Discrete & Continuous Models
& Applied Computational Science
ISSN 2658-7149 (Online), 2658-4670 (Print)

2025, 33 (1) 27–45
http://journals.rudn.ru/miph

Research article

UDC 004.85
PACS 07.05.Tp,
DOI: 10.22363/2658-4670-2025-33-1-27-45 EDN: AFZDUC

Statistical and density-based clustering techniques in the context of anomaly detection in network systems: A comparative analysis

Aleksandr S. Baklashov[1, 2], Dmitry S. Kulyabov[1, 3]

[1] RUDN University, 6 Miklukho-Maklaya St, Moscow, 117198, Russian Federation

[2] V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, 65 Profsoyuznaya St, Moscow 117997, Russian Federation

[3] Joint Institute for Nuclear Research, 6 Joliot-Curie St, Dubna, 141980, Russian Federation

(received: November 25, 2024; revised: December 10, 2024; accepted: December 12, 2024)

Abstract. In the modern world, the volume of data stored electronically and transmitted over networks continues to grow rapidly. This trend increases the demand for the development of effective methods to protect information transmitted over networks as network traffic. Anomaly detection plays a crucial role in ensuring net security and safeguarding data against cyberattacks.

This study aims to review statistical and density-based clustering methods used for anomaly detection in network systems and to perform a comparative analysis based on a specific task. To achieve this goal, the authors analyzed existing approaches to anomaly detection using clustering methods. Various algorithms and clustering techniques applied within network environments were examined in this study.

The comparative analysis highlights the high effectiveness of clustering methods in detecting anomalies in network traffic. These findings support the recommendation to integrate such methods into intrusion detection systems to enhance information security levels.

The study identified common features, differences, strengths, and limitations of the different methods. The results offer practical insights for improving intrusion detection systems and strengthening data protection in network infrastructures.

Key words and phrases: intrusion detection systems, network systems, clustering methods

For citation: Baklashov, A. S., Kulyabov, D. S. Statistical and density-based clustering techniques in the context of anomaly detection in network systems: A comparative analysis. *Discrete and Continuous Models and Applied Computational Science* **33** (1), 27–45. doi: 10.22363/2658-4670-2025-33-1-27-45. edn: AFZDUC (2025).

1. Introduction

With the growing frequency and complexity of attacks targeting information systems [1], such as DDoS and data breaches, having a system to protect information from these types of attacks becomes a vital aspect of network design. Anomaly detection serves as a key element in ensuring the security of information systems, as anomalies in network traffic often indicate unauthorized access attempts or other forms of intrusion. That is why developing effective methods to detect deviations in network traffic behavior remains a crucial challenge.

In [2], the authors provided a comprehensive overview of methods, systems, and tools for anomaly detection in network traffic. That study placed particular attention on classifying the approaches available at the time, including clustering-based methods. However, due to its publication date, the review does not fully reflect recent advances in data processing and modern algorithms. The study in [2] also overlooked key aspects of density-based clustering methods such as DBSCAN, HDBSCAN, etc.

Therefore, actualization and in-depth study of clustering methods in the context of modern network traffic paradigms presents a highly relevant research direction. Recent approaches offer new opportunities to improve the accuracy and efficiency of anomaly detection.

Researchers now explore a wide range of anomaly detection techniques. For example, some researchers detect it using deep unfolding methods to reconstruct normal and anomalous data flows based on sparse and full-dimensional components [3]. The others use approaches such as Isolation Forest and autoencoders to detect anomalies [4].

Many researchers focus on neural network-based techniques. In [5], the authors investigate deep learning to address the issue of false positives in anomaly detection. At the same time, the others combine traditional approaches with machine learning techniques [6]. These methods have demonstrated strong performance in recognizing different data patterns, making them particularly effective for solving cybersecurity challenges.

This study proposes a clustering-based approach for network intrusion detection. The proposed method aims to serve as the first line of defense against network attacks within intrusion detection systems (IDS), which monitor events occurring within information systems or their individual components.

The objective of this study is to analyze existing clustering methods for anomaly detection and to perform a comparative assessment. To achieve this, the study analyzes and evaluates several clustering algorithms and summarizes their properties in a comparison table. An experimental section follows, presenting results for each method applied to a specific dataset.

The article includes several sections, each addressing a specific aspect of the research:

The section “Types of intrusion detection systems and anomaly detection methodology” defines IDS, outlines main IDS types, and introduces clustering methods.

The section “Methods and instruments” presents a detailed review and analysis of clustering techniques. Subsections cover partitioning methods (e.g., k-means, k-medoids), hierarchical clustering, and density-based clustering (DBSCAN, HDBSCAN, OPTICS). A summary table at the end of this section facilitates the comparison of these methods.

The section “Practical application of clustering methods in network anomaly detection systems” presents the comparative analysis results of six clustering algorithms, tested on a real dataset. This section includes results and interpretation of the metrics obtained for each clustering method experimentally.

The section “Results” presents the metrics obtained by application of the clustering methods to the specific dataset. Data is presented in the summary table, heatmap and text format.

The section “Discussion” summarizes the experimental findings and justifies the selection of the most suitable clustering method for network anomaly detection.

The section “Conclusion” outlines the main outcomes and discusses directions for future research.

2. Types of intrusion detection systems and anomaly detection methodology

An intrusion detection system (IDS) is software or hardware that analyzes network traffic or computer activity to identify potential unauthorized access attempts, attacks, or intrusions into computer systems or networks [7]. IDS detects a wide range of threats, including intrusions, viruses, worms, denial-of-service (DoS) attacks, and other anomalous behaviors, and alerts administrators about it, forcing them to enable timely defensive actions [8].

Researchers classify intrusion detection systems into two main types based on detection methods and the way of deployment [9, 10]:

1. Network-based intrusion detection systems (NIDS) analyze network traffic for anomalies by intercepting data at the network adapter level or via network devices such as switches and routers. NIDS can detect attacks before they reach the target system.
2. Host-based intrusion detection systems (HIDS) run on individual computers and monitor activity at the operating system level, including file system changes, registry modifications, event logs, and other system parameters. HIDS typically detect attacks targeting a specific host and may offer additional insights about system compromise.

This study focuses on clustering methods as a key tool for identifying anomalies in network-based intrusion detection systems (NIDS). Dividing network traffic into clusters that represent normal and abnormal behavior plays a critical role in designing effective NIDS and ranks among the most successful techniques for detecting network anomalies.

This clustering approach enhances both the accuracy and efficiency of IDS work.

The section titled “Methods and instruments” presents a comparison of six clustering algorithms: k-means, k-medoids, hierarchical clustering, DBSCAN, HDBSCAN, and OPTICS.

This analysis aims to further selection of the most appropriate method for anomaly detection in NIDS based on their performance, accuracy, strengths, and limitations.

3. Methods and instruments

This chapter presents a comparative analysis of clustering methods applicable to the stated problem (see Fig. 1).

