Received: August 9, 2023. Revised: September 8, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Hybrid Model-Based Cauchy and Machine Learning Algorithms for IoT-Intrusion Detection System

Qassim Abd A. Hadi[1] **Ali Saeed D. Alfoudi**[1,2]* **Ahmed M. Mahdi**[1,3]

[1]*College of Computer Science and Information Technology, University of Al-Qadisiyah, Iraq*
[2]*Liverpool John Moores University, College of Computer Science, United Kingdom*
[3] *The University of Szeged, College of Science and Informatics, Hungary*
* Corresponding author's Email: a.s.alfoudi@qu.edu.iq

Abstract: The internet of things (IoT) cybersecurity presents a crucial challenge in our daily lives. An intrusion detection system (IDS) is valuable for protecting IoT data against malicious attacks. Moreover, intrusion datasets are often imbalanced in the number of attacks, increasing the bias in machine learning towards classes with high frequency. Normally, this case affects a training model's performance and ability to make a correct prediction. This paper presents a hybrid model that merges dynamic evolving cauchy clustering (DECS) with ranking classification. The DECS model operates based on the self-similarity principle, strategically distributing data into clusters to counteract the impact of imbalanced data. Furthermore, the rank classification algorithm predicts classes for new attacks. The NF-ToN-IoT dataset was used to test the validity performance of the proposed model and compared with standard machine learning algorithms (K-nearest neighbours, random forest, and decision tree). The proposed model outperforms standard cauchy clustering regarding mean square error (MSE), exhibiting a noteworthy reduction to 0.0534. Furthermore, the silhouette score, which indicates clustering quality, notably improved, reaching 0.4707. Additionally, the proposed model attained an accuracy rate of 67.77% and an F1-score of 76.52%, while the standard Random Forest achieved better accuracy at 66.34% and an F1-score of 68.48%.

Keywords: Machine Learning (ML), DECS, IoT, IDS, Anomaly-IDS, zero-day attack, stream mining.

1. Introduction

The internet of things (IoT) is a network of interconnected devices that communicate with each other through the internet. These devices collect data, undergo analysis and processing operations, and receive back decisions, resulting in significant savings in effort, time, and cost [1]. Additionally, the (IoT) is widely recognised as one of the most crucial advancements in our daily lives, primarily because of its significance and the confidentiality of the data it handles. The concept of security was introduced, and it is now widely recognised as one of the most important tools for preventing unauthorised access to a network [2]. One of the protection mechanisms that are utilised in the internet of things (IoT) is known as the intrusion detection system (IDS).

The process of tracking and monitoring data flow over a network is called an intrusion detection system (IDS).

Additionally, the identification of attacks and unauthorised cases, as well as the transmission of alerts and reports to the administrator of the network [3]. As stated in [4], its mission begins as soon as the data enters the network to begin the process of monitoring and data tracking. The position of the intrusion detection system (IDS) is located directly after the firewall, as shown in Fig. 1. The intrusion detection system performs its functions in real-time and continues to operate continuously to analyse data and discover records that appear to be suspicious.

It's worth mentioning that signature intrusion detection system (SIDS) and anomaly intrusion

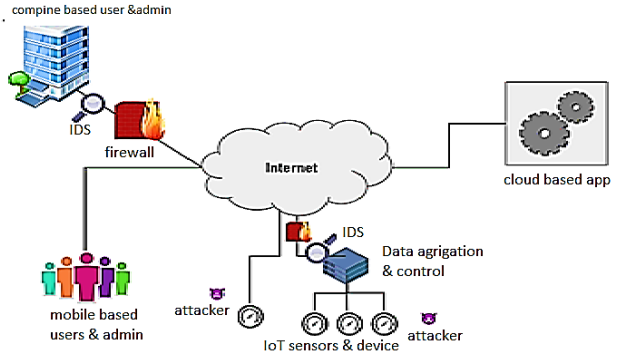

Figure. 1 A scenario describes the relationship between IDS and IoT [8]

detection system (AIDS) are the two basic methods used by IDS to detect attacks. This intrusion detection system (IDS) matches network traffic to a known database to detect attacks and threats. It then compares the network traffic to signatures to identify potential threats. One of the limitations of this sort of system is that it can detect existing dangers effectively; however, it cannot recognise new and emerging threats that were not anticipated in advance, which is one of the system's primary weaknesses [5].

Accordingly, this article will primarily focus on the second type of intrusion detection system, anomaly-based IDS. The anomaly-based IDS detects threats that deviate from normal behaviour and utilises machine learning algorithms to identify and adapt to abnormal patterns. Furthermore, this system generates a model that matches network traffic or activity, enabling the identification of potential threats indicative of an attack [6, 7] argued that this type of system is crucial for detecting threats and unwanted traffic that were previously unknown. However, one of its drawbacks is that it may generate false expectations when the network activity or traffic deviates from the established model.

In the (IoT) era, the proliferation of large communication devices is rapidly increasing. Ensuring communication security in an IoT environment, which previously relied on IDS, presents various challenges and opens up prospects for future research endeavours. Undoubtedly, the field of IDS has witnessed extensive research, yet numerous vital aspects warrant further exploration and development [9]. Therefore, Intrusion Detection Systems must become more accurate and capable of detecting a wide array of intrusions while minimising false alarms and addressing other associated challenges [8]. Numerous challenges can be encountered in this field, including but not limited to scalability, heterogeneity, a dynamic environment, anomaly detection, false positives, imbalanced data, and handling unknown attacks. Malicious threats and network intrusions often possess unique spatial and temporal advantages, making them significant challenges that constantly threaten network security. Many network intrusion detection systems (NIDSs) that rely on machine learning algorithms often require a substantial number of high-quality samples for effective training. The relationship between network traffic and the corresponding traffic categories must be accurately determined to achieve reliable results.

1.1 Motivation

This paper will primarily focus on the two most significant challenges, which is considered one of the most common problems facing the Internet of Things environment:

1. Imbalanced data and unknown attacks, such as zero-day attacks: The most crucial challenge addressed in this article is the issue of data imbalance in the (IoT) environment. Researchers face the challenge of dealing with diverse and imbalanced data collected from multiple and different devices, posing a significant obstacle to effective intrusion detection in such environments [10].
2. Another challenge that warrants focus is within the intrusion detection system of the IoT. This challenge pertains to a group of attacks that pose a threat to the network, yet they are unknown and undefined in advance. Additionally, the timing of these attacks, like zero-day attacks, cannot be predicted in advance. Consequently, detecting them using traditional methods becomes challenging, making them a crucial area of focus for ongoing research efforts [11].

