Received: April 21, 2023. Revised: June 2, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Adaptive DBSCAN with Grey Wolf Optimizer for Botnet Detection

Duaa Haider Mustafa[1]* **Idress Mohammed Husien[1]**

[1]*Department of Computer Science, College of Computer Science and Information Technology, University of Kirkuk, Kirkuk, Iraq*
* Corresponding author's Email: stch21m003@uokirkuk.edu.iq

Abstract: As the number of devices linked to the Internet (IOT devices) has dramatically increased, botnet attacks are becoming one of the most serious threats on the Internet. Many studies have been proposed for botnet detection based on machine learning. However, most of these existing studies focus on offline botnet detection using supervised machine learning methods. Since botnet attacks are committed in real time, they require online detection. Also, classification may not be practical for IOT applications such as botnet detection for many reasons that will be discussed in this paper. In order to overcome this limitation in the existing models, we propose an online botnet detection technique using unsupervised hybrid DBSCAN-GWO architecture. In this model, DBSCAN's eps parameter is generated automatically for each data stream using grey wolf optimizer which searches for the optimum eps value to give the best clustering quality for each data stream adaptively. After finding clusters in each data stream, a comparison is made between the clusters depending on the difference between their values to find the botnet clusters for each data stream adaptively. This model is evaluated using N_BaIot datasets of six different IOT devices. The results show the efficiency of DBSCAN-GWO model in detecting botnet data in all datasets compared to the regular DBSCAN with 3 different eps values and OPTICS clustering algorithms, as the best accuracy reaches 98%, which is also compared to a number of existing techniques which are the semi-supervised K-means clustering algorithm of 79.60%, DBSCAN clustering algorithm of 80%, and clustering-based semi-supervised machine learning approach of 96.66% for detecting anomalies and DDOS attacks.

Keywords: Online botnet detection, Unsupervised machine learning, DBSCAN, Grey wolf optimizer, Automatic parameter tuning, Eps parameter, Density based clustering.

1. Introduction

Over the past few years, there has been a swift rise in the quantity of devices that are linked to the Internet. These smart devices, such as smartphones, smart home devices, security cameras, webcams, and sensors, work and are connected to the internet 24/7 under insufficient protection. That makes these devices vulnerable to a hazardous attack which is called a botnet [1]. Several methods have been suggested for identifying botnets, including signature-based, anomaly-based, and machine learning-based detection. In signature-based detection, the detection system has a database that stores the most known botnets' signatures. So, the incoming traffic is considered a botnet when its signature matches the stored (known) botnet. This way of detection is useless for unknown botnets. Anomaly-based detection involves detecting the presence of malicious bots in a network by analyzing various irregularities in network traffic. Anomaly-based detection aims to identify botnets by detecting these abnormalities [2]. Anomaly-based intrusion detection, whether at the network or host level, has several drawbacks. These include a high rate of false positives and vulnerability to attacks delivered in a way that evades detection. To address these issues and improve anomaly detection accuracy, machine learning has been suggested as a way to automate the process. In machine learning botnet detection, the detection can be achieved based on supervised and unsupervised machine learning. Most ML-based proposed studies use classification (supervised ML) [3]. Supervised ML is not practical for botnet

detection since botnets mostly infect IOT devices. Data generated by IoT devices can be highly variable and subject to noise and errors. This can make it challenging to train an accurate classification model. Another problem is that classification requires predefined labels, which are not always available. Also, IoT devices often generate unstructured data, such as sensor readings or image and video feeds, which can be difficult to analyze and classify using supervised ML algorithms. Finally, the most challenging point is that data generated by devices connected to the internet requires real-time processing, which is very challenging for classification algorithms [4]. On the other hand, clustering is very useful for analyzing IOT data according to its ability to discover patterns in data, detect anomalies, and scale, with no predefined labels required. Finally, clustering algorithms can be designed to operate in real time. As a conclusion, unsupervised ML is more appropriate for botnet detection [5]. Clustering algorithms are divided into many categories, such as density-based, distribution-based, centroid-based, and hierarchical-based clustering. Density-based clustering is currently the most commonly used approach in detection, as it involves identifying clusters of data objects that are densely concentrated in a specific region. These clusters are differentiated from one another based on areas where there is a comparatively lower density of objects. Objects that are situated in these less dense regions are usually regarded as outliers or noise [6]. This paper's chosen algorithm for online botnet detection is DBSCAN, a density-based clustering algorithm. This algorithm was selected for various reasons, which will be discussed in the following section. The problem with DBSCAN is that it requires setting parameters to operate. For an automatic and adaptive generation of (eps) parameter, DBSCAN-GWO is proposed. A technique for identifying botnets on the internet that relies on clustering (an unsupervised machine learning approach). The data enters the proposed model as data streams, and each data stream is processed individually. The best result the DBSCAN algorithm can give depends on the value of its parameter Eps. Values of this parameter should be selected to give the best results. Most previous studies chose to set this parameter manually; keep changing the values until they get the desired results. This can't be applied to the real world online botnet detection system [7]. To address this problem, grey wolf optimizer is utilized to choose the optimum value of DBSCAN's (Eps) parameter that gives the best clustering quality for each data stream automatically and adaptively. Which in turn will give the best detection accuracy. The evaluation used 6 datasets of type N_BaIot related to six different devices; the botnets were of type Mirai and Gafgyt. The proposed DBSCAN-GWO model achieves online unsupervised botnet detection with the automatic parameter tuning of the DBSCAN's eps parameter adaptively for each data stream using grey wolf optimizer. The paper is structured as follows: Section 2 provides an overview of the existing methods through a literature survey, while section 3 introduces the proposed DBSCAN-GWO method. In section 4, the results and discussion pertaining to the DBSCAN-GWO method are presented. Section 5 concludes the research by outlining the findings and discussing future directions.

2. Related work

Machine learning has many use cases, such as image processing, speech recognition, catching Email spam and catching malware. This research focuses on the application of machine learning in the field of attack detection. Machine learning can be divided into two main categories, namely supervised ML and unsupervised ML [8]. This section will review the number of most popular ML algorithms in field of attack detection published for the last 3 years.

