Received: December 17, 2020. Revised: January 29, 2021.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Grey Wolf Optimization Parameter Control for Feature Selection in Anomaly Detection

Hussein Almazini[1]* **Ku Ruhana Ku-Mahamud[1]**

[1]*School of Computing, University Utara Malaysia, Kedah, Malaysia*
* Corresponding author's Email: h.almazni22@gmail.com

Abstract: The performance of different mechanisms utilised to perform anomaly detection depends heavily on the group of features used. Therefore, dealing with a multi-dimensional dataset that typically contains a large number of attributes has caused problems to classification accuracy. Not all attributes in the dataset can be used in the classification process since some features may lead to low performance of classifiers. Feature selection (FS) is a good mechanism that minimises the dimension of high-dimensional datasets. Modified binary grey wolf optimization (MBGWO) is a metaheuristic algorithm that has been successfully used for FS. However, the MBGWO algorithm has a drawback in selecting sub-optimal feature sets from an original set of features. This drawback is related to the linearly decreasing value of a parameter where there is no control between the exploration and exploitation processes. This study proposed an enhanced binary grey wolf optimization (EBGWO) algorithm for FS in anomaly detection by controlling the balancing parameter. The new method focused on obtaining a value for a parameter that controlled the trade-off between exploration and exploitation. Evaluation of the proposed method was on the NLS-KDD dataset with different attack classes and compared with other benchmark algorithms, such as binary bat algorithm, binary particle swarm optimization, and four variants of grey wolf optimiser for FS. The experimental results indicated that EBGWO was superior than other algorithms, where it obtained 19 features only out of a total of 41 features with 87.46% classification accuracy. The proposed algorithm can be applied to detect anomaly in network intrusion and outliers in data that are significant but difficult to find.

Keywords: Metaheuristic, Grey wolf optimization, Feature selection, Classification, Anomaly detection.

1. Introduction

Anomaly detection is the activity to detect known anomalies or attacks in information systems [1]. Anomaly detection systems work by observing the behavior of the full system, traffic, or objects and then comparing it against normal or predefined assumed behavior s. Any split from the standard behavior is considered as a possible attack [2]. Due to the difficulty in distinguishing different types of attacks in high-dimensional data, anomaly detection becomes the preferred approach [1]. Attackers have become competent to manufacture malware, which has the power to change their structure (polymorphism). Anomaly detection depends on machine learning to discover irregularities [3]. The core principle of machine learning that incorporates artificial intelligence is to allow a machine to learn on its own, thereby distinguishing abnormal and normal behavior s within the system. The performance of different mechanisms, utilised to accomplish anomaly detection within systems, depends greatly on the group of features utilised. A feature collection must supply a functional uniqueness through the existing classes within the data, permitting fast and good classification of those classes [4]. Redundant attributes may be present within the vector that are irrelevant in detecting anomaly [5–7]. Classification accuracy and increase in detection time are two things that will be affected when performing anomaly detection on irrelevant and redundant features [6, 8, 9]. The irrelevant attribute does not participate in the detection of any particular kind of attacks since it is incapable of holding important features for attack

detection [10, 11]. According to Kayacik [7], redundant and irrelevant attributes in the data will produce a small amount of knowledge concerning any of the classes within the dataset. Redundant attributes and irrelevant attributes/features will not provide any additional knowledge [4, 12, 13]. Feature selection (FS) is a procedure that removes irrelevant, redundant, or noisy data, and detects relevant features [14]. Feature selection improves predictive accuracy, increases comprehensibility, and speeds up data mining algorithms. Using FS mechanisms on the data before the features are analysed by the anomaly detection system can produce better detection accuracy [13]. Feature selection used in anomaly detection typically involves the use of supervised algorithms that require access to labelled data [8], [15–17].

Recent years have witnessed the use of metaheuristic algorithms such as ant colony optimization (ACO) in solving data mining problems for health care system [18], data clustering [19, 20], classification [21, 22] and travelling salesman issues [23]. Solving FS problems using metaheuristic algorithms is popular because near-optimal solutions could be obtained [24–28]. For example, Zhang [29] extended a bare-bone particle swarm optimization (PSO) algorithm for FS with binary variables to select the optimal features, which was named binary particle swarm optimization. Later, a new algorithm based on ACO and enriched with a new function as proposed by [30] produced better feature subsets. Bio-inspired optimization techniques were employed because of their robustness, simplicity, and efficiency to resolve complex optimization problems [31]. Grey wolf optimiser was first suggested by Mirjalili [32] to solve data extraction problems, which is a process in classification and FS [33]. One of the modern metaheuristic algorithms that have been successfully used for FS in terms of anomaly detection is modified binary grey wolf optimiser (MBGWO) [34]. However, MBGWO has a problem with the exploration and exploitation processes, which makes it insufficient in terms of finding a good quality solution. The behavior of the exploration and exploitation parameters is influenced by a value that decreases linearly from 2 to 0. This makes the search process become very limited because each wolf will have the same value in producing its solution [35]. The work in this study is implemented through two phases. Firstly, the decrease and increase strategy is applied to control the parameter of MBGWO by obtaining a value for a parameter that is utilised to control the trade-off between exploration and exploitation. Secondly, support vector machine is used for the classification process. The method based on indicators from the search process in determining the value for the parameter that influences the behavior of the exploration and exploitation processes to find the optimal features has resulted in better performance of the algorithm. This is the advantage of the proposed algorithm for feature selection.