The analysis focuses on three main types of clustering methods [11]:

1. Partitioning clustering;
2. Hierarchical clustering;
3. Density-based clustering.

To cluster network traffic into two categories this study evaluates the following methods:

- Two partitioning clustering methods: k-means and k-medoids;
- A hierarchical clustering method;
- Three density-based clustering methods: DBSCAN, HDBSCAN, and OPTICS.

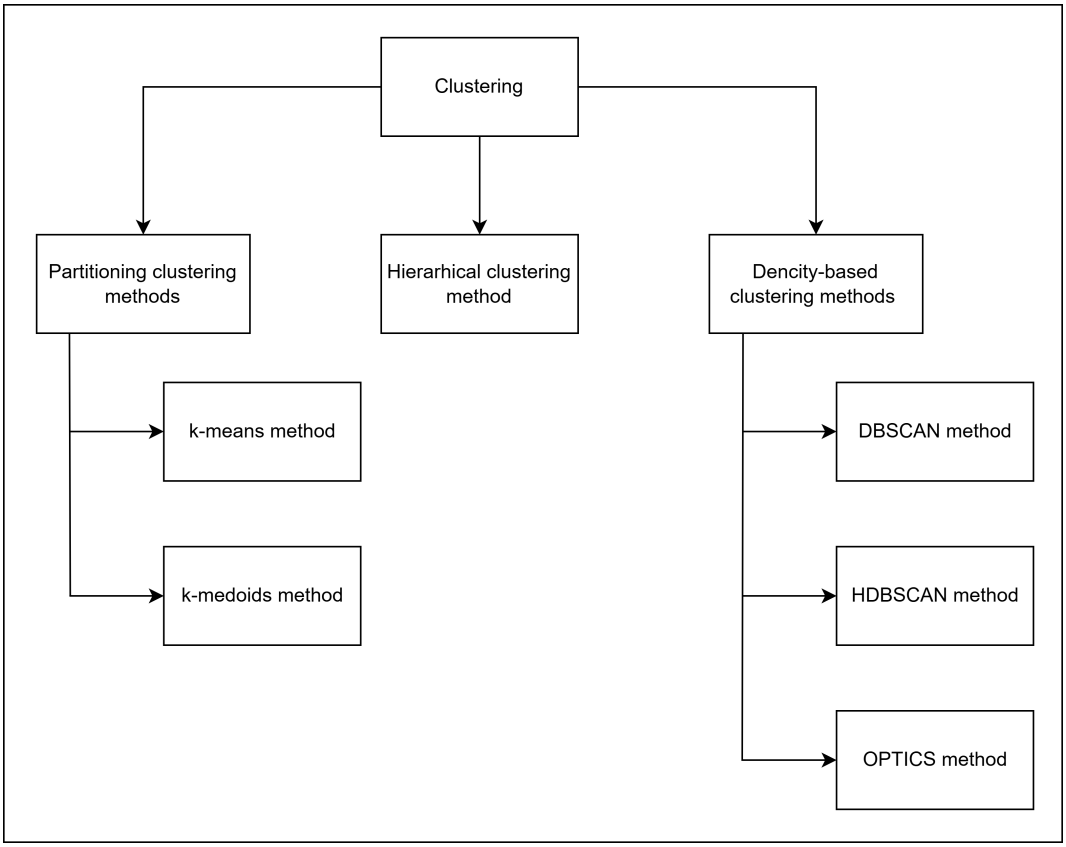

Figure 1. Clustering methods

3.1. Partitioning clustering methods

3.1.1. The k-means clustering method

The k-means method divides data into a predefined number of clusters k. The algorithm begins with selecting random centroids—mean points that represent each cluster. It then assigns each data point to the nearest centroid and after that recalculates centroids as the arithmetic mean of all points within the cluster. These steps repeat iteratively until convergence is achieved, after which the algorithm evaluates the clustering quality (see Fig. 2).

This method offers several advantages relevant to the current task, including simplicity, scalability, interpretability, and versatility.

However, it also introduces some limitations. The algorithm shows high sensitivity to initial centroid placement and outliers, which may distort the final results. Additionally, it does not guarantee an optimal solution [12].

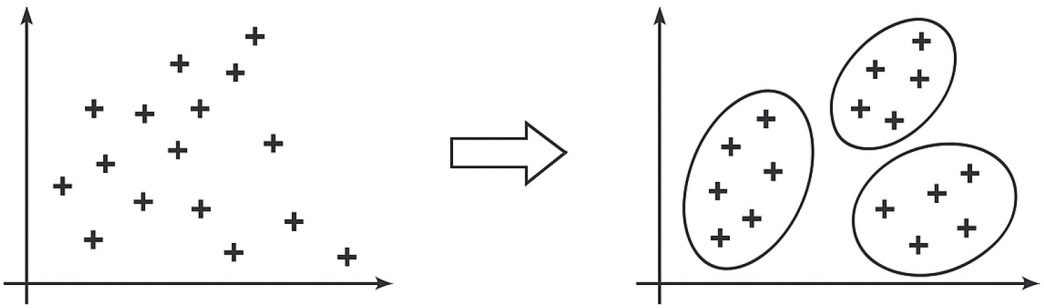

Figure 2. K-means method

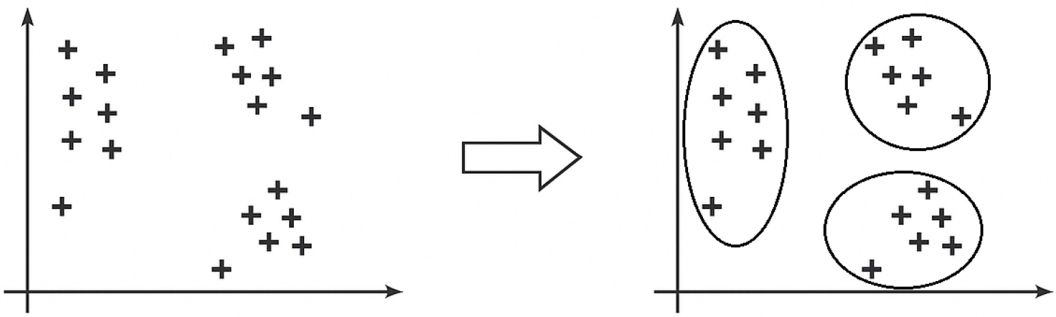

Figure 3. K-medoids method

3.1.2. The k-medoids clustering method

The k-medoids algorithm extends the k-means method by requiring that cluster centers (medoids) belong to the input data points. The algorithm begins by selecting k random points as initial medoids, where k denotes the predefined number of clusters. Then, it assigns each data point to the cluster by choosing the smallest distance from the medoid to the point, using a selected distance metric.

After assigning all points, the algorithm calculates the cost of the current clustering. It then attempts to replace one of the existing medoids with a non-medoid point and recalculates the cost. If the new cost exceeds or remains equal to the previous value, the algorithm reverts the change and stops. Otherwise, it accepts the new medoid and repeats the process from this step (see Fig. 3).

This method has several advantages. It executes a certain number of iterations. Compared to k-means, it provides more determined cluster centers since they correspond to actual data points. Also, this algorithm supports various distance metrics for cluster assignment.

