Vol. 7, No. 4, December 2025

Journal of Information Systems and Informatics

Published By Asosiasi Doktor Sistem Informasi Indonesia

Optimized K-Means Clustering for Web Server Anomaly Detection Using Elbow Method and Security-Rule Enhancements

Rahmawan Bagus Trianto[1], Muhammad Abdul Muin[2], Cahya Vikasari[3]

[1,2,3]Informatics Engineering, Department of Computer and Business, Politeknik Negeri Cilacap, Cilacap, Indonesia

Email: rahmawanbagustrianto@pnc.ac.id[1], abdulmuin@pnc.ac.id[2], cahyavikasari@pnc.ac.id[3]

Received: Oct 23, 2025

Revised: Nov 9, 2025

Accepted: Dec 7, 2025

Published: Dec 10, 2025

Corresponding Author:

Author Name*:

Rahmawan Bagus Trianto

Email*:

rahmawanbagustrianto@pnc.ac.id

DOI:

10.63158/journalisi.v7i4.1391

Abstract. Anomaly detection in web server environments is essential for identifying early indicators of cyberattacks that arise from abnormal request behaviors. Traditional signature-based mechanisms often fail to detect emerging or obfuscated threats, requiring more adaptive analytical approaches. This study proposes an optimized anomaly detection model using K-Means clustering enhanced with engineered security-rule features and the Elbow Method. Two datasets were used: a small dataset of 3,399 log entries from one VPS and a large dataset of 223,554 entries collected from three VPS nodes, all sourced from local production servers of the Department of Computer and Business, Politeknik Negeri Cilacap. The preprocessing pipeline includes timestamp normalization, removal of non-informative static resources, numerical feature scaling, and TF-IDF encoding of URL paths. Domain-driven security features entropy scores, encoded-payload indicators, abnormal status-code ratios, and request-rate deviations were integrated to improve anomaly separability. Experiments across five model configurations show that combining larger datasets with rule-based features significantly enhances clustering performance, achieving a Silhouette Score of 0.9136 and a Davies–Bouldin Index of 0.4712. The results validate the effectiveness of incorporating security-rule engineering with unsupervised learning to support early-warning threat detection in web server environments.

Keywords: anomaly detection, web server logs, K-Means, Elbow Method, security rules

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

1. INTRODUCTION

Web servers are critical infrastructures for delivering digital services and web-based applications, making them attractive targets for cyberattacks that exploit weaknesses in HTTP requests, server configurations, and application logic [1]–[5]. These malicious activities produce identifiable behavioral patterns in access logs, yet detecting such anomalies remains challenging due to the high volume and unstructured nature of log data, the rapid evolution of attack techniques, and the limited availability of labeled datasets in real operational environments [6], [7].

Unsupervised learning has therefore gained attention, with K-Means frequently adopted for its ability to discover natural groupings in high-dimensional data [8]–[12]. However, its performance depends on selecting an appropriate number of clusters, commonly addressed using the Elbow Method [8]. Moreover, meaningful anomaly detection requires more than basic log attributes; sophisticated threats often demand richer feature engineering that captures semantic, statistical, and behavioral cues [13]. Security-oriented features such as entropy measurements, encoded-payload indicators, uncommon HTTP methods, and abnormal status-code ratios offer additional discriminative power that conventional attributes cannot provide [14].

Prior studies on identifying web-based attacks have explored numerous techniques, including graph learning [1], [3], abstract syntax tree–based analysis [2], deep learning architectures [2], [7], [14]–[17], and OSINT-based methodologies [4]. Although valuable, many rely on labeled datasets and seldom integrate domain-specific security heuristics into the feature space. As a result, their effectiveness in identifying subtle or emerging attack behaviors remains limited.

Only a small number of studies have attempted to unify clustering techniques with engineered security rules, and even fewer have combined semantic URL patterns, statistical characteristics, and IDS/IPS-inspired indicators within a single model. This gap is particularly evident when dealing with large-scale institutional logs that exhibit complex and heterogeneous traffic patterns. Such limitations highlight the need for a more comprehensive modeling approach capable of capturing diverse and evolving threat behaviors.

To address these challenges, this study proposes an optimized clustering model that integrates enhanced security-rule features with K-Means and the Elbow Method to strengthen anomaly detection in web server logs. By bridging IDS/IPS heuristics with unsupervised learning, the model aims to improve anomaly separability and provide deeper insights into malicious activity patterns. The research objectives include: (1) constructing a security-rule–driven feature engineering pipeline; (2) determining the optimal clustering structure; (3) evaluating performance using established metrics such as Silhouette Score, Davies–Bouldin Index (DBI), and Within-Cluster Sum of Squares (WCSS); and (4) characterizing the anomaly groups found in real-world institutional traffic to support operational cybersecurity.

2. METHODS

The methodological framework of this research follows a structured pipeline, as illustrated in Figure 1, which outlines the sequential stages from initial literature review to final evaluation. The flowchart provides a visual representation of the interdependent processes, beginning with conceptual grounding and progressing through data preparation, feature engineering, clustering, model optimization, and anomaly interpretation.

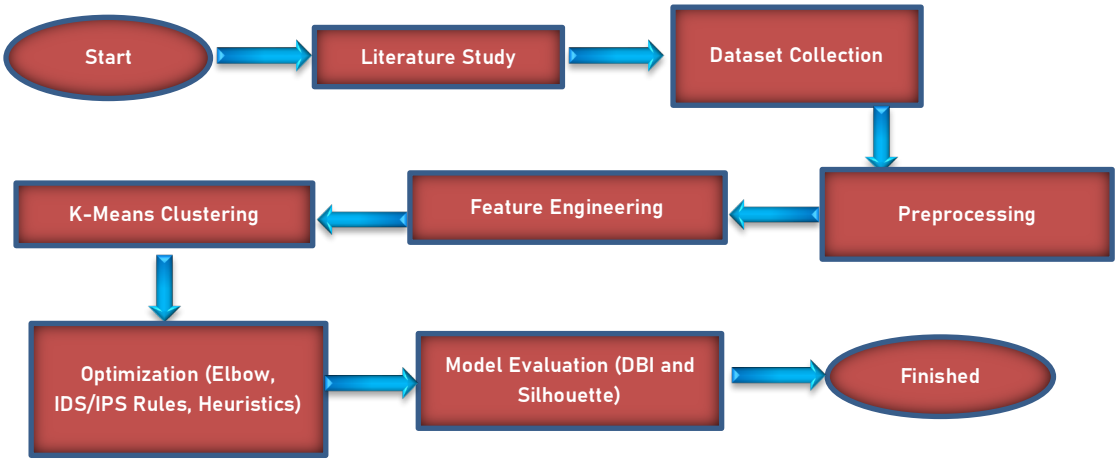

Figure 1. Research Methodological Framework

This flowchart serves as the guiding architecture for the methodological stages described in the following subsections.

3.1. Data Collection

Consistent with the flowchart, the process begins with a literature study, which establishes the theoretical foundation for selecting clustering algorithms, security-rule features, and evaluation metrics. Insights from this stage inform the subsequent dataset collection, where access logs are gathered from production web servers at Department of Computer and Business Politeknik Negeri Cilacap in combined log format. The dataset contains fields such as IP address, request path, HTTP method, status code, response size, user-agent, referrer, and timestamp. Two datasets of different scales were collected to evaluate the impact of data volume on clustering performance and anomaly separation. The small-scale dataset was sourced from a single VPS and consists of 3,399 log entries, representing routine academic and administrative web traffic.