The remaining parts of this paper is organised as follows. Sections 2 and 3 present the grey wolf optimiser (GWO) and the new method, respectively. The data and experimental design are presented in Section 4. Results of the experiment are presented in Section 5. Lastly, Section 6 summarises the conclusions and future directions of the study.

2. Grey wolf optimizer

The GWO algorithm begins the optimization operation with a collection of random positions where each position is kept in a vector. In every repetition, the first phase computes the fitness value of every position of the alpha, beta, and delta wolves. There are three vectors and three variables to keep the locations and fitness values of the wolves. The three vectors, *alpha*, *beta*, and *delta* wolves must be updated before the location updating process. For the updated wolf location, the distance between the three wolves/agents and the current solution should be calculated first. New locations for the wolves are calculated based on the three best locations as follows [35]:

$$\vec{\mathrm{X}}_{(\mathrm{t}+1)} = \frac{\vec{\mathrm{X}}_1+\vec{\mathrm{X}}_2+\vec{\mathrm{X}}_3}{3} \tag{1}$$

where $\vec{\mathrm{X}}_1, \vec{\mathrm{X}}_2, \vec{\mathrm{X}}_3$ are defined as:

$$\begin{aligned} \vec{\mathrm{X}}_1 &= |\vec{\mathrm{X}}_\alpha - \vec{\mathrm{A}}_1.\vec{\mathrm{D}}_\alpha| \\ \vec{\mathrm{X}}_2 &= |\vec{\mathrm{X}}_\beta - \vec{\mathrm{A}}_2.\vec{\mathrm{D}}_\beta| \\ \vec{\mathrm{X}}_3 &= |\vec{\mathrm{X}}_\delta - \vec{\mathrm{A}}_3.\vec{\mathrm{D}}_\delta| \end{aligned} \tag{2}$$

$$\begin{aligned} \vec{\mathrm{D}}_\alpha &= |\vec{\mathrm{C}}_1.\vec{\mathrm{X}}_\alpha - \vec{\mathrm{X}}| \\ \vec{\mathrm{D}}_\beta &= |\vec{\mathrm{C}}_2.\vec{\mathrm{X}}_\beta - \vec{\mathrm{X}}| \\ \vec{\mathrm{D}}_\delta &= |\vec{\mathrm{C}}_3.\vec{\mathrm{X}}_\delta - \vec{\mathrm{X}}| \end{aligned} \tag{3}$$

The variables, $\vec{X}_\alpha, \vec{X}_\beta, \vec{X}_\delta$ represent three best positions at iteration t, $\vec{A}_1, \vec{A}_2, \vec{A}_3$ and $\vec{C}_1, \vec{C}_2, \vec{C}_3$ are the coefficient vectors, which are computed as follows [32]:

$$\vec{\mathrm{A}} = 2\vec{\mathrm{a}}.\vec{\mathrm{r}}_1 - \vec{\mathrm{a}} \tag{4}$$

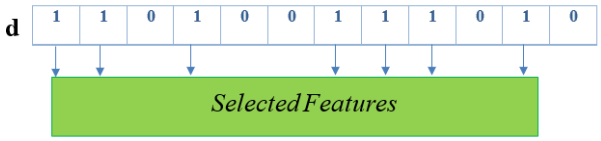


Figure 1. Solution representation

$$\vec{C} = 2\vec{r}_2 \quad (5)$$

where $\vec{a}$ will linearly decrease from 2 to 0 over the course of iterations and $\vec{r}_1$, $\vec{r}_2$ are the random vectors in [0, 1]. The equation for updating the value of the parameter (a) is given by [32]:

$$a=2-t(\frac{2}{T}) \quad (6)$$

Global search (exploration) in GWO is produced when A>1 or A< 1, while the local search (exploitation) is produced when -1<A<1. The value of A is dependent on the parameter (a) that linearly decreases from 2 to 0. As a result of the random mechanisms in this variable, the field changes in the range of [-2, 2] for variable A. Overall, as above, the balancing of exploration and exploitation is required to obtain the global optimum by applying a stochastic method.

Consequently, it is impossible to apply GWO to resolve FS issues without modification. There must be a process to switch the general algorithm into its binary version. There have been several binary versions suggested in the literature like crossover () [33], sigmoid() [36], and tanh() functions [37]. The major phase in explaining GWO for FS is to describe the feature subset represented in the solution. Fig. 1 shows the depiction of the features. The location of the solution may take a value of "1" or "0.", where if the amount is equal to 0, the attribute is not chosen. However, if the value equals to 1, the attribute is chosen. Therefore, the scope of the feature subset is expressed by a value of 1.