However, k-medoids also introduces some drawbacks. It remains sensitive to the initial choice of medoids, and the randomness in selecting replacement candidates may lead to inconsistent results across different runs [13].

3.1.3. The hierarchical clustering method

The hierarchical clustering algorithm begins by treating each data point as an individual cluster. It then iteratively merges the closest clusters until all points belong to a single cluster. At the end of the

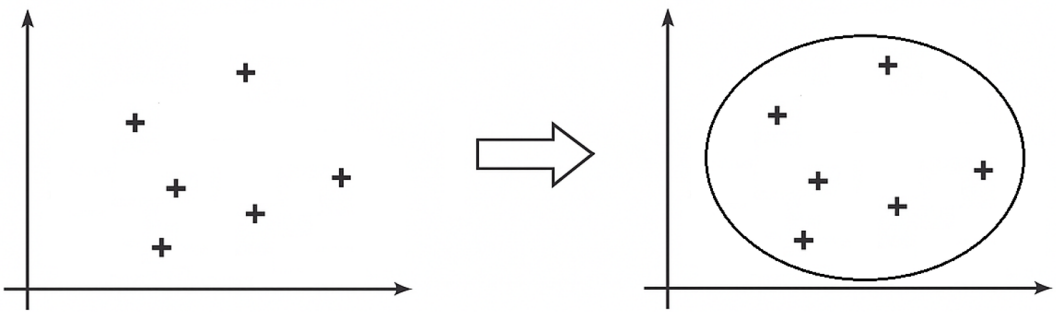

Figure 4. Hierarchical method

process, the algorithm constructs a dendrogram that illustrates the hierarchy of cluster merging and allows the selection of an optimal number of clusters (see Fig. 4).

This method provides several advantages, including a high level of interpretability and versatility.

However, it also has some drawbacks. The algorithm suffers from high computational complexity and is sensitive to the choice of distance metric [12].

3.2. Density-based clustering methods

3.2.1. The DBSCAN clustering method

The DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm identifies clusters by locating areas of high point density. The process begins by selecting a random point and defining its neighborhood. If the number of neighboring points exceeds a predefined threshold MinPts (minimum number of neighbors), the point becomes a core point.

The algorithm then forms clusters by uniting core points and cluster-related boundary points. Points that do not belong to any cluster and are not part of dense areas are treated as outliers or noise (see Fig. 5).

This method offers several advantages relevant to anomaly detection tasks, such as strong performance on large datasets.

However, DBSCAN has some limitations. It shows sensitivity to the choice of parameters and struggles to cluster data with varying densities or scales. In addition, the computational cost increases with large values of the input parameters [12].

3.2.2. The HDBSCAN clustering method

The HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) algorithm extends DBSCAN by incorporating a hierarchical clustering approach. After running the DBSCAN core procedure, HDBSCAN iteratively merges clusters based on the distances between them, forming a hierarchy of density-connected areas (see Fig. 6).

Overall, HDBSCAN offers a powerful clustering technique with several advantages over the classical DBSCAN. These include automatic determination of the number of clusters and greater robustness to parameter selection.

At the same time, applying HDBSCAN may require increased computational resources and can encounter limitations when dealing with datasets that exhibit highly irregular or complex structures [14].

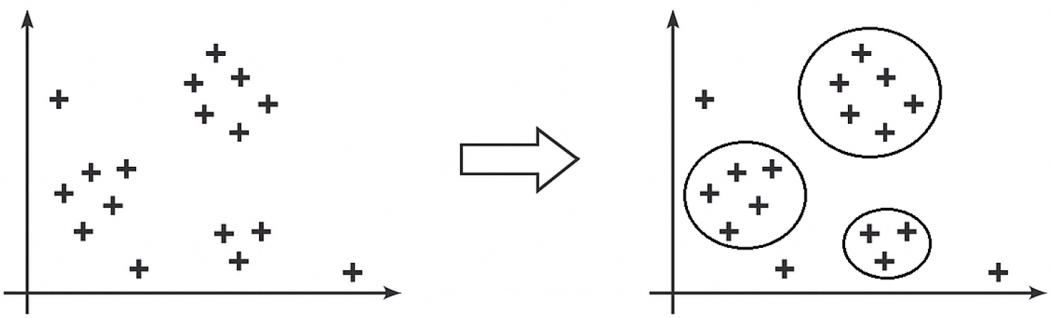

Figure 5. DBSCAN

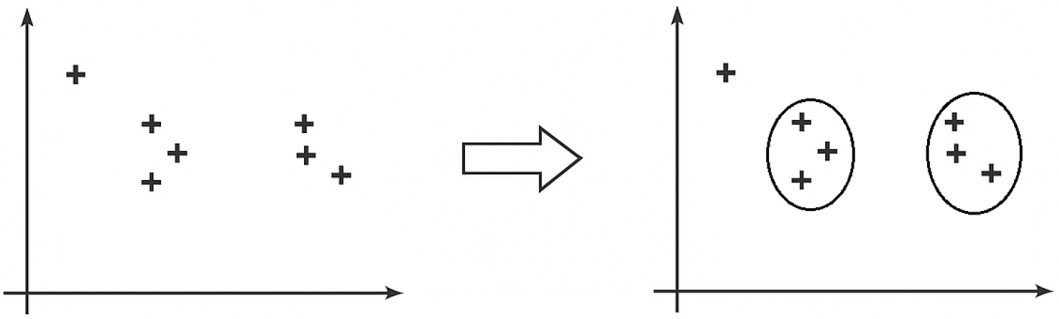

Figure 6. HDBSCAN

3.2.3. The OPTICS clustering method

The OPTICS (Ordering Points To Identify the Clustering Structure) algorithm builds upon the ideas introduced in DBSCAN, allowing the detection of clusters with varying densities by ordering points and dynamically selecting parameters. The algorithm requires setting a minimum number of neighbors (MinPts) and optionally an ε-radius. Compared to DBSCAN, OPTICS is less sensitive to the exact choice of these parameters.

After setting the parameters, the algorithm identifies the neighbors of each data point within the ε-radius and calculates the density of each point based on the number of its neighbors. OPTICS then generates an ordered list of points by iteratively traversing the dataset, starting from an random unvisited point and proceeding through its neighbors. This list forms the basis for constructing a dendrogram and selecting a density threshold that separates clusters [15] (see Fig. 7).

OPTICS offers several advantages, including the ability to detect clusters of arbitrary shape, robustness to noise and outliers, and no need to specify the number of clusters in advance. It also supports a variety of distance metrics for cluster formation.

However, the algorithm still requires careful tuning of ε and MinPts, and its computational complexity becomes significant on large datasets [14].

3.3. Comparative table

Let's make a summary table of the data (Table 1).

