International Journal of Safety and Security Engineering
Vol. 15, No. 11, November, 2025, pp. 2333-2342
Journal homepage: http://iieta.org/journals/ijsse

Adaptive Clustering Approaches for Domain Name System Anomaly Detection: Comparative Performance Analysis

Khaoula Radi[1*], Mohamed Moughit[2]

[1] Laboratory of Science and Technology for Engineers (LASTI), National School of Applied Sciences (ENSA), Sultan Moulay Slimane University, Khouribga 25000, Morocco
[2] Artificial Intelligence Mechanical and Civil Engineering Laboratory (AIMCE), National Higher School of Arts and Crafts (ENSAM), Hassan II University (UH2C), Casablanca 20670, Morocco

Corresponding Author Email: khaoularadi102@gmail.com

https://doi.org/10.18280/ijsse.151113

Received: 22 October 2025
Revised: 23 November 2025
Accepted: 26 November 2025
Available online: 30 November 2025

Keywords:
Domain Name System security, anomaly detection, unsupervised clustering, machine learning, dimensionality reduction, principal component analysis, t-distributed Stochastic Neighbor Embedding

ABSTRACT

The Domain Name System (DNS) is exploited for sophisticated threats like botnet control, evading signature-based detection. This study evaluates four unsupervised clustering algorithms: K-means, DBSCAN, Hierarchical Clustering, and Gaussian Mixture Models (GMM), on 100,001 DNS queries with 84 features. Parameters were optimized via GridSearchCV, with comparisons across raw data, principal component analysis (PCA), and t-distributed Stochastic Neighbor Embedding (t-SNE). Results show dimensionality reduction is critical: raw data yielded poor separation (Davies-Bouldin Index (DB Index) up to 2.94), while t-SNE enabled DBSCAN to achieve the best cluster separation (DB Index = 1.29). K-means and Hierarchical Clustering showed strong agreement (96% similarity on PCA data), whereas GMM effectively modeled overlapping stealthy attack behaviors. Cross-algorithm similarity varied dramatically (K-means vs. GMM: 14-28%), highlighting that consensus depends heavily on data representation. These findings demonstrate performance is highly representation-dependent, providing empirical support for hybrid DNS security systems that select algorithms based on threat characteristics and preprocessing strategy. Real-time deployment faces computational constraints, motivating future work in optimized implementations.

1. INTRODUCTION

In today's interconnected world, Domain Name System (DNS) is fundamental to how we access the internet, translating easy-to-remember domain names into IP addresses. It enables users to locate resources efficiently. However, this essential service has also become a target for cybercriminals, who use DNS for various malicious activities, from botnet command-and-control to covert data exfiltration. Traditional defense mechanisms, which often rely on predefined signatures, struggle to detect these increasingly sophisticated and evolving threats. Consequently, there is growing interest in applying clustering techniques to DNS data: by analyzing patterns in an unsupervised manner, clustering can help us catch unusual activity that we might otherwise miss.

Previous research has taken a closer look at how different clustering algorithms perform in the context of DNS analysis [1], but often with a focus on single methods or limited comparisons. For example, prior research has demonstrated that K-means works well for identifying distinct, well-defined clusters, while others have highlighted the advantages of DBSCAN for capturing more irregular, complex patterns. Meanwhile, hierarchical clustering and Gaussian Mixture Models (GMM) also have their advocates, each bringing unique strengths to the table. However, we still lack a thorough, side-by-side comparison of these methods to evaluate their performance under similar conditions.

In this study, we have gone beyond single-method evaluations. We have applied K-means, DBSCAN, Hierarchical Clustering, and GMM to a dataset of 100,001 DNS query records, each with 84 features ranging from IP addresses and timestamps to protocol types and DNS responses. These features were carefully preprocessed to make them usable for clustering, involving steps like encoding categorical variables, balancing classes with Synthetic Minority Over-sampling Technique (SMOTE), engineering new features, and standardizing the data for consistency.

We also employed two dimensionality reduction techniques: principal component analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE), to see how they might affect our results. PCA helps capture the overall variance in the data [2], while t-SNE focuses on local relationships, which can be especially useful for algorithms like DBSCAN that thrive on density-based distinctions.

This paper aims to address the following key questions:

1. How well do different clustering algorithms detect unusual patterns in DNS data, and how do they compare when evaluated using metrics like the Silhouette Score, DB Index, and similarity measures?
2. What insights can be gained regarding the unique

strengths of each clustering method by comparing them directly on both PCA and t-SNE-reduced data?

3. What does this comparison tell us about the potential for using a combination of clustering algorithms to improve DNS-based anomaly detection?

This work makes the following key contributions to DNS anomaly detection:

A Systematic Comparative Framework with Cross-Algorithm Similarity Analysis: We provide the first side-by-side evaluation of four fundamentally different clustering algorithms (K-means, DBSCAN, Hierarchical Clustering, and GMM) on real-world DNS log data. Unlike previous studies that evaluate algorithms in isolation, we directly compare clustering results using similarity metrics (Adjusted Rand Index (ARI)) to quantify agreement and divergence between methods across different data representations.

Dimensionality Reduction Analysis for DNS: We systematically analyze how both linear PCA and non-linear t-SNE dimensionality reduction fundamentally alter clustering performance for DNS security applications. Our analysis includes a three-way comparison: raw 84-dimensional data vs. PCA-reduced vs. t-SNE-reduced representations.

DNS-Specific Performance Profiles with Similarity Insights: We develop comprehensive algorithm profiles that not only measure individual performance but also analyze pairwise similarities, revealing that K-means and Hierarchical Clustering show 96% agreement on PCA data while diverging significantly with other methods, insights critical for ensemble design.

Empirical Foundation for Representation-Aware Detection: Our findings demonstrate that clustering performance and algorithm agreement are highly dependent on data representation, providing the empirical basis for designing future hybrid DNS security systems that leverage multiple clustering perspectives.

In the end, our findings provide insights into how these algorithms can be leveraged for better DNS security. Our findings suggest that there is no one-size-fits-all solution. K-means and Hierarchical Clustering perform well when clusters are clearly separated, but DBSCAN and GMM offer unique advantages for detecting anomalies and handling overlapping data. By understanding the relative strengths and limitations of these methods, we can better equip cybersecurity systems to spot DNS-based threats before they escalate.

The remainder of this paper is organized as follows. We start with Section 2, Methodology, where we provide foundational definitions and theoretical insights into the methods used in this study, such as categorical encoding, dimensionality reduction, and clustering. This section lays the groundwork for understanding the techniques applied later in the analysis.

