Intelligent Automation & Soft Computing

Tech Science Press

DOI: 10.32604/iasc.2023.036247
Article

An Efficient Approach Based on Remora Optimization Algorithm and Levy Flight for Intrusion Detection

Abdullah Mujawib Alashjaee*

Department of Computer Sciences, Faculty of Computing and Information Technology, Northern Border University, Rafha, 91911, Kingdom of Saudi Arabia

*Corresponding Author: Abdullah Mujawib Alashjaee. Email: abdullah.alashjaee@nbu.edu.sa

Received: 22 September 2022; Accepted: 06 January 2023

Abstract: With the recent increase in network attacks by threats, malware, and other sources, machine learning techniques have gained special attention for intrusion detection due to their ability to classify hundreds of features into normal system behavior or an attack attempt. However, feature selection is a vital preprocessing stage in machine learning approaches. This paper presents a novel feature selection-based approach, Remora Optimization Algorithm-Levy Flight (ROA-LF), to improve intrusion detection by boosting the ROA performance with LF. The developed ROA-LF is assessed using several evaluation measures on five publicly available datasets for intrusion detection: Knowledge discovery and data mining tools competition, network security laboratory knowledge discovery and data mining, intrusion detection evaluation dataset, block out traffic network, Canadian institute of cybersecurity and three engineering problems: Cantilever beam design, three-bar truss design, and pressure vessel design. A comparative analysis between developed ROA-LF, particle swarm optimization, salp swarm algorithm, snake optimizer, and the original ROA methods is also presented. The results show that the developed ROA-LF is more efficient and superior to other feature selection methods and the three tested engineering problems for intrusion detection.

Keywords: Feature selection; metaheuristic algorithms; intrusion detection; Remora optimization algorithm; Levy flight

1 Introduction

With the increased use of internet services, cybersecurity issues have become one of the most serious challenges that pose specific risks not only to individuals but also to business operations [1]. A variety of security mechanisms, such as firewalls, intrusion detection prevention systems, encryptions, and antivirus, are used by organizations and enterprises to deal with such cybersecurity attacks on their networks [2–4]. These mechanisms prove themselves as powerful methods for preventing many types of attacks. However, they cannot perform analysis for every network packet, and thus they cannot reach the desired detection performance [5]. To overcome these shortcomings and achieve optimal security

requirements for a network, researchers employed Machine Learning (ML) approaches to look inside packet payloads and detect such attacks with high accuracy and a low false positive rate [6].

Selecting an Optimal Feature Subset (OFS) assists the learning process by ML techniques to achieve better performance results. Nature-inspired algorithms are mostly Meta-Heuristics (MH) optimization methods inspired by nature. These methods gained special attention from many scholars in different applications due to their great potential to specify OFS [7]. They are also effective, reliable, and gradient-free stochastic optimization techniques that have successfully solved various numerical and combinatorial optimization problems with diverse frameworks [8,9].

MH inspiration sources are broken down into three types [10,11]: swarm-based algorithms, evolutionary-based algorithms, and physics-based algorithms. Some popular MH methods, including Multi-Verse Optimizer (MVO) [12], Particle Swarm Optimization (PSO) [13], Salp Swarm Algorithm (SSA) [14], genetic algorithm [15], whale optimization algorithm [16], Snake Optimizer (SO) [17], ROA [18], are some examples of applied MH methods for Feature Selection (FS).

MH algorithms can be combined to achieve better results in different applications. The authors in [19] combined the reptile search algorithm with ROA for data clustering. In another work [20], a modified version of the ROA method using Brownian motion is introduced for image segmentation. In [21], ROA with an autonomous foraging mechanism is used to explore search space and effectively enhance global optimization solutions of the ROA. In [22], the authors combined Gorilla Troops Optimizer (GTO) with Bird Swarms (BS) to boost the capability of the GTO for FS. They evaluated their proposed GTO-BS using several evaluation measures on four Intrusion Detection (ID) datasets: Network Security Laboratory Knowledge Discovery and Data Mining (NSL-KDD), Block out traffic network (Botnet), Canadian Institute of Cybersecurity (CIC-IDS-2017), University of New South Wales Network Botnet (UNSW-NB15) and Botnet-IoT. In [23], an efficient FS method named Dynamic Feature Selector (DFS) is introduced for filtering insignificant variables. The DFS used statistical analysis and feature importance tests to reduce model complexity and improve prediction accuracy using two ID datasets.

MH methods use two principles that are characteristic of all optimization techniques, which are exploration and exploitation. In the first principle, the algorithm attempts to discover different regions in the search area, while the exploitation searches around the obtained solution from the first phase to find the best candidates. However, experiment results show that ROA is weak in exploring search space broadly. In this paper, an improved version of ROA, namely ROA-LF, is presented for the purpose of selecting OFS for the application of ID. The ROA-LF combines the original ROA with LF to enhance the exploration process and maintain a balance between exploration and exploitation in the original ROA method's structure. The main contributions of this work could be summarized as follows:

- An improved version of ROA using LF, named ROA-LF, is proposed for ID,
- LF strategy is applied to enhance the ability of the ROA to explore search space more effectively and avoid getting stuck in local optima,
- The ROA-LF is examined using five open-access datasets for ID and three well-known engineering optimization problems,
- The ROA-LF's efficacy is confirmed when compared to other MH methods and the tested engineering problems.

The remainder of this paper is organized as follows: Section 2 provides a brief overview of ROA and LF. Section 3 describes the developed ROA-LF method, followed by the experimental results

and statistical comparison with other popular FS methods shown in Section 4. Section 5 presents the conclusion of this work.

2 Method

This section provides an overview of the ROA and LF.

2.1 ROA

ROA [18] is a new MH method that mimics the concept of parasitism of Remora. Exploration and exploitation of ROA are briefly described in this section.

2.1.1 Exploration (Free Travel)

■ Swordfish Optimization Strategy (SFO)

In the case where the Remora sticks to the swordfish, its location is updated using the following:

$$R_i^{t+1} = R_{ibest}^t - (rand\,(0,1)) \left(\frac{R_{ibest}^t - R_{rand}^t}{2} - R_{rand}^t \right) \tag{1}$$

where t is the number of current iterations; R_{ibest}^t refers to the best-obtained solution and R_{rand}^t indicates a random location, and *rand* (0, 1) is a random number in the range of 0–1.

■ Attack experience

Remora takes small steps in the vicinity of the host end to identify whether to change or not change the host based on fitness. This behavior mathematically can be presented as:

$$R_{att} = R_i^t + \left(R_i^t - R_{pre} \right) * randn \tag{2}$$

where R_{att} and R_{pre} are the position of the previous generation and the test step, respectively, and *randn* is the small global random step of the Remora.

Then Remora randomly checks the change in the fitness values between the current response $\left(f(R_i^t)\right)$ and the tested response $(f(R_{att}))$. If $(f(R_i^t) > f(R_{att})$, then the Remora selects one of the feeding methods for local optimization, while if $(f(R_i^t) < f(R_{att})$, Remora picks the host.

