Journal homepage: https://itce.vn.ua/en

Information Technologies and Computer Engineering **Vol. 23, No. 2. 2026**

Article's History: Received: 13.02.2026 Revised: 29.05.2026
Accepted: 25.06.2026 Published: 03.08.2026

ISSN 1999-9941; e-ISSN 2078-6387
UDC 004.4
DOI: 10.31649/vitce/2.2026.24

Software implementation of a network traffic monitoring system based on cluster analysis

Samira Budzhelida*
Student
National University "Zaporizhzhia Polytechnic"
69063, 64 Universytetska Str., Zaporizhia, Ukraine
https://orcid.org/0009-0002-4837-1022

Valeriy Dubrovin
PhD in Technical Sciences, Professor
National University "Zaporizhzhia Polytechnic"
69063, 64 Universytetska Str., Zaporizhia, Ukraine
https://orcid.org/0000-0002-0848-8202

Larysa Deineha
Senior Lecturer
National University "Zaporizhzhia Polytechnic"
69063, 64 Universytetska Str., Zaporizhia, Ukraine
https://orcid.org/0000-0003-0304-4327

Abstract. The growth in the volume and heterogeneity of network traffic complicates the timely detection of attacks, whereas signature-based approaches do not cover new or modified threats. Interpretable and reproducible methods for analysing data flows are needed, and they must be suitable for integration into monitoring systems. The purpose of the study was to develop and implement a monitoring system that separates normal connections from anomalous ones without prior data labelling and provides transparent decision-making criteria. The research methodology was based on the standardisation of network traffic features and the use of the unsupervised k-means clustering algorithm, followed by anomaly detection through deviations from centroids. On a synthetic set of events with a small proportion of violations, the system consistently formed two compact clusters corresponding to typical and atypical behaviour. The centroid of normal traffic was characterised by lower values for the volumes of transmitted and received data and lower connection activity; the centroid of anomalous traffic had substantially higher values across all features. The combined post-clustering rule reduced false positives that arise during legitimate large transfers, for example, backups, while maintaining a high proportion of correctly detected rare events. Comparative experiments demonstrated comparable or higher precision and recall than classical outlier detection approaches, along with a stable millisecond-level processing time for one thousand records. The sensitivity analysis confirmed robustness to the choice of distance threshold and preprocessing parameters. The experimental sample contained 1,000 records, of which 980 corresponded to normal network traffic and 20 corresponded to anomalous events (2%). Clustering formed two clusters, with the smaller cluster grouping 20 records that corresponded to atypical network behaviour. The proposed approach does not require reference labels, scales easily, provides transparent explanations through centroids and distances, is suitable for batch and stream processing, and can serve as a basic component in production anomaly detection pipelines

Keywords: anomalous connection detection; feature standardisation; centroid models; thresholding by distance to centroid; silhouette coefficient; comparison with outlier detection methods; stream data processing

Suggested Citation:
Budzhelida, S., Dubrovin, V., & Deineha, L. (2026). Software implementation of a network traffic monitoring system based on cluster analysis. *Information Technologies and Computer Engineering*, 23(2), 24-34. doi: 10.31649/vitce/2.2026.24.

*Corresponding author (*budzhelida.samira@gmail.com*)

Introduction

In the context of the rapid growth of cyber threats, network attacks remain one of the key challenges for information systems and corporate networks. Increasing volumes of network traffic, the development of cloud services, and the spread of the internet significantly complicate the monitoring of network activity. Under these conditions, the timely detection of anomalous behaviour in a network becomes critical for ensuring the stability of the information infrastructure and preventing cyber incidents. Conventional intrusion detection systems based on signature analysis can effectively identify only known types of attacks and have limited capacity to respond to new or modified threats. This creates a need for intelligent data analysis methods that automatically detect atypical behaviour in network traffic. Among such intelligent methods, machine learning algorithms occupy a central place. They are actively used in various areas of data analysis, including computer vision, signal processing, and cybersecurity, where they are applied to the automatic detection of complex patterns in large datasets (Nafea *et al.*, 2024).

The issue of detecting anomalies in network traffic is actively examined in the academic literature. In particular, T. Jafarian *et al*. (2020a) presented a systematic review of anomaly detection mechanisms in Software-defined Networking (SDN), in which the researchers classified existing approaches into several main categories, including methods based on flow analysis, entropy, deep learning, and hybrid models. This classification emphasised the diversity of approaches to network traffic analysis and highlighted the relevance of using intelligent algorithms for automated threat detection.

One of the first studies in which clustering was applied to network traffic analysis was conducted by G. Münz *et al*. (2007). The researchers proposed using the k-means algorithm for the analysis of NetFlow data and demonstrated the possibility of automatically detecting anomalous network traffic flows. In the studies from 2020-2023, considerable attention was paid to the use of machine learning methods for detecting anomalies in network traffic. In particular, A.B. Nassif *et al*. (2021) analysed more than 290 studies on anomaly detection problems in different application domains in a systematic review. The researchers emphasised that machine learning methods, in particular, unsupervised algorithms, are effective for tasks where anomalous events occur rarely and the formation of labelled datasets is difficult or economically impractical. The study also stressed that clustering algorithms detect hidden regularities in large datasets and can be used as a basis for automated monitoring systems.

The further development of this subject was presented by A. Yaseen (2023), who examined the role of machine learning methods in modern cybersecurity systems. The researcher established that behavioural models for network traffic analysis detect atypical activity patterns more effectively than conventional signature-based approaches. The study considered various machine learning algorithms, in particular, Support Vector Machine (SVM), Random Forest, and neural networks, which can analyse complex nonlinear relationships in network data. It also stressed the need to develop scalable algorithms capable of processing large data streams in real time.

S. Wang *et al*. (2021) focused on practical aspects of using machine learning algorithms for network traffic analysis to improve cybersecurity. The researchers experimentally demonstrated that clustering methods can group events and automatically detect anomalous patterns characteristic of intrusions. The practical conclusion of the study was that preprocessing and data standardisation are not merely desirable but mandatory stages. They ensured the necessary stability of models and help avoid result distortion caused by different scales of input features, which ultimately increases the reliability of threat detection.

