International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

Analysis of Cyber Attack Patterns on Elastic Logs Using Cybersecurity Framework 2.0 with Supervised and Unsupervised Learning Methods

Rikko Nugroho[1], Widya Silfianti[2]
[1,2]Business Information System, Gunadarma University, Depok, Indonesia, Indonesia-16424

Abstract**—***This study aims to analyze Elastic Security Information and Event Management (SIEM) aligned with the NIST Cybersecurity Framework (CSF) 2.0, particularly the Detect function, and to evaluate the quality of detection results using machine learning. The research was conducted in a virtual laboratory environment consisting of an Elastic SIEM Server, Ubuntu Web Server, Kali Linux, Windows Endpoint, and Windows Local User. Three cyberattack scenarios were evaluated: Brute Force Attack, SQL Injection, and Malware Activity. There are two ways to collect data logs: logs from the Ubuntu Web Server were collected using Filebeat, while logs from the Windows Endpoint were collected using Winlogbeat. The logs were processed through Logstash, stored in Elasticsearch, and visualized using Kibana. The dataset contained 13,150 log records, consisting of 5,733 normal records and 7,417 attack records. Decision Tree and Random Forest were used to classify normal and attack activities, while K-Means was used to analyze data clustering patterns. The results showed that Elastic SIEM successfully generated alerts for all three attack scenarios. Decision Tree achieved an accuracy of 88.4%, attack-class precision of 0.965, recall of 0.825, F1-score of 0.889, and AUC of 0.944. Random Forest achieved an accuracy of 87.8%, attack-class precision of 0.941, recall of 0.835, F1-score of 0.885, and AUC of 0.901. K-Means formed four clusters based on data-characteristic similarities. Overall, Decision Tree provided better classification performance, while Random Forest achieved slightly higher attack recall.*

Keywords**—** *Brute force: Decision Tree: Elastic SIEM: NIST CSF 2.0: RandomForest: SQL Injection.*

I. INTRODUCTION

The development of information technology has led organizations to increasingly rely on digital systems for business processes, services, communications, data storage, and decision-making. This development aligns with the concept of Society 5.0, which positions technology as a means to support human needs, sustainability, and organizational resilience. Society 5.0 builds on the foundations of Industry 4.0, emphasizing a human-centered, resilient, and sustainable approach [1].

The increased use of digital systems is also accompanied by increasing cybersecurity risks. Web applications, servers, endpoints, cloud services, and network devices can become targets of attacks if they are not equipped with adequate security monitoring mechanisms. AwanPintar.id's report shows that during the first half of 2025. there were 133,439,209 cyberattacks or threats detected in Indonesia. These attacks used various techniques, such as brute force, social engineering, open port exploitation, and exploitation of Common Vulnerabilities and Exposures vulnerabilities [2]. In addition, the 2024 Kominfo-CSIRT report recorded anomalous traffic of 2,117 GBps and 27 cyber incidents, with the dominant incident being the injection of malicious files [3]. This data shows that cyber threats remain a problem that requires rapid monitoring and detection capabilities.

In business information systems, cyber threats can affect service availability, data security, and user trust. This research focuses on three types of threats, namely Brute Force Attack, SQL Injection, and Malware Activity. Using SSH brute force and SQL injection scenarios in testing Wazuh-based Security Information and Event Management on web applications [4]. Meanwhile, Wazuh is a log monitoring platform, Shuffle for incident response automation, and YARA for identifying malware based on signatures [5].

One of the challenges in the threat detection process is the large volume of logs generated by servers, applications, operating systems, endpoints, firewalls, and network devices. If logs are still stored separately and analyzed manually, the process of identifying and investigating incidents can take longer. Security Information and Event Management can be used to centrally collect, analyze, correlate, and visualize security logs from various sources. With this mechanism, suspicious activity can be monitored more effectively and in real time [4].

One platform that can be used to build a SIEM is Elastic SIEM, which is based on the Elastic Stack. The Elastic Stack consists of Elasticsearch, Logstash, and Kibana. Logstash is used to receive and process data from various sources, Elasticsearch is used to store and search data, and Kibana is used to display dashboards and data visualizations. The Elastic Stack can be implemented as an open-source SIEM to support threat detection and response to security incidents [6].