1.2 contribution

The main contribution of this study is utilising a hybrid model by:

1. To address the challenge of imbalanced data, the dynamic evolving cauchy clustering algorithm (DECS) is employed, leveraging self-similarity to distribute the data. This approach ensures that all data is appropriately allocated to clusters based on similarity.
2. The ranking classification method relies on a voting mechanism, and a novel classifier has been devised that selects the best cluster using probability laws to estimate the degree of cluster belonging to the test data. The proposed model has demonstrated high performance through extensive testing and yielded promising results.

1.1. Evaluation metrics

The evaluation process of the proposed model holds the utmost significance in assessing its accuracy and effectiveness in identifying attacks. To calculate accuracy, recall, precision, and F1-score, a confusion matrix is employed for evaluating the proposed model. By utilising this matrix, researchers can objectively measure the system's performance regarding these crucial metrics. The proposed model demonstrated promising outcomes in terms of accuracy and efficiency in effectively identifying and detecting attacks.

Paper organisation

The organisation of the rest of the paper seems well-structured. Section 2 provides a review of previous studies related to the current work. Section 3 details the proposed solution, and lastly, section 4 showcases the experiment results and analysis of the proposed model.

2. Related work

Despite the maturity of the field of IDS research, current IDS solutions are deemed insufficient for widespread use in IoT deployments. In response to this challenge, we put forth novel approaches for intrusion detection that are adaptable to and capable of overcoming the limitations specific to the IoT environment. Securing devices and networks through the implementation of security measures, such as intrusion detection systems (IDS), is an enduring and crucial topic that has been extensively addressed over time. This method is trustworthy, and for some time, it has been a vital component in protecting networks and devices. Despite developments in technology such as the (IoT), implementing (IDS) has remained an essential step towards achieving a secure network or device.

Traditional IDSs could not fully manage the new and complex security concerns presented as a result of technological improvements, especially the internet of things (IoT). These challenges frequently involve attacks that have not been seen before, which conventional security systems have difficulty detecting. The precision and effectiveness of the suggested method in detecting previously unknown attacks gives it significant potential as a solution to these difficulties, and it offers tremendous promise in this regard. The proposed model addresses the issues of imbalanced data and outliers head-on by using an algorithm called dynamic evolving cauchy clustering (DECS).

Skrjanc et.al [12] highlighted the importance of addressing imbalanced and extreme data, which can give rise to various challenges, one of the most significant being the model's inherent bias towards one category over another. They developed a model that uses cosine-based Cauchy groups, which have successfully performed when dealing with imbalanced data and noise. The suggested model underwent careful analysis and testing using the DARPA data to ensure its accuracy. One of the weaknesses of this study is how it handles the stream clustering of data; specifically, it does not adapt well enough to the changes that take place over time. Because of this limitation, the accuracy with which anomalies and outliers are detected can be compromised, which is especially problematic in constantly changing situations.

Karatas et.al [13] Focused on the problem of data imbalance in datasets, which occurs when class distributions are not uniform and makes it difficult for classifiers to do their jobs. The researchers came up with a solution for this problem called the synthetic minority oversampling (SMOTE) model, which was designed to reduce the number of outliers and noise in the data. However, one of the weaknesses of the proposed model lies in its high complexity, which can impact its practical applicability and scalability. Additionally, the experimental results indicated that SMOTE did not perform optimally for intrusion detection systems (IDS).

Thi-Thu-Huong Le et.al [14] directed their efforts towards enhancing the attack detection performance of IDS using extensive IoT-based IDS datasets, while also offering interpretations for predictions made by machine learning (ML) models. The Shapley additive explanations (SHAP) method is integrated into the eXplainable AI (XAI) framework to provide annotations and interpretations for classification decisions made by DT and RF models. This approach is effective for interpreting the final decision of a group tree approach. Also, it assists cybersecurity experts in rapidly enhancing and evaluating the accuracy of their judgments, guided by the interpretations of the results. A weakness of the proposed model arises when the number of features and data expands, rendering the model's calculation infeasible and posing challenges in its application.

Chiba et al. [15] focused on anomaly depending on NIDS by suggesting a model based on a backpropagation neural network (BPNN) to determine unknown attacks like zero-day attacks. The presented model uses KDD CUP'99 datasets. One weakness of the proposed model is its vulnerability to overfitting, as it tends to focus solely on training data without effectively generalising to unfamiliar datasets. This limitation poses a risk to the system's ability to detect emerging threats. Instances,

where the system encounters novel data noise differing from the training dataset, can weaken the model's accuracy, as such variations are not adequately recognised.

Sarhan et al. [16] produced and utilised several datasets with network traffic information. The idea is to use Netflow's attributes to standardise NIDS datasets. Cisco employs a standard named NetFlow to collect traffic data while it traverses the network. The characteristics gathered through this process have proven effective in pinpointing network attacks. To enhance this, a fresh dataset comprising NetFlow features was generated from PCA files, including NF-ToN-IoT crafted from ToN-IoT. Notably, the accuracy of detecting attack classes with limited training data remains relatively low.

Pu et al. [17] focused on using a hybrid unsupervised machine learning model through two technologies, subspace clustering (SSC) and one-class support vector machine (OCSVM) models, to detect unknown attacks, outliers, and noise. The model's performance was assessed using the dataset, yielding favourable outcomes. However, it did not ascertain the crucial features instrumental in achieving these results. Furthermore, in line with anomaly-based methods, the detection of zero-day attacks entails modelling regular network traffic, thereby flagging deviations from this anticipated abnormal behaviour, which encompasses zero-day attacks. When the anomalies closely align with the data's standard distribution, accurately identifying such anomalies may pose challenges for the one-class support vector machine (OCSVM) algorithm.

Duan et al. [18] introduced an innovative real-time network intrusion detection system (NIDS) approach that addresses these challenges. The proposed model is built upon the foundation of a "dynamic graph neural network (DGNN) ", leveraging semi-supervised learning techniques to capture both the temporal and spatial characteristics of the network traffic flow. The limited inclusion of only 30% of labels in the suggested model is indeed one of its shortcomings. This low proportion of labelled data can significantly impact the accuracy of the findings, as it might lead to insufficient representation of various threats and anomalies in the network.