Table 1

Comparison of Clustering Methods

Method	Core Principle	Parameters	Outliers	Advantages	Disadvantages
k-means	Finds cluster centers and minimizes deviation from points.	Number of clusters	Sensitive to outliers	Simplicity, scalability, interpretability, versatility	Sensitive to initial conditions and outliers, no guarantee of optimality
k-medoids	Selects and updates medoids	Number of clusters	Less sensitive than k-means	Clusters defined by actual data points, supports various distance metrics	Sensitive to initial medoid selection, random replacement
Hierarchical Clustering	Merges and splits clusters based on inter-point distances	Number of clusters (optional)	Outliers affect hierarchy formation	High interpretability, versatility	High computational complexity, sensitive to distance metric
DBSCAN	Separates high- and low-density regions	ε, MinPts	Isolates outliers into separate clusters	Good performance on large datasets	Sensitive to parameters, computational complexity
HDBSCAN	Builds hierarchy of density-based clusters	*min_cluster_size*, *cluster_selection_epsilon*	Detects and ignores outliers	Automatic cluster number selection, robust to parameters	Higher computational demands than DBSCAN
OPTICS	Estimates density and performs ordered traversal	ε, MinPts	Robust to noise and outliers	Noise resilience, various distance metrics, arbitrary shape detection	Sensitive to parameters, computationally intensive

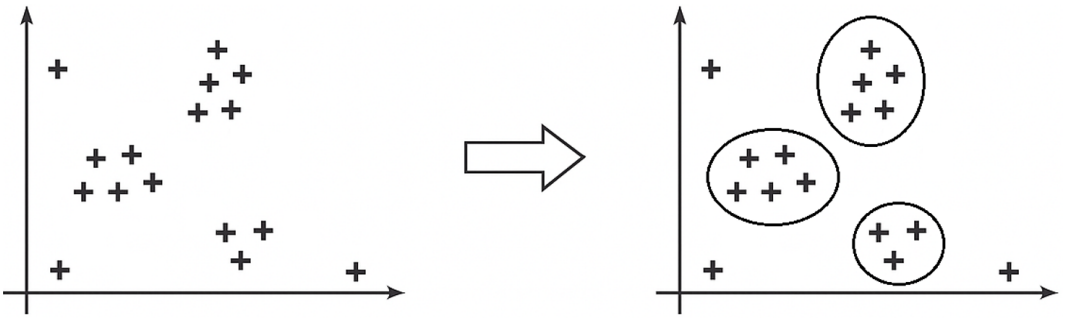

Figure 7. OPTICS

4. Practical application of clustering methods in network anomaly detection systems

This chapter presents a comparative analysis of six clustering methods described in the previous chapter that is conducted through experimental application to the task of separating network traffic into two categories: normal and anomalous. The analysis is based on the NSL-KDD dataset, which is an improved version of the well-known KDD Cup 1999 dataset [16, 17]. The NSL-KDD dataset contains 41 attributes and includes a label indicating whether the connection is normal or an anomaly [18]. The evaluation metrics and visualizations obtained from the experiments allow assessing the effectiveness, advantages, and limitations of each method, as well as identifying the most suitable approach for a Network Intrusion Detection System (NIDS).

4.1. Experiment description

The experiment was conducted using the NSL-KDD dataset, which contains both numerical and categorical features of network traffic, such as connection duration, number of bytes sent and received, as well as categorical features like protocol type and flags. The data underwent preprocessing: numerical features were normalized using StandardScaler, and categorical features were encoded using OneHotEncoder. Dimensionality reduction was performed using PCA, retaining 10 principal components to accelerate computation and facilitate analysis [19].

The clustering procedure was used to partition the data into groups corresponding to normal and anomalous traffic. The quality of clustering was evaluated using the following metrics: precision, recall, F1-score, execution time, silhouette coefficient, Calinski-Harabasz index (*ch_index*), number of clusters (*n_clusters*), noise ratio, cluster purity, and Precision-Recall AUC ($PR - AUC$). A cluster was classified as anomalous if more than 50% of the traffic within it was anomalous.

4.2. The analysis of clustering methods

4.2.1. K-means

In the K-means method, the number of clusters was fixed at $n_{\text{clusters}} = 20$. As shown in Figure 8, the clusters exhibit clear separation; however, their sizes vary significantly, with a few large clusters dominating the distribution. The algorithm demonstrates sensitivity to outliers, which can distort centroid positions and reduce clustering quality, as it does not isolate noise into a separate category.

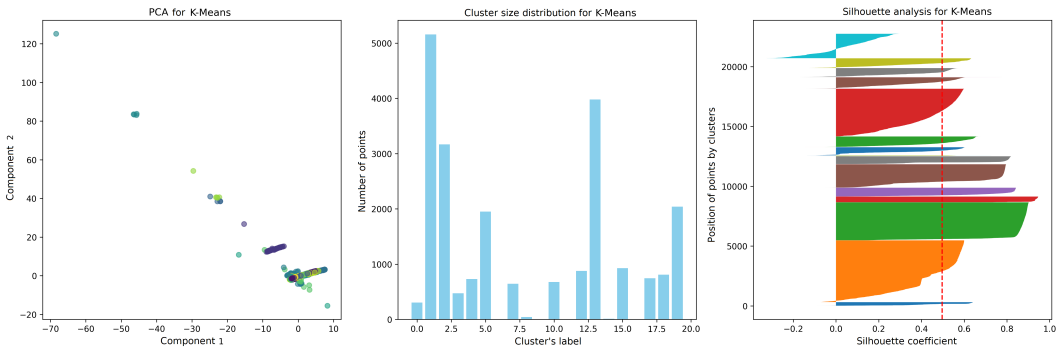

Figure 8. K-means—results

To evaluate clustering performance, the experiment used F1-score, execution time, and the silhouette coefficient. The F1-score reached 0.858, indicating a balanced correlation between precision and recall. The algorithm achieved exceptional computational efficiency, with an execution time of just 0.032 seconds. A silhouette coefficient of 0.496 suggests a tolerable level of compactness and separation among clusters.

The Calinski–Harabasz index further supports the presence of well-defined cluster structures. This method proves effective for processing large-scale datasets and performs well when anomalies form dense groups. Also, it demonstrates high speed and ease of implementation. However, its inability to explicitly handle noise stints its applicability in scenarios with significant outlier presence.

4.2.2. K-medoids

In the experiment with the k-medoids method, the number of clusters was set to 20, and the Manhattan distance metric was used for evaluating distances between objects. Unlike k-means, this method relies on medoids instead of centroids, making it less sensitive to outliers. The clustering visualizations (see Fig. 9) show a more balanced cluster size distribution, although the compactness becomes lower due to the heterogeneous data density.

The evaluation used F1-score, execution time, cluster purity, and silhouette coefficient. The F1-score reached 0.883, surpassing k-means and indicating higher clustering quality. However, the execution time amounted to 28.499 seconds, highlighting a significant drawback in computational efficiency. Cluster purity reached 0.868, reflecting a strong correspondence between the formed clusters and the actual data labels. However, the method's sensitivity to the initial medoid selection introduced variability, resulting in a low silhouette value of 0.221.

