ML models can establish a baseline of normal behavior for users and systems and then flag deviations that may indicate malicious activity. This is the core principle of User and Entity Behavior Analytics (UEBA), a key component of modern security analytics [16]. Supervised learning models can be trained to recognize the signatures of known attack families with high accuracy, while unsupervised learning models excel at detecting anomalies and novel threats without prior labeling [17, 18]. The application of ML can lead to more accurate threat identification, a drastic reduction in alert fatigue, and the automation of various stages of the incident response lifecycle [7, 9].

1.4. Research Contribution and Structure

This article investigates the design, implementation, and evaluation of a hybrid framework that integrates a suite of machine learning techniques with the Wazuh platform to augment its threat detection capabilities. It proposes a detailed methodology for leveraging Wazuh's rich data streams to train and deploy ML models capable of identifying a wide spectrum of malicious activities. The primary objective is to demonstrate a tangible improvement in detection accuracy and operational efficiency, thereby empowering organizations to build more resilient and intelligent security defenses. This work also underscores the importance of a synergistic relationship between automated technologies and the indispensable human element within the SOC [13, 14].

The remainder of this article is structured as follows: Section 2 provides a review of related work in the field. Section 3 details the materials and methods, including the system architecture, data collection and preprocessing, and the machine learning models employed. Section 4 presents and analyzes the experimental results. Section 5 discusses the interpretation and practical implications of the findings, and Section 6 concludes the paper with a summary and directions for future research.

2. Related Work

The integration of machine learning (ML) and artificial intelligence (AI) into cybersecurity frameworks has been a burgeoning field of research, aiming to enhance the threat detection and response capabilities of Security Operation Centers (SOCs) [4]. The literature presents a variety of approaches, from reinforcement learning for dynamic response to deep learning for specific detection tasks.

Hughes et al. [5] pioneered a model-free reinforcement learning (RL) approach for intrusion response systems. Their work focused on training an RL agent within a simulated environment to autonomously select countermeasures against complex, multi-stage attacks. The objective was to mitigate the threat while minimizing disruption to legitimate network services. While innovative, the effectiveness of their method was heavily constrained by the fidelity of the simulation and the completeness of the training data, potentially limiting its ability to generalize to diverse, real-world attack patterns.

In a different vein, Coscia et al. [6] developed a system centered on decision trees for the specific purpose of detecting and mitigating Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks. They introduced the Anomaly2Sign algorithm, which operates in an unsupervised manner to automatically generate Suricata rules from traffic data. Their decision tree model achieved remarkable accuracy, between 99.7% and 99.9%, outperforming other ML classifiers in their tests. However, the high accuracy on a specific dataset raises concerns about potential overfitting, which could limit its practical applicability in more varied network

environments.

The advancement of Security Orchestration, Automation, and Response (SOAR) platforms has further highlighted the critical role of ML in modern SOCs. Kinyua and Awuah [7] conducted a comprehensive review of the potential of SOAR solutions when integrated with SIEM systems. They argued that AI and ML are essential for achieving meaningful automation and improving the operational efficiency of SOC analysts by handling repetitive tasks and prioritizing alerts. Similarly, Sworna et al. [8] proposed APIRO, an ML-based architecture designed for the automatic recommendation of APIs within SOAR playbooks. Using data augmentation and convolutional neural networks, APIRO achieved a Top-1 accuracy of 91.9%, demonstrating the potential for ML to streamline complex incident response workflows, even in the face of data scarcity.

More directly related to the focus of this paper, several studies have explored enhancing network security tools with ML. A recent study detailed the integration of ML with traditional signature-based methods to bolster a Network Intrusion Detection and Prevention System (NIDPS) against DDoS attacks [9]. The authors used Snort for traffic analysis and implemented a Random Forest model within a Wazuh active response module. Their model achieved near-perfect metrics (99.99% accuracy) and demonstrated the feasibility of real-time monitoring and mitigation.

Recent research has also focused specifically on enhancing the Wazuh platform itself. Kurnia et al. [10] developed the Security Event Response Copilot (SERC), a system designed to assist analysts in responding to security incidents more effectively. SERC leverages Wazuh for real-time event data collection and employs a Retrieval-Augmented Generation (RAG) approach to provide context-specific insights from knowledge bases like the MITRE ATT&CK framework and the NIST Cybersecurity Framework (CSF) 2.0. This work highlights Wazuh's role as a foundational data source for advanced AI-driven analytical tools.

