Received: October 31, 2021. Revised: December 6, 2021.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Evolving Dynamic Fuzzy Clustering (EDFC) to Enhance DRDoS_DNS Attacks Detection Mechnism

Riyadh Rahef Nuiaa[1,2] **Selvakumar Manickam**[2]*
Ali Hakem Alsaeedi[3] **Dhiah Eadan Jabor Al-Shammary**[3]

[1]*Department of Computer, college of education for pure sciences, Wasit University, Iraq*
[2]*National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia, Malaysia*
[3]*College of computer science and information technology, Universitas of Al-Qadisiyah, Iraq*
* Corresponding author's Email: selva@usm.my

Abstract: Distinguishing between network traffic activity, intrusion, and normal bahavour is very difficult and very time-consuming. An analyst has to review all the large and wide data to find the order of intrusion in the network connection. Therefore, a method that can detect network intrusion and reflect the current network traffic is required. In this paper, a new EDFC model (Evolving Dynamic Fuzzy clustering) algorithm is generated to improve and enhance the detection mechanism. The proposed model contains two parts: the cluster part and evolving part. This paper's main objective is to design and implement a novel and data density-based clustering scheme that provides high system performance and persistent grouping of data with high similarity and performance on big data for efficient machine learning. Compared to previous techniques, the suggested model's performance with several standard datasets such as the UNSW-NB15 dataset, KDD99 dataset, and NSLKDD dataset indicates a higher silhouette coefficient. In the EDFC model two metrics have been used to verify the quality of clusters, and these are the silhouette coefficient and the number of clusters. The EDFC model has achieved a high silhouette coefficient, and a low number of clusters compare to other models. Our focus is to enhance the detection mechanism for the DRDoS_DNS attacks. Therefore, The EDFC model has been implemented on the standard CICDDoS2019 dataset which contains DRDoS_DNS attacks and achieved a silhouette coefficient of 0.76 and a number of cluster 13.

Keywords: Dynamic clustering, Silhouette coefficient, Density peak, Optimal cluster number, Fuzzy membership.

1. Introduction

Data mining plays a crucial role in potentially high-value applications and systems. Data optimization is at the core of machine learning systems. Many machine learning systems that work with very large data sets produce inadequate prediction results. Even if training for machine learning is done offline, the long processing time is a major drawback, as one has to wait much longer to obtain optimal data sets. In addition, the large amount of data potentially decreases the significance of data that has a small amount. A large amount of data potentially increases the prediction of the machine learning model from it. Therefore, data are grouped according to their similarity (clustering), limiting the effectiveness of categories with a large amount of data to categories with a large amount of data.

Data mining in traffic management offers significant advantages in terms of traffic forecast and making appropriate decisions. Moreover, this becomes more practical as manual analysis of such vast amounts of data is unfeasible due to the frequent influx [1, 2].

For discovering knowledge in unstructured multivariate and multidimensional data, clustering is a key technique in data mining [3]. When using clustering, patterns or groupings of objects that are similar across a dataset are looked for. The objects in each cluster are similar to one another, but they are not the same as the objects in other clusters [4, 5].

Large-scale data can be gathered, combined, analyzed, and used for various reasons and purposes using the Big Data analytics concept. Indeed, the use of Big Data in government is becoming commonplace. In addition, many nations now see Big Data as a future growth engine and solve current economic and social challenges [6-8]. Therefore, clusters should be used to solve these problems [9].

DBSCAN (density-based spatial clustering of noise-affected applications) and its variants are more effective at recognizing clusters of various shapes from noisy datasets when no prior knowledge about the number of clusters is available among existing clustering methodologies [10]. The most significant unsupervised learning technique is clustering, which provides a rich conceptual and computational framework for data analysis and interpretation [11]. The goal of clustering is to reduce dissimilarity between instances in the same cluster while increasing it between clusters [12].

Clustering is often used in data mining, cybersecurity, pattern recognition, picture segmentation, genetic disease diagnosis, and other applications [13]. The issues with the number of missing clusters have been solved by [10]. These attempts to develop strategies that reduce the number of missing clusters in the ANID system, making DPC realistic.

Evolving systems are mostly concerned with time-varying settings and nonstationary data processing employing computationally efficient recursive algorithms. They're especially well-suited to online, real-time applications, as well as scenarios or operational settings that often change [14, 15]. The nature of the data streams heavily influence the evolving techniques. For example, in the case of cyber-attacks, samples are drawn at random from several classes. In order to deal with such issues, all necessary processes for adding, merging, and deleting clusters should be put in place [16]. With its dynamic nature and simplicity of calculation, the technique is also well-suited to dealing with large-scale data challenges [17]. The eCauchy algorithm described here requires only a few starting parameters, such as the density minima and maxima. It adds, merges, splits, and removes clusters as samples flow through the data stream, changing the model's structure progressively as they are added or merged or separated [18]. The silhouette value considers both cluster cohesiveness and cluster separation while calculating its value. A positive value for objects in the [-1, 1] range means that the object is more like its own cluster than the next closest cluster [1, 19].

Researchers pay more attention to reducing human errors when analysing big data such as network traffic. therefore, the clustering techniques are used in several important fields because they can distinguish between traffic patterns.

The contribution of this paper is to design and implementation of a novel and data density-based clustering scheme that provides high system performance, the persistent grouping of data with high similarity, and performance on big data for efficient machine learning. The main goals of this model are listed as below:

- Obtaining a powerful clustering model for significantly big datasets for machine learning.
- Significantly reducing the long time spent on the training phase.
- Producing persistent group of data, with aspecific parameter only, needs to be changed for correlation parameter.
- The EDFC model is offered as a potential alternative data clustering model.
- Enhancing the mechanism for detecting cyber-attacks by applying the similarity, congruence and homogeneity between data in one cluster, and will reduce the number of clusters. therefore contributing to improving the efficiency of attacks detection.