Table 1 summarises the limitations of related works.

Table 1. Summarise of related works

Ref.	Proposed model	weakness
[12]	Evolving Cauchy Possibility Clustering	A method that balances weight evenly can lower the success of noise reduction in filters that use balanced components, which in turn makes the algorithm work less efficiently.
[13]	SMOTE	The model fails to prioritise low numbers of attacks in detection due to the absence of balanced techniques.
[14]	SHAP	The proposed SHAP method isn't suitable for dynamic changes in data behaviour, as it requires increased computational complexity with growing data behavior, making it unfit for dynamic data changes often seen in intrusion attacks
[15]	BPNN	lacks generalisation in bais to maijority class attacks
[16]	Standard ML classifier	Lack in optimising and enhancing the ability of the IDS to the dynamic change of the malicious behavior over the IoT networks
[17]	-subspace clustering (SSC) -One-Class Support Vector Machine (OCSVM)	Due to its high complexity, the model cannot be integrated into an IoT system. The complexity of the model requires significant computational and memory resources that are not feasible for typical IoT devices with limited resources. Therefore, developing IoT solutions should take a leaner and more resource-efficient approach.
[18]	DGNN	The limitation of DGNNs in IDS is that they may have difficulty distinguishing between different intruder classes. Statistical attack characteristics are often overemphasised while inherent attack topologies are ignored.

3. Proposed methodology

In this section, we investigate the primary techniques employed in the proposed model, as shown in Fig. 2. The approach suggested in this study involves a hybrid model combining clusters and

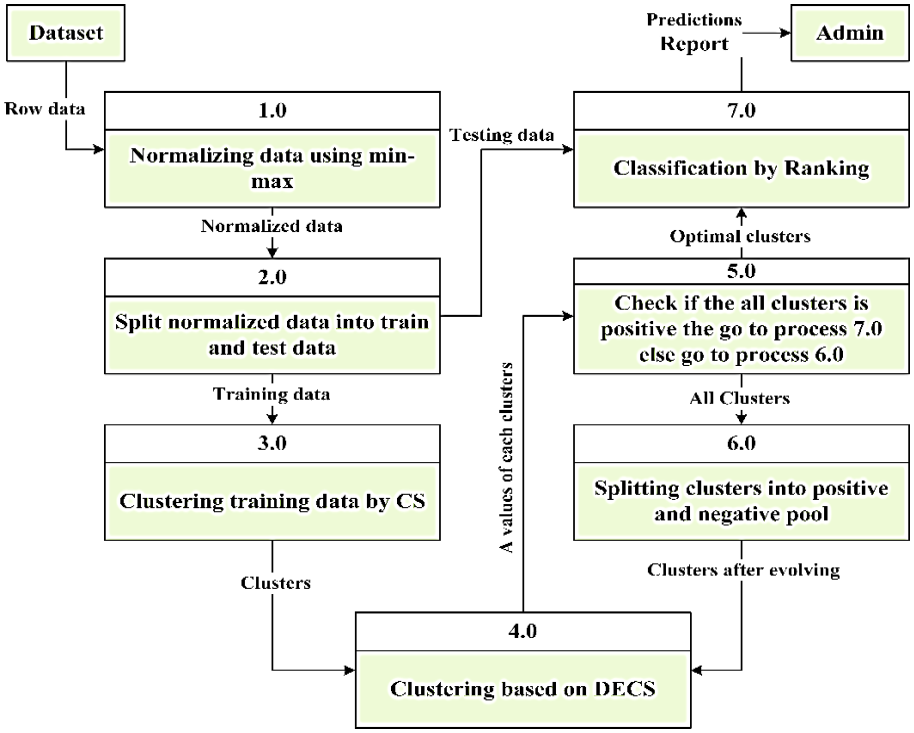

Figure. 2 The main steps of the proposed model

classification methods. Furthermore, clusters are employed to isolate training data that exhibit high similarity. The dynamic evolving cauchy clustering algorithm (DECS) is utilised for this purpose [19]. It relies on self-similarity for data distribution, effectively addressing the issue of imbalanced data. Thus, data distribution across clusters is based on similarity, allowing the clusters to evolve through the clustering algorithm to achieve optimal clustering. Moreover, the process involves training the most suitable clusters using classification algorithms to enhance accuracy. This simplifies the detection of unfamiliar attacks, achieved through ranking classification that relies on a voting mechanism. A new classifier was developed, which chooses the optimal cluster using probability laws related to the extent of cluster affiliation. This estimation is utilised to predict outcomes for test data. The proposed model has been rigorously tested and has yielded impressive results.

3.1 Preparing data

Min-max normalization is a technique utilised during the data preprocessing phase. It involves scaling the data to fit within a designated range, typically [0, 1]. This normalisation process aids in enhancing the prediction process, particularly when the data's range is well-defined and specific. Assuming we have some rows that contain all identical values, the normalisation for this row is zero, and zero values are given to all row values [20]. The formula 1 for min-max normalisation is:

$$\tilde{x} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

A noteworthy characteristic of Min-Max Normalization is its capability to preserve links and associations within the data. In this study, two features from the dataset, namely "IPV4-SRC-ADDR" and "IPV4-DST-ADDR", will be excluded. This decision is due to these features being represented as objects rather than numeric attributes.

3.2 Split normalization data

At this stage, the data used for normalisation has been split into two sets: training data and test data. The separation was based on a percentage of 80% for training data and 20% for test data. However, selecting a high rate of training data can lead to an overfitting problem. This occurs when the classifier becomes biased and too specialised to the training data, resulting in poor performance on unseen test data.

3.3 Trained data by using cauchy clustering algorithm

In this stage, we employed an unsupervised machine-learning algorithm called the Cauchy, a

clustering algorithm. The goal of clustering is to get meaningful information from a collection of data by grouping the data points most similar to one another based on the similarities and differences among those data points. Clustering is an effective method for finding patterns and structures in data that might not be immediately obvious. It enables the identification of complex relationships and hidden insights within the database [21]. Each data point is assigned to a group by the clustering method based on its closeness to other data points in that group and its distance from data points in different groups. Various clustering methods, such as the commonly used Euclidean distance, employ distinct metrics to ascertain proximity. Ultimately, each data point becomes a member of a particular cluster [22]. The Cauchy clustering algorithm, on the other hand, relies on density. It groups dense points within a single cluster.

