A Combined Anomaly Base Intrusion Detection Using Memetic Algorithm and Bayesian Networks

H. M. Shirazi, A. Namadchian, and A. khalili Tehrani

***Abstract*—Anomaly base intrusion detection systems (IDSs) detection rate trend and enjoy relatively numerous false negatives and false positives. In this study, we aim to achieve a linear classification function using Memtic algorithm, to minimize errors of such IDSs and to improve such systems, as well.**

A combined system is offered in this paper which tries to find the optimum subset for detecting intrusion of any set of four attack classes of Knowledge Discovery in Database 99 (KDD99) by using of both correlation analysis amongst features and information theory. Then proper classification function is measured for each attack class through a Memetic algorithm. Bayesian networks are employed to combine results of any function in order to achieve the final classification.

Kdd99 dataset and its refined version, NSL-kdd, were used to estimate the offered system, our findings showed 93.42 detection rate. Likewise, NSL-kdd estimation shows the suggested system for R2L attack class has succeeded to classify 86.60% of records which have not been classified correctly by the previous algorithms.

***Index Terms*—Anomaly base intrusion detection; KDD99; correlation analyzing; NSL-kdd; memetic algorithm; Bayesian networks ;classification function.**

I. Introduction

Today, concerning highly usages of intrusion methods in networks, more interests have been attracted to intelligent IDSs. For intrusion detection systems, there are two analysis methods mainly, signature-based and anomaly detection methods. In signature based systems, while informing about attack patterns, the analyzer seeks for a alarm which represents a kind of abnormality.

Initially anomaly-based intrusion detection systems form normal behaviors' profiles (of system, network and/or its users) and then detect any violation or distortion of such normal profiles as abnormal and aggressive behavior [1]. Here we intend to deal with anomaly-based intrusion detection systems.

Anomaly-based intrusion detection systems are modeled variously [2]. Such models usually consist of two parts: display algorithm and classification algorithm. Display algorithm is used to show inputs and also to develop a space for feature selection. Classification algorithm labels input vector as normal or abnormal.

Manuscript received July 15, 2012; revised August 31, 2012.

The authors are with the Department of Computer, Malek-Ashtar University of Technology, Tehran, I. R. Iran (e-mail: shirazi@mut.ac.ir; amin.namadchyan@gmail.com; alireza_khalili2001@yahoo.com).

Data mining uses various algorithms, such as clustering and classification, along with analyzing dataset to elicit a form of knowledge and represent it in anomaly detection model [24]. In this study both data mining and classification techniques were used in anomaly-based intrusion detection systems.

Knowledge Discovery in Database 99 (KDD99), as a standard dataset, was used for simulation.This dataset includes 5,000,000 network connections. The training part of dataset includes 494021 connections out of which 20 percent are normal connections. Every connection includes independent fields and a (normal or attack) label. Any attack is categorized in one of the four following classes: User to root (U2R), remote to local (R2L), probe and denial of service [20].

Proper features of data mining must be extracted to create the model. A reasonable survey has been conducted in [3, 4, 5, 6, 7] in this field. In [8, 9] it has been tried to select proper features for anomaly-based intrusion detection systems through correlation analysis.

Guangzhi Qu et al [8] have conducted a good study in which the proposed algorithm selects an optimum set of features for intrusion detection through a new criterion for dependency i.e. QY (xi,xj). However only U2R and R2L attacks have been considered in this study and there was no action to combine results and to achieve final classification function. Moreover, time complexities of training phase were left without analysis.

Hai Nguyen et al [9] have tried to turn the feature selection problem into a polynomial mixed 0-1 fractional programming problem using correlation analysis criterion, hence to obtain proper features for any attack class. However in this study a classification function has been proposed for each attack class and its results have not been combined; thus in order to evaluate his algorithm, the author has partitioned kdd99 dataset into four sets based on four attack classes. Likewise, algorithm implementation time has not been estimated.

