Computers, Materials & Continua
DOI: 10.32604/cmc.2022.027475
Article

Tech Science Press

Bio-inspired Hybrid Feature Selection Model for Intrusion Detection

Adel Hamdan Mohammad[1,*], Tariq Alwada'n[2], Omar Almomani[3], Sami Smadi[3] and Nidhal ElOmari[4]

[1]Computer Science Department, The World Islamic Sciences and Education University, Amman, Jordan
[2]Network and Cybersecurity Department, Teesside University, Middlesbrough, United Kingdom
[3]Information and Network Security, The World Islamic Sciences and Education University, Amman, Jordan
[4]Software Engineering, The World Islamic Sciences and Education University, Amman, Jordan
*Corresponding Author: Adel Hamdan Mohammad. Email: Adel.hamdan@wise.edu.jo
Received: 18 January 2022; Accepted: 23 March 2022

Abstract: Intrusion detection is a serious and complex problem. Undoubtedly due to a large number of attacks around the world, the concept of intrusion detection has become very important. This research proposes a multilayer bio-inspired feature selection model for intrusion detection using an optimized genetic algorithm. Furthermore, the proposed multilayer model consists of two layers (layers 1 and 2). At layer 1, three algorithms are used for the feature selection. The algorithms used are Particle Swarm Optimization (PSO), Grey Wolf Optimization (GWO), and Firefly Optimization Algorithm (FFA). At the end of layer 1, a priority value will be assigned for each feature set. At layer 2 of the proposed model, the Optimized Genetic Algorithm (GA) is used to select one feature set based on the priority value. Modifications are done on standard GA to perform optimization and to fit the proposed model. The Optimized GA is used in the training phase to assign a priority value for each feature set. Also, the priority values are categorized into three categories: high, medium, and low. Besides, the Optimized GA is used in the testing phase to select a feature set based on its priority. The feature set with a high priority will be given a high priority to be selected. At the end of phase 2, an update for feature set priority may occur based on the selected features priority and the calculated F-Measures. The proposed model can learn and modify feature sets priority, which will be reflected in selecting features. For evaluation purposes, two well-known datasets are used in these experiments. The first dataset is UNSW-NB15, the other dataset is the NSL-KDD. Several evaluation criteria are used, such as precision, recall, and F-Measure. The experiments in this research suggest that the proposed model has a powerful and promising mechanism for the intrusion detection system.

Keywords: Intrusion detection; Machine learning; Optimized Genetic Algorithm (GA); Particle Swarm Optimization algorithms (PSO); Grey Wolf Optimization algorithms (GWO); FireFly Optimization Algorithms (FFA); Genetic Algorithm (GA)

1 Introduction

The popularity of the Internet makes it an excellent means of communication. Several studies indicate that the number of Internet users around the world has been increased [1,2]. Also, most organizations worldwide need connections to the Internet to perform their operations and services. One other fact is that the number of individual users who need to connect their personal computers and devices to the Internet is amplified. Thus, internet security has become a major concern to protect devices and applications [3,4]. Furthermore, machines and applications face various security threats. Threats have several forms and shapes, such as computer viruses, spyware, ransomware, and others [5,6]. Intrusion detection is a serious topic related to information and network security. Besides, intrusion detection aims to detect intrusion before spreading and causing damage [7,8].

Cybersecurity is a term used to describe a set of technologies, procedures, and processes designed to protect computers, applications, networks, and data from unauthorized access [9,10]. Usually, security systems consist of firewalls, antivirus applications, and intrusion detection systems. In addition, intrusion detection is a security branch that aims to monitor, discover, and identify any unauthorized access [11–13].

This research presents a new multilayer bio-inspired feature selection model for intrusion detection using an optimized genetic algorithm. Feature selection is a crucial factor for the success or failure of intrusion detection. Several studies talk about using feature selection for intrusion detection. Besides, several methods and techniques are used for feature selection, such as PSO, GWO, FFA, and GA [14–20]. The proposed model in this research employs PSO, GWO, FFA, and GA in a new emergent way. The proposed model in this research uses a multilayer feature selection model. At layer 1, PSO, GWO, and FFA are used to generate sets of features. Then at layer 2, optimized GA is used to select one of the generated sets of features based on the feature set's priority. Modifications were done on GA to assign priority values for each feature set. Besides, the experiment is this research done on two benchmark datasets (UNSW-NB15 [21] and NSL-KDD [22,23]).

Fig. 1 below shows the abstract view of the proposed model. **The contribution of this research can be summarized into the following points**:

- Using four optimization algorithms in a new manner.
- Three optimizing algorithms will be used to generate feature sets (layer 1).
- Modification is applied on GA to assign a priority value for each feature set based on the F-Measure value.
- Modification is applied on GA to select a set from generated feature sets based on the priority value (Layer 2).
- A priority value will be assigned to each feature set, and this priority value will be modified from the optimized GA algorithm according to F-Measure.

The rest of this research is organized as the following: Section 2 presents up-to-date related studies. In Section 3, bio-inspired optimization algorithms and machine learning classifiers used in this research will be presented. Section 4 demonstrates the proposed model. In Section 5, datasets used in this experiment are presented. Section 6 presents the proposed model experiments and results. Finally, Section 7 presents the conclusion and future works.

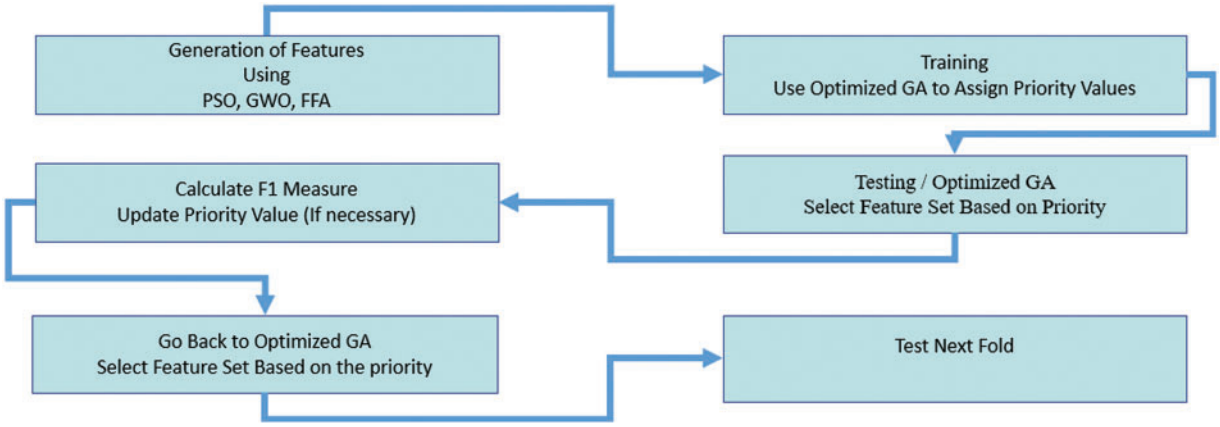

Figure 1: Abstract view of the proposed model

2 Related Studies

This section presents up-to-date related studies for intrusion detection using bio-inspired and machine learning algorithms.