This approach achieves a strong balance between precision and anomaly coverage, as evidenced by high F1-score and purity. It proves more robust than k-means in noisy environments, although it similarly lacks explicit mechanisms for isolating noise points. The method's resource intensity must also be taken into account when applying it to large-scale data.

4.2.3. Hierarchical clustering

Hierarchical clustering begins by treating each data point as an individual cluster and gradually merges them based on distances between objects using the Ward linkage method until the desired number of clusters (10) is formed. As illustrated in Fig. 10, the resulting clusters exhibit a clear

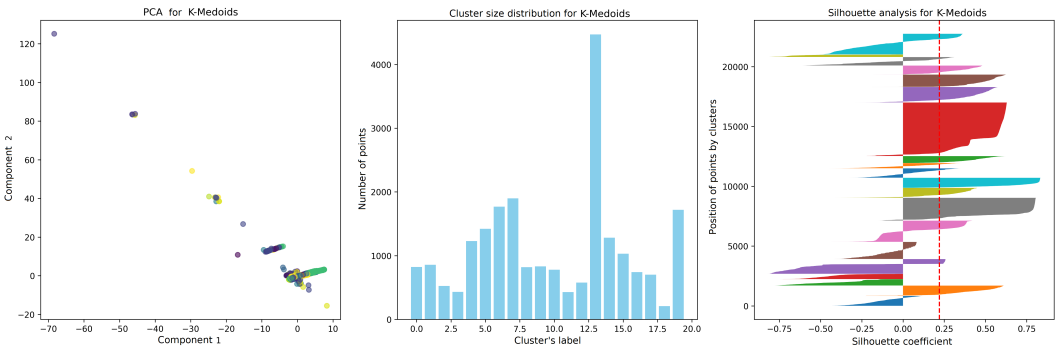

Figure 9. K-medoids—results

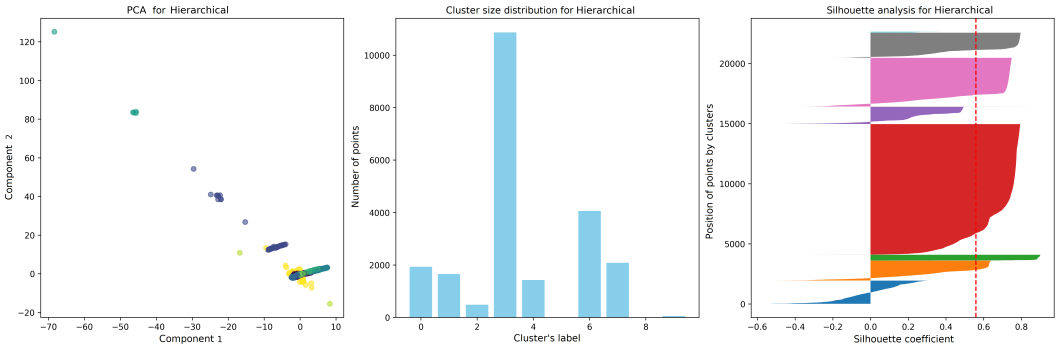

Figure 10. Hierarchical clustering—results

hierarchical structure; however, the cluster size distribution remains imbalanced, with one dominant large cluster.

Evaluation used the following metrics: F1-score, precision, Calinski–Harabasz index, silhouette coefficient, and execution time. The F1-score reached 0.824, indicating a reasonable balance between precision and recall. Precision achieved a high value of 0.939, reflecting strong classification accuracy. The Calinski–Harabasz index was 10159.3, suggesting excellent cluster separability and compactness. The silhouette coefficient was 0.559, which confirms passable intra-cluster cohesion and inter-cluster separation. However, the method required 17.538 seconds to complete, limiting its suitability for time-sensitive applications.

Hierarchical clustering allows us to identify internal data structure and relationships between clusters. Despite its interpretability and clustering effectiveness, the high computational complexity constrains its applicability to large-scale datasets or real-time systems.

4.2.4. DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) identifies clusters based on data density using the predefined parameters $\epsilon = 0.3$ (radius) and $min\_samples = 20$ (minimum number of neighbors). With these settings, DBSCAN detected a noise fraction of 0.14 relative to the total number of data points. As shown in Fig. 11, the resulting cluster size distribution is unbalanced,

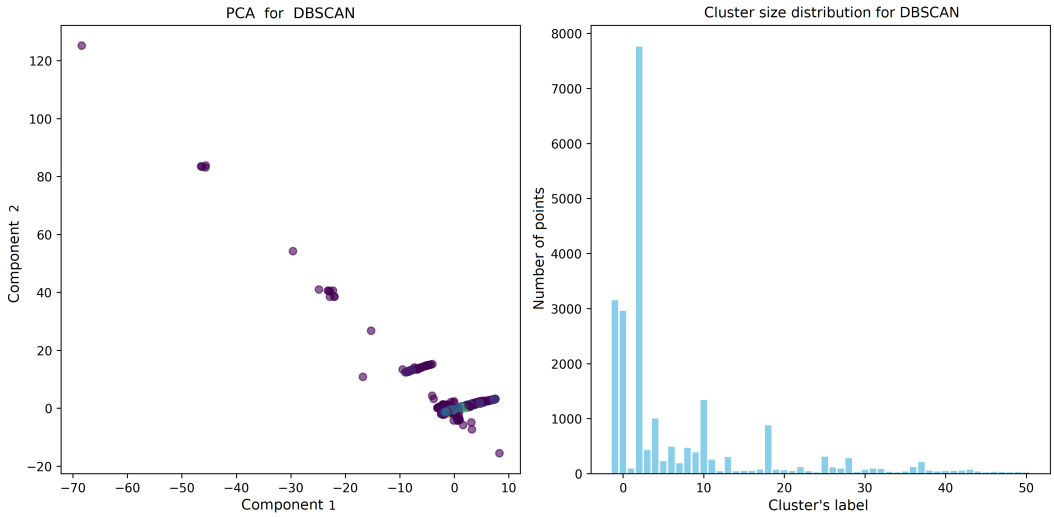

Figure 11. DBSCAN—results

with a single dominant cluster. Nevertheless, the remaining clusters maintain a moderate degree of compactness.

The clustering quality was assessed using F1-score, precision, execution time, and noise ratio. The F1-score reached 0.889, indicating high overall clustering performance. Precision was 0.86, suggesting an acceptable level of errors in the distribution of objects into clusters. Execution time amounted to 2.536 seconds, demonstrating good computational efficiency. However, the method's performance is highly sensitive to parameter tuning, which complicates its practical application. The silhouette coefficient is not a reliable measure for density-based methods and thus was not used as a primary evaluation metric.

DBSCAN offers an effective balance between computational speed and clustering quality. Yet, the requirement for careful parameter selection imposes additional complexity in real-world deployments.

4.2.5. HDBSCAN

HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) extends DBSCAN by building a hierarchy of clusters with automatic determination of the optimal number of clusters, which amounted to 243 in this case. The method employs parameters *min_cluster_size* = 5 and *cluster_selection_epsilon* = 0.08. Unlike DBSCAN, HDBSCAN replaces the fixed radius ϵ with a minimum cluster size, thereby offering greater flexibility in clustering process. With these parameters, the method identified noise accounting for 0.131 of the total number of instances.

