2528-9705

Örgütsel Davranış Araştırmaları Dergisi
Journal Of Organizational Behavior Research
Cilt / Vol.: 7, Sayı / Is.: S, Yıl/Year: 2022, Kod/ID: 22S0-850

Intrusion Detection in computer networks based on improved artificial neural networks using an artificial bee colony algorithm

Seyed Hedayat Mohammadi

Graduated in Computer Science from Yasouj Branch Azad University

E-mail: Hedayat0148@gmail.com

ABSTRACT

Today, computer networks monitor and manage critical infrastructures such as banking, transportation, commerce, and telecommunications. As a result, securing these systems against planned attacks is crucial. Most of these attacks exploit software errors and security vulnerabilities in the target system. It is impossible to eliminate software errors, so all software has security gaps. This study aims to detect intrusion in computer networks using the improved artificial neural network approach based on the artificial bee colony algorithm. The NSL-KDD dataset is used to test and evaluate the proposed model. The training set includes 24 types of attacks, while the test set includes 14 distinct attacks that do not exist in the training set. The results proved that the proposed method is more efficient than other methods in all cases where the data have a different observation probability. Although in some cases, our proposed method is less efficient than the basic method, in most cases, it is one of the best methods available. After selecting important features and deleting irrelevant data, the power and accuracy of the classification in the proposed method increase significantly.

Keywords: *intrusion detection, artificial neural networks, artificial bee colony algorithm, security gaps*

INTRODUCTION

An intrusion detection system is a set of tools, methods, and documentation helping identify, detect, and report unauthorized network activity [1]. The intrusion has different meanings. Some experts see intrusion as a failed attack, while others offer different definitions of intrusion and attack. As a rule, the intrusion is defined as: "An active set of interrelated events aimed at unauthorized access to information, alteration of information or damage to the system so that the system becomes inaccessible. This definition also includes both successful and unsuccessful efforts". "Intrusion detection" is not appropriate for intrusion detection systems, as these systems cannot really detect intrusion but rather introduce network activity as intrusion, which may not be intrinsic. An intrusion detection system is a small part of a protection system that is not considered a self-sufficient and autonomous system.

Evolutionary algorithms are widely used in pattern selection to detect intrusion. The basic idea of evolutionary algorithms is as follows: An initial population of chromosomes (a set of solutions) is generated (in our discussion, they are the same set of patterns). Based on an objective function (in pattern selection, the objective function is equivalent to a classifier), individuals in the population are evaluated, and the best chromosomes are selected for mutation and cross-over to create new chromosomes (to maximize the objective function). This algorithm is repeated a certain number of iterations (generations) determined by the user, and the best chromosomes are selected from the last generation.

Zhang and Sun have used taboo search (TS) to select a pattern [2]. The TS algorithm is applied to the initial set. Some solutions are identified as taboo (i.e., they should not be changed). In Si, other solutions (non-taboo solutions) are evaluated by a classifier in order to find the best solution. Finding a solution from Si is to evaluate the neighboring sets. Sub-sets with only one component different from Si are replaced by sub-sets that are better classified. Once this process is complete, S will be considered the best solution.

In the reference [3], 6 weighted features based on experience are selected to participate in the classification rules. A simple genetic algorithm is then used to infer classification rules from the network audit data. The support-confidence framework is also used as a fitness function to judge the quality of each rule.

In [4], to build an experimental IDS system for multiclass classification, the incoming traffic pattern is pre-processed, and redundant samples are removed. Then a feature selection algorithm based on a genetic algorithm is presented, which leads to minimizing the computational complexity of the classifier. Finally, a neurotree model is used as the classification engine, with a higher detection rate than NN * and C45.

[5] A SVM-based genetic algorithm is used to determine the feature set automatically. In this method, the objective function corresponding to a specific chromosome equals f = error_rate. The ultimate goal is to minimize this objective function. The results proved that GA could reduce the error rate by 9%.