To ensure a more targeted SIEM implementation, this study used the National Institute of Standards and Technology Cybersecurity Framework 2.0 as a reference. NIST CSF 2.0 consists of six main functions: Govern, Identify, Protect, Detect, Respond, and Recover [7]. This study focuses on the Detect function because it is related to the ability to find, analyze, and identify activities that have the potential to become security incidents. The Detect function is implemented through collecting logs from servers and endpoints, creating detection rules, and monitoring alerts using Elastic SIEM.

Previous research on SIEM has generally focused on successful log collection, dashboard visualization, and alert

Rikko Nugroho and Widya Silfianti, "Analysis of Cyber Attack Patterns on Elastic Logs Using Cybersecurity Framework 2.0 with Supervised and Unsupervised Learning Methods," *International Research Journal of Advanced Engineering and Science*, Volume 11, Issue 3, pp. 36-41, 2026.

generation. However, the alerts generated may not fully reflect actual attacks. Therefore, quantitative evaluation is needed to determine the ability of the detection data to distinguish between normal and attack activity. Ojoje, Aimufua, Bassey, and Musa [8] used accuracy, precision, recall, F1-score, ROC-AUC, and false positive rate metrics to evaluate the model's effectiveness in detecting network intrusions.

Machine Learning can be used as an approach to test the validity of detection results generated by SIEM. Decision Tree and Random Forest algorithms can be used in anomaly-based intrusion detection in network traffic [9]. Both algorithms can perform classification based on data patterns and characteristics. In addition to classification methods, K-Means can be used to group data based on similar characteristics. Classification and clustering approaches can be used to understand the patterns and distribution of attacks on network security data [10].

In this study, Decision Tree and Random Forest were used to classify Elastic SIEM detection data into normal and attack activity categories. The performance of both algorithms was evaluated using accuracy, precision, recall, F1-score, and Receiver Operating Characteristic/Area Under the Curve. These metrics are necessary to more measurably assess the model's ability to detect anomalies. K-Means was used to identify log data grouping patterns based on characteristic similarities, not to measure classification accuracy [11].

Based on previous research, there is a need to integrate SIEM implementation with Machine Learning-based detection result evaluation. Therefore, this study implemented Elastic SIEM in a virtual laboratory environment by referring to the Detect function in NIST CSF 2.0. Testing was conducted using Brute Force Attack, SQL Injection, and Malware Activity scenarios. The obtained log and alert data were then validated using Decision Tree, Random Forest, and K-Means. This study aims to determine the success of Elastic SIEM implementation in detecting threats and to measure the quality of the detection data quantitatively.

The contribution of this research is to combine the implementation of Elastic SIEM, the implementation of the Detect function in NIST CSF 2.0, and machine learning testing in a single research environment. This approach not only demonstrates the system's ability to generate logs and alerts but also provides measurable evaluation results regarding the classification of normal and attack activity and the clustering patterns of cyberthreat data.

II. Theoretical Basic

A. Cybersecurity and Threats

Cybersecurity aims to protect systems, networks, applications, and data from unauthorized access, damage, or service disruption. Its implementation is necessary to maintain the confidentiality, integrity, and availability of information to support organizational operations [12].

This research focuses on Brute Force Attacks, SQL Injection, and Malware Activity. Brute Force Attacks are carried out through repeated authentication attempts to gain access to a system [13]. SQL Injection is carried out by inserting malicious SQL commands into web application input [14]. Meanwhile, Malware Activity is characterized by suspicious process execution, file changes, or network communications inconsistent with normal activity [15].

B. Security Information and Event Management and Elastic SIEM

Security Information and Event Management is a system used to collect, store, analyze, and correlate security logs from various sources. SIEM supports real-time monitoring, suspicious activity detection, alert generation, and centralized incident analysis [16].

This research used Elastic SIEM based on the Elastic Stack, which consists of Filebeat, Winlogbeat, Logstash, Elasticsearch, and Kibana. To send logs from endpoints and servers, Logstash processes the data, Elasticsearch indexes and stores the logs, and Kibana shows warnings and dashboards. Filebeat and Winlogbeat are utilized [6], [17]. The system was implemented in a virtual laboratory environment to allow controlled security testing without impacting production systems.

C. NIST Cybersecurity Framework 2.0

The NIST Cybersecurity Framework 2.0 consists of the Govern, Identify, Protect, Detect, Respond, and Recover functions [7]. This research focuses on the Detect function because it relates to the process of collecting logs, monitoring activity, creating detection rules, and analyzing alerts.