A new binary variant of GWO (bGWO) algorithm was suggested in [33] in finding an optimal subset of features. The bGWO algorithm uses two different approaches to find a new position for the wolf. In the beginning, the first three best wolves' movements are binarised and a stochastic crossover operation is applied to the binarised value. Later, a sigmoidal function is then applied to the result of the stochastic crossover operation. This concept in bGWO is used to find the best subset of features that can improve the classification accuracy. However, the proposed bGWO do not have any automatic tuning for the balancing parameter which led to insignificant features were obtained leading to low classification accuracy.

An optimization population-based technique called modified GWO is suggested for choosing the optimal set of services, whereby a crossover procedure of the genetic algorithm (GA) is combined with the GWO algorithm [38]. The combination process improves the searchability as each agent can share its information with other pack members. The suggested algorithm makes a good trade-off between the local and global searches. However, the quality of the solution (classification accuracy) is unlikely to be high because the balancing parameter do not have any automatic control.

The MBGWO algorithm that was presented in [34] for FS in anomaly detection generally attaches the best fourth wolf (*omega*) to share the position updates of the search agents. The effort decreases the effectiveness of the best three wolves' positions by adding the best fourth wolf to share the updating decision. This leads to improving the performance of the updating decision of best search agents in finding the optimal solution. In MBGWO, the value of the parameter, *a*, decreases linearly for all search agents, making the search process become very limited because each wolf will have the same value in producing its solution. However, the algorithm has several drawbacks in selecting sub-optimal features set from an original set of features.

In all the previous studies referred to, the value of parameter, (a), decrease linearly for all the search agents, that makes the search process very limited because each wolf will have the same value in producing its solution. In this study, we proposed a method to obtain a value for the parameter that control the trade-off between exploration and exploitation to avoid the limitation in the existing techniques and improve the EBGWO for FS in anomaly detection.

3. The proposed method

The proposed method focuses on obtaining a value for the parameter, *a*, which is utilised to obtain a good trade-off between exploration and exploitation. The proposed method is called enhanced binary GWO (EBGWO), where enhancement is made on the exploration and exploitation processes of the wolves in the MBGWO algorithm. This study formulated an approach based on indicators from the search process in determining the value for the parameter that influences the behavior of the exploration and exploitation processes in finding the optimal features. There are three main things to consider in implementing the

proposed method: i) The value of the parameter, *a*, will depend on the agent feedback that acts as an indicator in the search space. Therefore, the value will not necessarily linearly change. ii) Each agent will obtain an increment indicator for the parameter when it leaves an area with good fitness value to an area with worse fitness value. Conversely, when each agent leaves a worse area and moves to a good or better one, it will receive a decrement indicator for the parameter selection. However, the agent's fitness value is the indicator that determines whether this agent should increase or reduce the value of the parameter. iii) In order for each agent to adapt its exploration and exploitation values, there are two rewards, i.e. decrement and increment. The decrement reward usually occurs when the solution is improved in subsequent iterations. The achievement of iteration at time *t* is realised by the ability of the search wolf from the fitness value at time *t* that is more preferable than its fitness value at time *t*−1. This decreases its exploration rate and consequently decreases its confidence. The proposed equation to reduce the value of the parameter, *a*, from 1 to 0 is given by:

$$a=1-t(\frac{2}{T}) \quad (7)$$

where *t* is the number of iteration and *T* is the maximum number of iterations. In contrast, the increment reward is when the solution does not improve in subsequent iterations. This will lead to the increment of the parameter's value. Another status that may drive to such a reward is when a continuous worse fitness occurs in which the wolf may want to scout in a new area in the search space, hoping to obtain an improved solution. This increases its exploration rate and, thus, increases its confidence. The proposed equation to increase the value of the parameter, *a*, from 1 to 2 is as follows:

$$a=1+t(\frac{2}{T}) \quad (8)$$

where *t* represent the iteration number and *T* is the maximum number of iterations. The new method replaces the linearly changing process of the MBGWO algorithm to form the EBGWO algorithm. Fig. 2 shows the EBGWO algorithm where the new method is highlighted.

Initialise a population of wolves' positions at random [0; 1].
Initialise a, A, and C
Find the best solutions (alpha, beta, delta, and omega)
while stopping criteria not met do
For each Wolf
Calculate and find alpha, beta, delta, and omega.
Calculate the best position.
end
For each Wolf
Evaluate the location of individual wolves.
If fitness value at time t worse than fitness value at time t−1
Increase the parameter (a)
else
decrease the parameter (a)
end if
Update Parameters A and C.
Evaluate positions of individual wolves.
Update alpha, beta, delta, and omega
End while
Return best solution

Figure. 2 EMGWO algorithm

4. Data and experimental design

Evaluation of the proposed method was conducted where its performance was compared with four GWO variants and two commons algorithms. The performance metrics used for comparison were classification accuracy and number features in the subset. The NSL-KDD dataset, which is the modified version of the KDD99 dataset [39], was used in the performance evaluation. The NSL-KDD dataset was utilised because it is useful and effective in comparing several anomaly detections approaches. In the NSL-KDD dataset, there were 41 features suggested for each record. Every attack was classified under one of the roots: denial of service (DoS), probe, user to root attack (U2R), and remote to local (R2L). Fig. 3 displays the definition of every attack type.