In [6], a new approach to combining SVM and CSOACN to exploit the benefits of both is proposed. Subsequently, the proposed algorithm is tested and evaluated using a standard KDD99 dataset. Experimental results show that CSVAC (a combination of support vectors with ant colony) is better than SVM and CSOACN in demand rate, efficiency, and execution time.

In [7], an intrusion detection technique using a hyper-graph-based genetic algorithm (HG-GA) (for parameterization) and support vector machine (SVM) (for feature selection) is proposed.

In [8], a new method based on multi-criteria linear programming and particle swarm optimization is introduced to increase attack detection accuracy. Multi-criteria linear programming is a classification method based on mathematical programming that has a great ability to solve real data mining problems. However, regularizing the parameters is an essential step in the training phase.

Security is one of the vital challenges in many systems, especially in migrating to the cloud. In this regard, an organization must handle security threats and related risks. The lack of direct control over assets and their potential management by the service provider makes cloud security more difficult than traditional models such as master-client models. As a result, security is one of the main challenges in the cloud computing environment. This is the most important security risk in hybrid and public cloud models and private models provided by third parties. The use of cloud services requires the transfer of responsibility, control of information and systems of the organization to an external service provider. Like other computing models, security in cloud computing is defined in terms of three distinct aspects, including data confidentiality, precision, and accessibility. Organizations may become overly dependent on one service provider. They may also face many challenges in transferring data and services into the organization or to another service provider. The use of new cloud computing services in the event of safety accidents reduces the necessary speed and agility of the organization. This paper aims to detect

intrusion in computer networks using the improved artificial neural network approach based on the artificial bee colony algorithm.

Research Methodology

The steps of implementing the evolutionary algorithm in the proposed method are as follows:

1. Initial population: At the starting point of the evolutionary algorithm, the base values for the parameters must be specified. The particles of the algorithm are equivalent to the neural network parameters.

2. Particles: In the structure of the algorithm used in the proposed method, each particle represents a set of neural network settings. In this process, we have 9 features in our intrusion detection dataset. So the number of hidden layer neurons is between 5 and 27.

- Learning function: The learning function is selected from 9 existing learning functions.
- Error function: The error calculation method is achieved based on four criteria.
- Transfer function: The transfer function in the second layer is selected based on eight available functions.

It is assumed that different segmentation of training and test data may affect the performance of the whole system. We consider three separate particles to determine the number of training, testing and evaluation data.

- Training data rate: The rate of division of main data into training data is equal to 0.4 to 0.7
- Validation data rate: The rate of division of main data into validation data is equal to 0.2 to 0.4
- Test data rate: The rate of division of main data into test data is equal to 0.1 to 0.3. This criterion is calculated by adding the seventh and eighth particles and subtracting them from the number 1. As a result, each chromosome is considered a neural network.

3. Selection methods: There are different methods for selecting particles in evolutionary algorithms. In this research, the selection roulette selection function has been used.

4. Fitness function: In this phase, the evaluation of networks built based on the fitness function is achieved. An evolutionary algorithm or GA is a programming technique using hereditary evolution as a problem-solving model.

5. Stop condition: There are two ways to stop the algorithm. Often after a certain number of iterations, the algorithm stops running. In some cases, the performance stops when a certain solution is reached, or the fitness level in a generation reaches a certain value.

The proposed algorithm can be summarized as follows:

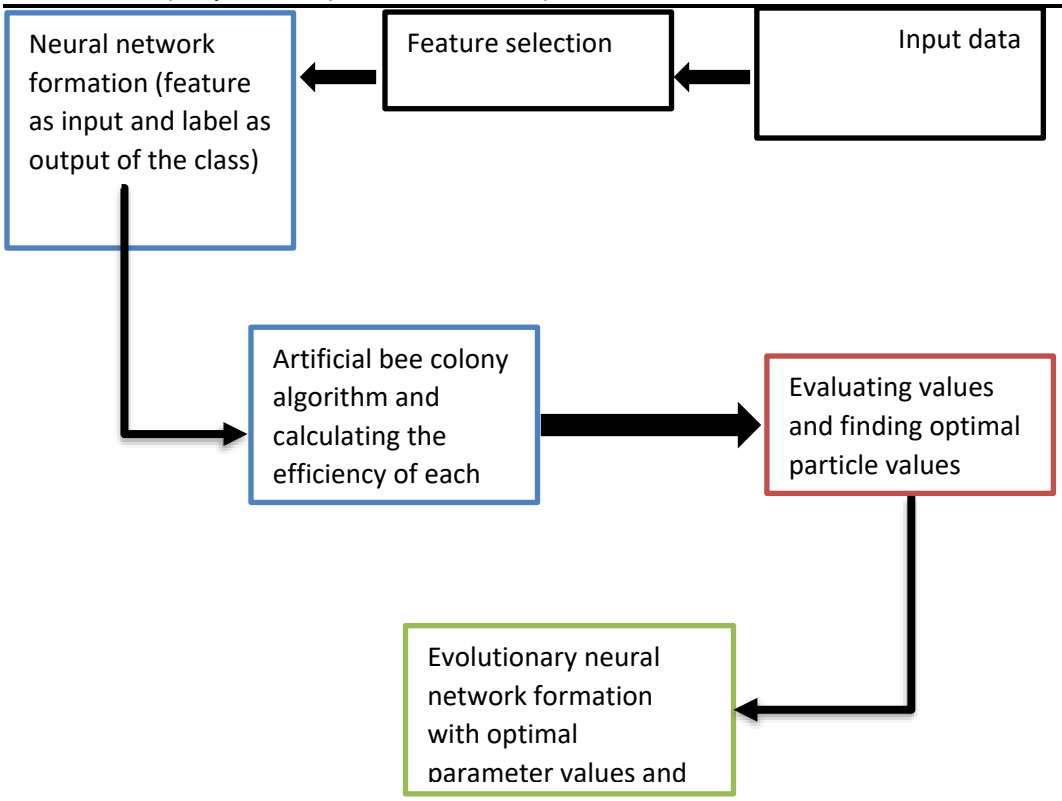


Figure 1: Proposed algorithm

The proposed method is evaluated on the NSL-KDD dataset [60]. This dataset contains selected records from KDD-CUP99 [61]. Challenges in the original dataset, like duplicate records in this dataset, have been addressed. Since 1999, the KDD-CUP99 dataset has been the most common dataset for evaluating anomaly detection and intrusion detection methods. This dataset was created by Stoolfo et al. [62] in the DARPA'98 Intrusion Detection Systems Assessment Program [63].

DARPA'98 contains approximately 5 million records of various connections, each containing 100 bytes. The KDD training set contains approximately 4900000 vectors of different connections, each containing 41 features. Each vector is classified into two classes: normal or attack. Each attack belongs to one of the following categories:

1) Denial of Service (DoS) attack: An attacker overwhelms a processing resource or memory so that it cannot execute authorized requests.

2) User to Root (U2R) attack: An attacker starts by accessing an authorized user's account (via password eavesdropping, dictionary attack, or social engineering) and accessing the system root by finding vulnerabilities in the system.

3) Remote to Local (R2L) attack: The attacker can send packets on the network but does not have an account on the network machines. So attacker accesses the system by examining the vulnerabilities of the system as a user.

4) Surveillance attacks: Such an attack attempts to obtain information about a computer network to threaten network security.

Most importantly, the probability distribution of the test set is not the same as the probability distribution of the training set. The test set has certain types of attacks not present in the training

set. This process brings the evaluations closer to reality. In other words, the training set includes 24 types of attacks, while the test set includes 14 distinct attacks that do not exist in the training set. Table (1) shows these classifications.

Table 1: Classification of attacks in the NSL-KDD dataset

Class	Sub-Class
NORMAL	-
PRB	ipsweep, nmap, portsweep, satan
DOS	back, land, neptune, pod, smurf, teardrop
U2R	buffer_overflow, loadmodule, multihop, perl, rootkit
R2L	ftp_write, guess_passwd, imap, phf, spy, warezclient, warezmaster

The approximate distribution of different data classes in the training and test data is shown in Table (2). Due to the rounding of percentages, the total number is not equal to 100%.

