VOL. 13, NO. 6, MARCH 2018 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

A FEATURE SELECTION APPROACH USING BINARY FIREFLY ALGORITHM FOR NETWORK INTRUSION DETECTION SYSTEM

Rana F. Najeeb and Ban N. Dhannoon
Computer Science Department, College of Science, AL-Nahrain University, Baghdad, Iraq
E-Mail: stcs-rfn16@sc.nahrainuniv.edu.iq

ABSTRACT

The number of attacks in recent times has tremendously increased due to the increase in Internet activities. This security issue has made the Intrusion Detection Systems (IDS) a major channel for information security. The IDS's are developed to in the handling of attacks in computer systems by creating a database of the normal and abnormal behaviours for the detection of deviations from the normal during active intrusions. The issue of classification time is greatly reduced in the IDS through feature selection. This paper is proposing the implementation of IDS for the effective detection of attacks. Based on this, the Firefly Algorithm (FA), a new binary feature selection algorithm was proposed and implemented. The FA selects the optimal number of features from NSL dataset. Additionally, the FA was applied with multi-objectives depending on the classification accuracy and the number of features at the same time. This is an efficient system for the detection of attacks reduction of false alarms. The performance of the IDS in the detection of attacks was enhanced by the proposed classification and feature selection algorithms.

Keywords: intrusion detection system, feature selection, firefly algorithm, classification.

1. INTRODUCTION

The security of computer systems has become an issue in recent times due to the developments in the fields of e-business, social networking, e-learning and online trading. Many high-profile company networks and web services have been effectively brought down by hackers and intruders. To secure network infrastructures and internet communication, many methods have been proposed and developed, including the use of firewalls, virtual private networks, and encryption. Intrusion detection method was recently added to these techniques. Within the past few years, the Intrusion detection methods have been employed synergistically with machine intelligence. With intrusion detection methods, information can be easily collected and used from known attacks to ascertain the possibility of attacks on a network or host [1, 2].

Intrusion detection methods (IDM) are systems for the identification and handling of malicious computer and network resource usage. This includes the unauthorized intrusion of the exterior system and internal user's un-authorized behaviors. The IDM was developed to ensure the security of computer systems by discovering and inform in gun-authorized and abnormal situations as well as network security violation. There are two categories of intrusion detection techniques which are anomaly detection technique and misuse detection technique. Misuse detection technique identifies intrusion by the matching of the attack features through the attacking feature library. Its speed of intrusion detection is high with a low chance of false alarms; though it does not identify non-designated attacks in the feature library, and cannot detect several new attacks. In anomaly detection techniques, the usual features of user's behaviors are stored in the database, and the behavior of the current user is compared to those stored in the database. In the presence of a high rate of divergence, it can be said that an abnormal situation has occurred. It has the advantage of being comparatively irrelevant with the system and its strong versatility possibility of detecting novel attacks. But because a complete description of the behavior of all the users in the system cannot be provided by conducting a normal contour, the user behaviour soften change, and there is a high chance of false alarms [3, 4, 5].

During the development of an IDS that uses machine learning technique, one of the major factors to be considered is the design of appropriate features that represent activities and differentiate normal network usage from attacks [6, 7, 8]. Even though there are many features that have been proposed, the lack of publicly available data sets makes the objective evaluation and fair comparison of the proposed features difficultand similarly delays the systematic investigations into the effect of features on IDSs. To solve this problem, MIT Lincoln laboratory [9] formulated 1999 KDDCUP data set, while Tavallaee *et al.* [10, 11] modified it to develop the NSL_KDD data set. After these, the performance of IDS proposed by many researchers has been subsequently evaluated objectively using the KDD'99 and NSL_KDD data sets.