The further development of methods for the intelligent analysis of network traffic was presented by M.A. Ferrag *et al*. (2020) in a large-scale comparative study on deep learning architectures for intrusion detection systems. The researchers described the theoretical principles of discriminative models (RNN, DNN, CNN) and generative models (autoencoders, RBM, DBM, DBN) and conducted their experimental testing on the modern CSE-CIC-IDS2018 and Bot-IoT datasets. A substantial contribution of the study is the systematisation of 35 cybersecurity datasets classified into seven categories, which significantly simplifies the selection of relevant datasets for further cybersecurity research. The experimental results confirmed the effectiveness of deep learning models in both binary and multiclass classification tasks for network attacks. Comparison with conventional machine learning algorithms, such as the naive Bayes classifier, SVM, and Random Forest, led the researchers to justified conclusions about the prospects of applying deep learning to cybersecurity tasks.

The study by Z. Ahmad *et al*. (2020) presents a systematic review of machine learning and deep learning methods for detecting network intrusions, based on publications from 2017-2020. The researchers analysed a wide range of algorithms, from classical methods (SVM, decision trees, and the k-nearest neighbour method) to modern deep learning architectures (RNN, CNN, autoencoders, and DBN), and outlined the strengths and weaknesses of each based on empirical data. An important contribution is its analysis of the datasets used. The researchers reported that 60% of studies still rely on outdated datasets (KDD Cup'99 and NSL-KDD), which limits the practical value of the results. The main conclusion of the review was that, despite high overall accuracy, most models have difficulties detecting rare attack types because of data imbalance. The researchers also noted the dominance of deep learning approaches (80% of the publications considered) and outlined promising directions for further research, in particular, the development of lightweight models for IoT and cyber-physical systems.

The problems of integrating intelligent algorithms into modern cybersecurity systems were considered by D. Dasgupta *et al*. (2022). The researchers examined the use of machine learning for the automated analysis of behavioural characteristics of network traffic and indicated that combining statistical methods with machine learning algorithms increases the accuracy of anomaly detection compared with conventional approaches. The key conclusion was the need to create models that provide high data analysis efficiency and retain the interpretability of results for security specialists, which is critical for understanding threats.

I.H. Sarker (2021) conducted a comprehensive empirical analysis of the effectiveness of machine learning models for detecting network anomalies and cyberattacks. The researcher proposed a two-level modelling architecture: binary classification to separate anomalous traffic from normal traffic and multiclass classification to identify specific attack types. An important methodological feature of the study is the use of correlation-based feature selection, which reduces the computational complexity of models without loss of accuracy. Experiments conducted on the UNSW-NB15 and NSL-KDD datasets established that ensemble methods, especially XGBoost and Random Forest, achieve the best results. The researcher concluded that the success of intelligent security systems is determined not only by the choice of algorithm but also by the quality of data preprocessing and the careful selection of relevant features.

Despite a considerable number of studies in this field, several problems remain insufficiently explored Most studies focus on complex machine learning models that require substantial computational resources and complex data preparation. However,. simple and interpretable clustering methods remain promising for practical application in network traffic monitoring systems, especially under conditions involving the processing of large data streams. Thus, the literature analysis led to two key conclusions that shaped the methodology of this study: first, clustering algorithms, in particular, k-means, are most suitable for detecting rare anomalies in the absence of labelled data; second, the effectiveness of clustering critically depends on the prior standardisation of features, which removes the influence of different measurement scales. The choice of these methodological solutions in this study was directly based on the conclusions of the analysed papers. The purpose of the study was to develop and experimentally examine a software system for monitoring network traffic based on cluster analysis for the automatic detection of anomalous events.

Materials and Methods

This study defined and preprocessed a set of behavioural features characterising network traffic. The k-means clustering algorithm was then applied to group network events, and the effectiveness of the proposed approach for detecting anomalous connections was evaluated. The empirical part was performed on a generated sample of flow events that modelled both typical and atypical network activity. The dataset comprised 1,000 records, of which 980 characterised normal traffic, whereas 20 events reproduced anomalous behavioural patterns. This distribution was deliberately designed to reproduce the realistic class imbalance typical of anomaly detection tasks, where the share of rare events is minimal (2%). A similar property of anomalous events, namely their statistical rarity and substantial deviation from the typical data distribution, is a fundamental characteristic of anomaly analysis tasks, as described in detail in the classical reviews by V. Hodge & J. Austin (2004) and V. Chandola *et al*. (2009). Four flow-level behavioural features were defined for each record: *src_bytes* – volume of outgoing bytes; *dst_bytes* – volume of incoming bytes; *count* – number of connections in the observation window; *srv_count* – intensity of requests to the server. These features belong to the key descriptors of behavioural flow analysis and are widely used in clustering-based anomaly detection systems. The heterogeneity of these features creates the need for scale adjustment before applying metric methods.

Data preprocessing involved a standardisation procedure to remove the influence of different feature scales and ensure the correct calculation of Euclidean distances at subsequent clustering stages. The importance of preprocessing and informative feature selection in network anomaly detection systems was also stressed by T. Jafarian *et al*. (2020b), who used the NetFlow protocol and relevant feature-selection methods before applying machine learning models to traffic analysis in software-defined networks. The researchers demonstrated that ignoring preprocessing leads to the dominance of features with large numerical ranges and a decline in attack detection accuracy, a conclusion confirmed by the results of this research. The StandardScaler transformation was used to normalise the data; it brings each feature to a zero mean and unit standard deviation:

$$X_{scaled} = \frac{X-\mu}{\sigma}, \quad (1)$$

where μ is the mean value of the feature, and σ is the standard deviation. The procedure consists of two stages: calculating the statistical characteristics (μ and σ) for the whole dataset and transforming each feature value using the formula above.

The operation of the transformation is illustrated using the following example: for the *src_bytes* = [200, 300, 400], the mean is $\mu = 300$, and the standard deviation is $\sigma \approx 81{,}65$. The normalised values are therefore:

$200 \rightarrow \approx -1.22$

$300 \rightarrow 0$

$400 \rightarrow \approx +1.22$

After standardisation, the values of all features in the space acquire commensurate scales, which prevents features with large numerical ranges (for example, *src_bytes*, *dst_bytes*) from dominating features with a narrower distribution (*count*, *srv_count*). Standardisation is a critical

condition for the stable operation of the k-means algorithm because it minimises the sum of squared distances in the feature space. There were no missing values in the synthetic dataset because all observations were generated programmatically using NumPy library functions; therefore, no further imputation methods were applied in the study.