Every NSL-KDD instance contained a network connection with 41 known features (e.g. type, service, flag, and protocol), that were labelled as normal or one of the four kinds of attack (e.g. probe, DoS, U2R, and R2L). Fig. 4 shows the NSL-KDD dataset class distribution, while Table 1 displays the distribution of attack types in the NSL-KDD dataset [34].

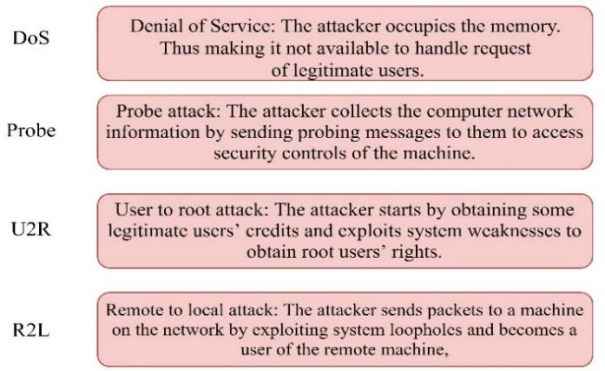
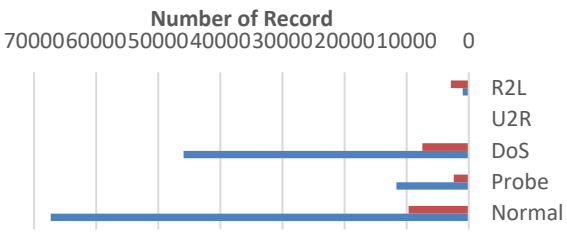


Figure. 3 Attack type definition



	Normal	Probe	DoS	U2R	R2L
Test+	9711	2421	7458	67	2887
Train+	67343	11656	45927	52	995

Figure. 4 NSL-KDD dataset class distribution

Table 1. Distribution of attack types in NSL-KDD

Attack	**Attack type**	**NSL-KDD**	
		Training data	**Testing data**
DoS	Neptune	8282	4657
	Smurf	529	665
	Pod	38	41
	Teardrop	188	12
	Land	1	7
	Back	196	359
	Apache2	-	717
	Udpstorm	-	2
	Process-table	-	685
	Mail-bomb	-	293
U2R	Bu_er-overow	6	20
	Load-module	1	2
	Perl	0	2
	Rootkit	4	13
	Spy	1	-
	Xterm	-	13
	Ps	-	17
	Http-tunnel	-	133
	Sql-attack	-	2
	Worm	-	2
	Snmp-guess	-	331
R2L	Guess-password	10	1231
	Ftp-write	1	3
	Imap	5	1
	Phf	2	2
	Multihop	2	18
	Warezmaster	7	944
	Warezclient	181	-
	Snmpgetattack	-	178
	Named	-	17
	Xlock	-	9
	Xsnoop	-	4
	Send-mail	-	14
Probe	Port-sweep	587	157
	IP-sweep	710	141
	Nmap	301	73
	Satan	691	735
	Saint	-	319
	Mscan	-	996

In this study, the data in NSL-KDD were divided into two datasets, namely KDDTrain+ and KDDTest+. The size for each dataset was 125,973 and 22,544, respectively. The ratio between the training data and testing data was 85:15. The first phase of the experiment compared the performance of the proposed method with several variants of GWO, i.e. modified GWO (MGWO) by Chandra [38], binary GWO (bGWO) by Emary [33], and MBGWO by Alzubi [34]. These algorithms were chosen for comparison because they use linearly decrease method for parameter, a, which make the search process limited in selecting the optimal features. The MGWO is a hybrid Algorithm while bGWO and MBGWO are non-hybrid algorithms similar to EBGWO. Moreover, these algorithms which are variants of GWO from the swarm intelligence family and performed well for FS. This is to show that better results can be obtained by EBGWO. The results were compared based on the average number of features (ANF) and the average accuracy performance (AAP), which are calculated as follows [33, 34, 40]:

$$AAP = \frac{\sum_{i=1}^{run} AC_i}{run}$$

$$ANF = \frac{\sum_{i=1}^{run} NF_i}{run}$$

The number of features is *NF*, and the accuracy rate is *AC*. Support vector machine was utilised as a classifier because it is one of the most commonly used classifiers for classification in anomaly detection [41, 42].

Fitness function is calculated for each algorithm as follows [33]:

$$Fitness = AC\,.a + (1/NF)\,.b,$$

where the number of features is *NF*, and a and b are two parameters in which $a \in [1,0]$ and $b = 1 - a$.

In the second phase of the experiment, the average attributes were selected and the average accuracy in EBGWO was compared with the binary PSO (bPSO) [43], bGWO [33], binary bat algorithm (BBA) [44], and MBGWO. In this phase, the dataset that was used was a combination of KDDTest-21 and KDDTrain+_20Percent [39]. KDDTrain+_20Percent was the subset of the training dataset, while KDDTest-21 was a subset of the test dataset. The total number of instances in this new dataset were 37,042. For the experiment, 80% of the combination data were randomly chosen as the training data, while the remainder 20% were used as the testing data.