However, there are 41 features in the connection vectors processed from raw tcpdump data in the KDD'99 and NSL_KDD data sets. These data sets have therefore been considered as having too many features for real time deployment in IDS. Researches on IDSs have recently overcome this problem by using only the feature parts that have attracted several research attentions. This is referred to as feature selection problems and has elicited the proposing of many feature selection methods. With feature selection, the computation time can be reduced, prediction performance can be improved, and the machine learning data or pattern recognition applications can be understood.

However, in many proposed feature selection methods, the central focus has been on the analysis of the individual feature relevance to the data set using analysis measures such as dependency ratio, information gain or correlation coefficient [13, 14, 15]. In these methods, features are usually ranked in a suggested metrics order, and then removed based on the ranking results [12]. These approaches do not explicitly consider the combinatorial properties of features, despite the capability of the combinatorial properties of features to lead to emergent effects on the performance of IDSs; and this is a major drawback of these methods. In other words, important features with less individual information but highly informative when in combination with other features could be eliminated [15, 16]. The reason for adopting the relevance of individual features to the data as a feature selection criteria in the proposed metho dsdespite knowing the issues with these approaches is due to the large number of possible feature subsets. As the number of features in the KDDCUP data set is 41, the total number of feature subsets is up to 241-1. Therefore, it is difficult to get the optimal feature subsets for IDSs based on the evaluation of the individual performance of the features rather than a collectively.

This study proposed a novel feature selection method (the wrapper type) based on the binary Firefly Algorithm (FA) and Naïve Bayesian Classifier (NBC). This paper is organized as follows: Section 2 presented the description and properties of the KDD and NSL data sets, while section 3 presented the details of the proposed feature selection algorithm. Section 4 compared the performances of the selected feature subsets and 41 features using an MLP. Finally, section 5 presented the major conclusions and recommendations for future research directions.

2. DATA SET

In this paper, the NSL_KDD data set was used to assess the performance of the subsets selected by running the proposed algorithm. The original KDD'99 data set widely used for the evaluationof the performance of IDSs is made up of the test and train data sets, each with nearly 300 thousand and 5 million instances, respectively. Table 1 showed the instances of attack from 41 features that belong to one of the four forms of attack (Denial of Service, User to Root, Remote to Local, and Probing Attacks).

Table-1. Categories of attack reserved in KDD-Cup '99.

Attack	Description
Denial of Service (Dos)	These attacks exhaust the network traffic or computing resources, denying the legitimate users of the services provided.
User to Root (U2r)	These attacks try to bypass the network after sniffing the ordinary users.
Remote to Local (R2l)	These attacks try exploit the target server vulnerability to access the ordinary users.
Probing	These attacks collect network activity information in an attempt to avoid security management.

These 41 features are additionally grouped into 3 groups which are the basic, contents, and traffic features. Features in the basic group include *duration*, service, and *protocoltype*. They show the attribute of being extracted from a TCP/IP connection. Those in the content group are features like *Num_failed_logins*, *logged_in*, *num_compromised*, and *su_attempted.* They are used mainly for the detection of suspicious behaviors such as login failure in a network. The traffic featuresare computed by monitoring the network connection for about 2s. They are divided into two groupsbased on the host (same host features) and based on the service (same service features). Those in the 'same host features' include *serror_rate and rerror_rate;*while those in the "same service features" include *srv_serror_rate and srv_rerror_rate.* A detailed information on these features is available at the corresponding websites [6, 7].

There are some advantages of the NSL_KDD data set over the KDD'99 data set even though it is a subset of KDD'99 data set. At first, no redundant records exist in the NSL_KDD data set as in KDD'99 train and test data set; so, there will be nobias in the learning algorithm based-IDS based towards more frequent records. Records from each attack category are adjusted based on its level of difficulty with respect to attack detection; making it easier to evaluate different detection methods with improved accuracy. Secondly, there is a reasonable amount of records in the NSL_KDD data set which made it possible to objectively compare different detection methods while avoiding the arbitrariness that occurs when using randomly selected data parts.