2.1.2 Exploitation (Thoughtful Nutrition)

■ WOA Strategy

According to the WOA, the position of the Remora attached to the whale is updated as follows:

$$R_{i+1} = D * exp^{\alpha} * \cos(2\pi x) + R_i \tag{3}$$

$$\alpha = rand\,(0,1) * (\alpha - 1) + 1 \tag{4}$$

$$\alpha = -\left(1 + \frac{t}{T}\right) \tag{5}$$

$$D = \lceil R_{best} - R_i \rceil * \mathrm{R} \tag{6}$$

where D presents the distance between the hunter and prey, α is a random number $[-1,\ 1]$, α is a linear number [=1 and −2], and t is the number of iterations.

■ Host nutrition

"Host feeding" is a small step in the exploitation process, which creates a solution space that converges gradually around the host, refining and enhancing the ability of local optimization. This stage can be mathematically modeled as follows:

$$R_i^t = R_i^t + A \tag{7}$$

$$= B * \left(R_i^t - C * R_{best}\right) \tag{8}$$

$$B = 2 * V * rand\ (0, 1) - V \tag{9}$$

$$V = 2\left(1 - \frac{t}{T}\right) \tag{10}$$

where A is a small step between the fish adhesive and the host, C is the coefficient of stickiness to indicate its position, and it is within the range of [0, 0.3].

2.2 LF

LF is a linear combination of two random independent variables (y_1, y_2), identically distributed with the same Probability Density Function (PDF) and is defined as [24–29]:

$$L_{y,ld}\,(y) = \sqrt{\frac{y}{2\pi}}\ \frac{1}{(y - ld)^{\frac{3}{2}}}\ exp\left(-\frac{y}{2y - ld}\right),\ \cdots ld < y < \infty \tag{11}$$

where y is a scale parameter and ld is the location parameter of the Levy distribution.

LF, which is used to produce a random walk, has step lengths (Sl) that are drawn from a Levy distribution length density distribution, and it can be given as:

$$L_\alpha\,(Sl) \approx \frac{1}{S^{\beta+1}}\ \cdots \lfloor S \rfloor\ "1 \tag{12}$$

where β is a power law.

In the first stage of LF, stochastic variables σ_{y_1} and σ_{y_2} with standard deviations are generated

$$\sigma_{y_1}\,(\alpha) = \left[\frac{G\,(1 + \beta)\,sin\,[\pi\beta/2]}{G\,(1 + \beta/2)\,\beta 2^{\beta-1/2}}\right]^{\frac{1}{\beta}} \quad \text{and} \quad \sigma_{y_2} = 1 \tag{13}$$

where $G\,(.)$ is the gamma function, and then the variable V is generated using,

$$V = \frac{y_1}{|t_2|^{1/\beta}} \qquad 1 < \beta < 2 \tag{14}$$

3 Proposed Method

This section explains the structure of the developed ROA-LF, which combines ROA and LF. Like any MH, ROA suffers from a balance between exploration and exploitation, which leads to it being trapped in a local optimum. To tackle this weakness and to enhance the global and local searching capability of the ROA, LF is used. The LF is integrated into the ROA's structure to extend its search ability and make it capable of visiting new locations in the search space. This helps the ROA to avoid becoming trapped in locally optimal solutions and balance between exploration and exploitation. The flowchart of the introduced ROA-LF is provided in Fig. 1, and the pseudocode is in Algorithm 1.

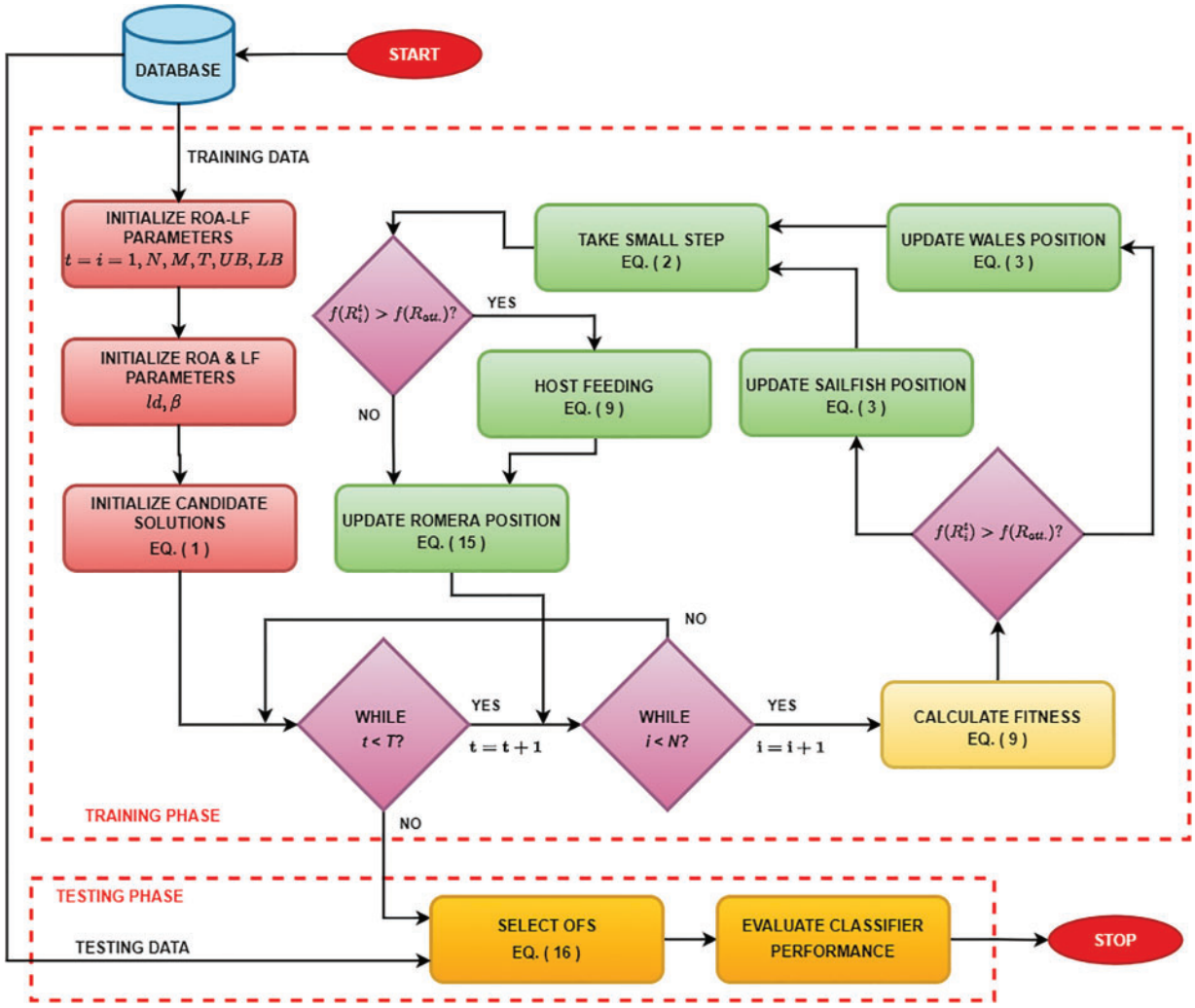

Figure 1: Structure of the ROA-LF approach

Initially, the dataset is divided into two mutually exclusive and exhaustive subsets: training and testing. The training data is used for optimization and classifier training, while the testing data is used for performance evaluation. The entire method can be understood in two phases: the training phase and the testing phase. The training phase starts by initializing hyper-parameters of the method, such as the maximum number of iterations T, problem dimensionality M, and some constants of ROA and LF methods. The training data's lower LB and upper UB limits are calculated for each dimension. Further, N Romera positions are initialized randomly in the range LB–UB, as in Eq. (1).

