Data and Metadata. 2025; 4:739
doi: 10.56294/dm2025739

ORIGINAL

Hybrid Intrusion detection model-based density clustering approach and deep learning for detection of malicious traffic over network

Modelo híbrido de detección de intrusiones basado en agrupación por densidad y aprendizaje profundo para la detección de tráfico malicioso en la red

Ola Ali Obead[1], Hakem Beitollahi[2]

[1]Department of Information Networks, College of Information Technology, University of Babylon. Iraq.
[2]School of Computer Engineering, Iran University of Science and Technology. Tehran, Iran.

Cite as: G Ali Obead O, Beitollahi H. Hybrid Intrusion detection model-based density clustering approach and deep learning for detection of malicious traffic over network. Data and Metadata. 2025; 4:739. https://doi.org/10.56294/dm2025739

Submitted: 28-04-2024 **Revised:** 25-08-2024 **Accepted:** 22-03-2025 **Published:** 23-03-2025

Editor: Dr. Adrián Alejandro Vitón Castillo

Corresponding Author: Ola Ali Obead

ABSTRACT

Intrusion detection in modern network environments poses significant challenges due to the increasing volume and complexity of cyber-attacks. This study proposes a hybrid approach integrating density-based clustering with deep learning to identify malicious traffic over the network. The proposed framework consists of two steps: clustering and classifying data. in clustering, the proposed model uses density clustering techniques to pre-process and segment network traffic into coherent clusters, thereby reducing data noise within clusters. The deep learning model analyses these clusters, accurately distinguishing between benign and malicious activities. The proposed model was tested over the benchmark dataset CIRA-CIC-DoHBrw-2020. The performance of the proposed model compared with standard machine learning models and the number of states of the artworks. The experiment result demonstrates that our hybrid model significantly improves detection accuracy and reduces false-positive rates compared to existing methods.

Keywords: Clustering; Density Clustering; Domain Name System; Deep Learning; Malicious Traffic.

RESUMEN

La detección de intrusiones en entornos de redes modernas plantea desafíos significativos debido al creciente volumen y complejidad de los ciberataques. Este estudio propone un enfoque híbrido que integra la agrupación basada en densidad con aprendizaje profundo para identificar tráfico malicioso en la red. El marco propuesto consta de dos etapas: agrupación y clasificación de datos. En la fase de agrupación, el modelo utiliza técnicas de agrupación por densidad para preprocesar y segmentar el tráfico de red en grupos coherentes, reduciendo así el ruido de los datos dentro de los grupos. Luego, el modelo de aprendizaje profundo analiza estos grupos, distinguiendo con precisión entre actividades benignas y maliciosas. El modelo propuesto fue evaluado utilizando el conjunto de datos de referencia CIRA-CIC-DoHBrw-2020. Su rendimiento se comparó con modelos de aprendizaje automático estándar y con los métodos más avanzados en el estado del arte. Los resultados experimentales demuestran que nuestro modelo híbrido mejora significativamente la precisión de detección y reduce las tasas de falsos positivos en comparación con los métodos existentes.

Palabras clave: Agrupación; Agrupación por Densidad; Domain Name System; Aprendizaje Profundo; Tráfico Malicioso.

INTRODUCTION

The Internet has become essential to our lives, underpinning critical infrastructure, communication, education, commerce, and many online services.[1] Life in the current era has significantly depended on the means of service available using the Internet, as the Internet was used to transfer important information, financial transactions, correspondence, and essential documents. However, there has been a significant increase in Internet users, as shown on the site (https://www.statista.com). The number of Internet users in 2024 increased by 66 %. Figure 1 shows the Worldwide internet user penetration from 2014 to January 2024.

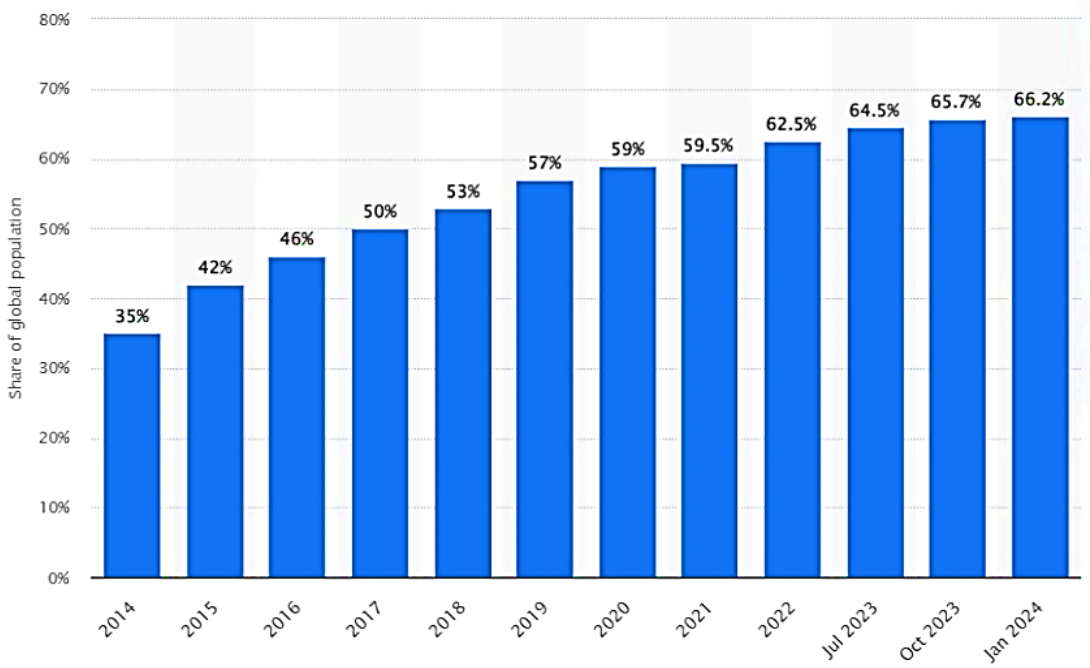


Figure 1. Worldwide internet user penetration from 2014 to January 2024, according to https://www.statista.com

Due to the importance of information transmitted via the Internet and the reliance on companies that provide online services, these companies and information are exposed to many attacks of theft and sabotage.[2] However, this reliance also attracts malicious actors who seek to disrupt or exploit vulnerabilities.

