Intelligent Systems with Applications 16 (2022) 200114

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications

An improved binary manta ray foraging optimization algorithm based feature selection and random forest classifier for network intrusion detection

Ibrahim Hayatu Hassan [a], Mohammed Abdullahi [a,*], Mansur Masama Aliyu [b], Sahabi Ali Yusuf [a], Abdulrazaq Abdulrahim [a]

[a] Department of Computer Science, Ahmadu Bello University, Zaria, Nigeria
[b] E-Library, Kebbi State University of Science and Technology, Aliero, Nigeria

ARTICLE INFO

Keywords:
Manta ray foraging optimization algorithm
Intrusion detection
Anomaly detection
Feature selection
Classification
Random forest

ABSTRACT

The growth within the Internet and communications areas have led to a massive surge in the dimension of network and data. Consequently, several new threats are being created and have posed difficulties for security networks to correctly discover intrusions. Intrusion Detection System (IDs) is one amongst the foremost essential events for security arrangements in network environments, and it is commonly applied to spot, track, and detect malevolent threats. Detecting intruders using metaheuristics and machine learning methodologies in recent trend offers improved discovery rate. Therefore, this paper presented an intrusion detection model using an improved Binary Manta Ray Foraging (BMRF) Optimization Algorithm based on adaptive S-shape function and Random Forest (RF) classifier. The BMFR is envisioned to identify the most relevant features and remove redundant and irrelevant ones from the intrusion detection datasets. Furthermore, the RF is used for feature evaluation and to build the intrusion detection model. The proposed method was validated and compared with other methods using two IDs benchmark datasets, which include NSL-KDD and CIC-IDS2017 datasets. The result indicates that the presented model selected 38 features with 99.6% precision, 94.3% recall, 96.9% f-measure, and 99.3% accuracy for the CIC-IDS2017 dataset. Moreover, for the NSL-KDD dataset, the presented model selected 22 features with 96.8%, 96.2%, 96.5%, and 98.8% for precision, recall, F-measure, and accuracy. In addition, a statistical significance test reveals a significance difference between the presented model and the compared methods in terms of F-measure.

1. Introduction

The communication model over networks is important in carrying sensational information for a variety of uses; intruders are drawn to the network to steal the information or impede the system. The attacker can dramatically reduce the value or entirely disable affected system accessibility. The key objective of the attack is to render the victim unable to use the resources. In most cases, targets may be web servers, CPUs, storage, or network resources. Because attacks to computer systems and networks are becoming more pervasive in today's trending growth with online sales and the dominance of e-market and e-commerce, network security is critical for securing the communication models (Anitha & Kaarthick, 2019; Froehlich & Kent, 1998; Onah, Abdulhamid, Abdullahi, Hassan, & Al-Ghusham, 2021; SaiSindhuTheja & Shyam, 2021). Fig. 1 depicts the number of data breaches and individual affected between 2015 and 2021 (Podcast, 2021).

An Intrusion Detection System (IDs) is a protocol for network security that detects, prevents, and repels unapproved access to a communication or computer network. IDs play a critical role in keeping a network safe and secure. Aside from that, the primary goal of an IDs is to ensure the network system's accessibility, authenticity, and privacy. Intrusion detection refers to a method that try to categorize network traffic into different types of attacks. It entails binary or multi-classification of preferred audit data as well as other system aspects. The success of an IDs is strongly determined on its ability to maximize detection accuracy while minimizing false alarm rate and detection

* Corresponding author.
E-mail addresses: ihhassan@abu.edu.ng (I.H. Hassan), abdullahilwafu@abu.edu.ng (M. Abdullahi), mmasama@acm.org (M.M. Aliyu), sahabiali@yahoo.com (S.A. Yusuf), aabdulrahim@abu.edu.ng (A. Abdulrahim).

https://doi.org/10.1016/j.iswa.2022.200114
Received 21 November 2021; Received in revised form 8 June 2022; Accepted 13 August 2022
Available online 18 August 2022

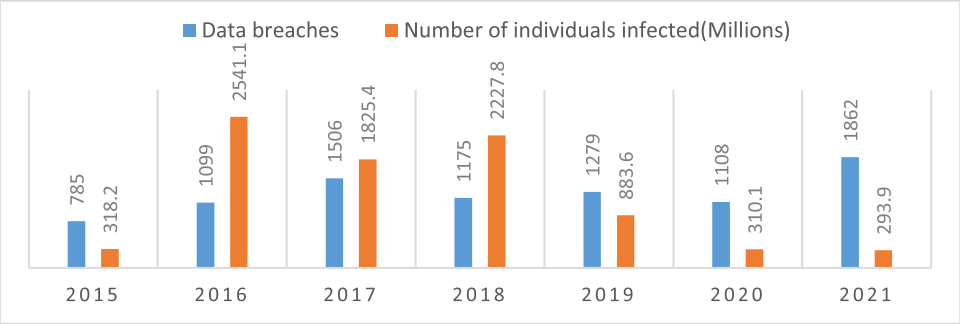

Fig. 1. Amount of computer and network data breaches and number of individuals infected.

time. Intrusion detection systems have remained a hot research topic in the cyber security due to their success in detecting new and unfamiliar attacks in practice. (Aburomman & Ibne, 2016; Gan, Duanmu, Wang, & Cong, 2013; Hasan, Islam, Zarif, & Hashem, 2019).

Normally, IDs models are categorized into two types: misuse and anomaly intrusion detection systems (Zakeri & Hokmabadi, 2019). Instructions are recognized on the basis of restrictions of system faults and identified as attack signatures for a misuse type of IDs. Still, it does not able to identify attacks which are new or unknown. In contrast, anomaly IDSs are dependant on usual patterns and exploits them to identify some action that significantly separates from usual patterns. By comparing existing intrusion patterns into consideration for examination, misuse intrusion detection technique identifies intrusions with previously identified patterns, otherwise, anomaly intrusion detection technique recognizes patterns based on the analysis of data acquired from normal practice (Jin et al., 2020). It can be important that all unusual patterns are considered as a possible attack even if they are not recognized as attacks; consequently, chances of getting false positives results may be unexpected in anomaly-based detection.

Many researchers view intrusion detection as a difficult problem in terms of classification and feature selection (wivedi, Vardhan, & Tripathi, 2020). To improve the performance of IDSs, several feature selection methods have been presented and implemented. In recent decades, a wide range of filter and wrapper approaches have been developed to create intelligent intrusion detection systems capable of improving network security and detecting modern attacks (Mohammadi, Mirvaziri, Ghazizadeh-Ahsaee, & Karimipour, 2019). Filter methods determine the inherent relationships between the input features and the significant class, and then remove irrelevant features from the original features. This category of feature selection can only remove irrelevant features, therefore, it cannot handle redundant features. Some of the filter techniques employed in the area of intrusion detection include Chi-square (Thaseen & Kumar, 2017), Information gain (Abdullah, Balamash, Alshannaq, & Almabdy, 2018; Salo, Nassif, & Essex, 2019). Moreover, numerous metaheuristics optimization approaches, such as particle Swarm Optimization (PSO) (Syarif, 2016), Firefly (FF) Optimization Algorithm (Al-Yaseen, 2019), Grasshopper Optimization Algorithm (GOA), Grey Wolf Optimization Algorithm (GWO) (Alzubi, Anbar, Alqattan, Al-betar, & Abdullah, 2019), Differential Evolution (DE) (Balasaraswathi, Sugumaran, & Hamid, 2017), and Genetic Algorithm (GA) (Aslahi-Shahri et al., 2016), have been implemented as wrapper techniques for feature selection in intrusion detection, requiring a classification algorithm to select the most relevant features, and hence eliminating redundant and irrelevant features (Mafarja et al., 2019). Anomalies are often not easy to detect, and most anomaly detection methods find it difficult to find important patterns. In this perspective, several authors have explored various machine learning engines and classifiers for training and validation of network traffic data for IDs development (Tidjon, Frappier, & Mammar, 2019).

Several researchers have applied optimization techniques and machine learning methods for IDs. However, according to the No Free Lunch theorem optimization by Ebrahimpour and Eftekhari (2017), there cannot be a single algorithm that solves all optimization problems. As a result, the current algorithms for IDs are unable to solve all problems. This motivated us to develop a new IDs method based on a recently proposed meta-heuristic algorithm, Manta ray foraging (MRF) optimization and Random Forest classifiers. The following are the paper's main contributions:

1 Proposed a Binary Manta Ray Foraging optimization algorithm using adaptive S-shaped transfer functions for feature selection.
2 Execute feature selection using the proposed Binary Manta ray foraging optimization algorithm using NSL-KDD and CICIDS2017 network traffic datasets.
3 Developed network intrusion detection model with the selected features based on Random Forest classifier.
4 Examine performance analysis and comparison of the presented IDs with GA, PSO, GWO, and GOA using Accuracy, Recall, Precision, F-measure, and execution time.
5 Statistical justification of the results achieved compared to that of GA, PSO, GWO, and GOA using *t*-test.

The remaining part of the paper is organized as follows: Section 2 contains a comprehensive literature analysis, Section 3 outlines the research background, Section 4 contains the proposed models for network intrusion detection, Section 5 contains the experimental result and discussion, and Section 6 gives the conclusion remarks.

2. Literature review

A number of research works have been presented for intrusion detection using metaheuristics optimization algorithms for feature selection and machine learning algorithms for feature evaluation and intrusion detection. Few of the works were presented and discussed in this part.

Bayu and Kyung-Hyune (2018) presented an intrusion detection model combining a random forest (RF) and a particle swarm optimization (PSO). The PSO is used as a search strategy in the wrapper feature selection technique to select the most informative feature subset, and the RF is used as the learning algorithm in the work presented. The presented model's efficiency is assessed using the NSL-KDD dataset in terms of precision, recall, false alarm rate, and accuracy. The statistical significance test revealed significant differences between the presented model and other classifiers (rotation forest (RoF) and deep neural network (DNN)). The test shows that the presented model outperforms the compared algorithm. In Mehrnaz, Babak, and Iraj (2018), the authors proposed a new anomaly detection model based on the artificial bee colony (ABC) and the AdaBoost algorithm. Using the NSL-KDD and ISCXIDS2012 intrusion detection datasets, the ABC algorithm is used to select the most appropriate features for anomaly detection, and AdaBoost is used for feature evaluation and anomaly classification. In terms of accuracy and detection rate, the results show that the proposed IDs

outperform multi-objective PSO (MPSO) and Genetic algorithm (GA). A new IDs for extracting the most relevant features for intrusion detection based on the oppositional-based laplacian grey wolf optimization algorithm (OLGWO) was proposed in Anitha and Kaarthick (2019). The selected features are used to train a support vector machine (SVM) classifier, which is used to categorize the various attacks. The KDD99 dataset is used to evaluate the proposed system's performance, and the results are compared to existing methods in terms of detection time, detection rate, false positive, and false negative.