Fitness values for all candidate solutions are calculated using Eq. (9). If Romera's previous position is better than the updated positions, then the Whale position update is implemented. Else Sailfish position update is implemented. To check if the uodate is optimum or not, a small step is taken in the test direction. The fitness value of Romera's position and the newly tested position is calculated. If the new test direction results in a larger fitness value, then Host feeding is implemented. If the test direction provides a smaller fitness value, then Romera's position is updated using LF.

LF is introduced into the exploration phase of the ROA to enhance its exploration ability further. For t^{th} iteration and i^{th} Romera is updated after LF as follows:

$$R_i(t) = R_i(t) + V(t) \odot R_i(t) \tag{15}$$

where $R_i(t)$ is candidate solution of i^{th} Romera at t^{th} iteration, V is the LF parameter, and $\odot$ indicates the dot product. The above process is repeated for all Romeras. When all Romera positions are updated, the global best solution is saved, and the next iteration starts. For new iterations, the entire process is repeated. The optimization stops when the maximum number of iterations T is reached.

The performance of the updated Romera position fitness is calculated by using the Fitness Function (FF), which is a K-Nearest Neighbor (KNN) classifier with five neighbors and a threshold value of 0.5, as recommended by the work of [30]. The Romera with the smallest fitness as a result of the least number of selected features and maximum accuracy is the best one and is defined as:

$$FF(R_i) = \lambda \times E + (\lambda - 1) \times \frac{|OFS_i|}{M} \tag{16}$$

where E is the classification error rate of the KNN classifier with five neighbors, $|OFS_i|$ is the number of selected features and M is the total number of features in the dataset, and λ controls the relative importance of classification error and the number of selected features. The value of α varies in the range of [0,1] and is set to 0.99, as recommended by [31].

The global best Romera's position is used to generate OFS by simple thresholding. It can be noted that a universal threshold of 0.5 is used during optimization, and the absolute value of the threshold used during the optimization does not change the OFS.

Algorithm 1: Pseudocode of the developed ROA-LF algorithm.

1. Group the complete data into mutually exclusive & exhaustive training & testing sets.
Training Phase
2. Load training examples
3. Calculate UB and LB and decide fitness function FF (f)
4. Initialize ROA and LF parameters ld, β
5. Initialize Romeras Eq. (1)
6. for $t = 1$ to T do
7. for $i = 1$ to N do
8. Calculate the fitness value (f) of the current Romera using Eq. (16)
9. if $f(R_i^t) > f(R_{att})$ then
10. Update the position of attached whales using Eq. (3)
11. else
12. Update the position of the attached Sailfishes using Eq. (1)
13. end if
14. Make a one-step prediction by Eq. (2);
15. if $(f(R_i^t) > f(R_{att})$ then
16. Host feeding mode for Romera using Eq. (9)
17. end if
18. Update the Romera position using Eq. (15)
19. end for
20. end for

(Continued)

Algorithm 1: Continued

21. Use a threshold of 0.5 to calculate OFS for best Romera
Testing Phase
22. Load testing examples
23. Use OFS to select only significant features
24. Evaluate the classifier performance

The testing phase starts with test data. The number of features is optimized using the position of the final global-best Romera received at the end of the training phase. The OFS of the test data is given as input to the trained classifier for performance evaluation.

4 Experimental Results

For assessing the effectiveness of the introduced ROA-LF, its capability is compared with other methods comprising PSO [13], SSA [14], SO [17], and ROA [18] on five ID datasets, and the results are provided in this section. Python scikit-learn environment setup on Windows 10 operating system with 32 GB RAM and 3.13 GHz processor speed is used to implement the experiments.

4.1 Parameter Settings

The ROA-LF is compared with other popular FS methods, with the number of expected candidate solutions and the maximum iterations set to 20 and 100, respectively. Also, each method is executed for 20 independent runs for statistically significant results. The MH methods used for comparison include PSO, SSA, SO, and ROA. The parameter setup of these MAs is detailed in Table 2. The parameter selection was based on the parameters used by the original author in the article or the parameters widely used by various researchers.

Table 1: Parameter settings

Method	Parameters
PSO	$c_1 = c_2 = 2$, $w_{min} = 0.1$ and $w_{max} = 0.9$
SSA	c_2 and c_2 are random values in range 0–1
SO	$c_1 = 0.5$, $c_2 = 0.05$, $c_3 = 2$, x_{max}, & x_{min} as per the dataset
ROA	$ld = 1$ and $\beta = 2$

Table 2: Characteristics of three standard open-access datasets used in the experiments

Dataset	Instances	Features	Classes	Domain
BreastEW	569	30	2	Biology
Churn	3150	16	2	Telecom
HeartEW	270	13	2	Biology

4.2 Standard Datasets

For quantitative and qualitative evaluation of the introduced ROA-LF, three standard open-access datasets from the UCI repository, suggested by several researchers in the literature, are used. Table 2 summarizes the details of the used datasets. For optimization experiments, each dataset is grouped into 80%–20% of the samples used for training and testing, respectively.

Table 3 summarizes the performance comparison of the ROA-LF and other methods in terms of statistical inferences of fitness values for three standard datasets. All methods are also ranked based on average, STD, best, fitness, and worst fitness values, in that order. Remember that a method with the smallest fitness value will be ranked first and vice versa. The table shows that the ROA-LF gained the first rank in all three datasets, indicating its superior performance over other methods. On average, SSA shows the second-best performance for all three datasets, followed by ROA and SO, respectively. PSO obtained the worst rank for all three datasets, indicating poor performance compared to other methods. These results prove the ROA-LF's capability to sustain a stable balance between the two main principles of MH methods.

Table 3: Best, worst, avg, and STD fitness values obtained by different methods

Dataset	Metric	PSO	SSA	SO	ROA	RSA-SO
BreastEW	Best	0.0492	0.0401	0.0491	0.0436	**0.0382**
	Worst	0.0562	0.0579	0.0578	0.0578	0.0491
	Avg.	0.0492	0.0421	0.0491	0.0436	**0.0401**
	STD.	0.0019	0.0044	0.0026	0.0044	**0.0018**
	Rank	5	2	4	3	1
Churn	Best	0.0418	0.0406	0.0415	0.0403	**0.0393**
	Worst	0.1346	0.0491	0.1346	0.0817	**0.0484**
	Avg.	0.0418	0.0406	0.0415	0.0403	**0.0393**
	STD.	0.0354	0.0025	0.0276	0.0119	**0.0018**
	Rank	5	3	4	2	**1**
HeartEW	Best	0.2865	0.0002	0.0003	0.0001	**0.0000**
	Worst	0.2692	0.0001	0.0002	0.0001	**0.0000**
	Avg.	0.1983	**0.0000**	0.0001	0.0001	**0.0000**
	STD.	0.0248	**0.0000**	0.0001	0.0000	**0.0000**
	Rank	5	2	4	3	**1**

Tables 4 and 5 compare all methods in terms of the testing accuracy and the number of selected features i.e., OFS. In Table 4, the ROA-LF shows the highest accuracy compared to other methods for all three datasets. The improved exploration of ROA can interpret as this because of LF integration. SO performs the second best, followed by SSA, ROA, and PSO.