Yin et al. [24] explore how to model intrusion detection using deep learning. The authors present (RNN-IDS) model using recurrent neural networks. Experiments were done using binary classification and multiclass classification. Also, comparisons are made with J48, Artificial Neural Networks (ANN), Random Forest (RF), Support Vector Machine (SVM), and other machine learning methods. Besides, experiments show that RNN-IDS is suitable for modeling a classification model with acceptable accuracy. Furthermore, experiments were done on the NSL-KDD dataset.

Ashahri et al. [25] present a hybrid model of Support Vector Machines (SVM) and a Genetic Algorithm (GA) for the intrusion detection problem. The number of features is reduced from 45 to 10. In this research, features are categorized into three priority values. The true positive rate is 97.3%, and the false positive is 0.017%. Finally, the KDDCUP99 dataset is used in this experiment.

Chung et al. [3] propose a hybrid intrusion detection system by using an intelligent swarm-based rough set (IDS-RS) for feature selection. IDS-RS aims to select the most relevant feature that can present the pattern of network traffic. Besides, in this research, a new weighted local search strategy incorporated in simplified swarm optimization (SSO-WLS) is developed. The dataset used is KDDCup99, and the authors say that SSO-WLS can improve performance. Also, the main idea of WLS is to improve the searching process. Finally, testing shows an accuracy of more than 93.3%

Wang et al. [26] propose an approach called (FC-ANN). This approach is based on artificial neural networks and fuzzy clustering. Also, Fuzzy Clustering (FC) is used to generate different training subsets. Based on different training subsets, different ANN models are trained to formulate different base models. Experiments were done with the KDDCUP99 dataset. Besides, machine learning classifiers, such as Naïve Bayesian and decision tree are used.

Çavuşoğlu et al. [27] propose a hybrid and layered intrusion detection system. Also, the author uses a combination of different machine learning and feature selection methods. The dataset used in this experiment is NSL-KDD. Besides, The number of attributes in this research is reduced by using two proposed feature selection methods. The proposed system has a high accuracy result.

Chen et al. [28] use the Support Vector Machine (SVM) with $tf \times idf$ and Artificial Neural Network (ANN) for intrusion detection. The dataset used in this research is BSM audit data from the DARPA 1998 intrusion detection evaluation program at MIT's Lincoln Labs. Experiments show that ANN with a simple frequency-based scheme achieved less performance than SVM.

Aljawarneh et al. [29] develope an enhanced J48 algorithm. Enhanced J48 is used in intrusion detection. For evaluation purposes, the authors use J48, Naïve Bayes (NB), Random Tree (RT), and NB-Tree. The dataset used in this experiment is NSL-KDD. Besides, the WEKA application is used for evaluation purposes. Detection accuracy in this research is 99.88% for the 10-fold cross-validation.

Almomani [7] develope a new model for intrusion detection. The author in this research uses GA, PSO, GWO, and FFA for feature selection. Evaluation of features generated is done based on Support Vector Machine (SVM) and J48. Besides, experiments in this research were done using the UNSW-NB15 dataset. Finally, measurements of performance are done based on precision, recall, and F-Measure.

Chen et al. [30] present a Support Vector Machine (SVM) intrusion detection model based on compressive sampling. Experiments in this research were done with the KDDCUP99 dataset. The compressed technology is used to realize network data compression. The authors conclude that compressed sensing's intrusion detection model had no significant change in detection rate. Furthermore, detection time, in this research, is reduced.

Haider et al. [31] propose an intrusion detection model based on real-time sequential deep extreme learning machine cybersecurity (RTS-DELM-CSIDS). This model initially determines the rating of security aspects contributing to their significance and then develops a comprehensive intrusion detection framework focused on the essential characteristics. The dataset used in these experiments is NSL-KDD, and the results show 96.22% and 92.73 accuracies.

3 Bio-Inspired Optimization Algorithms and Machine Learning Classifiers

Several bio-inspired algorithms are available, such as PSO, GWO, FFA, and GA. These algorithms show emergent results in feature reduction. PSO was created by Kennedy et al. [32], and it can search a vast space of candidate's solutions. PSO is not guaranteed to find an optimal solution. Besides, PSO uses a fitness function for evaluation purposes [33,34]. GWO is developed by Mirjalili in 2014 [35]. GWO simulates the hunting process of gray wolves. Four types of gray wolves simulate the nature of wolves which are are alpha, beta, delta, and omega. Besides, the steps of attacking the victim are as follows: searching for the victim, creating a circle around the victim, and finally attacking the victim [35–37]. FFA is a new metaheuristic algorithm. FFA was first introduced by Yang in 2008 [38,39]. FFA optimizer algorithm has been applied to several optimization problems and used to solve several real words problems, such as scheduling, environment, and economics [40–42]. GA is a heuristic search optimization algorithm. GA is used to solve different kinds of complex real-life problems in different areas, such as economics, engineering, multimedia, information security, management, and engineering [43–45]. Furthermore, GA is a metaheuristic algorithm inspired by biological processes. Also, GA is a population-based search algorithm. GA utilizes several concepts, such as fitness function, mutation, and crossover [46–48].

Machine Learning classifiers (MLC) algorithms are used for classifying data. Different methods and techniques are used for classification, such as Naïve Bayesian, Support Vector Machines (SVM), Artificial Neural Network (ANN), Decision Tree (C4.5), or (J48) classifier, K-Nearest Neighbor, and Random Forests (RF) [5,8,11,49–55]. In this research, J48 and Random Forest classifiers are used. J48

or C4.5 is a widely used machine learning algorithm. J48 is a decision tree classifier. J48 is considered a type of ID3 developed by Quinlan [50]. Random Forest Classifier (RFC) is an easy, flexible, and machine learning algorithm that can produce high-quality results. Also, the random forest is one of the most used algorithms. RF can be used for classification and regression.

4 The Proposed Model

This section proposes the new multilayer bio-inspired feature selection model for intrusion detection using an optimized genetic algorithm. The proposed model is divided into two layers. At layer 1, PSO, GWO, and FFA algorithms are used independently; as in Fig. 2 below, each algorithm is used to generate a set of features. For example, the first group of features is saved in set number 1, the second group of features is saved in set number 2, and so on. Each algorithm is executed for a specific number of iterations specified by the users. In this experiment, each algorithm is executed for 50 iterations to generate 50 independent sets; the total number of generated sets is 150.