The clustering begins by establishing the first cluster and assigning the initial data point. Subsequently, the second data point is assessed by comparing it with the point within the existing cluster. If their density is substantial, the second point joins the same cluster. Otherwise, another cluster is created ,and the process continues on all data, thus, ,there are two decisions, either create a new cluster or update the current cluster. The Cauchy density of sample i for cluster j is defined as ρ_i^j for batch data, the density is typically defined as the sum of distances between the current sample, $R(i)$, and all previous samples belonging to the specific cluster, which can be calculated according to Eq (2).

$$\rho_i^j = \frac{\begin{pmatrix} \cos\left(R(x)-\beta_j\right)\alpha_t^2\, N^j \\ +\, cos\left(R(x)-\beta_j\right)N^j\left(R(i)-\beta_j\right)^T \\ *\left(\varepsilon^j\right)^{-1}\left(R(x)-\beta_j\right) \\ +\left(R(x)-\beta_j\right)q(m-1) \end{pmatrix}}{\cos\left(R(x)-\beta_j\right)\alpha_t^2\, N^j} \tag{2}$$

The centre of the j[th] cluster can be defined as β_j and the R_i^j can be represented as follows the i represent the sample and the j represent the j^{th} cluster, according to Eq.(3):

$$\beta_j = \sum_{i=1}^{N^j} \frac{R_i^j}{N^j} \qquad N_j \neq 0 \tag{3}$$

Then calculated, the covariance matrix, according to Eq. (4), can be referred to it by $\varepsilon_{N^j}^j$, which represents the dimension n*n of j^{th} cluster. The covariance matrix measures how much two random variables change together. It is used to calculate the covariance between points of a data matrix.

$$\varepsilon_{N^j}^j = \begin{pmatrix} \frac{1}{N^j-1}\sum_{i=1}^{N^j} \left(R_i^j - \beta_{N^j}^j\right)^{T-1} \\ *\left(-\beta_{N^j}^j + R_i^j\right) \end{pmatrix} \tag{4}$$

When we calculate the density between the point and the points in the cluster, a high density is achieved; in this case, this point is added to the cluster. Thus, the number of cluster elements will increase by one. This can be calculated according to Eq. (5).

$$\beta_{N^j+1}^j = \frac{\beta_{N^j}^j\left(N^j+1\right)+\left(R(x)-\beta_j\right)}{N^j+1} \tag{5}$$

After that, we apply a non-normalisation to the covariance matrix, according to Eq. (6)

$$F_{N^j+1}^j = \begin{pmatrix} F_{N^j}^j + \left(R(x)-\beta_j\right) \\ *\left(R(x)-\beta_{N^j+1}^j\right)^{T-1} \\ *\left(-\beta_j + R(x)\right) \end{pmatrix} \tag{6}$$

Then, according to the Eq. (7), the covariance matrix is calculated:

$$\varepsilon_{N^j+1}^j = \frac{F_{N^j+1}^j}{N^j} \qquad N_j \neq 0 \tag{7}$$

3.4 Clustering evolving

During this stage, we implement an evaluation process for the clusters generated by the clustering algorithm. The aim is to reduce the total number of clusters and identify the optimal grouping of clusters. Clusters exhibiting highly similar characteristics are segregated at this juncture. A threshold is then defined, serving as a criterion for categorising the data into two groups: positive and negative. The degree of similarity is established based on the comparison of the value to the threshold limit. When the value significantly exceeds the threshold limit, it is assigned to the positive group. Conversely, if the value is notably lower than the threshold limit, it is allocated to the negative group. Eq. (8) elucidates this procedural concept. Variables are used that are set during the training phase, through which the dividing line between positive and negative assembly is determined.

$$G_{xi} = \begin{cases} 1 & if\ P_{ir} - \frac{\sum_{j=1}^{n}\|x_i-y_j\|^2}{n-1} \geq 0 \\ -1 & otherwise \end{cases} \tag{8}$$

where: $j \neq i$ the index of the point in the same cluster, Pir the nearest point to xi in another cluster.

At this stage, we redistribute the data in the negative groups to the data in the positive groups according to the following Eq. (9).

$$A_j = \begin{cases} 1 & if \ \frac{1}{m} \sum_{i=1}^{n} \sum_{j=1}^{m} G_{yj} \geq 0 \\ 0 & other\ wise \end{cases} \quad (9)$$

After merging the negative clusters, we do this by finding the distance between the point in the negative pool and the clusters, according to the following Eq. (10).

$$d_{xi} = \frac{\sum_{j=1}^{n} \|x_i - y_j\|^2}{n} \quad (10)$$

Then we take the least distance between the point and its nearest positive cluster, according to the following Eq. (11)

$$cluster_{index} = arg_{min}(d_{(xi)_j}) \quad (11)$$

3.5 Matching the test point with the best clusters:

In this phase, illustrated in Fig. 2, classifiers are trained using the data extracted from the clusters that emerged from the evaluation process. These selected clusters are deemed to be the most optimal. The matching process operates according to the subsequent steps:

- Firstly, we calculate the distance between the test point and collect the points in the clusters to find out the nearest point in the clusters relative to the test point, using the Euclidean distance and according to Eq. (12)

$$Z_{xi} = \frac{\sum_{j=1}^{n} \sqrt{(X_i - X_j)^2}}{n} \quad (12)$$

- After that, we calculate the normalisation of the values resulting from measuring the distance to make it within a range between [0,1] because there may be large distances between one test point and the clusters, and we perform the normalisation process to make the value small, according to the Eq. (13).

$$F_i = \frac{Z_i - Z_{i\,(min)}}{Z_{i\,(max)}} \quad (13)$$

- Then, find the inverse of normalised data, according to Eq. (14).