There are similar categories (4) of attacks in the NSL_KDD data as the KDD'99 data set, with each data instance having 41 features. In this paper, NSL_KDD data set was converted to only two categories (normal and attack); with the attack category containing all the categories except for normal. This paper is, therefore, presenting a feature selection method for binary classification problem of IDS, employing a data set of 10,000 records.

Firefly algorithm (FA)

The FA is a nature-inspired biological global stochastic approach of optimization developed by Yang [17]. It is a meta-heuristic approach based on Firefly population, with each Firefly representing a potential

ARPN Journal of Engineering and Applied Sciences

search space solution. The FA copies the mating and light flash-based information exchange mechanisms of Fireflies. In this section, the major attributes of Fireflies, the artificial FA, as well as the variants introduced to the basic algorithm already proposed were presented. Three idealized rules which describe the behavior of the artificial Fireflies were proposed by Yang [17] as:

- Fireflies are unisex and can be attracted to each other irrespective of the sex.
- The degree of attractiveness is related to the intensity of the emitted light; therefore, Fireflies with lights of lesser intensity will be made to move towards lights with higher intensities. Attractiveness decreases with increase in the distance between fireflies. They will randomly move when there is no brighter Firefly within the surrounding.
- The brightness of the light from the Fireflies is a function of the landscape of the fitness function. The brightness can be proportional to the fitness function value of the maximization problem. From these criteria, a summary of the basic steps of the FA can be presented as the pseudo-code illustrated in Figure-1.

Firefly Algorithm

Objective function $f(\mathbf{x})$, $\mathbf{x} = (x_1, ..., x_d)^T$
Generate initial population of fireflies $\mathbf{x}_i$ $(i = 1, 2, ..., n)$
Light intensity I_i *at* $\mathbf{x}_i$ *is determined by* $f(\mathbf{x}_i)$
Define light absorption coefficient γ
while *(t <MaxGeneration)*
for $i = 1 : n$ *all n fireflies*
  for $j = 1 : i$ *all n fireflies*
    if $(I_j > I_i)$, *Move firefly i towards j in d-dimension;* **end if**
    Attractiveness varies with distance r via $\exp[-\gamma r]$
    Evaluate new solutions and update light intensity
  end for j
end for i
Rank the fireflies and find the current best
end while
Postprocess results and visualization

Figure-1. Pseudocode of firefly algorithm.

3. THE PROPOSED ALGORITHM

The major motivation towards building a feature selection algorithm is to find better subset features for better performance accuracy. With the traditional wrapper model like the FA, all Fireflies are initialized with randomly selected features, but in the proposed model, all the Fireflies in the swarm will be initialized in a binary sequence. The major steps in the proposed algorithm as follows: -

3.1 Initialization

This step initiates all the Fireflies in the swarm by a random number in the range of [0,1]. These random numbers represent the position of each Firefly, and are calculated using Equation 1.

$$X = (UB - LB) \times Rand(0,1) + LB \quad (1)$$

where UB and LB represent the upper bound (1.0) and lower bound (0.0), respectively. The generated sequence will be converted into abinary sequence using the sigmoid function as follows: -

$$B_i = \begin{cases} 1, & sigmoid\ (X_i) > U\ (0,1) \\ 0, & otherwise \end{cases}$$

where X_i is the position of a Firefly, the sigmoid (X_i) is $1 / (1 + e^{-GRI})$, and U is the uniform distribution. Bi represents the binary sequence, where 1 implies that the feature will be selected, 0 implies the feature will not be selected. The Fireflies are initialized through these steps. Each Firefly has its own position based on the generated number of each one.