Another study by Omar (2020), proposed a wrapper feature selection model for a network intrusion detection system based on grey wolf optimization (GWO), PSO, GA, and the firefly optimization algorithm (FFA). A filtering-based method for the mentioned algorithms' mutual information. With the UNSW-NB15 dataset, the feature subsets are evaluated using SVM and J48 classifiers. For the experiments, thirteen (13) sets of rules were created. The experiment results show that Rules 12 and 13 produce better results in terms of accuracy, F-measure, and sensitivity, with 30 and 13 reduced features, respectively. The GA achieves good results in terms of TPR and FNR. While rules 11, 9, and 8 produce the best results in terms of FPR, PSO produces the best results in terms of precision and TNR. In another work, Mukaram, Mohammed, and Laith (2020) proposed the GWOSVM-IDS intrusion detection model for wireless sensor networks, which combines modified binary grey wolf optimization with a support vector machine (WSN). The proposed method is intended to improve intrusion detection accuracy, detection rate, and processing time in the WSN environment. The KDD 99 cub dataset is used to demonstrate the effectiveness of the proposed GWOSVM-IDS. According to the experimental results, the GWOSVM-IDS has the best performance in terms of feature generation, accuracy, false alarm rate, detection rate, and execution time.

In another study, Yuyang, Guang, Shanqing, and Mian (2020) proposed a framework for intrusion detection based on feature selection and ensemble machine learning techniques. Initially, a combination of correlation-based feature selection (CFS) and the Bat algorithm (BA) was proposed for feature selection. The BA selects a subset of features, and the CFS examines the features for correlation. The intrusion detection model was then built using an ensemble machine learning technique that included RF, Forest by penalizing attributes (Forest PA), and C4.5 classifiers. The proposed model's performance was evaluated using the benchmark intrusion detection datasets AWID, NSL-KDD, and CIC-IDS 2017. Using Accuracy, Detection rate, Precision, F-measure, Attack detection rate, and False alarm rate, the proposed method outperformed feature selection methods such as GA, PSO, Modified Bat algorithm for feature selection (MBAFS), Information gain (IG), and Gain ratio (GR). In Shubhra, Manu, and Sarsij (2021), the authors proposed an intrusion detection model based on SVM and GOA optimization algorithm. The SVM classifier is used by the GOA as a fitness function to obtain notable features and to optimize the SVM parameters. The EFSGOA method's performance is evaluated using the KDD cup 99 and NSL-KDD datasets. The experiment analysis shows that the proposed EFSGOA outperformed the Genetic algorithm (GA) and Particle swarm optimization in terms of accuracy, detection rate, false alarm rate, and CPU time (PSO).

Onah et al. (2021) demonstrated an intrusion detection system in a fog computing environment that employs a Genetic Algorithm and a Naive Bayes classifier. Using the NSL-KDD dataset, the performance of the presented IDs was measured and compared to that of other classifiers in terms of accuracy, precision, f-measure, and execution time. According to the results, the presented IDs have a lower F-measure than SVM, Random Forest, and Decision Tree. However, in terms of execution time, the presented method outperforms the other classifiers, implying that the presented IDs will be very useful where speed of processing is required. Talita, Nataza, and Rustam (2021) proposed an intrusion detection system based on PSO for feature selection and a Naive Bayes classifier for feature subset evaluation and detection model development using KDD Cup 99 dataset. The best experimental result was obtained with 38 features and an accuracy of 99.12%. The proposed method, however, was not compared to any other method.

Al-Saqqa, Al-Fayoumi, and Qasaimeh (2021) presented an evolutionary search algorithm-based intrusion detection model for malicious traffic. To detect attacks, the evolutionary search algorithm is used to select the most relevant feature subsets. Two classifiers (Naive Bayes and J48) are used to compare the system's performance before and after feature selection. The NSL-KDD dataset is used for the experimental evaluation. The results show that using the evolutionary search algorithm to select features improves the intrusion detection system in terms of detection accuracy and detection of unknown attacks. Furthermore, time performance is achieved by shortening training time, which has a positive impact on overall system performance. Pankaj, Mahesh, Emmanuel, and Prajjval (2021) developed an improved anomaly-based intrusion detection model for the Internet of Things network (IoT). The proposed model selects the most appropriate IoT network features by combining GWO and PSO. Based on the features chosen, an RF detection model was created. The proposed model's performance was evaluated using the datasets KDDcup99, NSL-KDD, and CICDS-2017. For multiclass classification, the GWO+PSO+RF intrusion detection model achieved a mean accuracy of 99.66%.

Shukla et al. (2021) proposed an anomaly detection model based on the self-adaptive grasshopper optimization algorithm (GOA) and SVM. Extensive experiments are carried out on standard intrusion detection datasets such as NSL-KDD, AWID, and CIC-IDS 2017 to evaluate the proposed method's performance. In terms of detection rate, false-positive rate, and accuracy for solving IDs problems, the comparative simulation results show that the proposed algorithm outperforms the basic grasshopper optimization algorithm and other commonly used evolutionary techniques. SaiSindhuTheja and Shyam (2021) proposed an effective DoS attack discovery model developed using Oppositional Crow Search Algorithm (OCSA). The proposed OCSA is a combination of oposition base learning (OBL) and Crow search algorithm. The OCSA is used for feature selection and recurrent neural network (RNN) for classification. The proposed system was evaluated using the KDD cup99 dataset based on accuracy, f-measure, recall and precision. A comparative analysis shows the superiority of this method in comparsion with GA, classical CSA, and FA.

Rashmita et al. (2021) design an intrusion detection system in an IoT network using Glowworm Swarm Optimization (GSO) algorithm with Principle Component Analysis (PCA). The PCA is used for feature extraction, and the GSO to classify the intrusions into different classes. The proposed method was evaluated and compared with Artificial Neural Network (ANN), SVM, Back Propagation Neural Network (BPNN), and PSO using NSL-KDD dataset. The result shows the superiority of the proposed method in contrast to compared methods with respect to detection rate (DR), accuracy, precision, recall, and False alarm rate (FAR). Hussein and Ku (2021) presented an intrusion detection model using an enhanced binary grey wolf optimization (EBGWO) algorithm for extracting relevant features based on NSL-KDD dataset. The proposed method achieved better results compared to Bat algorithm (BA), Binary particle swarm optyimization algorithm (BPSO), Modified binary grey wolf optimization algorithm (MBGWO), Binary grey wolf optimization algorith (bGWO), Modified grey wolf optimization algorith (MGWO), and grey wolf optimization algorith (GWO) with respect to accuracy and number of features selected.

Based on the review conducted, it can be seen that various metaheuristics optimization and machine learning algorithms were used for network intrusion detection. However, most of these algorithms have some drawbacks such as trapping into local optima and also on the basis of No Free Lunch theorem optimization by Ebrahimpour and Eftekhari (2017), there cannot be a single algorithm that solves all optimization problems. Therefore, this study proposed an Improved Binary Manta Ray Foraging based attribute selection and random forest classifier for network intrusion detection. Fig. 1 depicts the summary of some of the reviewed related works.

Table 1
Summary of Related Works.

References	Feature Selection Method	Classifier Used	Classification Type	Metrics Used	Dataset Used	Compared Algorithms	Limitations	Advantage
Bayu & yung-Hyune (2018)	PSO	RF	Binary	Accuracy, precision, FAR, DR	NSL-KDD	RoF, DNN	The performance of the study was evaluated on a single dataset, which this may not give a fair conclussion	Achieved better result than the compraed methods
Mehrnaz et al. (2018)	ABC	AdaBoost	Multi-class	Accuracy, detection rate	NSL-KDD, ISCXIDS2012	MPSO, GA	The two dataset used was not current and have smaller number of attacks.	Obtain a moderate results
Anitha & Kaarthick (2019)	OLGWO	SVM	Multi-class	DR, FPR, FNR	KDD99	CSAVC, CSOACN, SVM	The dataset used has redundant connections records which can result in machine learning bias and also the detection rate is not really good.	Achieved better result than the compraed methods
Shubhra et al. (2021)	EFGOA	SVM	Multi-class	Accuracy, DR, FAR, CPU time	KDD cup 99, NSL-KDD	GA, PSO	The two dataset used was not current and have smaller number of attacks.	Obtain a moderate results
Omar (2020)	PSO, GA, GWO, FFA	SVM, J48	Multi-class	TPR, TNR, FPR, FNR	UNSW-NB15	PSO, GA, GWO, FFA	The datset does not exhibit a good characteristics of real-time network traffic, and has less number of real data	Obtained higher accuracy than the compared approaches
Yuyang et al. (2020)	BA	RF, forest PA, C4.5	Multi-class	Accuracy, DR, ADR, F-measure, precision, FAR	AWID, NSL-KDD, CICDS-2017	IG, GR, GA, PSO, MBAFS	The convergence speed of the proposed method is slow and there exist a problen stagnation	This study axhieved higher detection rate and accuracy compared to other methods on NSL-KDD, AWID, and CICDS-2017datasets
Onah et al. (2021)	GA	Naïve Bayes	Multi-class	F-score, execution time, DR, TPR, FPR, ROC	NSL-KDD	SVM, RF, Decision tree	This method obtained a comparative accuracy, however, the f-measure is not very good, which is not good for multi-classification problem	Obtain a good accuracy then the compared methods
Talita et al. (2021)	PSO	Naïve Bayes	Binary	Accuracy, running time, Number of features	KDD cup 99	Nill	The limitation of this study id that, the dataset used has redundant connections records which can result in machine learning bias. Additionally no comparison was conducted.	The performance of this approach was compared with any other method, so we could not ascertain its advantage over other methods
Mukaram et al. (2020)	Improved Binary GWO	SVM	Binary	Number of features accuracy, FAR, DR execution time	NSL-KDD	PSO, GWO	The performance of the study was evaluated on a single dataset, which this may not give a fair conclussion	This study axhieved higher detection rate and accuracy compared to other methods on NSL-KDD
Al-Saqqa et al. (2021)	GA	Naïve Bayes, J48	Multi-class	Accuracy, DR, Execution time	NSL-KDD	Nill	The performance of the study was evaluated on a single, which this may not give a fair conclussion	The performance of this approach was compared with any other method, so we could not ascertain its advantage over other methods
Pankaj et al. (2021)	GWO, PSO	RF	Multi-class	Accuracy	KDDcup99, NSL-KDD, CICDS-2017	Nill	Accuracy was used as the evaluation metric for evaluation. However, it has been stated in the literature that accuracy is not a good measure in multi-classification problem.	The performance of this approach was compared with any other method, so we could not ascertain its advantage over other methods
Shukla (2021)	Adaptive GOA	SVM	Multi-class	Accuracy, DR, FPR	NSL-KDD, AWID, CICDS-2017	Classical GOA	Slow convergence	Achived better deterction rate than the classical GOA
(SaiSindhuTheja & Shyam, 2021)	OCSA	RNN	Binary	Accuracy, f-measure, recall, precision	KDDcup99	GA, CSA, FA	The limitation of this study id that, the dataset used has redundant connections records which can result in machine learning bias	Obtained higher accuracy than the compared approaches
Rashmita et al. (2021)	PCA	GSO	Multi-class	Accurac, precision, recall, DR, FAR	NSL-KDD	ANN, BPNN, PSO, SVM	The performance of the study was evaluated on a single dataset, which this may not give a fair conclussion	This study axhieved higher detection rate and accuracy compared to other methods on NSL-KDD datasets