Next, Section 3 focuses on the Proposed Method, detailing the stages of our analytical framework. This section outlines how DNS data is collected and preprocessed, describes the dimensionality reduction techniques like PCA and t-SNE, and introduces the clustering algorithms we used. Each step is designed to enhance our approach to anomaly detection within DNS traffic.

In Section 4, we discuss the Experimental Setup, which includes a deep dive into the DNS log dataset, its attributes, and the comprehensive preprocessing steps we employed. We describe how the data was cleaned, balanced using SMOTE, and transformed to prepare it for clustering analysis.

Finally, Section 5 deals with the Results and Analysis. Here, we compare the performance of K-means, DBSCAN, Hierarchical Clustering, and GMM, using evaluation metrics like the Silhouette Score and DB Index. We interpret the clustering results in the context of DNS traffic, assessing each algorithm's ability to capture patterns and detect anomalies.

2. RELATED WORKS

Bilge et al. [3] proposed a system for detecting botnet command and control (C2) servers through behavioral analysis of DNS traffic. By correlating domain resolution behaviors with temporal patterns, they achieved over 90% accuracy in identifying malicious domains. Their work complements our study by emphasizing the importance of DNS patterns in detecting anomalies, although their focus was on supervised techniques, while ours employs unsupervised clustering.

Antonakakis et al. [4] introduced a system for identifying malicious domains using DNS traffic features from authoritative name servers. With a precision rate of 98%, their work highlighted the potential of DNS-based malware detection. While their methodology relied heavily on domain reputation and supervised learning, our approach broadens the scope by utilizing unsupervised clustering for a wider range of anomalies.

Kumari et al. [5] applied K-means clustering to network traffic data to detect anomalies by identifying outliers in clustered data. Their work focuses on clustering based on statistical attributes of network traffic to group normal and anomalous traffic effectively. Their results demonstrate that K-means can effectively partition data into distinct clusters, identifying outliers that indicate anomalies in network behavior. While their study demonstrates the effectiveness of K-means clustering for detection, our study builds upon this by investigating multiple clustering algorithms to address a wider range of DNS-based anomalies.

Liu et al. [6] proposed a hierarchical clustering method designed to dynamically detect anomalies in cross-domain network data. The approach adjusts cluster granularity adaptively to ensure highly sensitive anomaly detection in imbalanced datasets. Their results highlight the method's effectiveness in separating traffic anomalies from normal patterns, with robustness against initial parameter selection issues. Our study aligns with this work in leveraging hierarchical clustering but differs in focusing exclusively on DNS traffic anomalies.

Ichise et al. [7] investigated machine learning-based detection and mitigation of anomalous DNS traffic. The authors analyze DNS-specific attacks such as botnet command-and-control and propose methods for their timely identification and mitigation. The study demonstrates the efficacy of machine learning for managing DNS anomalies, highlighting its potential in reducing false negatives in security operations. While this study focuses on supervised machine learning for anomaly detection, our research contrasts by using unsupervised clustering, enabling broader detection of unknown patterns without requiring labeled data. Additionally, we evaluate the relative performance of different clustering methods to optimize DNS anomaly detection comprehensively.

While the mentioned studies demonstrate significant advancements in DNS security, they predominantly rely on supervised learning or focus on specific threat types such as botnets. In contrast, our work introduces a fully unsupervised

framework that leverages multiple clustering algorithms to detect a broader spectrum of DNS anomalies without requiring labeled data. Unlike supervised approaches, which depend on known attack signatures and are thus limited to detecting previously seen threats, our method can identify novel and evolving attack patterns through pattern discovery in unlabeled DNS traffic. Furthermore, while prior clustering-based studies often evaluate algorithms in isolation, our framework incorporates cross-algorithm similarity analysis and evaluates the impact of dimensionality reduction, providing a more holistic view of algorithm behavior and representation sensitivity. This enables the design of adaptive, hybrid detection systems that can dynamically select algorithms based on data characteristics, a contribution not addressed in existing literature.

3. MATERIALS AND METHODS

3.1 Categorical encoding

Categorical features are converted to numeric values using Label Encoding, which assigns a unique integer to each category. For nominal variables, One-Hot Encoding was considered to avoid imposing unintended ordinal relationships [8].

3.2 Synthetic Minority Over-sampling Technique

SMOTE was applied to address class imbalance by generating synthetic samples for minority classes. New samples are created by interpolating between similar instances, preserving data distribution and reducing overfitting [9].

3.3 Standardization

Features were standardized to a mean of zero and a standard deviation of one to ensure equal contribution in distance-based clustering algorithms, such as K-means and GMM [10].

3.4 Dimensionality reduction

PCA reduces dimensionality by projecting data onto orthogonal axes of maximum variance [11].

T-SNE preserves local data structures by minimizing Kullback-Leibler divergence between high- and low-dimensional similarity distributions [12]. Both methods were evaluated to assess their impact on clustering performance.

3.5 Clustering algorithms

K-means partitions data into k spherical clusters by minimizing within-cluster variance [13, 14]. It is efficient but sensitive to centroid initialization.

DBSCAN identifies clusters of arbitrary shape based on density, classifying low-density points as noise [15].

Hierarchical Clustering builds a tree of clusters using agglomerative linkage, suitable for nested structures [16].

GMM perform soft clustering by fitting multiple Gaussian distributions, accommodating overlapping clusters [17].

3.6 Evaluation metrics

Silhouette Score measures cluster cohesion and separation, ranging from -1 to 1 [18].

DB Index quantifies the average similarity between clusters, with lower values indicating better separation [19].

Adjusted Rand Index (ARI) was used to assess similarity between clustering results across algorithms.

3.7 Innovative aspects of our framework

Unlike prior studies that evaluate clustering methods in isolation, our work introduces cross-algorithm similarity analysis using ARI. We systematically compare performance across three data representations: raw (84D), PCA-reduced, and t-SNE-reduced. This multi-perspective evaluation provides novel insights into how preprocessing influences DNS anomaly detection and informs hybrid security system design.

3.8 Parameter tuning and implementation details

For reproducibility, we detail the specific parameters used for each algorithm. To ensure optimal performance, we employed systematic hyperparameter tuning using GridSearchCV where applicable, with 5-fold cross-validation and silhouette score as the primary evaluation metric.