The k-means algorithm proposed by J. MacQueen (1967) was used to group records by similarity in the behavioural space. It provides efficient clustering when the purpose is to divide observations into compact clusters with minimal within-cluster variation. In this study, the implementation of network traffic clustering was performed in the Python programming language using the scikit-learn, NumPy, and pandas libraries, which provide the necessary tools for data preprocessing, clustering, and interpretation of results. All experiments were conducted in a reproducible environment with a fixed random state, which ensured result stability. The software implementation included three key stages:

- feature standardisation using StandardScaler, which aligns variable scales and ensures the correct calculation of Euclidean distances in the feature space;
- k-means clustering with k-means++ initialisation, the number of clusters k = 2, the n_init = 10 parameter, and the max_iter = 300 iteration limit;
- calculation of distances to centroids and application of a threshold rule (90th percentile) for the final identification of anomalies in the "small" cluster. The code fragment below reproduces the main stages of data processing and anomalous record identification.

```
    from sklearn.preprocessing import Standard-
Scaler
    from sklearn.cluster import KMeans
    import numpy as np

    # Feature scaling
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)

    # Clustering using k-means
    kmeans = KMeans(
    n_clusters=2,
    init="k-means++",
    max_iter=300,
    n_init=10,
    random_state=42
    )
    labels = kmeans.fit_predict(X_scaled)

    # Calculation of distances to centroids
    distances = np.linalg.norm(
    X_scaled - kmeans.cluster_centers_[labels],
    axis=1
    )

    # Threshold value for anomaly detection
    threshold = np.percentile(distances, 90)

    # Identification of the smaller cluster (can-
didate for anomalies)
    minority_cluster = np.argmin(np.bincount(la-
bels))

    # Detection of anomalous records
    anomalies = np.where(
     (labels == minority_cluster) & (distances >
threshold)
    )[0]
```

The code fragment implements the full processing cycle: from scaling and cluster formation to the detection of deviations based on a combined rule. The use of scikit-learn tools ensures the reproducibility of experiments, transparency of parameters, and the possibility of further integrating the model into batch or stream traffic processing systems. Anomalous records are defined as points that belong to the smaller cluster and simultaneously exceed the threshold distance to the corresponding centroid. The k-means algorithm performs iterative minimisation of the loss function:

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \| x - \mu_i \|^2, \quad (2)$$

where C_i is the set of elements of the cluster, and μ_i is its centroid. The main steps of the algorithm are:

1. Centroid initialisation. The k-means++ method was used, which reduced the effect of an unsuccessful choice of initial centroids and accelerated convergence.

2. Assignment of points to clusters. Each event is assigned to the nearest centroid by Euclidean distance.

3. Centroid update. The new centroid is calculated as the mean value of the points included in the cluster.

4. Convergence check. The process is repeated until the change in centroids or the loss function becomes smaller than the threshold value.

In this study, the number of clusters was fixed at $k = 2$, which corresponded to the problem statement, namely, the separation of a group of normal observations and a group of potential anomalies. The random number generator was fixed for experiment reproducibility. Since the share of anomalous events in the sample was small, it was reasonable to assume the presence of a compact "large" cluster corresponding to normal traffic and a smaller cluster in which potential anomalies were concentrated. Therefore, at the first post-processing stage, the cluster with the smaller size was marked as a candidate anomalous cluster.

An additional distance criterion was applied to reduce false-positive detections. For each record, the Euclidean distance to the centroid of its own cluster was calculated. An event was considered anomalous only if it belonged to the smaller cluster and its distance exceeded the 90th percentile of the within-cluster distribution. This two-stage scheme combines structural information (rarity) with metric information (distance from the centre) and substantially reduces the probability of marking legitimate activity peaks as anomalies.

A comprehensive analysis of the results was conducted to evaluate clustering quality and the accuracy of anomaly identification. In particular, the compactness and separability of the formed clusters in the standardised feature space were evaluated, and the distribution of within-cluster distances was examined. Visual analysis methods were also used, in particular, scatter plots of points in src_bytes-dst_bytes coordinates, indicating membership in the relevant clusters and histograms of distances to centroids showing the threshold value. Separate attention was paid to the

assessment of time costs for algorithm execution, in particular, the duration of training and prediction for a dataset of 1,000 records. Full specification of model parameters and data processing procedures makes the research completely reproducible in an independent computing environment.

Results and Discussion

k-means clustering formed two compact clusters, reproducing the expected bimodality of the dataset. The cluster sizes corresponded to the sample imbalance: the larger cluster aggregated the vast majority of records, and the smaller cluster contained rare events. The centroid of the "normal" cluster was characterised by lower src_bytes and dst_bytes values, along with lower count and srv_count activity; the centroid of the "anomalous" cluster had substantially higher mean values for all four features. Mean Euclidean distances to centroids were lower for the "normal" cluster and higher for the "anomalous" cluster, indicating good separation of the groups in the standardised feature space.

Data scaling was performed using the StandardScaler tool from the sklearn library. This process brought all features (data columns) to a single scale, with a mean value (μ) of 0 and standard deviation (σ) of 1. In this dataset, the features (src_bytes, dst_bytes, count, and srv_count) had different value ranges:

1. src_bytes and dst_bytes were measured in bytes and varied from 50 to 10,000;
2. count denotes the number of connections and, in this dataset, ranged from 1 to 100;
3. srv_count denotes the frequency of requests to the server and, in this dataset, ranged from 1 to 50.

Without scaling, features with larger values, for example, src_bytes, would dominate features with smaller values, for example, srv_count. This could disrupt the operation of the clustering algorithm because distances between points in space would be determined mainly by large values. As a result, the algorithm would form clusters that reflect not the real structure of the data but only differences in the scales of individual features. Standardisation ensures an equal contribution of all variables to the distance calculation process and increases the stability of the k-means algorithm. The clustering results are presented in Table 1, and examples of the detected anomalous records are given in Table 2.