1.1 Motivation

Identifying network traffic behaviour, incursion, and normal behaviour can be a time-consuming and challenging process. To determine the order of infiltration in the network connection, an analyst must analyze all huge and wide data. Since network intrusion detection and network traffic reflection are necessary, a method is needed. When the data size is very big and constantly changing, this will lead to difficulty in the process of analysis and classification. Therefore, unsupervised learning could be one of the best solutions. clustering was used in a wide field on data analysis.

1.2 Evaluation strategy

In the EDFC model CICDDoS2019, a dataset has been utilized as a benchmark dataset. Therefore, the proposed model EDFC has been implemented on several benchmark datasets such as DRDoS_DNS, part of the benchmark CICDDoS2019 dataset, UNSW-NB15 dataset, KDD99 dataset, and NSLKDD dataset. The result shows that the EDFC high silhouette coefficient and a low number of clusters. Therefore based on the two facts above and the result, our EDFC model is better than other models.

1.3 Paper organization

The remaining sections of this paper are as follows: Section 2 explains related works; the proposed model and its methodology are presented in Section 3; Section 4 contain the experiments and results of the EDFC model, the conclusion and future work of the EDFC model are shown in Section 5.

2. Related work

In this section, the authors will be review related and similar research papers in terms of dataset and technologies used. The comparisons will focus on two metrics: the silhouette coefficient and the number of clusters because it is the core-based metric of the proposed model. Therefore, other metrics are ignored.

Ref.[20], This research offers an IDS based on integrated classification and assesses its performance on both traditional offline datasets and online real-time datasets. When compared to other current decision tree-based models for detecting these five categories, the performance of the proposed integrated classification-based model is significantly better. Cluster quality is determined using the silhouette coefficient. The cluster configuration with the highest silhouette measurement value for the lowest value of k is chosen for further investigation. When there are 15 clusters, the silhouette value is =0.60, the best among the rest. There are ten categories in the UNSW-NB15 dataset (9 attack types and 1 normal).

Ref.[21], This study presents a new misuse-based intrusion detection system to protect the network from five different types of attacks: Exploit, DOS, Probe, Generic, and Normal. The KDD99 or NSL-KDD 99 dataset is used in the majority of related publications on IDS. These datasets are no longer useful for detecting recent forms of attacks and are of no use. The UNSW-NB15 dataset is used as an offline dataset in this paper to construct an intrusion detection model for identifying malicious network activity. The UNSW-NB15 (benchmark dataset) performance evaluation of the suggested approach demonstrates a greater silhouette coefficient. When there are 15 clusters, the silhouette value is =0.60, which is the best of the bunch compared to other clusters. According to the performance analysis results, clustering is a very useful technique for analyzing similarities in behavior between different categories and thus for improving IDS performance. The proposed work provides a framework for developing effective IDS to protect the system from internal and external hostile threats.

Ref.[22], the Researchers developed an Edge-based Hybrid Intrusion Detection Framework (EHIDF) to address current intrusion detection issues for a mobile edge computing environment. Different detection modules with different classifiers make up the proposed detection framework. It is capable of detecting unknown or novel mobile edge infrastructure assaults. For future testing, the highest silhouette coefficient value for the least value of k is chosen. When the number of clusters is 11, the silhouette value is at its highest (0.7). Three intrusion detection modules with three different classifiers make up the proposed framework. The C4.5 classifier is used by the Signature Detection Module (SDM), the Naivebased classifier is used by the Anomaly Detection Module (ADM), and the Meta-AdaboostM1 method is used by the Hybrid Detection Module (HDM). By identifying new unknown assaults, the created EHIDF can solve current detection problems.

Ref.[23], to counteract the negative economic effects of the COVID-19, an information-driven dynamic clustering system was cearted in this research. Data Analytics, Dynamic Clustering, and Data Security are the three main components of the proposed methodology. A clustering technique has been presented, and it has been extensively simulated in four different scenarios in order to determine its benefits and drawbacks.

Ref. [24] When it comes to IoT network planning and architecture, genetic algorithms with a dynamic clustering method are expected to be a highly effective energy-saving solution. Dynamic clustering finds the cluster head's (CH's) energy level during data transmission and utilizes it. It is shown in this study that dynamic clustering computing-based IoT implementation may efficiently serve real-world applications such as smart transportation, smart grid, and smart cities. Frame relay nodes (RN) and the dynamic clustering-based methodology are enhanced in the proposed way to select the most preferable sensor node (SN) among the cluster's nodes. A Genetic Analysis technique is employed in this situation. The simulations show that the suggested method outperforms the DCRN clustering algorithm regarding slot usage, throughput, and data transmission standard deviation.

3. The proposed system

The proposed system called the EDFC model aims to enhance the detection mechanisms by reducing the number of clusters that will be verified.

Therefore, when the similarity between the data in one cluster is at the highest level, it will lead to achieving the highest value of silhouette coefficient. also, the number of clusters is another important metric because It will reduce the number of nodes that will be checked. The process of reducing the number of clusters will continue until the optimal number of clusters is reached. This novel model is tested and implemented on many standard datasets. The model has three main points: the creating cluster phase, reducing the number of clusters phase, then the threshold comparison. Each phase has various subphases, which are explained in detail in the subsection below. The EDFC model can be used in wide fields to minimize the number of nodes, points, elements, etc. In the experiments, the value of the threshold(theta) that is equal to 0.4 was taken. Figure 1 shows how the EDFC model works and shows each of its stages.

