Computer Engineering and Intelligent Systems
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol.10, No.4, 2019

DOI: 10.7176/CEIS

Intrusion Detection System Based on Combination of Optimized Genetic and Firefly Algorithms in Cloud Computing Structure

Parnian Zare
Department of Information Technology Engineering, Shiraz Branch, Islamic Azad University, Shiraz, Iran

Abstract
Attackers or hackers are always looking to attack networks. Optimizing and securing system settings prevents hackers from accessing networks to a great extent. Intrusion Detection Systems (IDS), firewalls, and Honey Pot (Honey Pot) are technologies that can prevent hacking attacks on the networks. IDS or Intrusion Detection System analyzes all activities on the network and uses the information available on its database in order to determine if the activity is allowed or considered unauthorized. It also determines whether this activity can harm your network or not and eventually notify such activities by sending alarms or alerts to the system administrator. The main purpose of intrusion detection system is to classify data and network traffic. Thus, the detection of penetration in these systems is essentially a classification operation, so if the classification operation can be improved, the performance of intrusion detection system could get increased. For this reason, we have used the ECOC algorithm to improve classification performance by categorizing general problem into trivial classes. Improvement means that by breaking down the problem into smaller classes and assigning a separate classifier to each class, the power and accuracy of the classification operation increases, thereby overall system performance would improve. Other important factor which enhance diagnostic performance is the use of appropriate features in training and testing classifications. For this reason, we used firefly and genetic algorithms to select the proper features of each classification in each level. The main goal of this research is to provide an intrusion detection system with better penetration detection and performance. Based on the results obtained from the system diagnosis, our proposed system has been able to increase the detection rate up to 5% in comparison with other intrusion detection systems.
Keywords: Intrusion Detection System, Genetic Algorithm, Firefly Algorithm
DOI: 10.7176/CEIS/10-4-02
Publication date:May 31st 2019

1. Introduction

Today, most vital infrastructures such as telecommunication, transportation, business and banking are managed by computer networks, so the security of these systems is very important for planned attacks. Most of these attacks exploit software errors and system security gaps. Since the complete elimination of software errors is not possible, each software includes security issues, which is known as software vulnerabilities. Researchers have been trying to find these vulnerabilities in order to identify system penetration gaps and then providing system protection through preventive or confrontive strategies.

Each dictionary has a meaning and concept for penetration. There are also many discussions over the meaning and influence of intrusion in computer science. Many consider intrusion as an unsuccessful attack, while others consider other definitions. As a result, penetration can be defined as "an active set of related events with the aim of unauthorized access to information, information conversion and system detriment in order to make the whole system unusable." This definition includes both successful efforts and unsuccessful efforts. "

An intrusion detection system could be considered as a set of tools, methods, and documentation which identify and report unauthorized or unregistered activities through the network. The "Intrusion Detection" heading is not appropriate for such systems, since they may perceive specific action as an intrusion which is not fundamentally a penetrate. Furthermore, intrusion detection systems are not self-sufficient or independent because they take in to account as a small part of the computer protective system.

2. Research and related work

In a related research, [21], a new fuzzy method based on semi-monitored learning is presented to improve classification performance with the use of unlabeled examples. In [20] A hybrid intrusion detection system is proposed to identify internal and external attacks. In this system, signature recognition algorithms are used to identify internal attacks and fuzzy firefly algorithms to detect external attacks. [18] is a feature extraction algorithm which improves classification performance in intrusion detection systems. This algorithm is capable of supporting linear and nonlinear data. Furthermore, the system applies a hybrid algorithm using PSO for weight generation and classification combination. In [19], a new algorithm utilized PSO[1] for parameter and feature selection, subsequently SVM is used as a classifier.

[1] Firefly Optimization

DOI: 10.7176/CEIS

In [3], an intrusion detection system is proposed using the decision tree algorithm and post-propagation neural network which has acceptable diagnostic accuracy. In [4] a composite system based on the post-propagation neural network and decision tree algorithm is designed using the KDD CUP 99 dataset. The results show that the intrusion detection system is not able to detect all types of network attacks by using the neural network without a decision tree.