Intrusion Detection Systems (IDSs) is a digital security tool for computer networks.[3,4] They watch over the network traffic and sound an alarm if they spot anything suspicious. The IDSs depend on specific patterns and struggle to keep up with the ever-changing DDoS attacks. These attacks sneak in using normal rules and team up with other computers to cause even more trouble.[5]

Various types of Artificial intelligence (AI) models have been used in IDS. Whether they are algorithms based on deep learning or based on data mining algorithms, Support vector machine (SVM), Decision tree (DT), and Naïve Bayes (NB), among other algorithms, in determining the intrusion attacks. When artificial intelligence algorithms are applied to assess intrusion attacks, they suffer from two problems: the overlap of the attack data with natural data and the high dimensions of the data.[6] Moreover, due to their dynamic and evolving nature, Traditional, signature-based intrusion detection systems (IDSs) are often ineffective against DDoS attacks.[7,8,9] The intrusion detection model-based machine learning can learn from historical data and adapt to new attack patterns, offering greater flexibility and resilience than traditional methods.[10]

Ahmed et al.[11] proposed a method for detecting Distributed Denial-of-Service (DDoS) attacks on the application layer. Relevant features indicative of potential DDoS attacks is extracted from the preprocessed data. The proposed model classified the intrusion attacks based on the employed Multi-Layer Perceptron (MLP). The attack data is significantly less than normal traffic data, so the model might learn to favor the majority class and struggle to detect attacks accurately.

In [12], the authors proposed a proactive feature selection model for DRDoS detection using DNS responses. The model employs enhanced optimization algorithms to select relevant features and reduce dimensionality. It utilizes machine learning algorithms (k-NN, random forest, SVM) for classification based on features chosen. The authors used re-randomization when the algorithm reached stagnation.

The authors in [13] proposed a clustering method for classifying the DNS traffic to intrusion and normal. The proposed model reduces the number of producing clustering from the first stage based on frequent Euclidean distance and threshold. The limitation of this work is that the authors reused the same metrics across distributed data on clusters and, in an evolving process, made the clustering algorithm rely heavily on the threshold. Moreover, the reduced number of clustering in this manner increased overlap in final clusters and reduced the

effect of the IDS model in distinguishing intrusion attacks.

A deep learning framework for DDoS attack detection was proposed in [14], utilizing a contractive auto coder to effectively represent average traffic data. The model employs a stochastic threshold technique based on reconstruction error to identify anomalies within the dataset. However, directly applying machine learning algorithms without data preprocessing introduces a bias towards majority classes, reducing the system's effectiveness. This constitutes a weakness of the proposed model.

The study in [6] introduced a model to detect Distributed Reflection Denial-of-Service (DRDoS) attacks on DNS infrastructure based on comping three steps: feature selection, clustering, and classification. The proposed model selected the optimal features based on the wrapper model and adaptive threshold. In the second step, the distributed data on clusters based on frequent Euclidean distance utile the final clusters achieve a value less than the threshold that has been determined. The proposed model used machine learning to classify and detect intrusion attacks. The proposed model selected features based on standard optimization algorithms without enhancing the central problem of these algorithms, which is stagnation at the local optimum. Therefore, the algorithms essentially suffer from a lack of exportation, and the adaptive threshold has no significant enhancement search engine of the algorithm. Moreover, using thresholds in each stage of the proposed model makes the proposed model depend heavily on thresholds. In addition, the clustering grouped data with the same behavior, making segregating them difficult. Therefore, machine learning algorithms in this model suffer from a low positive rate.

Empowering Clustering with Machine Learning

Clustering is central to improving the detection and prevention of intrusion attacks by reducing data complexity, improving segmentation, and increasing detection accuracy.[15,16] The mathematical foundations of clustering algorithms provide robust tools for data analysis and model training, which are central to this approach.[17] These techniques enable the development of more efficient and accurate intrusion detection systems that protect network infrastructures from evolving cyber threats. In addition, clustering enhances deep learning models by improving feature learning, addressing data imbalances, and enabling multi-layered analysis — all of which contribute to more effective and efficient DRDoS detection. Training the model on the augmented data X' enhances its ability to extract relevant features and improves overall performance because augmenting the input features X with clustering features C increases the mutual information between the inputs and the target labels Y. Mathematically, this is demonstrated by the inequality $I(X';Y)=I((X,C);Y)=I(X;Y)+I(C;Y|X)\geq I(X;Y)$, where $I(C;Y|X)\geq 0$ due to the non-negativity of conditional mutual information. This means that the clustering features C, derived from X, provide additional information about Y that X alone may not capture. Consequently, adding C cannot decrease the mutual information and can potentially increase it if $I(C;Y|X)>0$, thereby providing mathematical evidence that training on X' enhances feature extraction and overall model performance in IDS detection. Figure 2 illustrates the comparative mutual information before and after feature selection applied to the intrusion detection dataset.

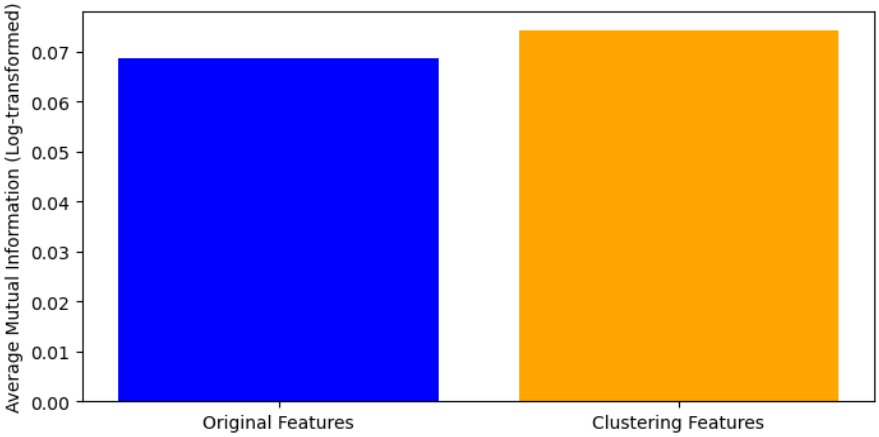


Figure 2. Comparative Mutual Information Before and After Feature Selection on the Intrusion Detection Dataset

Training the model on the augmented data enhances its ability to extract relevant features and improves overall performance due to the following reasons:

- Improving Context: clustering features provide information about the data's structure.
- Improved Separability: augmented features make classes more distinguishable in the feature space.
- Better Generalization: the model is guided toward meaningful patterns, reducing overfitting.
- Enhanced Minority Detection: emphasizes rare but important patterns crucial for detecting attacks.

https://doi.org/10.56294/dm2025739

Empower Cluster of the Train Data

Dynamic clustering is a novel and practical method for filtering data, and it is an ongoing process.[18] Via this dynamic approach, the (computer) models get training data, and data analysis is improved via data density and homogeneity. This method of data analysis, mainly through clustering algorithms, can dynamically enable a model to identify which features should be extracted[19,20] and thus allow the learning and prediction phases to be more accurate.