Muhammad [9] suggests a strategy to detect botnets in their early stages. Initially, the method employs feature selection techniques to choose the most suitable features. These features are then utilized to assess the performance of machine learning classifiers in detecting botnets. Then, random forest (RF), support vector machine (SVM) and logistic regression classifiers are applied for the detection of botnets. The limitation of this study is that it processes fixed data set, requires predefined labels and it uses classification which is not suitable for botnet detection for many reasons mentioned in the first section. To address the limitations of classification, unsupervised ML algorithms are suggested. Next, a number of the most recent papers that have used clustering algorithms in the field of detecting malicious attacks will be reviewed.

Cui [10] suggests a defence mechanism for detecting and preventing DDoS attacks targeting SDN controllers by analyzing traffic distribution. The method utilizes the K-means clustering algorithm to detect such attacks. The algorithm generates the current network's traffic distribution and detects attacks by analyzing the proportion of low-traffic flows. By using K-Means as an unsupervised machine learning algorithm, the detection method is adaptable and can detect various types and scales of attacks. The results demonstrate the effectiveness of

Table 1. An evaluation of DBSCAN algorithm in relation to alternative ML algorithms for botnet detection

Ref.	Algorithm/ Model	Botnet Detection Algorithms Requirements				
		Data Stream	Evolving Data	Handling Outliers	Arbitrary Shape Clusters	Less sensitive to noisy datasets
[9]	RF SVM LR (Classifiers)	×	×	×	×	×
[10]	K-Means	✓	×	×	×	×
[11]	BRICH Clustering	✓	×	✓	×	×
[12]	OPTICS	✓	✓	✓	✓	×
[13]	DBSCAN	✓	✓	✓	✓	✓

the proposed approach in detecting and preventing DDoS attacks. However, the used clustering algorithm k-means has many limitations such as it doesn't identify arbitrarily shaped clusters, doesn't handle outliers, not evolving data and is very sensitive to noisy datasets.

Another clustering algorithm was proposed by Lang [11] which improves the feature trees of BIRCH clustering. This study proposes a new cluster feature that eliminates the numeric problem and is easy to maintain, without incurring significant additional costs. This feature simplifies many computations and improves efficiency. The cluster features can be readily utilized in other approaches based on BIRCH, such as streaming data algorithms, without requiring significant modifications. Birch clustering is efficient for processing data streams, and handling outliers and big data. However, the used clustering algorithm is not specifically designed to identify arbitrarily shaped clusters and is very sensitive to noisy datasets.

Subudhi [12] builds a system for detecting unauthorized activities in databases is developed, employing OPTICS clustering and ensemble learning. The system consists of two main stages: training and testing. In the training phase, the features of the input dataset are processed, and OPTICS clustering is utilized to generate behavioral profiles. In the testing phase, transactions are assessed against these profiles to determine if they match any of them. However, the used OPTICS clustering is very sensitive to noisy datasets, which degrades the performance of the clustering model and increase the misclassification.

Deng [13] compares DBSCAN and K-means clustering algorithms in the field outlier detection. It evaluates the efficiency and performance of DBSCAN clustering. As a conclusion, for building an IDS, DBSCAN is more efficient since it can handle outliers, identify arbitrary shaped clusters, less sensitive to noisy datasets and it is possible to adapt DBSCAN for evolving data by updating the clusters as new data points become available as illustrated in Table 1. However, this study processes fixed dataset and sets the values of DBSCAN's parameters (eps,minpt) manually by testing which values can give the best results which is not practical for attack detection since these values must be changed for each data streams adaptively and automatically.

The contributions of this paper:

- Achieve online botnet detection using machine learning.
- Utilize grey Wolf optimization algorithm to tune DBSCAN's parameter (eps) automatically.
- For every data stream, the selection of the parameter value (eps) of DBSCAN is adjusted accordingly.

3. Methodology

In botnet detection, the attacks are committed in real time. Supervised ML is not a practical way to detect botnets in this case. Classification requires predefined labels which may not be available [14]. This study suggests a data stream botnet detection method based on unsupervised ML. The clustering algorithm used is hybrid DBSCAN-GWO, grey wolf optimizer is utilized to generate the optimum value of DBSCAN's epsilon. In the traditional DBSCAN, the value of eps is set manually, which is not practical to apply in real-world botnet detection systems. Each data set requires a different epsilon value for the best clustering quality possible. Next, we will discuss in detail the phases of the model.

3.1 The DBSCAN algorithm

This subsection presents the depiction and attributes of the basic DBSCAN clustering algorithm. DBSCAN is a widely employed clustering algorithm

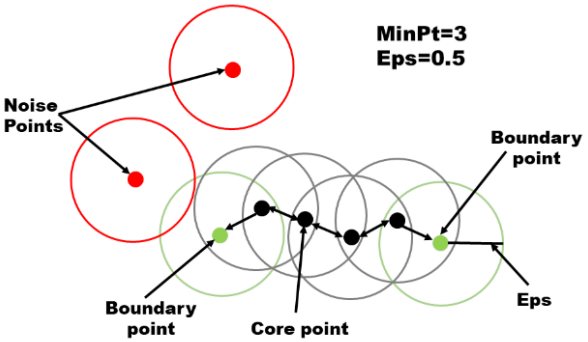


Figure. 1 DBSCAN clustering algorithm simulation

that not only identifies clusters of arbitrary shapes accurately but also effectively detects noise points [15]. We can locate a particular clustering center by carefully calculating the clustering number and investigating the central points. This method utilizes the concept of spatial density to distinguish between points belonging to a category and those that are noise by counting the number of nearby points. By changing the range of parameters, this method, which detects outliers, can classify data points and identify far-off outliers while also enabling the identification of noisy points [3]. DBSCAN's central idea is that nodes within a cluster must be near, and those within various clusters must be distant. There are two parameters to be considered for DBSCAN: MinPts and Epsilon (see Fig. 1). Below are some definitions related to the DBSCAN clustering algorithm:

i. Eps: The distance separating neighborhoods. A distance between two locations that is less than or equivalent to eps is regarded as a neighboring distance.
ii. MinPts: The smallest quantity of points necessary to form a cluster.
iii. Density reachable: If a series of core points connects density-reachable from one another. Based on the previously mentioned MinPts and Epsilon parameters, each point is classified as a core point, a boundary point, or an outlier.
iv. Core point: A point is classified as a core point if it has at least minPts neighboring points within a distance of eps, including itself.
v. Boundary point: A point is considered a boundary point if it I s not a core point but can still be reached by a core point and has fewer than minPts points in its neighborhood.
vi. Noise: A point that fails to meet the conditions for being categorized as a core or boundary point, and cannot be reached from any core points, is considered an outlier or noise.