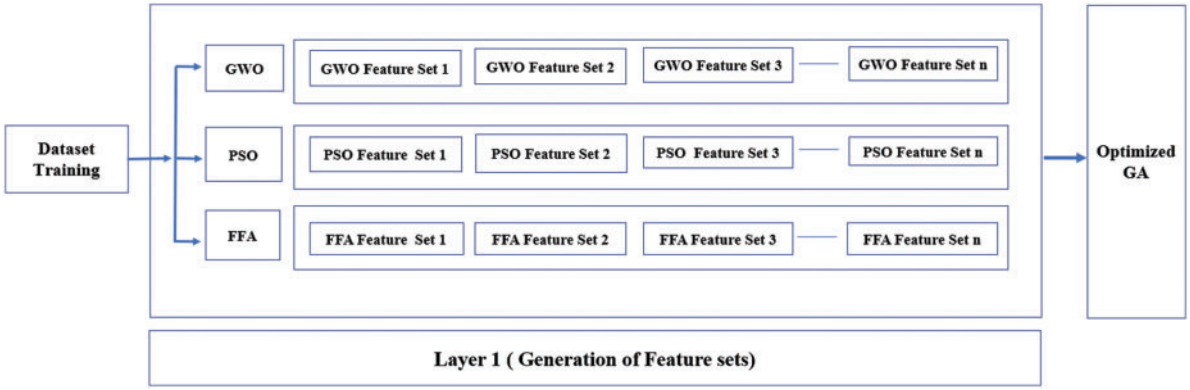

Figure 2: Layer 1 generation of feature sets

At layer 1 experiments, as shown in Fig. 3 below, optimized GA is used as the following: the training dataset is divided into a few numbers of folds selected by the user. Optimized GA is used to choose one feature set from the features set created at layer 1. Training is conducted, and F-Measure is calculated at the end of the fold. Based on the F-Measure value, the selected feature set will be given a priority value according to the following criteria: if F-Measure is greater than or equal to 90%, then the selected feature set will be given a priority value = high. If the F-Measure is greater than or equal to 80% and less than 90%, then the selected feature set will be given a priority value = medium. Otherwise, the selected feature set will be given a priority value = low. After completing the selected fold, optimized GA must select another feature set from the sets of features that have not been given a priority value. By the end of this phase, all feature sets will be assigned a priority value.

At the layer 2 experiments, as in Fig. 4 below, optimized GA is used as follows: the testing dataset is divided into numbers of folds selected by the user. Optimized GA is used to choose one set of generated feature set from layer 1. Selection of feature set at this stage will be according to priority value. If for any reason, the result of the F-Measure for any fold is not suitable with the priority assigned with the selected feature set, an update for priority value will occur.

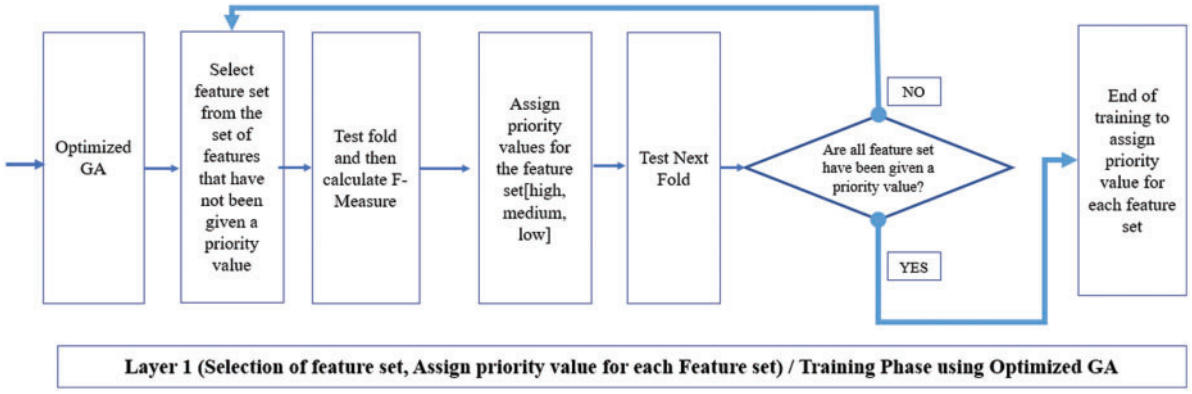

Figure 3: Layer 1 optimized GA/Training phase

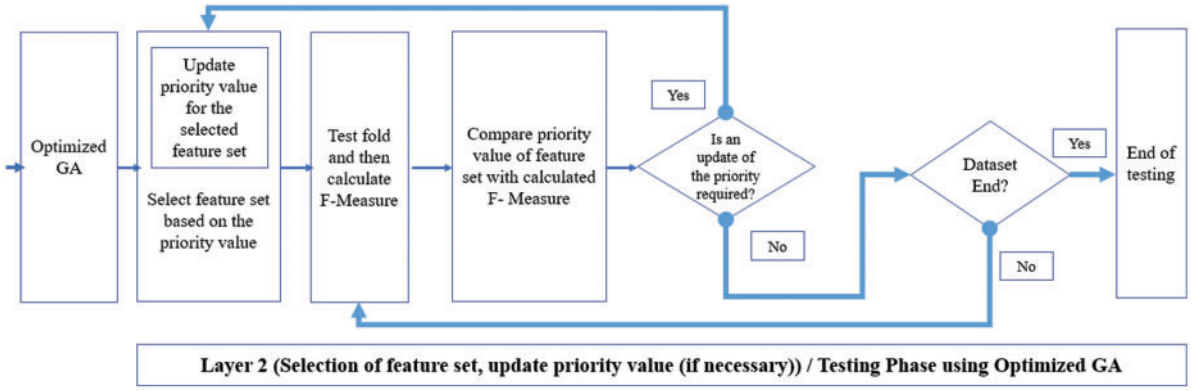

Figure 4: Layer 2 optimized GA/Testing phase

The overall proposed system is shown in Fig. 5 below. The proposed model works with two layers of bio-inspired feature selection model. GWO, PSO, and FFA will work at layer 1. These three bio-inspired algorithms aim to create feature sets and assign a priority value for each feature set with the help of optimized GA. At layer 2 experiments, the proposed model will use optimized GA. Optimized GA is modified to select feature sets based on their priority. At the end of each fold, F-Measure is calculated, modification on the feature set priority may occur based on F-Measure value and feature set priority. Besides, experiments in this research were performed with two well-known datasets.