The dataset reflects lightweight operational workloads and provides an initial baseline for model evaluation. The large-scale dataset was gathered from three VPS instances, resulting in a combined total of 223,554 log entries. This dataset captures broader and more diverse traffic patterns typical of multi-service institutional environments, including higher request variance, periodic load spikes, and irregular access behaviors. All log files were collected directly from the local server infrastructure, ensuring data authenticity and consistency. Sensitive information such as IP addresses or session identifiers was anonymized during preprocessing to maintain privacy while preserving structural integrity for analysis.

3.2. Preprocessing

The raw access logs collected from the institutional servers required several preprocessing steps to ensure data quality and analytical consistency before feature engineering and clustering. The preprocessing pipeline consisted of four main stages: cleaning, normalization, transformation, and structured parsing. These steps collectively ensured that the data were free from corruption, semantically consistent, and suitable for subsequent modeling. A summarized overview of each stage is presented in Table 1.

1) First, redundant entries, corrupted lines, and incomplete HTTP records were removed. Log fields containing sensitive information such as IP addresses and session identifiers were anonymized to preserve privacy without altering behavioral patterns.

2) Second, all timestamps were converted into a uniform format and mapped to derived temporal attributes, including hour-of-day and request frequency intervals. Request methods, status codes, and user-agent strings were normalized to standardized categorical forms to reduce noise arising from inconsistent server logging behaviors.
3) Third, URL paths and query components were lowercased, tokenized, and stripped of tracking parameters to ensure consistent semantic representation. Encoding residues such as URL encoding, Base64 segments, or hexadecimal payloads were identified and preserved for subsequent security-rule detection.
4) Finally, each log entry was parsed into a structured tabular format consisting of core HTTP attributes, metadata fields, and preprocessed semantic components. This structured representation served as the foundation for the feature engineering process and enabled efficient computation of clustering metrics and security-rule indicators.

Table 1. Summary of Preprocessing Steps

Stage	Technique	Purpose	Outcome
Data Cleaning	Removal of corrupted/incomplete log lines; anonymization of sensitive fields	Ensure data integrity and protect privacy	Clean and privacy-preserved dataset
Normalization	Standardizing timestamps, request methods, status codes, and user-agent strings	Reduce noise due to inconsistent log formats	Uniform and comparable attribute values
Transformation	Lowercasing, URL tokenization, removal of tracking parameters, detection of encoded payloads	Improve semantic consistency and expose hidden malicious patterns	Normalized path/query components with encoded-segment indicators
Structured Parsing	Extraction of HTTP attributes into tabular format	Enable efficient feature engineering and clustering	Structured dataset suitable for modeling

3.3. Feature Engineering

Following preprocessing, the next stage in the pipeline involves applying feature engineering.

3.3.1. Term Frequency–Inverse Document Frequency (TF-IDF)

Term Frequency–Inverse Document Frequency (TF-IDF) is applied to enhance the representational quality of textual elements within access logs, particularly request paths and query parameters [18]–[20]. These components often carry semantic signals that reveal user intent, navigation patterns, or indications of malicious behavior. TF-IDF transforms such textual data into weighted numerical vectors that reflect term importance across the dataset. TF-IDF serves three primary functions:

1) capturing semantic distinctions between benign and abnormal patterns;
2) preserving rare yet security-critical tokens such as encoded fragments or injection strings; and
3) improving clustering separability by providing high-dimensional semantic richness.

The resulting sparse matrix is later integrated with numerical and rule-based features to form the complete feature set used for K-Means clustering.

3.3.2. Security Rule Feature Engineering

While TF-IDF captures semantic relationships, effective anomaly detection requires features grounded in domain-specific security reasoning. To address this, a set of security-rule features derived from IDS/IPS logic is engineered to highlight behavioral irregularities that may be invisible to text-based representations. These includes:

1) entropy-based indicators for detecting obfuscated payloads;
2) rate-based metrics reflecting abnormal request bursts;
3) status-code deviation ratios representing unsuccessful access attempts;
4) flags for uncommon or suspicious HTTP methods; and
5) detection of encoded payload patterns such as Base64, hexadecimal, or heavy URL-encoding.

Together, these features embed security expertise directly into the feature matrix, improving the model's ability to detect subtle irregularities and distinguish malicious from benign traffic.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

Table 2. Summary of Feature Engineering Components

Category	Feature Type	Description / Technique	Purpose	Expected Outcome
Semantic Features (TF-IDF)	Term Weighting of URL Paths & Queries	Converts textual components into weighted vectors based on frequency and distinctiveness	Capture semantic cues, emphasize rare attack-related tokens	Improved semantic separability and richer feature space for clustering
Security Rule Features	Entropy Score	Measures randomness in URLs and query strings	Detect obfuscated, encoded, or polymorphic payloads	Identification of high-entropy anomaly candidates
	Rate-Based Metrics	Request frequency per IP and time window	Identify scanning, brute-force bursts, or bot-driven traffic	Detection of abnormal traffic spikes
	Status Code Ratios	Ratio of 4xx/5xx errors to 2xx successes	Measure unusual failure patterns from probing attempts	Highlight clients generating disproportionate errors
	Uncommon Method Flags	Binary indicators for OPTIONS, TRACE, PUT, DELETE	Detect enumeration or misuse attempts	Identification of suspicious HTTP operations
	Encoded Payload Detection	Heuristics for Base64, hex sequences, URL-encoded strings	Detect concealed or obfuscated content	Enhanced recognition of hidden malicious payloads

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

3.4. K-Means Clustering

Once the complete feature matrix is constructed combining TF-IDF vectors, normalized numerical attributes, and security-rule–based features the dataset proceeds to the K-Means clustering stage. K-Means is selected due to its computational efficiency, scalability, and suitability for high-dimensional data commonly found in log analytics. The algorithm partitions the dataset into k clusters by minimizing the sum of squared distances between each data point and its assigned centroid. The objective of the algorithm is to partition n data points $x_1, x_2, \ldots, x_n$ into k clusters $C_1, C_2, \ldots, C_k$ such that intra-cluster similarity is maximized and inter-cluster similarity is minimized. The clustering stage of this research employs the K-Means algorithm, selected for its efficiency, simplicity, and suitability for partitioning large-scale numerical datasets such as institutional web server logs. K-Means is a centroid-based clustering method that aims to group observations into k non-overlapping clusters, where each cluster is represented by a centroid corresponding to the mean position of all points within that cluster. The algorithm operates by minimizing the Within-Cluster Sum of Squares (WCSS), which represents the aggregated squared Euclidean distance between data points and their assigned cluster centroids. Formally, the objective function is expressed in Equation 1.

$$\min_{C_1,\ldots,C_k} \sum_{i=1}^{k} \sum_{x \in Ci} \|x - \mu_i\|^2 \quad (1)$$

where Ci denotes the i-th cluster, x represents a feature vector derived from the engineered log attributes, and μi is the centroid of cluster Ci, defined in Equation 2.