In [5], for the classification of normal network activities and the Dos and Probe attacks, a multi-layer neural network is used for off-line system design and a multi-stage system is proposed for classification of normal data and related attacks. The results show that the system performance is better than one stage system. In [6], writers use the K-mean clustering algorithm to divide the dataset into several sub-spaces and then use a set of MLPs for each space. Their model has shown an acceptable performance on the KDD CUP 99 dataset. In [18], researchers have introduced a classification model that includes MLP and RBF, whose results have shown a passable performance on the network data.

The goal of most researchers is to identify different types of attacks from normal data [19-12]. In most researches, decision tree has been used along with other classification algorithms to improve the performance of the systems. Given this, by the use of composite IDS which combines the capabilities of various classification algorithms excellent results were achieved. What is important is the application of simple and efficient methods for designing intrusion detection systems that can be implemented in real networks. Combined techniques, despite having acceptable performance, may not have a proper operational efficiency on different types of networks in terms of low identification speed. Regarding this issue, if a combined method maintains performance with the same speed, an optimal intrusion detection system can be implemented.

Proposed algorithm

Our proposed algorithm is presented in this section. The algorithm consists of three main steps: in the first step, the ECOC algorithm has been tried to improve the performance of the data classification. In the second step, we have used genetic and firefly algorithms to improve the performance of each classifier in order to select the most appropriate feature. Finally, in the third step, using the Hamming distance, we determine the class of each data.

In the first step

Using the ECOC technique and algorithm, we classify data classes into 15 classes with the strategy of separating paired-data class. The data classes contain five different classes, including a normal data class, and four data classes of attack types which are DOS, Probe, R2L, and U2R. The pattern of paired-class separation presented by the matrix utilizing ECOC algorithm in table1. In fact, the purpose of using ECOC algorithm is to divide and simplify the system in a way that all data types could classify in different combinations so that various information and analysis could be extracted.

Another unique feature of this algorithm would be data classifications in the simplest possible way. In the proposed classification method each class should only include two types of data classes. The pattern of the ECOC algorithm presents a MASK or a template using matrix, which can be random in most cases.

The columns of this matrix specify the categories or data classes, and each column represents all data types. Each column would consider same labeled-data as one group in order to maintain paired-class strategy. As a result, the pattern matrix included only two values which are 0 or 1.

For example, in each of the 15 classes, all of the data classes with value of 1 will be placed in one class and similarly all of the data classes with the value of 0 will be placed in another class.

Table 1. The pattern matrix generated by the ECOC algorithm

Class number Class type	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Normal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DOS	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
Probe	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0
R2L	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
U2R	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

In the second step

After identifying paired classes, extraction phase and feature selection are performed by genetic and firefly algorithms in each class. In this step, classifiers which are mentioned as matrix columns, have the function of selecting features. Each classifier has the task of classifying a paired class. Although each data has 41 features, we do not require all of them since the data classification method is simplified. For this reason, we use the genetic and firefly algorithms to derive the necessary attributes of each category. Feature extraction operations are calculated for each classifier separately by genetic and firefly algorithms, and ultimately, we consider approved features as the required characteristics of each classifier.

In fact, selected joint features obtained from two algorithms would be perceived as main attributes of each class. The process of feature selection in each classifier which includes genetic and firefly algorithms seeking a pattern or mask for attributes is presented in table2.

Table 2. Template matrix or mask of features

feature #1	Feature #2	Feature #3	---	Feature #41
1	1	0		0

In this mask, attributes specified with value 1 are selected and similarly attributes specified with value 0 are not selected. Finally, by using the decision tree and the selected attributes, we train each classifier and consider the function or accuracy of each tree's decision in terms of fitness function.

Finally, in the third step

In this step, the type of input data class must be specified. The output of the input data is similar to the following table. It has 15 columns, which are equal to the number of classifications.

Table 3 Generated template matrix of input data output

Class number Class type	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Sample 1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sample 2	1	1	1	1	0	1	1	1	0	0	0	0	0	0	1
Sample 3	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0
….	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
Sample n	1	0	1	1	1	0	1	1	0	0	0	0	1	0	0

To determine the class of each sample data, we compare the table rows of each sample data with five rows in the table. In fact, we calculate Hamming distance. Then we assign each sample data to a class that has a lower Hamming distance or higher similarity.