(*continued on next page*)

Table 1 (*continued*)

References	Feature Selection Method	Classifier Used	Classification Type	Metrics Used	Dataset Used	Compared Algorithms	Limitations	Advantage
Hussein and Ku (2021)	EBGWO	SVM	Multi-class	No. features, accuracy	NSL-KDD	BAT, PSO, MBGWO, bGWO, MGWO, GWO	The result shows an accuracy of 87.46%, which this indicate a room for improvement and more consideration of better multi-classificationmetrics.	Shows a moderate accuracy result in contrast to other compared methos

3. Background of the research

3.1. Manta ray foraging (MRF) optimization algorithm

MRF optimization algorithm was initially presented in by Zhao, Zhang, and Wang (2020), and it was inspired by Manta rays' food searching behaviour. The manta ray is one of the most well-known sea organisms. It feeds on plankton, which is composed of tiny animals found in the water. Manta rays have three food-finding mechanisms: chain, cyclone, and somersault foraging.

3.1.1. Chain foraging

Manta rays line up in an organized fashion to make a hard chain for grasping prey planktons in chain foraging. The MRF optimization algorithm assumes that the best location is one with a maximum concentration of plankton, which is the target prey for the manta ray chain to consume. Except for the first manta ray, the MRF algorithm updates the manta ray's location based on best location and the manta ray in front of it. Eq. (1) shows the chain foraging update mechanism.

$$q_n^{iter+1} = \begin{cases} q_n^{iter} + rand_1\left(q_b^{iter} - q_n^{iter}\right) + \gamma\left(q_b^{iter} - q_n^{iter}\right) & n = 1 \\ q_n^{iter} + rand_2\left(q_{n-1}^{iter} - q_n^{iter}\right) + \gamma\left(q_b^{iter} - q_n^{iter}\right) & n = 2, \ldots, N \end{cases} \tag{1}$$

$$\gamma = 2 * rand_3 * \sqrt{|\log(rand_4)|} \tag{2}$$

where q_n^{iter} is the location of the n^{th} manta ray at iteration *iter*, $rand_{1,\ 2,\ 3,4}$ are randomly generated numbers in the interval [0, 1] that are dissimilar from each other, γ define the weight coefficient, and q_b^{iter} define the plankton with the largest concentration. In chain foraging, the position update is determined by the best plankton location and the prior manta ray in the sequence.

3.1.2. Cyclone foraging

In cyclone foraging, the manta ray population forms a spiral by creating a head-to-tail link, when they discover plankton with the biggest concentration. With this knowledge, each manta ray moves not only in the direction of the plankton, but also in the direction of the manta ray in front of it. . Eq. (3) represent cyclone foraging update mechanism mathematically.

$$q_n^{iter+1} = \begin{cases} q_b^{iter} + rand_5\left(q_b^{iter} - q_n^{iter}\right) + \propto\left(q_b^{iter} - q_n^{iter}\right) & n = 1 \\ q_b^{iter} + rand_6\left(q_{n-1}^{iter} - q_n^{iter}\right) + \propto\left(q_b^{iter} - q_n^{iter}\right) & n = 2, \ldots, N \end{cases} \tag{3}$$

$$\propto = 2 * e^{rand_8} * \left(\frac{MaxIter - iter + 1}{iter}\right) * \sin(2 * \pi * rand_8) \tag{4}$$

where $\propto$ represent the weight coefficient, *MaxIter* defines the maximum number of iterations, and $rand_{5,6}$ represent the dissimilar generated numbers in the interval [0, 1]. This stage of the MRF algorithm serves as its main driving mechanism of exploration and exploitation. Using the best plankton as a reference point in this stage allows for searching the productive areas around the existing best solution, which offers the exploitation abilities of the algorithm. It also provides important support for the exploration capability by requiring the manta rays to move to a random position in the search regions that should be not only far away from their current location, but also the best prey location. This allows for a more thorough exploration of the global search region and aids the MRF algorithm in guiding the Manta rays through previously unexplored areas of the search area. This proposed mechanism is depicted in Eqs. (5) and (6).

$$q_{rp}^{iter} = Lw + rand_9(UP - Lw) \tag{5}$$

$$q_n^{iter+1} = \begin{cases} q_{rp}^{iter} + rand_{10}\left(q_{rp}^{iter} - q_n^{iter}\right) + \propto\left(q_{rp}^{iter} - q_n^{iter}\right) & n = 1 \\ q_{rp}^{iter} + rand_{11}\left(q_{n-1}^{iter} - q_n^{iter}\right) + \propto\left(q_{rp}^{iter} - q_n^{iter}\right) & n = 2, \ldots, N \end{cases} \tag{6}$$

where q_{rp}^{iter} represent the created random position in the acceptable limits, *UP* and *Lw* are the upper and lower limits respectively in a particular location, and $rand_{5,\ 6,\ 7,8,\ 9,\ 10,11}$, represent dissimilar randomly generated numbers in the interval [0, 1].

3.1.3. Somersault foraging

Somersault foraging is a regular, local, random, and cyclic movement which supports manta rays consume more plankton. The position of the highest concentration of plankton so far is selected to be the reference point, and each manta ray moves around this point and tumbles to a new location (Turgut, 2020). Eq. (7) is a mathematical representation of this model.

$$q_n^{iter+1} = q_n^{iter} + smsf * \left(rand_{12} * q_b^{iter} - \left(rand_{13} * q_n^{iter}\right)\right. \tag{7}$$

where *smsf* represent the somersault factor and $rand_{12,13}$, represent dissimilar randomly generated numbers in the interval [0, 1]. In this paper, the *smsf* is given a value of 2 as described in (Turgut, 2020).

Algorithm 1. MRF Optimization Algorithm

while *criteria !satisfied* ***do***
for $n = 1, \ldots, N$ ***do***
if $rand_1 < 0.5$ ***then***
if $(iter/MaxIter) < rand_2$ ***then*** **//Cyclone foraging**
$q_{rp}^{iter} = LW + rand_9(UP - LW)$

$$q_n^{iter+1} = \begin{cases} q_{rp}^{iter} + rand_{10}(q_{rp}^{iter} - q_n^{iter}) + \propto(q_{rp}^{iter} - q_n^{iter}) & n = 1 \\ q_{rp}^{iter} + rand_{11}(q_{n-1}^{iter} - q_n^{iter}) + \propto(q_{rp}^{iter} - q_n^{iter}) & n = 2, \ldots, N \end{cases}$$

else

$$q_n^{iter+1} = \begin{cases} q_b^{iter} + rand_5(q_b^{iter} - q_n^{iter}) + \propto(q_b^{iter} - q_n^{iter}) & n = 1 \\ q_b^{iter} + rand_6(q_{n-1}^{iter} - q_n^{iter}) + \propto(q_b^{iter} - q_n^{iter}) & n = 2, \ldots, N \end{cases}$$

endif
else **// Chain foraging**

$$q_n^{iter+1} = \begin{cases} q_n^{iter} + rand_1(q_b^{iter} - q_n^{iter}) + \gamma(q_b^{iter} - q_n^{iter}) & n = 1 \\ q_n^{iter} + rand_2(q_{n-1m}^{iter} - q_n^{iter}) + \gamma(q_b^{iter} - q_n^{iter}) & n = 2, \ldots, N \end{cases}$$

endif
// Calculate fitness for n^{th} **manta ray** $f(q_n^{iter+1})$
if $f(q_n^{iter+1}) < f(q_b^{iter})$ ***then***
$q_b^{iter} = q_n^{iter+1}$
endif
endfor
// Somersault foraging
for $n = 1, \ldots, N$ ***do***

(*continued on next page*)

(*continued*)

$q_n^{iter+1} = q_n^{iter} + smf * (rand_{12} * q_b^{iter} - (rand_{13} * q_n^{iter})$
// Calculate fitness for n^{th} manta ray $f(q_n^{iter+1})$
if $f(q_n^{iter+1}) < f(q_b^{iter})$ ***then***
$q_b^{iter} = q_n^{iter+1}$
endif
endfor
endwhile

3.2. Proposed binary mrf (BMRF) optimization algorithm

Here, the proposed binary MRF optimization algorithm using adaptive S-shaped transfer function was presented. The description of the proposed BMRF is given below.