The comparative analysis using OFS is shown in Table 5 for all three datasets. The ROA-LF selected the least number of features in OFS for all three datasets. This confirms the efficiency of the proposed ROA-LF in eliminating features that are not significant for binary classification. SSA

shows the second smallest OFS, followed by SO and ROA. PSO selects the highest number of features in OFS and hence, the least perming method.

Table 4: Results of RSA-SO and other methods in terms of classification accuracy

Dataset	PSO	SSA	SO	ROA	ROA-LF
BreastEW	95.4001	95.6496	95.3722	95.3726	**96.1319**
Churn	89.6476	96.3150	93.2201	92.9740	**96.3486**
HeartEW	73.7800	99.2293	99.3587	84.5661	**99.6521**

Table 5: Comparison between RSA-SO and other methods in terms of average OFS

Dataset	PSO	SSA	SO	ROA	ROA-LF
BreastEW	3	3	3	9	**2**
Churn	14	11	12	13	**11**
HeartEW	13	3	5	5	**1**

4.3 ID Datasets Descriptions

Five real datasets from ID applications are selected to assess ROA-LF efficiency. These datasets are widely used for ID [22,23], and they include Knowledge Discovery and Data Mining Tools Competition (KDD-CUP99), NSL-KDD, Intrusion Detection Evaluation Dataset (ISCXIDS2012), Botnet, and CIC-IDS2018. The main characteristics of those datasets are given in Table 6.

Table 6: The characteristics of the datasets

Dataset	Source	No. of features	No. of samples
KDD-CUP99	[32]	43	494,020
NSL-KDD	[33]	43	125,973
ISCXIDS2012	[34]	8	11,68,079 (train) + 6,29,274 (test)
Botnet	[35]	8	77,796 (train) + 1,63,660 (test)
CIC-IDS2018	[36]	80	1,048,575

The KDD-CUP99 dataset includes Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R), and probing attack properties. It contains seven weeks of network traffic, has about five million lines, and is one of the most widely used datasets for ID assessment. It contains 43 features and 494,020 samples.

The NSL-KDD is an upgraded version of KDD-CUP99, with a 43-dimensional feature in each record. It does not contain unnecessary and repetitive records according to the original KDD-CUP99 dataset and uses the same properties as the KDD-CUP99. It contains 43 features and 125,973 samples.

The ISCXIDS2012 dataset comprises seven days from Friday, 11/6/2010, to Thursday, 17/6/2010, of routine and malicious network activities collected using 21 interconnected Windows workstations. The dataset is labeled for normal (2,381,532) and malicious (68,792) activities. A variety of multi-stage

attacks were simulated to generate traces. It contains eight features and 11,68,079 (train) + 6,29,274 (test) samples.

The botnet dataset comprises non-overlapping subsets of the ISOT dataset created by merging different available datasets, such as the French chapter of the Honeynet Project, Ericsson Research in Hungary, and Lawrence Berkeley National Laboratory. It comprises traces of malicious botnets (Storm and Zeus) and everyday activities. 15% and 25% of the ISOT dataset are used in training and test datasets. A subset of normal activities from the ISCXIDS2012 dataset is used in the training dataset. In addition, a subset of normal and botnet activities is included in the test dataset. Four botnet traces (Neris, RBot, Virut, and NSIS) from Botnet traffic produced by the Malware Capture Facility Project are included in the training dataset, while seven botnets (Neris, RBot, Virut, NSIS, Menti, Sogou, and Murlo) are included in the test dataset. It contains eight features and 77,796 (train) + 1,63,660 (test) samples.

The CIC-IDS2018 dataset includes seven attacks: Brute-force, Heartbleed, Botnet, DoS, DDoS, Web attacks, and network infiltration from inside. An infrastructure of 50 machines is used to attack 420 machines and 30 servers from 5 departments of the victim organization. The dataset captures each machine's network traffic and system logs and is represented using 80 features extracted from the captured activities using CICFlowMeter-V3. It contains 80 features and 1,048,575 samples.

The datasets contain many records for routine activities and network attacks. Using an iterative FS such as MH methods will be computationally expensive. Hence, only 10% of the dataset is used for FS evaluation while maintaining the ratio of natural activities and network attacks.

4.4 Experimental Results and Discussion

In order to examine the effectiveness of the ROA-LF as an FS method, the real-world datasets provided in Table 1 are used, and its efficacy is evaluated using fitness values (best, worst, average (Avg.), standard deviation (STD.)), classification accuracy, and the number of the OFS.

Table 7 provides a summary of the obtained results by the ROA-LF against the other methods. The Friedman test is performed for ranking the MH methods, and ranks are presented in the table. The ROA-LF gives the best fitness values in four datasets and the smallest, worst fitness value in three out of five datasets, while the original ROA achieved both the best and worse fitness values on the CIC-IDS2018 dataset. Also, the ROA-LF has both better Avg and STD of fitness values in four datasets and achieved the first rank in four datasets. The PSO ranked first in one dataset, while SSA achieved the best STD result for the ISCXIDS2012 dataset. These results prove the ROA-LF's stability in balancing the exploration and exploitation principles.

Table 7: Fitness values achieved by different MH methods for five publicly available ID datasets

Dataset	Metric	PSO	SSA	SO	ROA	ROA-LF
KDD-CUP99	Best	0.0338	0.0322	0.0249	0.0214	**0.0191**
	Worst	0.0516	0.0536	0.0404	0.0474	**0.0392**
	Avg.	0.0335	0.0398	0.0277	0.0307	**0.0263**
	STD.	0.0096	0.0069	0.0074	0.0093	**0.0066**
	Rank	4	5	2	3	**1**

(Continued)

Table 7: Continued

Dataset	Metric	PSO	SSA	SO	ROA	ROA-LF
NSL-KDD	Best	0.0606	0.0596	0.0624	0.0586	**0.0581**
	Worst	0.0718	0.0678	0.0723	0.0680	**0.0662**
	Avg.	0.0610	0.0613	0.0653	0.0610	**0.0598**
	STD.	0.0081	0.0053	0.0048	0.0033	**0.0021**
	Rank	3	4	5	2	**1**
ISCXIDS2012	Best	0.0417	0.0421	0.0421	0.0403	**0.0392**
	Worst	0.0639	0.0583	0.0614	0.0662	**0.0638**
	Avg.	0.0537	0.0523	0.0523	0.0558	**0.0520**
	STD.	0.0073	**0.0058**	0.0061	0.0102	0.0086
	Rank	4	2	3	5	**1**
Botnet	Best	0.0463	0.0481	0.0472	0.0427	**0.0423**
	Worst	0.0545	0.0597	0.0568	**0.0509**	0.0521
	Avg.	0.0486	0.0506	0.0492	0.0469	**0.0453**
	STD.	0.0030	0.0025	0.2019	0.0028	**0.0024**
	Rank	3	5	4	2	**1**
CIC-IDS2018	Best	0.0341	0.0362	0.0372	**0.0303**	0.0326
	Worst	0.0600	0.0539	0.0581	**0.0444**	0.0509
	Avg.	**0.0344**	0.0407	0.0422	0.0349	0.0371
	STD.	0.0072	0.0053	0.0061	0.0053	**0.0045**
	Rank	**1**	4	5	2	3

Table 8 compares different MH algorithms in terms of mean and Std of accuracy. The developed ROA-LF shows the least STD accuracy in all used datasets, which reflects the stability of the ROA-LF compared to PSO, SSA, SO, and ROA. The mean of accuracy is the highest for the developed ROA-LF in three out of five datasets. The SSA method gained the best mean accuracy result in the ISCXIDS2012 dataset and PSO in the CIC-IDS2018 dataset. Overall results indicate that the LF strategy improves the ROA's performance.