4- Evaluation

To assess the evaluation of the proposed method, the network traffic dataset is used. This dataset is collected by the Lincoln MIT Laboratory's Technology and Cytology Unit. The main KDDCUP'99 dataset consists of 41 entries that are presented as datasets and class labels. It also includes five different classes as shown in table5. These features are basically divided into three categories: (1) features that are extracted from the TCP / IP connection of the transport capacity check, and are named as the base features; (2) features that could access the load capacity of the TCP packet and also monitoring the suspicious behavior within the load capacity section are identified as content features; (3) time based and host-based traffic features are designed to evaluate attacks using more than two seconds intervals and a date-based window. Features are determined as below:

1) 1-10 base features
2) 11-22 content features
3) 23-31 time-based traffic features
4) 32-41 host-based traffic features

DOI: 10.7176/CEIS

Table 4. description of input features

	#	Input feature	Data type	#		Input feature	Data type
Basic Features	1	Period of time	Continuous	Time based Traffic Features	23	Counts	Continuous
	2	Protocol type	symbolic		24	srv_count	Continuous
	3	services	symbolic		25	serrorate_r	Continuous
	4	flag	symbolic		26	Error rate srv	Continuous
	5	src_bytes	continuous		27	Error rate Again	Continuous
	6	dst_bytes	continuous		28	Error rate again srv	Continuous
	7	ground	symbolic		29	same_srv_rate	Continuous
	8	Wrong part	continuous		30	diff_srv_rate	Continuous
	9	instantaneous	continuous		31	srv_diff_host_rate	Continuous
	10	hot	Continuous				
Content Features	11	Number of failed logins	Continuous	Host based Traffic Features	32	dst_host_count	Continuous
	12	Logging in	symbolic		33	dst_host_srv_count	Continuous
	13	Number of compromises	Continuous		34	dst_host_same_srv_rate	Continuous
	14	root_shell	Continuous		35	dst_host_diff_srv_rate	Continuous
	15	su_attempted	Continuous		36	dst_host_same_src_port_rate	Continuous
	16	Root numbers	Continuous		37	dst_host_srv_diff_host_rate	Continuous
	17	File creation number	Continuous		38	dst_host_serror_rate	Continuous
	18	num_shells	Continuous		39	dst_host_srv_serror_rate	Continuous
	19	File Access Number	Continuous		40	dst_host_rerror_rate	Continuous
	20	num_outbound_cmds	Continuous		41	dst_host_srv_rerror_rate	Continuous
	21	is_hot_login	symbolic				Continuous
	22	Guest login	symbolic				Continuous

Table 5. Attack Types

DOS	U2R	R2L	PROBE Attack
return	Perl	FTP write	IP sweep
Ping of death	Buffer overflow	Guess the password	NMAP
Neptune	Load module	IMAP	Port sweep
Smurf	Rootkit	Multi HOP	Satan
Land		Phf	
Teardrop		SPY	
		Wareclient	
		Warezmaster	

As mentioned, the KDD CUP 99 dataset is used to generate training and test datasets. This dataset has more than 4 million records, which is too much for the simulation process. Hence, according to [23], the training dataset contains 60593 normal records, 49115 DOS records, 1917 Probe records, 899 R2L records, 26 U2R records which are randomly selected from the main dataset. This dataset also provides 41 entries for each input data, and five normal classes as an output data which are DOS, Probe, R2L, and U2R attack classes.

There are various criteria for evaluating performance of intrusion detection systems. The proposed evaluation process utilizes recall, precision, and approximate mean (FM).

Accuracy = The number of related retrieved documents / Total number of retrieved documents. Recall = The number of related retrieved documents /The total number of related documents in the database.