The Identify function is used to determine assets and log sources, while the Protect function is used to set up basic protection in the test environment. The Detect function is then applied through Elastic SIEM to detect Brute Force Attacks, SQL Injection, and Malware Activity.

D. Machine Learning

Machine Learning is used to study data patterns and perform classification or clustering. This study used Decision Trees, Random Forests, and K-Means to test Elastic SIEM detection data [18].

Decision Trees are a classification algorithm that generates predictions based on conditions in data features. Random Forests combine multiple Decision Trees and determine the final result through majority voting, resulting in a more stable model and reducing the risk of overfitting [19]. Both algorithms are used to classify data into normal and attack categories.

K-Means is an unsupervised learning algorithm that clusters data based on characteristic similarities to centroids [20]. In this study, K-Means was used to examine clustering patterns in log data and was not used to measure classification accuracy.

E. Model and Dataset Evaluation

The classification results of Decision Tree and Random Forest were evaluated using a Confusion Matrix consisting of True Positives, True Negatives, False Positives, and False Negatives [21]. The metrics used included accuracy, precision, recall, F1-score, and ROC/AUC. These metrics were used to measure the model's accuracy and its ability to distinguish between normal activity and attacks [8], [11].

Rikko Nugroho and Widya Silfianti, "Analysis of Cyber Attack Patterns on Elastic Logs Using Cybersecurity Framework 2.0 with Supervised and Unsupervised Learning Methods," *International Research Journal of Advanced Engineering and Science*, Volume 11, Issue 3, pp. 36-41, 2026.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

Before modeling, the dataset underwent cleaning, labeling, categorical data transformation, normalization, and feature selection [22]. The dataset was then divided into 80% training data and 20% testing data for Decision Tree and Random Forest testing [23].

F. Previous Research

Previous research has shown that SIEM can be used to collect logs, display dashboards, and detect brute force attacks, SQL injections, and malware activity [4]–[6]. Other research has shown that Decision Trees and Random Forests can be used for attack classification, while K-Means can be used to identify clustering patterns in security data [9], [10].

The difference in this research lies in combining Elastic SIEM, the Detect function in NIST CSF 2.0, and Machine Learning validation in a single laboratory environment. This approach not only assesses the success of alerts but also quantitatively evaluates the quality of the detection data.

III. Research Methods

A. Research Stages

This research uses an experimental method in a virtual laboratory. The research stages include literature review, laboratory design, Elastic SIEM implementation, NIST CSF 2.0 implementation, attack scenarios, log and alert collection, preprocessing, machine learning testing, evaluation, and conclusion drawing.

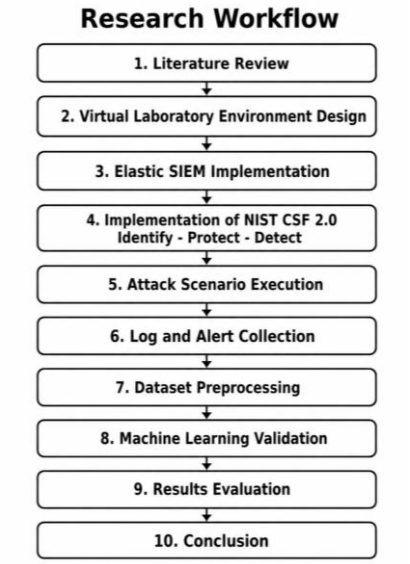

Fig. 1. Research Workflow

B. Virtual Laboratory Environment Design

The research environment uses a 192.168.100.0/24 network with a gateway of 192.168.100.1. Elastic SIEM Server (192.168.100.15) serves as the monitoring center; Ubuntu Web Server (192.168.100.16) serves as the target for Brute Force Attacks and SQL Injection; Kali Linux (192.168.100.17) serves as the attacker; Windows Endpoint (192.168.100.18) serves as the target for Malware Activity; and Windows Local User (192.168.100.2) serves as the source of normal activity.