3.1 Creating cluster

The first stage is how to create the first cluster from the dataset points by the algorithm I, selecting a point from the dataset, i.e. X, selecting the nearest other points to X, i.e. Y. Therefore, the point will not be compared with itself. The next step is to measure the distance between those points by Eq. (1).

$$d(x_i, y_j) = \sqrt{\sum_{j=1}^{n} (x_i - y_j)^2} \qquad (1)$$

Where X, Y are points in the dataset.

The next step in the creating stage is to measure the distance between those points to find the minimum distance between them by the Eq. (2)

$$c_{xy} = argmin(d_{x_i, y_j}) \qquad (2)$$

Where x_i point in a cluster, *r* is other clusters and $j \neq i \quad y, x \in$ point data pool

If these points are found with a minimum distance, then they are assigned to their cluster at the first comparison. But, if the model has at least one cluster, then some conditions will be checked before deciding to which cluster these points belong.

3.2 Editing cluster

Homogeneity between single cluster points is very important, as the Eq. (3) can be measured.

$$c_{id} = \frac{\sum_{j=1}^{n} d(x_i, x_j)}{n-1} \qquad (3)$$

Where $j \neq i$

This equation shows the points that belong to the same class.

In some of the following steps of the proposed model, the distance between a point in a cluster to its nearest point in another cluster should be checked, and it can be calculated using Eq. (4).

$$ind_j = \frac{\sum_{i=1}^{k} c_{ir} - c_{id}}{k} \qquad (4)$$

Where *K* number of points in cluster *j*, c_{ir} is the nearest point in the other clusters to this point.

Calculating the fuzzy coefficient of each cluster and figures out how closely the elements are related to each other in the same cluster by Eq. (5). otherwise, the operation will be repeated from Eq. (1) for points that do not belong to any cluster.

$$FC_j = \begin{cases} 1 \; if \; ind_j > \theta \\ -1 \; otherwise \end{cases} \qquad (5)$$

The points that belong to a negative value in the Eq. (5) don't belong to any cluster. A new cluster will be created by Eq. (6) containing the point. Therefore, the center (μ) of cluster j is calculating by Eq. (7).

$$F_i = \frac{d(x_i, y_j) + d(x_i, c_j)}{2} \qquad (6)$$

Where y_j is the nearest point in cluster j for x_i, $c_j$the center for cluster j.

3.2.1. adding a point to a cluster:

depending on the algorithm I and the Eq. (2), if point Y has a cluster, point X will be added to the cluster to which Y belongs.

But if point Y does not have a cluster, then the operation will proceed to the create cluster operation to create a new cluster that contains both points X and Y.

3.2.2. Remove cluster

The evolving technique has two parts in the proposed model:
the first part is to reduce the number of clusters until they reach the optimal number of clusters. The second part deals with clusters with a negative value and how to treat and get rid of them.

The evolving technique reduces the number of clusters with negative values until they are all eliminated.

Eq. (6) eliminates the cluster with a negative value in the next epoch until belonging to the cluster with a positive value.

The second algorithm describes how the evolving technique works in the proposed model to reduce the number of clusters until the optimal number is reached and clusters with a negative value are eliminated.

3.2.3. Cluster centre

The cluster centre will be changed based on add a point to a cluster:

When a new cluster is created, then the cluster centre will be calculated by the Eq. (7)

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad (7)$$

Where: n is the number of points in cluster j.

Otherwise, when a point is involved with an existing cluster, the cluster center will be updated, and the new center (μ) of cluster j containing X, Y point from Eq. (1) will be calculated by the Eq. (8).

$$\mu_{j+1} = \mu_j + \frac{1}{n_{j+1}}(x - \mu_j) \qquad (8)$$

Where: x is a new point that joins in clustering j.

4. Experiments

To verify the efficiency of the proposed model (EDFC), it was compared with other models with the same dataset. The results indicate that the proposed model is better in terms of silhouette coefficient and the number of clusters. All comparison results are shown in Table 1.

4.1 Evaluation metrics

Some measurement criteria are used to evaluate the suggested EDFC model. The silhouette coefficient and clusters number are two evaluation indicators used to assess the model's performance. The EDFC model has been compared with three similar models that used the same dataset (UNSW-NB15). The results show that the EDFC model achieved a higher silhouette coefficient and low clusters number, unlike the other models.