Table 8: Classification accuracy of the developed ROA-LF and other MH methods

Dataset	Measure	Method				
		PSO	SSA	SO	ROA	ROA-LF
KDD-CUP99	Mean	0.9756	0.9685	0.9828	0.9749	**0.9788**
	STD	0.0085	0.0058	0.0056	0.0080	**0.0051**
NSL-KDD	Mean	0.9473	0.9472	0.9418	0.9466	**0.9489**
	STD	0.0072	0.0047	0.0054	0.0047	**0.0034**
ISCXIDS2012	Mean	0.9521	**0.9547**	0.9535	0.9487	0.9525
	STD	0.0048	0.0071	0.0056	0.0078	**0.0042**

(Continued)

Table 8: Continued

Dataset	Measure	Method				
		PSO	SSA	SO	ROA	ROA-LF
Botnet	Mean	0.9572	0.9565	0.9579	0.9577	**0.9606**
	STD	0.0055	0.0063	0.0052	0.0053	**0.0041**
CIC-IDS2018	Mean	**0.9709**	0.9656	0.9644	0.9704	0.9677
	STD	0.0060	0.0042	0.0049	0.0042	**0.0035**

The results of the proposed ROA-LF and the other MH algorithms based on the mean and STD of the OFS selected by the corresponding MH algorithm are shown in Table 9. The ROA-LF selected the least mean OFS in four out of five datasets, while for ISCXIDS2012 dataset, PSO, SA, ROA, and the developed ROA-LF selected the same mean number of OFS. Similarly, the STD of the number of OFS is the least by ROA-LF in four of five datasets, indicating better stability. For the ISCXIDS2012 dataset, SO, ROA and ROA-LF show similar STD of OFS.

Table 9: Average OFS of the developed ROA-LF and other MH methods

Dataset	Measure	Method				
		PSO	SSA	SO	ROA	ROA-LF
KDD-CUP99	Mean	40	37	46	25	**23**
	STD	5	8	8	6	**4**
NSL-KDD	Mean	38	39	33	35	**31**
	STD	4	7	6	7	**3**
ISCXIDS2012	Mean	5	6	5	**4**	**4**
	STD	**2**	3	**2**	**2**	**2**
Botnet	Mean	5	6	6	4	**5**
	STD	2	3	**2**	**2**	**2**
CIC-IDS2018	Mean	45	53	56	45	**41**
	STD	10	9	10	9	**8**

The convergence behavior of the developed ROA-LF is shown in Fig. 2. The ROA-LF shows a faster convergence rate than the other methods on four out of five datasets, while the original ROA needs fewer iterations to reach the optimal solution on the CIC-IDS2018 dataset. This indicates that the use of LF can effectively improve the convergence ability of ROA and thus obtain better optimization results. These results prove the suitability of the developed ROA-LF as an FS for ID.

Boxplot is a visual representation of data distribution of the results in terms of accuracy in three quartiles: lower, middle, and upper. A boxplot of all the methods over five datasets is shown in Fig. 3. This Figure shows that the median accuracy of ROA-LF is higher than other MH methods in three out of five datasets, while upper accuracy is higher for four out of five datasets. This confirms the stability of the developed ROA-LF compared to the other comparison algorithms.

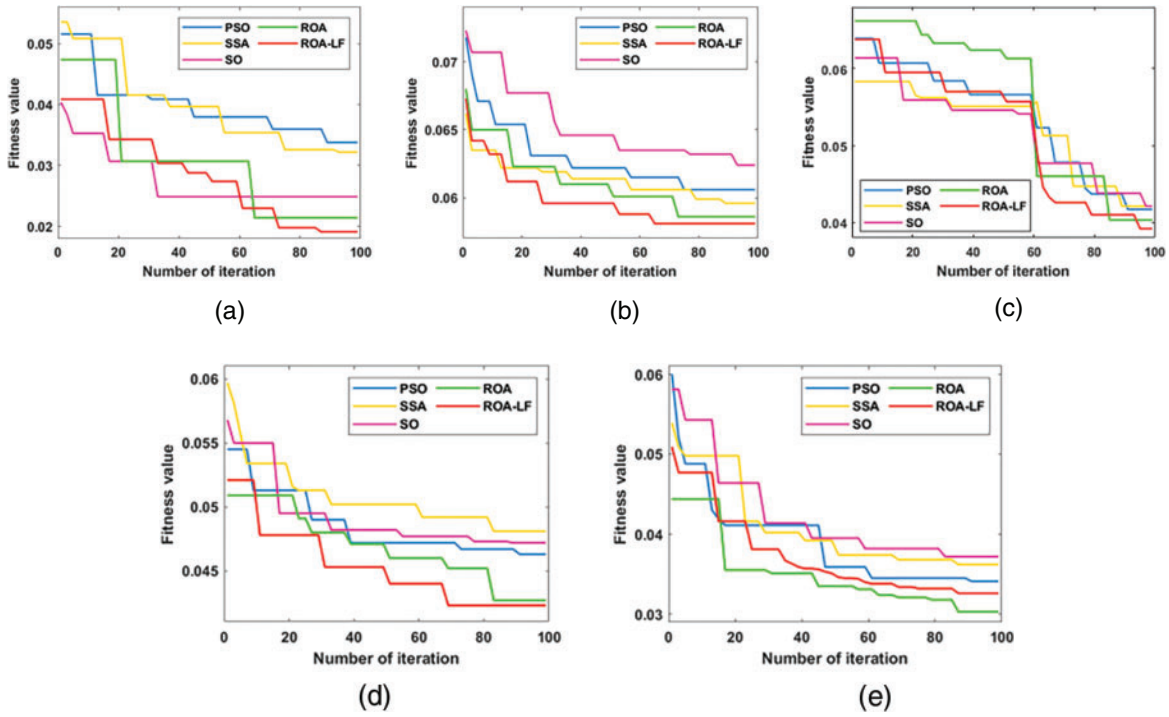

Figure 2: Convergence curves of the ROA-LF and the other MH algorithms for (a) KDD-CUP99, (b) NSL-KDD, (c) ISCXIDS2012, (d) Botnet, and (e) CIC-IDS2018