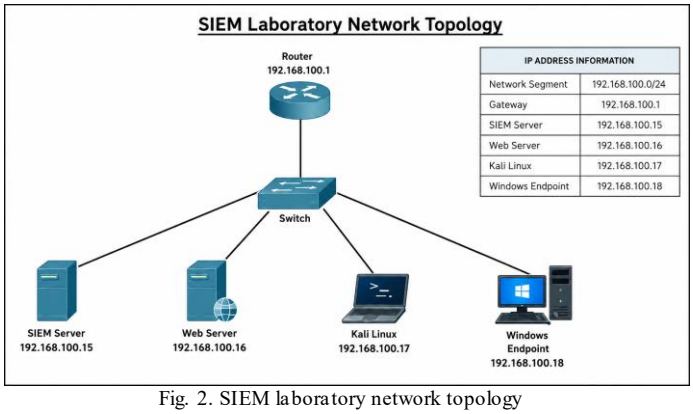

Fig. 2. SIEM laboratory network topology

C. Elastic SIEM Implementations

Elastic SIEM is built using Elasticsearch, Logstash, Kibana, Filebeat, and Winlogbeat. Filebeat collects SSH authentication logs and Apache access logs, while Winlogbeat collects Windows Event Logs and Windows Defender logs. Logs are processed through Logstash, stored in Elasticsearch, and visualized through Kibana.

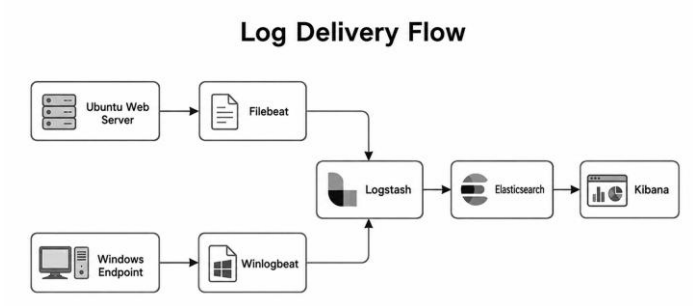

Fig. 3. Log delivery flow

D. Implementation of NIST CSF 2.0

The implementation of NIST CSF 2.0 focuses on Identify, Protect, and Detect. Identify is used to identify assets, services, log sources, and threats. Protect includes authentication, access restrictions, Windows Defender, agent configuration, and the separation of laboratory from production systems. Detect is the primary focus through log monitoring, detection rules, and alert analysis.

E. Attack Scenario Execution

A brute force attack was performed from Kali Linux to an SSH Ubuntu Web Server through repeated authentication. SQL injection was performed on the web application using UNION SELECT, OR 1=1, and SLEEP patterns. Malware activity was performed on a Windows endpoint using simulated process or file activity that generated Windows Event Log and Windows Defender events.

F. Log and Alert Collection

Logs and alerts were collected after all scenarios were executed. Data came from SSH authentication logs and Apache access logs on Ubuntu Web Server, as well as Windows Event Logs and Windows Defender logs on Windows Endpoints. Filebeat and Winlogbeat sent data via Logstash to Elasticsearch, then the logs and alerts were verified using Kibana based on timestamp, agent type, host

Rikko Nugroho and Widya Silfianti, "Analysis of Cyber Attack Patterns on Elastic Logs Using Cybersecurity Framework 2.0 with Supervised and Unsupervised Learning Methods," *International Research Journal of Advanced Engineering and Science*, Volume 11, Issue 3, pp. 36-41, 2026.

name, source IP, event dataset, outcome, process name, attack type, and label. Normal and attack activity data were then exported in CSV format and compiled into a research dataset of 13,150 logs.

G. Dataset Preprocessing

The dataset consists of 5,733 normal data sets and 7,417 attack data sets, including 2,419 Brute Force Attacks, 2,861 SQL Injections, and 2,137 Malware Activities. Preprocessing was performed by checking for empty and duplicate values, assigning normal and attack labels, encoding categorical data, normalizing numeric data, and selecting features. The dataset was then divided into 80% training data and 20% testing data for Decision Tree and Random Forest testing.

H. Machine Learning Validation

Testing was conducted using the KNIME Analytics Platform. Decision Trees and Random Forests were used for normal and attack classification through preprocessing, data splitting, learning, predictor, scorer, and ROC curve stages. K-Means was used for clustering after encoding and normalization. PCA was used to display the clustering results in a scatter plot, with the number of clusters set at four.

I. Results Evaluation

Decision Tree and Random Forest were evaluated using Confusion Matrix, accuracy, precision, recall, F1-score, and ROC/AUC. K-Means results were analyzed based on the number of members, centroids, data distribution, and cluster characteristics. K-Means was not evaluated using classification metrics because it does not use class labels.