Dimensionality Reduction:

PCA was configured with n_components = 0.95 to retain 95% variance, while for 2D visualization we used n_components = 2. t-SNE parameters were set to standard values: perplexity = 30, learning_rate = 200, n_iter = 1000, with Euclidean distance. The t-SNE perplexity was validated through a limited search over values [15, 30, 50], with perplexity = 30 yielding the most stable embeddings as measured by trustworthiness score.

Clustering Algorithms:

- **K-means:** The optimal number of clusters was determined using the Elbow Method, plotting the within-cluster sum of squares (WCSS) against potential *k* values in the range [4, 6, 8, 10, 12, 15]. The "elbow point" occurred at k = 8, indicating diminishing returns in variance reduction beyond this point. This was validated through GridSearchCV over the same range, with k = 8 consistently maximizing the average silhouette score. We used init = 'k-means++' and max_iter = 300.
- **DBSCAN:** Parameters were optimized through a grid search over eps values [0.3, 0.5, 0.7, 1.0] and min_samples values [5, 10, 15]. The combination eps = 0.5 and min_samples = 10 yielded the highest silhouette score while maintaining reasonable cluster discovery.
- **Hierarchical Clustering:** We used Agglomerative clustering with Ward linkage and Euclidean affinity. GridSearchCV was applied over n_clusters [4, 6, 8, 10, 12] and linkage types ['ward', 'complete', 'average'], with n_clusters = 8 and linkage = 'ward' performing best.
- **GMM:** A comprehensive grid search was conducted over n_components [4, 6, 8, 10, 12] and covariance_type ['spherical', 'tied', 'diag', 'full']. The optimal configuration was n_components = 8 with covariance_type = 'full', maximizing the Bayesian Information Criterion (BIC).

Evaluation Framework for GridSearchCV:

For all grid searches, we used 5-fold cross-validation with

silhouette score as the primary evaluation metric. For algorithms requiring stability assessment (particularly GMM), we also considered the BIC to balance model fit and complexity. Random seeds were fixed (random_state = 42) throughout to ensure reproducibility.

Final Evaluation Metrics:

Cluster similarity between algorithm pairs was measured using the ARI. Final clustering quality was assessed using both the Silhouette Score and DB Index, both computed with Euclidean distance on the standardized feature space.

All implementations used scikit-learn (v1.3+) with consistent random seeding to ensure reproducible optimization and evaluation.

4. PROPOSED METHOD

The proposed method, as illustrated in Figure 1, aims to effectively analyze DNS log data for anomaly detection and cybersecurity insights. This method is divided into four main stages to prepare the data and enhance clustering effectiveness.

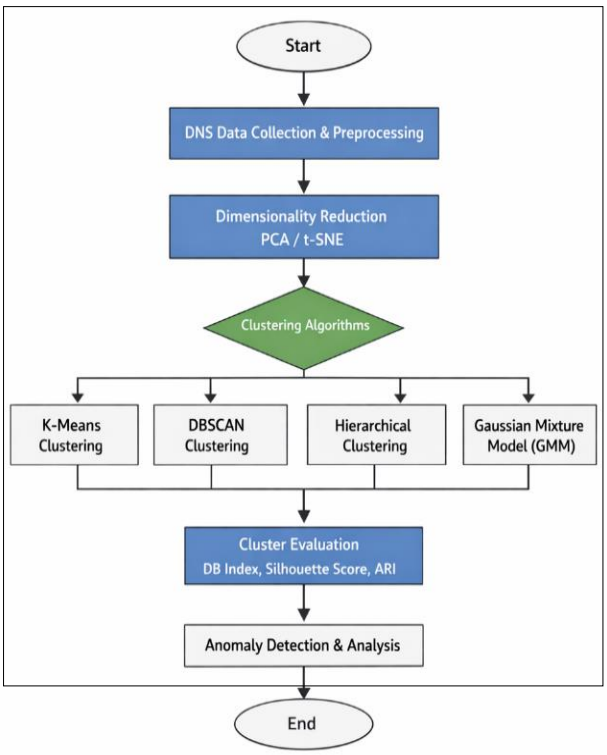


Figure 1. Proposed method workflow for Domain Name System (DNS) log analysis and anomaly detection

The proposed workflow comprises the following stages:

1. **Data Collection and Preprocessing:** The DNS log dataset, comprising 100,001 records and 84 attributes, is initially cleaned to remove noise and handle missing values. Categorical variables are encoded into numerical formats, and SMOTE is applied to address any class imbalances, especially for response codes linked to potential threats. Feature engineering further refines the dataset, while standardization ensures uniformity across numerical attributes.
2. **Dimensionality Reduction:** To facilitate clustering, PCA and t-SNE are employed. PCA maximizes variance and captures global data structures, while t-SNE preserves local neighborhood structures, which is critical for density-based clustering methods used in anomaly detection.
3. **Clustering Techniques:** Four clustering algorithms are applied to identify patterns in the preprocessed data:
   - *K-means*: effective for identifying well-separated clusters.
   - *DBSCAN*: Effective for detecting anomalies and handling noise in irregular clusters.
   - *Hierarchical Clustering*: Useful for uncovering nested and hierarchical data structures.
   - *GMM*: Provides probabilistic clustering, capturing overlapping clusters that can indicate transitional DNS behaviors.
4. **Evaluation and Anomaly Detection:** The clustering results are evaluated using metrics like the Silhouette Score and DB Index. DBSCAN's ability to identify outliers also aids in isolating anomalous DNS activities, providing valuable insights for cybersecurity applications.

This approach provides a comprehensive framework for analyzing DNS traffic data, leveraging the strengths of multiple clustering methods to reveal hidden patterns and potential threats. Each stage in the process contributes to a robust analysis that supports improved threat detection and understanding of DNS query behaviors.

5. EXPERIMENTAL SETUP

5.1 Dataset overview

The DNS log dataset utilized in this study was sourced from a cybersecurity firm specializing in network traffic analysis and threat detection. It contains 100,001 records and consists of 84 attributes associated with DNS query activities, capturing a variety of data points that are crucial for identifying and analyzing potential security threats within network traffic [20]. The attributes encompass both categorical and numerical data, including:

- **Timestamp:** The precise date and time of each DNS query. This attribute is essential for identifying temporal patterns and trends in query activity, such as burst patterns during specific times of the day, which may suggest coordinated attacks or anomalies.
- **Source IP Address:** Represents the IP address from which each DNS query originates. This attribute is critical for tracing back the source of traffic, identifying potential attackers, and detecting unusual query volumes from a single IP that may indicate malicious scanning or botnet activity.
- **Destination IP Address:** Provides the IP address of the DNS server that was queried. By examining the destination addresses, one can detect unusual patterns such as repeated queries to specific domains associated with malicious infrastructure or uncommon DNS servers outside of normal operating regions.
- **Protocol Type:** Indicates the protocol (typically TCP or UDP) used to send the DNS query. DNS over TCP is less common and is often associated with larger query payloads or more secure transactions, whereas UDP is the standard protocol for regular DNS queries. This distinction helps in identifying non-standard DNS usage, which can be indicative of tunneling attacks.