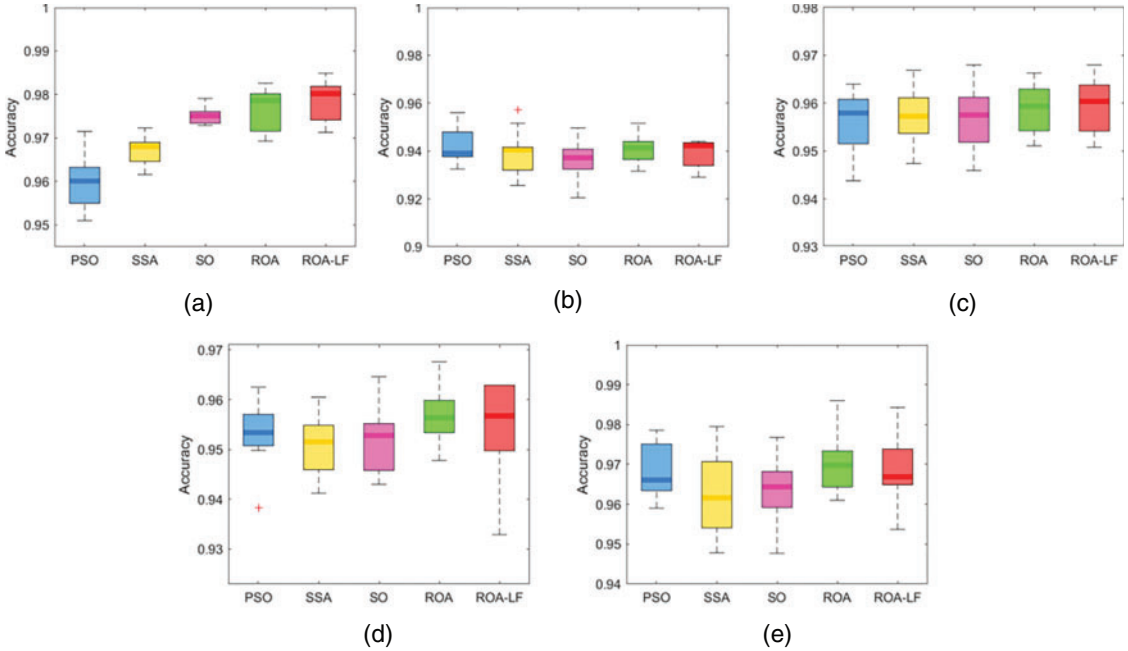

Figure 3: Box plots of the ROA-LF and the other MH algorithms for (a) KDD-CUP99, (b) NSL-KDD, (c) ISCXIDS2012, (d) Botnet, and (e) CIC-IDS2018

4.5 Real-World Engineering Problems

In this section, the ROA-LF method is applied to solve two real-world engineering problems with constraints, and these problems include Cantilever Beam Design [37], Three-Bar Truss Design [38], and Pressure Vessel Design [39].

4.5.1 Cantilever Beam Design (CBD) Problem

The proposed ROA-LF is applied to solve the CBD problem, which has five main parameters that need to be specified during the optimization process. Fig. 4 shows the CBD problem design. The mathematical representation of this problem can be formulated as follows:

Minimize

$$f(x) = 0.6224\,(x_1 + x_2 + x_3 + x_4 + x_5) \tag{17}$$

Subject to:

$$g(x) = \frac{60}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0 \tag{18}$$

where $(0.01 \leq x_i \leq 100, i = 1, 2, 3, 4, 5)$.

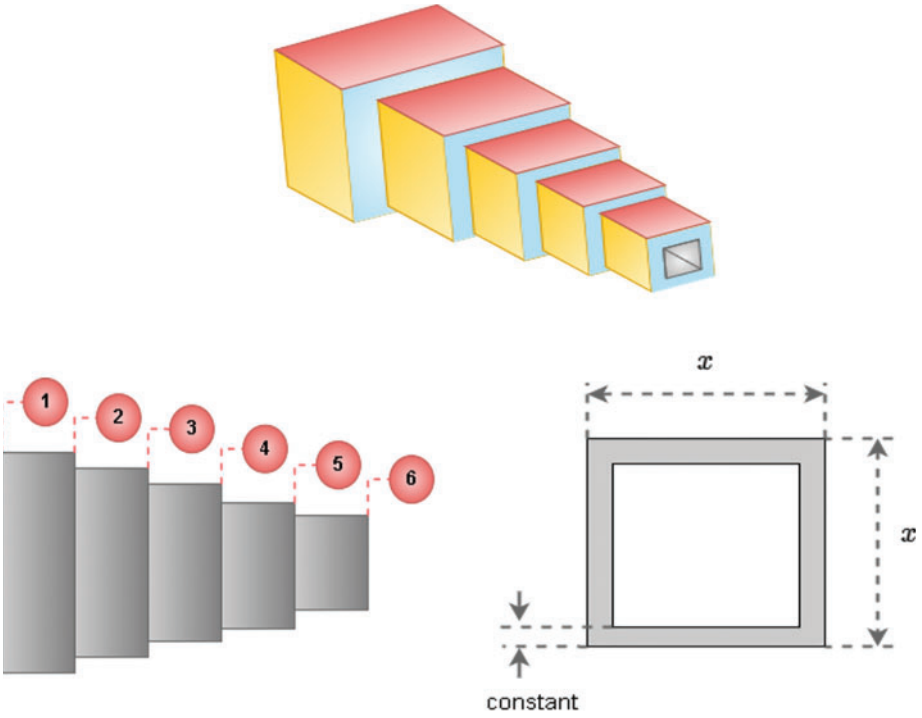

Figure 4: The CBD problem

Table 10 gives the results of the ROA-LF and the other methods for solving the problem of CBD. The ROA-LF has the smallest weight compared to PSO, SSA, SO, and ROA, while SO ranked second.

Table 10: Results of ROA-LF and other methods for CBD problem

Method	Best weight for variables					Optimal weight
	x_1	x_2	x_3	x_4	x_5	
PSO	5.5122	5.5653	4.6476	3.5543	2.0423	13.2706
SSA	6.3791	3.9871	8.6664	3.6680	1.7987	15.2484
SO	5.9832	4.7939	4.6247	3.4697	2.0584	13.0268
ROA	6.0231	5.4457	4.2770	3.5853	2.1767	13.3865
ROA-LF	5.5456	4.8966	4.4228	3.5007	2.1396	**12.7625**

4.5.2 Three-Bar Truss Design (TBTD) Problem

The optimal design of a TBTD seeks to minimize the structure weight subject to supporting a total load acting vertically downward. Two design variables and the structural geometry of the problem are given in Fig. 5. The objective function of this problem can be written as follows:

Minimize

$$f(x) = \left(2\sqrt{2x_1} + x_2\right) * l \tag{19}$$

Subject to:

$$g_1(x) = \frac{\sqrt{x_1}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0$$

$$g_2(x) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0 \tag{20}$$

$$g_3(x) = \frac{1}{\sqrt{2}x_2 + x_1}P - \sigma \leq 0$$

where $l = 100$ cm, $P = 2\,\frac{\text{kN}}{\text{cm}^2}$, $\sigma = 2\,\frac{\text{kN}}{\text{cm}^2}$, and $0 \leq x_i \leq 1, i = 1.2$.

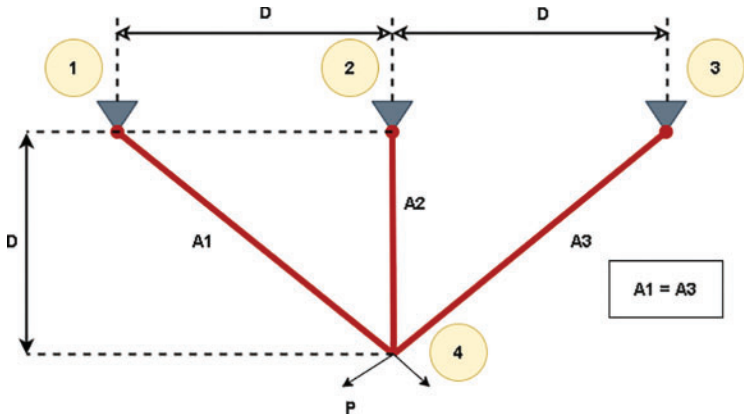

Figure 5: TBTD problem

The results of the ROA-LF for solving the problem of TBTD are provided in Table 11. The ROA-LF provides the best solution since it gained the smallest weight in comparison to PSO, SSA, SO, and ROA methods. This indicates the suitability of the developed ROA-LF for the TBTD engineering problem.