Table 2: Approximate distribution of training and test data in the NSL-KDD dataset

Class	Test	Training
NORMAL	19 %	48 %
PRB	1 %	20 %
DOS	73 %	26 %
U2R	0.07 %	0.02 %
R2L	5 %	5 %

One of the data normalization process tasks is to convert text feature values to numeric values. Table (1) reflects the characteristics of the NSL-KDD dataset. Some of the 41 feature values in NSL-KDD are text values. The support vector machine uses only numeric data for training and testing, so text feature values must be converted to numeric values. Features with text values include (B) Protoclo_type, (C) Service, and (D) Flag, which are represented by numbers 2, 3, and 4 in Table (2), respectively.

For example, for Protoclo_type feature values, tcp is 1, udp is 2, and icmp is 3. In data normalization, after converting these three features from text format to numeric format, the main challenge is to convert data to binary form or normalized continuous form. In the proposed intrusion detection system, the normal continuous form is used. To normalize the features, a statistical analysis was performed on each feature based on the data in the NSL-KDD and the maximum, and minimum values for the values of each feature were determined. Then, based on Equation (1), normalization was realized in the range [1, 0].

$$Nf = \frac{f-MinF}{MaxF-minF} \qquad (1)$$

where F is the desired feature, f represents the value of the feature, maxF is equal to the maximum value of the feature F, and minF is equal to the minimum value of the feature F.

Table 3: NSL-KDD data set features

Label	Name of feature	Label	Name of feature
A	Duration	**V**	Is_guest_login
B	Protocol_type	**W**	Count

C	Service	X	Sev_count
D	Flag	Y	Serror_rate
E	Src_byte	Z	Sev_serror_rate
F	Dst_byte	AA	Rerror_rate
G	Land	AB	Srv_rerror_rate
H	Wrong_fragment	AC	Same_srv_rate
I	Urgent	AD	Diff_srv_rate
J	Hot	AE	Srv_diff_host_rate
K	Num_failed_login	AF	Dst_host_count
L	Logged_in	AG	Dst_host_srv_count
M	Num_comprised	AH	Dst_host_same_srv_rate
N	Root_shell	AI	Dst_host_diff_srv_rate
O	Su_attempted	AJ	Dst_host_same_src_port_rate
P	Num_root	AK	Dst_host_srv_diff_host_rate
Q	Num_file_creations	AL	Dst_host_server_rate
R	Num_shells	AM	Dst_host_srv_serror_rate
S	Num_access_files	AN	Dst_host_rerror_rate
T	Num_cutbounds_cmds	AO	Dst_host_srv_rerror_rate
U	Is_host_login		

Precision and Recall are two well-known parameters used in evaluating data mining and machine learning algorithms. Precision is defined as follows:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Furthermore, Recall is defined as follows:

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

where TP represents data correctly assigned to the positive class, FP represents data incorrectly assigned to the positive class, and FN represents data incorrectly assigned to the negative class.

Results

The advantage of using the feature selection step in the proposed model is the reduction of training time and testing of the support vector machine, reducing the computational costs and necessary computer resources such as memory and CPU time. In this regard, training and testing time without feature selection and feature selection have been calculated and shown in Table (4). The unit of measurement of the time is milliseconds.

Precision, recall and F-score based on attack class for 41 features and eight features are shown in Tables (4) to (6). As can be seen from the results, feature selection has led to increased precision, recall, and F-score of the intrusion detection system. In all 5 classes, the detection rate with 8 features is higher than the detection rate with 41 features. In particular, the percentage of attack detection for U2R and R2L classes using feature selection increases significantly. The system also uses important features to have a higher ability to detect new and unknown attacks, i.e., attacks that are not encountered in the training process and are only present in the test data.