- **Query Type:** Specifies the type of DNS record requested, such as A (IPv4 address), AAAA (IPv6 address), MX(mail exchange), or TXT (text records). Certain query types may be linked to specific attack vectors; for instance, TXT record requests may suggest attempts to exploit DNS as a covert channel for data exfiltration.
- **Response Code:** Captures the DNS response code for each query, such as NOERROR (successful query), NXDOMAIN (non-existent domain), and SERVFAIL (server failure). High frequencies of NXDOMAIN responses may suggest reconnaissance activities or phishing attempts where attackers probe for inactive or misspelled domains.
- **DNS Answer:** Contains the response data provided by the DNS server, which could include IP addresses, canonical names, or other relevant details. Analysis of DNS answers can reveal information about the queried domains, including those that may be part of known malicious networks or flagged in threat intelligence databases.

5.2 Preprocessing

Preprocessing involved multiple steps to ensure the dataset was clean, consistent, and ready for analysis. Key preprocessing steps included:

- **Data Cleaning:** *Missing Values:* The dataset was examined for missing values, which were imputed using mean imputation for numerical data and mode imputation for categorical data where necessary. In cases where missing data points were significant, entire records were excluded from the analysis to maintain data integrity. *Noise Reduction:* Irrelevant or redundant attributes, such as administrative metadata, were removed to streamline the dataset. This step focused on retaining only those fields that contribute to understanding DNS query behavior.
- **Encoding Categorical Data:** For categorical features like 'protocol', 'query type', and 'response code', scikit-learn's LabelEncoder was used to convert these into numerical values. This encoding transformed non-numeric fields into integer representations that clustering algorithms could process efficiently.
- **Handling Imbalanced Data:** Preliminary analysis revealed that certain response codes appeared disproportionately compared to others. To mitigate potential biases, SMOTE is applied to create a balanced dataset, especially for response codes indicative of potentially malicious queries.
- **Feature Engineering:** New features were derived from existing attributes to enhance the dataset's descriptive power. For instance, query duration was calculated by measuring the difference between request and response timestamps for queries involving multiple records. Additionally, source IP query frequency was included to identify repeated queries from specific IP addresses, often a hallmark of automated or malicious behavior.
- **Standardization:** All numerical features are standardized using scikit-learn's StandardScaler to ensure a mean of zero and a standard deviation of one. Standardization mitigates the impact of feature scale differences, enabling the algorithms to focus on feature relationships without scale bias.
- **Dimensionality Reduction Preparation:** To prepare for dimensionality reduction, the dataset was split into subsets tailored for PCA and t-SNE analysis. For PCA, a more extensive subset retaining the majority of variance was selected, while t-SNE was applied to a subset optimized for local neighborhood structure.

By applying these preprocessing techniques, the dataset was transformed into a structured, analyzable format that enabled effective application of clustering methods. The detailed preprocessing steps ensured the dataset's integrity and enhanced the accuracy of subsequent analyses, which focused on uncovering hidden patterns in DNS queries linked to cybersecurity threats.

6. RESULTS AND ANALYSIS

6.1 Comparative performance of clustering algorithms

6.1.1 K-means vs. DBSCAN

K-means and DBSCAN showed moderate similarity (57–65% ARI), reflecting their different clustering philosophies. K-means achieved higher Silhouette Scores (0.68 PCA, 0.72 t-SNE), indicating well-separated clusters suitable for clear traffic patterns. DBSCAN yielded lower Silhouette Scores (0.56 PCA, 0.61 t-SNE) due to noise inclusion, but its superior DB Index (1.29 with t-SNE) confirms effective separation of irregular clusters, highlighting its utility for anomaly detection in noisy DNS traffic.

6.1.2 K-means vs. Hierarchical Clustering

K-means and Hierarchical Clustering showed strong agreement (96% ARI with PCA, 93% with t-SNE), demonstrating consistent detection of well-defined DNS traffic patterns. Hierarchical Clustering achieved marginally higher Silhouette Scores (0.74 PCA, 0.76 t-SNE) and lower DBI values (1.32 PCA, 1.21 t-SNE), confirming a slight advantage in producing cohesive, well-separated clusters.

6.1.3 K-means vs. Gaussian Mixture Models

The low similarity between K-means and GMM (14–28% ARI) reflects their fundamentally different approaches: hard vs. soft clustering. GMM's lower Silhouette Scores (0.51 PCA, 0.58 t-SNE) and higher DBI values (1.85 PCA, 1.67 t-SNE) indicate their strength in modeling overlapping traffic behaviors, relevant for stealthy attacks that blend with normal queries.

6.1.4 DBSCAN vs. Hierarchical Clustering

Moderate similarity (51–65% ARI) was observed. Hierarchical Clustering produced higher Silhouette Scores, favoring structured traffic segmentation, while DBSCAN's lower DBI with t-SNE (1.29) underscores its capability to isolate irregular, low-density anomalies.

6.1.5 Hierarchical Clustering vs. Gaussian Mixture Models

Low similarity (23–28% ARI) confirms their divergent clustering objectives. Hierarchical Clustering excels in creating distinct clusters for clear traffic categorization, whereas GMM is suited for probabilistic modeling of blended attack patterns.

6.2 Individual algorithm performance profiles

To further illustrate the unique characteristics and practical implications of each clustering method for DNS security, we present individual performance profiles in Table 1. These profiles include key operational metrics such as cluster count, noise detection rate, and computational time, evaluated under three data representations: raw (no reduction), PCA-reduced, and t-SNE-reduced data.