J. Conclusion

Conclusions are drawn based on the successful implementation of Elastic SIEM, the detection rule's ability to generate alerts for the three scenarios, the Decision Tree and Random Forest classification results, and the K-Means clustering pattern. This stage is used to answer the problem formulation and determine the best-performing model based on all evaluation metrics.

IV. Results and Discussion

A. Elastic SIEM Implementation and Detection Results

The Elastic SIEM implementation was successfully implemented in a virtual laboratory with a 192.168.100.0/24 network. Elasticsearch, Logstash, Kibana, Filebeat, and Winlogbeat ran as a single log collection and analysis pipeline. Ubuntu Web Server logs were received through the filebeat-* index, while Windows Endpoint logs were received through the winlogbeat-* index. These results demonstrate that the Detect function in NIST CSF 2.0 has been implemented through log monitoring, detection rule creation, and alert analysis.

The alert dashboard shows three rules that successfully generated alerts: Potential Brute Force Attack Detected, SQL Injection Attempt Detected via Apache Logs, and Malware Activity Detected by Windows Defender. Brute Force Attacks are identified through repeated authentication failures, SQL Injection through suspicious request patterns in the Apache access log, and Malware Activity through process events, files, and Windows Defender.

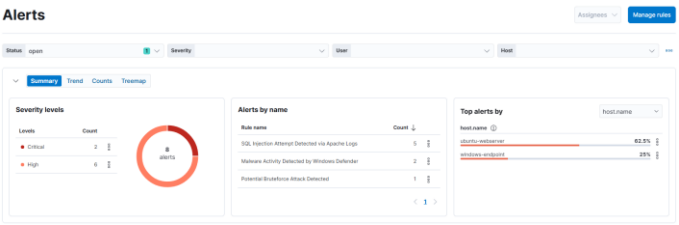

Fig. 4. Dashboard alert Elastic SIEM

TABLE 1. Summary of detection results

Attack	Source Log	Results
Brute Force	SSH authentication log	warning appears
SQL Injection	Apache access log	warning appears
Malware Activity	Windows Event Log/Defender	warning appears

B. Dataset Composition

The research dataset consists of 13,150 logs, consisting of 5,733 normal logs and 7,417 attack logs. The attack logs consist of 2,419 brute force attacks, 2,861 SQL injections, and 2,137 malware activities. The normal and attack logs are used for Decision Tree and Random Forest classification.

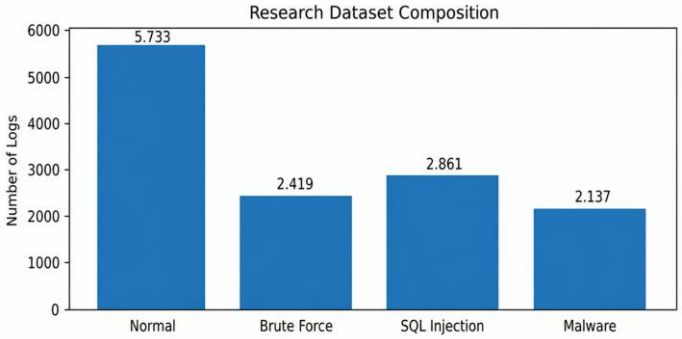

Fig. 5. Research Dataset Composition

C. Decision Tree and Random Forest Classification Results

Testing was conducted using the KNIME Analytics Platform after the dataset was subjected to Column Filtering, Missing Value Filtering, One-to-Many Filtering, Normalization, and an 80:20 data split. The testing flow connected the learner and predictor of each model with a Scorer and ROC Curve.

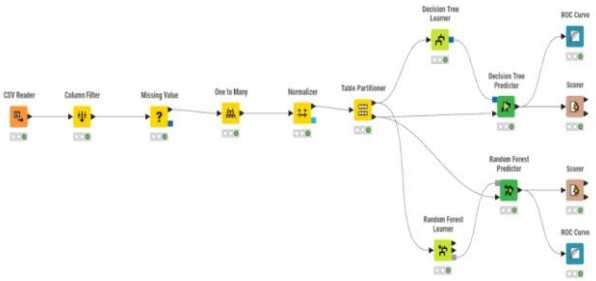

Fig. 6. Decision Tree and Random Forest testing workflow

Decision Tree correctly classified 1,223 attack data sets and 1,102 normal data sets. A total of 260 attack data sets were predicted as normal and 45 normal data sets were

Rikko Nugroho and Widya Silfianti, "Analysis of Cyber Attack Patterns on Elastic Logs Using Cybersecurity Framework 2.0 with Supervised and Unsupervised Learning Methods," *International Research Journal of Advanced Engineering and Science*, Volume 11, Issue 3, pp. 36-41, 2026.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

predicted as attack. Random Forest correctly classified 1,238 attack data sets and 1,070 normal data sets, with 245 false negatives and 77 false positives.