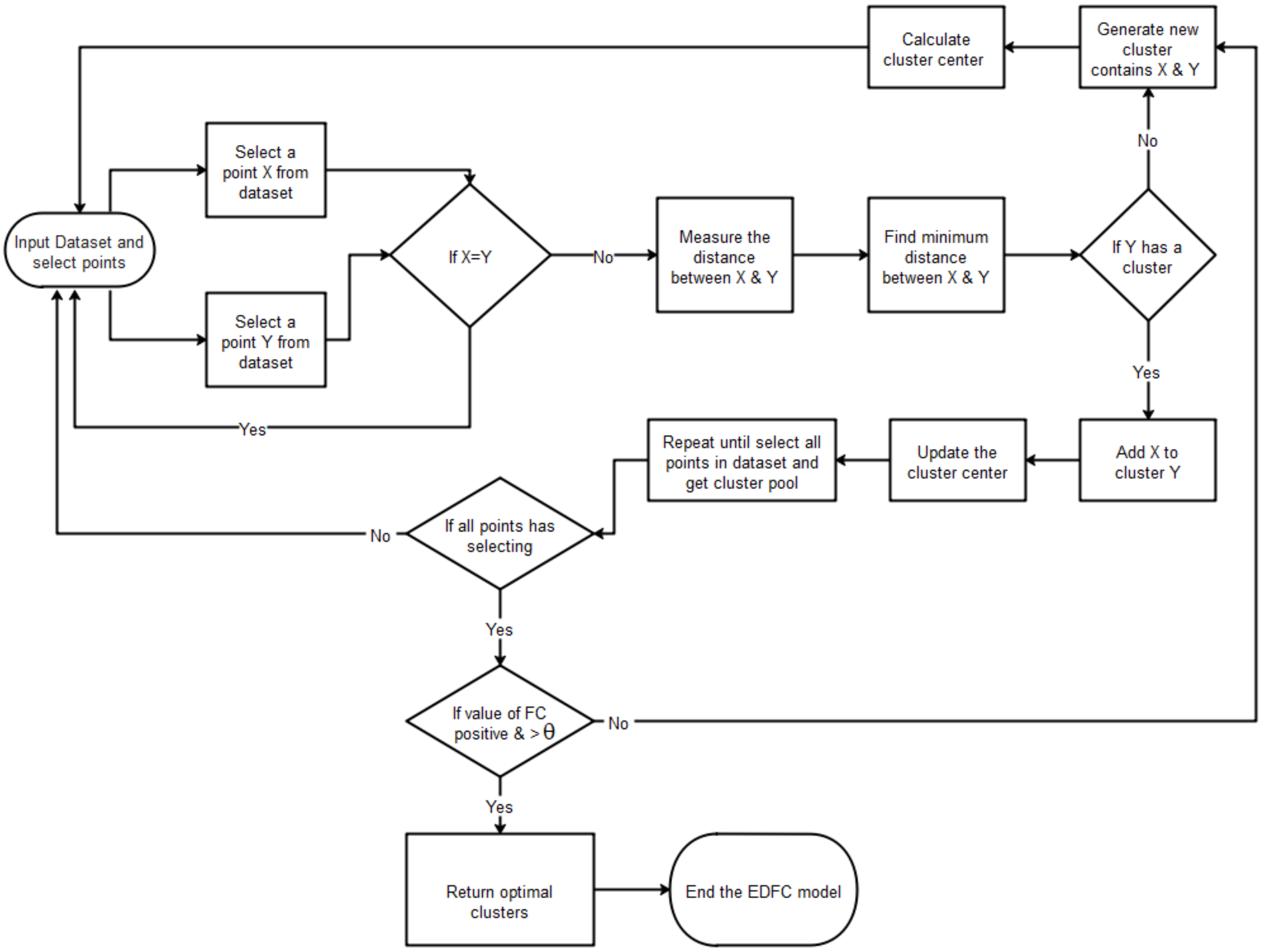
Calculate cluster center
Generate new cluster contains X & Y
No
Select a point X from dataset
Input Dataset and select points
If X=Y
No
Measure the distance between X & Y
Find minimum distance between X & Y
If Y has a cluster
Select a point Y from dataset
Yes
Yes
Repeat until select all points in dataset and get cluster pool
Update the cluster center
Add X to cluster Y
No
If all points has selecting
Yes
If value of FC positive & > θ
No
Yes
Return optimal clusters
End the EDFC model

Figure. 1 Flowchart for the proposed EDFC model

Table 1. The silhouette value for the EDFC model compared to silhouette values and the number of clusters of other models

No.	**Ref.**	**Model**	**year**	**Dataset**	**silhouette coefficient**	**Cluster number**
1	[20]	Integrated model	2020	UNSW-NB15	0.60	15
2	[21]	Statistical Analysis	2020	UNSW-NB15	0.60	15
3	[22]	EHIDF model	2021	UNSW-NB15	0.70	11
4	our model	EDFC		UNSW-NB15	0.83	2

```
Generate Clusters Algorithm I
input:   dataset
output: clusters
procedure EDFC I
clustering pool ← [ ]
        for x ← in dataset do
            for y ← in dataset and x≠y do
                $d_i$ ← distance between x and y  eq (1)
                $C_{x,y}$ ← the point that has minimum in $d_i$  eq(2)
               if y has cluster then:
                   assign x to cluster’s y
                  update the cluster center μ  eq (8)
               else:
                  generate new cluster include x and y
                  calculate the cluster center μ  eq (7)
return clustering pool
```

```
Evolving Clusters Algorithm II
input: clustering pool
output: optimal clusters
procedure EDFC II
E ←  eq (5)                    // evaluate cluster
   for j ← in clustering pool do
       if $E_j$ < 1:
         for i ← in j do
             $F_i$ ← fuzzy membership  eq (6)
             assign x to cluster that has minimum F
return optimal clusters
```