Table 1. Clustering results: number of points, centroid coordinates, and mean distance to centroid for each cluster

Cluster	Number of points	src_bytes	dst_bytes	count	srv_count	Mean distance to centroid
0	980	1,544.29	1,486.13	9.81	4.98	1.16
1	20	7,975.70	7,867.20	76.25	36.70	2.67

Note: the table presents generated clustering results: the size of each cluster, centroid coordinates in the original feature scale, and mean distance values to the centroid
Source: developed by the researchers

Table 2. Anomalous network traffic records detected using the k-means method

Record no.	src_bytes	dst_bytes	count	srv_count	Distance to centroid
980	5,207	9,655	94	25	3.805003
981	7,633	9,840	77	35	1.641567
982	9,117	9,195	50	25	3.542815
983	5,153	7,555	98	36	3.008567
984	8,366	7,305	94	29	2.240655
985	9,314	9,508	53	37	2.704821
986	9,764	6,088	56	46	3.250347
987	6,518	7,642	98	41	2.436200
988	7,600	6,177	58	37	2.160244
989	7,195	9,916	85	22	3.393031
990	9,498	9,757	68	39	2.129318
991	9,691	7,994	50	46	3.264403
992	8,554	5,559	59	40	2.546097
993	7,414	7,964	85	44	1.660048
994	7,849	6,271	58	47	2.864852
995	6,742	6,813	94	34	2.129602
996	6,306	7,915	95	34	2.219929
997	8,849	5,609	91	23	3.510660
998	9,168	7,461	81	48	2.408686
999	9,576	9,120	81	46	2.439567

Note: anomalous records are defined as elements of the cluster with the smallest number of points, which was formed during clustering (k = 2). Distance_to_Centroid values reflect the Euclidean distance of each record to the cluster centroid; larger values indicate a substantial deviation from normal traffic. Clustering was performed on standardised data, but the table presents the original feature values
Source: developed by the researchers

A scatter plot of network traffic in the coordinates of the src_bytes and dst_bytes features was constructed for a visual analysis of the clustering results. This visualisation helps assess cluster separability and the spatial distribution of normal and anomalous traffic in a two-dimensional feature projection. The graphical representation of the clustering results is presented in Figure 1.

As a result of the k-means algorithm (Table 1, Fig. 1), network traffic was successfully divided into two clusters: normal and anomalous. Most observations belonged to the normal traffic cluster, which contained 980 points (98% of the total data volume) and was characterised by moderate values of the main parameters of network activity. The src_bytes and dst_bytes values for these records were within the typical range, corresponding to ordinary volumes of data transfer in the network, whereas the count and srv_count indicators reflected standard connection and server request intensity. The second cluster contained 20 points (2% of the total sample) and was interpreted as anomalous traffic. These records were characterised by substantially larger volumes of transmitted and received data, along with a higher number of connections and server requests, indicating atypical network activity. Such characteristics may correspond to different types of network anomalies, in particular, mass data transfers or increased request intensity.

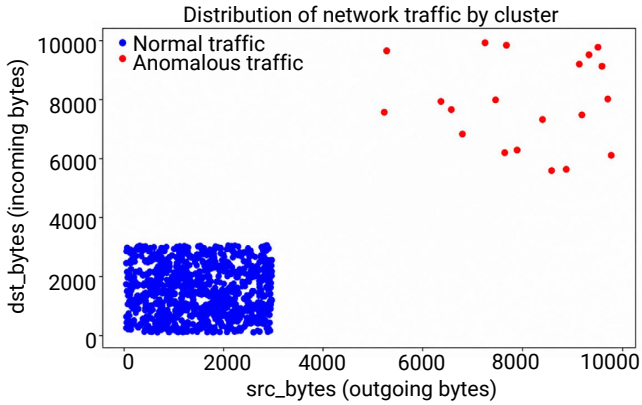

Figure 1. Distribution of network traffic by src_bytes and dst_bytes features after k-means clustering
Note: the diagram is provided as visual confirmation of the separation of normal and anomalous network traffic by the src_bytes and dst_bytes features. The horizontal axis (x) shows the number of outgoing bytes (src_bytes), whereas the vertical axis (y) shows the number of incoming bytes (dst_bytes). Each point corresponds to one network connection, and its coordinates are determined by the values of these features. The colour of the points indicates cluster membership: blue points correspond to normal traffic, whereas red points indicate anomalous connections. Clustering was actually performed in the four-dimensional space of standardised features, whereas the figure presents a two-dimensional projection of the data. The dataset was generated for training purposes to demonstrate the principles of anomaly detection in network traffic
Source: developed by the researchers based on generated data

The findings demonstrate the effectiveness of the applied clustering approach. The algorithm correctly separated rare atypical observations into a distinct cluster, and data processing remained computationally efficient: clustering the dataset of 1,000 records took several milliseconds on a modern computer. Figure 1 illustrates a clear spatial separation of anomalous points from the main mass of normal traffic, and the mean distances to centroids confirm the compactness of the formed clusters. The distribution of Euclidean distances of points to their corresponding centroids was examined for a more detailed analysis of cluster structure. The corresponding histogram is presented in Figure 2.

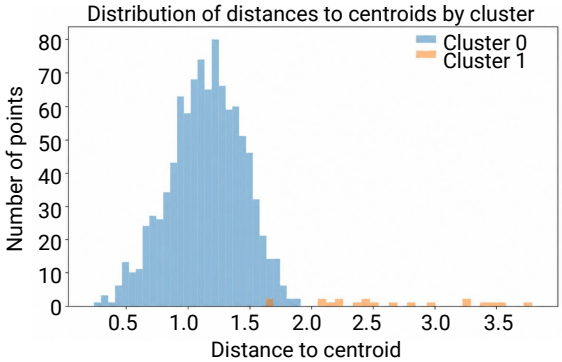

Figure 2. Distribution of distances to centroids for clusters formed using the k-means method
Note: the histogram illustrates the distribution of Euclidean distances of points to centroids for each cluster. Most observations belong to the normal traffic cluster (cluster 0), whereas anomalous events form a separate group with larger distance values
Source: developed by the researchers

The applied combined post-processing rule (membership in the "small" cluster and exceeding the ninetieth percentile of within-cluster distance) reduced the frequency of potential false positives associated with large but legitimate data transfers, while preserving a high proportion of correctly detected rare events. The graphical materials illustrate key aspects of separation. The src_bytes–dst_bytes scatter plot clearly shows a compact area of typical activity and distant points of atypical behaviour. The tabular material therefore includes cluster sizes, centroid coordinates in the original feature scale, mean within-cluster distances, and the share of records marked as anomalous after thresholding.