Table 2. Parameters settings

Parameters	Unit/Description
Number of runs	20
Total number of wolves	12
Number of iterations	20
SVM classifier	RBF
a value	0.6
b value	0.4

Fitness function was again calculated for each algorithm. The anomaly detection metrics that were used to compare the performance were average accuracy (*AC*), number of selected features and error rate (*ER*) using the following equations [33, 34, 40]:

$$AC = \frac{TP+TN}{TP+TN+FP+FN}$$

$$ER = \frac{FP+FN}{TP+TN+FP+FN}$$

where *TN*, *TP*, *FN*, and *FP* are the confusion matrices that describe the classification results (true or false) [34]. The matrices are:

- *TN*: Correctly classified as normal
- *FN*: Intrusions that are classified as normal
- *TP*: Correctly classified as intrusion
- *FP*: Normal behavior but classified as intrusion

The error rate metric was used to test for the convergence of the algorithm [45, 46]. The experiment was conducted in MATLAB R2018b with parameter settings as shown in Table 2.

5. Results and discussion

The results of the first phase of the experiment in which the full dataset was used are shown in Table 3. The results exposed that EBGWO obtained the highest accuracy with the smallest number of features. The new method for the parameter selection value was able to choose the optimal subset of features by considering previous knowledge during the search process. In EBGWO, the choice of the parameter selection value (parameter control) from the feedback in the search space was considered. Whereas in all other GWO variants, the parameter selection value process was from the linear decreasing strategy without feedback from the search space. The EBGWO results emphasised that the parameter selection value had an effect in the decision-making process for the trade-off between exploration and exploitation.

Table 3. First phase results for feature selection

Algorithm	AAP	ANF
EBGWO	87.46%	19
MBGWO	81.58%	26
bGWO	81.07%	26
MGWO	79.66%	24
GWO	79.66%	28

Table 4. Comparison of average accuracy

Class	1	2	3	4	5
Normal	98.31 %	98.26 %	98.23 %	98.41 %	98.41 %
DoS	99.54 %	99.42 %	99.40 %	99.73 %	99.73 %
Probe	98.71 %	98.66 %	98.58 %	98.88 %	98.88 %
U2R	99.74 %	99.59 %	99.59 %	99.77 %	99.77 %
R2L	97.60 %	97.36 %	97.33 %	97.33 %	97.33 %

1.Proposed EBGWO Algorithm
2.MBGWO Algorithm
3.bGWO Algorithm
4.bPSO Algorithm
5.BBA Algorithm

Table 5. Comparison of average selected feature

Class	1	2	3	4	5
Normal	20	20	21	26	25
DoS	16	17	18	23	23
Probe	16	16	18	22	21
U2R	13	12	12	18	18
R2L	17	18	19	23	22

1.Proposed EBGWO Algorithm
2.MBGWO Algorithm
3.bGWO Algorithm
4.bPSO Algorithm
5.BBA Algorithm

The experimental outcomes of the second phase of the experiment are presented in Tables 4 and 5, which showed that the EBGWO algorithm outperformed other algorithms in terms of average subset of selected features and average accuracy. The balance between searching a significant number of features with the highest accuracy was obtained by the EBGWO algorithm. This reflected the advantage of the proposed method in obtaining the value for the parameter that controlled the trade-off between the exploration and exploitation processes. The results indicated that EBGWO obtained the highest accuracy with the smallest number of features. In summary, EBGWO selected fewer numbers of features that were useful for the detection procedure with highest accuracy.

The performance of EBGWO and MBGWO in terms of their convergence for five classes; Normal,