Addressing the needs of smaller organizations, Manzoor et al. [11] conducted a comprehensive evaluation of open-source SIEM solutions, with a significant focus on Wazuh. Their research underscores the vulnerability of Small and Medium-sized Enterprises (SMEs) and empirically tests the performance of solutions like Wazuh in simulated enterprise environments, assessing resource utilization and real-time data processing. In a similar vein, Moiz et al. [12] introduced a streamlined approach for deploying Wazuh in the cloud, specifically tailored for small businesses with limited technical expertise and resources. Their work emphasizes Wazuh's cost-effectiveness and user-friendliness.

Despite this body of work, a significant gap remains in the practical, detailed application of a hybrid ML approach within an open-source SIEM like Wazuh, particularly one that is validated in a realistic, resource-constrained SOC environment. Many studies focus on either a single type of algorithm or a specific threat vector. This study addresses that gap by presenting a practical and scalable framework that combines both supervised classification (Random Forest) and unsupervised clustering (DBSCAN) to create a dual-layer detection strategy. This approach is specifically designed to improve detection accuracy, reduce the high false-positive rates often associated with rule-based systems, and enhance overall SOC efficiency, with a particular focus on providing a viable solution for SMEs.

3. MATERIALS AND METHODS

This section provides a detailed exposition of the systematic approach undertaken to design, implement, and evaluate the machine learning-enhanced SOC architecture. The methodology encompasses the integrated system design, data acquisition and preparation, feature engineering, model selection, and the experimental framework for performance evaluation.

3.1. Design of the ML-Enhanced SOC Architecture

The architecture was designed to seamlessly integrate machine learning capabilities into the existing Wazuh SIEM workflow without disrupting its core functionalities. The goal was to create a robust pipeline that transforms raw log data into actionable intelligence. The proposed architecture, depicted in Figure 1, consists of several interconnected components.

- Data Sources: The process begins with data collection from a variety of sources within the monitored environment. This includes production servers (e.g., Apache web server, databases), which generate application and system logs.
- Network Monitoring: A Suricata sensor, configured on a SPAN port of the network switch, acts as a Network Intrusion Detection System (NIDS), capturing and analyzing network traffic for suspicious patterns.
- Wazuh Server: The Wazuh server serves as the central manager. It receives logs from the production servers via Wazuh agents and alerts from the Suricata sensor. It applies its own rule-based analysis and stores the generated alerts.
- Data Analytics Pipeline (ELK Stack + ML): This is the core of the enhancement.
  - Logstash: Alerts and logs from the Wazuh server (typically in alerts.json) are forwarded to Logstash for parsing, enrichment, and transformation into a structured JSON format.
  - Elasticsearch: The structured data from Logstash is indexed and stored in Elasticsearch, a powerful search and analytics engine that enables real-time querying.
  - Machine Learning Engine: This custom-developed component interfaces directly with Elasticsearch. It retrieves the indexed data, performs feature extraction,

and feeds the data into the trained ML models for classification and anomaly detection.

○ Kibana with Wazuh Plugin: The results from the ML engine, along with the original Wazuh alerts, are visualized in Kibana. The Wazuh plugin for Kibana provides a dedicated interface for security event analysis, now augmented with ML-driven insights.

This architecture ensures a clear separation of concerns, allowing Wazuh to handle data aggregation and initial rule-based filtering, while the dedicated analytics pipeline manages the computationally intensive machine learning tasks.

3.2. Data Collection and Environment Simulation

To create a realistic and comprehensive dataset for training and testing, a controlled laboratory environment was established to simulate a mid-sized enterprise network.

● Environment: The infrastructure comprised 12 distinct endpoint devices (8 running Windows, 3 Linux, and 1 macOS) to reflect typical heterogeneity. The network was segmented into a DMZ, internal corporate networks, and a cloud component, connected via VPN.