The results are consistent with the practice of using interpretable clustering approaches in network monitoring tasks, where transparent decision-making criteria and stable response time are important. I.H. Sarker (2021) conducted a large-scale empirical comparison of ten machine learning algorithms, in particular, Naive Bayes, SVM, Random Forest, and XGBoost, on the widely recognised UNSW-NB15 and NSL-KDD datasets. The study indicated that although ensemble methods (Random Forest and XGBoost) provide the highest classification accuracy (over 85% for multiclass classification), the decisive factor in effectiveness is not so much algorithmic complexity as the quality of data preprocessing and the selection of relevant features. An important difference is that the study by I.H. Sarker used a classification approach with labelled data, whereas the present study used a clustering approach (k-means) that does not require prior data labelling. Despite this methodological difference, the common conclusion is the decisive role of preprocessing: feature standardisation avoids the dominance of parameters with large numerical ranges, in particular, traffic volume over the number of packets, and ensures the correct formation of clusters in which anomalous events (less than 5% of the sample and approximately 2% in the conducted research) form a separate group. This conclusion is consistent with the broader approach to constructing intrusion detection systems presented by B. Molina-Coronado *et al*. (2020), who emphasised within the Knowledge Discovery in Databases framework that the effectiveness of intrusion detection systems depends on the coherence of all data-processing stages, from preprocessing and feature selection to the choice of analysis algorithm. The researchers noted that both classification methods and clustering approaches can demonstrate high effectiveness if this process is properly organised.

In the field of intrusion detection systems, the study by C. Zhang *et al*. (2025) systematised key machine learning technologies used in cybersecurity. Among the key strands, the researchers identified zero-day attack detection through anomaly analysis, explainable artificial intelligence (xAI), encrypted traffic analysis, Graph Neural Networks, and the use of Large Language Models. Although clustering methods are not the central subject of this review, the researchers recognised their effectiveness for the initial detection of anomalous patterns in large datasets, especially under conditions of substantial class imbalance. A similar regularity was also observed in this study: anomalous records constituted a small proportion of the total sample (less than 5%, actually 2%), yet they formed a separate compact cluster that differed substantially from the main mass of normal traffic in the values of key parameters.

The study by S.M. Miraftabzadeh *et al*. (2023) is the most relevant to the approach used; it includes a large-scale analysis of the application of clustering algorithms in complex technical systems. The researchers considered more than 440 publications and systematised seven categories of clustering methods: centroid-based, hierarchical, probabilistic, density-based, grid-based, graph-based, and other modified approaches. Within the experimental research, eight clustering algorithms were compared on synthetic time series. The results showed that some alternative methods, in particular, CLIQUE and Bayesian Gaussian Mixture, could achieve substantially higher clustering accuracy (up to 100% according to the Normalised Mutual Information, NMI, metric), whereas the classical k-means algorithm showed lower values under certain conditions. The researchers also stressed that clustering quality depends substantially on data preprocessing, in particular, feature scaling and dimensionality reduction. The obtained results confirm this conclusion: feature standardisation avoids the dominance of parameters with large numerical ranges and ensures the correct formation of clusters in the space of network traffic characteristics.

Similar approaches to the use of clustering algorithms in cybersecurity tasks were also presented by J. Chen *et al*. (2022), who proposed a method for detecting malware based on an improved clustering algorithm and data self-similarity analysis. The further development of ideas for improving clustering algorithms was presented by Y. Yuan & Y. Li (2022), who developed a hybrid method for identifying network anomalies. The method combined the k-means algorithm with optimisation methods, namely Particle Swarm Optimisation and a Genetic algorithm. This approach avoids the problem of local optima and improves detection accuracy in complex, multidimensional network datasets compared with the use of the classical k-means algorithm.

In the fundamental review by R. Chalapathy & S. Chawla (2019), the researchers proposed a classification of Deep Anomaly Detection methods, including hybrid models, One-Class Neural Networks, semi-supervised approaches, and unsupervised approaches. The researchers demonstrated that deep learning models, in particular, autoencoders, Long short-term memory recurrent neural networks, and Generative adversarial networks, can detect complex nonlinear dependencies in large datasets. However, the use of such models often requires substantial computational resources and large training samples. In operational network traffic monitoring systems, where processing speed and result interpretability are critical, simpler algorithms can remain an effective alternative. The k-means algorithm is characterised by relatively low

computational complexity, implementation simplicity, and a transparent clustering mechanism, which enables rapid event grouping and the analysis of centroid coordinates for result interpretation.

Conclusions

The findings confirm that the stated purpose was achieved and demonstrate the possibility of effectively applying the k-means algorithm to analyse behavioural characteristics of network connections. A set of behavioural features of network traffic was developed, and data preprocessing and standardisation were performed, ensuring the correctness of further calculations in the multidimensional feature space. The k-means clustering algorithm was applied to the prepared data, forming two stable clusters that reflected normal and anomalous network activity. The conducted analysis established that most observations belonged to the normal traffic cluster, whereas anomalous records formed a separate compact group characterised by increased values of transmitted data volume and network connection intensity. The use of an additional combined rule for interpreting clustering results, which considered membership in the smaller cluster and the exceedance of the threshold distance to the centroid, increased the accuracy of anomalous event detection and reduced the probability of false positives.

The obtained results indicate the appropriateness of using interpretable clustering methods for automated network traffic monitoring. The k-means algorithm effectively detected anomalous behavioural patterns even in cases of substantial data imbalance, where anomalous events constituted only a small proportion of the sample. The algorithm also has relatively low computational complexity and enables rapid data processing, which makes it suitable for integration into operational systems for analysing network activity.

The limitations of the study are associated with the use of a generated dataset and the limited number of features describing network activity. Under real-world conditions, improving analysis accuracy may require the inclusion of additional characteristics of network traffic, in particular, temporal parameters, packet statistics, and behavioural features of different types of network services. Prospects for further research include expanding the set of behavioural features, examining the effectiveness of the algorithm on real network data, and developing hybrid approaches that combine clustering methods with modern machine learning and deep learning algorithms to improve the accuracy and adaptability of anomaly detection systems.

Acknowledgements

None.