Table 2. Definition of all notations

Notation	description
$\tilde{x}$	The Normalisation equation
x	The single value in the specific feature
ρ_i^j	The density by using DECS
β^j	The centroid of the cluster
α_t^2	The standard division
$R(i)$	The current sample
Σ^j	The covariance matrix
R(X)	The feature vector of x
R(i)	The feature vector of the ith data point
Nj	The number of data points in the Jth cluster
q	Quantum
m	The dimensionality of the feature vector
Pir	variable that represents the ith point's proximity
n	The total number of points
Gxi	The variable that takes on a value of either 1 or -1
dxi	Euclidean distance between two points(x and y)
argmin	The function returns the argument that minimises the value of the function inside it.
$d_{(xi)j}$	Distance between the data point xi and the centroid of the jth cluster
R_i	The inverse of normalised data
$\check{R}$	Probability for each test point

$$R_i = 1 - F_i \quad (14)$$

- Lastly, using Eq. (15), determine the probability for each test point within each cluster. Subsequently, identify clusters that yield similar predictions and aggregate these predictions. After collecting the guesses, we take the highest value as the final value. We repeat the process for each of the test points.

$$\check{R} = \frac{R}{\sum R_i} \quad (15)$$

Table 2 illustrates the description of all notation of formulas in the proposed model

4. Experimental results

In this section, we will review the description of the database used to analyse the essential results that were extracted from the implementation of the model.

4.1 Data description

The NF-ToN-IoT (Net Flow -ToN-IoT) dataset was used in this study. The Net flow version of the UNSW-ToN-IoT dataset is called NF-ToN-IoT. This dataset has 14 features and 1,379,274 samples, with 1,108,995 (80.4%) attacks flow and 270,279 (1.9%) benign flows [23]. This dataset comprises various types of attacks, including benign, backdoor, DDoS, DoS, injection, MitM, password, ransomware, XSS, and scanning. Queensland university gathered this dataset intending to standardise network-security datasets to achieve interoperability and more comprehensive studies. The original writers, Drs. Mohanad Sarhan, Siamak Layeghy, and Marius Portmann deserve full recognition [16].

4.2 Evaluation metrics:

During this phase, we assess the critical metrics employed in our study to test the performance quality of the proposed model. The following section and Table 3 describe the most significant metrics to evaluate the proposed solution.

4.2.1. Mean square error (MSE):

The MSE is one of the metrics used to measure the quality of clustering results by determining the correlation of data points within each cluster [24]. In this work, MSE is used as follows:

- First, following the execution of the data distribution process across clusters and the extraction of optimal clusters, along with establishing central points for each cluster, the next step involves calculating squared distances. This is accomplished by summing the squared differences between data points [25].
- We sum the squared distances generated by all data.
- Finally, we divide the total resulting from the previous stage by the total number of data and thus result in the average squared distance between the group points and the central point.

4.2.2. Silhouette score:

Silhouette criterion can be calculated by using the following steps:

The silhouette criterion is computed by calculating the silhouette score for all data points within each cluster. This process follows the subsequent steps [26]:

- We calculate the distance between each point and all other points in the cluster, and denote it with a symbol" μ".

Table 3. Evolution metrics

Metrics	formula
MSE	$\frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{m}\lvert X_j-\mu_i\rvert^2$
Silhouette score	$\frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{m}\frac{Xj-\mu k}{\max(Xj,\mu k)}$
Accuracy	$\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{TN}+\mathrm{FN}}$
Detection Rate (DR) or Recall	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$
Precision	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$
F1-score	$2*\frac{Recall*Precision}{Recall+Precision}$

- Next, for each point in the cluster, the average distance between that point and every other point in the closest neighbouring cluster is determined. Its symbol is "Xj" and is known as the "inter-cluster distance".
- Determine each point's silhouette value inside the cluster.

Determine the cluster's average silhouette value for each point.

4.2.3. Accuracy

Accuracy serves as a key metric for evaluating the precision of the proposed model's predictions. It is determined by dividing the count of accurate predictions by the total number of predictions made. Given the unbalanced nature of Internet of Things (IoT) data, relying solely on accuracy may prove inadequate. The preponderance of intrusive data outweighing the occurrences of natural data could lead the accuracy metric to provide misleading results, potentially overlooking the minority of intrusion instances [16].

4.2.4. Detection rate (DR) or recall

A crucial metric, often referred to as sensitivity or recall, assesses the classification model's efficacy. It quantifies the proportion of accurately classified positive cases (True positives) with the entire count of positive instances within the dataset (True positives + False negatives).

4.2.5. Precision

The metric known as positive predictive value (PPV) or precision is an essential measure in the internet of things (IoT) environment. The positive

predictive value (PPV) or precision calculates the percentage of correct cases out of the total number of cases that the model identifies as positive (true positive + false positive). A higher percentage of the positive predictive value (PPV) or precision indicates a lower rate of misidentifying normal cases as attacks, which leads to a lower percentage of false positives.

4.2.6. F1-score

The F1 score is a valuable measure used to assess the performance of the proposed model. By combining the precision and recall metrics, the F1-score achieves a balance in the model's effectiveness, effectively reducing false negatives and false positives.

4.3 Analysis results

This section examines the solutions applied in our study and the corresponding outcomes. The primary concern addressed within the proposed model is the challenge of imbalanced data and outliers. The rationale behind these data and values stems from the characteristics of the internet of things (IoT) environment, which entails streaming and voluminous data. As a result, implementing effective techniques becomes imperative to address this challenge. In the proposed model, firstly, execution preprocessing by using Min-max normalisation to limit the data within a specific range between [-1, 1]. After that, the DECS algorithm was used, which creates clusters according to similarity, and according to Eq. 2. The cosine function was used, which is characterised by the same wide range and a continuous function, Thus including all expected values and in addition to that the range of this function is between [-1, 1]. We conclude from this that the problem of imbalanced data and outliers has been processed.

Also, the second problem that is being addressed is that there is a group of unknown and unspecified attacks in advance due to not knowing the date of their appearance. Therefore, a classifier based on probability was proposed, as it calculates the probability of the formation of the point in all clusters, and the highest percentage of chance is taken, and thus discovering such unknown attacks more precisely.