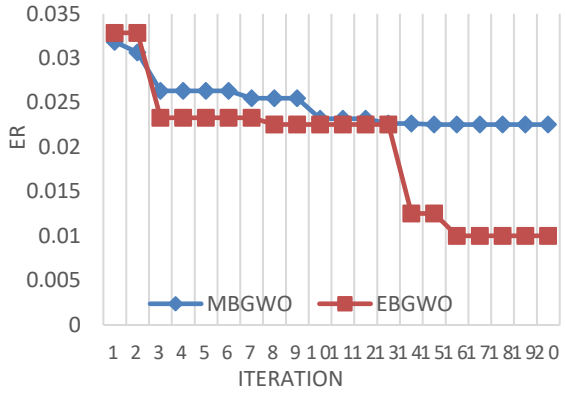


Figure. 5 Convergence of MBGWO and EBGWO for Normal class

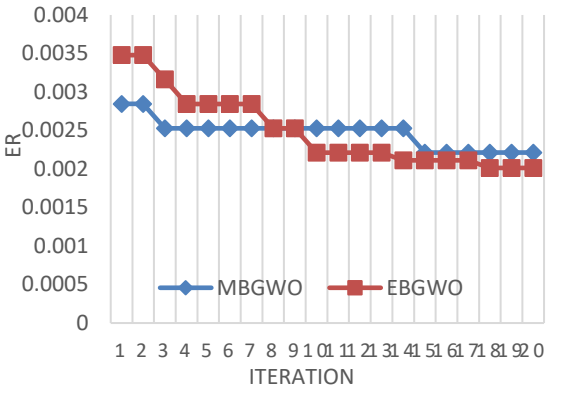


Figure. 8 Convergence of MBGWO and EBGWO for U2R class

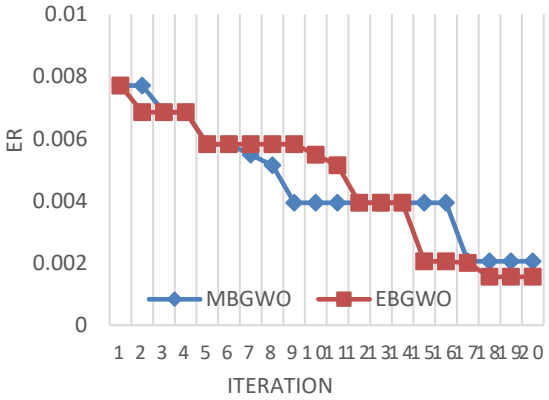


Figure. 6 Convergence of MBGWO and EBGWO for DoS class

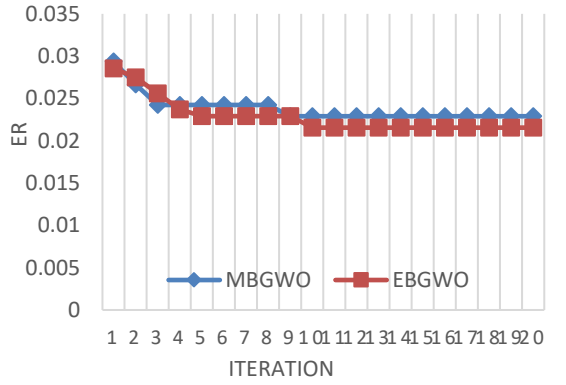


Figure. 9 Convergence of MBGWO and EBGWO for R2L class

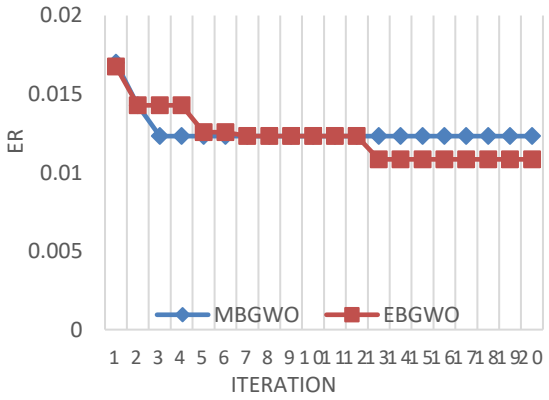


Figure. 7 Convergence of MBGWO and EBGWO for Probe class

DoS, Probe, U2R and R2L, are displayed in Figs. 5–9. Within a single run, the MBGWO algorithm showed that it lacked in minimising the error rate of the attacks as it could only reach a limited number of optimal features. Therefore, it required running for many times to achieve a small number of selected features with high accuracy. This was due to the fact that MBGWO linearly decreasing parameter did not consider any previous knowledge during the search process. However, the proposed EBGWO method proved that it could seek many optimal features within a run with minimum error rate. This was because the proposed parameter control in EBGWO forced the search wolves to jump out from non-promising regions to discover more areas in the search space. Furthermore, trade-off between the exploration and exploitation processes depended on the knowledge during the search process.

6. Conclusion

The proposed EBGWO has enhanced the linearly decreasing parameter in MBGWO for FS in anomaly detection. The general advantage of this enhancement is that the detection of the anomaly occurs with high accuracy by selecting the most efficient features. In addition, the effectiveness of EBGWO has been shown where only the most relevant subset content features were utilised, which resulted in the highest classification accuracy. The experimental results indicated that EBGWO was superior than other algorithms, where it obtained 19 features only out of a total of 41 features with 87.46% classification accuracy. The results also indicated that EBGWO is

superior in terms of classification accuracy and number of chosen features when assessed on the NLS-KDD dataset with different attack classes. Finally, future research can focus on the hybridisation of EBGWO with stochastic local search method to obtain improvements in each candidate solution.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The draft has been prepared by the 1st author while the review and editing has performed by the 2nd author.

Acknowledgments

The researchers thank the Malaysian Ministry of Higher Education for financially supporting this research under the Grant Scheme (TRGS /1/2018/UUM/02/3/3 (S/O code14163).

References

[1] A. Alghuried, "Model for Anomalies Detection in Internet of Things (IoT) Using Inverse Weight Clustering and Decision Tree", *Masters dissertation, Technological University Dublin*, 2017. doi:10.21427/D7WK7S

[2] K. Prabha and S. Sudha, "A Survey on IPS Methods and Techniques", *Int. J. Comput. Sci. Issues*, Vol. 13, No. 2, pp. 38–43, 2016.