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in Ci} x \quad (2)$$

The algorithm proceeds iteratively via two alternating phases:

1) Assignment Step

   Each data point is assigned to the cluster with the nearest centroid based on Euclidean distance. This ensures that points exhibiting similar behavioral patterns such as request frequency, entropy levels, TF-IDF semantics, and security-rule indicators are grouped together.

2) Update Step

Cluster centroids are recalculated as the mean of all points currently assigned to each cluster. This repositioning incrementally improves cluster compactness and separation.

These two steps repeat until convergence, which occurs when cluster assignments no longer change significantly or when the centroid movement falls below a predefined threshold. The algorithm's convergence efficiency makes it well suited for high-volume web server logs where feature dimensionality is increased by TF-IDF vectors and security-rule attributes. On the other literatures [8]–[12], [21], the clustering process involves the following steps:

1) Initialization

   Cluster centroids are initialized using the k-means++ method to enhance convergence stability and reduce sensitivity to random seeds.

2) Assignment Step

   Each log entry is assigned to the closest centroid based on Euclidean distance, which is effective for continuous and normalized numerical representations used in this study.

3) Update Step

   Centroids are recalculated iteratively by computing the mean of all feature vectors within each cluster.

4) Convergence Criteria

   The algorithm repeats the assignment and update steps until minimal centroid shifts occur or a maximum number of iterations is reached.

K-Means capability to reveal natural groupings within unlabeled institutional web logs supports the identification of behavioral clusters that may correspond to normal traffic, suspicious anomalies, or potential attack segments.

3.5. Evaluation Metrics

To assess the quality and validity of the clustering results, this study employs three complementary evaluation metrics: Silhouette Score, Davies–Bouldin Index (DBI), and Within-Cluster Sum of Squares (WCSS). These metrics jointly evaluate cluster compactness, separation, and overall structural coherence, ensuring that the K-Means model produces meaningful behavioral partitions within web server log data.

2.5.1. Silhouette Score

The Silhouette Score measures the degree of confidence in the cluster assignment of each data point by comparing intra-cluster cohesion with inter-cluster separation. For each observation x, the silhouette coefficient $s(x)$ is defined as shown in Equation 3.

$$s(x) = \frac{b(x) - a(x)}{max\{a(x),b(x)\}} \quad (3)$$

where:

a(x) is the average distance between x and all other points within the same cluster (intra-cluster cohesion),

b(x) is the minimum average distance between x and all points in the nearest neighboring cluster (inter-cluster separation).

The resulting silhouette value ranges from −1 to +1:

Values close to +1 indicate well-separated and cohesive clusters.

Values around 0 imply overlapping cluster boundaries.

Values below 0 suggest misclassification.

This metric is particularly important for anomaly detection, as high silhouette scores indicate that anomalous behaviors form distinct and meaningful clusters rather than ambiguous groupings.

2.5.2. Davies–Bouldin Index (DBI)

The Davies–Bouldin Index evaluates clustering quality based on the ratio between intra-cluster dispersion and inter-cluster separation. For k clusters, DBI is defined in Equation 4.

$$DBI = \frac{1}{k} \sum_{i=1}^{k} \underset{j \neq i}{max} \left(\frac{S_i + S_j}{M_{ij}}\right) \quad (4)$$

where:

Si represents the average distance of all points in cluster Ci from its centroid (intra-cluster dispersion),

Mij is the distance between the centroids of clusters Ci and Cj (inter-cluster separation).

Lower DBI values indicate superior cluster structure characterized by compact clusters and large inter-cluster separation. Given the high-dimensional feature space (TF-IDF + security-rule features), DBI provides an essential quantitative measure to ensure that the clustering solution does not suffer from noisy dispersion or centroid overlap.

2.5.3. Within-Cluster Sum of Squares (WCSS)

The Within-Cluster Sum of Squares measures the total variance within clusters and serves as the primary criterion for determining the optimal number of clusters using the Elbow Method. Formally, it is expressed as K-Means point above. WCSS decreases as k increases, but the rate of decline slows after a certain point. The "elbow" in the WCSS curve marks the value of k beyond which additional clusters yield diminishing improvements. This study uses this curve to identify the most appropriate cluster number that balances model complexity and interpretability.

3. RESULTS AND DISCUSSION

This section presents the clustering outcomes from five experimental models designed to identify anomalous patterns including web shell activity, scanning attempts, brute-force behaviors, and payload anomalies in web server logs. Each model varies by dataset size and feature composition, while evaluation is conducted using Silhouette Score, Davies–Bouldin Index (DBI), and Within-Cluster Sum of Squares (WCSS).

3.1. Model 1 - Small Dataset, Without Security Rules

The first experiment, which utilized a small dataset and excluded all IPS/IDS-derived rule features, resulted in an optimal cluster configuration of three groups. Based on the elbow curve on Figure 2, the reduction in WCSS begins to stabilize at k=3, indicating that additional clusters would not yield significant gains in compactness. The silhouette visualization on Figure 3. Silhouette Plot for Model 1 further illustrates moderate cohesion among data points, with one cluster appearing more dispersed and thus likely representing anomalous traffic.

Overall metric performance characterized by a silhouette coefficient of approximately 0.72 and a DBI close to 0.86 suggests that cluster boundaries remain insufficiently separated. Two clusters predominantly represent normal traffic but are split due to

limited diversity in the dataset, while the remaining cluster corresponds to anomalous behavior. These results confirm that, without security-rule features, the model struggles to form well-defined decision boundaries, and anomaly isolation remains weak.

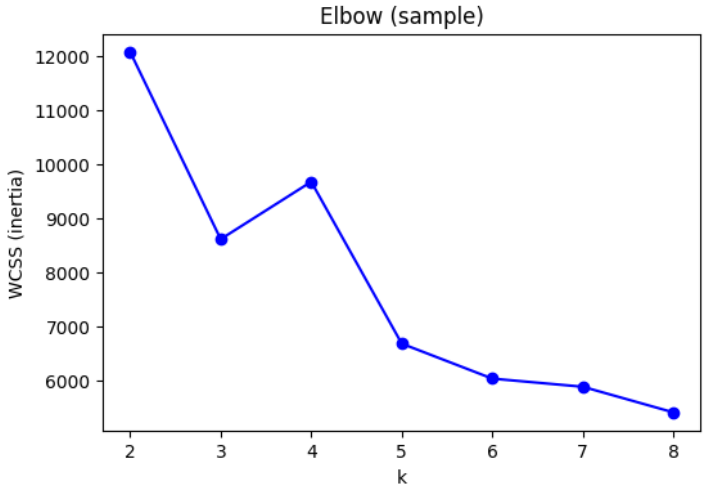

Figure 2. Elbow for Model 1

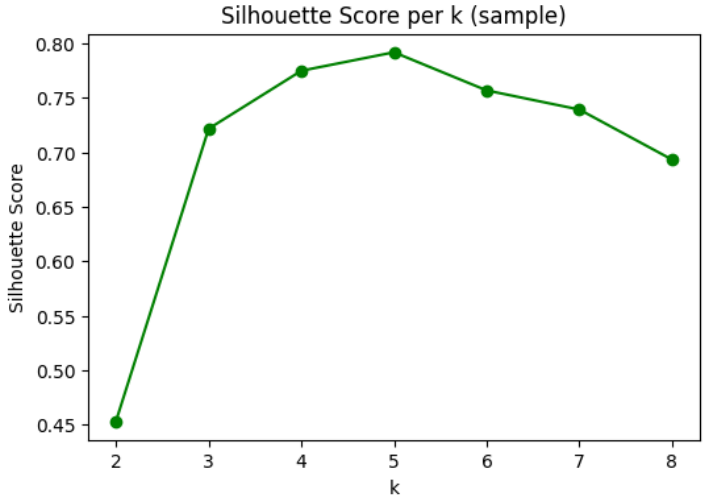

Figure 3. Silhouette for Model 1

3.2. Model 2 - Large Dataset, Without Security Rules

In the second experiment, the use of a substantially larger dataset led to the identification of six optimal clusters, as shown in the corresponding elbow curve on Figure 4 for Elbow Curve for Model 2. The silhouette plot reveals significantly improved compactness among clusters, yielding a high silhouette score of approximately 0.89. However, the DBI value of around 0.78 shows that inter-cluster separation is still not ideal. Silhouette and DBI Score show on Figure 5 and Figure 6.