Fig. 3 shows a remarkable improvement in the total silhouette score for the entire clustering process on the NF-ToN-IoT dataset. Before performing the evolving process for clusters, the silhouette score stood at 0.3185. However, after conducting five iterations of the evolving operations, the silhouette

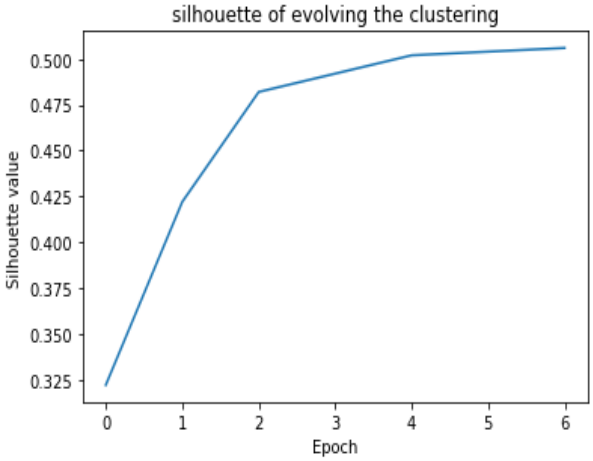

Figure. 3 Silhouette score of the proposed solution on NF-ToN-IoT

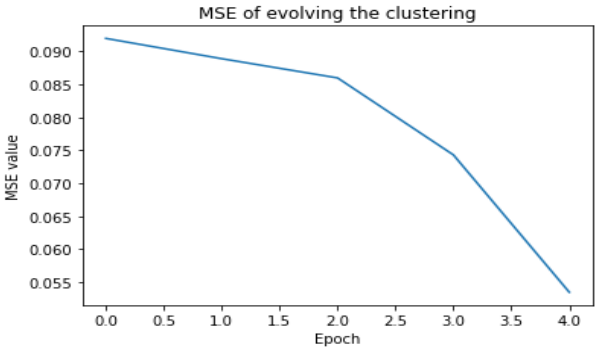

Figure. 4 Mean square error of the proposed solution on NF-ToN-IoT

score significantly increased to approximately 0.4704. The significant improvement in separating similar data within a single cluster, as evidenced by the increased silhouette score, is of great significance for the anomaly.

In Fig. 4, an apparent reduction in the total mean square error (MSE) value can be observed, decreasing the value from its initial state to 0.0909 after performing evolving operations. The updated mean square error (MSE) result reaching 0.0534 indicates a notable decrease and a substantial improvement following the evolving operation.

Table 4 and Fig. 5 illustrates that the proposed solution achieved the highest accuracy compared to other standard machine learning classifiers such as decision tree (DT), random forest (RF), k-nearest neighbors (KNN), and Naive Bayes (NB). Specifically, the proposed model attained a notable accuracy of 67.77%, outperforming the least accurate classifier, NB, which reached 55.51%. Moreover, the proposed model's effectiveness and accuracy are evident in its ability to identify and uncover attacks.

In Fig. 6, concerning the F1-score metric, the proposed solution stands out with the highest value

Table 4. The evaluation metrics of the proposed model with standard classifiers

metrics	DT	NB	RF	KNN	Proposed solution
Accuracy	64.91	55.51	66.43	65.18	67.77
precision	65.13	82.28	69.56	70.4	87.86
recall	64.91	55.51	67.43	65.18	67.77
F1-score	65.02	66.29	68.48	67.69	76.52

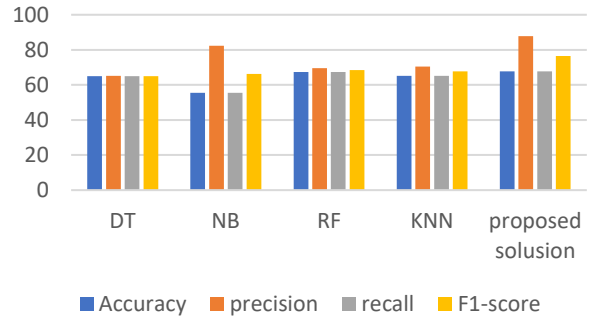

Figure. 5 shows the evaluation metrics of the proposed model and standard classifiers.

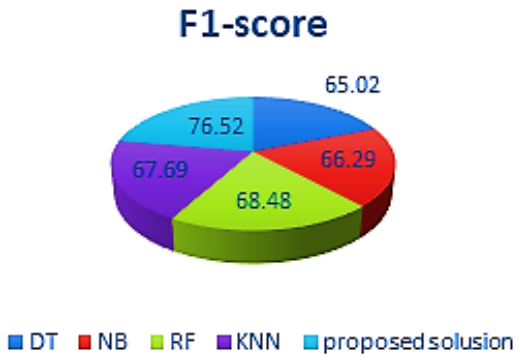

Figure. 6 Comper of the proposed model and standard classifiers in term F1-score

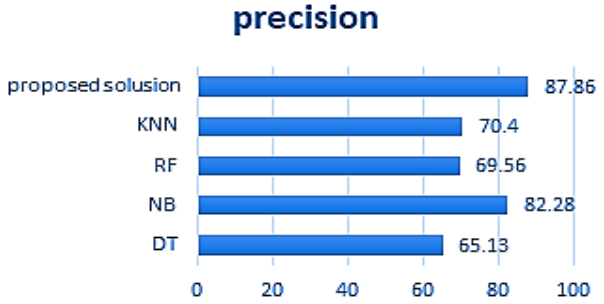

Figure. 7 Comper of the proposed model and standard classifiers in terms of precision

compared to other standard machine learning classifiers. The proposed model achieved an impressive F1-score of 76.52%, while the decision tree (DT) classifier obtained the lowest value, estimated at 65.02%.

In Fig. 7, regarding the precision metric, it is evident that the proposed solution demonstrated the highest precision value compared to other standard machine learning classifiers. The proposed model achieved an impressive precision rate of 87.86%, while the decision tree (DT) classifier obtained the lowest precision, estimated at 65.13%.

Table 5 and Fig. 8 display the prediction outcomes for classes of NF-ToN-IoT using both the proposed model and various ML classifiers. Notably, in the case of the DoS attack, the machine learning classifiers achieved a 96.839% accuracy, with the (NB) classifier leading at this task. Our proposed classifier has achieved a remarkable general accuracy of 99.057%. In the context of MITM attacks, the machine learning classifiers attain a 99.855% accuracy, primarily led by the Naive Bayes (NB) classifier. Notably, our proposed classifier surpasses this with an impressive accuracy of 99.892%. Turning to password attacks, the machine learning classifiers achieve a 62.234% accuracy, also led by the NB classifier. However, our classifier achieves a higher accuracy of 88.204%.