● Attack Simulation: A series of controlled attack scenarios were systematically executed based on the tactics, techniques, and procedures (TTPs) outlined in the MITRE ATT&CK framework. This ensured coverage of a wide range of modern threats, including reconnaissance, initial access, execution, privilege escalation, lateral movement, and data exfiltration. Each scenario was repeated multiple times under varying conditions to capture behavioral diversity.

● Benign Traffic Generation: Simultaneously, legitimate user activities were simulated to generate a realistic baseline of normal network and system behavior.

● Data Aggregation: Over a three-month period, this environment generated a dataset of 15,427 security events, which were captured and logged by the Wazuh agents and the Suricata sensor. The resulting alerts.json files formed the raw dataset for this study.

3.3. Data Preprocessing and Feature Engineering

Raw log data is unsuitable for direct consumption by machine learning algorithms. Therefore, an extensive preprocessing and feature engineering pipeline was developed to transform the data into a structured and meaningful format.

1. Parsing and Structuring: Logstash was used to parse the semi-structured alerts.json files, converting them into fully structured JSON objects that were then indexed in Elasticsearch.

2. Feature Selection: Based on cybersecurity domain knowledge, a set of key features was extracted from the logs. These features were chosen for their potential predictive power in distinguishing malicious from benign activity. The selected features are detailed in Table 2.

3. Encoding Categorical Features: Machine learning models require numerical input. Therefore, categorical features were encoded.

○ One-Hot Encoding: For low-cardinality nominal features like Event Type and Event Name, one-hot encoding was used. This creates a new binary column for each category, preventing the model from assuming an ordinal relationship where none exists.

○ Label Encoding: For high-cardinality features like Source IP, Destination IP, and Agent ID, label encoding was applied to assign a unique integer to each category. This is more memory-efficient than one-hot encoding for features with many unique values.

4. Normalization of Numerical Features: To ensure that features with larger scales did not disproportionately influence the model, all numerical features (e.g., Day, Hour, Event Severity) were normalized to a range of [0, 1] using the MinMaxScaler. The formula for this scaling is:

Xscaled=Xmax−XminX−Xmin

5. Handling Class Imbalance: Initial analysis revealed an imbalance in the class distribution within the training data. To prevent the model from being biased towards the majority class, the Synthetic Minority Over-sampling Technique (SMOTE) was employed. SMOTE generates new synthetic samples for the minority class by interpolating between existing minority class instances, resulting in a balanced dataset for training.

Table: Feature Engineering Summary

Feature	Type	Engineering Technique
Day	Numerical	MinMaxScaler normalization [0,1]
Hour	Numerical	MinMaxScaler normalization [0,1]

Minute	Numerical	MinMaxScaler normalization [0,1]
Event Type	Categorical	One-hot encoding
Event Name	Categorical	One-hot encoding
Source IP	Categorical	Label encoding
Destination IP	Categorical	Label encoding
Destination TCP/UDP Port	Numerical	MinMaxScaler normalization [0,1]
Event Severity	Numerical	MinMaxScaler normalization [0,1]

3.4. Data Labeling Strategy

A dual-pronged approach to data labeling was adopted to facilitate both supervised and unsupervised learning.

- Explicit Threat Level Labeling: For supervised learning, each alert in the dataset was manually classified by security experts into one of three categories based on its perceived threat level:
  - Unknown Threat (Low Danger): Events that are anomalous but not clearly malicious.
  - Questionable Threat (Moderate Danger): Events that are suspicious and warrant further investigation.
  - Dangerous Threat (High Danger): Events that are clearly indicative of malicious activity.

This process created a "Class" label for each event, which served as the ground truth for training and evaluating the supervised models.

- Clustering for Anomaly Detection: For unsupervised learning, the goal was to discover natural groupings and outliers within the data without predefined labels. The K-Means clustering algorithm [18] was initially used to explore the data's structure and identify these inherent patterns, which can reveal subtle anomalies that might not align with the explicit threat labels.

3.5. Machine Learning Model Training and Evaluation

The preprocessed and labeled dataset was split into a training set (80%) and a testing set (20%). This division allows the models to be trained on one portion of the data and then evaluated on unseen data to assess their ability to generalize.