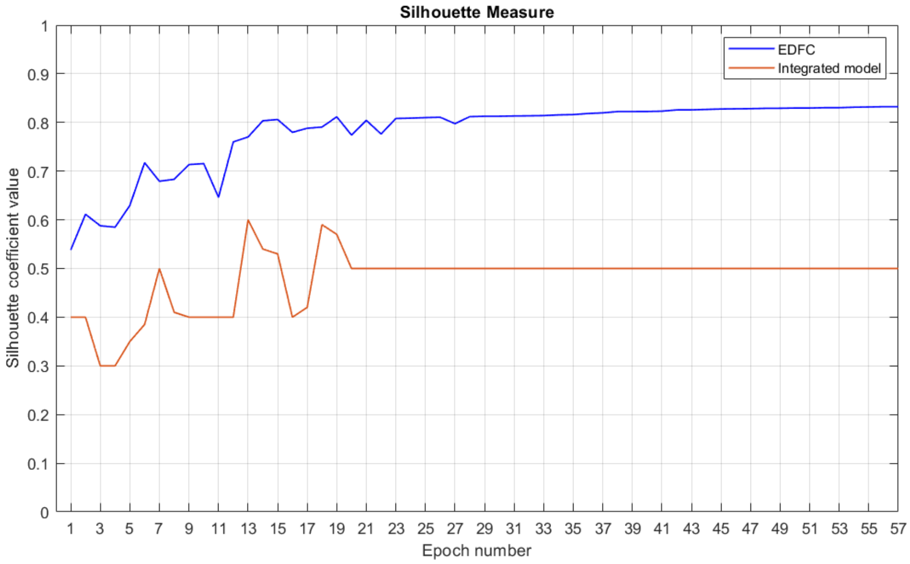

Figure. 2 EDFC model vs Integrated model

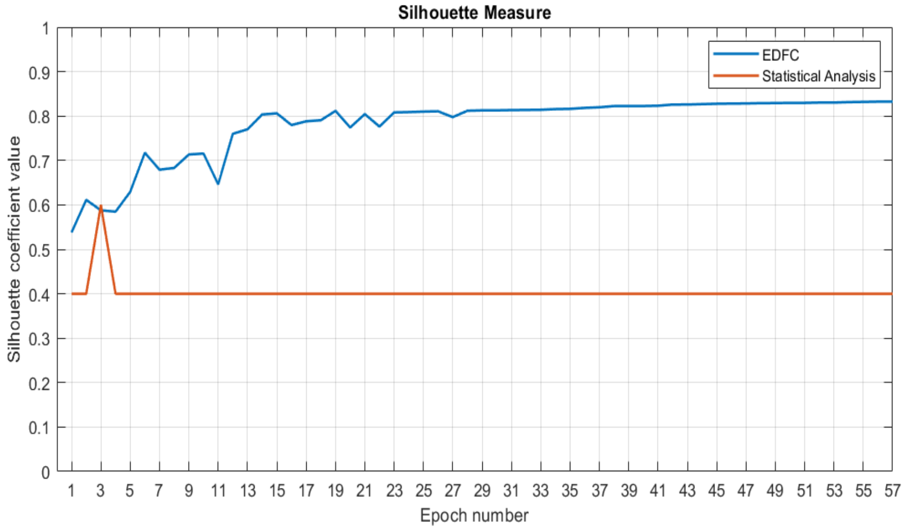

Figure. 3 EDFC model vs Statistical Analysis model

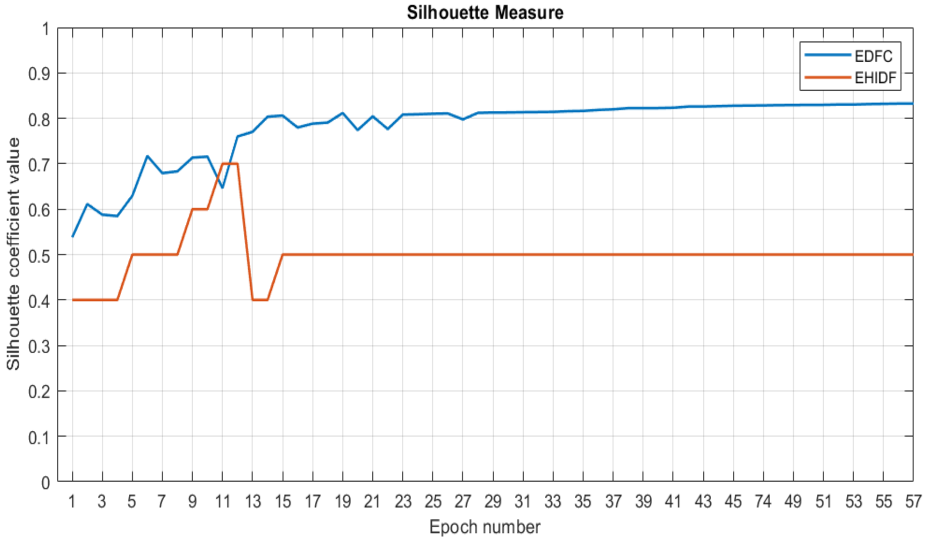

Figure. 4 EDFC model vs EHIDF model

Fig. 2 shows the comparison between the proposed model EDFC and the Integrated model from the ref. [20], the silhouette coefficient for the Integrated model is 0.6, while the cluster number is 15.

The EDFC model achieves a silhouette coefficient value that is 0.83 when the number of clusters is 2. Therefore, from the above figure, it can be said that the EFDC model is better than the other model in terms of silhouette coefficient and clusters number.

Fig. 3 shows the comparison between the proposed model EDFC and the Statistical Analysis model from the ref. [21], the silhouette coefficient for the Statistical Analysis model is 0.6, while the cluster number is 15.

The EDFC model achieves a silhouette coefficient value that is 0.83, when the number of clusters is 2. Therefore, from the above figure, it can be said that the EFDC model is better than the other model in terms of silhouette coefficient and clusters number.

Fig. 4 shows the comparison between the proposed model EDFC and the EHIDF model from the ref. [22], the silhouette coefficient for the EHIDF model is 0.7, while the cluster number is 11.

The EDFC model achieves a silhouette coefficient value that is 0.83, when the number of clusters is 2. Therefore, from the above figure, it can be said that the EFDC model is better than the other model in terms of silhouette coefficient and clusters number. To measure the performance of the EDFC model, it was tested on several standard datasets as shown in Table 2.