Funding

The study received no funding.

Conflict of Interest

None.

References

[1] Alang, K., Hassan, S.Z., Katkam, V., & Hassan, S. (2025). Real-time ML and LLM optimization: Orchestrating scalable workflows in distributed commerce environments. In *2025 international conference on computing technologies & data communication* (pp. 1-7). Hassan: IEEE. doi: 10.1109/ICCTDC64446.2025.11158822.

[2] Ali, A., & Ghanem, M.C. (2025). Beyond detection: Large language models and next-generation cybersecurity. *SHIFRA*, 2025, 81-97. doi: 10.70470/SHIFRA/2025/005.

[3] Ali, T., & Kostakos, P. (2023). Huntgpt: Integrating machine learning-based anomaly detection and explainable AI with large language models (LLMs). *ArXiv*. doi: 10.48550/arXiv.2309.16021.

[4] Aljumaily, M., Abd, H., & Majeed, E. (2025). Enhancing user and entity behavior analytics in SIEM systems using AI-powered anomaly detection: A data-driven simulation approach. *International Journal of Mechatronics, Robotics, and Artificial Intelligence*, 1(2), 82-93. doi: 10.33971/ijmrai.1.2.11.

[5] Amer, L. (2025). AI in cyber security: A dual perspective on hacker tactics and defensive strategies. *Cyber Security: A Peer-Reviewed Journal*, 8(3), 198-213. doi: 10.69554/CLXC9075.

[6] Arjunan, T. (2024). Detecting anomalies and intrusions in unstructured cybersecurity data using natural language processing. *International Journal for Research in Applied Science & Engineering Technology*, 12(2), 1023-1029. doi: 10.22214/ijraset.2024.58497.

[7] Boddu, R., & Lamppu, S. (2024). *Microsoft unified XDR and SIEM solution handbook: Modernize and build a unified SOC platform for future-proof security*. Birmingham: Packt Publishing Ltd.

[8] Brandao, P.R. (2025). Exploring the role of artificial intelligence in detecting advanced persistent threats. *Computers,* 14(7), article number 245. doi: 10.3390/computers14070245.

[9] da Costa, F.H., Medeiros, I., Menezes, T., da Silva, J.V., da Silva, I.L., Bonifácio, R., Narasimhan, K., & Ribeiro, M. (2022). Exploring the use of static and dynamic analysis to improve the performance of the mining sandbox approach for android malware identification. *Journal of Systems and Software*, 183, article number 111092. doi: 10.1016/j.jss.2021.111092.

[10] Desetty, A.G. (2024). Unveiling hidden threats with ML-powered user and entity behavior analytics (UEBA). *Turkish Journal of Computer and Mathematics Education*, 15(1), 44-50.

[11] Donepudi, S., Lakshmi, U.P., Kumar, N.P., Lalitha, S., Shaik, R., & Devi, D.A. (2025). Efficient LLM inference on mcp servers: A scalable architecture for edge-cloud ai deployment. *Journal of Theoretical and Applied Information Technology*, 103(13), 4885-4895.

[12] Esposito, G. (2025). *LLMs in the SIEM loop: A contract-based framework for threat detection with an evaluation on Windows telemetry and MITRE ATT&CK mapping*. Torino: Polytechnic University of Turin.
[13] Fuentes, J., Ortega-Fernandez, I., Villanueva, N.M., & Sestelo, M. (2025). Cybersecurity threat detection based on a UEBA framework using Deep Autoencoders. *AIMS Mathematics*, 10(10), 23496-23517. doi: 10.3934/math.20251043.
[14] Guduru, S. (2025). Autonomous cyber defense: LLM-Powered incident response with LangChain and SOAR integration. *International Journal of Computer Science and Information Technology Research*, 6(1), 72-82. doi: 10.63530/IJCSITR_2025_06_01_008.
[15] Hakonen, P. (2022). *Detecting insider threats using user and entity behavior analytics*. (Master's thesis, JAMK University of Applied Sciences, Jyväskylä, Finland).
[16] Hassanov, I., Virtanen, S., Hakkala, A., & Isoaho, J. (2024). Application of large language models in cybersecurity: A systematic literature review. *IEEE Access*, 12, 176751-176778. doi: 10.1109/ACCESS.2024.3505983.
[17] Huang, F., Xiong, H., Chen, S., Lv, Z., Huang, J., Chang, Z., & Catani, F. (2023). Slope stability prediction based on a long short-term memory neural network: comparisons with convolutional neural networks, support vector machines and random forest models. *International Journal of Coal Science & Technology*, 10(1), article number 18. doi: 10.1007/s40789-023-00579-4.
[18] Hussain, M.J. (2024). A survey based on behavior analysis of artificial intelligence using machine learning process. In *2024 4th international conference on sustainable expert systems* (pp. 1694-1701). New York: IEEE. doi: 10.1109/ICSES63445.2024.10763264.
[19] Ibrahim, N., & Kashef, R. (2025). Exploring the emerging role of large language models in smart grid cybersecurity: A survey of attacks, detection mechanisms, and mitigation strategies. *Frontiers in Energy Research*, 13, article number 1531655. doi: 10.3389/fenrg.2025.1531655.
[20] ISO/IEC 27001:2022. (2022). *Information security, cybersecurity and privacy protection – information security management systems – requirements*. Retrieved from https://www.iso.org/standard/27001.
[21] Jaffal, N.O., Alkhanafseh, M., & Mohaisen, D. (2025). Large language models in cybersecurity: A survey of applications, vulnerabilities, and defense techniques. *AI*, 6(9), article number 216. doi: 10.3390/ai6090216.
[22] Jiang, X., Jia, R., & Zhang, F. (2025). Deep learning-based user behavior anomaly detection and threat early warning in cloud computing environments. *Academia Nexus Journal*, 4(3).
[23] Karras, A., Theodorakopoulos, L., Karras, C., Theodoropoulou, A., Kalliampakou, I., & Kalogeratos, G. (2025). LLMs for cybersecurity in the big data era: A comprehensive review of applications, challenges, and future directions. *Information*, 16(11), article number 957. doi: 10.3390/info16110957.
[24] Kasri, W., Himeur, Y., Alkhazaleh, H.A., Tarapiah, S., Atalla, S., Mansoor, W., & Al-Ahmad, H. (2025). From vulnerability to defense: The role of large language models in enhancing cybersecurity. *Computation*, 13(2), article number 30. doi: 10.3390/computation13020030.
[25] Katreddy, S.S. (2023). Optimizing AI/ML workloads in cloud environments: A scalable approach. *International Journal of Intelligent Systems and Applications in Engineering*, 11(11), 710-719.
[26] Kethireddy, R.R. (2022). AI-powered insider threat detection with behavioral analytics with LLM. *International Journal of Science and Research*, 11(10), 1449-1453. doi: 10.21275/SR221013110718.
[27] Khan, M.Z.A., Khan, M.M., & Arshad, J. (2022). Anomaly detection and enterprise security using user and entity behavior analytics (UEBA). In *2022 3rd international conference on innovations in computer science & software engineering* (pp. 1-9). New York: IEEE. doi: 10.1109/ICONICS56716.2022.10100596.
[28] Malik, V., Khanna, A., Sharma, N., & Nalluri, S. (2024). Advanced persistent threats (APTs): Detection techniques and mitigation strategies. *International Journal of Global Innovations and Solutions*. doi: 10.21428/e90189c8.91e89a3e.
[29] Mareedu, A. (2025). Autonomous Security Operations Centers (SOC): AI agents for threat triage, response, and orchestration. *International Journal of Emerging Research in Engineering and Technology*, 6(2), 63-70. doi: 10.63282/3050-922X.IJERET-V6I2P108.
[30] Mihailescu, M.I., Nita, S.L., Rogobete, M., & Marascu, V. (2023). Unveiling threats: Leveraging user behavior analysis for enhanced cybersecurity. In *2023 15th international conference on electronics, computers and artificial intelligence* (pp. 1-6). New York: IEEE. doi: 10.1109/ECAI58194.2023.10194039.
[31] Mir, A.W., & Kumar, K.R. (2022). An enhanced implementation of security management system (SSMS) using UEBA in Smart Grid based SCADA systems. In J.K. Mandal, S. Misra, J.S. Banerjee & S. Nayak (Eds.), *Proceedings of 2nd global conference on artificial intelligence and applications: Applications of machine intelligence in engineering* (pp. 1-11). Boca Raton: CRC Press. doi: 10.1201/9781003269793.
[32] Mohanty, R.K. (2025). Deep learning for analyzing user and entity behaviors: Techniques and applications. In N. Marriwala, S. Jain, V. Shukla & D. Kumar (Eds.), *Hybrid soft computing techniques for machine learning and optimization* (pp. 121-148). Hershey: IGI Global Scientific Publishing. doi: 10.4018/979-8-3693-6864-0.ch006.
[33] Motlagh, F.N., Hajizadeh, M., Majd, M., Najafi, P., Cheng, F., & Meinel, C. (2024). Large language models in cybersecurity: State-of-the-art. *ArXiv*. doi: 10.48550/arXiv.2402.00891.