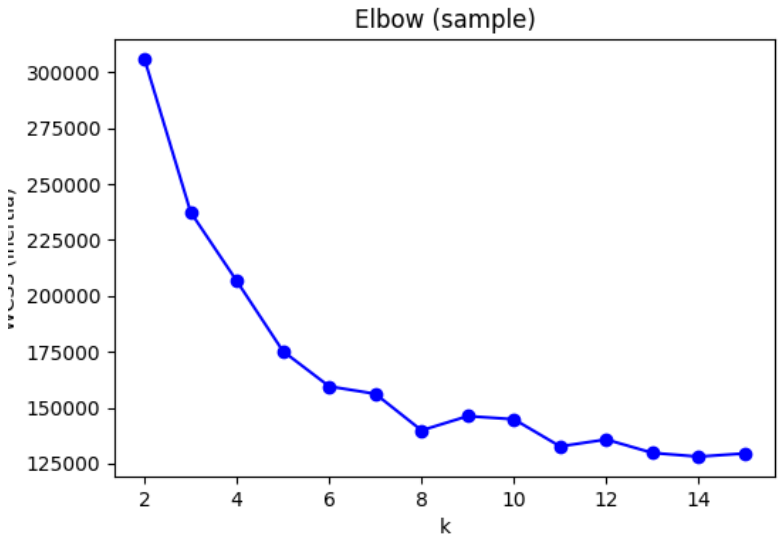

Figure 4. Elbow Curve for Model 2

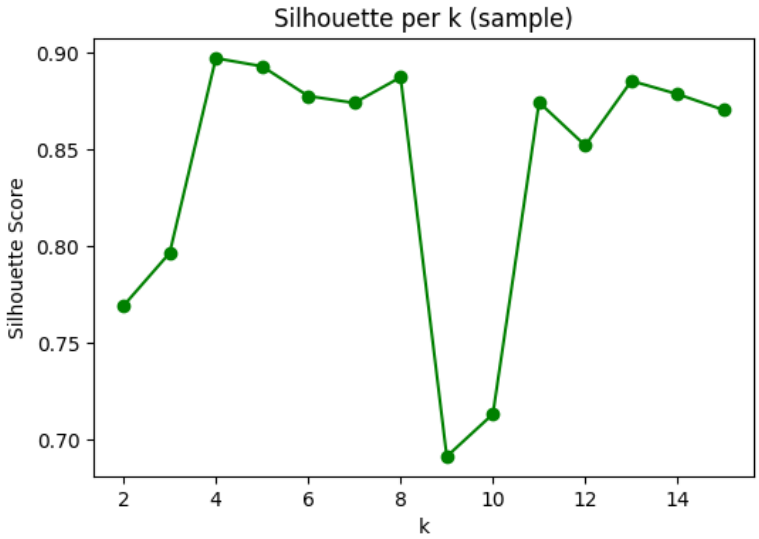

Figure 5. Silhouette Score for Model 2

Although the metrics appear strong, closer analysis indicates that the model's apparent performance is driven by the consistency of normal traffic rather than by effective discrimination of subtle or sophisticated anomalies. This reinforces the notion that basic log features alone are insufficient for detecting fine-grained malicious behavior.

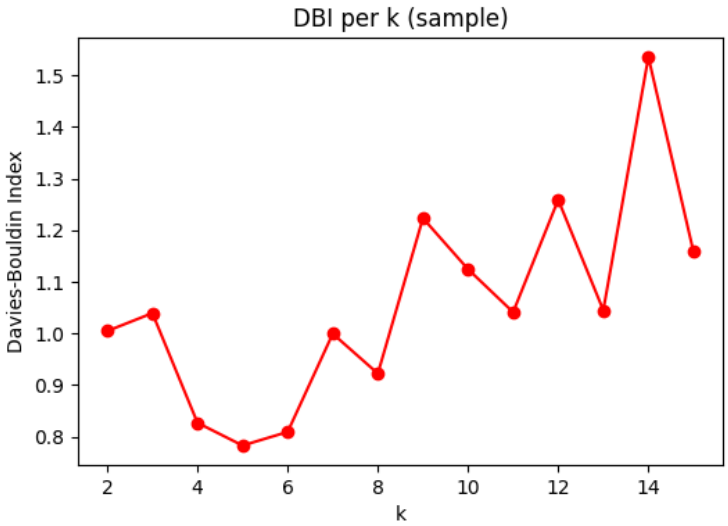

Figure 6. DBI Score for Model 2

3.3. Model 3 - Small Dataset with Security Rules

The third experiment introduced IPS/IDS-derived security-rule features into the small dataset. This modification substantially altered the cluster landscape, producing five distinct clusters. The elbow curve at Figure 7 shows a noticeable inflection at k=5, while the silhouette at Figure 8 displays more compact cluster shapes compared to Models 1 and 2.