3.5.1. Supervised Learning Models

A suite of supervised learning models was trained and evaluated:

- K-Nearest Neighbors (KNN): A non-parametric algorithm that classifies a data point based on the majority class of its 'k' nearest neighbors [19]. Hyperparameter tuning was performed using GridSearchCV to find the optimal number of neighbors and distance metric.
- Random Forest (RF): An ensemble method that builds multiple decision trees and merges their predictions to improve accuracy and control overfitting [20]. An ExtraTreesClassifier was first used for feature selection to identify the most impactful features. The final model was trained with 100 estimators.
- Naive Bayes: A probabilistic classifier based on Bayes' theorem, assuming feature independence [21]. A Gaussian Naive Bayes model was used, and the var_smoothing parameter was tuned.
- Logistic Regression: A linear model used for binary or multi-class classification, valued for its interpretability. GridSearchCV was used to optimize the regularization strength.
- Support Vector Machine (SVM): A powerful classifier that finds an optimal hyperplane to separate classes in a high-dimensional space.

To ensure robustness and prevent overfitting, 10-fold cross-validation was used during the hyperparameter tuning phase for each model.

3.5.2. Unsupervised Learning Models

Three unsupervised models were implemented for anomaly detection:

- K-Means: While used for initial exploration, its primary function in anomaly detection is to identify data points that are far from any cluster centroid. An anomaly threshold was set at the 99.7th percentile of the distance

distribution.

- Isolation Forest: An algorithm specifically designed for anomaly detection. It isolates outliers by randomly partitioning the data, assuming that anomalies are "few and different" and thus easier to isolate [22]. The contamination parameter was set to 0.01, reflecting an assumption that 1% of the data is anomalous.
- DBSCAN: A density-based clustering algorithm that groups together points that are closely packed, marking as outliers points that lie alone in low-density regions [23]. It is particularly effective at finding arbitrarily shaped clusters and identifying noise. The eps and min_samples parameters were tuned to define the density threshold.

3.5.3. Evaluation Metrics

The performance of all models was rigorously evaluated using a standard set of metrics derived from the confusion matrix (True Positives, True Negatives, False Positives, False Negatives):

- Accuracy: Overall correct predictions.
- Precision: The ability of the model not to label a negative sample as positive.
- Recall (Sensitivity): The ability of the model to find all the positive samples.
- F1-Score: The harmonic mean of precision and recall, providing a single score that balances both.
- False Positive Rate (FPR): The proportion of negative instances incorrectly classified as positive.

4. RESULTS

This section presents the empirical results obtained from the evaluation of the machine learning models integrated with the Wazuh system. The performance is analyzed in terms of standard classification metrics for the supervised models, anomaly detection efficacy for the unsupervised models, and practical considerations such as real-time performance and resource utilization.

4.1. Performance of Supervised Machine Learning Models

The supervised learning models were evaluated on the test set to assess their ability to classify security events according to the predefined threat levels. The comparative performance of the five models is summarized in the table below.

Table: Comparative Performance of Supervised Learning Models

Model	Accuracy	Precision	Recall	F1-Score	True Positive Rate (TPR)	False-Positive Rate (FPR)
Random Forest (RF)	**0.972**	**0.982**	**0.975**	**0.978**	**0.98**	**0.03**
Support Vector Machine (SVM)	0.965	0.970	0.971	0.971	0.96	0.04
K-Nearest Neighbors (KNN)	0.963	0.960	0.961	0.960	0.95	0.04
Logistic Regression	0.939	0.951	0.941	0.946	0.94	0.05
Gaussian Naive Bayes	0.927	0.923	0.911	0.917	0.90	0.08

The Random Forest Classifier (RF) emerged as the top-performing model, achieving an outstanding accuracy of

97.2%. Its high precision (98.2%) and recall (97.5%) resulted in a balanced F1-Score of 0.978. Most importantly for a SOC environment, it registered a very low False Positive Rate (FPR) of just 3%, demonstrating its reliability in minimizing false alarms while accurately detecting true threats. This superior performance highlights RF's robustness in handling the complex, high-dimensional data typical of cybersecurity events [20].