3.2 Fitness function

The fitness function of the proposed algorithm is to minimize the error rate of the classification performance over the validation set of given training data, as shown in Equation 2 while maximizing the number of non-selected features (irrelevant features). To calculate the fitness function, a classifier should be used. In this case, the Naïve Bayesian Classifier was applied to get the accuracy.

$$\text{Error} = \sigma * \frac{|\#\text{Features}|}{[\#\text{All Features}]} + (1 - \sigma) * \frac{\text{Err}[\#\text{Features}]}{\text{Err}[\#\text{All Features}]} \quad (2)$$

where *#Features* is the selected features; *Err* is the classifier error rate; in other words, the 5-fold cross validation error rate after training the Naïve Bayesian; and σ is a constant value limited to the range [0,1], which regulates the importance of the classification performance to the number of selected features. After calculating the error, the intensity of each Firefly is calculated using Equation 3.

$$I\,(F_i) = \frac{1}{1 + Error^2} \quad (3)$$

3.3 Attractiveness calculation

The attractiveness β of each Firefly can be defined using Equation 4.

$$\beta(r) = \beta_0 \times e^{-\gamma r^2} \quad (4)$$

where r represents the distance between two Fireflies and can be calculated using Equation 5, and β_0 represents the attractiveness at r =0 (Initial Case).

$$r_{ij} = |X_i - X_j| \quad (5)$$

where X represents the real values of the position of the Fireflies which have been calculated by the information gain ratio equation. The distance is calculated using the hamming distance method, by subtracting each bit of Firefly *i* from Firefly *j*. The distance in this method is represented by the difference between the binary strings of the two Fireflies. This method will improve the Firefly algorithm for working with the binary sequence (features) better than working with the continuous values (positions).

3.4 Position updating

Each Firefly in the swarm moves towards the brighter Firefly; in other words, Fireflies (*Fi*) are attracted by the brighter Firefly. This step can be called position updating which can be determined using Equation 6.

$$X_i = X_i + \beta \times (X_j - X_i) + \alpha \times (Rand - \tfrac{1}{2}) \quad (6)$$

where P_i in the first part of the equation represents the current position, and the second part contains the attractiveness between the position of F_i and F_j. Gr represents the information gain ratio values for all the features that have been calculated in the first step. The third part contains the randomization with α, where $\alpha \in [0,1]$. The randomness parameter is decremented by another constant rate δ, where $\delta \in [0.95\,, 0.97]$, so that at the final stage of the optimization,α has its minimum value as in Equation (7).

$$\alpha = \alpha \times \delta \quad (7)$$

4. EXPERIMENTS SETUP

This part is divided into two parts; the first part showed the results of the proposed algorithm over different types of testing, while the second part showed the comparison between the proposed algorithm and another two well-known algorithms. Two kinds of metrics - accuracy were measured to analyze the performance of the feature subsets generated by the proposed selection algorithm. They are commonly defined as follows.

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

4.1 The results of the proposed algorithm

This part showed the results of the proposed algorithm in terms of the performance accuracy and number of selected features. The experiments contain two main factors - the number of iterations and number of Fireflies in the swarm (or swarm size). The results presented in the tables (Tables 2, 3, and 4) showed that the FA can improve the performance of the Naïve Bayesian Classifier in all the cases. Each table contained different values of swarm size (SS) and fixed values of the number of iterations (IT).

From the Tables, it can be noticed that the major improvements in the accuracy occurred when the swarm size was increased at the same time the number of iterations affected the results, but with a little improvement.

Table-2. The results of the proposed FA with 250 iterations.

SS	*IT*	*Case*	*SF*	*RF*	*BS*	*ACC*	*AVE*
10	250	Best	15	26	1110000100010110010100001001010110000010010	96.16	94.22
		Worst	15	26	0111100110010001001101000000000100001001 10	94.23	
20	250	Best	16	25	0110011110010010100000010001001100001101 0	96.066	95.26
		Worst	17	24	0111001000111011000100010001001000011011 0	94.500	
30	250	Best	13	28	0010000100010010010000011011000100011001 0	96.33	95.58
		Worst	14	27	0011000110010000100111101000000000110010	94.80	
40	250	Best	15	26	0110000110110101000000000010001000011011 1	96.46	95.63
		Worst	16	25	0111100101010100101001100011001000001000 0	94.90	

ARPN Journal of Engineering and Applied Sciences

Table-3. The results of the proposed FA with 500 iterations.