The original MRF algorithm was proposed by Zhao et al. (2020) to handle continuous optimization issues. But, in an attribute selection space, attributes are binary in nature, therefore two conditions occur: selection and non-selection, this motivate us to suggest the BMRF algorithm, as binary can best denote the attribute space (Zarshenas & Suzuki, 2016). Here, the MRF algorithm is transformed in to the BMRF algorithm by use of an adaptive S-shape transfer function (ASSTF), which makes it appropriate for feature selection spaces (Cao, Guo, & Yongquan, 2017; Islam, Li, & Mei, 2017; Yi, Wang, & Wang, 2016). Each manta ray M ($m_1, m_2, \ldots, m_j, \ldots, m_n$) in the BMRF algorithm consists of a chain of binary values, which donate a candidate solution, n is the dimension, and m_j is either 1 or 0 indicating whether or not the attribute/feature is chosen. The BMRF algorithm updates the position by changing the probability, as opposed to the MRF, which updates the positions by using continues stages. The S-shaped function is the most basic and appropriate technique for transitioning from a continuous to a binary state (Mirjalili & Lewis, 2013). Table 2 lists the four most commonly used S-shape transfer functions (SSTF).

Example, take the first function, the common conversion method is as in Eqs. (8) and (9).

$$SSTF\left(q_n^{iter}\right) = \frac{1}{1 + e^{-\left(q_n^{iter}\right)}} \tag{8}$$

$$q_n^{iter+1} = \begin{cases} 0 \ if \ rand \ < \ SSTF\left(q_n^{iter}\right) \\ 1 \ if \ rand \ \geq SSTF\left(q_n^{iter}\right) \end{cases} \tag{9}$$

This paper presented the proposed BMRF with an adaptive S-shape transfer function (ASSTF), whereby the iterations of the BMRF algorithm are taken in to consideration with the transfer function, and dynamically modified the gradient to attain a refined turnover probability. For example, for the first transfer function in Table 1. The ASSTF in current position (q_n^{iter}) can be represented in Eqs. (10) though (11) and the new position is computed using Eq. (10).

$$ASSTF\left(q_n^{iter}\right) = \frac{1}{1 + e^{\frac{-\left(q_n^{iter}\right)}{k}}} \tag{10}$$

$$k = \left(1 - \frac{iter}{MaxIter}\right) * MaxIter_{max} + \frac{iter}{MaxIter} * MaxIter_{min} \tag{11}$$

$$q_n^{iter+1} = \begin{cases} 0 \ if \ rand \ < \ ASSTF\left(q_n^{iter}\right) \\ 1 \ if \ rand \ \geq ASSTF\left(q_n^{iter}\right) \end{cases} \tag{12}$$

where *iter* and *MaxIter* defines the existing iteration and the maximum number of iterations that the algorithm can run respectively. In this paper, the $MaxIter_{max}$ parameter is given a value 4 and $MaxIter_{min}$ as 0.01 as proposed in (Mafarja, Jaber, & Hammouri, 2017). At the initial phases of the iterations, a bigger *k* results in a lesser gradient in the curve, and a lesser change probability makes the algorithm give better attention to exploitation. At the later phases of the iterations, as *k* reduces, the curve gradient rises, the change probability rises, and the algorithm gives attention more on exploiting its capability to go out of optimal local solutions.

3.3. Random forest (RF) algorithm

RF, which was presented by Breiman (2001), is a decision tree (DT) algorithm that works through building several DT. RF can be described as a forest of DTs in which each tree casts a vote for the most popular input vector class. Compared to other algorithms such as artificial neural networks and support vector machines, RF has less parameters to be defined when running. A pool of individual tree arranged classifiers can be represented as in Eq. (8).

$$\{RF(y, \ \alpha_n), \ n = 1, 2, \ldots i, \ldots\} \tag{8}$$

where, *RF* is the classifier, $\{\alpha_n\}$ stands for identically independent distributed random vectors, and each DT has a vote for most famous class at input variable *y*. The nature and dimensionality of α hinge on its use in DT building.

The building of each of the DT that form the forest, is the key to the achievement of the RF. The diversity of RF could be obtained by sampling from the feature set, from the data set, or simply by randomly varying some decision tree parameters. Because of its high execution speed, RF is an appealing classifier, and it typically has much higher classification efficiency compared to a single decision tree. Mostly, the higher the number of trees in the forest, the more strongly it appears (Anyanwu & Shiva, 2009; Mahmudul, Md., M., & M., 2019).

4. Proposed model for network intrusion detection

The proposed IDs comprises some independent procedures. Fig. 3 illustrates the complete structure of the proposed model. The first procedure is the collection of the datasets and observation. Further, data pre-processing was carried out on the datasets. The data preprocessing techniques used here consists of encoding and normalization, sampling using k-means, and class oversampling. After that, the feature selection is based on a wrapper method using the proposed binary MRF optimization algorithm to remove irrelevant and redundant features from the dataset. Furthermore, the proposed model was developed using RF classifier and finally the model was evaluated using Accuracy, Precision, Recall, F-measure, and Execution time.

4.1. Description of the datasets

In this paper, two popular open-source network intrusion datasets (NSL-KDD and CIC-IDS2017) were used to train and test the proposed IDs. The NSL-KDD and CIC-IDS2017 were collected from Pahl and Aubet (2018) and Iman, Arash, and Ali (2018) respectively. The description of these datasets such as the number of instances and classes in each of the dataset are given in Table 3.

a. NSL-KDD Dataset: It is a modified and enhanced version of the KDD Cup'99 dataset that eliminates some of the problems associated with the KDD Cup'99 dataset (Tavallaee, Bagheri, Lu, & Ghorbani, 2009). The dataset contains 41 attributes and 148, 516 instances of Genuine and attacks. The attacks are classified into four types (Probe, DoS, U2R, and R2L) (Zeeshan, Adnan, Cheah, Johari, & Farhan, 2020). This dataset is used in this paper because of its numerous advantages, like non-redundant instance presence, which prevents unfair classification, and it can be used to effectively contrast various IDS methods. b. CIC-IDS2017 Dataset: It was created in 2017 by the Canadian Institute of Cyber Security (CIC). It includes both traditional attacks and modernized real-world network intrusions. The dataset contains 83 features and 2830,743 instances of Genuine and various attacks types. The network traffic flow is examined using CICFlowMete based on timestamps, destination IP addresses, source, protocols, and attacks. Furthermore, the dataset includes popular attack scenarios such as Infiltration,

Table 2
S-shape transfer functions.

Name	Function
$SSTF_1$	$\frac{1}{1+e^{-x}}$
$SSTF_2$	$\frac{1}{1+e^{-2x}}$
$SSTF_3$	$\frac{1}{1+e^{-\frac{x}{2}}}$
$SSTF_4$	$\frac{1}{1+e^{-\frac{x}{3}}}$

Table 3
Number of Instances and classes in the NSL-KDD and CIC-IDS2017 datasets.

NSL-KDD Class	Instances	CIC-IDS2017 Class	Instances
Probe	14,077	Genuine	2273,097
Genuine	77,053	Infiltration	36
DoS	53,383	DoS	380,699
U2R	254	Brute Force	13,835
R2L	3749	PortScan	158,930
		Bot	1966
		WebAttack	2180
Total	148, 516		2830,743

Table 4
Number of instances selected using k-means.

NSL-KDD Class	Instances	CIC-IDS2017 Class	Instances
Probe	884	Genuine	18,225
Genuine	4598	Infiltration	36
DoS	3181	DoS	3042
U2R	254	Brute Force	96
R2L	3749	PortScan	1255
		Bot	1966
		WebAttack	2180
Total	12,666		26,800

Table 5
Number of instances in the datasets after the complete pre-processing steps.

NSL-KDD Class	Instances	CIC-IDS2017 Class	Instances
Probe	1000	Genuine	18,225
Genuine	4598	Infiltration	1000
DoS	3181	DoS	3042
U2R	1000	Brute Force	1000
R2L	3749	PortScan	1255
		Bot	1966
		WebAttack	2180
Total	13,528		28,668

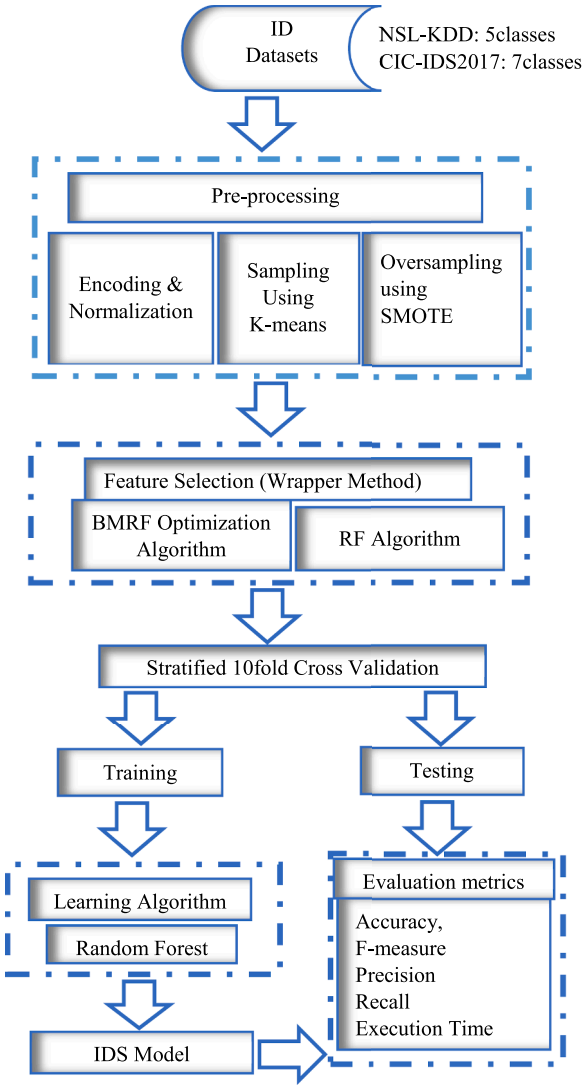

Fig. 3. Proposed Network Intrusion Detection Framework.

PortScan, Denial of Service (DoS), BruteForce, bot, and webAttack (Li, Abdallah, & Abdallah, 2021).