3.2 The grey wolf optimizer

Grey wolf optimizer (GWO) is a meta-heuristic search algorithm introduced by Mirjalili et al. [16].

3.2.1. Inspiration of the algorithm

The GWO algorithm is inspired by grey wolves' social behaviour and hunting patterns in their natural habitat. Grey wolves are top predators that occupy the highest level in the food chain, and their behavior and social organization have influenced the design of the GWO algorithm. On average, gray wolves prefer to reside in packs consisting of 5 to 12 individuals. A highly rigid social dominance hierarchy exists among all group members. The algorithm begins by creating an initial population, which is randomly generated. According to the algorithm, this population is then divided into four categories, namely alpha, beta, delta, and omega. The algorithm initially generates a random population and begins the search process. The population is divided into four categories, namely alpha, beta, delta, and omega. After a certain number of iterations, the top three solutions are represented by the letters α, β, and δ, while the remaining population is represented by ω. To obtain better solutions, the wolves in the ω category must surround and approach the α, β, and δ categories [17].

3.2.2. Main phases of grey wolf hunting:

- The act of pursuing, following and getting closer to the target animal.
- Chasing, surrounding, and bothering the target animal until it comes to a halt.
- The act of assaulting the target animal.

The GWO (grey wolf optimizer) algorithm was created by utilizing a mathematical model that represents grey wolves' social ranking system and hunting habits.

3.2.3. Mathematical formulas of Encircling the prey:

$$\vec{D} = \left|\vec{C} \cdot \vec{X}_P(t) - \vec{X}_p(t)\right| \quad (1)$$

$$\vec{X}(t+1) = \vec{X}_P(t) - \vec{A} \cdot \vec{D} \quad (2)$$

Values of $\vec{A}$, $\vec{C}$ coefficient vectors are given by the equations below:

$$\vec{A} = 2\vec{a}\vec{r_1} - \vec{a} \quad (3)$$

$$\vec{C} = 2\vec{r_2} \quad (4)$$

$\vec{a}$: Linearly decreased in the range [2, 0] for successive iterations.

$\vec{r_1}\vec{r_2}$: random vectors in the range 0,1.

3.2.4. Mathematical formulas of hunting process:

In every cycle of the GWO algorithm, the omega wolves modify their positions by considering the positions of the alpha, beta, and delta wolves, who have greater expertise in detecting potential prey locations.

$$\begin{aligned} \vec{X}_1 &= \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha), \\ \vec{X}_2 &= \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta), \\ \vec{X}_3 &= \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta), \end{aligned} \tag{5}$$

$$\begin{aligned} \vec{D}_\alpha &= |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}|, \\ \vec{D}_\beta &= |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}|, \\ \vec{D}_\delta &= |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}|, \end{aligned} \tag{6}$$

$$\vec{X}(t+1) = \frac{X_1 + X_2 + X_3}{3}, \tag{7}$$

Where $\overrightarrow{X}\alpha$ is the position of the alpha, $\overrightarrow{X}$ β is the position of the beta, $\overrightarrow{X}$ δ is the position of the delta, $\overrightarrow{C}$ 1, $\overrightarrow{C}$ 2, and $\overrightarrow{C}$ 3 and $\overrightarrow{A}$ 1, $\overrightarrow{A}$ 2, and $\overrightarrow{A}$ 3 are all random vectors, $\overrightarrow{X}$ is the position of the current solution, and t indicates the number of iterations. Next step is attacking the prey, attacking the target is completed by gradually reducing $\vec{\alpha}$ from 2 to 0. As a result, the volatility of $\overrightarrow{A}$ is also decreased.

3.2.5. Attacking prey (exploitation):

To represent the wolf's behavior of attacking prey that has stopped moving in mathematical terms, the value of $\vec{\alpha}$ is gradually reduced during iterations. $\vec{\alpha}$ is a randomly generated value within the range of [-2a, 2a], where the variable "a" is reduced from 2 to 0 throughout the iterations. |A|<1 force the wolves to attack the prey (exploitation).

3.2.6. Searching for prey (exploration)

If |A|>1, then the gray wolves are compelled to diverge from the current prey in order to search for a more suitable prey. This behavior is known as exploitation, as it involves exploring new possibilities in the search space. The GWO algorithm also incorporates the vector $\vec{C}$, which contains a random value between 0 and 2, and contributes to the exploration aspect of the algorithm. If C>1, it emphasizes attacking behavior, while C<1 de-emphasizes attacking behavior, promoting a more exploratory behavior.

3.3 Data streaming

In IoT environment, traffic data enters the Internet in real-time and forms an infinite number of streams. The streaming concept can be described as a series of objects already present, continuous, and organized (explicitly or implicitly by timestamp or entry time) [1]. To simulate online detection, the data set is treated as streams. In each stream, 250 rows are processed individually; results of experiments demonstrate that this number yields the most effective clustering outcomes. As the first stream enters the model, it is clustered into groups using DBSCAN depending on the eps value chosen by GWO. The used mechanism to find benign and botnet clusters will discussed in the next sections.

3.4 Objective function

Linear programming problems involve optimizing a real-valued function known as the objective function, subject to a set of constraints that define feasible solutions. The objective function must be either maximized or minimized within these constraints. It is an algebraic equation that expresses the goal of the issue and can be scaled up or down [18]. The objective function to be optimized in this work is the clustering metric Silhouette Coefficient. Its value is between the range (-1, 1) [19]. The GWO algorithm explores a range of values, specifically between 0.1 and 4 which represent (eps), to identify the value that produces the highest Silhouette Coefficient value. To achieve this, GWO iteratively applies these values to the DBSCAN algorithm and evaluates Silhouette Coefficient value using the cluster labels assigned to that data stream. This process is repeated by testing different values within the specified range to obtain the highest possible Silhouette Coefficient value. The eps value that corresponds to this optimal quality is then utilized in the clustering process performed by DBSCAN. By employing this approach, we ensure the formation of the most effective clusters for each data stream.