Table 4: Training and testing time of the proposed model with and without using feature selection

	41 features	8 features	Time reduction
Training time	238454 ms	71320 ms	70.09%
Testing time	24820 ms	3989 ms	83.94%

Table 5: Precision of detection of proposed system based on attack class

	Normal	DoS	PRB	U2R	R2L
41 features (artificial bee colony algorithm)	76.71 %	97.72 %	99.86 %	53.49 %	63.39 %
8 features (artificial bee colony algorithm)	96.65 %	97.29 %	99.87 %	55.86 %	99.04 %

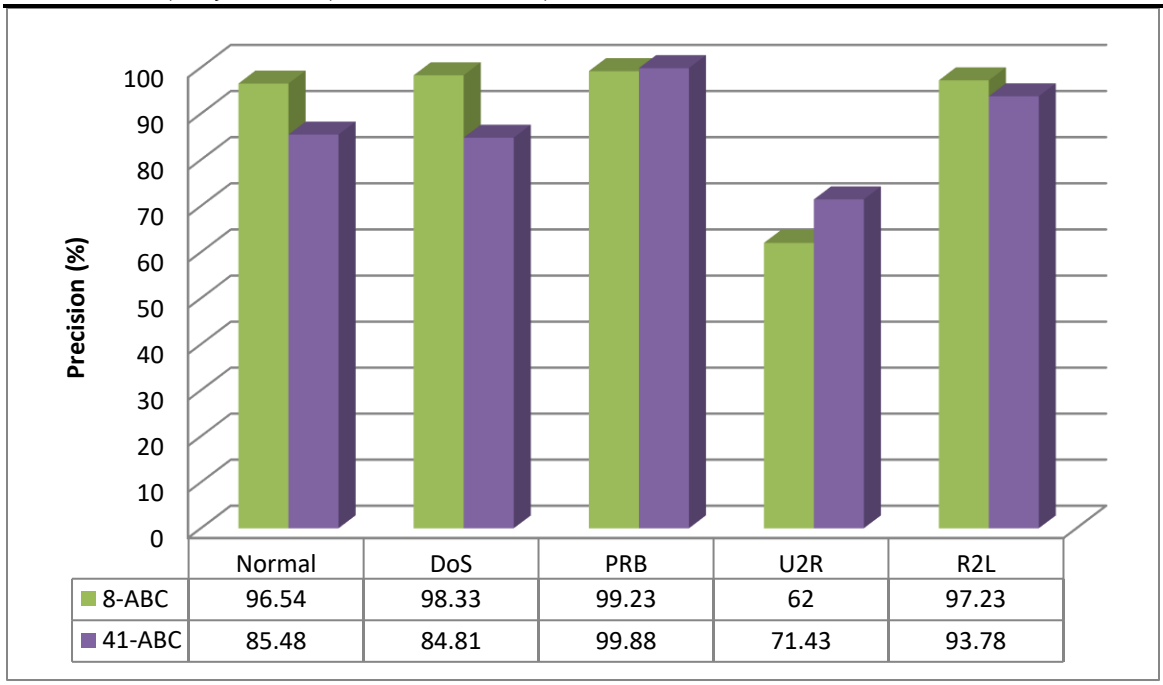
Table 6: Recall of detection of proposed system based on attack class

	Normal	DoS	PRB	U2R	R2L
41 features (artificial bee colony algorithm)	97.99 %	86.27 %	97.65 %	40.07 %	53.55 %
8 features (artificial bee colony algorithm)	98.39 %	84.10 %	98.50 %	46.40 %	65.40 %