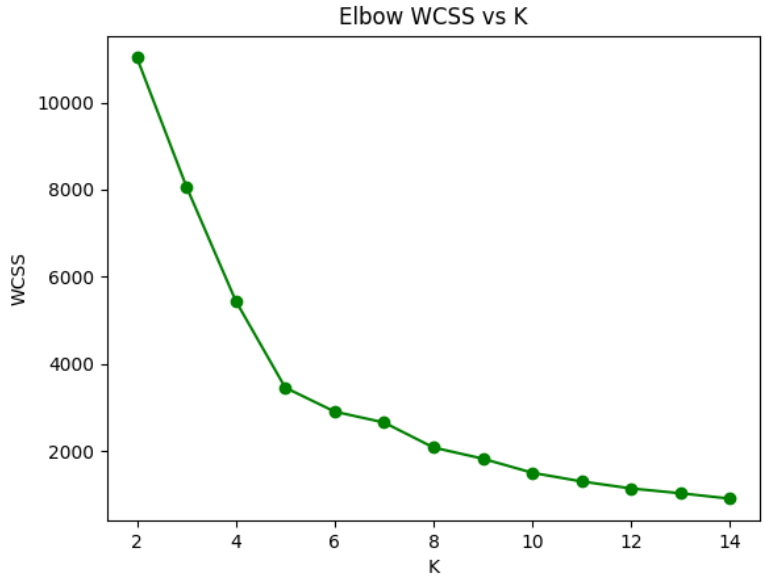

Figure 7. Elbow Curve for Model 3

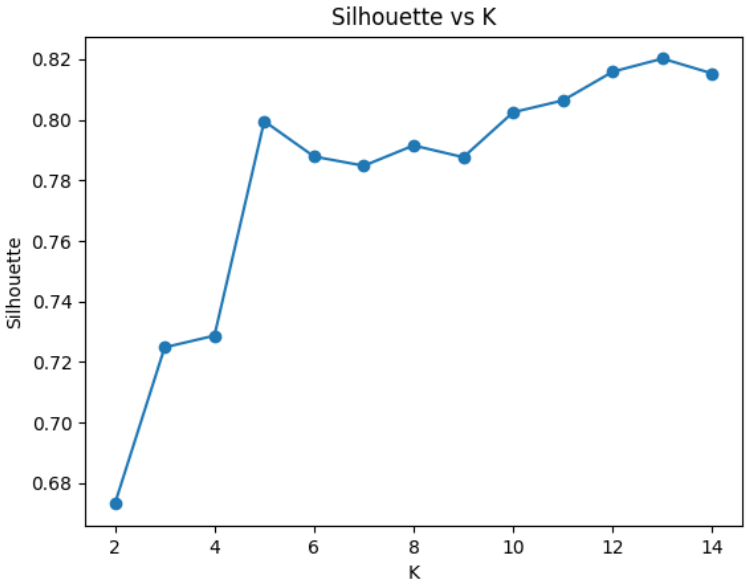

Figure 8. Silhouette Score Curve for Model 3

The inclusion of rule-based features such as entropy indicators, abnormal status ratios, encoded payload detection, and method irregularities enabled the model to separate anomalous behaviors into multiple subgroups instead of combining them into a single class. One cluster is dominated by normal traffic, whereas the remaining clusters represent several anomaly patterns including directory probing, brute-force attempts, web shell access signatures, and suspicious encoded payload activity. The improved silhouette score (around 0.80) and significantly lower DBI value (approximately 0.58) confirm that security-rule features play a critical role in distinguishing malicious behaviors, even when the volume of data is limited.

3.4. Model 4 - Large Dataset with Security Rules

The fourth experiment, which combined a large dataset with a full set of security-rule features, resulted in an optimal configuration of eight clusters. As depicted in Figure 9 Elbow Curve for Model 4, the decrease in WCSS becomes marginal after the eighth cluster. The silhouette Figure 10 Silhouette for Model 4 exhibits strong cohesion among cluster members, and the DBI value of around 0.59 at Figure 11 indicates substantially better inter-cluster separation than models without rule-based features.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

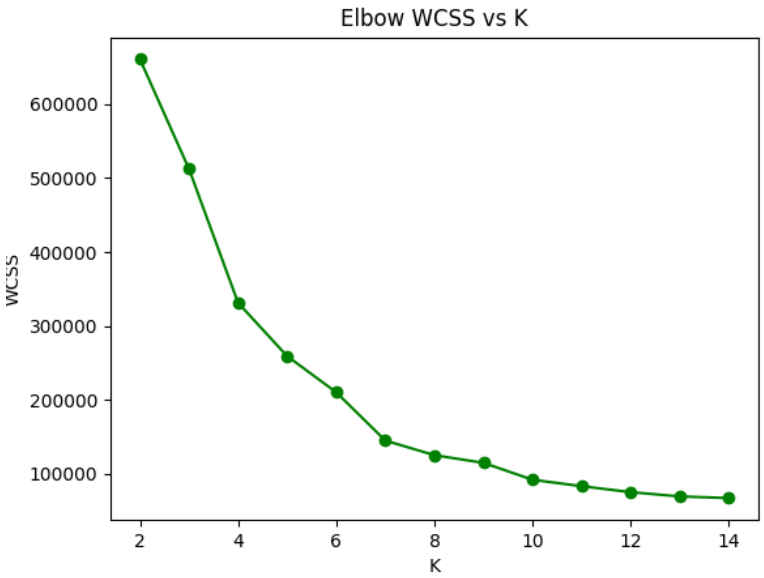

Figure 9. Elbow Curve for Model 4

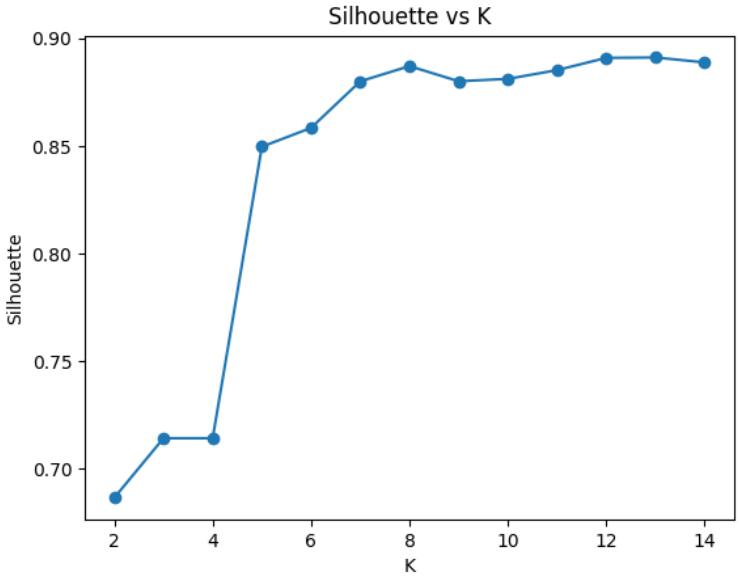

Figure 10. Silhouette for Model 4

Model 4 reveals that normal traffic consolidates into a single, highly homogeneous cluster. Meanwhile, the anomaly clusters split into several semantically meaningful groups such as scanning behavior, brute-force traffic, exploit attempts, suspicious file uploads, encoded payload anomalies, and interactive web shell like activity. This model most accurately reflects the complexity of real-world traffic, demonstrating that both dataset size and rule-derived features are crucial in capturing diverse attack signatures.

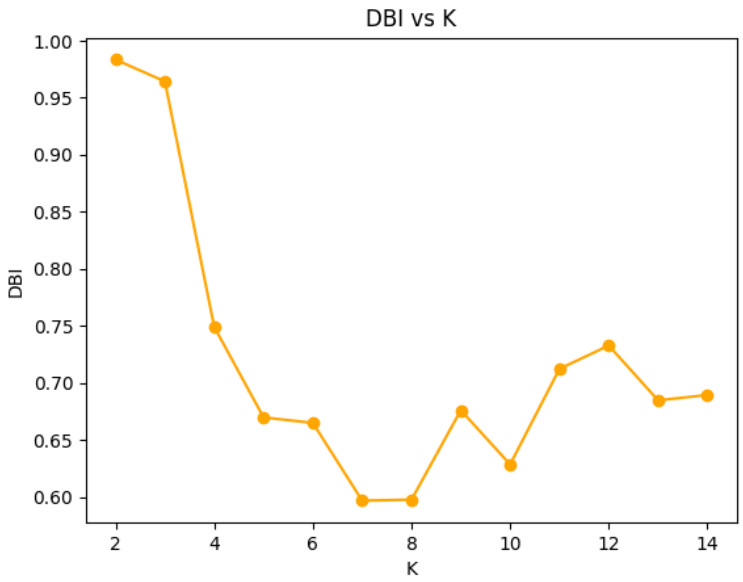

Figure 11. SBI Score for Model 4

3.5. Model 5 - Large Dataset with Reduced Security Rules

In the fifth experiment, two binary rule features (suspicious path indicator and user-agent anomaly indicator) were intentionally removed to evaluate their impact on model performance. Surprisingly, the model exhibited the highest clustering quality among all experiments.