This strategy intertwines with data density and homogeneity, which dictate the refining process and assist in expediting user-friendly data. Dynamic clustering is a form of evolutionary methodology that helps in iterative sample improvement, enabling the model to discover increasingly complex patterns and relationships within the dataset.[21] This iterative process continues until the data is sufficiently manipulated to make it consistent with the underlying structures, giving the algorithm a better data set for recognizability and precision.

However, there are further complications, such as defining appropriate data density and homogeneity metrics. Training data should be diverse, balanced, optimized, and measurable to prevent bias within the model and for the model to generalize. A careful combination of this stage and its avoidance significantly boosts model success; therefore, popular balancing techniques for producing accurate and reliable results like SMOTE or cost-effective learning have been applied against data imbalance, mainly in the fields of cyber defense and network flow analysis, but recent studies show weaknesses in these solutions.[19] For example, SMOTE can artificially increase the size of the minority class (malicious traffic), potentially causing overfitting and generating synthetic examples that may not accurately capture the patterns of real-world attacks, ultimately reducing the generalizability of the model to real-world use cases.

Datasets (CIRA-CIC-DoHBrw-2020)

The CIRA-CIC-DoHBrw-2020 dataset to test the validity of the proposed model for the following objects:

- Testing and validity of the proposed clustering algorithm in the split data into homogeneity groups.
- Test and validate the proposed balance algorithm when balancing data in each cluster.
- Testing and validity of the integration deepening with the above algorithm in detecting malicious traffic over the Network.

METHOD

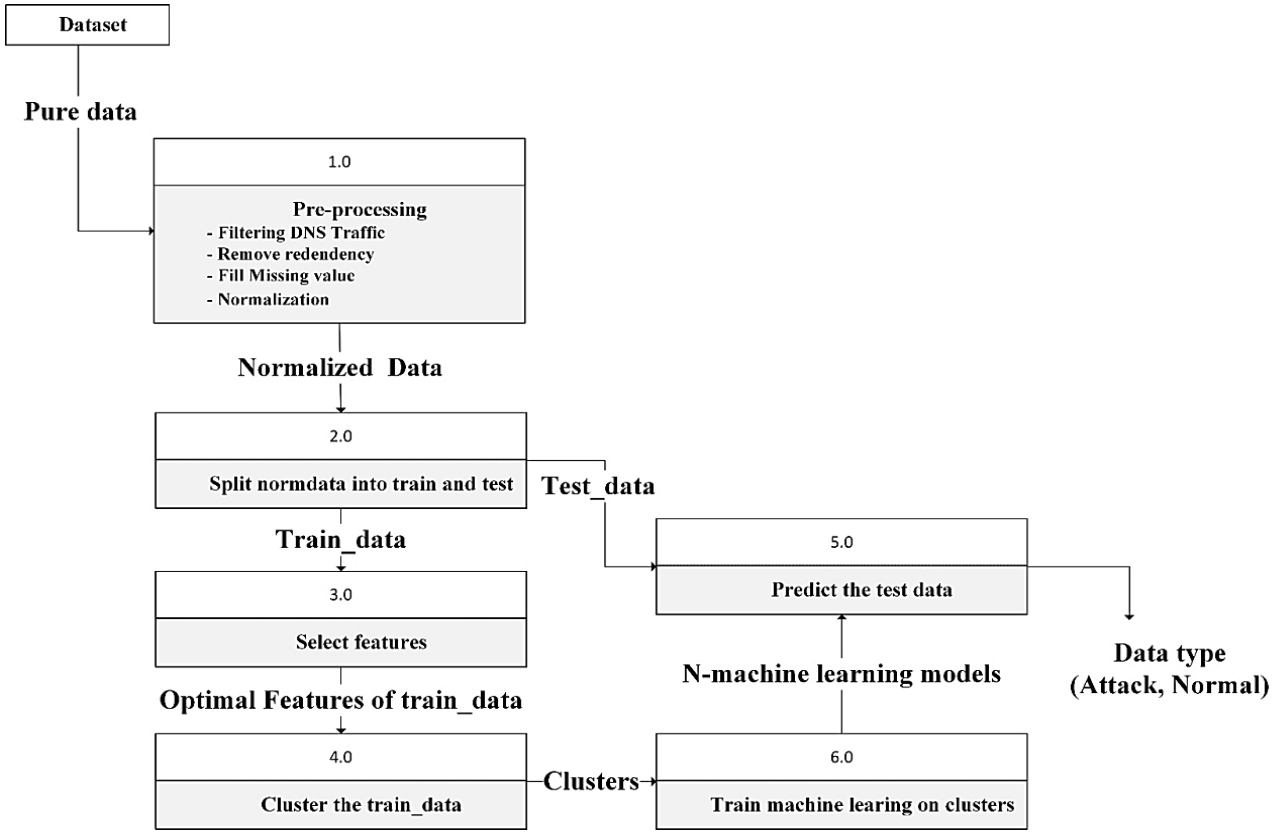


Figure 3. The basic architecture of the proposed model

The process starts with data preparation, which involves collecting, cleaning, and formatting the data. Next, preprocessing steps include normalizing, extracting features, and splitting data. These steps are essential

to ensure the data is in the correct format and structure for the rest of the process. Figure 3 shows the basic steps of the proposed model.

Preprocessing

This step consists of four basic steps that result in preparing the data to be worked on using artificial intelligence algorithms.

Step 1 Remove redundancy

Identifying and removing duplicate or near-duplicate records within the dataset is essential. Redundant entries can distort the model's understanding of the data by overemphasizing specific patterns, leading to skewed or biased results. When duplicates remain, the model may appear to perform better than it does because it effectively "learns" the same information multiple times. By systematically eliminating these redundancies, we ensure that each observation carries equal weight, thus providing a cleaner, more representative dataset and supporting a fairer, more accurate model.

Step 2 Fill Missing value

Addressing missing values is crucial for maintaining the integrity of the dataset. Depending on the nature and extent of the missing data, you may either remove records with large portions of missing information or impute (estimate) those values. A standard and straightforward imputation method is mean-based imputation. Suppose a feature has valid (non-missing) observations: x_1, x_2,…, x_n. The mean $\bar{x}$ of these valid observations is computed as shown in equation 1:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad (1)$$

For each missing entry x_m in the same feature, you then replace it with $\bar{x}$ as shown in equation 2:

$$x_m = \bar{x} \quad (2)$$

This ensures the dataset remains the same size while providing a reasonable central value estimate for missing observations. However, alternative methods (e.g., median or mode imputation, regression-based imputation, or advanced model-based approaches) may be more appropriate if missing values comprise a substantial portion of the dataset or the data distribution is skewed. In cases where the missing data is extreme and may introduce bias even after imputation, removing those records entirely can be more effective, provided the removal does not compromise the dataset's representativeness.