SS	*IT*	*CASE*	*SF*	*RF*	*BS*	*ACC*	*AVE*
10	500	Best	15	26	11100101000101100010000100010000000010111	96.26	95.51
		Worst	20	21	01100011110100001011100100100001101110101	94.40	
20	500	Best	14	24	11100101101100100000010000000100000011001	96.26	95.78
		Worst	14	27	10100101000100001100000011010011000010001	95.26	
30	500	Best	11	30	11100001001100001010000010100100000000000	96.33	95.82
		Worst	15	27	10100111000110001000000110100000000110110	95.36	
40	500	Best	13	28	01100111100101000000010001000100000010001	96.50	95.89
		Worst	19	22	11101011001111000110000100100010000011011	95.26	

Table-4. The results of the proposed FA with 1000 iterations.

SS	*IT*	*Case*	*SF*	*RF*	*BS*	*ACC*	*AVE*
10	1000	Best	13	28	11100011000100000000100010000100000110011	96.50	95.65
		Worst	15	26	00101101000110100101000000100000100110101	94.83	
20	1000	Best	13	28	01100000110100000100000100100011000011001	96.40	95.78
		Worst	15	26	01101110000110000011100010100000000110001	95.40	
30	1000	Best	17	24	00100011000100010101110100100010000110111	96.20	95.86
		Worst	16	25	01101100001101000001011000110010000010101	95.43	
40	1000	Best	15	26	01100001000110001000010110100110000110001	96.63	96.01
		Worst	16	25	01111101011100000000010100000101000110010	95.60	

Table-5 summarized the results by comparing the best results of each experiment with the original accuracy (all features).

Table-5. Results of comparing the best results of each experiment with the original accuracy.

SS	IT	SF	RF	ACC
Original	-	41	0	89.6
40	250	15	26	96.46
40	500	13	28	96.50
40	1000	15	26	96.63

Table-5 showed that the best results were obtained by the maximum swarm size of 40. The results were increasing but with no major difference, when comparedwith the results based on the number of iterations. We can conclude that the proposed method needed for 40 Fireflies in the swarm but with 250 iterations to decrease the time.

4.2 Benchmarking the proposed method with other algorithms

The proposed IDS model is anomaly based and has two main stages - the pre-processing stage, which involved the wrapper feature selection process that combines BBAL with the detection classifier (NBC); the second stage is the detection step which showed the performance measures obtained by the classifier with previously selected feature subsets. To test our model, a personal computer with a core i7 processor, speed 2.2 GHz, and 4 GB of RAM running under windows 10 operating system was used. Also, for the ranking, the proposed algorithm was benchmarked with two other algorithms (Binary Particle Swarm Optimization (BPSO) and Binary Bat algorithm (BBA)). The results of these other two algorithms were lifted from previous studies [reference].

The three algorithms had their individual parameters and use specific values, as follows:

Swarm size = 10, Maximum number of iteration = 200.
For BFA:
Bmin = 0.0, G = 1.0, A = 0.2, D = 0.96.
For BBA:
In the case of BBA, we setup:

- the maximum loudness A0 =0.5, and the minimum pulse rate r0 = 0.5,
- the frequency ranges between 0.8 and 1.0,
- = 0:1 and _= 0:9

BPSO: -
- learning factors are c1 = 2.3 and c2 = 1.8,
- inertia weight was reduced from 0.9 to 0.5.

Table-6 showed the results of all the algorithms.

Table-6. The results of all the algorithms.