Regarding the XXS attack, the machine learning classifiers attained an accuracy of 88.712%. In contrast, our classifier demonstrates improved performance, achieving an accuracy of 92.619%.

4.4 Compare with other techniques

This section discusses the result of the proposed model compared to other intrusion detection studies.

We observed specific differences when comparing the proposed model with the studies in [16] and [24] over the NF-ToN-IoT dataset. In [24], the authors employed silhouette scores and scatter plots to select optimal clusters. They utilised the k-means clustering technique to distribute the data across the groups based on their distances. Table 6 compares the proposed model to two other studies [16] and [27] on two metrics - Accuracy and F1 score. The proposed solution has the highest accuracy of 67.77% compared to 56.34% for [16] and 67.16% for [27]. The proposed model also has the highest performance in terms of F1-score of 76% compared to 60% for [16] and 63% for [27]. Based on these two metrics, the proposed solution outperforms the other two studies on both accuracy and F1 score.

To prove the validity of the proposed model, we compared two datasets: NSL_KDD and UNSW_NB15. The second dataset, UNSW_NB15, contains a much larger number of data packets, about

Table 5. Comparison results on the NF-ToN-IoT for the proposed model and ML classifiers.

classes	DT	NB	RF	KNN	Proposed solution
Benign	100	100	100	100	99.993
backdoor	99.971	99.964	99.971	99.949	98.644
DDoS	87.682	88.48	88.842	83.898	87.378
DoS	98.506	96.839	98.506	98.441	99.057
injection	74.473	74.966	75.335	74.364	71.362
MITM	99.869	99.855	99.877	99.891	99.892
password	83.303	62.234	84.681	85.05	88.204
Ransomware	100	100	99.993	99.993	99.993
scanning	97.317	95.824	97.586	98.013	98.405
XXS	88.712	92.066	90.075	90.771	92.619

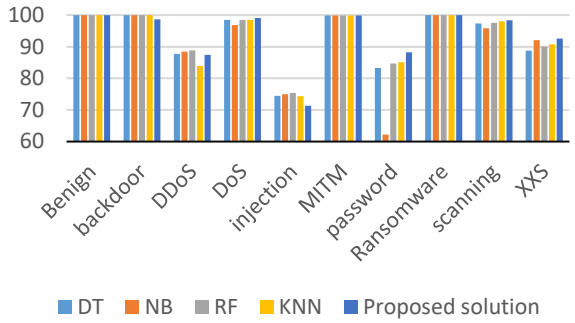

Figure. 8 Shows the comparison between the proposed modle and machine learning classifiers for each attacks detection

Table 6. Comparison between the proposed model and previous studies over NF-ToN-IoT

Sq.	Ref	Metrics	
		DR	F1-score
1	[16]	56.34%	60%
2	[27]	67.16%	63%
3	proposed Model	**67.77%**	**76%**

Table 7. Comparison between the proposed model and previous studies over NSL_KDD

Sq.	Ref	Metrics	
		Accuracy	F1-score
1	[28]	80.75	80.99
2	[29]	**82.08**	**81.75**
3	proposed Model	79.21	78.91

two million. It includes nine major types of attacks. Part of the training process involves sorting graphs based on this dataset with 20% training. This process helps eliminate the data imbalance and increase the homogeneity of data distribution. The paper uses a model that uses the random state method to provide 20% of the training data and 20% of the test data from partitioned sections of this dataset for training and testing the model. Table 7,8 compares the proposed model with [28, 29].

Table 8. Comparison between the proposed model and previous studies over UNSW_NB15

Sq.	Ref	Metrics	
		Accuracy	F1-score
1	[28]	94.92	95.08
2	[29]	90.21	91.54
3	proposed Model	**97.83**	**96.5**

Compared to the model presented in [28], our proposed model achieved absolute superiority in accuracy and F1 score. Compared with the model shown in [29], our proposed model achieved a higher score in one of the two datasets, while the other slightly favoured the first model. However, our proposed model achieved high superiority in the comparisons due to its accuracy in separating outliers in the data. This means that our model was able to accurately identify and separate data points significantly different from the rest of the data, which is crucial for improving the overall accuracy of the model.

5. Conclusion

This article introduces an actionable and highly adaptable approach that yields promising results in accuracy and efficiency. This study's suggested method employs a hybrid model comprising clusters and classification. Clusters are utilized to identify training data that share significant similarities. The clusters are further refined and optimized through the clustering algorithm to achieve the most optimal clustering. Afterwards, the proposed classifier associates the test points with these refined and optimal clusters. In the IoT environment's intrusion detection systems (IDS), an essential challenge arises from the presence of imbalanced data and outliers. These data can significantly hinder the efficiency and accuracy of the model. Therefore, the proposed dynamic evolving cauchy clustering algorithm (DECS) is one of the crucial solutions to address such issues. A set of attacks, including zero-day attacks, presents a challenge in intrusion detection due to their unpredictable timing and unknown characteristics. Detecting such attacks can be particularly difficult. The evaluation results demonstrate the proposed model's significant impact in determining its effectiveness and usability. Additionally, the

efficiency and accuracy of the model in handling noisy and unbalanced data and its significance in identifying attacks are notable compared to standard machine learning algorithms and previous studies. The actual evaluation of the model's efficacy emerges when it is applied to generalize its performance on invisible and unobservable data, including previously unknown attacks.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

The Contributions of the first author are Conceptualisation, methodology software, the second author validation, formal analysis, investigation, and resources; third author project administration, and funding acquisition.

References

[1] D. Berte, "Defining the iot", In: *Proc. of the International Conference on Business Excellence*, Vol. 12, No. 1, pp. 118-128, 2018.

[2] W. Zhou, Y. Jia, A. Peng, Y. Zhang, and P. Liu, "The effect of iot new features on security and privacy: New threats, existing solutions, and challenges yet to be solved", *IEEE Internet of things Journal*, Vol. 6, No. 2, pp. 1606-1616, 2018.

[3] Z. Yang, X. Liu, T. Li, D. Wu, J. Wang, Y. Zhao, and H. Han, "A systematic literature review of methods and datasets for anomaly-based network intrusion detection", *Comput. Secur.*, Vol. 116, p. 102675, 2022.