Figure 12 shows that the cluster size distribution is relatively balanced, with several large groups and a moderate number of smaller ones. However, cluster compactness remained limited due to heterogeneous density across groups.

The evaluation relied on F1-score, recall, PR-AUC, and execution time. The F1-score reached 0.924, the highest among all methods considered. Recall was 0.988, reflecting excellent sensitivity in anomaly detection. The PR-AUC value of 0.932 further confirmed outstanding overall performance. Execution time was 4.194 seconds, acceptable for mid-scale problems.

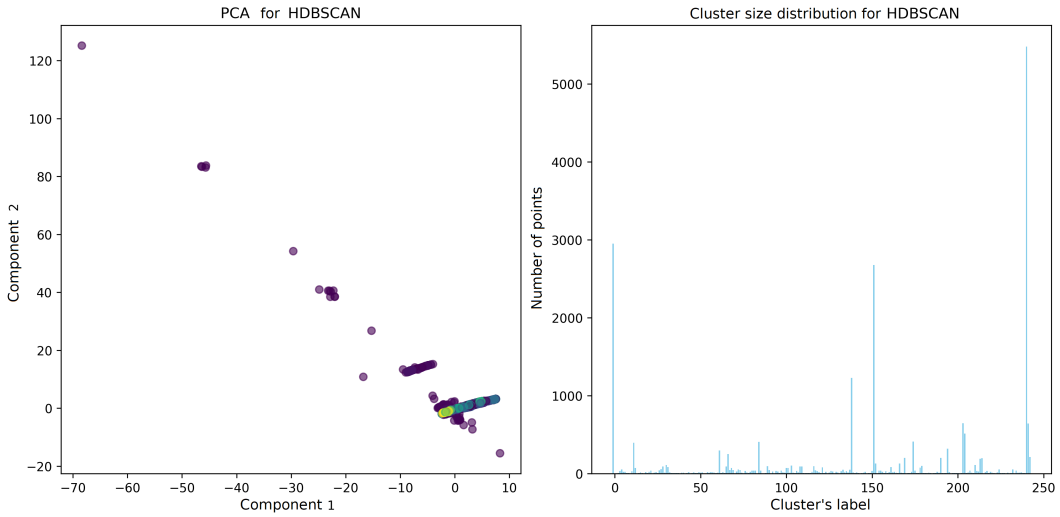

Figure 12. HDBSCAN—results

The combination of high recall and F1-score demonstrates the method's ability to detect nearly all anomalies, while the presence of a noise cluster assists outlier identification. Overall, HDBSCAN provides strong performance in clustering tasks involving heterogeneous densities and noisy data.

4.2.6. OPTICS

OPTICS (Ordering Points To Identify the Clustering Structure) is a density-based clustering method that utilizes the parameters *min_samples* $= 4$ and $\xi = 0.01$, resulting in the formation of 2066 clusters and a high noise level with a noise ratio of 0.386. As shown in Figure 13, the cluster size distribution is imbalanced, with one dominant large cluster and numerous small groups. Despite this, the main cluster maintains high compactness.

Clustering quality was assessed using F1-score, recall, execution time, and the silhouette coefficient. The F1-score reached 0.853, indicating a satisfactory result. Recall achieved 0.987, highlighting the method's strong anomaly detection capability. However, the execution time amounted to 41 seconds, which significantly reduces the method's feasibility for real-time applications. Furthermore, the high noise proportion of 0.39 emphasizes its sensitivity to data structure.

OPTICS effectively identifies clusters of arbitrary shapes and exhibits robustness to noise. Nevertheless, its computational complexity and sensitivity to parameter tuning limitations for deployment in performance-critical environments.

5. Results

The comparative analysis of clustering methods was conducted based on evaluation metrics presented in the heatmap (Figure 14) and the clustering visualizations. Table 2 summarizes the key performance indicators across all methods.

For the task of distinguishing between normal and anomalous traffic, it is essential to balance precision and recall, which is reflected in the F1-score, while also considering computational

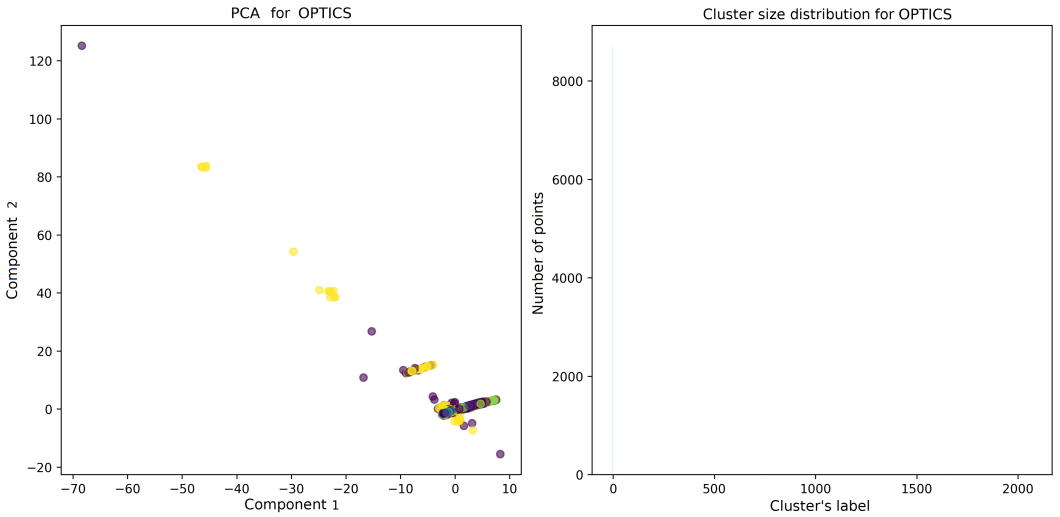

Figure 13. OPTICS—results

Table 2

Comparison of clustering metrics

	HDBSCAN	DBSCAN	K-Medoids	K-Means	OPTICS	Hierarchical
precision	0,868	0,860	0,895	0,896	0,751	0,939
recall	0,988	0,920	0,870	0,823	0,987	0,734
F1	0,924	0,889	0,883	0,858	0,853	0,824
time	4,194	2,536	28,499	0,032	41,203	17,538
silhouette	−0, 088	0,192	0,221	0,496	−0, 038	0,559
ch_index	229,125	829,498	1396,012	12286,905	15,215	10159,276
n_clusters	243	51	20	20	2066	10
noise_ratio	0,131	0,140	0	0	0,386	0
purity	0,846	0,774	0,868	0,845	0,601	0,822
pr_auc	0,932	0,913	0,783	0,692	0,873	0,912

efficiency, robustness to noise, and ease of parameter setting. Among the evaluated methods, HDBSCAN achieved the highest F1-score (0.92), followed by DBSCAN (0.89), K-Medoids (0.88), K-Means (0.86), OPTICS (0.85), and Hierarchical Clustering (0.82). In terms of execution time, K-Means (0.03 seconds) and DBSCAN (2.5 seconds) demonstrated the fastest performance, whereas OPTICS (41 seconds), K-Medoids (28 seconds), and Hierarchical Clustering (17 seconds) required significantly more time. HDBSCAN ranked second in speed with a runtime of 4 seconds. Regarding noise handling, HDBSCAN (0.13) and DBSCAN (0.14) effectively isolated outliers, while OPTICS (0.39) marked a substantial portion of data as noise. K-Means, K-Medoids, and Hierarchical Clustering