Step 3 Normalization

Normalization using the standard deviation method is essential in preparing numerical features for machine learning models. This method, often called z-score normalization, standardizes the data by centering it around zero with a standard deviation of one. The equation 3 used for z-score normalization:

$$x_{norm} = \frac{x - \mu}{\sigma} \quad (3)$$

Where x is the original value of the feature, μ is the mean of the feature values, and σ is the standard deviation of the feature values.

Step 4 Split into train and test

The dataset must be divided into training, validation, and test sets to assess the model's performance effectively. The proposed model divides the dataset into 70 % training, 10 % validation, and 20 % testing.

Modified DBSCAN with Intensity-Based Post-Processing (M-DBSCAN**)**

While the standard DBSCAN algorithm can automatically identify dense regions and label outliers, it offers limited options for refining the resulting clusters once the initial partitioning is complete. In this work, we propose a modified DBSCAN procedure that applies an additional intensity-based post-processing step to reassign data points between clusters, thereby improving overall cluster coherence.

The following are the main steps of the proposed M-DBSCAN:

Step 1 Initial Clustering with DBSCAN

The proposed MD-IP algorithm begins by applying DBSCAN with user-specified parameters and minSamples.

https://doi.org/10.56294/dm2025739

Each data point assigned to one of several clusters (llabeledby non-negative integers) or classified as noise (label -1). Formally, for every point x_i, DBSCAN checks whether (equation 4) to determine whether x_i is a core point, or whether it should be treated as part of a border cluster or noise.

$$\left|\{x_j \mid d(x_i, x_j) \leq \epsilon\}\right| \geq \text{minSamples} \quad (4)$$

Step 2 Calculation of Cluster Intensity

Once clustering is complete, we compute an intensity measure for each identified cluster C_k (ind (C_k)). The intensity of a cluster is defined as data-based separation and cluster correlation, as shown in equation 5.

$$D(C_k) = \frac{1}{m \times n}\left(\sum_{j=1}^{n}\sum_{i=1}^{m}(\|x_j - \mu_j\|) - \|x_j - \mu_i\|\right) \quad (5)$$

Where: μ_k is mean of the cluster C_k. Equation 6 calculates the μ_k .

$$\mu_k = \frac{1}{n}\sum_{i=1}^{n} x \quad (6)$$

$$Cor(C_k) = \frac{1}{m \times n}\sum_{j=1}^{n}\sum_{i=1}^{m}\left(\frac{\|x_j - \mu_j\| \times \|x_j - \mu_i\|}{\sqrt{\|x_j - \mu_j\|^2 \times \|x_j - \mu_i\|^2}}\right) \quad (7)$$

$$ind(C_k) = D(C_k) + Cor(C_k) \quad (8)$$

Clusters with a higher average distance to their mean have lower density and are considered less “intense” in cohesive structure.

Step 3 Redistribution of Points from Low-Intensity Clusters

Algorithm 1 (pseudocode below) checks whether any cluster’s intensity is below a specified threshold τ. If a cluster C_{low} fails to meet this intensity threshold, each point $x_i \in C_{low}$ is examined and moved to the nearest cluster that satisfies. If a cluster C_{low} fails to meet this intensity threshold, each point $x_i \in C_{low}$ is examined and moved to the nearest cluster that satisfies this, as shown in equation 9.

$$\text{Intensity}\left(C_{\text{high}}\right) \geq \tau \quad (9)$$

The distance measure used here is the Euclidean distance between the point x_i and the centroid μ_k of each potential “high-intensity” cluster C_{high}. This procedure mitigates the problem of having multiple small, scattered, or loosely connected clusters by consolidating them into more coherent clusters. After redistributing these points, the intensities are recalculated, and the process repeats until all clusters have an intensity above τ or no further improvement is possible.

Algorithm 1

Modified DBSCAN with Intensity-Based Redistribution (M-DBSCAN).

1 Perform standard DBSCAN on scaled dataset with parameters ϵ and minSamples.
2 Compute the intensity of each non-noise cluster (equation 8)
3 While any cluster’s intensity is below threshold τ:
  a. Identify a cluster C_{low} with Intensity (C_{low})<τ.
  b. For each point x_i in C_{low}:
  i. Find cluster C_{high} among the other clusters such that $\|x_i - \mu_{high}\|$ is minimized and Intensity (C_{high})≥τ.
  ii. Reassign x_i to C_{high} if such a cluster exists.
  c. Recompute all cluster intensities.
4 Output: Final clusters.

Train machine learning on clusters

During this stage, machine learning algorithms are individually trained on each cluster, allowing customization that reflects the unique characteristics of each data subset. This approach enhances machine learning effectiveness by adapting models to specific patterns within each cluster, leading to a deeper and more accurate understanding. Furthermore, deep learning algorithms excel at further separation within a single cluster, where data are more similar. Their ability to extract complex features and handle large datasets enables finer differentiation of similar data points. Figure 4 shows the framework of training CNN independently on each cluster.