Silhouette Coefficient metric is calculated according to the formula below:

$$Silhouette\ Coefficient = \frac{b-a}{max(a,b)} \tag{8}$$

a = The average distance between each point within a cluster.

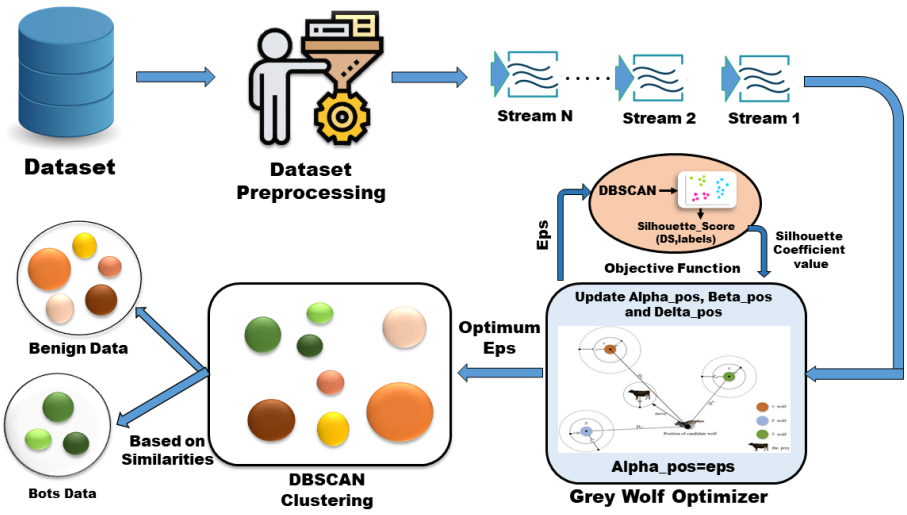


Figure. 2 DBSCAN-GWO model framework

b= The average distance between all clusters.

3.5 Hybrid DBSCAN-GWO

In this research, we propose a novel approach for detecting botnets using machine learning. Our proposed model incorporates the grey wolf optimization technique to automatically and adaptively adjust the eps parameter of the DBSCAN clustering algorithm for each data stream as shown in Fig. 2. When a stream enters the model, we need to determine the appropriate eps value. We achieve this by employing the GWO algorithm, which optimizes the objective function (Silhouette Coefficient) to maximize the clustering quality. The GWO algorithm searches for the eps value that yields the highest Silhouette Coefficient value, thus maximizing the objective function. Once we find the best eps value for a particular data stream, it is utilized in DBSCAN to form clusters where points close to each other belong to the same cluster, while points far apart are assigned to different clusters. This process aids in identifying botnet clusters amidst benign clusters at a later stage. We repeat this procedure for each data stream, adjusting the eps value based on the characteristics of that specific stream. Consequently, the tuning of DBSCAN's eps parameter becomes adaptive and automatic, leading to optimal results. Our model closely simulates real-world botnet detection techniques, making it highly applicable in practical botnet detection applications.

3.6 Detecting botnet clusters

For every data stream, the clustering process generates multiple clusters as its output. Since unsupervised machine learning lacks predefined labels, we adopt an approach to determine normal and abnormal data based on the characteristics of the clusters. The largest cluster in each data stream is treated as the normal data reference point. Clusters that closely resemble the normal cluster are considered normal, while those that significantly differ from the normal clusters are identified as bot clusters or outliers. To differentiate between normal and bot clusters, we compare the values of each cluster with those of the largest cluster. If the number of similar values surpasses a predefined threshold, which falls within the range of 50 to 250, the cluster is classified as normal and merged with the largest cluster. Conversely, if the number of similar values falls below the threshold, the cluster is identified as a bot cluster. This process is repeated iteratively until all clusters have been processed for each data stream.

4. Experiments and detection results

4.1 Dataset overview

Datasets play a central role in evaluating the efficiency of the clustering algorithms. When the values of the botnet data are close to the benign, it is more challenging to detect the botnet attack. So, dataset selection affects model evaluation. For this study, the N_BaIot datasets [20] were chosen which contain six subdatasets belongs to six different IOT devices. N_BaIot dataset were created by UCI Machine Learning Repository in 2018 and were generated by six IOT devicesEach dataset contains 3 files; 1 is the benign data, 2 is the type Mirai botnet, and the last is the type Gafgyt botnet. The malicious data can be divided into 6 attacks by the two mentioned botnets. For each dataset, our model is

tested to detect 6 types of attacks.

4.2 Evaluation metrics

The experiments were conducted on two different data sets. We performed the model on Philips B120N10 Baby Monitor data set and Simple Home XCS7 1002 Security Camera data set. To assess DBSCAN-GWO in botnet detection, the following evaluation metrics were utilized:

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (9)$$

$$Presicion = \frac{TP}{(FP+TP)} \quad (10)$$

$$Recall = \frac{TP}{(TP+FP)} \quad (11)$$

$$F1 - Measure = \frac{2 \times Precision\ Recall}{(Precision+Recall)} \quad (12)$$

$$Detection\ Rate = \frac{TP}{TP+FN} \times 100 \quad (13)$$

TP, TN, FP, and FN represent the counts of true positives, true negatives, false positives, and false negatives, correspondingly. Accuracy refers to the number of correctly predicted data points, while precision measures the number of accurate positive predictions. Recall measures the number of correct positive predictions out of all possible positive predictions, and the detection rate represents the percentage of reported attacks that have been detected [21]. These metrics proved their efficiency as they are utilized in evaluating most ML based models [21-23].