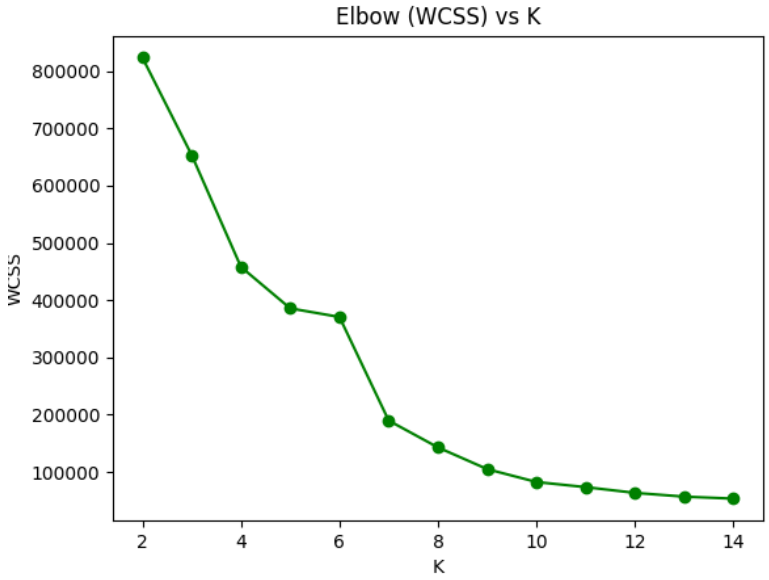

Figure 12. Elbow Curve for Model 5

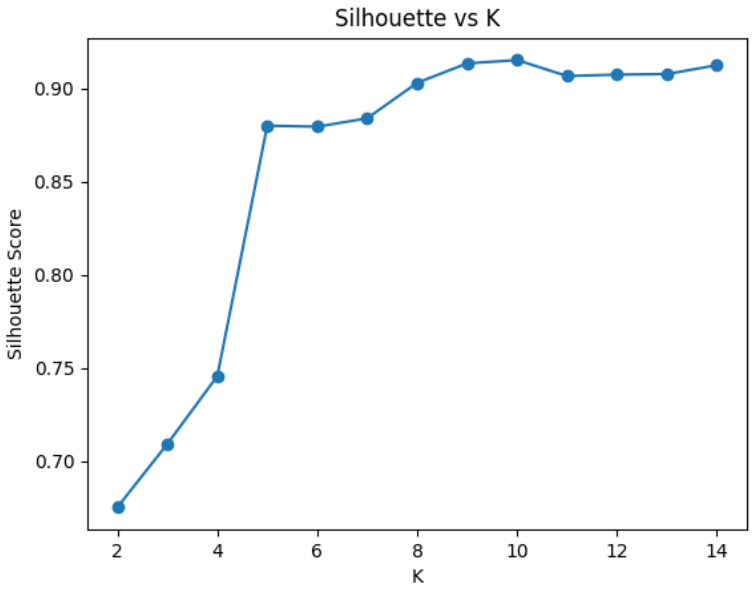

Figure 13. Silhouette Score Curve for Model 5

The elbow curve (Figure 12 Elbow Curve for Model 5) suggests an optimal value of k=9, and the corresponding silhouette plot (Figure 13 Silhouette Plot for Model 5) shows exceptionally clean cluster separation. The silhouette coefficient reaches approximately 0.91, while the DBI (Figure 14 DBI Score for Model 5) falls to around 0.47 the best values across all experiments.

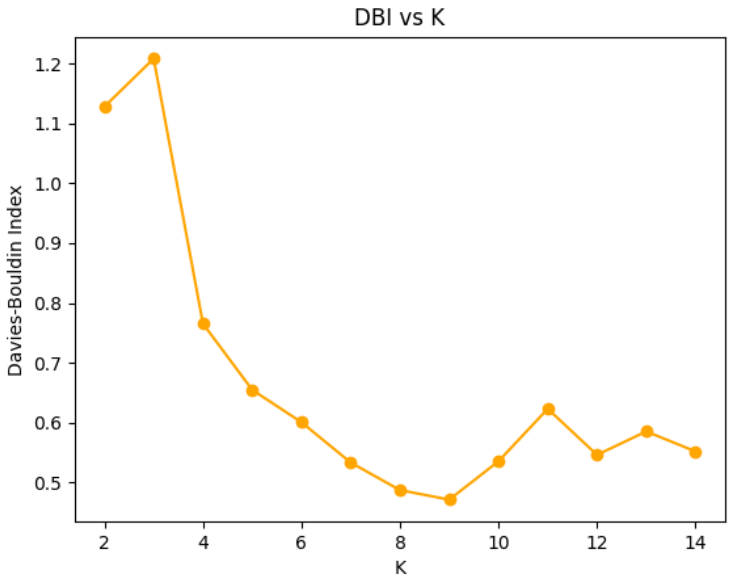

Figure 14. DBI Score Curve for Model 5

This suggests that removing noisy binary features reduces cluster distortion and improves cohesion. Two clusters represent normal traffic segments with distinct behavior profiles, while the remaining clusters correspond to various categories of anomalous activities. These include web shell-related access patterns, scanning activity, stealthy encoded payloads, brute-force sequences, and anomalous upload behaviors. The results indicate that core rule-based features such as entropy, abnormal request-rate deviations, status-code irregularities, and payload encoding signals carry substantial discriminative power even when auxiliary binary rules are removed.