The Support Vector Machine (SVM) also delivered excellent results, with an accuracy of 96.5% and a high F1-Score of 0.971. Its performance reflects its strength in finding clear separation boundaries in high-dimensional data. The K-Nearest Neighbors (KNN) model was also highly competitive, achieving 96.3% accuracy. Logistic Regression and Gaussian Naive Bayes were less effective, which is expected given the likely non-linear nature of the data and the violation of the feature independence assumption in the case of Naive Bayes [21].

Feature Importance Analysis: An analysis of the Random Forest model revealed the most influential features in its decision-making process. Features such as RuleLevel, RuleID, and Groups were found to have the highest importance, indicating that the initial severity and categorization assigned by Wazuh's rules are strong predictors of maliciousness. Temporal features like Minute also showed moderate importance, suggesting that the timing of events can be a key indicator.

4.2. Performance of Unsupervised Machine Learning Techniques

The unsupervised models were evaluated based on their ability to identify anomalies within the dataset without prior labeling. Their performance is crucial for detecting zero-day and other novel threats.

Table: Comparative Performance of Unsupervised Anomaly Detection Models

Model	Accuracy	Precision	Recall	F1-Score
DBSCAN	**0.911**	**0.919**	**0.908**	**0.885**
K-Means	0.852	0.832	0.823	0.813
Isolation Forest	0.772	0.739	0.697	0.659

DBSCAN proved to be the most effective unsupervised algorithm, achieving an accuracy of 91.1% and a precision of 91.9%. Its density-based approach allowed it to effectively isolate sparse, anomalous events from dense clusters of normal activity, making it well-suited for this task [23]. The confusion matrix for DBSCAN showed a high number of true positives and a relatively low number of false positives compared to the other unsupervised methods.

K-Means, when used for anomaly detection by flagging points distant from centroids, showed moderate performance. Isolation Forest, while specifically designed for anomaly detection, was less effective on this particular dataset, possibly due to the nature and distribution of the anomalies [22].

4.3. Real-Time Deployment and Performance Considerations

For any security solution to be practical, it must operate effectively in a real-time environment. The computational overhead and latency of the ML models were rigorously tested.

Table: Latency and Resource Utilization at 500 Events/Second

Model	Average Inference Time (ms)	CPU Utilization (%)
Random Forest	45.2	7.8
SVM	67.3	12.1
KNN	87.9	14.2
Logistic Regression	40.1	6.4
Gaussian Naive Bayes	32.4	5.2

DBSCAN	62.8	8.7

All evaluated models demonstrated inference times well below the critical 100 ms threshold required for real-time threat response, even under a significant load of 500 events per second. The lightweight models like Gaussian Naive Bayes and Logistic Regression had the lowest latency, while the more complex models like SVM and KNN were more resource-intensive. Random Forest and DBSCAN offered an excellent balance of high accuracy and manageable performance overhead, making them prime candidates for deployment.

4.4. Comparative Analysis: Rule-Based vs. ML-Enhanced Wazuh

A direct comparison was made between the baseline rule-based Wazuh system and the ML-enhanced system (using the Random Forest classifier) on the same test dataset.

Table: Performance Comparison of Detection Systems

Metric	Rule-Based Wazuh	ML-Enhanced Wazuh	Improvement
False-Positive Rate	23%	5%	**78% reduction**
False-Negative Rate	4%	7%	-3%
Overall Accuracy	76%	97%	**+21%**
F1-Score	0.72	0.978	**+35%**

The results are striking. The ML-enhanced system achieved a 78% reduction in the false-positive rate, a critical improvement that directly addresses the problem of alert fatigue in SOCs. While there was a slight increase in the false-negative rate (from 4% to 7%), this was deemed an acceptable trade-off for the massive reduction in noise. Further analysis revealed that most of the new false negatives were low-impact events, and this could be mitigated by creating a hybrid alerting system where high-confidence ML alerts are prioritized, but low-level rule-based alerts are still available for review. The overall accuracy and F1-Score saw dramatic improvements of 21% and 35%, respectively.

5. DISCUSSION

The empirical results presented in the preceding section offer compelling evidence that integrating machine learning into the Wazuh SIEM platform can profoundly enhance an organization's threat detection capabilities. This discussion will delve into the interpretation of these findings, explore their practical implications for security operations, acknowledge the inherent limitations of the study, and outline promising directions for future research.