1. $\text{Recall} = \frac{\text{Tp}}{\text{TP+F}}$
2. $\text{Precision} = \frac{\text{Tp}}{\text{TP+FP}}$
3. $\text{FM} = \frac{\text{2.Recall.Precision}}{\text{Recall+Precision}}$

in which:

TP = The number of attack records classified as attacks.
TN = The number of normal records that are classified as normal.
FP = The number of attack records classified as normal.
FN = The number of normal records classified as attacks.

Obtained results which are mentioned below highlight the advantages of using genetic and firefly algorithms. An important subject to be addressed is that the intrusion detection system has the task of

DOI: 10.7176/CEIS

categorizing and classifying different data with different characteristics. Therefore, specifying system performance for each group of data can express the overall performance more explicitly. As shown in the table6, the performance of each data class varies with other classes since the required features for classification task are different, and more importantly, the ratio of different data classes varies in the internet traffic. This is why the intrusion detection systems detect normal data better than other attacks data classes. So, if we look precisely, we'll see that higher data ratio in different classes leads to better system performance in other specific classes because of training enhancement due to data ratio development.

The system function is very important in detecting a U2R attacks, Because the amount of U2R related data in a traffic dataset is limited on the internet. Therefore, training and testing these data is more difficult than other data groups. But ultimately, the performance of the intrusion detection system is generally reported.

Table 6. System Performance Results

Class5 Normal data	Class4 U2R Attack data	Class3 R2L Attack data	Class2 Probe Attack data	Class1 DOS Attack data	Evaluation Criteria
98.19	**23.07**	**97.26**	**86.61**	**99.77**	**Recall**
99.44	**57.69**	**29.25**	**91.81**	**99.14**	**Precision**
98.81	**32.96**	**44.53**	**89.13**	**99.45**	**FM**

In Table 7 we plan to compare performance of the proposed system with a number of related systems. As shown in the table below, different algorithms used for implementation process. By examining other similar systems, we conclude that hybrid systems will bring better results since each algorithm has its own strengths and weaknesses, therefor if one can use the strengths of each algorithm, a system with acceptable performance would be provided.

Table 7. Comparison of the performance of different systems

classification	Performance level
J48	81.05
Naive Bayes	76.56
NB tree	82.02
Random forests	80.67
Random tree	81.59
Multi-layer perceptron	77.41
SVM	69.52
Fuzziness based semi-supervised	84.12
Proposed Method	99

5. Conclusion

In this paper, by using a huge amount of data in a cloud computing environment, we proposed a method based on combination of firefly and genetic algorithms to detect intrusions in cloud computing structure with an acceptable accuracy.

The firefly algorithm is used to select the initial population which get involved in the genetic algorithm. Furthermore, it could improve the genetic algorithm by applying early randomized chromosome population. Cloud computing is a large and complex environment, including hardware, software, and security. The success or failure of cloud services depends on users' trust. Trusting that their data and processes are protected in a safe and secure environment. In this research, the most critical part is to ensure a secure environment, by providing a basic view of hardware and software security policies. In future, high degree existence of cloud computing, encourages attackers to penetrate due to the large amount of data and resources. Although using further attack recognition techniques would minimize related losses, for future work, utilizing open standards to prevent conflicts and lock-in problems and setting up specific security standards for cloud computing structure are highly recommended.

References

1. Bukharov, Oleg E., and Dmitry P. Bogolyubov. "Development of a decision support system based on neural networks and a genetic algorithm." *Expert Systems with Applications* 42, no. 15 (2015): 6177-6183.
2. Kim, G., Lee, S., & Kim, S. (2014). A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Systems with Applications*, *41*(4), 1690-1700.
3. Tammi, Wasima Matin, Noor Ahmed Biswas, Ziad Nasim, Khadizatul Zannat Shorna, and Faisal Muhammad Shah. "Artificial Neural Network based System for Intrusion Detection using Clustering on Different Feature Selection." *International Journal of Computer Applications* 126, no. 12 (2015).
4. Balamurugan, V., & Saravanan, R. (2017). Enhanced intrusion detection and prevention system on cloud