In the proposed algorithm, initially proper features of each class are selected using correlation analysis criterion and Memetic algorithm (MA). Then linear classification function is obtained through Memetic algorithm. This algorithm helps system to bypass local minima and become convergent, faster. Therefore, unlike papers which used genetics [8, 12, 15] we used all KDD training datasets and enhanced our precision as well.

The architecture and components of proposed system are introduced in section 2. Section 3 deals with Memetic algorithm and its characteristics. Features are elicited using Memetic algorithm and correlation criterion in section 4. The proposed algorithm to obtain linear classification function for

each class is introduced in section 5. In section 6 the trend to combine results of classification functions is implied using Bayesian networks, and finally discussion and conclusion are presented in section 8.

II. Algorithm of the Proposed System

Our proposed algorithm is composed of four phases. In the first phase, proper features of each class are selected using a Memetic algorithm which its evaluation function is represented by "e(S) in equation (1). In the second phase, optimum linear classification function is obtained for each class through a Memetic algorithm (figure 3). Architecture of this phase is shown in Fig.1.

Our objective in the third phase is to combine the mentioned functions and obtain a classification that is able to report occurring or not occurring of an attack based on the features. For this purpose, Bayesian networks are used. This algorithm classifies data based on information theory [18, 19]; thus it appears that there is more coordination with other parts and a better respond is resulted. Its architecture is demonstrated in Fig. 1.

After obtaining a proper classification, system may isolate attacks from ordinary traffic through this function. KDD99 dataset was used to test this system.

Fig. 1. Architecture of the proposed system

III. Memetic Algorithm (MA)

A meme was designed by Richard Dawkins as a similar form of a gene in genetic algorithm in 1976, however it had some differences. [16, 17].

Since anomaly detection problem may be considered as problem space searching, evolutionary algorithms can be used to solve this problem. Based on studies in [12, 8 and 22], genetic algorithm has reflected appropriate results.

Considering genetic algorithms disadvantages, e.g. getting stuck in local minima, using MA, which has solved this problem somehow [17], may enhance both speed and precision of intrusion detection.

IV. Selecting Proper Features

The quality of analyzed data plays a significant role in enhancing the precision of data mining algorithms. Data quality would be analyzed through two aspects: data dependency and data redundancy.

Presence of independent data and extra features of data mining model led to a weak prediction, numerous calculations and decreased effectiveness. Generally, selecting proper features for earning algorithms in data mining decreases data dimension and algorithms consumed space hence its speed will be increased. Sometimes it increases classification precision [4]. On the other word a proper subset of features is a subset that depends on decision variable and its members have the least dependency [5].

Feature selection problem would be modeled as problem space searching in which any form of problem space is a subset of possible features and time complexity has componential relation with some features; as a result, feature selection problem is a NP problem. For example, our problem space for 41 features will be 2^{41} [3, 10].

Features selection problem solving methods would be categorized through some standards. A known category divides such methods into three classes: filter, wrapper and a combination of both [10]. Accordingly, used method in this study is a combined method.

In order to select a proper subset of features we used a criterion in [8]:

$$e(S) = \frac{\sum_{\forall j \in I_m} I(Y; X_j)}{H(Y)} - Q(X_i, X_j). \quad (1)$$

where;

S: a subset of features. Higher e implies lower dependency between the features of S and their higher dependency to decision variable, Y [8].

MA was used to select proper features (section 2). Each chromosome is a 41-bit string. Zero value of each bit indicates that feature does not belong to "S", and if a bit is equal to 1, the certain feature belongs to a subset of "S," Equation 1 is used in fitness function and gaining more e(S) by chromosome brings about better respond.