[34] Mustafa, A.M. (2024). *Leveraging AI for confident classification and prioritization of intrusion detection system alerts*. (Master's thesis, American University of Beirut, Beirut, Lebanon).
[35] Naqvi, B., Perova, K., Farooq, A., Makhdoom, I., Oyedeji, S., & Porras, J. (2023). Mitigation strategies against the phishing attacks: A systematic literature review. *Computers & Security*, 132, article number 103387. doi: 10.1016/j.cose.2023.103387.
[36] Odozor, L.A., Ransome-Kuti, O.S., Odeniran, Q., Olisa, A.O., Berko, S.N., & Abaya, J.T. (2025). Data-driven incident response: Enhancing detection and containment through adversarial reasoning and malware behavior analytics. *International Journal of Innovative Science and Research Technology*, 10(9), 218-230. doi: 10.38124/ijisrt/25sep154.
[37] Önal, V., Arslan, H., & Canay, Ö. (2025). Anomaly detection in SIEM data: User behavior analysis with artificial intelligence. In P. Bhambri & A.J. Anand (Eds.), *Handbook of AI-driven threat detection and prevention: A holistic approach to security* (pp. 269-289). Boca Raton: CRC Press. doi: 10.1201/9781003521020.
[38] Pitkar, H. (2025). Cloud security automation through symmetry: Threat detection and response. *Symmetry*, 17(6), article number 859. doi: 10.3390/sym17060859.
[39] Putra, F.P.E., Ubaidi, Zulfikri, A., Arifin, G., & Ilhamsyah, R.M. (2024). Analysis of phishing attack trends, impacts and prevention methods: literature study. *Brilliance: Research of Artificial Intelligence*, 4(1), 413-421. doi: 10.47709/brilliance.v4i1.4357.
[40] Rahman, N. (2024). *Leveraging large language models for network traffic analysis: Design, implementation, and evaluation of an LLM-powered system for cyber incident reconstruction.* (Master's thesis, University of Turku, Turku, Finland).
[41] Razavi, H., Ouaissa, M., Ouaissa, M., Nakouri, H., & Abdelgawad, A. (2025). *AI-driven cybersecurity: Revolutionizing threat detection and defence systems*. Boca Raton: CRC Press. doi: 10.1201/9781003631507.
[42] Regulation (EU) No. 2016/679 of the European Parliament and of the Council "On the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data, and Repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance)". (2016, April). Retrieved from https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng.
[43] Saraiva, M., & Mateus-Coelho, N. (2022). CyberSoc framework a systematic review of the state-of-art. *Procedia Computer Science*, 204, 961-972. doi: 10.1016/j.procs.2022.08.117.
[44] Sarker, I.H. (2024). Generative AI and large language modeling in cybersecurity. In *AI-driven cybersecurity and threat intelligence: Cyber automation, intelligent decision-making and explainability* (pp. 79-99). Cham: Springer. doi: 10.1007/978-3-031-54497-2_5.
[45] Semerikov, S.O., Vakaliuk,T.A., Kanevska, O.B., Moiseienko, M.V., Donchev, I.I., & Kolhatin, A.O. (2025). LLM on the edge: The new frontier. In *Proceedings of the 5th edge computing workshop* (pp. 137-161). Zhytomyr: CEUR-WS.
[46] Shakil, N.A.F., Mia, R., & Ahmed, I. (2023). Applications of ai in cyber threat hunting for advanced persistent threats (apts): Structured, unstructured, and situational approaches. *Journal of Applied Big Data Analytics, Decision-Making, and Predictive Modelling Systems*, 7(12), 19-36.
[47] Sharma, G., Thakur, A., & Tiwari, C. (2024). Developing a comprehensive framework for user and entity behavior analytics (UEBA): Integrating advanced machine learning and contextual insights. *Journal of Communication Engineering & Systems*, 14(2), 20-31.
[48] Sokyrka, I., Kukulevskyi, I., & Tolbatov, A. (2025). Authentication methods using behavioral analytics and machine learning for internet of things devices. *Electronic Professional Scientific Journal "Cybersecurity: Education, Science, Technique",* 2(30), 35-49. doi: 10.28925/2663-4023.2025.30.941.
[49] Subrahmanyam, S. (2025). Behavioral analysis for threat detection. In P. Bhambri & A.J. Anand (Eds.), *Handbook of AI-driven threat detection and prevention: A holistic approach to security* (pp. 95-115). Boca Raton: CRC Press. doi: 10.1201/9781003521020.
[50] Suprun, O., & Karpenko, N. (2025). Information security in the context of user behavior analysis. In *International scientific-practical conference "Problems of computer sciences, software modeling and security of digital systems"* (pp. 104-107). Lutsk: Lesya Ukrainka Volyn National University.
[51] Thelwall, M. (2025). Research quality evaluation by AI in the era of large language models: Advantages, disadvantages, and systemic effects – an opinion paper. *Scientometrics*, 130(10), 5309-5321. doi: 10.1007/s11192-025-05361-8.
[52] Trivedi, A., Gupta, R., & Jangal, K. (2025). Research paper on cybersecurity and insider threat detection: The role of user behavior analytics (UBA) in modern defense strategies. *International Journal for Research in Applied Science & Engineering Technology,* 13(1), 455-466. doi: 10.22214/ijraset.2025.66298.
[53] Vieira, L.D.S.L. (2025). *Development of a web application for real-time inference in AI models for autonomous driving*. (Master thesis, University of Porto, Porto, Portugal).
[54] Wairagade, A., & Ranjan, S. (2025). User behavior analysis for cyber threat detection: A comparative study of machine learning algorithms. In *2025 13th international symposium on digital forensics and security* (pp. 1-6). New York: IEEE. doi: 10.1109/ISDFS65363.2025.11011949.