[3] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection", *IEEE Commun. Surv. Tutorials*, Vol. 18, No. 2, pp. 1153–1176, 2016.

[4] D. K. Bhattacharyya and J. K. Kalita, *Network Anomaly Detection: A Machine Learning perspective*, No. November. Crc Press, 2013.

[5] V. Engen, J. Vincent, and K. Phalp, "Exploring discrepancies in findings obtained with the KDD Cup '99 data set", *Intell. Data Anal.*, Vol. 15, No. 2, pp. 251–276, 2011.

[6] F. Amiri, M. Rezaei Yousefi, C. Lucas, A. Shakery, and N. Yazdani, "Mutual information-based feature selection for intrusion detection systems", *J. Netw. Comput. Appl.*, Vol. 34, No. 4, pp. 1184–1199, 2011.

[7] H. G. Kayacik, A. N. Zincir-Heywood, and M. I. Heywood, "Selecting features for intrusion detection: A feature relevance analysis on KDD 99 intrusion detection datasets", In: *Proc. of PST 2005 - 3rd Annu. Conf. Privacy, Secur. Trust. Conf. Proc.*, Vol. 94, No. January, pp. 1723–1722, 2005.

[8] S. Zargari and D. Voorhis, "Feature selection in the corrected KDD-dataset", In: *Proc. of - 3rd Int. Conf. Emerg. Intell. Data Web Technol. EIDWT 2012*, pp. 174–180, 2012.

[9] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, and K. Dai, "An efficient intrusion detection system based on support vector machines and gradually feature removal method", *Expert Syst. Appl.*, Vol. 39, No. 1, pp. 424–430, 2012.

[10] H. Liu, L. Yu, and S. Member, "Toward integrating feature selection algorithms for classification and clustering", *Knowl. Data Eng. IEEE Trans.*, Vol. 17, No. 4, pp. 491–502, 2005.

[11] D. A. Ostrowski, "Feature selection for twitter classification", In: *Proc. of. - 2014 IEEE Int. Conf. Semant. Comput. ICSC 2014*, No. 97, pp. 267–272, 2014.

[12] S. Agarwal, "Data mining: Data mining Concepts and Techniques", In: *Proc. of IEEE International Conf. on on Machine Intelligence and Research Advancement*, pp. 203-207, 2014.

[13] J. Song, "Feature Selection for Intrusion Detection System", *Dr. Diss. Aberystwyth Univ.*, 2016.

[14] V. Kumar and S. Minz, "Feature Selection : A literature Review", *SmartCR*, Vol. 4, No. 3, 2014.

[15] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "An Efficient Intrusion Detection System Based on Feature Selection and Ensemble Classifier", *arXiv Prepr. arXiv*, Vol. 14, No. 8, pp. 1–12, 2019.

[16] H. F. Eid, M. A. Salama, A. E. Hassanien, and T. H. Kim, "Bi-layer behavioral-based feature selection approach for network intrusion classification", In: *Proc. of International Conf. on Security Technology*, Vol. 259 CCIS, pp. 195–203, 2011.

[17] W. Fan, N. Bouguila, and D. Ziou, "Unsupervised anomaly intrusion detection via localized Bayesian feature selection", In: *Proc. of 2011 IEEE 11th International Conf. on Data Mining*, pp. 1032–1037, 2011.

[18] J. Wahid and H. F. A. Al-mazini, "Classification of Cervical Cancer Using Ant-Miner for Medical Expertise Classification of Cervical Cancer Using Ant-Miner for Medical Expertise Knowledge Management", In: *Proc. of Knowledge Management International Conf. (KMICe)*, No. 7, pp. 393–397, 2018.

[19] A. M. Jabbar, K. R. Ku-Mahamud, and R. Sagban, "An improved ACS algorithm for data