V. Classification Functions for Each Class

As it can be seen in Fig.1 in our proposed system MA[1] has been used to obtain linear classification functions, in a way that length of each chromosome is proportional to the number of selected features for that class. Values of each gene are coefficients of linear classifier function and are in the range of

[1] Memetic algorithm

(-1023,1023). Evaluation function for each chromosome is as Eq.2 which reflects classification precision:

$$CR = \frac{TP + TN}{Size\_of\_dataset} \quad (2)$$

In evaluation function, firstly any feature is normalized through a method presented in [22]. Then its attack or normal nature is measured using IS-attack function (Fig.2).

```
Bool IS_attack(Selected Feature ,chromosome )
{
        If
∑_{k=0}^{n-1} gene_k × Normal(Selected Feature_k) < gene_n  then
is attack else is not attack
}
Float Fittness (chromosome)
{
        sum=0;
        For every connection record in learning data set if
data row is attack and IS_attack(
Selected Feature ,chromosome ) then sum++
Return CR = sum / size of data set
}
Memetic algorithm Pseudocode
{
Initialise P randomly (P = Population)
For I = 1 to m (m = population size)
Perform local search in the neighbourhood of i
(I being the current chromosome)
Evaluate fitness of I and its neighbourhood explored
Make i the best chromosome found
End For
Repeat
        Select parents from P
Generate offspring applying recombination
        To the parent selected
        If an individual is selected to undergo mutation,
                Then apply local search
        Evaluate fitness of current individual and its
neighbours
        Adopt best chromosome
Until stop best chromosome fittness > σ
```

Fig. 2. Classification algorithm of each class

Where σ is a value between 0 -1 interval which is selected by the user based on his required precision rate. The more this value is close to 1, the more precise classification functions will be but it will take more time. The value of σ vector for {Normal, DOS, U2R, R2L, probe} will be selected as $P = \{0.96, 0.87, 0.88, 0.98, 0.98\}$ respectively.

VI. Using Bayesian Networks to Combine Results

Whenever connection information (selected feature) is bestowed to classification function of each class, it may have different results due to the mentioned connection belonging to attacks or normal traffic. Thus, we need an algorithm which concludes such results and makes decision about normal or abnormal condition of a connection. To do this, Bayesian networks were employed.

In the training phase, we evaluated data of each connection using classifier function of each class and then stored results in a database. Then the output of the classifier function is given to the Bayesian network (from the Weka [25]) as input. For the experiments, we applied Weka's default values as the input parameters of these methods.

The network is erected after training phase. In operational phase outputs are imported to the developed network from classifier functions of each class and the final result is fruited.

VII. Evaluation of the Proposed System

To evaluate the proposed system, a proper dataset should be selected in the first phase which either meet the necessary standards or be comparable with other works. Kd99 [20] is a proper dataset, however this database in not only old but suffers some drawbacks too. Its drawbacks were analyzed in [13] and after solving them, a dataset so called NSL-Kdd [21] was introduced.

Both aforementioned sets were used to test our system in order to both preserve our work's comparability with its previous counterparts and to negate some drawbacks of kd99 in our evaluation.

A. Evaluation of Intrusion Detection precision

Initially proper features should be selected out of 41 features of KDD (Such features are listed in APPENDIX A).

Results from implementation of 1st phase of the proposed algorithm are summarized in Table I. For MA, number of population, mutation probability and crossover probability were selected as 50, 0.02 and 0.8 respectively and DSCG local searching function was used.

TABLE II: Selected Features of Each Class (for Feature Names, See Appendix A)

Class	Identifications for KDD99	Identifications for NSL-kdd
Normal	5,23,6,36,12	5,3,6,30,4
DOS	23,3,6,12,36,24	23,3,6,12,36
Probe	5,6,41,27,35,29	5,6,27,35,12
U2R	3,33,17,14	3,33,17,14,32
R2L	3,36,10	3,36,37,22

Training part of kdd99 was used for training phases and test part of it was employed for testing purpose. Results of the proposed system are presented and compared with the similar systems in Table II.