[4] A. Blaise, M. Bouet, V. Conan, and S. Secci, "Detection of zero-day attacks: An unsupervised port-based approach", *Computer Networks*, Vol. 180, p. 107391, 2020.

[5] K. Pedapalli, B. Gupta, and P. Mahajan, "Climate change and tourism: a paradigm for enhancing tourism resilience in SIDS", *Worldw. Hosp. Tour. Themes*, Vol. 14, No. 5, pp. 431–440, 2022.

[6] W. Yassin, N. I. Udzir, A. Abdullah, M. T. Abdullah, H. Zulzalil, and Z. Muda, "Signature-Based Anomaly intrusion detection using Integrated data mining classifiers", In: *Proc. of 2014 International Symposium on Biometrics and Security Technologies (ISBAST)*, pp. 232-237, 2014.

[7] F. Abbasi, M. Naderan, and S. E. Alavi, "Anomaly detection in Internet of Things using feature selection and classification based on Logistic Regression and Artificial Neural Network on N-BaIoT dataset", In: *Proc. of 2021 5th International Conference on Internet of Things and Applications (IoT)*, pp. 1-7, 2021.

[8] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges", *Cybersecurity*, Vol. 4, pp. 1-27, 2021.

[9] R. R. Nuiaa, S. Manickam, and A. H. Alsaeedi, "A Comprehensive Review of DNS-based Distributed Reflection Denial of Service (DRDoS) Attacks", *State-of-the-Art*, Vol. 2022, No. 6, 2022.

[10] J. Hancock, T. Khoshgoftaar, and J. Johnson, "Evaluating classifier performance with highly imbalanced Big Data", *Journal of Big Data*, Vol. 10, No. 1, p. 42, 2023.

[11] R. Ahmad, I. Alsmadi, W. Alhamdani, and L. Tawalbeh, "Zero-day attack detection: a systematic literature review", *Artificial Intelligence Review*, pp. 1-79, 2023.

[12] I. Škrjanc, S. Ozawa, T. Ban, and D. Dovžan, "Large-scale cyber attacks monitoring using Evolving Cauchy Possibilistic Clustering", *Applied Soft Computing*, Vol. 62, pp. 592-601, 2018.

[13] G. Karatas, O. Demir, and O. K. Sahingoz, "Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset", *IEEE Access*, Vol. 8, pp. 32150-32162, 2020.

[14] T. Le, H. Kim, H. Kang, and H. Kim, "Classification and explanation for intrusion detection system based on ensemble trees and SHAP method," *Sensors*, Vol. 22, No. 3, p. 1154, 2022.

[15] Z. Chiba, N. Abghour, K. Moussaid, A. E. Omri, and M. Rida, "A novel architecture combined with optimal parameters for back propagation neural networks applied to anomaly network intrusion detection", *Computers & Security*, Vol. 75, pp. 36-58, 2018.

[16] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "Netflow datasets for machine learning-based network intrusion detection systems", In: *Proc. of Big Data Technologies and Applications: 10th EAI International Conference, BDTA 2020, and 13th EAI International Conference on Wireless Internet, WiCON 2020, Virtual Event*, Vol. 10, pp. 117-135, December 11, 2020.

[17] G. Pu, L. Wang, J. Shen, and F. Dong, "A hybrid unsupervised clustering-based anomaly

detection method", *Tsinghua Science and Technology*, Vol. 26, No. 2, pp. 146-153, 2020.

[18] G. Duan, H. Lv, H. Wang, and G. Feng, "Application of a dynamic line graph neural network for intrusion detection with semisupervised learning", *IEEE Transactions on Information Forensics and Security*, Vol. 18, pp. 699-714, 2022.

[19] S. Hadi, A. Alsaeedi, R. Nuiaa, S. Manickam, and A. Alfoudi, "Dynamic Evolving Cauchy Possibilistic Clustering Based on the Self-Similarity Principle (DECS) for Enhancing Intrusion Detection System", *International Journal of Intelligent Engineering & Systems*, Vol. 15, No. 5, 2022, doi: 10.22266/ijies2022.1031.23.

[20] S. Patro and K. Sahu, "Normalization: A preprocessing stage", *arXiv preprint arXiv:1503.06462*, 2015.

[21] M. Du, R. Wang, R. Ji, X. Wang, and Y. Dong, "ROBP a robust border-peeling clustering using Cauchy kernel", *Information Sciences*, Vol. 571, pp. 375-400, 2021.

[22] A. A. Obaidi, M. Jubair, I. Aziz, M. Ahmad, S. Mostafa, H. Mahdin, A. A. Tickriti, and M. Hassan, "Cauchy Density-Based Algorithm for VANETs Clustering in 3D Road Environments", *IEEE Access*, Vol. 10, No. May, pp. 76376–76385, 2022,

[23] Y. Wang, Z. Han, J. Li, and X. He, "BS-GAT Behavior Similarity Based Graph Attention Network for Network Intrusion Detection", *arXiv preprint arXiv:2304.07226*, 2023.

[24] M. Shutaywi and N. N. Kachouie, "Silhouette analysis for performance evaluation in machine learning with applications to clustering", *Entropy*, Vol. 23, No. 6, p. 759, 2021.

[25] H. Choi, N. Qureshi, and D. Shin, "Comparative Analysis of Electricity Consumption at Home through a Silhouette-score prospective", In: *Proc. of 2019 21st International Conference on Advanced Communication Technology (ICACT): IEEE*, pp. 589-591, 2019.

[26] K. Shahapure and C. Nicholas, "Cluster quality analysis using silhouette score", In: *Proc. of 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA): IEEE*, pp. 747-748, 2020.

[27] W. Lo, S. Layeghy, M. Sarhan, M. Gallagher, and M. Portmann, "E-graphsage: A graph neural network based intrusion detection system for iot", In: *Proc. of NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, pp. 1-9, 2022.

[28] S. Alkafagi, "Build Network Intrusion Detection System based on combination of Fractal Density Peak Clustering and Artificial Neural Network", *Journal of Al-Qadisiyah for Computer Science and Mathematics*, Vol. 15, No. 1, pp. 111-126, 2023.

[29] Y. Yang, K. Zheng, C. Wu, X. Niu, and Y. Yang, "Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks", *Applied Sciences*, Vol. 9, No. 2, p. 238, 2019