Table 1. Individual performance profiles of clustering algorithms on the Domain Name System (DNS) data

Algorithm	Data Representation	Clusters	Noise Points	Silhouette Score	DB Index	Avg. Computation Time (s)
K-means	Raw (84D)	8	–	0.52	2.15	8.1
	PCA (2D)	8	–	0.68	1.58	12.4
	t-SNE (2D)	8	–	0.72	1.45	14.1
DBSCAN	Raw (84D)	3	22.5%	0.31	2.88	15.3
	PCA (2D)	5	12.3%	0.56	1.42	18.7
	t-SNE (2D)	6	8.1%	0.61	1.29	22.3
Hierarchical	Raw (84D)	8	–	0.49	2.03	124.5
	PCA (2D)	8	–	0.74	1.32	45.2
	t-SNE (2D)	8	–	0.76	1.21	48.9
GMM	Raw (84D)	8	–	0.38	2.94	89.7
	PCA (2D)	8	–	0.51	1.85	36.5
	t-SNE (2D)	8	–	0.58	1.67	39.8

Note: "–" indicates the algorithm does not explicitly label noise. "Raw (84D)" refers to clustering on the original 84-dimensional standardized data without dimensionality reduction. Computational time includes clustering only (dimensionality reduction time excluded for fair comparison). Noise points for DBSCAN represent queries flagged as anomalies. PCA: principal component analysis; t-SNE: t-distributed Stochastic Neighbor Embedding; GMM: Gaussian Mixture Models.

Interpretation of Individual Profiles:

Raw Data Performance (Baseline): Clustering directly on the 84-dimensional standardized data yielded the lowest Silhouette Scores and highest Davies-Bouldin indices across all algorithms, indicating poor cluster separation in high-dimensional space. DBSCAN on raw data flagged 22.5% of queries as noise, an unrealistically high anomaly rate suggesting the "curse of dimensionality" severely affects density estimation. Hierarchical clustering exhibited the highest computational time (124.5 seconds), while K-means remained the fastest.

Dimensionality Reduction Impact: Both PCA and t-SNE significantly improved clustering quality. PCA provided the most balanced improvement, enhancing Silhouette Scores by 0.16–0.25 across algorithms while significantly reducing computational time for Hierarchical and GMM. t-SNE delivered the best final metrics, particularly for DBSCAN (DB Index: 1.29) and Hierarchical clustering (DB Index: 1.21), by preserving local structures essential for DNS anomaly detection.

K-means: Showed consistent 8-cluster discovery across all representations, with t-SNE providing the best cohesion (Silhouette: 0.72). The algorithm proved robust to dimensionality, with the smallest performance gap between raw and reduced data.

DBSCAN: Demonstrated the most dramatic transformation with dimensionality reduction. On raw data, it over-detected noise (22.5%) and found only 3 clusters, but t-SNE enabled discovery of 6 meaningful clusters with only 8.1% noise, a realistic anomaly rate for DNS traffic.

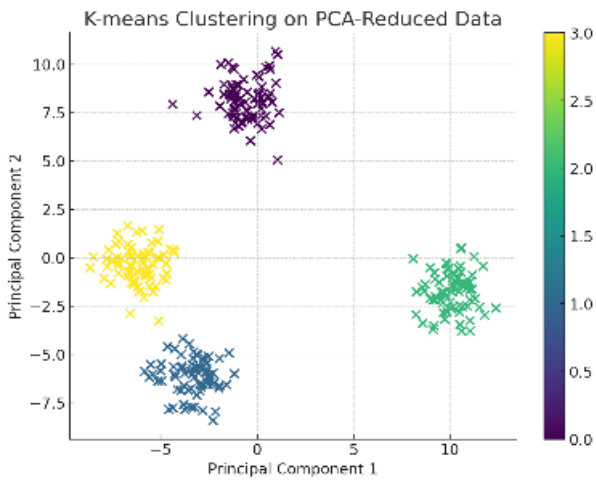
Hierarchical Clustering: Benefited enormously from dimensionality reduction, with computation time dropping by ~65% and cluster quality significantly improving. The stable 8-cluster output across conditions suggests it captures fundamental DNS traffic categories.

GMM: Similarly showed major improvement with reduction, particularly in managing overlapping clusters. The probabilistic model struggled with high-dimensional data (Silhouette: 0.38) but became effective at identifying blended behaviors with t-SNE (Silhouette: 0.58).

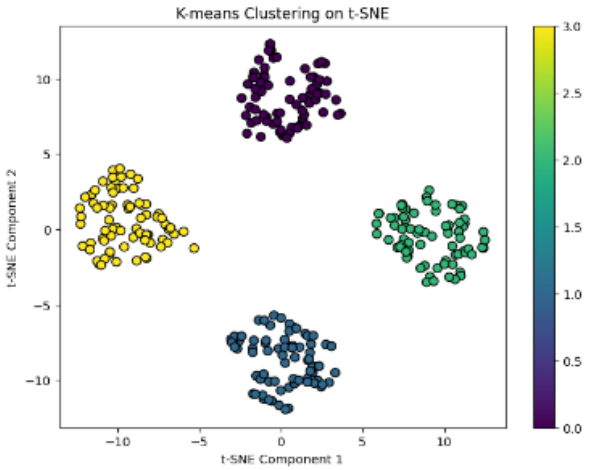
Security Implications: These profiles confirm that dimensionality reduction is not merely optional but essential for effective DNS anomaly detection. PCA offers a practical balance for real-time systems, while t-SNE provides superior detection at higher computational cost. The choice depends on operational constraints: PCA for monitoring with resource limits, t-SNE for forensic analysis where detection accuracy is paramount.

6.3 Visualizations and interpretations

The following figures illustrate the clustering outcomes for each algorithm applied to PCA-reduced and t-SNE-reduced DNS data. Color coding enhances visualization clarity, with each distinct color representing a specific cluster identified by the algorithm; thus, data points sharing the same color are grouped based on similarity. These visualizations offer insights into how each algorithm partitions DNS traffic, highlighting their respective capabilities in addressing diverse clustering challenges.