4.3 Results of experiments

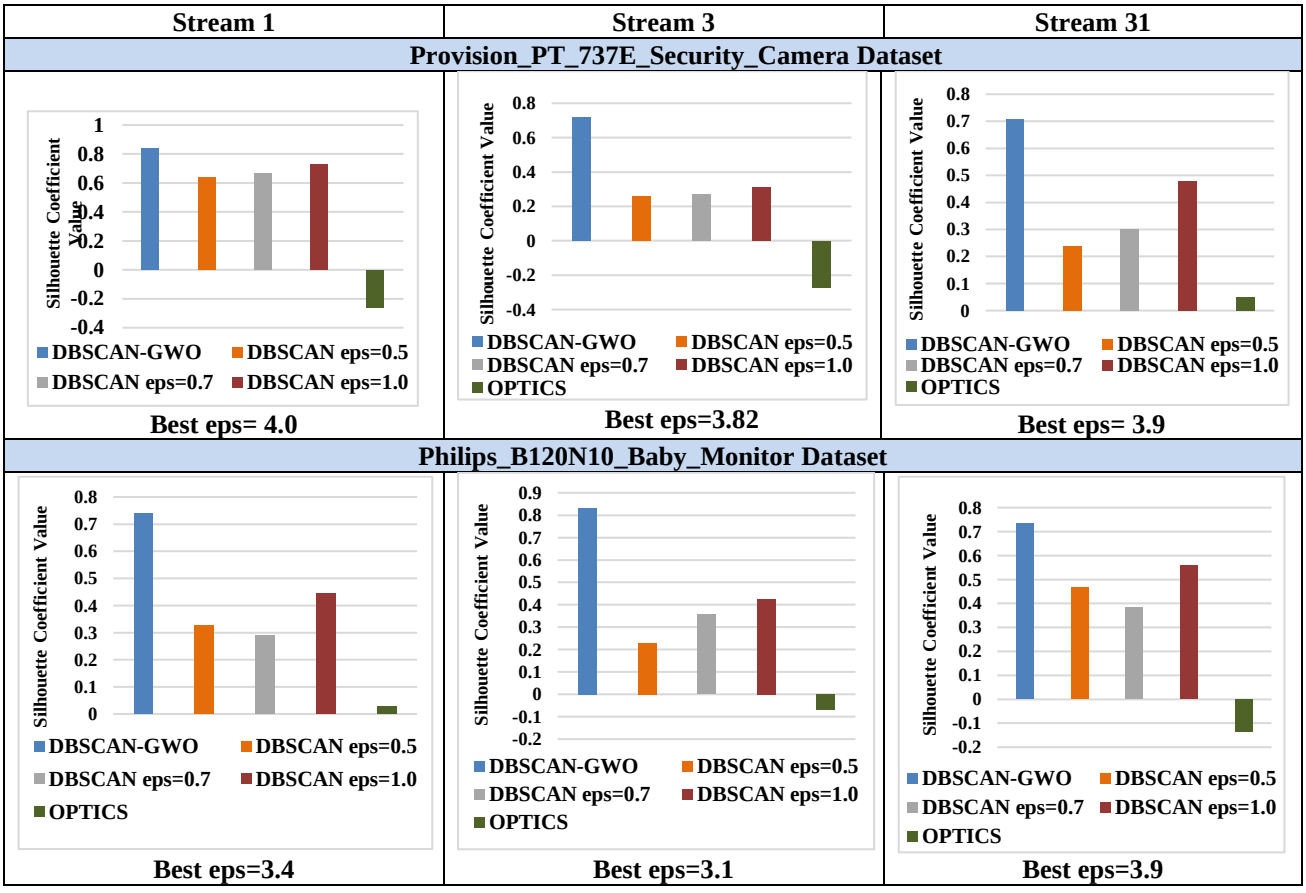
The effectiveness of the proposed DBSCAN-GWO method is assessed through experiments, which involve comparing the clustering quality and botnet detection accuracy with regular DBSCAN with different epsilon values and the OPTICS clustering algorithm. Two basic metrics were utilized for evaluation's sake. First is the Silhouette Coefficient metric and other evaluation metrics employed to assess the model's effectiveness in botnet detection accuracy. The findings indicate that the proposed model achieves the highest level of clustering quality measured by Silhouette Coefficient metric for each data stream. Unlike basic DBSCAN, where the value of eps is manually adjusted until the desired clusters are obtained, choosing an impropriate eps value gives bad clusters, and many important data points are considered as noise which will later cause misclassification problems as many normal data points are taken as noise and outliers. The proposed model automatically generates the optimal eps value for each dataset and data stream. Fig. 3 demonstrates that the proposed model produces the optimal Silhouette Coefficient value for each data streams. The accuracy of detecting botnets is heavily influenced by the generation of precise and high-quality clustering. To assess the clustering quality, we conducted evaluations using three random streams across six datasets, utilizing the Silhouette Coefficient metric. The DBSCAN-GWO algorithm yielded varying clustering quality for each stream, ranging from 0.1 to 0.8 for the first stream, 0.6 to 0.8 for the third stream, and 0.5 to 0.89 for stream 31. When employing DBSCAN with eps=0.5, the clustering quality for stream 1 ranged from -0.3 to 0.6, while for stream 3 it ranged from 0.2 to 0.6, and for stream 31 it ranged from -1.5 to 0.6. Applying DBSCAN with eps=0.7 resulted in clustering quality variations of -0.27 to 0.62 for stream 1, 0.3 to 0.6 for stream 3, and 0 to 0.6 for stream 31. Furthermore, utilizing DBSCAN with eps=1.0, the clustering quality varied from -0.15 to 0.7 for stream 1, 0.3 to 0.65 for stream 3, and 0.1 to 0.64 for stream 31. Notably, the OPTICS clustering algorithm exhibited the lowest clustering quality, with results ranging from -0.15 to 0.05 for stream 1, -0.25 to 0.23 for stream 3, and -0.25 to 0.1 for stream 31. These findings underscore the critical role of selecting an appropriate clustering algorithm for achieving accurate botnet detection.

Achieving high clustering quality will lead to perform correct clusters which will help in detecting the botnet clusters. Table 2 shows the results of the evaluation metrics of the proposed model in detecting botnets compared to DBSCAN with 3 different eps values chosen manually and OPTICS clustering algorithm for six datasets generated by six different IOT devices. Detection results show that the DBSCAN-GWO overcomes the rest of the algorithms in the accuracy of botnet identification and has the least misclassification percentage. The proposed DBSCAN-GWO model obtains accuracy of 0.97, 0.98, 0.83, 0.89, 0.88 and 0.91 for Provision_PT_737E_Security_Camera, Philips _B120N10Baby_Monitor, Samsung SNH_1011_NWebcam Dataset, Danmini_Doorbell Dataset, Simple Home_XCS7_1002_Security Camera Dataset and SimpleHome_XCS7_1003_WHT_Security_Camera respectively. Table 2 shows that our proposed method demonstrates superior performance in terms of detection accuracy, precision, f1 measure, and detection rate for compared to existing techniques.