Received: October 31, 2021. Revised: December 6, 2021.

Table 2. The silhouette values and clusters number for the EDFC model with different datasets

Dataset EDFC	DRDoS_DNS	KDD99	NSLKDD	UNSW-NB15
Silhouette value	0.76	0.66	0.93	0.83
Cluster number	13	3	2	2

Fig. 5 shows the observed results which indicate that the average silhouette coefficient is the largest = 0.76 when k=13, representing the best clustering quality, i.e. the optimal clustering number is 13 when implementing EDFC on standard DRDoS_DNS dataset.

Fig. 6 shows the observed results which indicate that the average silhouette coefficient is the largest = 0.93 when k=2, representing the best clustering quality, i.e. the optimal clustering number is 2 when implementing EDFC on standard NSL-KDD dataset.

Fig. 7 shows the observed results which indicate that the average silhouette coefficient is the largest = 0.83 when k=2, representing the best clustering quality, i.e. the optimal clustering number is 2 when implementing EDFC on standard UNSW-NB 15 dataset.

Fig. 8 shows the observed results which indicate that the average silhouette coefficient is the largest = 0.66 when k=3, representing the best clustering quality, i.e. the optimal clustering number is 3 when implementing EDFC on standard KDD99 dataset.

Fig. 9 shows that when implementing the EDFC model on the DRDoS_DNS, part of the benchmark CICDDoS2019 dataset, the number of clusters is reduced at each epoch until achieving the optimal number of clusters in the last epoch with the best silhouette coefficient.

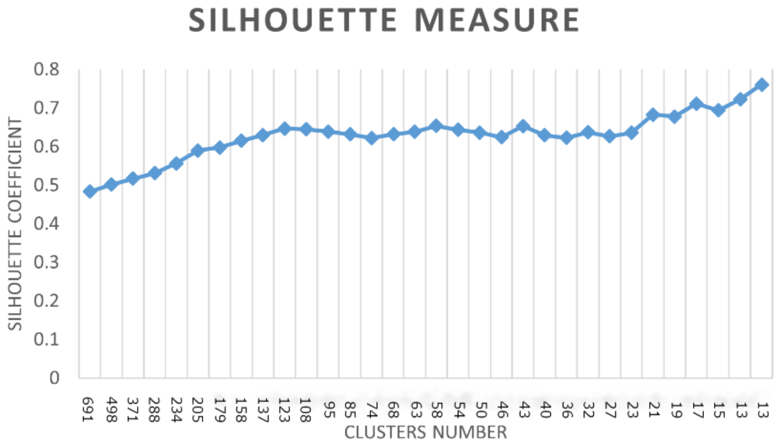

Figure. 5 Silhouette coefficient graph for selecting the optimal number of clusters

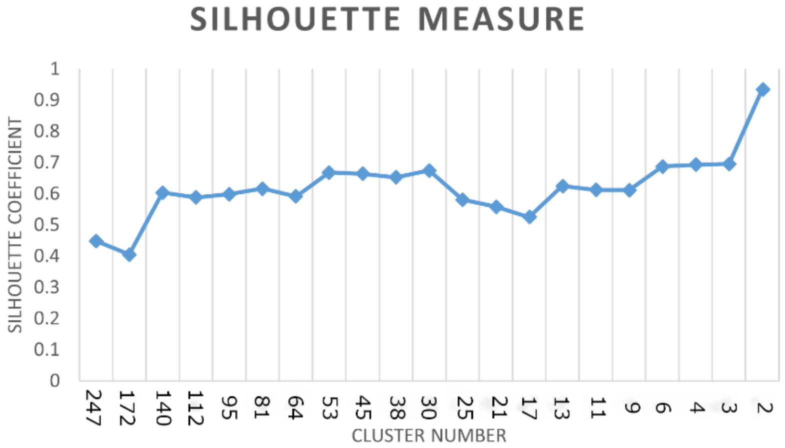

Figure. 6 Silhouette coefficient graph for selecting the optimal number of clusters

Received: October 31, 2021. Revised: December 6, 2021.

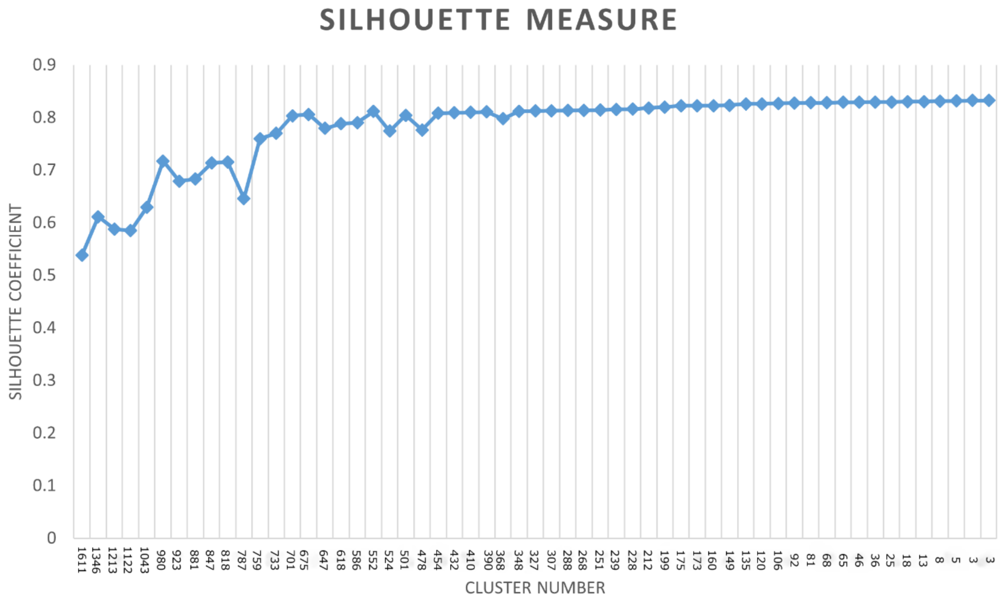

Figure. 7 Silhouette coefficient graph for selecting the optimal number of clusters