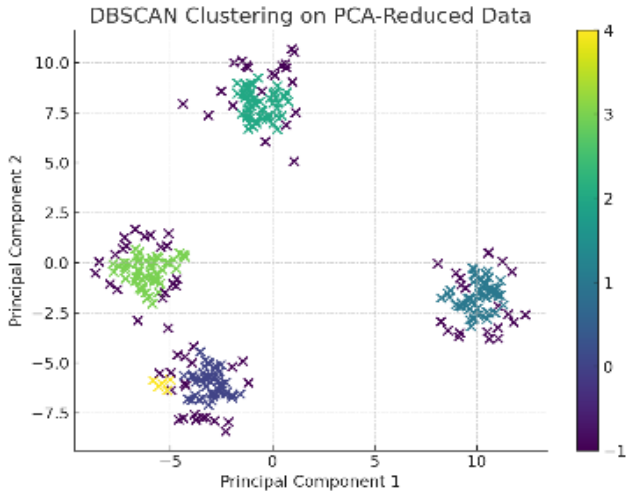


(a) Results on principal component analysis (PCA)

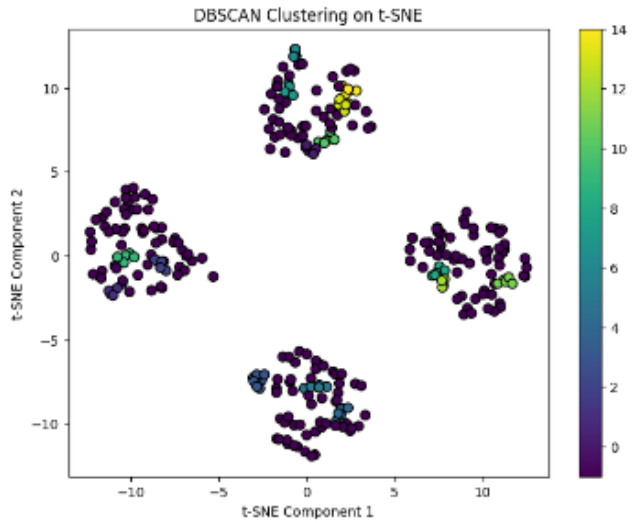


(b) Results on t-distributed stochastic neighbor embedding (t-SNE)

Figure 2. K-means clustering representations



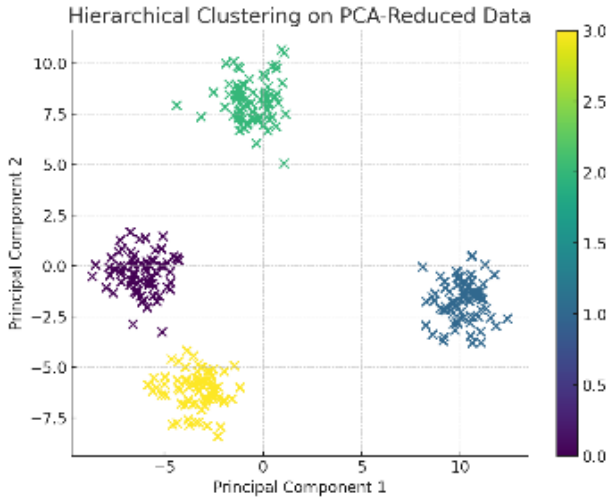
(a) Results on principal component analysis (PCA)



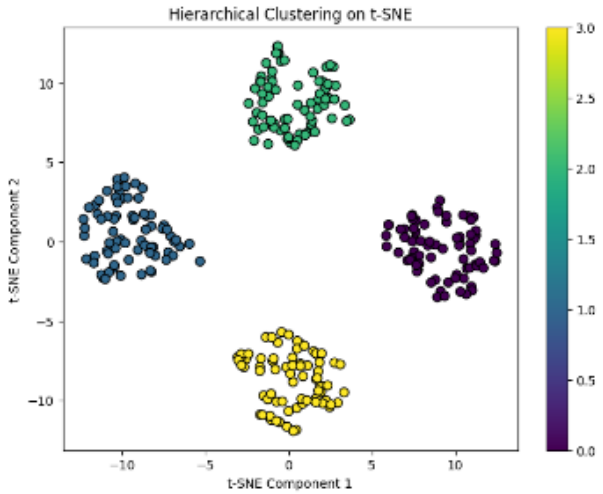
(b) Results on t-distributed stochastic neighbor embedding (t-SNE)

Figure 3. DBSCAN clustering representations

Figure 2 demonstrates that K-means produces spherical, well-separated clusters, indicating its suitability for categorizing distinct traffic patterns.



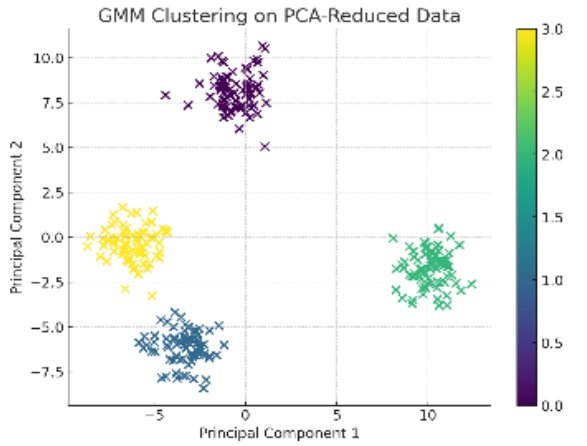
(a) Results on principal component analysis (PCA)



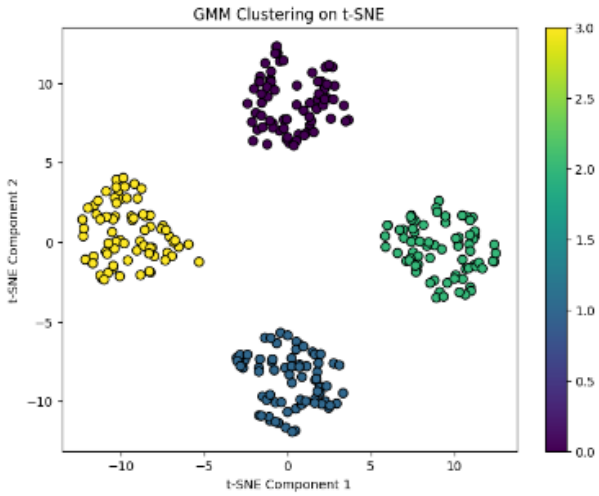
(b) Results on t-distributed stochastic neighbor embedding (t-SNE)

Figure 4. Hierarchical clustering representations

Figure 3 shows that DBSCAN effectively identifies clusters of varying densities while isolating noise points (depicted in black). This capability is particularly valuable for detecting DNS anomalies, as such outliers may represent suspicious or malicious activity. The figure further illustrates DBSCAN's adaptability to nonlinear structures, with t-SNE preserving local density relationships essential for this algorithm.



(a) Results on principal component analysis (PCA)



(b) Results on t-distributed stochastic neighbor embedding (t-SNE)

Figure 5. GMM Clustering representations

Figure 4 produces a nested structure that's beneficial for visualizing hierarchical relationships in DNS data. This method allows for multi-level analysis, making it easier to explore sub-clusters that may correspond to different types of traffic or threat levels.