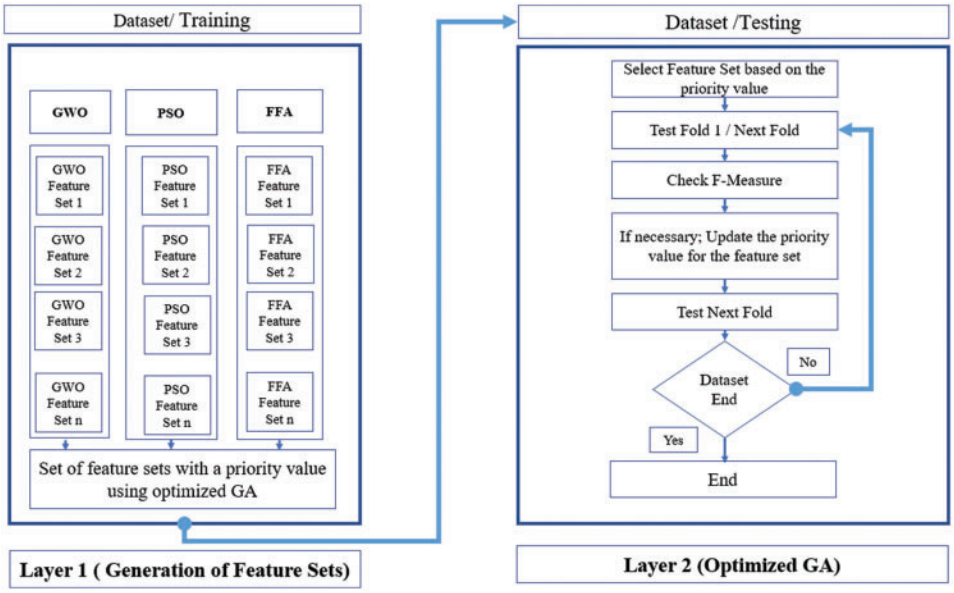

Figure 5: The proposed system

In this section, this study will present the GA algorithm modifications to fit the proposed model during the training phase.

Optimized GA for Proposed Model // Training Phase

Get Features set generated from layer 1 and give each feature set a number, which means optimized GA will get a set consists of 150 feature sets.
Step 1: initialize features sets {*get features sets from layer 1 experiments*}
Initialize features set ← {*FS1, FS2, FS3 . . . . FSn); n: number of features set selected by the user in layer 1 experiments*.
Step 2: Run GA to select one feature set, Name of the selected feature set is S-FS;
S-FS ← {*FS1, FS2, FS3 . . . . FSn*}
Step 3: continue learning with fold/Next Fold using the proposed model, using the same feature set, calculate F1 measure, set the priority of feature set based on F-Measure according to the following:
While Dataset fold is not empty, Do
With selected fold Do
Evaluate all records in the fold and calculate F-Measure
Check F-Measure ← *Calculate F-Measure*
Priority = high ← *If F-Measure Measure for S-FS >=90%.*
Priority = Medium ← *If F-Measure is greater than >=80 % and <90%.*
Priority = Low ← *If F-Measure is <80%.*
End of Fold
Step 3.1: select a random feature set from the sets of features that have not given priority.
R-FS: Random feature set ← {*set of features which have not been given a priority value*}
Step 3.2: continue training; test next fold; calculate F-Measure at the end of each fold; by the end of the training, all feature sets will have a priority value assigned for each feature set.
End While

In this section, the authors will present the GA algorithm modifications to fit the proposed model during the testing phase.

Optimized GA for Proposed Model // Testing Phase

Step1: Select Features set based on a priority value assigned
If set has a priority = high; then probability of selection is greater >=75% ← {all set of features which have been given a priority value = high}
If set has a priority = Medium; then probability of selection is >=50% and <75% ← {all set of features which have been given a priority value = medium}
If set has a priority = Low; then probability of selection is <50% {all set of features which have been given a priority value = low}
While Dataset folds are not empty Do
With selected fold Do
Use the selected feature set to evaluate all records in the fold
Check F1 ← Calculate F1
Based on the priority of features set selected and the results of F-Measure; update priority may occur, Based on the following criteria.
Update priority value for selected feature and set to Medium ← Priority = high & F-Measure is <90% and >= 80%.
Update priority value for selected feature and set to Low ← Priority = high & F-Measure is <80%
Update priority value for selected feature and set to Low ← Priority = Medium & F-Measure is<80%
Update priority value for selected feature and set to High ← Priority = Medium & F-Measure is>=90%
Update priority value for selected feature and set to Medium ← Priority = Low & F-Measure is >=80% and <=90
Update priority value for selected feature and set to High ← Priority = Low & F-Measure is >=90%.
Otherwise; NO update.
End of Fold
Step 2: continue testing; Select random features set based on priority value; calculate F1 at the end of each fold.
End While

5 Datasets

In this research experiment, the authors use two well-known datasets. The first dataset is the UNSW-NB15. UNSW-NB15 dataset is a comprehensive dataset constructed by Moustafa [21]. The list of features of UNSW-NB15 is listed in Tab. 1 below, and the number of records in the training set is 175,341 records, and the testing set is 82332 records.

The second dataset used in this experiment is the NSL-KDD [22]. NSL-KDD is an enhanced version of the KDD'99 dataset [23]. NSL-KDD does not include redundant records in the training set. The number of features in NSL-KDD contains 38 features (symbolic features are discarded). Tab. 2 below presents a list of features in the NSL-KDD dataset. Also, The number of records in the training "KDDTrain+" file is 125974, and the number of records in the testing "KDDTest+" file is 22545 records.

Table 1: List of features in training and testing UNSW-NB15 dataset

F no	F name	F no	F name	F no	F name
1	id	16	dloss	31	Response_body_len
2	dur	17	sinpkt	32	Ct_srv_src
3	proto	18	dinpkt	33	Ct_state_ttl
4	service	19	sjit	34	Ct_dst_ltm
5	state	20	djit	35	Ct_src_dport_ltm
6	spkts	21	swin	36	Ct_dst_sport_ltm
7	dpkts	22	stcpb	37	Ct_dst_src_ltm
8	sbytes	23	dtcpb	38	Is_ftp_login
9	dbytes	24	dwin	39	Ct_ftp_cmd
10	rate	25	tcprtt	40	Ct_flw_http_mthd
11	sttl	26	synack	41	Ct_src_ltm __
12	dttl	27	ackdat	42	Ct_srv_dst
13	sload	28	smean	43	Is_sm_ips_ports
14	dload	29	dmean	44	Attack_cat
15	sloss	30	Trans_depth	45	label