TABLE 2. Confusion matrix in the model

Model	TP	FN	FP	TN
Decision Tree	1.223	260	45	1.102
Random Forest	1.238	245	77	1.070

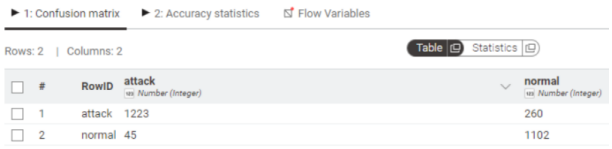

Fig. 7. Confusion matrix Decision Tree

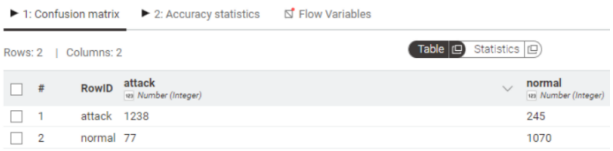

Fig. 8. Confusion matrix Random Forest

D. Model Performance Comparison

Testing was conducted using the KNIME Analytics Platform after the dataset was subjected to Column Filter, Missing Value Filter, One-to-Many Filter, Normalizer, and 80:20 data splitting.

TABLE 3. Comparison of evaluation results

Model	Accuracy	Precision Attack	Recall Attack	F1 Attack	AUC
Decision Tree	0,884	0,965	0,825	0,889	0,944
Random Forest	0,878	0,941	0,835	0,885	0,901

Decision Tree achieved 88.4% accuracy and an AUC of 0.944, while Random Forest achieved 87.8% accuracy and an AUC of 0.901. Decision Tree had higher precision and F1-score attack, resulting in more accurate attack predictions. Random Forest had slightly higher recall attack, indicating a slightly better ability to find attack data. Overall, Decision Tree performed better on this dataset.

Fig. 9. ROC Curve Decision Tree

Fig. 10. ROC Curve Random Forest

E. K-Means Clustering Results

K-Means was used as a supporting analysis to examine the similarity of log data characteristics. The testing process included preprocessing, K-Means Learner, K-Means Predictor, PCA, and Scatter Plot. PCA was used to reduce dimensionality so that cluster patterns could be visualized in two dimensions.

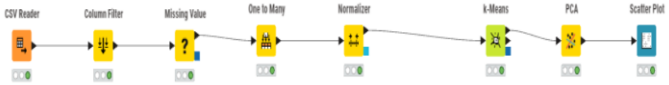

Fig. 11. K-Means testing workflow

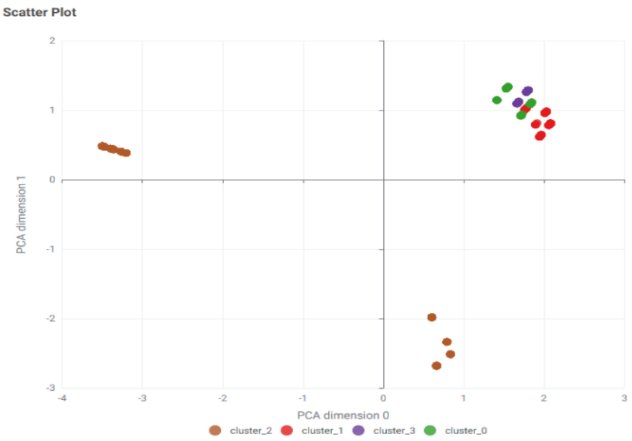

Fig. 12. K-Means results using PCA and Scatter Plot

The clustering results formed four clusters: cluster_0, cluster_1, cluster_2, and cluster_3. Cluster_2 appears relatively separate, while the other clusters are located in several adjacent areas. This indicates that some data have different characteristics, but others have similar patterns. Clusters cannot be directly considered normal labels or attack types because K-Means works without class labels.

F. Discussion

The results demonstrate that Elastic SIEM can receive logs from servers and endpoints and generate alerts for Brute Force Attacks, SQL Injection, and Malware Activity. Machine Learning validation showed that the detected data could be classified with accuracy above 85%. Decision Tree emerged as the best model based on accuracy and AUC, while Random Forest had a slight advantage in recall attacks. K-Means complemented the analysis by revealing patterns of closeness in data characteristics without replacing classification evaluation.