The results vary across different devices due to the specific data values involved. Detecting botnet clusters becomes more challenging when the values of benign data are more similar to the values of botnet data. Our proposed model encounters another challenge posed by noisy datasets. Certain IoT devices generate a significant amount of noise, which can have a negative impact on the model's performance in creating high quality clusters and detecting botnets as shown in Table 2. Addressing these two challenges can be considered as potential areas for future work.

4.4 Comparative analysis

The DBSCAN-GWO model proposed in this study is compared with three other similar methods for attack and anomaly detection, as shown in Fig. 4. The first method [13] utilizes the DBSCAN clustering algorithm to identify anomalies in a fixed dataset, where the value of eps is manually set for the entire dataset. This method achieved a detection accuracy of 80%. The second method [24] employs the k-Means clustering algorithm along with a set of classifiers to create a semi-supervised approach for detecting DDOS attacks. Three different centroids were manually selected for k-Means. The accuracy of this method reached 79.6%. The third method [25], which combines clustering and classifiers for DDOS attack detection, is also included for comparison. In this approach, k-Means is used to cluster unlabeled data, and classifiers such as k-nearest neighbours (knn), support vector machines (SVM), and random forests (RF) are employed to classify labelled data points. This method achieves an accuracy of 96.66%. Previous studies suffer from several limitations, including the utilization of fixed datasets and the manual selection of parameters for clustering algorithms. In the last two studies, clustering algorithms are combined with classifiers. In contrast, our proposed method demonstrates its effectiveness in real-world botnet detection systems by performing online botnet detection with no need to any predefined labels. We leverage the DBSCAN algorithm, known for its excellence in data analysis, as the primary clustering algorithm. To ensure the inclusion of accurate data points within clusters, we automatically tune the DBSCAN parameter (eps) using the grey wolf optimizer (GWO). This adaptive tuning of eps is applied to each data stream. The combination of these factors significantly impacts botnet detection accuracy, resulting in an impressive 98% accuracy in our proposed model.



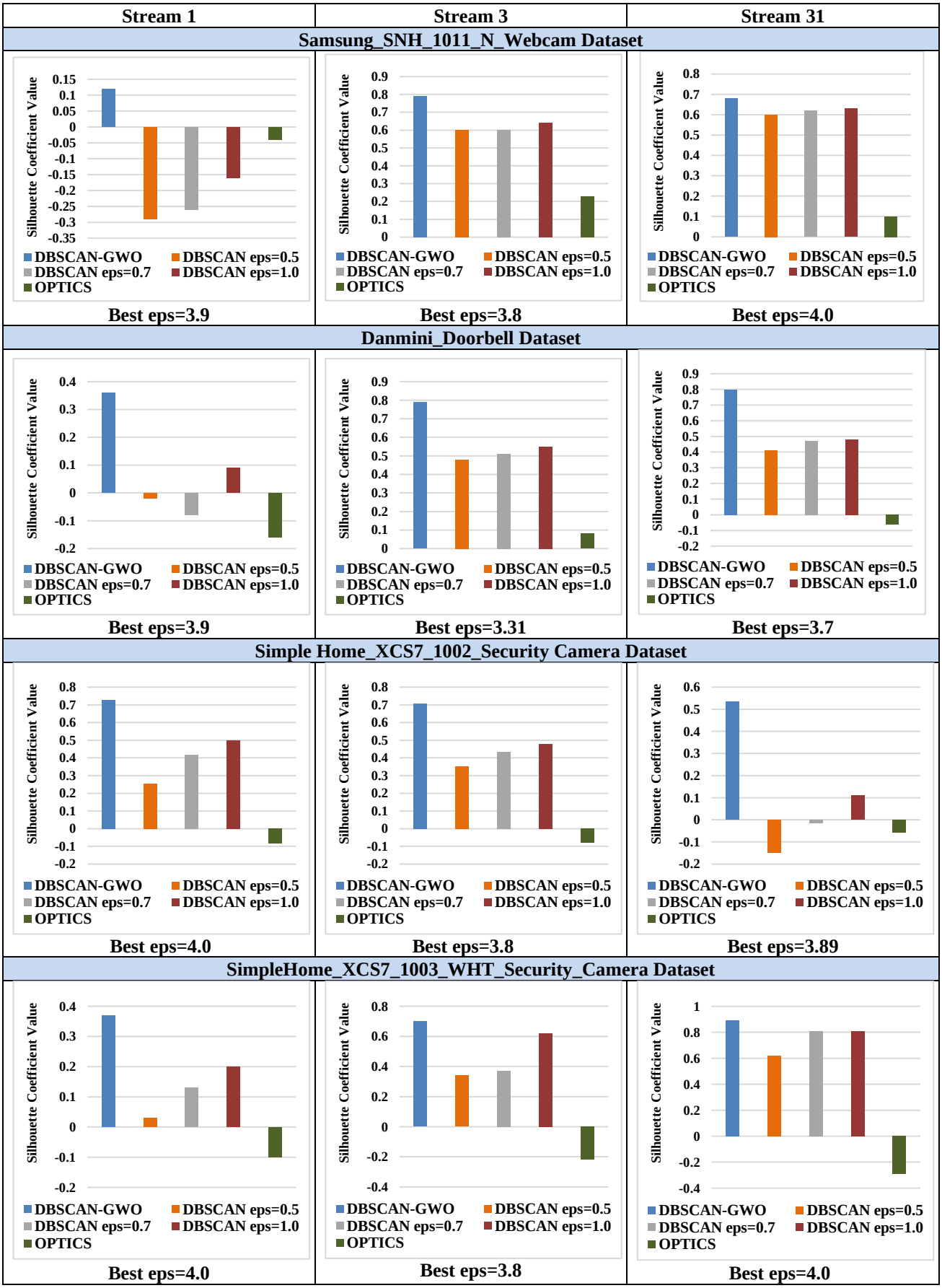


Figure. 3 Silhouette Coefficient value for the proposed model compared with other clustering techniques in 3 streams for six IOT devices' datasets