Table 2: List of features NSL-KDD dataset

F-no	F-name	F-no	F-name	F-no	F-name
1	duration	15	su_attempted	29	same_srv_rate
2	protocol_type	16	num_root	30	diff_srv_rate
3	service	17	num_file_creations	31	srv_diff_host_rate
4	flag	18	num_shells	32	dst_host_count
5	src_bytes	19	num_access_files	33	dst_host_srv_count
6	dst_bytes	20	num_outbound_cmds	34	dst_host_same_srv_rate
7	land	21	is_host_login	35	dst_host_diff_srv_rate
8	wrong_fragment	22	is_guest_login	36	dst_host_same_src_port_rate
9	urgent	23	count	37	dst_host_srv_diff_host_rate
10	hot	24	srv_count	38	dst_host_serror_rate
11	num_failed_logins	25	serror_rate	39	dst_host_srv_serror_rate
12	logged_in	26	srv_serror_rate	40	dst_host_rerror_rate
13	num_compromised	27	rerror_rate	41	dst_host_srv_rerror_rate
14	root_shell	28	srv_rerror_rate	42	class

6 Experiments and Results

In the following section, this study will present the evaluation metrics and results related to this research. All experiments were done using Dell Machine, Intel(R), Core i7-CPU 1.8 GHz, installed

memory (RAM) 16 GB, 64 Bit Operating System, Windows10. The Anaconda Python open source is used to run the experiments

6.1 Evaluation Metrics

In this part, this research work will demonstrate the evaluation metrics used in this research. For testing the evaluation and performance, the authors used regular criteria, such as True Positive (TP), False Positive (FP), False Negative, Precision (P), Recall (R), and F-Measure. To demonstrate these standard criteria, please see Tab. 3 below.

Table 3: Confusion matrix

		Predicted	
		Normal	Attack
Actual	Normal	a (TP)	b (FN)
	Attack	c (FP)	d (TN)

TPR (True Positive Rate): Quantity of normal data identified as normal.

FPR (False Positive Rate): Quantity of attack identified as normal.

FNR (False Negative Rate): Quantity of normal identified as attack

Precision: Ratio of the numbers of decisions that are correct.

Recall: Ratio of total relevant results correctly classified.

F- Measure: Testing the level of accuracy.

TPR= a/(a + b)	1
FPR=c/(c + d)	2
FNR=b/(a + b)	3
Precision = TP/(TP + FP)	4
Recall or (Sensitivity) = TP/(TP + FN)	5
F-Measure= 2 ∗ Precision ∗ Recall/(Precision + Recall)	6

6.2 Experiments Results Using UNSW-NB15 Dataset

Layer 1 experiment: training of the proposed model conducted as follows: the training dataset consists of 175,341 records, and the PSO, GWO, and FFA are used to generate the set of features. Each algorithm is run independently with Anaconda Python open-source, and each algorithm is used to create 50 feature sets. Each feature set will have a number, such as PSO-FS1, PSO-FS2 . . . . PSO-FS50, GWO-FS1, GWO-FS2 . . . GWO-FS50, FFA-FS1, FFA-FS2 . . . FFA-FS50. The results of this phase are 150 features set. For example:

PSO-FS_1 is consisting from 24 features= {f2, f4, f5, f7, f11, f12, f16, f17, f18, f19, f20, f22, f23, f24, f25, f26, f28, f30, f31, f34, f39, f40, f41, f42}.

GWO-FS_1 is consisting from 19 features= {f1, f4, f5, f6, f13, f16, f17, f22, f23, f26, f28, f29, f34, f36, f37, f38, f40, f41, f42}.

FFA-FS1 is consisting from 21 features = {f1, f2, f3, f6, f8, f9, f10, f11, f12, f13, f16, f19, f26, f28, f31, f32, f34, f35, f37, f41, f43}

Then using optimized GA, the training phase is started with the 150 feature set. Since the training dataset consists of 175,341 records, the training dataset is divided into 175 folds. Each fold consists of 1000 records except for the last fold, which consists of 1341 records. Optimized GA is used to select one feature set, and then training is conducted with the selected fold. At the end of training the fold, F-Measure is calculated. Based on the F-Measure value, a priority value will be assigned to the selected feature set. By the end of layer 1 experiments, each feature set will have a priority value assigned. Priority value will be assigned according to the following criteria.

Priority = *High*← F-Measure is >=90%

Priority = *Medium* ← F-Measure is >=80% and <90%

Priority = *Low*← F-Measure is <80%

Layer 2 experiment: testing of the proposed model conducted as follows: the testing dataset consists of 82332, and the dataset is split into 82 folds, and each fold consists of 1000 records except for the last fold, which consists of 1332. Optimized GA is used to select one feature set from the 150 sets. The selection of the feature set will be according to the priority value. The criteria for the selection of feature set is according to the following:

Probability of selection is >=75% ←*Priority* = ***High***

Probability of selection is >=50% and less <75% ← *Priority* = ***Medium***

Probability of selection is <50% ← *Priority* = ***Low***

The selected feature set is used with the 1000 record, and the F-Measure is calculated at the end of the fold. For any reason, if the F-Measure value is not compatible with the priority value, an update for the priority will occur. Updating the features set priority will be according to the following criteria:

- Update priority value for a selected feature and set it to *Medium* if the features set priority = *High* and F-Measure is <90% and >=80%.
- Update priority value for a selected feature and set it to *low* if the features set priority = *High* and F-Measure is <80%
- Update priority value for a selected feature and set it to *Low* if the features set priority = *Medium* and F-Measure is <80%
- Update priority value for a selected feature and set it to *High* if the features set priority = *Medium* and F-Measure is >=90%
- Update priority value for a selected feature and set it to *Medium* if the features set priority = *Low* and F-Measure is >=80% and <=90
- Update priority value for a selected feature and set it to *High* if the features set priority = *Low* and F-Measure is >=90%.

For experiment purposes, this study will demonstrate only the first 5 folds and the last 5 folds; Tab. 4 below will present experimental results using the J48 classifier and UNSW-NB15 dataset.

Table 4: UNSW-NB15 dataset/J48

Experiment	TPR	FPR	FNR	Precision	Recall	F-Measure
Fold 1	0.854	0.156	0.143	0.846	0.857	0.851
Fold 2	0.869	0.169	0.156	0.837	0.848	0.842
Fold 3	0.854	0.173	0.161	0.832	0.841	0.836
Fold 4	0.871	0.152	0.172	0.851	0.835	0.843
Fold 5	0.881	0.141	0.181	0.862	0.830	0.845
Fold ...						
Fold 78	0.856	0.132	0.122	0.866	0.875	0.871
Fold 79	0.871	0.123	0.147	0.876	0.856	0.866
Fold 80	0.881	0.147	0.159	0.857	0.847	0.852
Fold 81	0.891	0.134	0.149	0.869	0.857	0.863
Fold 82	0.903	0.149	0.143	0.858	0.863	0.861

Tab. 5 below will show experimental results using the Random Forest (RF) classifier. This study will demonstrate the results of the first 5 folds and the last 5 folds.