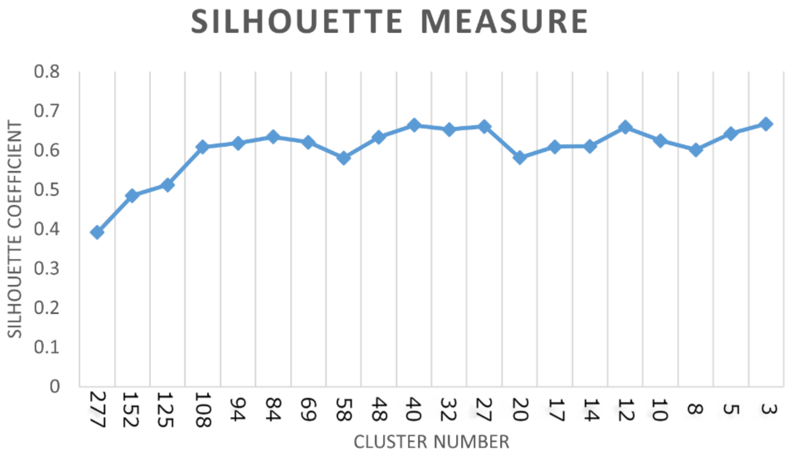

Figure. 8 Silhouette coefficient graph for selecting the optimal number of clusters

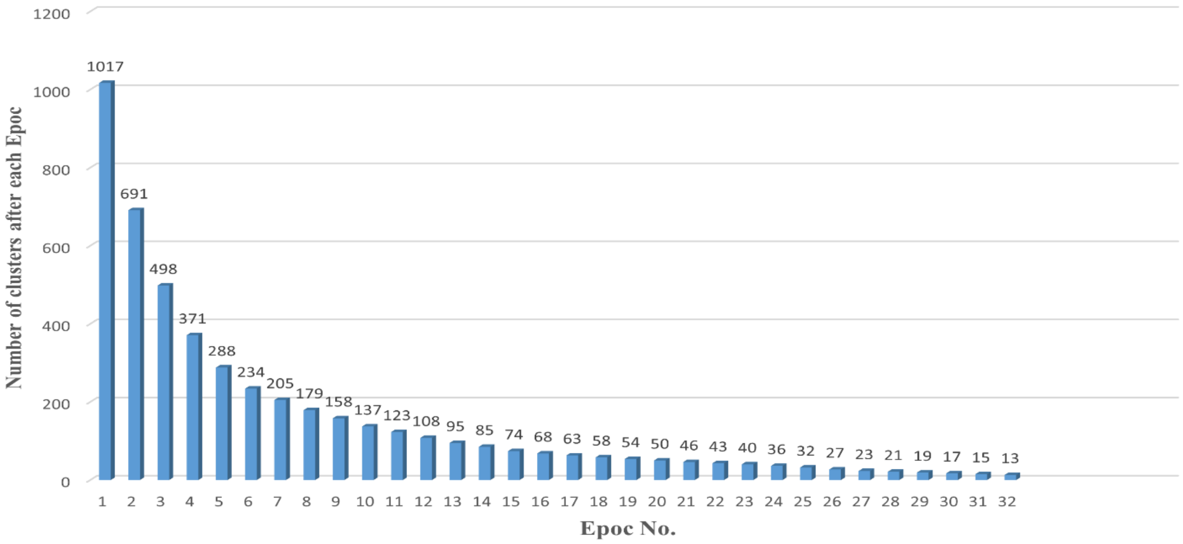

Figure. 9 The number of clusters during each epoch for the DRDoS_DNS dataset

5. Conclusion and future work

The main challenge in analyzing big data, especially the huge amount for network traffic, is distinguishing between normal and abnormal traffic. The clustering techniques are employed to solve these types of problems. The EDFC model is a new cluster algorithm tested on big datasets such as the CICDDoS2019 dataset, UNSW-NB15 dataset, KDD99 dataset, and NSLKDD dataset. The experiments results indicate that the EDFC model is better than other models based on two famous metrics: silhouette coefficient and clusters number.

The future work is to implement the EDFC model on a real online dataset and use other metrics to improve the model's performance.Callouts, figure and table captions should be 10-point non-boldface Times New Roman. Initially capitalize only the first word of each figure caption.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The paper background work, conceptualization, methodology and result analysis and comparison have been done by first Author. Dataset collection, implementation, preparing and editing draft, visualization have been done by third and fourth authors. The supervision, review of work and project administration, have been done by second author.

References

[1] D. Toshniwal, N. Chaturvedi, M. Parida, A. Garg, C. Choudhary, and Y. Choudhary, “Application of clustering algorithms for spatio-temporal analysis of urban traffic data”, *Transp. Res. Procedia*, Vol. 48, pp. 1046-1059, 2020.

[2] L. Yang, A. Moubayed, and A. Shami, “MTH-IDS: A Multi-Tiered Hybrid Intrusion Detection System for Internet of Vehicles”, *IEEE Internet Things J.*, 2021.