environment using hybrid classification and OTS generation. *Cluster Computing*, 1-13.
5. Chandrashekhar, A. M., & Raghuveer, K. (2014). Amalgamation of K-means clustering algorithm with standard MLP and SVM based neural networks to implement network intrusion detection system. In *Advanced Computing, Networking and Informatics-Volume 2* (pp. 273-283). Springer, Cham.
6. Eesa, A. S., Orman, Z., & Brifcani, A. M. A. (2015). A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems. *Expert Systems with Applications*, *42*(5), 2670-2679.
7. Osanaiye, O., Cai, H., Choo, K. K. R., Dehghantanha, A., Xu, Z., & Dlodlo, M. (2016). Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing. *EURASIP Journal on Wireless Communications and Networking*, *2016*(1), 130.
8. Eesa, A. S., Orman, Z., & Brifcani, A. M. A. (2015). A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems. *Expert Systems with Applications*, *42*(5), 2670-2679.
9. Abuadlla, Y., Kvascev, G., Gajin, S., & Jovanovic, Z. (2014). Flow-based anomaly intrusion detection system using two neural network stages. *Computer Science and Information Systems*, *11*(2), 601-622.
10. David E. Rumelhart, James McClelland, L. and the PDP research group. (editors), Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1: Foundations. MIT Press.
11. Aburomman, A. A., & Reaz, M. B. I. (2016). A novel SVM-kNN-PSO ensemble method for intrusion detection system. *Applied Soft Computing*, *38*, 360-372.
12. Vaidya, Harendra, Shahrukh Mirza, and Nayan Mali. "Intrusion Detection System." International Journal of Advanec Research in Engineering, Science & Technology 3 (2016).
13. Sahasrabuddhe, Atmaja, Sonali Naikade, Akshaya Ramaswamy, Burhan Sadliwala, and Pravin Futane. "Survey on Intrusion Detection System using Data Mining Techniques." (2017).
14. Dawle, Yashashree, Manasi Naik, Sumedha Vande, and Nikita Zarkar. "Database Security Using Intrusion Detection System." Database 2, no. 03 (2017): 01-06.
15. Kukreja, Kashish, Yugal Karamchandani, Niraj Khandelwal, and Kajal Jewani. "Intrusion Detection System." International Journal of Scientific and Research Publications (2015).
16. Pan, Shengyi, Thomas Morris, and Uttam Adhikari. "Developing a hybrid intrusion detection system using data mining for power systems." IEEE Transactions on Smart Grid 6, no. 6 (2015): 3104-3113.
17. Bhavsar, Yogita B., and Kalyani C. Waghmare. "Intrusion detection system using data mining technique: Support vector machine." International Journal of Emerging Technology and Advanced Engineering 3, no. 3 (2013): 581-586.
18. Ambusaidi, Mohammed A., Xiangjian He, Priyadarsi Nanda, and Zhiyuan Tan. "Building an intrusion detection system using a filter-based feature selection algorithm." IEEE transactions on computers 65, no. 10 (2016): 2986-2998.
19. Bamakan, Seyed Mojtaba Hosseini, Huadong Wang, Tian Yingjie, and Yong Shi. "An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos firefly optimization." Neurocomputing 199 (2016): 90-102.
20. Aburomman, Abdulla Amin, and Mamun Bin Ibne Reaz. "A novel SVM-kNN-PSO ensemble method for intrusion detection system." Applied Soft Computing 38 (2016): 360-372.
21. Desai, Anuja S., and D. P. Gaikwad. "Real time hybrid intrusion detection system using signature matching algorithm and fuzzy-GA." In Advances in Electronics, Communication and Computer Technology (ICAECCT), 2016 IEEE International Conference on, pp. 291-294. IEEE, 2016.
22. Yu, H., & Liu, K. (2017, January). Classification of multi-class microarray datasets using a minimizing class-overlapping based ECOC algorithm. In *Proceedings of the 5th International Conference on Bioinformatics and Computational Biology* (pp. 51-54). ACM.
23. Ashfaq, Rana Aamir Raza, Xi-Zhao Wang, Joshua Zhexue Huang, Haider Abbas, and Yu-Lin He. "Fuzziness based semi-supervised learning approach for intrusion detection system." Information Sciences 378 (2017): 484-497.