https://doi.org/10.56294/dm2025739

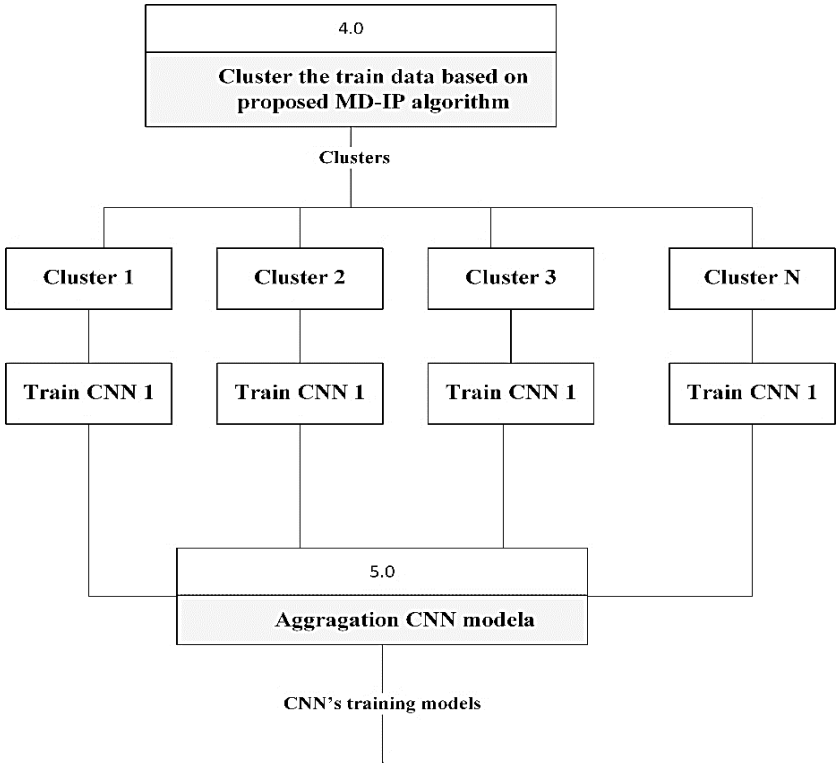


Figure 4. Framework of training CNN on clusters

Proposed CNN architecture

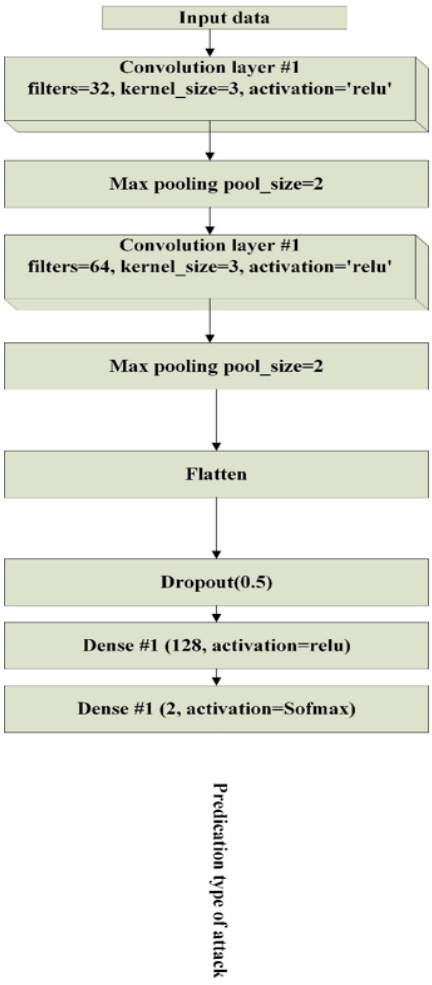


Figure 5. Proposed CNN architecture

https://doi.org/10.56294/dm2025739

In this CNN architecture for DNS intrusion attack prediction, we begin with an Overview and Motivation that highlights how large volumes of DNS traffic data can be processed to identify malicious behavior by leveraging the spatial (or spatiotemporal) feature-learning capability of CNNs. Next, the Input Representation phase involves data preprocessing (such as converting DNS traffic into a matrix or tensor format) and normalization or scaling, ensuring continuous features are standardized for efficient training. The Convolutional Layers then extract hierarchical features in two stages: first, a convolution with 32 filters (kernel_size=3, ReLU activation) detects initial localized patterns, followed by max pooling (pool_size=2) to reduce dimensionality and provide a degree of invariance; a second convolution, now with 64 filters (again kernel_size=3 and ReLU activation), learns more abstract features, and another max pooling layer further condenses these representations. Finally, in Flatten and Fully Connected Layers, the output is reshaped into a 1D vector, passed through a 50 % dropout to mitigate overfitting, and fed into a dense layer with 128 ReLU-activated units, aggregating high-level features. The network concludes with a final dense layer (2 units, softmax activation) that outputs the probability of normal versus malicious DNS traffic.

Predict the test data

The prediction process in the proposed model-based membership function considers the test point's relation to the cluster data and the prediction probability in overall CNNs.

A cluster-based membership function (which captures how well the test point x aligns with the data belonging to a specific cluster k) as shown in equation 10.

$$P_{x,k} = \frac{\sum_{i=1}^{n} e^{-\frac{\|x-y_i\|^2}{2\sigma_1^2}} + e^{-\frac{\|x-\mu_k\|^2}{2\sigma_2^2}}}{2} \quad (10)$$

Where σ_1^2 is the variance between point x and data of cluster k, and $2\sigma_2^2$ is the variance between point x and the centre of cluster k and both>0.

$$M_{x,k} = \sim (argmin_{P_x} + P_{x,k} (argmax_{P_x} - argmin_{P_x})) \quad (11)$$

The CNN-predicted probability (i.e., how likely CNN thinks x belongs to cluster k) as shown in equation 12.

$$Prediction_x = argmax (CNN_k * M_{x,k}) \quad (12)$$

RESULTS

This section discusses the experiments regarding the proposed models:

Evaluate the proposed Clustering (M-DBSCAN)

This section evaluates the proposed Modified DBSCAN (M-DBSCN) in two aspects: its comparison with the original DBSCN and the number of recent dynamic clustering models.

Table 1 illustrates the comparison of the M-DBSCAN and standard DBSCN in terms of four evaluation metrics: the number of clusters identified by each method, Adjusted Rand Index (ARI), Normalized Mutual Information (NMI) completeness, and V-measure.

ARI, NMI, completeness, and V-measure all range from 0 to 1 (with ARI also potentially going as low as -1, but typically staying above 0 in practice), and higher values in each indicate that the discovered clusters match the true class labels more closely. ARI focuses on correctly grouping and separating data relative to true labels, NMI measures how much information is shared between clusters and labels, completeness checks if each valid class is assigned to a single cluster, and V-measure combines completeness and homogeneity (ensuring each cluster mainly contains a single class). The number of clusters has no "ideal" universal value; it should be interpreted relative to the data and analysis goals.

Table 1. Compare the performance of the proposed M-DBSCAN and original DBSCAN

Method name	No. clusters	ARI	NMI	Completeness	V_Measure
DBSCAN	90	0,13	0,27	0,16	0,26
M-DBSCAN	4	0,36	0,29	0,23	0,29

The original DBSCAN produces 90 clusters, which suggests very fine-grained partitioning or potentially many small clusters. Meanwhile, the Modified DBSCAN produces 4 clusters, which indicates a more consolidated or less fragmented clustering. Generating fewer clusters may mean stricter density thresholds and an improved

parameter-setting procedure.

The ARI is higher for Modified DBSCAN (0,36) compared to Original DBSCAN (0,13). A higher ARI suggests that the Modified DBSCAN clusters align better with the reference labels.

NMI for Original DBSCAN is 0,26, while for Modified DBSCAN, it is 0,29. Although the difference is not as significant as with ARI, the Modified DBSCAN still shows an improvement. NMI is bounded between 0 and 1, and a higher NMI indicates a better agreement between the cluster assignments and the ground truth relative to the total information shared.

Completeness is higher for Modified DBSCAN (0,23) than for Original DBSCAN (0,16). The Completeness measures whether all points of a given valid class are assigned to the same cluster. Improved completeness suggests Modified DBSCAN keeps more "true class" members together.