Table 2. Evaluation metrics for the proposed model compared to other clustering techniques in terms of accuracy

N_BaIoT DataSets for six devices	Clustering Algorithms	Accuracy	Precision	Recall	F1	Detection Rate (%)
Provision_PT_737E_ Security_Camera Dataset	OPTICS	0.88	0.39	0.42	0.41	61
	DBSCAN-eps=0.5	0.67	0.19	0.72	0.32	71
	DBSCAN-eps=0.7	0.69	0.2	0.72	0.3	71
	DBSCAN-eps=1.0	0.75	0.23	0.72	0.36	73
	DBSCAN-GWO	**0.974**	**0.64**	**0.99**	**0.78**	**99**
Philips_B120N10_ Baby_Monitor	OPTICS	0.55	0.17	0.96	0.29	97
	DBSCAN-eps=0.5	0.82	0.31	0.71	0.43	74
	DBSCAN-eps=0.7	0.84	0.31	0.47	0.38	57
	DBSCAN-eps=1.0	0.88	0.4	0.39	0.39	68
	DBSCAN-GWO	**0.98**	**0.91**	**0.98**	**0.94**	**98**
Samsung_SNH_ 1011_N_Webcam Dataset	OPTICS	0.53	0.12	0.66	0.2	60
	DBSCAN-eps=0.5	0.84	0.36	0.87	0.51	87
	DBSCAN-eps=0.7	0.86	0.37	0.78	0.5	79
	DBSCAN-eps=1.0	**0.88**	**0.42**	0.78	**0.55**	80
	DBSCAN-GWO	0.835	0.35	**0.98**	0.52	**97**
Danmini_Doorbell Dataset	OPTICS	0.54	0.14	0.75	0.24	67
	DBSCAN-eps=0.5	0.56	0.15	0.8	0.26	73
	DBSCAN-eps=0.7	0.62	0.18	0.8	0.29	75
	DBSCAN-eps=1.0	0.77	0.27	0.79	0.4	79
	DBSCAN-GWO	**0.89**	**0.46**	**0.83**	**0.59**	**84**
Simple Home_XCS7_ 1002_Security Camera Dataset	OPTICS	0.53	0.14	**0.81**	0.24	72
	DBSCAN-eps=0.5	0.60	0.16	**0.81**	0.27	76
	DBSCAN-eps=0.7	0.68	0.20	**0.81**	0.32	78
	DBSCAN-eps=1.0	0.723	0.22	**0.81**	0.35	79
	DBSCAN-GWO	**0.88**	**0.44**	0.80	**0.56**	**82.5**
SimpleHome_XCS7_1003_ WHT_Security_Camera	OPTICS	0.53	0.14	0.74	0.24	62
	DBSCAN-eps=0.5	0.77	0.27	**0.79**	0.41	78
	DBSCAN-eps=0.7	0.7	0.22	**0.79**	0.34	77

N_BaIoT DataSets for six devices	Clustering Algorithms	Accuracy	Precision	Recall	F1	Detection Rate (%)
	DBSCAN-eps=1.0	0.68	0.21	0.78	0.33	76
	DBSCAN-GWO	**0.91**	**0.55**	0.75	**0.63**	**79**

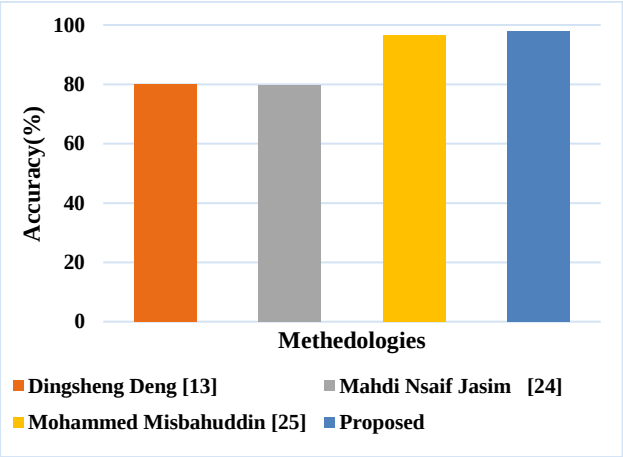


Figure. 4 Comparative analysis between the proposed method and other techniques

5. Conclusion

Botnet detection systems can be defined as software that monitors the traffic flow and detects viruses of type botnet whenever it is found. Since the data flow is in the form of the data stream, these systems must have online detection ability. Unsupervised ML (clustering) can process data streams. The suggested clustering algorithm, DBSCAN, efficiently finds arbitrarily shaped clusters and clusters with noise (outliers). The problem with DBSCAN is that it needs predefined parameters to perform, these parameters in most studies are set and changed manually to give the desired results. Manual parameter setting is not practical for building real-world botnet detection systems. In this study, grey wolf optimizer was proposed for the automatic selection of the eps (DBSCAN's parameter); finding the best value of eps depends on the ability of its value to maximize the objective function (cluster quality metric). For every data stream, a new value of eps is selected by GWO that suits this data stream. The appropriate choice of eps depends on the values in the dataset; eps can change for every data stream. The proposed model DBSCAN-GWO generates the best clusters in the data sets; generating good clusters affects the efficiency of finding the malicious clusters. After clustering the data stream, malicious clusters are recognized. Since it is unsupervised learning, no labels are available, so the biggest cluster is considered a benign and basic cluster because it represents the majority of data in the data stream. After recognizing the basic cluster, other clusters are compared with the basic cluster, and the clusters that are least similar to the basic cluster are considered malicious. Also, the noise cluster is considered as noise. This process is repeated for every data stream till the entire dataset is processed. The experimental outcomes demonstrate that the proposed DBSCAN-GWO algorithm achieves superior clustering results when compared to both the non-optimized DBSCAN and the OPTICS clustering algorithm. For botnet detection accuracy, the assessment criteria indicate that the proposed model surpasses other clustering algorithms and exhibits fewer misclassifications. Botnet detection Accuracy value varies from one dataset to another depending on the distance between malicious and benign data values. For further studies, the strategy of splitting the benign and botnet clusters can be improved, which will give higher accuracy in detecting botnets.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

The first author was responsible for the conceptualization, methodology, software development, validation, formal analysis, investigation, resource management, data curation, original draft preparation, as well as visualization of the results in the paper. The supervision, review and editing of the writing project administration, have been done by the second author.