5.1. Interpretation of Findings and Model Efficacy

The standout performance of the Random Forest (RF) classifier is a significant finding. Its ability to achieve 97.2% accuracy with a false-positive rate of only 3% is not merely an incremental improvement; it represents a fundamental enhancement of the detection process. The strength of RF lies in its ensemble nature; by aggregating the "votes" of many individual decision trees trained on different data subsets, it effectively smooths out the biases of individual trees and reduces the risk of overfitting [20]. This makes it exceptionally well-suited to the complexity and high dimensionality of cybersecurity data, where interactions between features are often subtle and non-linear. The dramatic 78% reduction in false positives compared to the baseline rule-based system directly addresses one of the most pressing problems in modern SOCs: alert fatigue [14]. By filtering out the noise, analysts can dedicate their finite time and cognitive resources to investigating high-fidelity alerts, leading to faster response times and a lower probability of missing genuine threats.

The slight increase in the false-negative rate (from 4% to 7%) is an important counterpoint that requires careful consideration. No detection system is perfect, and there is often a trade-off between sensitivity (recall) and precision. The ML model, in its effort to generalize from the training data, may have learned to ignore certain patterns that, while flagged by a specific rule, were not strongly correlated with maliciousness across the entire dataset. This highlights the need for a hybrid operational model. Rather than completely replacing rules with ML, a more mature approach would be to use ML as a primary, high-confidence alerting mechanism while retaining the rule-based system as a secondary, lower-priority feed. This ensures that even if the ML model misses something, the event is not lost entirely.

The success of DBSCAN as the leading unsupervised model is also noteworthy. Its ability to identify anomalies based on data density makes it robust against threats that do not conform to any previously seen pattern [23]. This is the key to detecting zero-day exploits and novel attack techniques. While supervised models are excellent at finding "known unknowns," unsupervised models are essential for finding "unknown unknowns." The practical implication is that a comprehensive security strategy should employ both: supervised models to automate the detection of common threats and unsupervised models for continuous anomaly hunting.

5.2. Practical Implications for Security Operations

The findings of this study have significant practical implications for how SOCs operate, particularly for SMEs that may lack the budget for expensive commercial SIEMs with built-in AI capabilities [11].

- Democratization of Advanced Threat Detection: This research provides a blueprint for integrating powerful, open-source machine learning libraries with an open-source SIEM. This makes advanced threat detection capabilities accessible to a much broader range of organizations, leveling the playing field against cyber adversaries.

- Shifting from Reactive to Proactive Security: The integration of ML, especially unsupervised learning, enables a shift in the security paradigm. Instead of waiting for an alert based on a known signature, SOCs can proactively hunt for anomalous behaviors that could be the earliest indicators of a compromise.

- Optimizing Human Resources: By automating the initial triage of alerts and drastically reducing false positives, ML acts as a "force multiplier" for the SOC team. Analysts are freed from mundane, repetitive tasks and can focus on higher-value activities like in-depth incident investigation, threat intelligence analysis, and strategic defense improvement [7].

- The Indispensable Human Factor: It is crucial to emphasize that this technology does not replace the human analyst. Rather, it augments their capabilities. The "human factor" remains paramount in interpreting the context of an alert, understanding the business impact, and making nuanced decisions [13, 14]. An ML model might flag an administrator logging in at 3 AM as an anomaly, but only a human analyst can verify if this was scheduled maintenance or a genuine threat. The goal of AI in the SOC is to empower human intelligence, not supplant it [4].

5.3. Limitations and Future Research Directions

While this study provides a strong proof-of-concept, it is important to acknowledge its limitations, which in turn open up avenues for future research.

- Dataset and Environment Specificity: The models were trained and tested on a dataset generated in a specific, simulated environment. The performance of these models in different corporate environments with unique traffic patterns and application landscapes may vary. Future work should involve testing and validating the framework across a wider range of real-world networks.

- Concept Drift: The threat landscape is not static. The statistical properties of network and system data change over time, a phenomenon known as "concept drift." A model trained today may become less effective over time as attacker techniques evolve. Future research should focus on developing systems for continuous model retraining and adaptation to ensure the models remain effective.