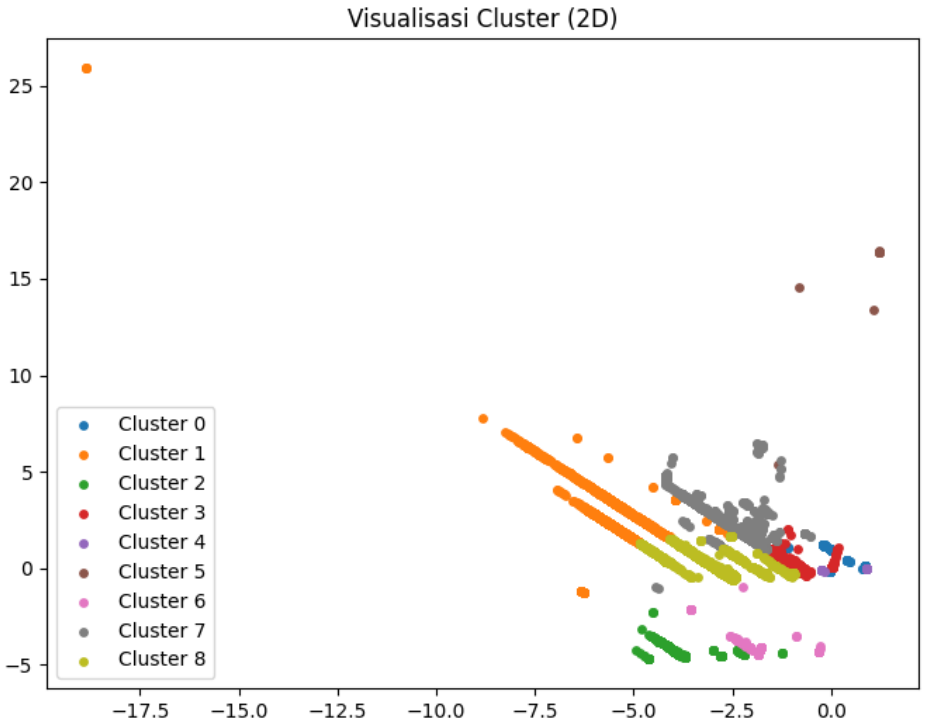

Figure 15. 2D cluster visualization for Model 5

Figure 15 presents the 2D cluster visualization of the best-performing model (model 5) after the removal of noisy binary features. The plot illustrates the distribution of log entries projected into two principal components, enabling a clearer view of the separation between normal and anomalous behavior groups. The visualization shows that the optimized feature set dominated by entropy, request-rate deviation, status-code irregularities, and encoded-payload indicators produces well-formed and interpretable cluster boundaries. Two clusters appear densely grouped with relatively low dispersion,

representing normal traffic segments exhibiting distinct but benign behavioral patterns. Meanwhile, the remaining clusters are more scattered and divergent, reflecting various anomaly categories identified in the dataset.

3.6. Comparative Analysis Summary

The comparison across all models reveals clear trends regarding the impact of dataset size and feature composition. A consolidated summary of model's performance is presented in Table 2, which shows the silhouette coefficient, DBI, and relative interpretability of each model. The table highlights the superior performance of Model 5, which achieves the highest silhouette score and the lowest DBI value, suggesting that the model forms highly cohesive and well-separated clusters. Models incorporating security-rule features consistently outperform those without them, and larger datasets improve cluster stability and allow finer separation of anomaly categories. However, the unexpected performance boost in Model 5 also reveals that not all rule features contribute equally; some rules may introduce noise or overfitting, particularly when their nature is binary or overly static.

Table 2. Summary of model's performance

Model	Silhouette Score	DBI	Remarks
1	0.7203	0.8632	Weakest baseline
2	0.8907	0.7831	Dominated by normal traffic
3	0.7996	0.5807	Security rules appear effective
4	0.8872	0.5979	Rules combined with large-scale data yield optimal performance
5	**0.9136**	**0.4712**	**Best-performing and most stable model**

3.7. Discussion

The experimental results demonstrate that both dataset size and the inclusion of security-rule–based features substantially influence the quality of anomaly detection in web server logs. Models trained without IPS/IDS-derived features struggled to isolate anomalies effectively, even when using larger datasets, because basic log attributes alone do not adequately capture behavioral characteristics associated with malicious activities. Although larger datasets improved cluster stability and produced higher silhouette

scores, much of this improvement stemmed from stronger modeling of normal traffic rather than meaningful separation of anomalous patterns.

In contrast, models incorporating security-rule features exhibited markedly improved separability, confirming that domain-specific behavioral signals such as entropy indicators, abnormal request-rate deviations, status-code anomalies, and encoded payload patterns play a critical role in distinguishing malicious from benign traffic. These enhanced feature sets enabled the models to detect not only broad categories of anomalies but also finer subtypes such as web shell interactions, scanning behaviors, brute-force attempts, and suspicious upload sequences.

The best-performing configuration (Model 5) further reveals that not all rule-based features contribute equally. The removal of certain binary indicators, such as suspicious path flags and user-agent anomaly markers, resulted in clearer cluster structures and reduced noise. This suggests that overly rigid or static rule features may distort the feature space, whereas more expressive, behavior-oriented metrics (e.g., entropy and abnormality ratios) provide more stable discriminative value. Overall, the findings highlight the importance of combining large-scale datasets with carefully engineered security features to achieve robust anomaly detection. The results also indicate that future work should prioritize dynamic and continuous feature engineering over static rule matching, particularly for detecting evolving or previously unseen attack vectors.

The findings of this study show several key differences and improvements compared with prior research:

1) Supervised vs. Unsupervised Approaches
   Previous works using deep learning or AST-based detection [2], [22] rely on labeled datasets, which are rarely available in operational environments. This study overcomes that limitation by using unsupervised clustering capable of detecting anomalies without annotation.
2) Feature Limitations in Prior Studies
   Earlier clustering-based studies typically used only TF-IDF or basic statistical features, offering limited detection sensitivity [1], [8], [14], [16], [23]. Our model integrates entropy, encoded payload indicators, rate deviations, and status-code irregularities, significantly improving anomaly separability.

3) Scope of Detectable Threats

Graph learning or OSINT-based systems [1], [4] focus on specific attack patterns. The proposed model captures a broader threat spectrum, including scans, brute-force attempts, stealthy payloads, and upload anomalies.

4) Operational Applicability

Many existing models require high computational resources or specialized datasets [3], [7], [14], [23]. This research provides a lightweight, server-log–driven method suitable for institutional environments with limited resources.

5) Novelty of Integration

Few prior studies combine clustering with IDS/IPS-inspired features [23]–[25]. This study uniquely bridges domain security heuristics with unsupervised learning, yielding clearer cluster boundaries and more stable results.

4. CONCLUSION

This study demonstrates the effectiveness of integrating clustering techniques with domain-specific security-rule features for detecting anomalous and potentially malicious activities in web server logs. Through five experimental configurations, the results reveal that dataset size and feature composition significantly influence the quality of cluster formation and anomaly separation. Models trained without security-rule enhancements struggled to reliably distinguish between normal and anomalous traffic, even when using large datasets. Conversely, the incorporation of behavioral security rules such as entropy-based indicators, abnormal status-code ratios, encoded payload detection, and request-rate anomalies substantially improved the model's ability to isolate diverse categories of attacks, including web shell access, scanning, brute-force patterns, and suspicious upload behaviors. Among all configurations, Model 5 produced the best overall performance, achieving the highest silhouette score and lowest DBI, indicating strong cohesion and separation. The model's stability, even with reduced binary rule features, highlights the greater importance of expressive behavioral features compared to static or binary indicators. These findings underscore the potential of combining unsupervised learning with engineered security features to build scalable, signature-independent anomaly detection systems capable of identifying both known and novel threat patterns in institutional web environments. Future work may extend this approach by exploring advanced representation techniques such as semantic embeddings, sequence-based

behavioral modeling, and session-level contextual features to further enhance model robustness and generalizability.

Future work may extend this approach by incorporating more advanced representation techniques, including semantic or contextual embeddings, sequence-aware behavioral modeling (e.g., LSTM or attention-based architectures), and session-level aggregation to capture long-range user interactions. Additional exploration of adaptive clustering mechanisms, real-time implementation strategies, and cross-institutional datasets may further enhance model robustness, generalizability, and operational readiness for deployment in production security monitoring environments.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Directorate of Research and Community Service (DPPM), Ministry of Higher Education, Research, and Technology of the Republic of Indonesia, for providing financial support for this research. The authors also extend their appreciation to Politeknik Negeri Cilacap for the continuous institutional support, research facilities, and collaborative environment that made this study possible.

REFERENCES

[1] P. Feng *et al.*, "GlareShell: Graph learning-based PHP webshell detection for web server of industrial internet," *Comput. Networks*, vol. 245, no. April, p. 110406, 2024, doi: 10.1016/j.comnet.2024.110406.