Table 7: F-score of detection of proposed system based on attack class

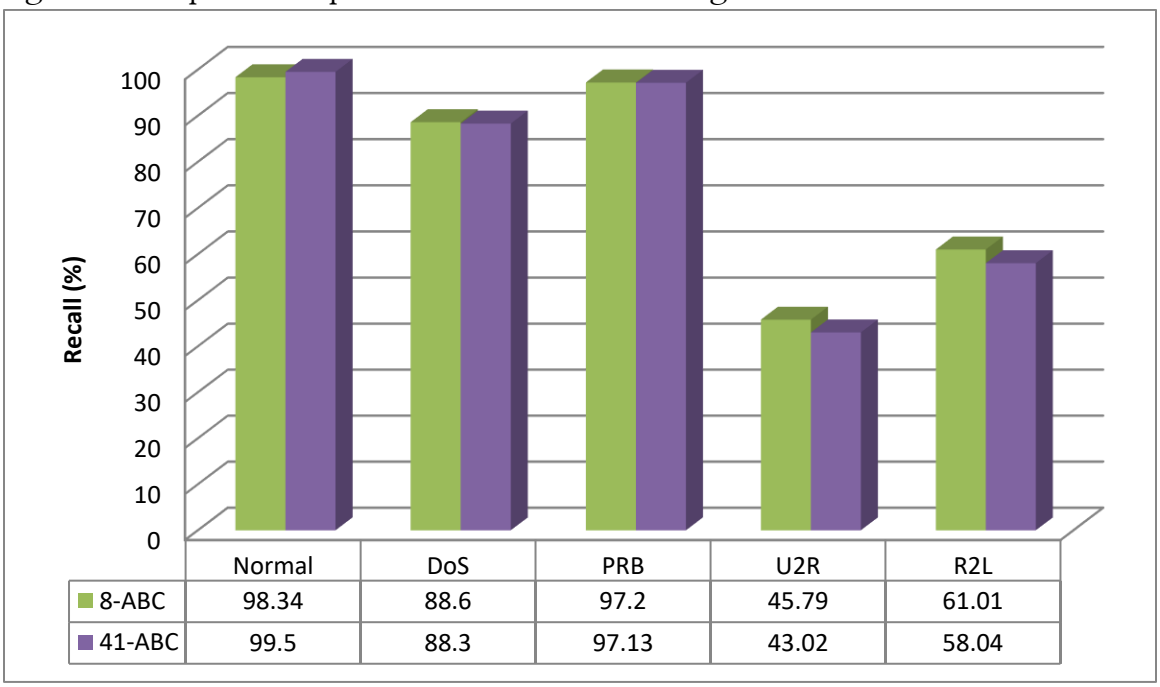
	Normal	DoS	PRB	U2R	R2L
41 features (artificial bee colony algorithm)	86.05 %	91.63 %	98.74 %	45.81 %	58.05 %
8 features (artificial bee colony algorithm)	97.51 %	90.11 %	99.18 %	50.69 %	78.77 %

In this study, the efficiency of the proposed method is compared with a multilayer perceptron neural (MLP) network based on a back-propagation (BP) algorithm. The BP learning algorithm uses the steepest descent (S.D) algorithm. Regulating the network parameters according to the error signals is calculated based on the feed of each pattern to the network. MATLAB software [73] has been used to implement this method. Comparison of precision, recall and F-score for detection using 41 features and 8 features for neural network based on artificial bee colony algorithm are shown in Figures (2) to (4), respectively. The figure shows that the value of the mentioned criteria of 8 features case with the neural network based on the artificial bee colony algorithm is higher than the others.