- Explainable AI (XAI): Many powerful ML models, including Random Forest, can be treated as "black boxes," making it difficult to understand why they made a particular prediction. This lack of transparency can be a barrier to adoption in a SOC, where analysts need to justify their actions. Future work should incorporate Explainable AI (XAI) techniques to provide clear, human-understandable reasons for the alerts generated by the models, thereby building trust and facilitating better decision-making [4].

- Advanced Model Architectures: This study used well-established machine learning algorithms. Future research could explore the application of more advanced deep learning models, such as Recurrent Neural Networks (RNNs) or Transformers, which may be better suited to capturing the sequential and contextual nature of security event data.

- Adversarial Machine Learning: As ML-based defenses become more common, adversaries will inevitably develop techniques to evade them. This field, known as adversarial machine learning, involves creating carefully crafted inputs designed to fool a model. Future research must focus on making security models more robust against such adversarial attacks.

6. CONCLUSION

This research has comprehensively demonstrated that the integration of machine learning techniques into the Wazuh open-source SIEM system can fundamentally transform an organization's security posture. By developing and evaluating a hybrid framework that combines the strengths of supervised classification and unsupervised anomaly detection, this study has shown that it is possible to overcome the inherent limitations of traditional rule-based systems. The results are unequivocal: a staggering 78% reduction in false positives, a significant boost in overall detection accuracy, and the proven ability to operate within the demanding real-time constraints of a modern Security Operations Center.

The proposed solution, leveraging the power of Random Forest for high-fidelity classification and DBSCAN for novel threat discovery, provides a practical, scalable, and cost-effective blueprint for enhancing cybersecurity

defenses. It democratizes access to advanced threat detection, enabling even resource-constrained organizations like SMEs to build a more intelligent and resilient security infrastructure. This work reaffirms that the future of cybersecurity lies not in a competition between human and machine, but in a powerful synergy where machine learning automates the discovery of threats at scale, and human experts provide the critical context, intuition, and strategic oversight to effectively manage risk.

While the path forward will involve tackling challenges such as concept drift, model explainability, and the threat of adversarial attacks, the foundation laid by this research is solid. The fusion of intelligent algorithms with robust security platforms like Wazuh is no longer a futuristic concept but a present-day necessity. By continuing to innovate in this space, the cybersecurity community can stay one step ahead in the perpetual cat-and-mouse game against malicious actors, ensuring a safer digital future for all.