V. Conclusion

Based on the results of the implementation, testing, and evaluation conducted in this study, the following conclusions can be drawn.

1. Elastic SIEM based on the NIST Cybersecurity Framework 2.0, specifically the Detect function, was successfully implemented in a virtual laboratory environment as a central monitoring and analysis center for logs from Ubuntu Web Server and Windows Endpoints.
2. Elastic SIEM successfully detected Brute Force Attacks,

Rikko Nugroho and Widya Silfianti, "Analysis of Cyber Attack Patterns on Elastic Logs Using Cybersecurity Framework 2.0 with Supervised and Unsupervised Learning Methods," *International Research Journal of Advanced Engineering and Science*, Volume 11, Issue 3, pp. 36-41, 2026.

SQL Injection, and Malware Activity based on the log characteristics of each scenario, as indicated by the appearance of alerts in the system.

3. Decision Tree and Random Forest were able to classify the detected data into Normal and Attack categories. Decision Tree achieved 88.4% accuracy, while Random Forest achieved 87.8% accuracy.
4. K-Means successfully grouped the log data into four clusters based on similar characteristics. However, the clusters cannot be directly considered normal classes or attack types because the clustering process does not use labels.
5. Decision Tree provides better overall performance with a precision attack of 0.965, a recall of 0.825, an F1-score of 0.889, and an AUC of 0.944. Random Forest obtains a precision attack of 0.941, a recall of 0.835, an F1-score of 0.885, and an AUC of 0.901. Random Forest is slightly better on the recall attack, while Decision Tree is superior on the other metrics.

REFERENCES

[1] D. Mourtzis, J. Angelopoulos, and N. Panopoulos, "A literature review of the challenges and opportunities of the transition from Industry 4.0 to Society 5.0," Energies, vol. 15, no. 17, Art. no. 6276, 2022, doi: 10.3390/en15176276.

[2] AwanPintar.id, Laporan Ancaman Digital di Indonesia Semester 1 Tahun 2025. PT Prosperita Sistem Indonesia, 2025. [Online]. Available: https://www.awanpintar.id/.

[3] Kominfo-CSIRT, Laporan Tahunan Kominfo-CSIRT 2024. Kementerian Komunikasi dan Digital Republik Indonesia, 2024. [Online]. Available: https://csirt.komdigi.go.id/.

[4] N. Hidayasari, Mansur, Kasmawi, and Z. Efendi, "Implementasi prototipe SIEM berbasis Wazuh pada website dengan pengujian FIM dan threat hunting," JITSI: Jurnal Ilmiah Teknologi Sistem Informasi, vol. 6, no. 4, pp. 372–382, 2025, doi: 10.62527/jitsi.6.4.523.

[5] M. R. Na'im and Y. Kurniawan, "Implementasi dan evaluasi sistem keamanan siber berbasis Wazuh, Shuffle, dan YARA di pusat data Diskominfo Kota Tangerang dengan metode PPDIOO," Router: Jurnal Teknik Informatika dan Terapan, vol. 3, no. 4, pp. 120–134, 2025, doi: 10.62951/router.v3i4.752.

[6] S. Regi and G. Kaur, "Improving threat detection with Elastic Stack based Security Information and Event Management," International Journal for Research in Applied Science and Engineering Technology, vol. 12, no. 4, pp. 2428–2433, 2024, doi: 10.22214/ijraset.2024.60266.

[7] National Institute of Standards and Technology, The NIST Cybersecurity Framework (CSF) 2.0, NIST Cybersecurity White Paper 29, 2024, doi: 10.6028/NIST.CSWP.29.

[8] O. A. Ojoje, G. I. O. Aimufua, S. I. Bassey, and U. Musa, "Performance evaluation and benchmarking of machine learning algorithms for network intrusion detection: An ensemble approach," SSR Journal of Engineering and Technology, vol. 2, no. 5, pp. 21–34, 2025, doi: 10.5281/zenodo.17399805.

[9] A. K. Sah and K. Venkatesh, "Anomaly-based intrusion detection in network traffic using machine learning: A comparative study of Decision Trees and Random Forests," in *2024 2nd International Conference on Networking and Communications (ICNWC)*, 2024, doi: 10.1109/ICNWC60771.2024.10537451.