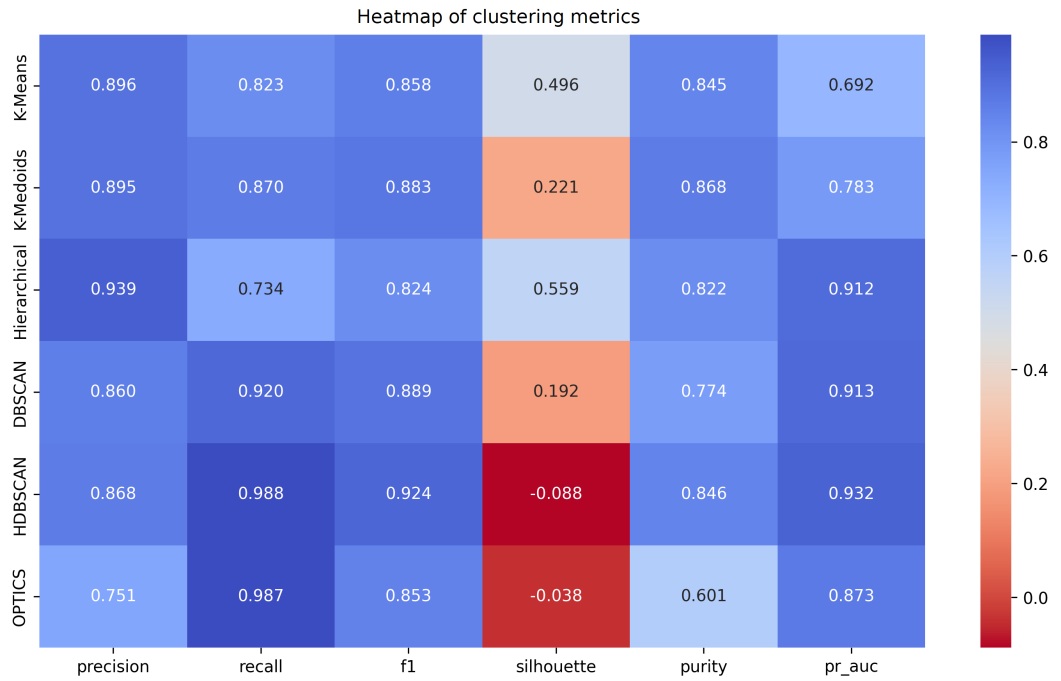

Figure 14. Heatmap

do not separate noise explicitly. In terms of configuration simplicity, K-Means, K-Medoids, and Hierarchical Clustering are more straightforward, requiring only the number of clusters as input, whereas DBSCAN, HDBSCAN, and OPTICS involve more complex parameter setting.

Thus, HDBSCAN represents the most suitable method for this task: it identifies nearly all anomalies (recall of 0.99), yields a strong F1-score (0.92), and effectively isolates suspicious points via noise detection. However, if execution time is a critical constraint, K-Means offers a viable alternative, achieving a runtime of just 0.03 seconds and high precision (0.94), albeit with lower recall (0.69).

6. Discussion

The analysis demonstrated that the most effective clustering methods for separating network traffic into normal and anomalous categories are HDBSCAN and K-Means. These methods achieved high F1-scores in the range of 0.85-0.92 and exhibited low execution times (0.03-4 seconds), making them suitable for use in network intrusion detection systems (NIDS). K-Means stands out due to its simplicity and speed, but its sensitivity to initial conditions can result in instability. K-Medoids is more robust to outliers but suffers from slow performance on large datasets. Hierarchical clustering offers a high degree of interpretability, yet its computational complexity limits its scalability for large-scale applications.

Density-based methods such as DBSCAN and OPTICS identified a substantial proportion of data as noise, which complicates their direct application to binary classification tasks in network traffic analysis. In contrast, HDBSCAN achieved an optimal balance between clustering quality (F1-score of 0.923) and anomaly detection capability (recall of 0.988), while maintaining a rapid runtime. These

results make HDBSCAN the most suitable clustering method for practical deployment in network intrusion detection systems.

In conclusion, for practical applications, HDBSCAN and K-Means are recommended, depending on the specific requirements for computational efficiency and robustness to noise.

7. Conclusion

This study conducted a comparative analysis of six clustering methods for the task of separating network traffic into normal and anomalous categories. Experimental results indicate that HDBSCAN is the most suitable method according to several criteria, including precision, recall, F1-score, and the ability to detect outliers. HDBSCAN achieved the highest F1-score (0.92) and recall (0.988), effectively identifying nearly all anomalies and handling noise robustly.

At the same time, the K-Means algorithm, having much lower computational complexity and execution time (0.03 seconds), is also an effective solution for time-sensitive applications. However, its sensitivity to the initial centroid selection and initial conditions, as well as its inability to detect noise stint its applicability. K-Medoids offers better robustness to outliers compared to K-Means, but its computational cost makes it less attractive for large-scale datasets.

Hierarchical clustering, OPTICS, and DBSCAN exhibit advantages such as the ability to detect clusters of arbitrary shape and considering noise. Nevertheless, their high computational complexity and sensitivity to parameter selection restrict their use in scenarios requiring rapid analysis of large datasets.

Future research directions include the development of hybrid approaches that combine the high accuracy and recall of density-based methods (e.g., HDBSCAN) with the speed and simplicity of partitioning-based methods (e.g., K-Means). Moreover, integrating clustering techniques with neural network architectures may further enhance the overall performance of anomaly detection systems.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used “Conceptualization, Aleksandr S. Baklashov and Dmitry S. Kulyabov; methodology, Dmitry S. Kulyabov; software, Aleksandr S. Baklashov; validation, Aleksandr S. Baklashov and Dmitry S. Kulyabov; formal analysis, Dmitry S. Kulyabov; investigation, Aleksandr S. Baklashov; resources, Dmitry S. Kulyabov; data curation, Aleksandr S. Baklashov; writing—original draft preparation, Aleksandr S. Baklashov; writing—review and editing, Aleksandr S. Baklashov and Dmitry S. Kulyabov; visualization, Aleksandr S. Baklashov; supervision, Dmitry S. Kulyabov.; project administration, Dmitry S. Kulyabov. All authors have read and agreed to the published version of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data sharing is not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kosmacheva, I., Davidyuk, N., Belov, S., Kuchin, Y. S., Kvyatkovskaya, Y., Rudenko, M. & Lobeyko, V. I. Predicting of cyber attacks on critical information infrastructure. *Journal of Physics: Conference Series* **2091** (2021).
2. Bhuyan, M. H., Bhattacharyya, D. K. & Kalita, J. K. Network Anomaly Detection: Methods, Systems and Tools. *IEEE Communications Surveys & Tutorials* **16,** 303–336 (2014).
3. Schynol, L. & Pesavento, M. *Deep Unrolling for Anomaly Detection in Network Flows* in (Dec. 2023), 61–65. doi:10.1109/CAMSAP58249.2023.10403513.