[55] Wang, F., Zhu, G., Yuan, C., & Huang, Y. (2024). LLM-enhanced cascaded multi-level learning on temporal heterogeneous graphs. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval* (pp. 512-521). New York: ACM. doi: 10.1145/3626772.3657731.

[56] Xu, H., Wang, S., Li, N., Wang, K., Zhao, Y., Chen, K., Yu, T., Liu, Y., & Wang, H. (2025). Large language models for cyber security: A systematic literature review. *ACM Transactions on Software Engineering and Methodology*. doi: 10.1145/3769676.

[57] Zhang, M., Shen, X., Cao, J., Cui, Z., & Jiang, S. (2025). Edgeshard: Efficient LLM inference via collaborative edge computing. *IEEE Internet of Things Journal,* 12(10), 13119-13131. doi: 10.1109/JIOT.2024.3524255.

Програмна реалізація системи моніторингу мережевого трафіку на основі кластерного аналізу

Саміра Буджеліда

Студент
Національний університет «Запорізька політехніка»
69063, вул. Університетська, 64, м. Запоріжжя, Україна
https://orcid.org/0009-0002-4837-1022

Валерій Дубровін

Кандидат технічних наук, професор
Національний університет «Запорізька політехніка»
69063, вул. Університетська, 64, м. Запоріжжя, Україна
https://orcid.org/0000-0002-0848-8202

Лариса Дейнега

Старший викладач
Національний університет «Запорізька політехніка»
69063, вул. Університетська, 64, м. Запоріжжя, Україна
https://orcid.org/0000-0003-0304-4327

Анотація. Зростання обсягів і різнорідності мережевого трафіку ускладнює своєчасне виявлення атак, тоді як підходи, що ґрунтуються на сигнатурах, не охоплюють нові або модифіковані загрози. Потрібні інтерпретовані й відтворювані методи аналізу потоків даних, придатні для інтеграції в моніторингові системи. Метою дослідження було розробити та програмно реалізувати систему моніторингу, яка без попереднього розмічування даних відокремлює нормальні з'єднання від аномальних та надає прозорі критерії прийняття рішень. Методологія дослідження ґрунтувалася на стандартизації ознак мережевого трафіку та застосуванні алгоритму неконтрольованої кластеризації k-means з подальшим виявленням аномалій за відхиленням від центроїдів. На синтетичному наборі подій із невеликою часткою порушень система стабільно формувала два компактні кластери, що відповідали типовій та нетиповій поведінці. Центроїд нормального трафіку характеризувалося меншими значеннями обсягів переданих і прийнятих даних, а також нижчою активністю з'єднань; центроїд аномального – істотно вищими показниками за всіма ознаками. Комбіноване правило після кластеризації зменшувало хибні спрацювання, які виникають під час легітимних великих передавань (наприклад, резервних копій), і водночас утримує високу частку правильно виявлених рідкісних подій. Порівняльні експерименти демонструвало співставну або кращу точність і повноту щодо класичних підходів до виявлення викидів, а також стабільний час обробки на рівні мілісекунд для тисячі записів. Аналіз чутливості підтвердив стійкість до вибору порога відстані та параметрів попередньої обробки. Експериментальна вибірка містила 1 000 записів, з яких 980 відповідали нормальному мережевому трафіку, а 20 – аномальним подіям (2 %). У результаті кластеризації сформовано два кластери, причому менший кластер об'єднав 20 записів, що відповідають нетиповій поведінці мережі. Запропонований підхід не потребує еталонних міток, легко масштабується, забезпечує прозорі пояснення через центроїди та відстані, придатний для пакетної та потокової обробки й може слугувати базовою ланкою у виробничих конвеєрах виявлення аномалій

Ключові слова: виявлення аномальних з'єднань; стандартизація ознак; центроїдні моделі; порогування за відстанню до центроїда; коефіцієнт силуету; порівняння з методами виявлення викидів; потокова обробка даних