Table 5: UNSW-NB15 dataset/Random Forest

Experiment	TPR	FPR	FNR	Precision	Recall	F-Measure
Fold 1	0.856	0.152	0.143	0.849	0.857	0.853
Fold 2	0.871	0.174	0.156	0.833	0.848	0.841
Fold 3	0.864	0.172	0.163	0.834	0.841	0.838
Fold 4	0.861	0.153	0.174	0.849	0.832	0.840
Fold 5	0.891	0.151	0.185	0.855	0.828	0.841
Fold ...						
Fold 78	0.857	0.147	0.111	0.854	0.885	0.869
Fold 79	0.871	0.132	0.123	0.868	0.876	0.872
Fold 80	0.889	0.146	0.149	0.859	0.856	0.858
Fold 81	0.841	0.139	0.143	0.858	0.855	0.856
Fold 82	0.913	0.139	0.151	0.868	0.858	0.863

Tabs. 6 and 7 below will shed light on features set priority values after the training phase and the number of features set that have changed their priority values during testing with the UNSW-NB15 dataset.

6.3 Experiments Results Using NSL-KDD Dataset

All the steps done with the UNSW-NB15 dataset are repeated with the NSL-KDD dataset. Training is done following similar steps as the UNSW-NB15 dataset.

Layer 1 experiment: Optimized GA is used in the training phase to select features set and to give each feature set a priority value. By the end of the training phase, each feature set will have a priority

value assigned to it. Since the training dataset consists of 125974 records, the training dataset is divided into 125 folds, and each fold consists of 1000 records except the last fold, which consists of 1974.

Table 6: Priority values for features set/UNSW-NB15 dataset

No of feature set	Low priority	Medium priority	High priority
119	×	✓	×
19	×	×	✓
12	✓	×	×
Total (150)			

Table 7: Feature set changed priority value/UNSW-NB15 dataset

No of features set	Modifications
3	Changed their priority from medium to high
4	Changed their priority from high to medium
1	Changed their priority from medium to low

Layer 2 experiment: the testing dataset consists of 22545, and the testing dataset is split into 22 folds, and each fold consists of 1000 records except for the last fold, which consists of 1545 records.

For experiment purposes, this study will demonstrate only the first 5 folds and the last 5 folds, and Tab. 8 below will show experimental results using the J48 classifier and NSL-KDD dataset.

Table 8: NSL-KDD dataset/J48

Experiment	TP rate	FP rate	FN rate	Precision	Recall	F-Measure
Fold 1	0.901	0.123	0.132	0.880	0.872	0.876
Fold 2	0.891	0.147	0.133	0.858	0.870	0.864
Fold 3	0.887	0.121	0.143	0.880	0.861	0.870
Fold 4	0.899	0.113	0.154	0.888	0.854	0.871
Fold 5	0.897	0.147	0.147	0.859	0.859	0.859
Fold ...						
Fold 18	0.951	0.091	0.098	0.913	0.907	0.910
Fold 19	0.956	0.089	0.087	0.915	0.917	0.916
Fold 20	0.964	0.087	0.078	0.917	0.925	0.921
Fold 21	0.956	0.056	0.091	0.945	0.913	0.929
Fold 22	0.971	0.054	0.087	0.947	0.917	0.932

Tab. 9 below will show experimental results using the Random Forest (RF) classifier. This study will demonstrate the results of the first 5 folds and the last 5 folds.

Table 9: NSL-KDD dataset/Random Forest

Experiment	TPR	FPR	FNR	Precision	Recall	F-Measure
Fold 1	0.889	0.132	0.132	0.871	0.871	0.871
Fold 2	0.874	0.123	0.123	0.877	0.877	0.877
Fold 3	0.897	0.135	0.134	0.869	0.870	0.870
Fold 4	0.891	0.142	0.124	0.863	0.878	0.870
Fold 5	0.852	0.129	0.147	0.869	0.853	0.861
Fold ...						
Fold 18	0.941	0.041	0.109	0.958	0.896	0.926
Fold 19	0.943	0.042	0.121	0.957	0.886	0.920
Fold 20	0.948	0.054	0.109	0.946	0.897	0.921
Fold 21	0.941	0.061	0.113	0.939	0.893	0.915
Fold 22	0.938	0.077	0.101	0.924	0.903	0.913

Tabs. 10 and 11 below will shed light on features set priority values after the testing phase and the number of features set that have changed their priority values during testing with the NSL-KDD dataset.

Table 10: Priority values for features set/NSL-KDD dataset

No of feature set	Low priority	Medium priority	High priority
115	×	✓	×
27	×	×	✓
8	✓	×	×
Total (150)			

Table 11: Feature set which changed priority value/NSL-KDD dataset

No of feature set	Modifications
4	Changed their priority from medium to high
4	Changed their priority from high to medium
0	Changed their priority from medium to low

As shown in Tabs. 4 and 5 above, results show that the effects of optimized GA are exposed. True positive (TP), False Positive (FP), False Negative (FN), Precision (P), and Recall (R) performance are enhanced as the number of tested folds is increased. Besides, experiments with the UNSW-NB15 dataset performance are not equivalent to experiments with the NSL-KDD dataset since the UNSW-NB15 dataset has more attributes and more attack classes.

Tabs. 8 and 9 above demonstrate the effects of optimized GA with the NSL-KDD dataset. As shown in Tabs. 8 and 9, results indicate that optimized GA has a positive impact, especially with the last folds, which means that GA can adapt itself with time.

Fig. 6 below shows F-Measure using the two datasets. The study will demonstrate the results of the first 5 folds and the last 5 folds only.