[10] C. A. Yoga, A. J. Rodrigues, and S. O. Abeka, "Hybrid machine learning approach for attack classification and clustering in network security," International Journal of Computer Applications, vol. 185, no. 31, pp. 45–51, 2023. [Online]. Available: https://www.ijcaonline.org/archives/volume185/number31/yoga-2023-ijca-923076.pdf

[11] M. S. Alsayfi, "Interpretable machine learning framework for anomaly detection in intrusion detection systems using XAI techniques," International Journal of Applied Mathematics, vol. 38, no. 5s, pp. 1191–1213, 2025, doi: 10.12732/ijam.v38i5s.667.

[12] P. A. Khairunnisa, N. Annisa, Yukandri, and J. Parhusip, "Perancangan sistem keamanan jaringan berbasis cybersecurity untuk mitigasi ancaman siber pada infrastruktur TI: Studi kasus di Indonesia," Teknik: Jurnal Ilmu Teknik dan Informatika, vol. 4, no. 2, pp. 9–16, 2024, doi: 10.51903/teknik.v3i1.570.

[13] F. I. Farrel, I. Mardianto, and A. S. Qamar, "Implementation of Security Information and Event Management Wazuh with active response and Telegram notification for mitigating brute force attacks on the GT-I2TI USAKTI information system," INTELMATICS, vol. 4, no. 1, pp. 1–7, 2024, doi: 10.25105/itm.v4i1.18529.

[14] V. Abdullayev and A. S. Chauhan, "SQL injection attack: Quick view," Mesopotamian Journal of Cybersecurity, vol. 2023, pp. 30–34, 2023, doi: 10.58496/MJCS/2023/006.

[15] N. H. Putri and D. I. Mulyana, "Deteksi dan prioritas mitigasi malware menggunakan Random Forest dan metode MCDM," Jurnal Informatika dan Teknik Elektro Terapan, vol. 13, no. 3S1, pp. 632–638, 2025, doi: 10.23960/jitet.v13i3S1.7762.

[16] T. I. Alatise and O. E. Nottidge, "Threat detection and response with SIEM system," International Journal of Communication and Information Technology, vol. 5, no. 1, pp. 36–38, 2024, doi: 10.33545/2707661X.2024.v5.i1a.78.

[17] R. Nezhmetdinov, S. Kotyrova, R. Nezhmetdinova, and I. Kovalev, "Automating log analysis for industrial equipment maintenance using Elastic Stack," E3S Web of Conferences, vol. 389, Art. no. 01064, 2023, doi: 10.1051/e3sconf/202338901064.

[18] I. H. Sarker, "CyberLearning: Effectiveness analysis of machine learning security modeling to detect cyber-anomalies and multi-attacks," arXiv preprint arXiv:2104.08080, 2021, doi: 10.48550/arXiv.2104.08080.

[19] W. H. Elashmawi, A. Sheta, and A. Al-Qerem, "Intelligent intrusion detection system using RF, SVM, and DT: A comparison-based KDD data set," Journal of Computer Science, vol. 21, no. 8, pp. 1749–1759, 2025, doi: 10.3844/jcssp.2025.1749.1759.

[20] J. Liu, W. Yinchai, T. C. Siong, X. Li, L. Zhao, and F. Wei, "A hybrid interpretable deep structure based on adaptive neuro-fuzzy inference system, decision tree, and K-means for intrusion detection," Scientific Reports, vol. 12, Art. no. 20770, 2022, doi: 10.1038/s41598-022-23765-x.

[21] L. I. Uzlah, R. A. Saputra, and Isnawaty, "Deteksi serangan siber pada jaringan komputer menggunakan metode Random Forest," JATI: Jurnal Mahasiswa Teknik Informatika, vol. 8, no. 3, pp. 2787–2793, 2024.

[22] J. Li, M. S. Othman, H. Chen, and L. M. Yusuf, "Optimizing IoT intrusion detection system: Feature selection versus feature extraction in machine learning," Journal of Big Data, vol. 11, Art. no. 36, 2024, doi: 10.1186/s40537-024-00892-y.

[23] N. S. M. A. Sitanggang, O. Pribadi, and Herman, "Implementation of the Decision Tree method for detecting attacks in networks," Journal of Artificial Intelligence and Engineering Applications, vol. 5, no. 1, pp. 277–283, 2025.