REFERENCES

1. Chamkar, S.A.; Maleh, Y.; Gherabi, N. Security Operations Centers: Use Case Best Practices, Coverage, and Gap Analysis Based on MITRE Adversarial Tactics, Techniques, and Common Knowledge. J. Cybersecur. Priv. 2024, 4, 777–793.
2. Mokalled, H.; Catelli, R.; Casola, V.; Debertol, D.; Meda, E.; Zunino, R. The Applicability of a SIEM Solution: Requirements and Evaluation. In Proceedings of the 28th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, Naples, Italy, 12–14 June 2019.
3. Sheeraz, M.; Paracha, M.A.; Haque, M.U.; Durad, M.H.; Mohsin, S.M.; Band, S.S.; Mosavi, A. Effective security monitoring using efficient SIEM architecture. Hum.-Centric Comput. Inf. Sci. 2023, 13, 1–18.
4. Khayat, M.; Barka, E.; Serhani, M.A.; Sallabi, F.; Shuaib, K.; Khater, H.M. Empowering Security Operation Center with Artificial Intelligence and Machine Learning–A Systematic Literature Review. IEEE Access 2025, 13, 19162–19197.
5. Hughes, K.; McLaughlin, K.; Sezer, S. Dynamic countermeasure knowledge for intrusion response systems. In Proceedings of the 2020 31st Irish Signals and Systems Conference (ISSC), Letterkenny, Ireland, 11–12 June 2020; pp. 1–6.
6. Coscia, A.; Dentamaro, V.; Galantucci, S.; Maci, A.; Pirlo, G. Automatic decision tree-based NIDPS ruleset generation for DoS/DDoS attacks. J. Inf. Secur. Appl. 2024, 82, 103736.
7. Kinyua, J.; Awuah, L. AI/ML in Security Orchestration, Automation and Response: Future Research Directions. Intell. Autom. Soft Comput. 2021, 28, 527–545.
8. Sworna, Z.T.; Islam, C.; Babar, M.A. APIRO: A framework for Automated Security Tools API Recommendation. ACM Trans. Softw. Eng. Methodol. 2023, 32, 1–42.
9. Toyin, O.; Adeola, M.O.; Oguntimilehin, A.; OB, A.; Aweh, O.M.; Obamiyi, S.E.; Akinduyite, C.O.; James, A.A. Intelligent Network Intrusion Detection and Prevention System (NIDPS): A Machine Learning and Network Security. In Proceedings of the 2024 IEEE 5th International Conference on Electro-Computing Technologies for Humanity (NIGERCON), Ado Ekiti, Nigeria, 26–28 November 2024; pp. 1–6.
10. Kurnia, R.; Widyatama, F.; Wibawa, I.M.; Brata, Z.A.; Nelistiani, G.A.; Kim, H. Enhancing Security Operations Center: Wazuh Security Event Response with Retrieval-Augmented-Generation-Driven Copilot. Sensors 2025, 25, 870.
11. Manzoor, J.; Waleed, A.; Jamali, A.F.; Masood, A. Cybersecurity on a budget: Evaluating security and performance of open-source SIEM solutions for SMEs. PLoS ONE 2024, 19, e0301183. [PubMed]
12. Moiz, S.; Majid, A.; Basit, A.; Ebrahim, M.; Abro, A.A.; Naeem, M. Security and threat detection through cloud-based Wazuh deployment. In Proceedings of the 2024 IEEE 1st Karachi Section Humanitarian Technology Conference (KHI-HTC), Tandojam, Pakistan, 8–9 January 2024; pp. 1–5.
13. Vilendeˇci´c, B.; Dejanovi´c, R.; Curi´c, P. The Impact of Human Factors in the Implementation of SIEM Systems. ´J. Electr. Eng. 2017, 5, 196–203.
14. Chamkar, S.A.; Maleh, Y.; Gherabi, N. The Human Factor Capabilities in Security Operation Center (SOC). EDPACS 2022, 66, 1–14.
15. Mughal, A.A. Building and securing the modern security operations center (soc). Int. J. Bus. Intell. Big Data Anal. 2022, 5, 1–15.
16. Önal, V.; Arslan, H.; Görmez, Y. Machine Learning and Event-Based User and Entity Behavior Analysis. In Proceedings of the 2024 32nd Signal Processing and Communications Applications Conference (SIU), Mersin, Turkiye, 15–18 May 2024; pp. 1–4.
17. Karampudi, B.; Phanideep, D.M.; Reddy, V.M.K.; Subhashini, N.; Muthulakshmi, S. Malware Analysis Using Machine Learning. In Intelligent Systems Design and Applications; Abraham, A., Pllana, S., Casalino, G., Ma, K., Bajaj, A., Eds.; Springer Nature: Cham, Switzerland, 2023; pp. 281–290.
18. Silic, M.; Delac, G.; Srbljic, S. Prediction of Atomic Web Services Reliability Based on K-means

Clustering. In ESEC/FSE, Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering, Saint Petersburg, Russia, 18–26 August 2013; ACM: New York, NY, USA, 2013; pp. 70–80.

19. Laaksonen, J.; Oja, E. Classification with Learning K-nearest Neighbors. In Proceedings of the IEEE International Conference on Neural Networks, Washington, DC, USA, 3–6 June 1996; Volume 3, pp. 1480–1483.

20. Breiman, L. Random forests. In Machine Learning; Springer: Berlin/Heidelberg, Germany, 2001; Volume 45, pp. 1–33.

21. Rish, I. An Empirical Study of The Naive Bayes Classifier. In IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence; Washington, DC, USA, 2001; Volume 3, pp. 41–46.

22. Liu, F.T.; Ting, K.M.; Zhou, Z.-H. Isolation Forest. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; pp. 413–422.

23. Schubert, E.; Sander, J.; Ester, M.; Kriegel, H.P.; Xu, X. DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. ACM Trans. Database Syst. (TODS) 2017, 42, 1–21.