References

[1] Z. Shao, S. Yuan, and Y. Wang, “Adaptive online learning for IoT botnet detection”, *Inf Sci (N Y)*, Vol. 574, pp. 84–95, 2021.

[2] M. Wazzan, D. Algazzawi, O. Bamasaq, A. Albeshri, and L. Cheng, “Internet of things botnet detection approaches: Analysis and recommendations for future research”, *Applied Sciences (Switzerland)*, Vol. 11, No. 12, MDPI AG, 02, 2021.

[3] Y. Xing, H. Shu, H. Zhao, D. Li, and L. Guo,

"Survey on Botnet Detection Techniques: Classification, Methods, and Evaluation", *Mathematical Problems in Engineering*, Vol. 2021. Hindawi Limited, 2021.

[4] I. A. Abdulmajeed and I. M. Husien, "Machine Learning Algorithms and Datasets for Modern IDS Design", In: *Proc. of 2022 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, Malang, Indonesia, pp. 335-340, 2022.

[5] G. Yu, Z. Cai, S. Wang, H. Chen, F. Liu, and A. Liu, "Unsupervised Online Anomaly Detection with Parameter Adaptation for KPI Abrupt Changes", *IEEE Transactions on Network and Service Management*, Vol. 17, No. 3, pp. 1294–1308, Sep 2020.

[6] H. P. Kriegel, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering", *Wiley Interdiscip Rev Data Min Knowl Discov*, Vol. 1, No. 3, pp. 231–240, May 2011.

[7] L. Wang, H. Wang, X. Han, and W. Zhou, "A novel adaptive density-based spatial clustering of application with noise based on bird swarm optimization algorithm", *Comput Commun*, Vol. 174, pp. 205–214, Jun 2021.

[8] I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions", *SN Computer Science*, Vol. 2, No. 3, Springer, May 01, 2021.

[9] A. Muhammad, M. Asad, and A. R. Javed, "Robust Early Stage Botnet Detection using Machine Learning", In: *Proc. of 2020 International Conference on Cyber Warfare and Security (ICCWS)*, Islamabad, Pakistan, pp. 1-6, 2020.

[10] J. Cui, J. Zhang, J. He, H. Zhong, and Y. Lu, "DDoS detection and defense mechanism for SDN controllers with K-Means", In: *Proc. of 2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC)*, Leicester, UK, pp. 394-401, 2020.

[11] A. Lang and E. Schubert, "BETULA: Fast clustering of large data with improved BIRCH CF-Trees", *Inf Syst*, Vol. 108, Sep 2022.

[12] S. Subudhi and S. Panigrahi, "Application of OPTICS and ensemble learning for Database Intrusion Detection", *Journal of King Saud University – Computer and Information Sciences*, Vol. 34, No. 3, pp. 972–981, Mar 2022.

[13] D. Deng, "Research on Anomaly Detection Method Based on DBSCAN Clustering Algorithm", In: *Proc. of 2020 5th International Conference on Information Science, Computer Technology and Transportation (ISCTT)*, Shenyang, China, pp. 439-442, 2020.

[14] P. Barthakur, M. Dahal, and M. K. Ghose, "A Framework for P2P Botnet Detection Using SVM", In: *Proc. of 2012 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, Sanya, China, pp. 195-200, 2012.

[15] A. Karami and R. Johansson, "Choosing DBSCAN parameters automatically using differential evolution", *International Journal of Computer Applications*, Vol. 91 No. 7, pp. 1-11, 2014.

[16] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer", *Advances in Engineering Software*, Vol. 69, pp. 46–61, 2014.

[17] S. Mirjalili, I. Aljarah, M. Mafarja, A. A. Heidari, and H. Faris, "Grey wolf optimizer: Theory, literature review, and application in computational fluid dynamics problems", *Studies in Computational Intelligence*, Springer Verlag, pp. 87–105, 2020.

[18] J. A. Fessler, "Optimization Methods for Magnetic Resonance Image Reconstruction: Key Models and Optimization Algorithms", *IEEE Signal Processing Magazine*, Vol. 37, No. 1, pp. 33-40, Jan 2020.

[19] M. Pant, T. Radha, and V. P. Singh, "Particle Swarm Optimization Using Gaussian Inertia Weight", In: *Proc. of International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, Sivakasi, India, pp. 97-102, 2007.

[20] Y. Meidan, "N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders", *IEEE Pervasive Computing*, Vol. 17, No. 3, pp. 12-22, Jul.-Sep 2018.

[21] I. A. Abdulmajeed and I. M. Husien, "MLIDS22- IDS Design by Applying Hybrid CNN-LSTM Model on Mixed-Datasets", *Informatica (Slovenia)*, Vol. 46, No. 8, pp. 121–134, Oct 2022.

[22] M. Alauthaman, N. Aslam, L. Zhang, R. Alasem, and M. A. Hossain, "A P2P Botnet detection scheme based on decision tree and adaptive multilayer neural networks", *Neural Comput & Applic*, Vol. 29, No. 11, pp. 991–1004, Jun 2018.

[23] M. G. Karthik and M. B. M. Krishnan, "Detecting Internet of Things Attacks Using Post Pruning Decision Tree-Synthetic Minority Over Sampling Technique", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 4, pp. 105–114, Aug 2021, doi: 10.22266/ijies2021.0831.10.

[24] M. N. Jasim and M. T. Gaata, "K-Means clustering-based semi-supervised for DDoS attacks classification", *Bulletin of Electrical*

Engineering and Informatics, Vol. 11, No. 6, pp. 3570–3576, Dec 2022.

[25] M. Aamir and S. M. A. Zaidi, "Clustering based semi-supervised machine learning for DDoS attack classification", *Journal of King Saud University - Computer and Information Sciences*, Vol. 33, No. 4, pp. 436–446, May 2021.