V-measure is also higher in Modified DBSCAN (0,29) versus Original DBSCAN (0,26). V-measure is a harmonic mean of homogeneity and completeness. Again, a higher value indicates better overall cluster quality in capturing the homogeneity (each cluster contains only members of a single class) and completeness (all members of a class are assigned to the same cluster).

The comparison suggests Modified DBSCAN performs better across all metrics, indicating higher agreement with ground-truth labels (where applicable) and forming more cohesive and comprehensive clusters. It is particularly noteworthy that it manages to do this with far fewer clusters, implying a more robust clustering solution that likely generalizes better and is easier to interpret.

Table 2. Compare the performance of the proposed M-DBSCAN with state-of-the-art models

Method Name	No. Clusters	Ari	Nmi	Completeness	V_Measure
DECS[22]	8	0,28	0,12	0,19	0,23
EDFC[13]	13	0,19	0,27	0,16	0,22
HCMD[21]	10	0,21	0,12	0,14	0,17
M-DBSCAN	4	0,36	0,29	0,23	0,29

Overall, M-DBSCAN delivers the best performance across all reported metrics (ARI=0,36, NMI=0,29, Completeness=0,23, and V-Measure=0,29) while producing the fewest clusters (4). In comparison, DECS forms 8 clusters with a moderately high ARI (0,28) but a relatively low NMI (0,12), indicating less shared information with the proper labels. EDFC has the highest number of clusters (13) and achieves a competitive NMI (0,27), but its ARI (0,19) falls behind other methods. Meanwhile, HCMD creates 10 clusters with moderate-to-low performance on all metrics (e.g., ARI=0,21, NMI=0,12). These results suggest that M-DBSCAN captures better alignment with ground truth (reflected by higher ARI and NMI) and offers stronger completeness and overall V-measure despite clustering the data into fewer groups.

Evaluate model detection performance

In this section, the proposed model of the balancing data is evaluated and compared with standard machine learning and deep learning algorithms in the detection classification report.

Table 3 illustrates the results of the compression of the proposed model with the standard machine learning algorithms and deep learning. The comparative performance of various machine learning algorithms—KNN, Naïve Bayes (NB), Decision Trees (DT), Support Vector Machines (SVM), Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), and the proposed model—is summarized in terms of precision, recall, F1-score, and overall accuracy for both benign and DNS-over-HTTPS (DoH) traffic classification. Naïve Bayes (NB) exhibits markedly lower performance, with a particularly poor precision for the benign class (9,72 %) and insufficient recall for DoH (37,75 %), resulting in the lowest accuracy (65,80 %) among all tested methods. This outcome suggests that NB's simplifying assumptions regarding feature independence do not capture the complex, potentially high-dimensional relationships needed to distinguish effectively between regular and encrypted malicious traffic.

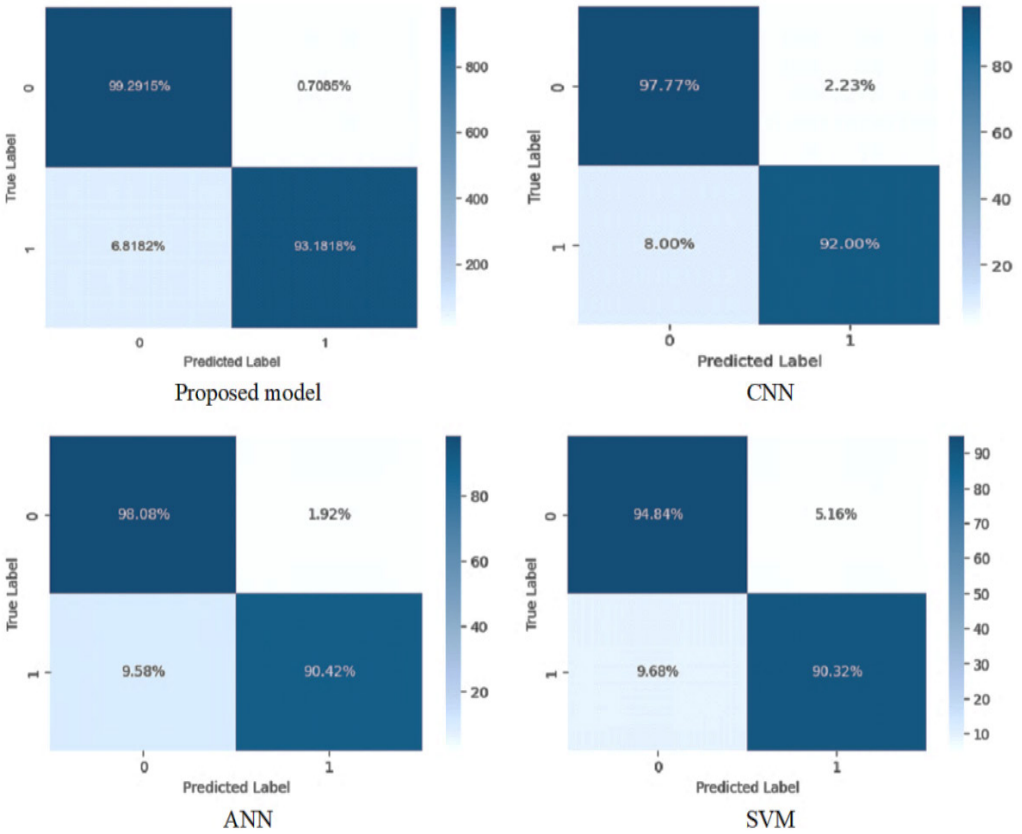
In contrast, KNN, DT, SVM, ANN, and CNN each show more robust and relatively balanced performance profiles, with accuracy values in the mid-90 % range. For instance, Decision Trees achieve 94,80 % accuracy, coupled with strong F1-scores above 94 % for both classes, indicative of an effective partitioning strategy. Similarly, CNN maintains an accuracy of 94,59 % while achieving F1 scores around 95 % for benign and DoH data, reflecting its capacity to learn spatial or local correlations in feature representations. Notably, the proposed model outperforms all baselines, boasting the highest accuracy (96,38 %) and exhibiting markedly high precision, recall, and F1-scores for both classes (e.g., 99,29 % recall for benign and 99,26 % precision for DoH). This superior performance is likely attributable to an enhanced ability to model the nuanced traffic

characteristics of DoH connections, possibly through advanced feature engineering, architecture design, or hyperparameter tuning. As a result, the proposed framework mitigates the critical false negatives and false positives that often plague network security systems, offering a more reliable tool for practical DoH detection and network defense.