4.2. Preprocessing

Machine learning study needs exploratory data analysis and data observation so that the dataset can be more suitable for classification and also help reduce the intricacy and error rate of the classification model. The CIC-IDS2017 and NSL-KDD datasets employ in this paper are

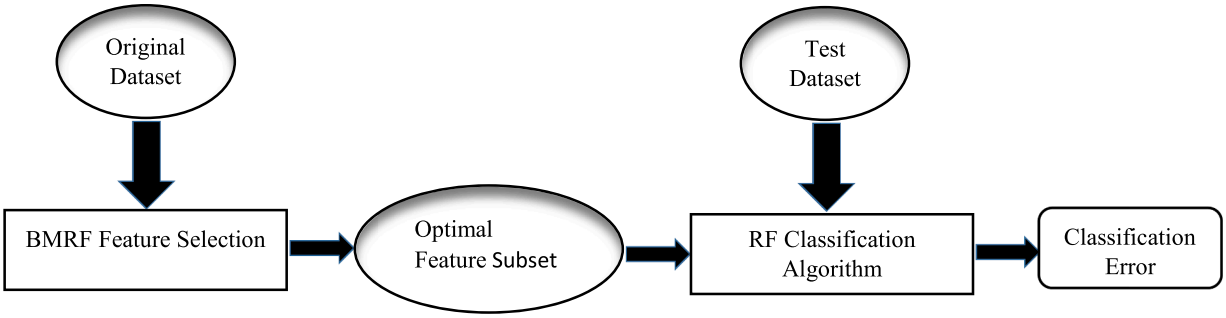

Fig. 2. BMRF Feature Selection Technique.

Table 6
Performance comparison of the eight BMRF with Random Forest classifier for Intrusion Detection using CIC-IDS2017 dataset.

Models	Metrics (%)	Genuine	Bot	Bruteforce	DoS	Infiltration	Portscan	WebAttack
$BMRFSSTF_1$+RF	Precision	95.8	74.6	97.6	99.0	**100**	97.0	88.8
	Recall	95.0	99.0	81.0	80.2	74.0	**100**	91.4
	Fmeasure	95.4	85.1	88.5	88.6	85.0	98.5	90.1
$BMRFSSTF_2$+RF	Precision	98.2	79.2	98.6	92.2	**100**	99.2	96.6
	Recall	95.4	95.4	81.0	97.8	74.0	**100**	90.8
	Fmeasure	96.8	86.5	88.9	94.8	84.8	99.8	93.6
$BMRFSSTF_3$+RF	Precision	78.0	99.0	**100**	95.4	0.0	67.8	**100**
	Recall	99.2	60.8	71.4	61.8	0.0	12.8	6.4
	Fmeasure	87.3	75.3	83.2	75.0	0.0	21.4	12.0
$BMRFSSTF_4$+RF	Precision	98.6	95.4	98.0	98.0	**100**	97.6	93.6
	Recall	99.0	95.8	89,4	98.6	57.0	96.6	94.0
	Fmeasure	98.8	95.6	93.5	98.0	72.5	97.1	94.0
$BMRFASSTF_1$+RF	Precision	99.0	**99.2**	**100**	**99.6**	**100**	**99.9**	99.6
	Recall	**100**	**99.0**	91.2	98.4	74.0	**100**	97.6
	Fmeasure	**99.5**	**99.1**	**95.4**	99.0	84.8	**99.9**	98.6
$BMRFASSTF_2$+RF	Precision	95.8	74.6	**98.6**	99.0	**100**	97.0	88.8
	Recall	95.0	**99.4**	81.0	80.2	**74.0**	**100**	91.4
	Fmeasure	95.4	85.4	88.9	88.6	84.8	98.4	90.0
$BMRFASSTF_3$+RF	Precision	**99.2**	97.6	**100**	99.0	**100**	99.2	**100**
	Recall	99.6	99.0	81.0	98.8	74.0	**100**	**97.8**
	Fmeasure	**99.2**	98.4	89.2	99.0	85.1	**99.6**	**98.9**
$BMRFASSTF_4$+RF	Precision	99.0	99.0	93.6	99.0	**100**	99.0	94.0
	Recall	99.0	97.6	85.4	**99.5**	74.0	99.8	93.0
	Fmeasure	99.0	98.3	89.3	99.2	85.1	99.4	93.5

Table 7
Performance comparison of the eight BMRF with Random Forest models for Intrusion Detection using NSL-KDD dataset.

Models	Metrics (%)	DoS	U2R	R2L	Probe	Genuine
$BMRFSSTF_1$+RF	Precision	99.2	77.2	95.0	94.8	96.4
	Recall	99.4	78.6	96.2	93.6	95.4
	Fmeasure	99.3	77.9	95.6	94.2	95.9
$BMRFSSTF_2$+RF	Precision	99.8	82.2	96.8	95.4	97.8
	Recall	98.2	79.0	97.8	96.8	97.2
	Fmeasure	98.9	80.6	**97.3**	96.1	97.5
$BMRFSSTF_3$+RF	Precision	**100**	81.2	96.4	96.0	97.6
	Recall	99.0	80.2	96.6	97.6	96.4
	Fmeasure	99.4	80.7	96.5	96.9	98.0
$BMRFSSTF_4$+RF	Precision	98.0	88.8	95.4	93.8	96.0
	Recall	97.5	84.4	96.0	94.2	95.2
	Fmeasure	97.7	86.5	95.7	93.9	95.6
$BMRFASSTF_1$+RF	Precision	**100**	89.4	96.4	97.6	96.8
	Recall	**99.8**	83.0	98.0	98.4	97.6
	Fmeasure	**99.9**	86.1	97.2	98.0	97.4.
$BMRFASSTF_2$+RF	Precision	99.8	**92.2**	96.4	97.8	98.0
	Recall	**99.8**	**88.0**	**98.2**	98.0	97.0
	Fmeasure	**100**	**90.1**	97.0	98.0	97.5
$BMRFASSTF_3$+RF	Precision	98.8	85.6	**96.8**	96.8	97.2
	Recall	99.6	81.8	97.2	95.4	96.8
	Fmeasure	99.2	83.6	97.0	96.1	97.0
$BMRFASSTF_4$+RF	Precision	99.4	89.2	95.8	96.0	97.6
	Recall	99.4	80.4	98.0	98.0	95.8
	Fmeasure	99.4	84.6	96.9	97.0	96.7

subjected through three pre-processing steps, which include encoding and normalization, sampling using k-means, and class oversampling using SMOTE. The steps are described below:

4.2.1. Data encoding and normalization

The intrusion datasets are firstly coded using label encoder, which is applied to convert categorical attributes into numerical attributes to help the inputs of machine learning models, because some machine learning models cannot directly support string attributes. Following that, the datasets are normalized using normal distribution, because the attributes in the network traffic flow dataset mostly have different scales, and machine learning models perform better on datasets with

Table 9
Average value of various models for CIC-IDS2017 dataset.

Models	Precision (%)	Recall (%)	F-measure (%)	Accuracy (%)	Execution Time(s)
BMRF+SVM	79.0	81.0	80.0	85.0	595.369
BMRF+Naïve Bayes	72.0	64.0	67.8	63.8	439.756
BMRF+XGBoost	94.8	89.2	91.9	95.8	447.454
BMRF+KNN	92.0	89.0	90.5	95.0	459.533
BMRF+RF (Proposed)	99.6	94.3	96.9	99.3	455.317

Table 10
Average value of various models for NSL-KDD dataset.

Models	Precision (%)	Recall (%)	F-measure (%)	Accuracy (%)	Execution Time
BMRF+SVM	66.0	39.0	49.0	39.22	1763.019
BMRF+Naïve Bayes	63.0	64.0	63.5	64.4	1697.433
BMRF+XGBoost	94.2	92.0	93.1	95.1	1704.434
BMRF+KNN	92.0	88.0	90.0	96.3	1699.642
BMRF+RF (Proposed)	96.8	96.2	96.5	98.8	1706.074

Table 8
Performance comparison with feature selection and without feature selection.

Datasets	Techniques	No. Features	Precision (%)	Recall (%)	F-measure (%)	Accuracy (%)
CIC-IDS2017	RF	83	85.2	92.5	88.7	90.2
	BMRF+RF(Proposed)	38	99.6	94.3	96.9	99.3
NSL-KDD	RF	41	93.5	93.5	93.5	93.3
	BMRF+RF(Proposed)	22	96.8	96.2	96.5	98.8