clustering", *Indones. J. Electr. Eng. Comput. Sci.*, Vol. 17, No. 3, pp. 1506–1515, 2020.
[20] A. M. Jabbar, K. R. Ku-Mahamud, and R. Sagban, "Modified ACS centroid memory for data clustering", *J. Comput. Sci.*, Vol. 10, No. 10, pp. 1439–1449, 2019.
[21] H. N. K. Al-behadili, K. R. Ku-Mahamud, and R. Sagban, "HYBRID ANT COLONY OPTIMIZATION AND ITERATED LOCAL SEARCH FOR RULES-BASED CLASSIFICATION", *J. Theor. Appl. Inf. Technol.*, Vol. 98, No. 04, pp. 657–671, 2020.
[22] H. N. K. Al-behadili, Ku-Mahamud, and R. Sagban, "Adaptive Parameter Control Strategy for Ant-Miner Classification Algorithm", *Indones. J. Electr. Eng. Informatics*, Vol. 8, No. 1, pp. 149–162, 2020.
[23] R. Sagban, K. R. Ku-Mahamud, M. Shahbani, and A. Bakar, "Reactive max-min ant system with recursive local search and its application to TSP and QAP", *Intell. Autom. Soft Comput.*, Vol. 23, No. 1, pp. 127–134, 2017.
[24] V. Agrawal and S. Chandra, "Feature selection using Artificial Bee Colony algorithm for medical image classification", In: *Proc. of 2015 Eighth International Conf. on Contemporary Computing (IC3)*, pp. 171–176, 2015.
[25] L. Y. Chuang, C.-H. Yang, and J. C. Li, "Chaotic maps based on binary particle swarm optimization for feature selection", *Appl. Soft Comput.*, Vol. 11, No. 1, pp. 239–248, 2011.
[26] H. M. Zawbaa, E. Emary, A. E. Hassanien, and B. Parv, "A wrapper approach for feature selection based on swarm optimization algorithm inspired from the behavior of social-spiders", In: *Proc. of 2015 7th International Conf. of Soft Computing and Pattern Recognition (SoCPaR)*, pp. 25–30, 2015.
[27] M. Sudana, R. Nalluri, Saisujana, H. Reddy, and V. Swaminathan, "An Efficient Feature Selection using Artificial Fish Swarm Optimization and SVM Classifier", In: *Proc. of 2017 International Conf. on Networks & Advances in Computational Technologies (NetACT)*, No. 7, pp. 407–411, 2017.
[28] J. André and G. Sargo, "Binary Fish School Search applied to Feature Selection", 2013.
[29] Y. Zhang, D. Gong, Y. Hu, and W. Zhang, "Feature selection algorithm based on bare bones particle swarm optimization", *Neurocomputing*, Vol. 148, pp. 150–157, 2015.
[30] R. Forsati, A. Moayedikia, R. Jensen, M. Shamsfard, and M. R. Meybodi, "Enriched ant colony optimization and its application in feature selection", *Neurocomputing*, Vol. 142, pp. 354–371, 2014.
[31] R. Tiwari, "Correlation-based Attribute Selection using Genetic Algorithm", *Int. J. Comput. Appl.*, Vol. 4, No. 8, pp. 28–34, 2010.
[32] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Advances in Engineering Software Grey Wolf Optimizer", *Adv. Eng. Softw.*, Vol. 69, pp. 46–61, 2014.
[33] E. Emary, "Binary Gray Wolf Optimization Approaches for Feature Selection", *Neurocomputing*, Vol. 172, No. 8, pp. 371–381, 2015.
[34] Q. M. Alzubi, M. Anbar, Z. N. M. Alqattan, and M. Azmi, "Intrusion detection system based on a modified binary grey wolf optimisation", *Neural Comput. Appl.*, pp. 1–13, 2020.
[35] Q. Gu, X. Li, and S. Jiang, "Hybrid Genetic Grey Wolf Algorithm for Large-Scale Global Optimization", *Complexity*, pp. 1–18, 2019.
[36] Q. Al-tashi, S. J. Abdulkadir, H. Rais, and S. Mirjalili, "Binary Optimization Using Hybrid Grey Wolf Optimization for Feature Selection", *IEEE Access*, pp. 1–9, 2019.
[37] A. Sahoo and S. Chandra, "Multi-objective Grey Wolf Optimizer for improved cervix lesion classification", *Appl. Soft Comput. J.*, Vol. 52, pp. 64–80, 2017.
[38] M. Chandra, A. Agrawal, A. Kishor, and R. Niyogi, "Web Service Selection with Global Constraints using Modified Gray Wolf Optimizer", In: *Proc. of 2016 International Conf. on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1989–1994, 2016.
[39] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set", In: *Proc. of 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pp. 1–6, 2009.
[40] E. Emary, H. M. Zawbaa, and C. Grosan, "Experienced Gray Wolf Optimization Through Reinforcement Learning and Neural Networks", *IEEE Trans. Neural Networks Learn. Syst.*, Vol. 29, No. 3, pp. 681–694, 2017.
[41] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary ant lion approaches for feature selection", *Neurocomputing*, Vol. 213, pp. 54–65, 2016.
[42] N. Kunhare, R. Tiwari, and J. Dhar, "Particle swarm optimization and feature selection for intrusion detection system", *Sadhana - Acad. Proc. Eng. Sci.*, Vol. 45, No. 1, pp. 2–14, 2020.
[43] S. Lee, S. Soak, S. Oh, W. Pedrycz, and M. Jeon, "Modified binary particle swarm optimization",

Prog. Nat. Sci., Vol. 18, No. 9, pp. 1161–1166, 2008.

[44] S. Mirjalili, S. M. Mirjalili, and X. S. Yang, "Binary bat algorithm", *Neural Comput. Appl.*, Vol. 25, No. 3–4, pp. 663–681, 2014.

[45] M. Mafarja, R. Jarrar, S. Ahmad, and A. A. Abusnaina, "Feature selection using Binary Particle Swarm optimization with time varying inertia weight strategies", In: *Proc. of the 2nd International Conf. on Future Networks and Distributed Systems*, pp. 1–9, 2018.

[46] Y. He, W. J. Ma, and J. P. Zhang, "The Parameters Selection of PSO Algorithm influencing On performance of Fault Diagnosis", In: *Proc. of MATEC Web of Conf.*, Vol. 63, p. 02019, 2016.