Table 11: Results of ROA-LF and other methods for TBTD problem

Method	Optimal weight for variables		Optimal weight
	A_1	A_2	
PSO	1.0000	1.2619	489.3098
SSA	1.3251	1.2166	447.2487
SO	1.1495	1.3596	439.2091
ROA	1.6482	1.1215	475.2698
ROA-LF	1.1566	1.2131	**425.2684**

4.5.3 Pressure Vessel Design (PVD) Problem

In this problem, the PVD seeks to minimize the total pressure constrained by material, shaping, and welding costs. This problem consists of four variables, as illustrated in Fig. 6, where T_s denotes the thickness of the shell, T_h is the head thickness, R represents the inner radius, and L is the length of the cylindrical section of the vessel. The objective function of the PVD can be written as follows:

Minimize

$$f(x) = 0.6224x_1x_2x_3 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \tag{21}$$

Subject to:

$$\begin{aligned} g_1(x) &= -x_1 + 0.0193x_3 \le 0 \\ g_2(x) &= -x_3 + 0.00954x_3 \le 0 \\ g_3(x) &= -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1,296,000 \le 0 \\ g_4(x) &= x_4 - 240 \le 0 \end{aligned} \tag{22}$$

where $(0 \le x_i \le 100, i = 1.2)$ and $(10 \le x_i \le 200, i = 3.4)$.

The results of the ROA-LF and the other comparative methods for the problem of PVD are given in Table 12. The ROA-LF has the smallest weight compared to PSO, SSA, SO, and the original ROA, while the SSA ranked second. The results reveal that ROA-LF can obtain excellent optimal values in this engineering problem, reflecting the applicability of ROA-LF to engineering problems.

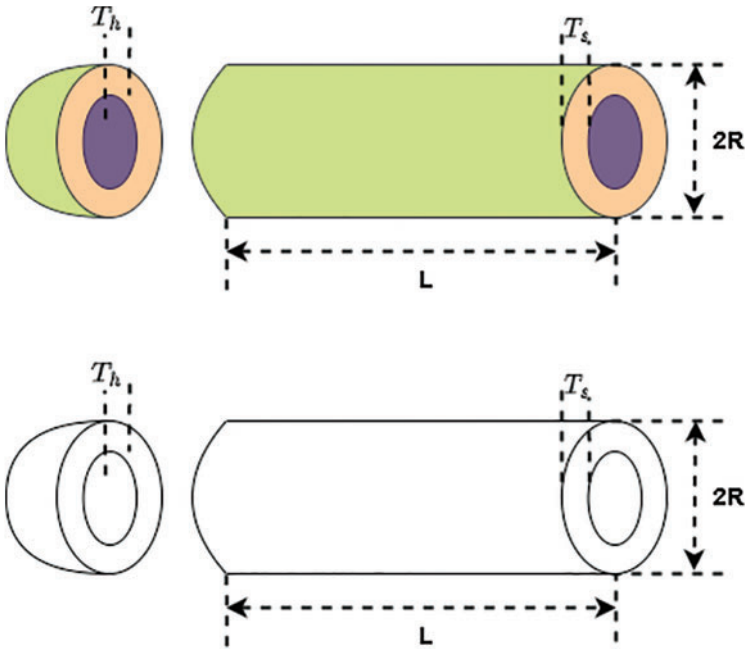

Figure 6: The PVD problem

Table 12: Results of ROA-LF and other methods results for PVD problem

Method	Optimal weight for variables				Optimal weight
	T_s	T_h	R	L	
PSO	1.0000	0.0000	120.0000	10.5012	2414.0478
SSA	1.5139	0.0000	63.1556	11.2165	2953.1495
SO	1.3265	0.0000	63.3515	11.4516	2275.4308
ROA	1.1368	0.0000	70.1692	10.3115	1841.2947
ROA-LF	1.1348	0.0000	63.3546	10.6153	**1661.9514**

5 Conclusion and Future Work

The existence of irrelevant or redundant data affects the performance of ML methods. This paper presents a novel FS method to improve the capability of the original ROA in exploration and exploitation using LF. The developed ROA-LF efficiency is validated using five open-access datasets in the ID domain: KDD-CUP99, NSL-KDD, ISCXIDS2012, Botnet, CIC-IDS2018, and three engineering problems. The developed ROA-LF performance is compared with the PSO, SSA, SO, and original ROA. The experimental results showed that the adaptive LF could improve ROA, thus improving its performance capability. The developed ROA-LF performs better than the other comparative methods in terms of fitness values, accuracy, number of the selected OFS, and convergence speed evaluation metrics. The statistical results show that ROA-LF is significantly more effective than the comparison algorithm.

Moreover, the results demonstrate that ROA-LF is applicable to the tested engineering optimization problems in real life with satisfactory optimization results compared to PSO, SSA, SO, and ROA alone. In future work, we will attempt to use developed ROA-LF as an FS method in other applications such as text mining, image segmentation, industry, and IoT. The introduced FS method can be improved by applying chaotic maps or combining it with other MH methods to speed up ROA's capability when searching for OFS and avoid getting stuck in the local optima. Moreover, the developed ROA-LF can be used for deep learning and ML model parameter tuning in medical applications such as Pancreatic Nodule Detection [40], and brain tumors [41].

Funding Statement: The author received no specific funding for this study.

Conflicts of Interest: The author declares that he has no conflicts of interest to report regarding the present study.