ALGORITHM	ACC. RATE	ERR. RATE	NO. FEATURES
BPSO	90.63%	9.37%	22
BBAL	91.61%	8.09%	15
BFA	92.02%	7.98%	14
NBC	89.9%	10.1%	41(ALL)

5. CONCLUSIONS

A wrapper feature selection method was proposed in this paper and applied to intrusion detection system. Further tests on the performance of the proposed method were performed using Naïve Bayesian Classifier. The NSL-KDD dataset was used and it empirically proved that the movement and randomization of the Firefly algorithm were enhanced by distance calculation through hamming distance method since the Firefly algorithm was initialized by a binary sequence, unlike the standard Firefly algorithm. This enhancement can offer better results in terms of performance accuracy and the number of selected features. Further works will focus on proposing and testing other modifications for improving the metaheuristics approaches for feature selection problems in general, and the intrusion detection system in particular.

REFERENCES

[1] Paliwal S., Gupta R. 2012. Denial-of-service, probing & remote to user (R2L) attack detection using genetic algorithm. Int. J. Comput. Appl. 60(19): 57-62.

[2] Sabhnani M., Serpen G. 2013. Application of machine learning algorithms to KDD intrusion detection dataset within misuse detection context. In: Proc. Int. Conf. Mach. Learn.: Model, Technol., and Appl, pp. 209-215.

[3] Azad C., Jha V.K. 2013. Data mining in intrusion detection: a comparative study of methods, types and data sets. Int. J. Inf. Technol. Comput. Sci. 5(8): 75-90.

[4] Parazad S., Saboori E., Allahyar A. 2012. Fast feature reduction in intrusion detection datasets. In: Proc. 35th Int. Conv., MIPRO. pp. 1023-1029.

[5] Kang, S.-H. 2015. A feature selection algorithm to find optimal feature subsets for detecting DoS attacks. In: Proc. 5th Int. Conf. IT Conv. Secur. pp. 352-354.

[6] Nguyen H.T., Petrovic S., Franke K. 2010. A comparison of feature-selection methods for intrusion detection. LNCS. 6258, 242-255.

[7] Chebrolu S., Abraham A., Thomas J.P. 2004. Hybrid feature selection for modeling intrusion detection system. LNCS. 3316, 1020-1025.

[8] Chebrolu S., Abraham A., Thomas J.P. 2005. Feature deduction and ensemble design of intrusion detection systems. Comput. Secur. 24(4): 295-307.

[9] KDD Cup 1999, 2007. Available on:http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[10] NSL_KDD data set: http://nsl.cs.unb.ca/NSL-KDD/.

[11] Tavallaee M., Bagheri E., Lu W., Ghorbani A.A. 2009. A detailed analysis of the KDD CUP 99 data set. In: Proc. 2009 IEEE Int. Conf. Comput. Intell. Secur. Def. Appl. pp. 53-58.

[12] Chandrashekar G., Sahin F. 2014. A survey on feature selection methods. Comput. Electr. Eng. 40(1): 16-28.

[13] Kayacik H.G., Zincir-Heywood A.N., Heywood M.I. 2005. Selecting features for intrusion detection: a feature relevance analysis on KDD 99 intrusion detection datasets. 3rd Annu. Conf. Priv. Secur. Trust, St. Andrews, New Brunswick, Canada.

[14] Olusola A.A., Oladele A.S., Abosede D.O. 2010. Analysis of KDD '99 intrusion detection dataset for selection of relevance features. In: Proc. World Congr. Eng. Comput. Sci. p. 1.

[15] Xu Z., King I., Lyu M.R.T., Jin R. 2010. Discriminative semi-supervised feature selection via manifold regularization. IEEE Trans. Neural Netw. 21(7): 1033-1047.

[16] Cuyon I., Elisseeff A. 2003. An introduction to variable and feature selection. J. Mach. Learn. Res. 3, 1157-1182.

[17] Xin-She Yang. 2009. Firefly Algorithms for Multimodal Optimization, Stochastic Algorithms: Foundations and Applications. pp. 169-178.