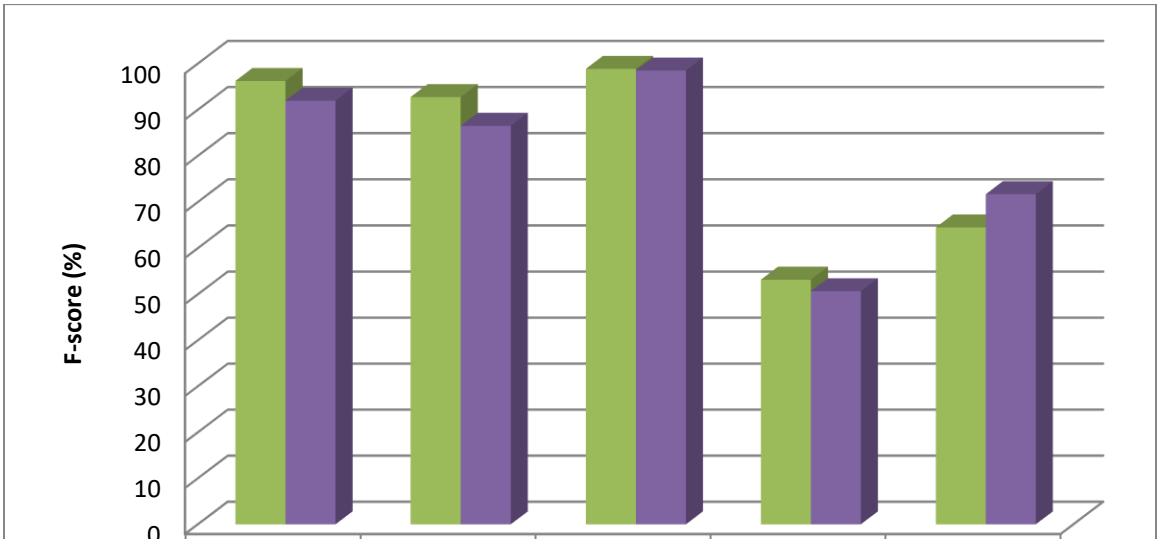
	Normal	DoS	PRB	U2R	R2L
8-ABC	96.54	98.33	99.23	62	97.23
41-ABC	85.48	84.81	99.88	71.43	93.78

Figure 2: Comparison of precision criteria for detecting attacks based on attack class



	Normal	DoS	PRB	U2R	R2L
8-ABC	98.34	88.6	97.2	45.79	61.01
41-ABC	99.5	88.3	97.13	43.02	58.04

Figure 3: Comparison of recall criteria for detecting attacks based on attack class



	Normal	DoS	PRB	U2R	R2L
8-ABC	96.23	92.74	98.86	53.17	64.49
41-ABC	91.95	86.51	98.48	50.66	71.7

Figure 4: Comparison of F-score criteria for detecting attacks based on attack class

Conclusion

Numerous experiments were performed to evaluate the proposed method in order to determine its strengths and weaknesses as well as to provide a comprehensive analysis of its performance. In these experiments, various parameters have been measured. The results of the experiments and the behavior analysis of the proposed method are presented in full. According to the extracted results, feature selection leads to increased accuracy, recall, and F scores in the model. In addition, the model training and testing time is significantly reduced, which in turn increases the speed of attack detection in the intrusion detection system. The results proved that the proposed method is more efficient than other methods in all cases where the data have a different observation probability. Although in some cases, our proposed method is less efficient than the basic method, in most cases, it is one of the best methods available. After selecting important features and deleting irrelevant data, the power and accuracy of the classification in the proposed method increase significantly.

References

[1] Endorf, Carl, Eugene Schultz, and Jim Mellander. Intrusion Detection & Prevention, McGraw-Hill, 2004.

[2] Bhutan, M.H.; Bhattacharyya, D.K.; Kalita, J.K., "Network Anomaly Detection: Methods, Systems and Tools," IEEE Communications Surveys & Tutorials, vol.16, no.1, pp.303-336, First Quarter 2014.

[3] Li, Tao, Chengliang Zhang, and Mitsunori Ogihara. "A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression." Bioinformatics 20, no. 15 (2004): 2429-2437.

[4] Tsoumakas, Grigorios, and Ioannis Katakis. "Multi-label classification: An overview." Dept. of Informatics, Aristotle University of Thessaloniki, Greece (2006).

[5] Hansen, Lars Kai, and Peter Salamon. "Neural network ensembles." IEEE transactions on pattern analysis and machine intelligence 12 (1990): 993-1001.
[6] Ravuri, Suman, and Andreas Stolcke. "A comparative study of recurrent neural network models for lexical domain classification." In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6075-6079. IEEE, 2016.
[7] Rowland, Craig H. "Intrusion detection system." U.S. Patent 6,405,318, issued June 11, 2002.
[8] Siwak, G., Siebenman, T. and Witt, M., SIWAK GREG, SIEBENMAN TED and WITT MAX, 2016. Intrusion Detection System. U.S. Patent 20,160,012,713.