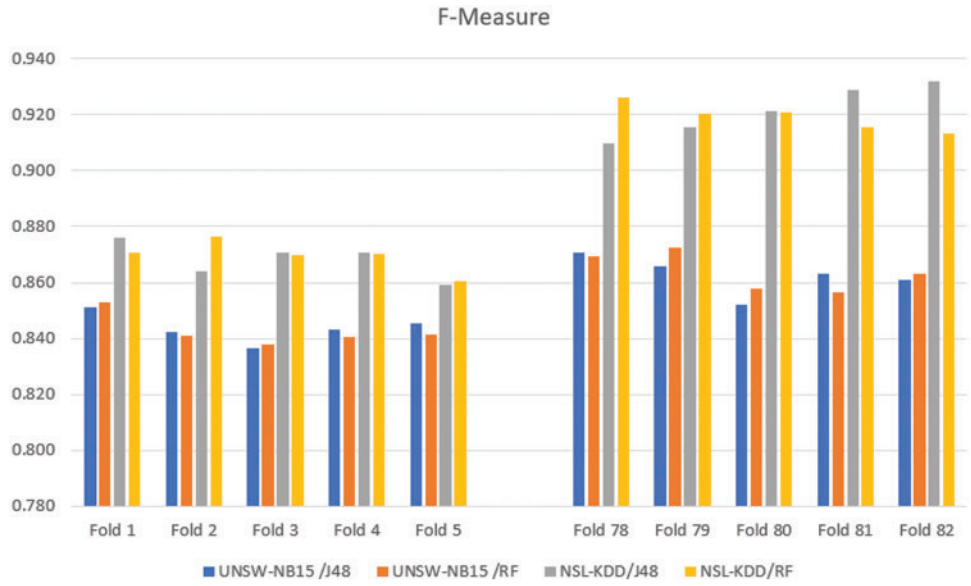

Figure 6: F-Measure using optimized GA

Demonstrated results in Tabs. 4, 5, 8, and 9 show that the multilayer bio-inspired feature selection model for intrusion detection using an optimized genetic algorithm is highly acceptable and recommended. Furthermore, Fig. 6 above indicates that F-Measure accuracy performance is noted. Also, results demonstrate that NSL-KDD dataset results show higher F-Measure values compared with UNSW-NB15 since UNSW-NB15 has more attributes and attack types.

7 Conclusion and Future Works

Intrusion detection systems are an emergent topic, and the increased number of attacks around the world increases the need for protecting our machines. This research proposes a new multilayer bio-inspired feature selection model for intrusion detection using an optimized genetic algorithm. The proposed model consists of two layers. **At layer 1**, a set of features is generated using three well-known metaheuristic algorithms. Algorithms used at layers 1 are (PSO), (GWO), and (FFA). These algorithms are used to create a set of features that will be used later by an Optimized Genetic algorithm (GA) in a new manner. At layer 1, optimized GA is used to assign a priority value for each feature set in the training phase. The optimized GA is used to assign a priority value for each feature set based on the F-Measure calculated at the end of the fold. By the end of the training phase and with the help of optimized GA, each feature set will have a priority assigned. **At layer 2**, optimized GA is used with the testing dataset. In this phase, any feature set will be selected according to its priority value. At the end of each fold, the F-Measure is calculated. Based on F-Measure, an update may occur on the priority value assigned to the selected feature set. Optimized GA's modifications were conducted to guarantee that optimized GA will continually adapt and learn based on the types of data received. Several criteria

are used to measure the performance of the proposed model. Several criteria are used, such as true positive, false positive, false negative, precision, recall, and F-Measure. Besides, F-Measure is used as a reference for assigning priority values for all feature sets. Furthermore, Two benchmark datasets are used in the proposed work: the NSL-KDD and the UNSW-NB15. The overall results related to precision and recall are promoted. Results with the NSL-KDD dataset are better than the UNSW-NB15 since the UNSW-NB15 has more attributes and attack types. Future work for the authors or other researchers could be evaluating the proposed model in the real world with the actual kind of attacks and checking how optimized GA will be adapted.

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

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] D. Massa and R. Valverde, "A fraud detection system based on anomaly intrusion detection systems for E-commerce applications," *Computer and Information Science*, vol. 7, no. 2, pp. 117–140, 2014.
[2] B. Luo and J. Xia, "A novel intrusion detection system based on feature generation with visualization strategy," *Expert Systems with Applications*, vol. 41, no. 9, pp. 4139–4147, 2014.
[3] Y. Chung and N. Wahid, "A hybrid network intrusion detection system using simplified swarm optimization (SSO)," *Applied Soft Computing*, vol. 12, no. 9, pp. 3014–3022, 2012.
[4] B. Luo and J. Xia, "A novel intrusion detection system based on feature generation with visualization strategy," *Expert System with Application*, vol. 41, no. 9, pp. 4139–4147, 2014.
[5] A. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.
[6] J. Khan and N. Jain, "A survey on intrusion detection systems and classification techniques," *International Journal of Scientific Research in Science, Engineering and Technology*, vol. 2, no. 5, pp. 202–208, 2016.
[7] O. Almomani, "A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA algorithms," *Symmetry*, vol. 12, no. 1046, pp. 1–20, 2020.
[8] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai *et al.,* "An efficient intrusion detection system based on support vector machines and gradually feature removal method," *Expert Systems with Applications*, vol. 39, no. 1, pp. 424–430, 2012.
[9] I. Sarker, Y. Abushark, F. Alsolami and A. Khan, "IntruDTree : A machine learning based cyber security intrusion detection model," *Symmetry*, vol. 12, no. 14, pp. 1–15, 2020.
[10] B. Morel, "Artificial intelligence and the future of cybersecurity," in *Proc. 4th ACM Workshop on Security and Artificial Intelligence*, Chicago, Illinois, USA, pp. 93–98, 2011.
[11] M. Blowers and J. Williams, "Machine learning applied to cyber operations," *Network Science and Cybersecurity*, vol. 55, pp. 155–175, 2014.
[12] A. Abraham, C. Grosan and C. Vide, "Evolutionary design of intrusion detection programs," *International Journal of Network Security*, vol. 4, no. 3, pp. 328–339, 2007.
[13] V. Hashemi, Z. Muda and W. Yassin, "Improving intrusion detection using genetic algorithm," *Information Technology Journal*, vol. 12, no. 11, pp. 2167–2173, 2013.
[14] R. Rahmani, M. Othman, A. Shojaei and R. Yusof, "Static VAR compensator using recurrent neural network," *Electrical Engineering*, vol. 96, no. 2, pp. 109–119, 2014.
[15] M. Sheikhan, Z. Jadidi and A. Farrokhi, "Intrusion detection using reduced-size RNN based on feature grouping," *Neural Computing and Applications*, vol. 21, no. 6, pp. 1185–1190, 2012.
[16] R. Ashfaq, X. Wang, J. Huang, H. Abbas and Y. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Information Science*, vol. 378, no. 1, pp. 484–497, 2017.
[17] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[18] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Network*, vol. 61, no. 3, pp. 85–117, 2015.
[19] W. Li, P. Yi, Y. Wu, L. Pan and J. Li, "A new intrusion detection system based on KNN classification algorithm in wireless sensor network," *Journal of Electrical and Computer Engineering*, vol. 2014, no. 5, pp. 1–8, 2014.
[20] N. Farnaaz and M. Jabbar, "Random forest modeling for network intrusion detection system," *Procedia Computer Science*, vol. 89, no. 1, pp. 213–217, 2016.
[21] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *IEEE In Proc. of the 2015 Military Communications and Information Systems Conf. (MilCIS)*, Canberra, Australia, pp. 1–6, 2015.
[22] M. Tavallaee, E. Bagheri, W. Lu and A. Ghorbani, "A detailed analysis of the KDD CUP 99 data Set," in *Second IEEE Symp. on Computational Intelligence for Security and Defense Applications (CISDA)*, Ottawa, ON, Canada, pp. 1–6, 2009.
[23] S. Hettich and S. Bay, "KDD cup 1999 data set," University of California Irvine, KDD repository, 1999. [Online]. Available: http://kdd.ics.uci.edu.
[24] C. Yin, Y. Zhu, J. Fei and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.
[25] B. AShahri, R. Rahmani, M. Chizari, A. Maralani, M. Eslami *et al.,* "A hybrid method consisting of GA and SVM for intrusion detection system," *Neural Computing and Applications*, vol. 27, no. 6, pp. 1669–1676, 2016.
[26] G. Wang, J. Hao, J. Ma and L. Huang, "A new approach to intrusion detection using artificial neural networks and fuzzy clustering," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6225–6232, 2010.
[27] U. Çavuşoğlu, "A new hybrid approach for intrusion detection using machine learning methods," *Applied Intelligence*, vol. 49, no. 7, pp. 2735–2761, 2019.
[28] W. Chen, S. Hsu and H. Shen, "Application of SVM and ANN for intrusion detection," *Computers & Operations Research*, vol. 32, no. 10, pp. 2617–2634, 2005.
[29] S. Aljawarneh, M. Yassein and M. Aljundi, "An enhanced J48 classification algorithm for the anomaly intrusion detection systems," *Cluster Computing*, vol. 22, no. S5, pp. 10549–10565, 2019.
[30] S. Chen, M. Peng, H. Xiong and X. Yu, "SVM intrusion detection model based on compressed sampling," *Journal of Electrical and Computer Engineering*, vol. 2016, pp. 1–6, 2016.
[31] A. Haider, M. Khan, A. Rehman, M. Rahman and H. Kim, "A real-time sequential deep extreme learning machine cybersecurity intrusion detection system," *Computers, Materials & Continua*, vol. 66, no. 2, pp. 1785–1798, 2020.
[32] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. of ICNN'95-Int. Conf. on Neural Networks*, Perth, Australia, 4, pp. 1941–1948, 1995.
[33] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. of IEEE Int. Conf. on Evolutionary Computation*, Anchorage, AK, USA, pp. 69–73, 1998.
[34] F. Marini and B. Walczak, "Particle swarm optimization (PSO). A tutorial," *Chemometrics and Intelligent Laboratory Systems*, vol. 149, pp. 153–165, 2015.
[35] S. Mirjalili, S. M. Mirjalili and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
[36] Z. Cui and X. Gao, "Theory and applications of swarm intelligence," *Neural Computing and Applications*, vol. 21, no. 2, pp. 205–206, 2012.
[37] Z. Zhang, K. Long, J. Wang and F. Dressler, "On swarm intelligence inspired self-organized networking: Its bionic mechanisms, designing principles and optimization approaches," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 513–537, 2014.
[38] R. Parpinelli and H. Lopes, "New inspirations in swarm intelligence: A survey," *International Journal of Bio-Inspired Computation*, vol. 3, no. 1, pp. 1–16, 2011.