TABLE II: Comparison of Our Results with Our Kdd99 Relevant Studies

Class	Our	GACL	GA	C5.0	CTREE
Normal	97.12	96.14	98.34	99.5	92.78
DOS	98.1	96.68	99.33	97.1	98.91
Probe	80.23	85.77	93.95	83.3	50.35
R2L	40.13	30.3	5.86	8.4	7.41
U2R	65.12	75.71	63.64	13.2	88.13

NSL-Kdd is, in fact, selected records of kdd99. Since there are no iterative records in training set of this dataset and training algorithms effectiveness is not based upon their decisions about iterative records, such algorithms made lower faults during offering multiples. Similarly number of selected records is based on the classification hard scattering in kdd99, thus training algorithms precision is analyzed in a larger and more precise span. In our dataset, numbers of records in both test and training sets are reasonable, thus intricate methods would be implemented without random selection of dataset records [13].

TABLE III: EFFECTIVENESS OF THE PROPOSED ALGORITHM ON NSL-KDD AND COMPARING IT WITH SEVEN ALGORITHMS BASED ON SUCCESSFULPREDICTION FIELD

successful Prediction / Class	0	1-5	6-10	11-15	16-20	21
Normal	N/A	33.33	90.11	78.46	75.08	98.78
DOS	71.4	52.55	55.56	16.56	58.15	92.88
Probe	0	15	37.9	77.77	72.88	68.42
R2L	86.6	56.44	59.24	49	77.87	N/A
U2R	78.57	62.86	33.33	33.33	N/A	N/A
Total	81.3	53.68	58.59	64.77	70.82	94.56

In order to evaluate system with NSL-Kdd dataset, initially training part of the dataset, KddTrain$^+$, was used for training phase and then test part, KddTest$^+$, was used to achieve the final result.

For NSL-Kdd dataset, SuccesfulPrediction field is a number in the range of 0 to 21 which indicates the number of algorithms among seven algorithms implemented in [13] (for three times) on NSL-kdd dataset, which have had successful classifications. Moreover, this field is a criterion to detect difficulty of classification of present records of NSL-Kdd.

In table 3, system precision in terms of SuccessfulPrediction field is shown. For instance, for R2L class in 0, our system has succeeded to classify 86.60% of records correctly. It is worthy to say that 0 in the mentioned field indicates that all seven algorithms have failed to classify these records in all three times.

N/A value in table 3 shows that there is no record for the mentioned class in SuccessfulPrediction field.

B. Evaluation of Time Complexity of Algorithm Implementation

In the first phase, using MA, proper features of all five current classes of kdd were extracted. This algorithm used 50 individuals and after 20000 function evaluations, i.e. eq. (1), the algorithm was converged and the proper set was resulted.

In Eq. (1), Q (x_i,x_j) is already stored in a two-dimensional array in which one can access all x_is and x_js through O(1). Since I_m is equal to number of features in the worst case; then $\sum_{\forall j \in I_m} I(Y; X_j)$ is measurable with O(m). Therefore, the runtime of our algorithm for each class is measured as below:

Number of operations = 20000 * 41 = 820000 (5)

Therefore, for five implementations for each class, a total number of 6560000 operations are needed. Using a 2GHz processor with 1GB memory, implementing this algorithm takes 12 to 13 minutes approximately and five implementations takes about 60-65 minutes. It is worthy to mention that using Memetic has led to a decrease in full search time, which was 4×2^{41}.

In the second phase, for obtaining classifier function, MA with 50 individuals was used and proper results were obtained after 20000 function evaluations which have been offered in Fig. 2.

Time complexity of implementing evaluation function for m features and n training records is in the order of O(mn). That's because time complexity of IS-attack function is in the order of O(m) and it is implemented as large as n times, i.e. equal to the number of training set records. As a result, total number of required operations for each Kdd class is equal to 20000*n*m.

The last training phase (3rd phase) is as $O(N^4)$ concerning used Bayesian networks with irregular nodes [23].