Table 3. Result of the proposed model with machine learning and deep learning algorithms

Model	Class Label	Precision	Recall	F1-Score	Model Accuracy
KNN	Benign	91,35	97,27	94,22	94,10
	DoH	97,16	91,01	93,98	
NB	Benign	9,72	94,53	73,19	65,80
	DoH	87,61	37,75	52,76	
DT	Benign	92,33	97,57	94,88	94,80
	DoH	97,49	92,09	94,72	
SVM	Benign	90,53	94,84	92,63	92,55
	DoH	94,72	90,32	92,46	
ANN	Benign	90,90	98,08	94,35	94,20
	DoH	97,97	90,41	94,04	
CNN	Benign	92,12	97,39	95,62	94,59
	DoH	97,01	91,67	95,38	
Proposed model	Benign	93,43	99,29	96,27	96,38
	DoH	99,26	93,18	96,13	

Figure 6 shows the result of the confusion matrix of competition algorithms. The confusion matrices for the Proposed Model, CNN, ANN, SVM, KNN, Naïve Bayes (NB), and Random Forest (RF) highlight significant differences in their performance when classifying benign (Class 0) and attack (Class 1) traffic.



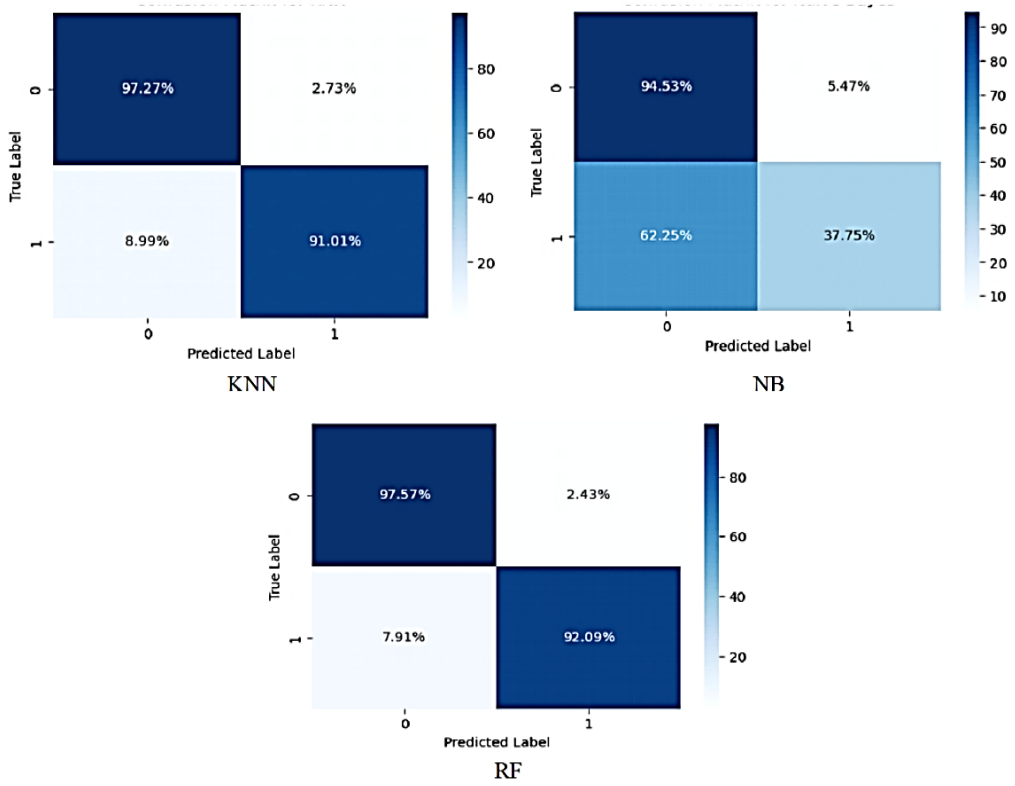


Figure 6. Confusion matrices of the competition algorithm (a-Proposed algorithms-CNN, C-ANN, D-SVM , E-KNN, I-NB, and G-RF

The Proposed Model demonstrates the most robust performance, achieving a recall of 93,18 % for Class 1 and a false-positive rate of only 0,71 % for Class 0, making it highly effective for detecting attacks with minimal misclassification. CNN follows closely with a Class 1 recall of 92,00 % and a false-positive rate of 2,23 %, reflecting a firm but slightly less balanced performance than the Proposed Model. ANN achieves a Class 1 recall of 90,42 % and a false-positive rate of 1,92 %, showing moderate effectiveness but underperforming compared to CNN and the Proposed Model. Conversely, SVM exhibits the weakest performance among these models, with a Class 1 recall of 90,32 % and the highest false-positive rate of 5,16 %, making it less suitable for high-precision and reliability scenarios. Among the traditional machine learning models, RF demonstrates strong and balanced performance, achieving 92,09 % recall for Class 1, a 7,91 % false-negative rate, and a low false-positive rate of 2,43 %, making it more reliable than KNN and NB. KNN achieves a Class 1 recall of 91,01 % but with an 8,99 % false-negative rate and a false-positive rate of 2,73 %, indicating decent but less optimal performance. NB, however, performs poorly, with a Class 1 recall of only 37,75 % and a high 62,25 % false-negative rate, severely limiting its utility in detecting attacks. The Proposed Model is the most effective for detecting benign and attack traffic, with CNN and RF as the best performers. SVM, KNN, and especially NB exhibit varying degrees of limitation in classification accuracy.

CONCLUSIONS

This proposal presented a hybrid model for detecting malicious traffic on networks. The proposed model addresses critical challenges in detecting and preventing such attacks, including high-dimensional datasets, data imbalance, and evolving attack strategies. By integrating metaheuristic optimization and machine learning techniques, the model significantly improved detection accuracy, false-positive reduction, and scalability.

Dynamic Clustering: to cluster data adaptively, a modified DBSCAN algorithm with intensity-based post-processing was employed. This technique ensured better segmentation and classification by managing data imbalance and reducing the overlap between standard and attack data. A tailored Convolutional Neural Network (CNN) architecture was introduced to learn complex patterns within the clustered data. This step further enhanced detection accuracy and minimized false-positive rates.