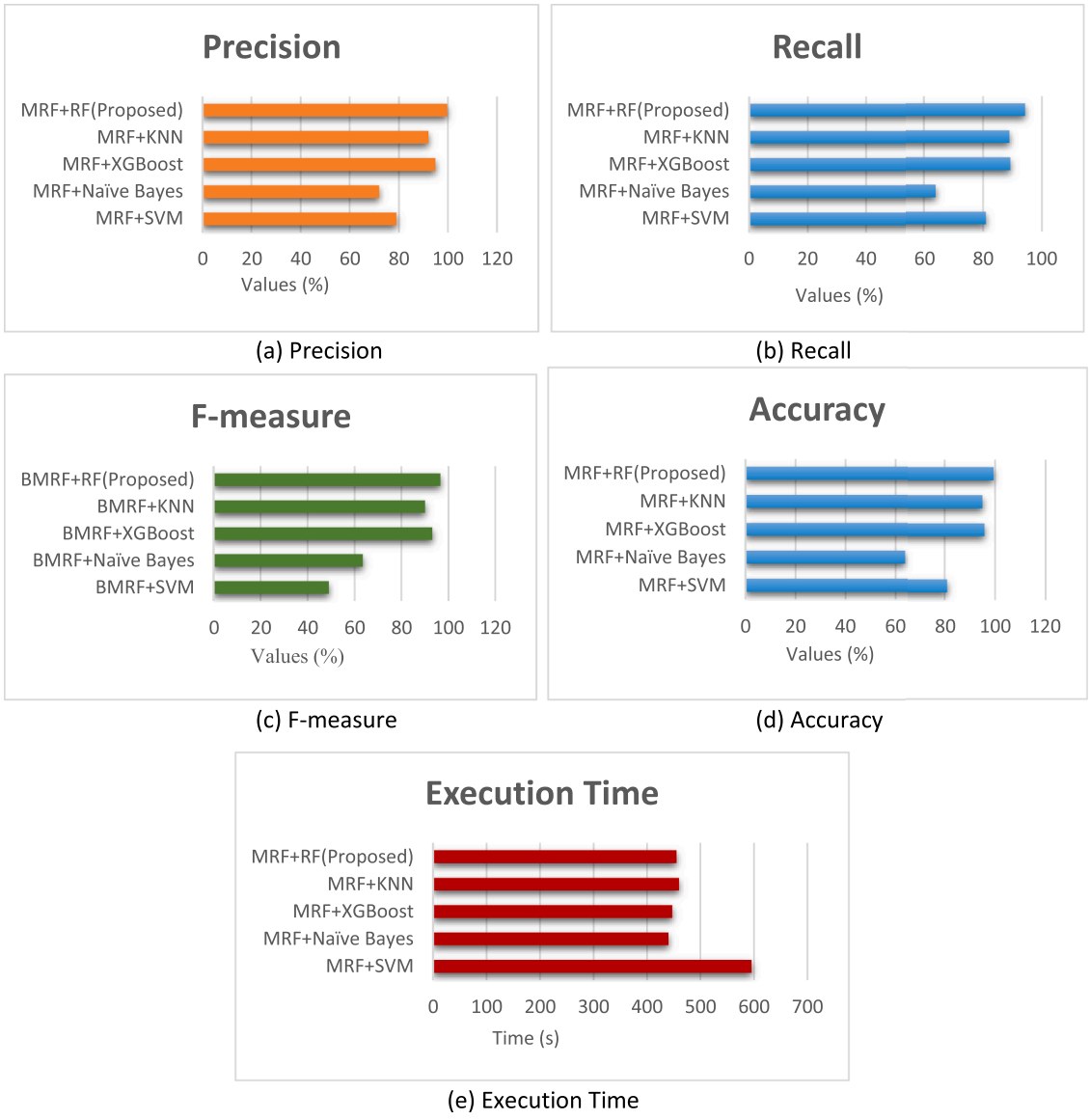

Fig. 4. Performance comparison of proposed IDS based on recall, precision, f-measure, accuracy, and execution time with other algorithms for CIC-IDS2017 dataset.

attribute values on the same scale (Khattab & Klaus, 2018). A biased machine learning model can be created from an unnormalized dataset with largely dissimilar attribute scales, which only prioritizes attributes with large scale values. The Z-score technique is used to normalize the intrusion detection dataset used in this study. The attributes are standardize so that there means and standard deviations will become 0 and 1 respectively using Z-score techniques. Eq (13) is used to calculate each normalized attribute value, y_n.

$$y_n = \frac{y - \mu}{\sigma}, \tag{13}$$

where y, is the initial attribute value, σ and μ represent the standard deviation and the mean values of the attribute respectively.

4.2.2. *Dataset sampling using k-means*

In practice, training Machine Learning Algorithms (MLA) on huge volume of network dataset is impractical and can take an inordinate quantity of time, particularly when fine-tuning the algorithms hyper-parameter, which requires training a MLA several times. For optimal training effectiveness enhancement drives, sampling is a common approach which can produce a small proportion of the entire dataset to reduce the difficulty of training the system (Faraoun & Boukelif, 2006; Li et al., 2021).

In the proposed intrusion detection model, to get a good dataset subset, a cluster sampling method based on k-means clustering procedure is applied. Dataset sampling using clustering technique is a method whereby the initial dataset instances are grouped into clusters; subsequently, from each cluster, a fraction of the dataset is selected cluster to generate the dataset subset (Faraoun & Boukelif, 2006). In contrast to random sampling technique, which picks all the instances randomly with the same probability, sampling using clustering techniques can produce good dataset subset as the thrown-out instances are commonly redundant instances.

Out of the known clustering algorithms, *k*-means is reported as the most used for dataset subset selection because of its simplicity, short computation time and also produce good clusters (Shi, Liu, & Guan, 2010). The *k*-means algorithm is applied to split the data into multiple clusters using distance metrics like Euclidean, Mahalanobis, and Manhattan (Moubayed, Injadat, Shami, & Lutfiyya, 2018, 2020). Because dataset samples from the same group are similar, selecting instances from different clusters may significantly decrease the dataset size without sacrificing useful information in the original dataset. The

Fig. 5. Performance comparison of proposed ID model based on recall, precision, f-measure, accuracy, and execution time with other algorithms for NSL-KDD dataset.

Table 11
Average value of various models for CIC-IDS2017 dataset.

Models	Precision (%)	Recall (%)	F-measure (%)	Accuracy (%)
GA+RF	91.0	90.3	90.6	96.1
PSO+RF	94.8	91.9	93.3	96.9
GWO+RF	85.1	79.4	82.2	92.4
GOA+RF	78.0	89.3	83.3	89.9
BMRF+RF (Proposed)	99.6	94.3	96.9	99.3

Table 12
Average value of various models for NSL-KDD dataset.

Models	Precision (%)	Recall (%)	F-measure (%)	Accuracy (%)
GA+RF	94.3	88.9	91.5	95.7
PSO+RF	95.0	93.1	94.0	97.0
GWO+RF	74.8	92.0	82.5	89.4
GOA+RF	89.1	90.8	89.9	90.8
BMRF+RF (Proposed)	96.8	96.2	96.5	98.8

primary objective of the *k*-means is to reduce the distance sum of squares amongst the dataset instances and the associated centroids, as shown in Eq. (14).

$$\sum_{i=0}^{m_p} \min_{c_j \in M_p} \left(y_i - c_j\right)^2, \tag{14}$$

where $(y_1, y_2 \ldots, y_m)$ represent the dataset instance matrix; c_j is the centre of cluster M_p; and m_p is the amount of instances in the cluster M_p.

The algorithm has a time complexity of $O(mpt)$, whereby m represents total instances in the dataset, p is the number of clusters, and t represents the algorithm stopping iteration (Shi et al., 2010).

After using the *k*-means to divide the datasets into a number of clusters, 10% of the instances in each cluster is selected using a random sampling technique. The selected subset from each cluster are combined, which form the final subset of the original dataset used in this study. The fraction of is chosen based on our resource constraints. Table 4 shows

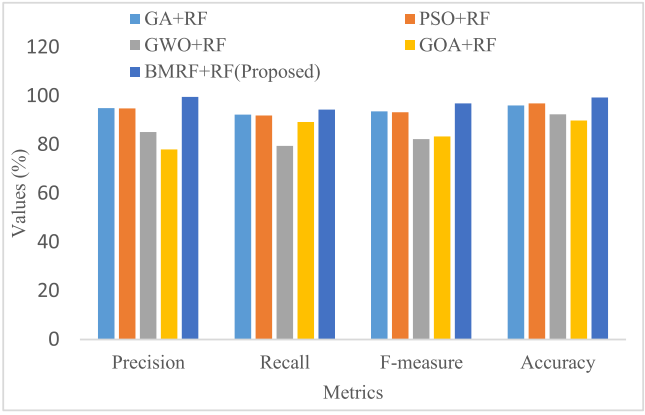

Fig. 6. Performance comparison of the proposed model with other meta-heuristics algorithms using CIC-IDS2017 dataset.

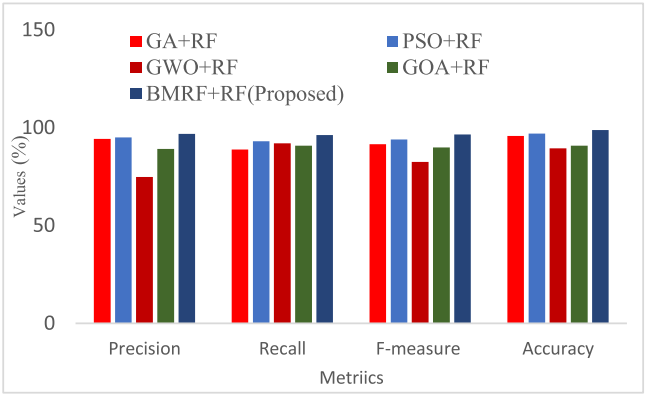

Fig. 7. Performance comparison of the proposed model with other meta-heuristics algorithms using NSL-KDD dataset.

Table 13
p-values produced by the *t*-test for the two datasets.

Dataset	GA+RF	PSO+RF	GWO+RF	GOA+RF
NSL-KDD	0.046	0.067	0.001	0.035
CIC-IDS2017	0.007	0.024	0.001	0.001

the number of instances in each class for the two datasets after sampling.

4.2.3. Class imbalance handling

Following normalization and sampling of the datasets, a method is used to deal with class imbalances in the datasets. Imbalance problems in a network traffic flow dataset are common because the proportion of genuine traffic flow in the real world is much greater than the proportion of attack traffic flow, giving rise to misleading models and a poor detection percentage (Zhenxiang et al., 2018).

Class imbalance challenges are primarily addressed by resampling strategies that include random sampling and SMOTE, that can produce new cases to balance the dataset (Onah et al., 2021; Zhenxiang et al., 2018). In contrast to random sampling, that just duplicates the instances and might result in over fitting, SMOTE by Chawla, Bowyer, Hall, and Kegelmeyer (2002) can generate good quality cases built based on idea of k Nearest neighbour (KNN); As a result, SMOTE is selected in this paper to effectively manage the disparity class problem in the datasets, resulting in high-quality minority class examples. Eq. (15) is used to generate the new instances in the minority classes. For example, if Y is an instance in the smaller class and Y_i is an instance selected randomly from neighbors of Y, an artificial instance Y_m is generated using Eq. (15).

The balance datasets using SMOTE is shown in Table 5, which is the final dataset used for feature selection and subsequently for building the proposed intrusion detection model.

$$Y_m = Y + r * (Y_i - Y),\ i = 1, 2, 3, \ldots, k \tag{15}$$

where r is a random number in the range of (0, 1) (Injadat, Moubayed, Nassif, & Shami, 2020)

4.3. Feature selection

Feature selection is a significant technique in developing network intrusion detection model whereby the important attributes are selected by eliminating the irrelevant and redundant ones. The difficulty and complexity of the model can be reduced by feature selection. In this study, we choose the features by utilizing the proposed BMRF optimization algorithm as the search strategy and random forest classifier as the learning algorithm in the wrapper feature selection method after the preprocessing procedure. The aim of our study is to reduce the number of features and guarantee the classification accuracy of the machine learning model. Consequently, we used the fitness function as shown in Eq. (16) to guide the BMRF algorithm search.

$$fitness = \alpha * CE + \beta * \frac{NCF}{TNF} \tag{16}$$

where, CE represent the classification error rate based on the current selected feature subset, which is obtain from the RF prediction, NCF represent the number of currently selected features, and TNF represent the total number of features in the dataset. Moreover, α and β denote the weight coefficients. A representation for the BMRF feature selection technique is given in Fig 2.