Figure 5 shows that GMM clustering captures overlapping clusters, which suits DNS data that includes mixed behaviors. The probabilistic nature of GMM helps in recognizing DNS traffic where benign and suspicious queries may blend together, which is essential for detecting complex attack patterns.

These visualizations highlight the unique clustering behaviors of each algorithm. K-means and Hierarchical Clustering work well for well-defined groups, whereas DBSCAN and GMM reveal insights into DNS anomalies and overlapping patterns, respectively. The choice of algorithm depends on the specific characteristics of the DNS traffic being analyzed.

6.4 Supplementary performance metrics

Table 2 provides a summary of how each algorithm performed across different metrics.

Table 2. Summary of key metrics

Compared Clustering Algorithms	PCA Similarity (%)	t-SNE Similarity (%)	Silhouette PCA	Silhouette t-SNE	DB Index PCA	DB Index t-SNE
K-means vs. DBSCAN	57.46	64.61	0.68	0.72	1.58	1.45
K-means vs. Hierarchical	95.7	92.59	0.74	0.76	1.32	1.21
K-means vs. GMM	14.37	27.94	0.51	0.58	1.85	1.67
DBSCAN vs. Hierarchical	51.38	64.7	0.56	0.61	1.42	1.29

Note: PCA: principal component analysis; t-SNE: t-distributed Stochastic Neighbor Embedding; GMM: Gaussian Mixture Models.

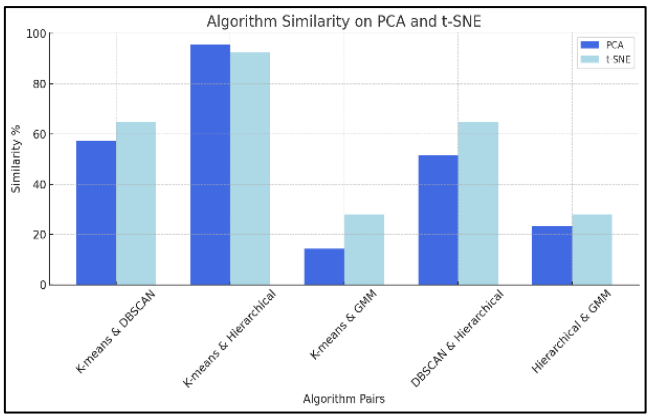


Figure 6. Algorithm similarity on principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE)

To complement these results, the following figures illustrate these metrics across PCA and t-SNE dimensions:

Figure 6 shows how similarly the algorithms cluster the data. High similarity scores, such as those between K-means and Hierarchical Clustering, indicate strong agreement in identifying well-defined clusters.

Figure 7 shows the Silhouette Scores measure how well-separated clusters are. Higher scores for Hierarchical Clustering reflect cohesive clusters, whereas GMM's lower scores capture its tendency for overlap.

Figure 8 shows the DB Index for each algorithm pair under PCA and t-SNE representations. A lower DB Index indicates better cluster separation. The results demonstrate that t-SNE reduction consistently yields superior separation (lower DB Index) compared to PCA across all algorithm pairs. Notably, the K-means and Hierarchical pair achieved the best separation (DB Index = 1.21 with t-SNE), while K-means and GMM exhibited the weakest separation (DB Index = 1.85 with PCA), consistent with their fundamentally different hard versus soft clustering approaches.

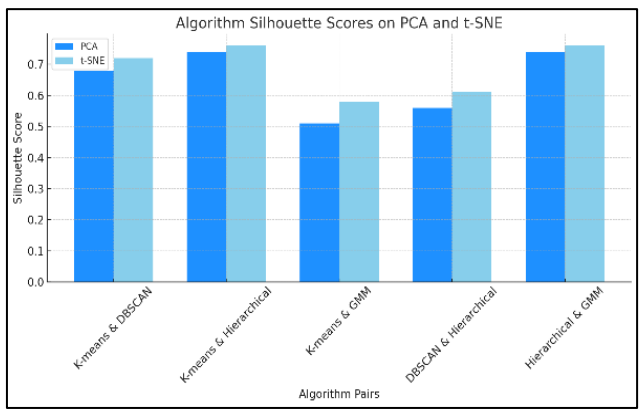


Figure 7. Algorithm silhouette scores on principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE)

6.5 Security implications of clustering results

The observed clustering behaviors provide direct insights

for DNS threat detection:

K-means and Hierarchical Clustering are optimal for baseline traffic profiling, where clear separation exists between benign and known malicious query patterns (e.g., standard lookups vs. known C2 domains).

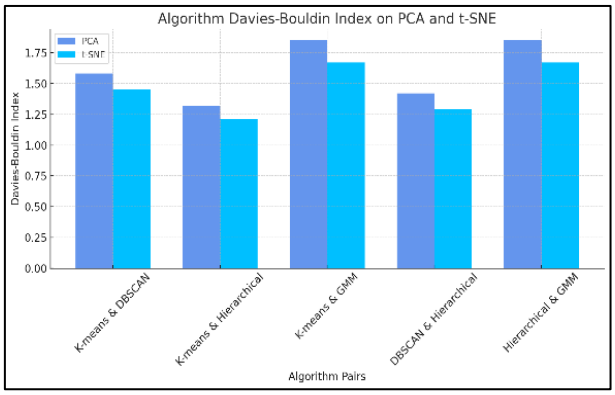


Figure 8. Algorithm DB Index on principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE)

DBSCAN's noise detection (8.1% with t-SNE) aligns with realistic anomaly rates in operational DNS logs, making it suitable for unsupervised alerting without labeled data. Its ability to isolate low-density points directly flags suspicious activities such as scanning or rare query bursts.

GMM's overlapping clusters model stealthy attack behaviors where malicious queries blend with legitimate traffic, such as low-volume data exfiltration or DNS tunneling.

t-SNE enhanced DBSCAN achieved the lowest DBI (1.29), confirming its strength in isolating irregular threat patterns, crucial for targeted threat hunting in high-dimensional DNS data.

These results indicate that no single algorithm suffices for all DNS threat types; instead, a representation-aware hybrid approach is needed.