[39] X. Yang, "Etaheuristic optimization: Nature-inspired algorithms and applications," in *Artificial Intelligence, Evolutionary Computing and Metaheuristics*. Vol. 427. Berlin: Springer, Part of the Studies in Computational Intelligence Book Series, 2013.
[40] X. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic Algorithms: Foundations and Applications*. Vol. 5792. Berlin: Springer, Part of the Lecture Notes in Computer Science Book, 2009.
[41] A. Ritthipakdee, A. Thammano, N. Premasathian and D. Jitkongchuen, "Firefly mating algorithm for continuous optimization problems," *Computational Intelligence and Neuroscience*, vol. 2017, pp. 1–10, 2017.
[42] X. Yang, "Firefly algorithm, stochastic test functions and design optimization," *International Journal of Bio-Inspired Computation*, vol. 2, no. 2, pp. 77–84, 2010.
[43] X. Yang and X. He, "Firefly algorithm: Recent advances and applications," *International Journal of Swarm Intelligence*, vol. 1, no. 1, pp. 36–50, 2013.
[44] S. Katoch, S. Chauhan and V. Kumar, "A review on genetic algorithm: Past, present, and future," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 8091–8126, 2021.
[45] K. Dhal, S. Ray, A. Das and S. Das, "A survey on nature-inspired optimization algorithms and their application in image enhancement domain," *Archives of Computational Methods in Engineering*, vol. 26, no. 5, pp. 1607–1638, 2019.
[46] C. Lee, "A review of applications of genetic algorithms in operations management," *Engineering Application of Artificial Intelligence*, vol. 76, no. 2, pp. 1–12, 2018.
[47] D. Whitley, "An executable model of a simple genetic algorithm," *Foundations of Genetic Algorithm*, vol. 2, pp. 45–62, 1993.
[48] T. Vidal, T. Crainic, M. Gendreau and C. Prins, "A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows," *Journal of Computers & Operations Research*, vol. 40, no. 1, pp. 475–489, 2013.
[49] K. Das, "Hybrid genetic algorithm: An optimization tool," in *Global Trends in Intelligent Computing Research and Development*. Pennsylvania: IGI global, pp. 268–305, 2013.
[50] X. Zhang, L. Jia, H. Shi, Z. Tang and X. Wang, "The application of machine learning methods to intrusion detection," *IEEE Spring Congress on Engineering and Technology*, vol. 2012, pp. 1–4, 2012.
[51] S. Salzberg, "C4.5: Programs for machine learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993," *Machine Learning*, vol. 16, pp. 235–240, 1994.
[52] A. Hamdan, "Intrusion detection using a new hybrid feature selection model," *Intelligent Automation & Soft Computing*, vol. 30, no. 1, pp. 65–80, 2021.
[53] M. Nasir, A. Javed, M. Tariq, M. Asim and T. Baker, "Feature engineering and deep learning-based intrusion detection framework for securing edge IoT," *The Journal of Supercomputing*, vol. 78, no. 6, pp. 8852–8866, 2022.
[54] O. Almomani, "A hybrid model using bio-inspired metaheuristic algorithms for network intrusion detection system," *Computers, Materials and Continua*, vol. 68, no. 1, pp. 409–429, 2021.
[55] O. Almomani, M. Almaiah, A. Alsaaidah, S. Smadi, A. Hamdan *et al.,* "Machine learning classifiers for network intrusion detection system: comparative study," in *Int. Conf. on Information Technolgy*, Amman, Jordan, pp. 440–445, 2020.