4.4. Classification

After finishing the feature selection procedure, we return the best feature subset from the original datasets, the attributes whose values is one (1) are kept for developing the model. The Random Forest classifier with 100 trees was used to evaluate the accuracy, precision, recall, and f-measure for the intrusion detection model using the reduced CICSID 2017 and NSL-KDD datasets. Stratified 10 fold cross fold validation, which means that 9/10 of the instances are used for training and 1/10 for testing is use. All the experiment use training and testing instances for each fold. It is also important to state that all the experimentations were runs 30 times in order to obtain a reasonable result.

4.5. Evaluation metrics

To evaluate the presented network intrusion detection model, the following five evaluation metrics were used as employed in evaluating various research works in IDs such as (Mehrnaz et al., 2018; Saranya, Sridevi, Deisy, Chung, & Khan, 2020; Talita et al., 2021).

Accuracy: Is the ratio of the number of intrusions ($T^{Positive}$) and normal activities ($T^{Negative}$) that are identified effectively to the number of the instances in the dataset as depicted in Eq. (17)

$$Accuracy = \frac{T^{Positive} + T^{Negative}}{T^{Positive} + F^{Positive} + T^{Negative} + F^{Positive}} \tag{17}$$

Recall: Is the proportion of the number of intrusions that are identified effectively (i. e. True positive ($T^{positive}$)) to the sum of the amount of intrusions that are recognized as intrusion ($T^{positive}$) and the amount of normal activities that are recognized as normal (i.e. True negative ($F^{Negative}$) as depicted in Eq. (18).

$$Recall = \frac{T^{positive}}{T^{positive} + F^{Negative}} \tag{18}$$

Precision: Is the proportion of the number of intrusions ($T^{positive}$) that

are identified to the total amount of intrusion ($T^{positive}$) and normal ($F^{positive}$) activities identified that as depicted in Eq. (19).

$$Precision = \frac{T^{positive}}{T^{positive} + F^{positive}} \tag{19}$$

F-measure: Define as the harmonic mean of recall and precision metrics which is depicted in Eq. (20).

$$F - measure = \frac{2 * Preciion * Recall}{Precision + Recall} \tag{20}$$

Execution Time**:** Define as the time spent to finish the classification of the various attacks which is computed using Eq. (21).

$$Execution_{time} = Start_{time} - End_{time} \tag{21}$$

5. Experimental results and discussion

In this part, various test and experiments have been conducted to determine the efficacy of the proposed network intrusion detection model. The hardware and software used is provided. Four performance comparisons was presented to showcase the efficiency of the proposed IDs. Firstly, the performance of the various BMRF optimization algorithm with 4 S-shaped and 4 adaptive S-shaped functions were presented. Secondly, performance comparison of RF with BMRF feature selection and without feature selection were also presented. Thirdly, performance of the BMRF optimization with other machine learning algorithms were also presented. Finally, the performance of the BMRF and other metaheuristics algorithms were presented. The statistical significance was also presented to show the statistical significance between the proposed method and the other techniques.

5.1. Experimental setup

The experiment was carried out on an Intel® Pentium® CPU B960 @ 2.20 GHz, with an Install memory (RAM) of 4.00 GB and 64-bit Operating System, x64-based processor. The test was carried out using Python Programming language in Jupyter Notebook.

5.2. Performance comparisons of various bmrf for detecting various attacks

To demonstrate the effectiveness of the proposed IDs, eight different IDs were developed and compared using precision, recall, and f-measure using four BMFR optimizations based on S-shape and four adaptive S-shape function for selecting appropriate number of features for network intrusion detection benchmark dataset and random forest as the classifier. Tables 6 and 7 show the results of this experiment for the NSL-KDD and CIC-IDS2017 benchmark datasets, respectively. The experiment reveals that the intrusion detection model using Binary manta ray with adaptive S-functions obtain the best precision, recall, and f-measure for most attacks and the normal traffic. For the subsequent comparison, the features selected by the binary manta ray with adaptive S-Shape transfer function ($BMRFASSTF_1$) chosen because it has the best average performance in terms of the precision, recall and F-measure for the CIC-IDS2017 dataset. For the NSL-KDD dataset, the $BMRFASSTF_2$ has the best average performance.

5.3. Performance comparison with feature selection and without feature selection

Here, the effectiveness of the proposed study with and without the feature selection was assess for both the two datasets. The results as shown in Table 8, the proposed method using the manta ray foraging optimization algorithm selected 22 attributes out of the 41 present in the NSL-KDD dataset with an accuracy of 98.8%, precision of 96.8, recall of 96.2%, and F-measure of 96.5%.

5.4. Performance comparison with other machine learning algorithms

In this paper, for building the IDs, we use the Random forest classifier. To demonstrate the efficacy of this classifier, the proposed IDs was contrasted with other machine learning classifiers that are applied to intrusion detection, such as *k*-nearest neighbour (KNN), Support vector machine (SVM), Naïve Bayes and XGBoost. The subsets generate by the $BMRFASSTF_1$ and $BMRFASSTF_2$ are used to train and test the model for the CIC-IDS2017 and NSL-KDD respectively. The percentage value for each model with respect to recall, accuracy, precision, f-measure, and execution time is depicted in Tables 9 and 10 and graphically shown in Figs. 4 and 5 for the CIC-IDS2017 and NSL-KDD benchmark datasets respectively. The experiment indicate that the proposed wok obtain better recall, precision, accuracy, and f-measure in contrast with all the compared techniques in the two datasets used. However, the proposed work was outperformed by Naïve Bayes and XGBoost for the CIC-IDS2017 and Naïve Bayes, KNN, and XGBoost for NSL-KDD dataset in terms of execution time.

5.5. Performance comparison with other metaheuristics techniques

To further assess the efficiency of our proposed intrusion detection model, we compare it with other well-known metaheuristic optimization algorithms applied to intrusion detection for feature selection such as GA, PSO, GWO, and GOA by performing experiment with the two datasets using the Random forest classifier. Tables 11 and 12 depicts the average performance of our proposed ID model as compared to the other methods for the CIC-IDS2017 and NSL-KDD respectively with respect to recall, precision, accuracy, and f-measure. The experimental results indicate that the proposed BMRF+RF intrusion detection model outperformed the GA, PSO, GWO, and GOA with 99.6% precision, 94.3% recall, 96.9% F-measure, and 99.3% accuracy for the CIC-IDS2017 dataset, for NSL-KDD dataset, the proposed model also shows better performance with 96.8%, 96.2%, 96.5%, and 98.8% with respect to recall, precision, accuracy, and f-measure respectively. Figs. 6 and 7 shows the graphical representation of this result.

5.6. Statistical assessment of the presented IDs using F-measure

To further prove the efficacy of the presented model, a statistical significance test was performed using *t*-test based on the F-measure result. This test is performed at 0.05% significance level for the two datasets and the results obtained, reported in Table 12, confirm the performance of our proposed model. The p values indicate that for the NSL-KDD dataset, there is statistical significance between the GA, GWO and GOA and the proposed model (i.e., $p<0.05$). However, the difference in the F-measure between the proposed model and PSO is just by chance, meaning there is no statistical significance between the two models with respect to the F-measure. Furthermore, the results show that there is a significance difference between the proposed model and other compared methods for the CIC-IDS2017 dataset (Table 13).

6. Conclusion

System and network security is one of the essential issues due to numerous threats and vulnerabilities on the internet. Consequently, intrusion detection is a vital part in system security. In this paper, we have presented a BMRF optimization algorithm and Random Forest classifier-based network intrusion detection model. The BMRF is a metaheuristic optimization algorithm employ for the feature selection procedure, which is a mixture of the MRF optimization algorithm and adaptive S-Shape transfer function. The feature space is searched till an optimal solution is achieved or stopping condition is reached. The optimal feature subset is used to train and test the intrusion detection model using the RF classifier. To assess the efficiency of the presented model, the NSL-KDD and CIC-IDS2017 intrusion detection datasets were

used. The experimental analysis indicates that in comparison with SVM, XGBoost, Naïve Bayes, and KNN, the presented model selected 38 features with 99.6% precision, 94.3% recall, 96.9% F-measure, and 99.3% accuracy for the CICID 2017 dataset. However, for the same dataset, the presented model was outperformed by Naïve Bayes and XGBoost in terms of execution time. Also, the presented model selected 22 features with 96.8%, 96.2%, 96.5%, and 98.8% for precision, recall, F-measure, and accuracy respectively for the NSL-KDD, with better execution time than SVM only. Furthermore, the statistical significance test reveals that there is a significance difference between the presented model and other metaheuristics methods in terms of F-measure. In the future, other classifiers other than RF can be investigated to improved performance of the classification process. Also, modification and hybridization of BMRF can be pursued for more efficient feature selection and other optimization problems. Furthermore, more efficient BMRF based feature techniques can be designed to address the issue of imbalance associated with the datasets.

CRediT authorship contribution statement

Ibrahim Hayatu Hassan: Conceptualization, Methodology, Software, Writing – original draft. **Mohammed Abdullahi:** Investigation, Writing – review & editing. **Mansur Masama Aliyu:** Validation, Writing – review & editing. **Sahabi Ali Yusuf:** Methodology, Supervision. **Abdulrazaq Abdulrahim:** Writing – review & editing.

Declaration of Competing Interest

None.

References

Abdullah, M., Balamash, A., Alshannaq, A., & Almabdy, S. (2018). Enhanced intrusion detection system using feature selection method and ensemble learning algorithms. *International Journal of Computer Science and Information Security (IJCSIS)*, 16.

Aburomman, A. A., & I, R. (2016). A novel SVM-KNN-PSO ensemble methods for intrusion detection system. *Applied Soft Computing*, 360–372, 10.1016/j.asoc.2015.10.011.

Al-Saqqa, S., Al-Fayoumi, M., & Qasaimeh, M. (2021). Intrusion Detection System for Malicious Traffic Using Evolutionary Search Algorithm. *Recent Advances in Computer Science and Communications*, 1381–1389.

Al-Yaseen, W. (2019). Improving intrusion detection system by developing feature selection model based on Firefly algorithm and support vector machine. *IAENG Internal Journal of Computer Science*, 534–540.

Alzubi, Q., Anbar, M., Alqattan, Z., Al-betar, M., & Abdullah, R. (2019). Intrusion detection based on a modified grey worlf optimization. *Neural Computing Apllication*, 6125–6137.