For test phase (4th phase) detection of attack or normal nature of connections is depended on number of selected features and since 41 features are selected in the worst case, this time is very trivial. However, since this is the only part which is conducted online, it highly affects the effectiveness of our intrusion detection system, but selecting an optimum subset of features in the training phase, we have optimized the time needed for this part of the algorithm as possible.

VIII. CONCLUSION

Anomaly-based intrusion detection system was presented in this study which its detection capacity is acceptable in contrast of other systems. Using Bayesian networks, which act based on information theory, helped results of each class to be combined well and MA optimized the algorithm's run time. In order to evaluate the performance of the proposed system, results were obtained on KDD and NSL-Kdd datasets.

This study aimed at offering a model for anomaly detection engine; however future works include using signature-based intrusion detection systems as a supervisor to form a proper dataset with operational environment.

Signature-based intrusion detection systems have a trivial false positive rate but their detection rate is very low because they are able to detect only those attacks which follow their pattern. Therefore, when a signature-based intrusion detection system recognizes a connection as a single attack, it is reliable mostly and it may even be used as an attack label to train an anomaly detection system. It helps us to train the system online as seen in the architecture of Fig. 3.

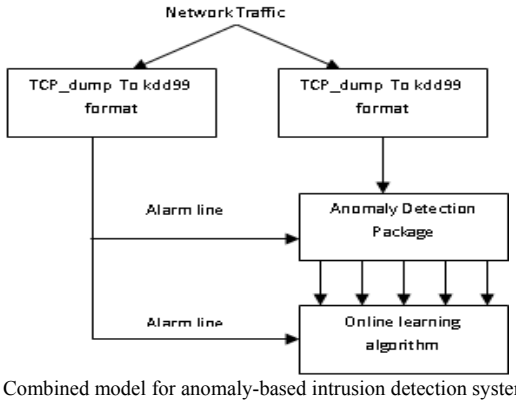

Fig. 3. Combined model for anomaly-based intrusion detection system and signature-based detection system

For datasets of this study only attacks were being classified and normal traffic has not been classified. In the case of using a dataset which classifies both attacks and normal traffic, modeling network's normal behavior will be easier; thus both precision and effectiveness of the system will be increased.

APPENDIX

Names and identifications (id) of selected features here we want to keep the order of features from the original KDD CUP'99 data set.

TABLE IV: KDD FEATURES

ID	fName
3	Service
4	Flag
5	src_bytes
6	dst_bytes
10	Hot
12	logged_in
14	root_shell
17	num_file_creations
22	is_guest_login
23	Count
24	srv_count
27	rerror_rate
29	same_srv_rate
30	diff_srv_rate
32	dst_host_count
33	dst_host_srv_count
35	dst_host_diff_srv_rate
36	dst_host_same_src_port_rate
37	dst_host_srv_diff_host_rate
41	dst_host_srv_rerror_rate

REFERENCES

[1] R. Coolen and H.A.M. Luiijf, "intrusion Detection: Generics and State-of-the-Art," *Resarch and Organization (RTO) Technical Report 49*, 2002.

[2] G. Gu, P. Fogla, D. Dagon, W. Lee, and B. Skoric. "Towards an information-theoretic framework for analyzing intrusion detection systems," *In Proceedings of the 11th European Symposium on Research in Computer Security (ESORICS'06)*, September 2006.

[3] M.A. Hall and G. Holmes, "Benchmarking Attribute Selection Techniques for Discrete Class Data Mining," *Proc. IEEE Trans. Knowledge and Data Eng.*, 2002.

[4] H.C. Peng, F. Long, and C. Ding , "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," *Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[5] Huan Liu and Motoda Hiroshi, "Feature Selection for Knowledge Discovery and Data Mining," *Proc. The Springer International Series in Engineering and Computer Science*, vol. 454, ISBN: 978-0-7923-8198-3, 1998.