4. Maheswari, G., Vinith, A., Sathyanarayanan, A. S., Sowmi, S. M. & Sambath, M. An Ensemble Framework for Network Anomaly Detection Using Isolation Forest and Autoencoders. *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS),* 1–6 (2024).
5. Olateju, O., Okon, S., Igwenagu, U., Salami, A., Oladoyinbo, T. & Olaniyi, O. Combating the Challenges of False Positives in AI-Driven Anomaly Detection Systems and Enhancing Data Security in the Cloud. *Asian Journal of Research in Computer Science* **17,** 264–292. doi:10.9734/ajrcos/2024/v17i6472 (June 2024).
6. Lavanya, A. & Sekar, D. Traditional Methods and Machine Learning for Anomaly Detection in Self-Organizing Networks. *International Journal of Scientific Research in Science, Engineering and Technology* **10,** 352–360. doi:10.32628/IJSRSET2310662 (Dec. 2023).
7. Sheela, S. N., Prasad, E., Srinath, M. V. & Basha, M. S. Intrusion Detection Systems, Tools and Techniques – An Overview. *Indian journal of science and technology* **8** (2015).
8. Al-Ghamdi, M. An Assessment of Intrusion Detection System (IDS) and Data-Set Overview: A Comprehensive Review of Recent Works. *Journal of Scientific Research and Development* **5,** 979–982 (Feb. 2021).
9. Rozendaal, K., Mailewa, A. & Dissanayake Mohottalalage, T. Neural Network Assisted IDS/IPS: An Overview of Implementations, Benefits, and Drawbacks. *International Journal of Computer Applications* **184,** 21–28. doi:10.5120/ijca2022922098 (May 2022).
10. Satilmiş, H., Akleylek, S. & Tok, Z. A Systematic Literature Review on Host-Based Intrusion Detection Systems. *IEEE Access* **PP,** 1–1. doi:10.1109/ACCESS.2024.3367004 (Jan. 2024).
11. Mahfuz, N. M., Yusoff, M. & Ahmad, Z. Review of single clustering methods. *IAES International Journal of Artificial Intelligence* **8,** 221–227 (2019).
12. Burkov, A. *Machine learning engineering* (True Positive, Sept. 2020).
13. Park, H.-S. & Jun, C.-H. A simple and fast algorithm for K-medoids clustering. *Expert Systems with Applications* **36,** 3336–3341. doi:10.1016/j.eswa.2008.01.039 (2009).
14. Campello, R., Kröger, P., Sander, J. & Zimek, A. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **10.** doi:10.1002/widm.1343 (Oct. 2019).
15. Ankerst, M., Breunig, M. M., Kriegel, H.-P. & Sander, J. OPTICS: ordering points to identify the clustering structure. *SIGMOD Rec.* **28,** 49–60. doi:10.1145/304181.304187 (June 1999).
16. Sahli, Y. Comparison of the NSL-KDD dataset and its predecessor the KDD Cup '99 dataset. *International Journal of Scientific Research and Management* **10,** 832–839. doi:10.18535/ijsrm/v10i4.ec05 (Apr. 2022).
17. L.Dhanabal & Shantharajah, D. S. P. *A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms* in. **4** (June 2015), 446–452.
18. Kunhare, N. & Tiwari, R. *Study of the Attributes using Four Class Labels on KDD99 and NSL-KDD Datasets with Machine Learning Techniques* in (Nov. 2018), 127–131. doi:10.1109/CSNT.2018.8820244.
19. Gorban, A., Kégl, B., Wunsch, D. & Zinovyev, A. *Principal Manifolds for Data Visualisation and Dimension Reduction, LNCSE 58* 338 pp. (Jan. 2008).

Information about the authors

Aleksandr S. Baklashov—Master's degree student Department of Probability Theory and Cybersecurity of RUDN University; Mathematician, V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences (e-mail: 1132239133@pfur.ru, phone: +7(977)4573881, ORCID: 0009-0000-9046-3225, ResearcherID: KLZ-4503-2024)

Dmitry S. Kulyabov—Professor, Doctor of Sciences in Physics and Mathematics, Professor of Department of Probability Theory and Cyber Security of RUDN University; Senior Researcher of Laboratory of Information Technologies, Joint Institute for Nuclear Research (e-mail: kulyabov_ds@pfur.ru, phone: +7(495)9520250, ORCID: 0000-0002-0877-7063, ResearcherID: I-3183-2013, Scopus Author ID: 35194130800)

УДК 004.85
PACS 07.05.Tp,
DOI: 10.22363/2658-4670-2025-33-1-27-45 EDN: AFZDUC

Статистические и плотностные методы кластеризации в задачах обнаружения аномалий сетевых систем: сравнительный анализ

А. С. Баклашов[1, 2], Д. С. Кулябов[1, 3]

[1] Российский университет дружбы народов, ул. Миклухо-Маклая, д. 6, Москва, 117198, Российская Федерация

[2] Институт проблем управления им. В. А. Трапезникова Российской академии наук, ул. Профсоюзная, д. 65, Москва, 117997, Российская Федерация

[3] Объединённый институт ядерных исследований, ул. Жолио-Кюри, д. 6, Дубна, 141980, Российская Федерация

Аннотация. В современном мире количество данных, хранящихся в электронном виде и передающихся по сети, непрерывно растёт. Это создаёт потребность в разработке эффективных методов защиты информации, передающейся в виде сетевого трафика. Выявление аномалий играет ключевую роль в обеспечении безопасности сетей и защите информации от кибератак.

Цель данной работы заключается в проведении обзора статистических и плотностных методов кластеризации, применяемых для определения аномалий в сетевых системах, и проведении их сравнительного анализа на конкретной задаче.

Для достижения цели исследования использовались методы анализа существующих подходов к обнаружению аномалий с помощью методов кластеризации. В исследовании рассматривались различные алгоритмы и методы кластеризации, применяемые в сетевых системах.

Результаты проведённого сравнительного анализа продемонстрировали высокую эффективность методов кластеризации в задачах обнаружения аномалий сетевого трафика, что позволяет рекомендовать их для интеграции в системы обнаружения вторжений с целью повышения уровня информационной безопасности.

Был проведён сравнительный анализ различных методов, выявлены их общие черты, различия, достоинства и недостатки.

Полученные результаты могут быть использованы для улучшения систем обнаружения вторжений и повышения уровня защиты информации в сетевых системах.

Ключевые слова: системы обнаружения вторжений, сетевые системы, методы кластеризации