[2] B. Xie, Q. Li, and Y. Wang, "PHP-based malicious webshell detection based on abstract syntax tree simplification and explicit duration recurrent networks," *Comput. Secur.*, vol. 146, no. June, 2024, doi: 10.1016/j.cose.2024.104049.

[3] Y. Xu, Y. Fang, Z. Liu, and Q. Zhang, "PWAGAT: Potential Web attacker detection based on graph attention network," *Neurocomputing*, vol. 557, no. 2019, p. 126725, 2023, doi: 10.1016/j.neucom.2023.126725.

[4] Yusuf Raharja, "Implementasi Metode Osint untuk Mengidentifikasi Serangan Judi Online pada Website," *J. Inform. Polinema*, vol. 10, no. 3, pp. 359–364, 2024, doi: 10.33795/jip.v10i3.4847.

[5] A. Kurniawan, B. S. Abbas, A. Trisetyarso, and S. M. Isa, "Classification of web backdoor malware based on function call execution of static analysis," *ICIC Express Lett.*, vol. 13, no. 6, pp. 445–452, 2019, doi: 10.24507/icicel.13.06.445.

[6] H. Kwon and J. W. Baek, "Text Select-Backdoor: Selective Backdoor Attack for Text Recognition Systems," *IEEE Access*, vol. 12, no. July, pp. 170688–170698, 2024, doi: 10.1109/ACCESS.2024.3436586.

[7] Y. Bai *et al.*, "Backdoor Attack and Defense on Deep Learning: A Survey," *IEEE Trans. Comput. Soc. Syst.*, vol. 12, no. 1, pp. 404–434, 2024, doi: 10.1109/TCSS.2024.3482723.

[8] R. B. Trianto, A. S. Nugroho, and E. Supriyadi, "Klasterisasi Menggunakan Algoritma K-Means dan Elbow pada Opini Masyarakat Tentang Kebijakan Sekolah Luring Tahun 2022," *INOVTEK Polbeng - Seri Inform.*, vol. 8, no. 1, p. 1, 2023, doi: 10.35314/isi.v8i1.2756.

[9] Y. Chen, P. Tan, M. Li, H. Yin, and R. Tang, "K-means clustering method based on nearest-neighbor density matrix for customer electricity behavior analysis," *Int. J. Electr. Power Energy Syst.*, vol. 161, no. July, 2024, doi: 10.1016/j.ijepes.2024.110165.

[10] K. E. Setiawan, A. Kurniawan, A. Chowanda, and D. Suhartono, "Clustering models for hospitals in Jakarta using fuzzy c-means and k-means," *Procedia Comput. Sci.*, vol. 216, no. 2022, pp. 356–363, 2022, doi: 10.1016/j.procs.2022.12.146.

[11] W. A. Prastyabudi, A. N. Alifah, and A. Nurdin, "Segmenting the Higher Education Market: An Analysis of Admissions Data Using K-Means Clustering," *Procedia Comput. Sci.*, vol. 234, no. 2023, pp. 96–105, 2024, doi: 10.1016/j.procs.2024.02.156.

[12] N. Rylko, M. Stawiarz, P. Kurtyka, and V. Mityushev, "Study of anisotropy in polydispersed 2D micro and nano-composites by Elbow and K-Means clustering methods," *Acta Mater.*, vol. 276, no. April, p. 120116, 2024, doi: 10.1016/j.actamat.2024.120116.

[13] X. Sun, X. Liu, C. Deng, H. Chu, G. Wang, and H. Zhao, "An Enhanced Density Peak Clustering Algorithm With Dimensionality Reduction and Relative Density Normalization for High-Dimensional Duplicate Data," *IEEE Access*, vol. 13, no. August, pp. 147242–147264, 2025, doi: 10.1109/ACCESS.2025.3596983.

[14] S. Tahvili, L. Hatvani, M. Felderer, F. G. de Oliveira Neto, W. Afzal, and R. Feldt, "Comparative analysis of text mining and clustering techniques for assessing functional dependency between manual test cases," *Softw. Qual. J.*, vol. 33, no. 2, pp. 1–36, 2025, doi: 10.1007/s11219-025-09722-7.

[15] S. Mostafaei, A. Ahmadi, and J. Shahrabi, "Dealing with data intrinsic difficulties by learning an interPretable Ensemble Rule Learning (PERL) model," *Inf. Sci. (Ny).*, vol.

595, pp. 294–312, 2022, doi: 10.1016/j.ins.2022.02.048.

[16] A. Hannousse and S. Yahiouche, "Handling webshell attacks: A systematic mapping and survey," *Comput. Secur.*, vol. 108, p. 102366, 2021, doi: 10.1016/j.cose.2021.102366.

[17] S. M, S. Anusuya, and L. K. Narayanan, "Enhancing Automatic Speech Recognition Accuracy Using a Gaussian Mixture Model (GMM)," *SSRN Electron. J.*, 2025, doi: 10.2139/ssrn.5089158.

[18] R. Nanda, E. Haerani, S. K. Gusti, and S. Ramadhani, "Klasifikasi Berita Menggunakan Metode Support Vector Machine," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 5, no. 2, pp. 269–278, 2022, doi: 10.32672/jnkti.v5i2.4193.

[19] D. F. AL-Hafiidh, I. F. Rozi, and I. K. Putri, "Peringkasan Teks Otomatis pada Portal Berita Olahraga menggunakan metode Maximum Marginal Relevance.," *J. Inform. Polinema*, vol. 8, no. 3, pp. 21–30, 2022, doi: 10.33795/jip.v8i3.519.

[20] D. H. Amalia and W. Yustanti, "Klasifikasi Buku Menggunakan Metode Support Vector Machine pada Digital Library," *J. Informatics Comput. Sci.*, vol. 3, no. 01, pp. 55–61, 2021, doi: 10.26740/jinacs.v3n01.p55-61.

[21] J. Heidari, N. Daneshpour, and A. Zangeneh, "A novel K-means and K-medoids algorithms for clustering non-spherical-shape clusters non-sensitive to outliers," *Pattern Recognit.*, vol. 155, no. May, p. 110639, 2024, doi: 10.1016/j.patcog.2024.110639.

[22] H. Zhang *et al.*, "Webshell traffic detection with character-level features based on deep learning," *IEEE Access*, vol. 6, pp. 75268–75277, 2018, doi: 10.1109/ACCESS.2018.2882517.

[23] B. Subba and P. Gupta, "A tfidfvectorizer and singular value decomposition based host intrusion detection system framework for detecting anomalous system processes," *Comput. Secur.*, vol. 100, p. 102084, 2021, doi: 10.1016/j.cose.2020.102084.

[24] M. Berhili, O. Chaieb, and M. Benabdellah, "Intrusion Detection Systems in IoT Based on Machine Learning: A state of the art," *Procedia Comput. Sci.*, vol. 251, pp. 99–107, 2024, doi: 10.1016/j.procs.2024.11.089.

[25] Z. T. Sworna, Z. Mousavi, and M. A. Babar, "NLP methods in host-based intrusion detection systems: A systematic review and future directions," *J. Netw. Comput. Appl.*, vol. 220, no. November 2022, p. 103761, 2023, doi: 10.1016/j.jnca.2023.103761.