References

[1] K. K. R. Choo, K. Gai, L. Chiaraviglio and Q. Yang, "A multidisciplinary approach to internet of things (IoT) cybersecurity and risk management," *Computer Security*, vol. 102, pp. 102136, 2021.
[2] E. Jaw and X. Wang, "Feature selection and ensemble-based intrusion detection system: An efficient and comprehensive approach," *Symmetry*, vol. 13, no. 10, pp. 1764, 2021.
[3] S. Krishnaveni, S. Sivamohan, S. S. Sridhar and S. Prabakaran, "Efficient feature selection and classification through ensemble method for network intrusion detection on cloud computing," *Cluster Computing*, vol. 24, no. 3, pp. 1761–1779, 2021.
[4] V. R. Balasaraswathi, L. Mary Shamala, Y. Hamid, M. Pachhaiammal Alias Priya, M. Shobana *et al.,* "An efficient feature selection for intrusion detection system using B-HKNN and C2 search based learning model," *Neural Process Letter*, vol. 54, no. 1, pp. 1–25, 2022.
[5] V. Ford and A. Siraj, "Applications of machine learning in cyber security," in *Proc. of the. 27th Int. Conf. on Computer Applications in Industry and Engineering*, Kota Kinabalu, Malaysia, vol. 118, pp. 64–82, 2014.
[6] A. Gupta, R. Gupta and G. Kukreja, "Cyber security using machine learning: Techniques and business applications," *Applications of Artificial Intelligence in Business Education and Healthcare*, vol. 954, pp. 385–406, 2021.
[7] H. Varaee and M. R. Ghasemi, "Engineering optimization based on ideal gas molecular movement algorithm," *Engineering and Computers*, vol. 33, no. 1, pp. 71–93, 2017.
[8] S. S. Band, S. Ardabili, A. S. Danesh, Z. Mansor, I. AlShourbaji *et al.,* "Colonial competitive evolutionary rao algorithm for optimal engineering design," *Alexandria Engineering Journal*, vol. 61, no. 12, pp. 11537–11563, 2022.
[9] M. Banaie-Dezfouli, M. H. Nadimi-Shahraki and Z. Beheshti, "R-GWO: Representative-based grey wolf optimizer for solving engineering problems," *Applied Soft Computing*, vol. 106, pp. 107328, 2021.
[10] R. Khalid and N. Javaid, "A survey on hyperparameters optimization algorithms of forecasting models in smart grid," *Sustain Cities Soc.*, vol. 61, pp. 102275, 2020.
[11] L. Abualigah and A. Diabat, "A comprehensive survey of the grasshopper optimization algorithm: Results, variants, and applications," *Neural Computing and Applications*, vol. 32, pp. 15533–15556, 2020.
[12] S. Mirjalili, S. M. Mirjalili and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, pp. 495–513, 2016.
[13] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. of Int. Conf. on Neural Networks*, Perth, WA, Australia, vol. 1, no. 1, pp. 1942–1948, 1995.
[14] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris *et al.,* "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.
[15] J. H. Holland, "Genetic algorithms," *Scientific Am.*, vol. 267, pp. 66–73, 1992.

[16] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
[17] F. A. Hashim and A. G. Hussien, "Snake optimizer: A novel meta-heuristic optimization algorithm," *Knowledge-Based Systems*, vol. 242, pp. 108320, 2022.
[18] H. Jia, X. Peng, and C. Lang, "Remora optimization algorithm," *Expert Syst. Appl.*, vol. 185, pp. 46–61, 2021.
[19] K. H. Almotairi and L. Abualigah, "Hybrid reptile search algorithm and remora optimization algorithm for optimization tasks and data clustering," *Symmetry*, vol. 14, no. 3, pp. 458, 2022.
[20] Q. Liu, N. Li, H. Jia, Q. Qi and L. Abualigah, "Modified remora optimization algorithm for global optimization and multilevel thresholding image segmentation," *Mathematics*, vol. 10, no. 7, pp. 1014, 2022.
[21] R. Zheng, H. Jia, L. Abualigah, S. Wang and D. Wu, "An improved remora optimization algorithm with autonomous foraging mechanism for global optimization problems," *Mathematical Biosciences and Engineering*, vol. 19, no. 4, pp. 3994–4037, 2022.
[22] S. S. Kareem, R. R. Mostafa, F. A. Hashim and H. M. El-Bakry, "An effective feature selection model using hybrid metaheuristic algorithms for iot intrusion detection," *Sensors*, vol. 22, no. 4, pp. 1396, 2022.
[23] M. Ahsan, R. Gomes, M. M. Chowdhury and K. E. Nygard, "Enhancing machine learning prediction in cybersecurity using dynamic feature selector," *Journal of Cybersecurity and Privacy*, vol. 1, no. 1, pp. 199–218. 2021.
[24] Z. Manbari, F. AkhlaghianTab and C. Salavati, "Hybrid fast unsupervised feature selection for high-dimensional data," *Expert Systems with Applications*, vol. 124, pp. 97–118, 2019.
[25] M. Shehab and L. Abualigah, "Opposition-based learning multi-verse optimizer with disruption operator for optimization problems," *Soft Computing*, vol. 26, pp. 11669–11693, 2021.
[26] A. G. Hussien and M. Amin, "A self-adaptive Harris Hawks optimization algorithm with opposition-based learning and chaotic local search strategy for global optimization and feature selection," *International Journal of Machine Learning and Cybernetics*, vol. 13, no. 2, pp. 309–336, 2022.
[27] Y. Yuan, X. Mu, X. Shao, J. Ren, Y. Zhao *et al.,* "Optimization of an auto drum fashioned brake using the elite opposition-based learning and chaotic k-best gravitational search strategy based grey wolf optimizer algorithm," *Applied Soft Computing*, vol. 123, pp. 108947, 2022.
[28] J. P. Nolan, "Modeling with stable distributions," in *Univariate Stable Distributions*, Cham: Springer, pp. 25–52, 2020.
[29] Y. Li, Y. Zhao and J. Liu, "A levy flight sine cosine algorithm for global optimization problems," *International Journal of Distributed Systems and Technologies*, vol. 12, no. 1, pp. 49–66, 2021.
[30] I. Al-Shourbaji, P. H. Kachare, S. Alshathri, S. Duraibi, B. Elnaim *et al.,* "An efficient parallel reptile search algorithm and snake optimizer approach for feature selection," *Mathematics*, vol. 10, pp. 2351, 2022.
[31] I. Al-Shourbaji, N. Helian, Y. Sun, S. Alshathri and M. Abd Elaziz, "Boosting ant colony optimization with reptile search algorithm for churn prediction," *Mathematics*, vol. 10, pp. 1031, 2021.
[32] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. of IEEE Conf. on Symp. on Computational Intelligence for Security and Defense*, Ottawa, ON, Canada, pp. 1–6, 2009.
[33] S. Sapre, P. Ahmadi and K. Islam, "A robust comparison of the KDDCup99 and NSL-KDD IoT network intrusion detection datasets through various machine learning algorithms," *arXiv preprint*, pp. 1–8, 2019.
[34] A. Shiravi, H. Shiravi, M. Tavallaee and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computer Security*, vol. 31, no. 3, pp. 357–374, 2012.
[35] E. B. Beigi, H. H. Jazi, N. Stakhanova and A. A. Ghorbani, "Towards effective feature selection in machine learning-based botnet detection approaches," in *Proc. of IEEE Conf. on Communications and Network Security*, San Francisco, CA, USA, pp. 247–255, 2014.
[36] I. Sharafaldin, A. H. Lashkari and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. of the 4th Int. Conf. on Information Systems Security and Privacy*, Funchal, Portugal, vol. 1, no. 2, pp. 108–116, 2018.

[37] H. Chickermane and H. C. Gea, "Structural optimization using a new local approximation method," *Int. J. Numerical Methods in Engineering*, vol. 39, no. 5, pp. 829–846, 1996.
[38] T. Ray and K. M. Liew, "Society and civilization: An optimization algorithm based on the simulation of social behavior," *IEEE Trans. Evolutionary Computation*, vol. 7, no. 4, pp. 386–396, 2003.
[39] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowledge Based Systems*, vol. 89, pp. 228–249, 2015.
[40] T. Thanya and S. W. Franklin, "Grey wolf optimizer based deep learning for pancreatic nodule detection," *Intelligent Automation and Soft Computing*, vol. 36, no. 1, pp. 97–112, 2023.
[41] S. Keerthi and P. Santhi, "Precise multi-class classification of brain tumor via optimization based relevance vector machine," *Intelligent Automation and Soft Computing*, vol. 36, no. 1, pp. 1173–1188, 2023.