[3] S. Pasupathi, V. Shanmuganathan, K. Madasamy, H. R. Yesudhas, and M. Kim, “Trend analysis using agglomerative hierarchical clustering approach for time series big data”, *J. Supercomput.*, pp. 1-20, 2021.

[4] D. T. Dinh, T. Fujinami, and V. N. Huynh, “Estimating the optimal number of clusters in categorical data clustering by silhouette coefficient”, In: *Proc. of International Symposium on Knowledge and Systems Sciences*, pp. 1-17, 2019.

[5] D. T. Dinh, V. N. Huynh, and S. Sriboonchitta, “Clustering mixed numerical and categorical data with missing values”, *Inf. Sci. (Ny).*, Vol. 571, pp. 418-442, 2021.

[6] E. S. Kim, Y. Choi, and J. Byun, “Big Data Analytics in Government: Improving Decision Making for R&D Investment in Korean SMEs”, *Sustainability*, Vol. 12, No. 1, p. 202, 2020.

[7] S. Heidari, M. Alborzi, R. Radfar, M. A. Afsharkazemi, and A. R. Ghatari, “Big data clustering with varied density based on MapReduce”, *J. Big Data*, Vol. 6, No. 1, pp. 1-16, 2019.

[8] H. I. Hayatu, A. Mohammed, and A. B. Isma’eel, “Big Data Clustering Techniques: Recent Advances and Survey”, *Mach. Learn. Data Min. Emerg. Trend Cyber Dyn.*, pp. 57-79, 2021.

[9] R. A. A. Habeeb, F. Nasaruddin, A. Gani, M. A. Amanullah, I. A. T. Hashem, E. Ahmed, and M. Imran, “Clustering-based real-time anomaly detection—A breakthrough in big data technologies”, *Trans. Emerg. Telecommun. Technol.*, p. e3647, 2019.

[10] Z. Ghaemi and M. Farnaghi, “A varied density-based clustering approach for event detection from heterogeneous twitter data”, *ISPRS Int. J. Geo-Information*, Vol. 8, No. 2, p. 82, 2019.

[11] X. Xu, S. Ding, L. Wang, and Y. Wang, “A robust density peaks clustering algorithm with density-sensitive similarity”, *Knowledge-Based Syst.*, Vol. 200, p. 106028, 2020.

[12] D. Ienco and G. Bordogna, “Fuzzy extensions of the DBScan clustering algorithm”, *Soft Comput.*, Vol. 22, No. 5, pp. 1719-1730, 2018.

[13] Y. Shi, C. Otto, and A. K. Jain, “Face clustering: representation and pairwise constraints”, *IEEE Trans. Inf. Forensics Secur.*, Vol. 13, No. 7, pp. 1626-1640, 2018.

[14] D. Leite, I. Škrjanc, and F. Gomide, “An overview on eVolving systems and learning from stream data”, *EVol. Syst.*, pp. 1-18, 2020.

[15] J. Maia, C. A. S. Junior, F. G. Guimarães, C. L. D. Castro, A. P. Lemos, J. C. F. Galindo, and M. W. Cohen, “EVolving clustering algorithm based on mixture of typicalities for stream data mining”, *Futur. Gener. Comput. Syst.*, Vol. 106, pp. 672-684, 2020.

[16] I. Škrjanc, S. Ozawa, T. Ban, and D. Dovžan, “Large-scale cyber attacks monitoring using eVolving cauchy possibilistic clustering”, *Appl. Soft Comput.*, Vol. 62, pp. 592-601, 2018.

[17] Z. Zhao, C. Li, X. Zhang, F. Chiclana, and E. H. Viedma, “An incremental method to detect communities in dynamic eVolving social networks”, *Knowledge-Based Syst.*, Vol. 163, pp.

404-415, 2019.

[18] I. Škrjanc, S. Blažič, E. Lughofer, and D. Dovžan, "Inner matrix norms in eVolving cauchy possibilistic clustering for classification and regression from data streams", *Inf. Sci. (Ny).*, Vol. 478, pp. 540-563, 2019.

[19] K. Lingelbach, S. Gado, D. Janssen, D. Piechnik, M. Eichler, D. Knopf, L. Hentschel, M. Schuler, D. Sernatinger, and M. Peissner, "Identifying the effects of COVID-19 on psychological well-being through unsupervised clustering for mixed data", In: *Proc. of Sixth International Congress on Information and Communication Technology*, pp. 883-895, 2022.

[20] V. Kumar, D. Sinha, A. K. Das, S. C. Pandey, and R. T. Goswami, "An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset", *Cluster Comput.,* Vol. 23, No. 2, pp. 1397-1418, 2020.

[21] V. Kumar, A. K. Das, and D. Sinha, "Statistical analysis of the UNSW-NB15 dataset for intrusion detection", *Computational Intelligence in Pattern Recognition*, pp. 279-294, 2020.

[22] A. Singh, K. Chatterjee, and S. C. Satapathy, "An edge based hybrid intrusion detection framework for mobile edge computing", *Complex Intell. Syst.*, pp. 1-28, 2021.

[23] M. A. Rahman, N. Zaman, A. T. Asyhari, F. A. Turjman, M. Z. A. Bhuiyan, and M. F. Zolkipli, "Data-driven dynamic clustering framework for mitigating the adverse economic impact of Covid-19 lockdown practices", *Sustain. Cities Soc.*, Vol. 62, p. 102372, 2020.

[24] S. Rani, S. H. Ahmed, and R. Rastogi, "Dynamic clustering approach based on wireless sensor networks genetic algorithm for IoT applications", *Wirel. Networks*, Vol. 26, No. 4, pp. 2307-2316, 2020.