Anitha, P., & Kaarthick, B. (2019). Oppositional based Laplacian grey wolf optimization algorithm with SVM for data mining in intrusion detection system. *Journal of Ambient Intelligence and Humanized Computing*, 1–12. https://doi.org/10.1007/s12652-019-01606-6

Anyanwu, M. N., & Shiva, G. N. (2009). *International Journal on Computer Science and Security*, 230–240.

Aslahi-Shahri, B., Rahmani, R., Maralani, A., Eslami, M., Golka, M., & Ebrahimi, A. (2016). A hybrid method consisting of GA and SVM for intrusion detection system. *Neural Computing Application*, 1669–1676.

Balasaraswathi, V. R., Sugumaran, M., & Hamid, Y. (2017). Feature selection techniques for intrusion detection using non-bio-inspired and bio-inspired optimization algorithms. *Journal of Communications and Information Networks*, 107–119.

Bayu, A. T., & Kyung-Hyune, R. (2018). An Integration of PSO-based Feature Selection and Random Forest for Anomaly Detection in IoT Network. In *MATEC Web of Conferences (IJCAET & ISAMPE 2017* (p. 01053). EDP Sciences.

Breiman, L. (2001). Random forests. *Machine Learning*, 5–32. https://doi.org/10.1023/A:1010933404324

Cao, H., Guo, Z., & Yongquan, Z. (2017). Binary symbiotic organism search algorithm for feature selection and analysis. *IEEE Access*, 1–27.

Chawla, N. V., Bowyer, K. W., Hall, L. A., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence*, 321–357.

Ebrahimpour, M., & Eftekhari, M. (2017). Ensemble of feature selection methods: A hesitant fuzzy sets approach. *Applied Soft Computing*, 300–312.

Faraoun, K. M., & Boukelif, A. (2006). Neural Networks Learning Improvement using the K-Means Clustering Algorithm to Detect Network Intrusions. *INFOCOMP Journal on Computational Science*, 28–36.

Froehlich, F., & Kent, A. (1998). *The froehlich/kent encyclopedia of telecommunications, 17*. CRC Press.

Gan, X. S., Duanmu, J. S., Wang, J. F., & Cong, W. (2013). Anomaly intrussion detection based on PLS feature extraction and core vector machine. *Knwoledge Based System*, 1–6.

Hasan, M., Islam, M. M., Zarif, M. I., & Hashem, M. (2019). Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet of Things, 7*, 10.1016/j.iot.2019.10 0 059.

Hussein, A., & Ku, R. K.-M. (2021). Grey Wolf Optimization parameter control for feature selection in anomaly detection. *International Journal of Intelligent Engineering and Systems*, 474–483. https://doi.org/10.22266/ijies2021.0430.43

Iman, S., Arash, H. L., & Ali, A. G.-n. (2018). Toward generating a new intrusion detection dataset and intru-sion traffic characterization. In *4th International Conference on In-formation Systems Security and Privacy (ICISSP)*. Portugal.

Injadat, M., Moubayed, A., Nassif, A. B., & Shami, A. (2020). Multi-stage optimized machine learning framework for network intrusion detection. *IEEE Transactions on Networking Service and Management*.

Islam, M. J., Li, X., & Mei, Y. (2017). A Time-Varying Transfer Function for Balancing the Exploration and Exploitation ability of a Binary PSO. *Applied Soft Computing*, 182–196.

Khattab, M. A., & Klaus, M.-. M. (2018). Intelligent intrusion detection in external communication systems for autonomous vehicles. *System Science and Control Engineering,*, 48–56.

Li, Y., Abdallah, M., & Abdallah, S. (2021). MTH-IDS: A Multi-Tiered Hybrid Intrusion Detection System for Internet of Vehicles. *IEEE Internet of Things Journal*, 1–17.

Mafarja, M. M., Jaber, I., & Hammouri, A. (2017). *Binary dragonfly algorithm for feature selection*. ICTCS.

Mafarja, M., Aljarah, I., Faris, H., Hammouri, A., Ala'M, A. Z., & Mirjalili, S. (2019). Binary grasshopper optimisation algorithm approaches for feature selection problems. *Expert System. Application,*, 267–286.

Mahmudul, H., Md, M. I., M, I., & M, M. H. (2019). Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet of Things*, 10 0 059.

Mehrnaz, M., Babak, S., & Iraj, M. (2018). Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms. *Journal of King Saud University - Computer and Information*. https://doi.org/10.1016/j.jksuci.2018.03.011

Mirjalili, S., & Lewis, A. (2013). S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation*, 1–14.

Mohammadi, S., Mirvaziri, H., Ghazizadeh-Ahsaee, M., & Karimipour, H. (2019). Cyber intrusion detection by combined feature selection. *Journal Information Security Application*, 80–88.

Moubayed, A., Injadat, M., Shami, A., & Lutfiyya, H. (2018). DNS Typo-Squatting Domain Detection: A Data Analytics & Machine Learning Based Approach. In *2018 IEEE Global. Communication Conference (GLOBECOM 2018)*. IEEE.

Moubayed, A., Injadat, M., Shami, A., & Lutfiyya, H. (2020). Student Engagement Level in e-Learning Environment: Clustering Using K-means. *American Journal of Distance Education*, 1–20.

Mukaram, S., Mohammed, O., & Laith, A. (2020). Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks. *Journal of Ambient Intelligence and Humanized Computing*, 1–18.

Omar, A. (2020). A Feature Selection Model for Network Intrusion Detection System Based on PSO, GWO, FFA and GA Algorithms. *Symmetry*, 1–20.

Onah, J. O., Abdulhamid, S. M., Abdullahi, M., Hassan, I. H., & Al-Ghusham, A. (2021). Genetic Algorithm based feature selection and Naïve Bayes for anomaly detection in fog computing environment. *Machine Learning with Applications*. https://doi.org/10.1016/j.mlwa.2021.100156

Pahl, M. O., .& Aubet, F. (2018,. December 29). *DS2OS traffic traces*. Retrieved from Kaggle: https://www.kaggle.com/francoisxa/ds2ostraffictraces.

Pankaj, K. K., Mahesh, C. G., Emmanuel, S. P., & Prajjval, G. (2021). A smart anomaly-based intrusion detection system for the Internet of Things (IoT) network using GWO–PSO–RF model. *Journal of Reliable Intelligent Environments*, 3–21.

Rashmita, K., K, M., Mary, S. C., J, A., Anitha, T., & Rajendran, T. (2021). A hybrid network anomaly detection system using glowworm swarm optimization with principal component analysis. *Research Square*, 1–12. https://doi.org/10.21203/rs.3.rs-408246/v1

SaiSindhuTheja, R., & Shyam, G. K. (2021). An efficient metaheuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment. *Applied Soft Computing Journal*. , Article 106997. https://doi.org/10.1016/j.asoc.2020.106997

Salo, F., Nassif, A., & Essex, A. (2019). Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection. *Computational Networks*, 164–175.

Saranya, T., Sridevi, S., Deisy, C., Chung, T. D., & Khan, M. A. (2020). Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review. *Procedia Computer Science, 171*(2020), 1251–1260.

Shi, N., Liu, X., & Guan, Y. (2010). Research on k-means clustering algorithm: An improved k-means clustering algorithm. In *3rd Int. Symp. Intell. Inf. Technol. Secur. Informatics (IITSI)* (pp. 63–67).

Shubhra, D., Manu, V., & Sarsij, T. (2021). Building an efficient intrusion detection system using grasshopper optimization algorithm for anomaly detection. *Cluster Computing*, 1–21. https://doi.org/10.1007/s10586-020-03229-5

Shukla, A. K. (2021). Detection of anomaly intrusion utilizing self-adaptive grasshopper optimization algorithm. *Neural Computing and Applications*, 7541–7561.

Syarif, I. (2016). Feature selection of network intrusion detection data using genetic algorithm and particle swarm optimization. *MITTER international journal of Engineering Technology*, 277–290.

Talita, A. S., Nataza, O. S., & Rustam, Z. (2021). Naïve Bayes Classifier and Particle Swarm Optimization Feature Selection Method for Classifying Intrusion Detection System Dataset. *Journal of Physics*, Article 012021.

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD cup 99 data set. In *IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA 2009)* (pp. 1–6). IEEE.

Thaseen, I., & Kumar, C. (2017). Intrusion detection model using fusion of Chi-square feature selection and multi class SVM. *Journal of King Saud University - Computer and Information Science*, 462–472.

Tidjon, L., Frappier, M., & Mammar, A. (2019). Intrusion detection systems: A cross-domain overview. *IEEE Communication Surveys and Tutorials*, 3639–3681.

Turgut, O. E. (2020). A novel chaotic mantaray foraging optimization algorithm for thermoeconomic design optimization of an airfin cooler. *SN Applied Sciences*, 1–36. https://doi.org/10.1007/s42452-020-04013-1

wivedi, S., Vardhan, M., & Tripathi, S. (2020). Distributed denial-of service prediction on iot framework by learning techniques. *Open Computer Science*, 220–230.

Yi, J. H., Wang, J., & Wang, G. G. (2016). Improved robabilistic neural networks with self-adaptive strategies for transformer fault diagnosis problem. *Advance Mechical Engineering*, 1–13.

Yuyang, Z., Guang, C., Shanqing, J., & Mian, D. (2020). Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Computer Networks*, 107247. 10.1016/j.comnet.2020.107247.

Zarshenas, A., & Suzuki, K. (2016). Binary coordinate ascent: An efficient optimization technique for feature subset selection for machine learning. *Knowledge Based System*, 191–201.

Zeeshan, A., Adnan, S. K., Cheah, W. S., Johari, A., & Farhan, A. (2020). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Wiley*, 1–29. https://doi.org/10.1002/ett.4150

Zhao, W., Zhang, Z., & Wang, L. (2020). Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications. *Engineering Applications on Artificial Intelligence*. , Article 103300. https://doi.org/10.1016/j.engappai.2019.103300

Zhenxiang, C., Qiben, Y., Hongbo, H., Shanshan, W., Lizhi, P., Lin, W., et al. (2018). Machine learning based mobile malware detection using ghly imbalanced network traffic. *Information Science*, 346–364. https://doi.org/10.1016/j.ins.2017.04.044