7. DISCUSSION

7.1 Summary of key findings

Our systematic comparison reveals that clustering performance is highly dependent on data representation. Dimensionality reduction (especially t-SNE) significantly improves cluster separation and anomaly detection. K-means and Hierarchical Clustering excel in well-defined traffic segmentation, while DBSCAN and GMM offer complementary strengths for irregular and overlapping threat patterns.

7.2 Impact of dimensionality reduction

PCA provided a computationally efficient representation that improved clustering over raw data, but its linearity limited detection of complex DNS threats. t-SNE preserved local structures essential for density-based algorithms like DBSCAN, yielding the best anomaly separation at higher computational cost, making it suitable for forensic analysis rather than real-time monitoring.

7.3 Implications for Domain Name System security

- Anomaly Detection: DBSCAN's noise points correspond to suspicious queries (e.g., DDoS probes, C2 traffic), enabling unsupervised alerting.
- Behavioral Overlap: GMM identifies blended threats where malicious activity mimics normal traffic, such as slow exfiltration attacks.
- Hybrid Recommendation: A layered approach using K-means/Hierarchical for traffic profiling and DBSCAN/GMM for deep anomaly analysis can balance detection breadth and depth, enhancing adaptive DNS security systems.

8. CONCLUSIONS

This study has systematically evaluated four clustering algorithms: K-means, DBSCAN, Hierarchical Clustering, and GMM, for DNS anomaly detection across raw, PCA-reduced, and t-SNE-reduced data. Our results demonstrate that clustering performance is highly dependent on data representation, with t-SNE significantly enhancing anomaly separation for density-based methods like DBSCAN. K-means and Hierarchical Clustering proved effective for well-separated traffic patterns, while DBSCAN and GMM offered advantages for detecting irregular and overlapping threats, respectively.

The findings suggest that no single algorithm is universally optimal for DNS security. A hybrid, representation-aware approach, combining methods based on traffic characteristics and preprocessing strategy, could improve detection robustness. However, real-world deployment faces challenges, including the computational cost of t-SNE and GMM, sensitivity to parameter tuning, and the static nature of the dataset used.

Future work should explore incremental clustering for streaming DNS data, ensemble methods to leverage multiple algorithmic perspectives, and integration with threat intelligence for validation. Further investigation into lightweight dimensionality reduction techniques and real-time applicability will be essential for operational deployment.

REFERENCES

[1] Khaoula, R., Imane, M., Mohamed, M. (2024). Improving cyber defense with DNS Query clustering analysis. In 2024 11th International Conference on Wireless Networks and Mobile Communications (WINCOM), Leeds, United Kingdom, pp. 1-6. https://doi.org/10.1109/WINCOM62286.2024.10656538

[2] Khaoula, R., Mohamed, M. (2022). Improving intrusion detection using PCA and K-means clustering algorithm. In 2022 9th International Conference on Wireless Networks and Mobile Communications (WINCOM), Rabat, Morocco, pp. 1-5. https://doi.org/10.1109/WINCOM55661.2022.9966426

[3] Bilge, L., Balzarotti, D., Robertson, W., Kirda, E., Kruegel, C. (2012). Disclosure: Detecting botnet command and control servers through large-scale NetFlow analysis. In Proceedings of the 28th Annual Computer Security Applications Conference, Orlando,

Florida, USA, pp. 129-138. https://doi.org/10.1145/2420950.2420969

[4] Antonakakis, M., Perdisci, R., Lee, W., Vasiloglou, N., Dagon, D. (2011). Detecting malware domains at the upper DNS hierarchy. In Proceedings of the 20th USENIX conference on Security (SEC'11). USENIX Association, USA, 27.

[5] Kumari, R., Sheetanshu, Singh, M.K., Jha, R., Singh, N.K. (2016). Anomaly detection in network traffic using K-mean clustering. In 2016 3rd International Conference on Recent Advances in Information Technology (RAIT), Dhanbad, India, pp. 387-393. https://doi.org/10.1109/RAIT.2016.7507933

[6] Liu, Y., Xu, H.P., Yi, H., Lin, Z., Kang, J., Xia, W.Q. (2017). Network anomaly detection based on dynamic hierarchical clustering of cross domain data. In 2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), Prague, Czech Republic, pp. 200-204. https://doi.org/10.1109/QRS-C.2017.39

[7] Ichise, H., Jin, Y., Iida, K., Takai, Y. (2018). Detection and blocking of anomaly DNS Traffic by analyzing achieved NS record history. In 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Honolulu, HI, USA, pp. 1586-1590. https://doi.org/10.23919/APSIPA.2018.8659739

[8] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12: 2825-2830. https://dl.acm.org/doi/10.5555/1953048.2078195.

[9] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16: 321-357. https://doi.org/10.1613/jair.953

[10] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12: 2825-2830. https://dl.acm.org/doi/10.5555/1953048.2078195

[11] Jolliffe, I.T., Cadima, J. (2016). Principal component analysis: A review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374(2065): 20150202. https://doi.org/10.1098/rsta.2015.0202

[12] Maaten, L.V.D., Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9: 2579-2605.

[13] Khaoula, R., Mohamed, M. (2023). K-means-dist: A novel approach for enhanced cybersecurity clustering using combined distance metrics. In 2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM), Istanbul, Turkiye, pp. 1-6. https://doi.org/10.1109/WINCOM59760.2023.10322902

[14] Wu, J. (2012). Advances in K-means Clustering: A Data Mining Thinking. Springer Science & Business Media. https://doi.org/10.1007/978-3-642-29807-3

[15] Ester, M., Kriegel, H.P., Sander, J., Xu, X.W. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, Oregon, USA, pp. 226-231. https://dl.acm.org/doi/10.5555/3001460.3001507.

[16] Johnson, S.C. (1967). Hierarchical clustering schemes. Psychometrika, 32(3): 241-254. https://doi.org/10.1007/BF02289588

[17] Dempster, A.P., Laird, N.M., Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B (Methodological), 39(1): 1-22. https://doi.org/10.1111/j.2517-6161.1977.tb01600.x

[18] Rousseeuw, P.J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20: 53-65. https://doi.org/10.1016/0377-0427(87)90125-7

[19] Davies, D.L., Bouldin, D.W. (1979). A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1(2): 224-227. https://doi.org/10.1109/TPAMI.1979.4766909

[20] Bilge, L., Sen, S., Balzarotti, D., Kirda, E., Kruegel, C. (2014). Exposure: A passive DNS analysis service to detect and report malicious domains. ACM Transactions on Information and System Security (TISSEC), 16(4): 1-28. https://doi.org/10.1145/2584679