[6] Huan Liu and Lei Yu, "Toward integrating feature selection algorithms for classification and clustering," *Proc.Knowledge and Data Engineering, IEEE Transactions* on vol.17, no.4, pp. 491- 502, (April 2005).

[7] Isabelle Guyon and Andr´e Elisseeff, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research 3 2003*, pp 1157-1182, 2003.

[8] G. Qu, S. Hariri and M. Yousif, "A new dependency and correlation analysis for features". *Proc. Knowledge and Data Engineering, IEEE Transactions on*, vol.17, no.9, pp. 1199- 1207, Sept. 2005.

[9] Hai Nguyen, K. Franke and S. Petrovic, "Improving Effectiveness of Intrusion Detection by Correlation Feature Selection," *Proc. Availability, Reliability, and Security, 2010. ARES '10 International Conference on* , vol., no., pp.17-24, 15-18 , Feb. 2010

[10] S. Das, "Filters, Wrappers and a Boosting-Based Hybrid for Feature Selection,"*Proc. 18th Int'l Conf. Machine Learning*, pp. 74-81, 2001.

[11] H. Liu and H. Motoda, "Computational Methods of Feature Selection," *Proc.Boca Raton: Chapman & Hall/CRC*, 2008.

[12] X.Qu, S.Hariri and M.Yousif, "An Efficient Network Intrusion Detection Method Based on Information Theory and Genetic Algorithm", *Proceedings of the 24th IEEE International Performance Computing and Communications,* 2005.

[13] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," in *Proc. Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, 2009.

[14] J. McHugh, "Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory," in *Proc.ACM Transactions on Information and System Security*, vol. 3, no. 4, pp. 262–294, 2000.

[15] S. Owais, V. Snasel, P. Kromer and A. Abraham, "Survey: Using Genetic Algorithm Approach in Intrusion Detection Systems Techniques," in *Proc. Computer Information Systems and Industrial Management Applications*, 2008. CISIM '08. 7th, vol., no., pp.300-307, 26-28 June 2008.

[16] Ulrike VÖllinger, Erik Lehmann and Rainer Stark, "Using Memetic Algorithms for the Solution of Technical Problems," *Proc. World Academy of Science, Engineering and Technology 59*, pp 428-433, 2009.

[17] Nicholas J. Radcliffe and Patrick D. Surry, "Formal Memetic Algorithms". *Proc. Evolutionary Computing: AISB Workshop*, Ed: T.C. Fogarty, Springer-Verlag LNCS 865, pp1-16, (1994)

[18] R.E. Neapolitan, "Probabilistic reasoning in expert systems: theory and algorithms", *JohnWiley & Sons*, 1990.

[19] G.F. Cooper, "Computational complexity of probabilistic inference using Bayesian belief networks" *Proc. In Artificial Intelligence*, 42 (pp. 393-405), 1990.

[20] "KDD Cup 1999," Available on: http://kdd.ics.uci.edu/databases/ kddcup99/kddcup99.html, January 2010.

[21] "Nsl-kdd data set for network-based intrusion detection systems," Available on: http://nsl.cs.unb.ca/NSL-KDD/, May 2010.

[22] H.M. Shirazi, and Y. Kalaji, "An Intelligent Intrusion Detection System Using Genetic Algorithms and Features Selection," *Majlesi Journal of Electrical Engineering,* vol. 4,no. 1, pp 33-43, March 2010.

[23] Jie Cheng, David Bell and Weiru Liu," sa Learning Bayesian Networks from Data: An Efficient Approach Based on Information Theory," 1997.

[24] Daniel T. Larose, "Discovering knowledge in data: an introduction to data mining," *Published by John Wiley & Sons*, Inc.2005 ISBN 0-471-66657-2, 2005.

[25] "Waikato environment for knowledge analysis (weka) version 3.5.7."Available on: http://www.cs.waikato.ac.nz/ml/weka/, Sep, 2010.