BIBLIOGRAPHIC REFERENCES

1. Hadi SM, Alsaeedi AH, Al-Shammary D, Alkareem Alyasseri ZA, Mohammed MA, Abdulkareem KH, et al. Trigonometric words ranking model for spam message classification. IET Networks. 2022. https://doi.org/10.1049/ntw2.12063

2. Hammi B, Zeadally S, Nebhen J. Security threats, countermeasures, and challenges of digital supply chains. ACM Computing Surveys. 2023., https://doi.org/10.1145/3588999

3. Schmitt M. Securing the Digital World: Protecting smart infrastructures and digital industries with Artificial Intelligence (AI)-enabled malware and intrusion detection. Journal of Industrial Information Integration. 2023;36:100520.https://doi.org/10.1016/j.jii.2023.100520

4. Hart M, Dave R, Richardson E. Next-Generation Intrusion Detection and Prevention System Performance in Distributed Big Data Network Security Architectures. International Journal of Advanced Computer Science and Applications. 2023;14(9).DOI:10.14569/IJACSA.2023.01409103

5. Geng J, Wang J, Fang Z, Zhou Y, Wu D, Ge W. A survey of strategy-driven evasion methods for PE malware: Transformation, concealment, and attack. Computers & Security. 2024;137:103595. https://doi.org/10.1016/j.cose.2023.103595Manickam S, Nuiaa RR, Alsaeedi AH, Alyasseri ZAA, Mohammed MA, Jaber MM. An enhanced mechanism for detection of Domain Name System-based distributed reflection denial of service attacks depending on modified metaheuristic algorithms and adaptive thresholding techniques. IET Networks. 2022:1-13. https://doi.org/10.1049/ntw2.12043

7. Al-E'mari S, Anbar M, Sanjalawe Y, Manickam S, Hasbullah I. Intrusion Detection Systems Using Blockchain Technology: A Review, Issues and Challenges. Computer Systems Science & Engineering. 2022;40(1). 10.32604/csse.2022.017941

8. Hasbullah I. Intrusion Detection Systems Using Blockchain Technology: A Review, Issues and Challenges. DOI:10.32604/csse.2022.017941

9. Balogun BF. An Enhanced Network Anomaly Intrusion Detection System Using Dimensionality Reduction Approach and Residue Number System: Kwara State University (Nigeria); 2023. Balogun BF. An Enhanced Network Anomaly Intrusion Detection System Using Dimensionality Reduction Approach and Residue Number System: Kwara State University (Nigeria); 2023.

10. Liu Q, Li P, Zhao W, Cai W, Yu S, Leung VC. A survey on security threats and defensive techniques of machine learning: A data driven view. IEEE access. 2018;6:12103-17. https://doi.org/10.1109/ACCESS.2018.2805680

11. Ahmed S, Khan ZA, Mohsin SM, Latif S, Aslam S, Mujlid H, et al. Effective and Efficient DDoS Attack Detection Using Deep Learning Algorithm, Multi-Layer Perceptron. Future Internet. 2023;15(2):76. https:// doi.org/10.3390/fi15020076

12. Nuiaa RR, Manickam S, Alsaeedi AH, Alomari ES. A new proactive feature selection model based on the enhanced optimization algorithms to detect DRDoS attacks. Int J Electr Comput Eng. 2022;12(2):1869-80. DOI: 10.11591/ijece.v12i2.pp1869-1880

13. Nuiaa RR, Manickam S, Alsaeedi AH, Al-Shammary DEJ. Evolving Dynamic Fuzzy Clustering (EDFC) to Enhance DRDoS_DNS Attacks Detection Mechnism. International Journal of Intelligent Engineering and Systems. 2022;15(1):509-19. DOI: 10.22266/ijies2022.0228.46

14. Aktar S, Nur AY. Towards DDoS attack detection using deep learning approach. Computers & Security. 2023;129:103251. https://doi.org/10.1016/j.cose.2023.103251

15. Nuiaa RR, Alsaeedi AH, Alkafagi SS, Alfoudi ASD. A Critical Review of Optimization MANET Routing Protocols. Wasit Journal of Computer and Mathematics Science. 2022;1(4). https://doi.org/10.31185/wjcm.94

16. Al Ogaili RRN, Raheem OA, Abdkhaleq MHG, Alyasseri ZAA, Alsaidi SAAA, Alsaeedi AH, et al. AntDroidNet Cybersecurity Model: A Hybrid Integration of Ant Colony Optimization and Deep Neural Networks for Android Malware Detection. Mesopotamian Journal of CyberSecurity. 2025;5(1):104-20. DOI: https://doi.org/10.58496/MJCS/2025/008

17. Abd Aliwie, A. N. (2025). Conversational Silence in Harold Pinter's The Birthday Party: A Pragmatic Perspective. International Journal of Arabic-English Studies. https://doi.org/10.33806/ijaes.v25i2.860

18. Shafi I, Chaudhry M, Montero EC, Alvarado ES, Diez IDLT, Samad MA, et al. A Review of Approaches for Rapid Data Clustering: Challenges, Opportunities and Future Directions. IEEE Access. 2024. DOI: 10.1109/ACCESS.2024.3461798

19. Wani AA. Comprehensive analysis of clustering algorithms: exploring limitations and innovative solutions. PeerJ Computer Science. 2024;10:e2286. DOI:10.7717/peerj-cs.2286

20. Alsaeedi AH, Hadi SM, Alazzawi Y. Adaptive Gamma and Color Correction for Enhancing Low-Light Images. International Journal of Intelligent Engineering & Systems. 2024;17(4). DOI: 10.22266/ijies2024.0831.15

21. Alfoudi AS, Aziz MR, Alyasseri ZAA, Alsaeedi AH, Nuiaa RR, Mohammed MA, et al. Hyper clustering model for dynamic network intrusion detection. IET Communications. 2022. https://doi.org/10.1049/cmu2.12523

22. Hadi SM, Alsaeedi AH, Nuiaa RR, Manickam S, Alfoudi ASD. Dynamic Evolving Cauchy Possibilistic Clustering Based on the Self-Similarity Principle (DECS) for Enhancing Intrusion Detection System. International Journal of Intelligent Engineering & Systems. 2022;15(5). DOI: 10.22266/ijies2022.1031.23

FINANCING

The authors did not receive financing for the development of this research.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

AUTHORSHIP CONTRIBUTION

Conceptualization: Ola Ali Obead, Hakem Beitollahi.
Data curation: Ola Ali Obead, Hakem Beitollahi.
Formal analysis: Ola Ali Obead, Hakem Beitollahi.
Research: Ola Ali Obead, Hakem Beitollahi.
Methodology: Ola Ali Obead, Hakem Beitollahi.
Project management: Ola Ali Obead, Hakem Beitollahi.
Software: Ola Ali Obead, Hakem Beitollahi.
Supervision: Ola Ali Obead, Hakem Beitollahi.
Validation: Ola Ali Obead, Hakem Beitollahi.
Display: Ola Ali Obead, Hakem Beitollahi.
Drafting - original draft: Ola Ali Obead, Hakem Beitollahi.
Writing - proofreading and editing: Ola Ali Obead, Hakem Beitollahi.