Neural Comput & Applic (2012) 21:1185–1190
DOI 10.1007/s00521-010-0487-0

ORIGINAL ARTICLE

Intrusion detection using reduced-size RNN based on feature grouping

Mansour Sheikhan · Zahra Jadidi · Ali Farrokhi

Received: 17 August 2010 / Accepted: 8 November 2010 / Published online: 25 November 2010

Abstract Intrusion detection is well-known as an essential component to secure the systems in Information and Communication Technology (ICT). Based on the type of analyzing events, two kinds of Intrusion Detection Systems (IDS) have been proposed: anomaly-based and misuse-based. In this paper, three-layer Recurrent Neural Network (RNN) architecture with categorized features as inputs and attack types as outputs of RNN is proposed as misuse-based IDS. The input features are categorized to basic features, content features, time-based traffic features, and host-based traffic features. The attack types are classified to Denial-of-Service (DoS), Probe, Remote-to-Local (R2L), and User-to-Root (U2R). For this purpose, in this study, we use the 41 features per connection defined by International Knowledge Discovery and Data mining group (KDD). The RNN has an extra output which corresponds to normal class (no attack). The connections between the nodes of two hidden layers of RNN are considered partial. Experimental results show that the proposed model is able to improve classification rate, particularly in R2L attacks. This method also offers better Detection Rate (DR) and Cost Per Example (CPE) when compared to similar related works and also the simulated Multi-Layer Perceptron (MLP) and Elman-based intrusion detectors. On the other hand, False Alarm Rate (FAR) of the proposed model is not degraded significantly when compared to some recent machine learning methods.

Keywords Partial connection · Recurrent neural network · Intrusion detection · Feature grouping

M. Sheikhan (✉)
Department of Communication Engineering,
Faculty of Engineering, Islamic Azad University,
South Tehran Branch, P.O. Box 11365-4435, Tehran, Iran
e-mail: msheikhn@azad.ac.ir

Z. Jadidi · A. Farrokhi
Department of Electronic Engineering,
Islamic Azad University, South Tehran Branch, Tehran, Iran
e-mail: z_jadidi@azad.ac.ir

A. Farrokhi
e-mail: ali_farrokhi@azad.ac.ir

1 Introduction

In recent decades, malicious behavior by some Internet users has prompted researchers to work on various intrusion detection techniques. Based on the source of information, two kinds of Intrusion Detection Systems (IDS) have been proposed: host-based and network-based [1]. Host-based IDS is served on host computer and network-based IDS monitors data exchanged between computers. On the other hand, if analyzing of events is considered, two kinds of IDS exist: anomaly-based [2] and misuse-based [3]. Anomaly-based IDS detects activities that differ from established patterns for users, and misuse-based IDS compares users' activities with the known behaviors of attackers.

Many soft computing approaches have been applied to the intrusion detection field. In this way, the anomaly-based detection techniques can be classified into three main categories: statistical-based [4], knowledge-based [5], and machine learning (e.g., Bayesian networks [6], Markov models [7], Artificial Neural Networks (ANNs) [8, 9], fuzzy logic [10, 11], genetic algorithms [12] and clustering and outlier detection [13]).

The detection techniques that are used in misuse-based IDS can also be classified into three similar categories: statistical-based [14], knowledge-based [15, 16], and machine learning (e.g., Bayesian networks [17], ANNs [18–22], fuzzy

logic [10], genetic algorithms [23], clustering [24], decision trees [25, 26], and hybrid systems [27–30]).

In this paper, a reduced-size structure of Recurrent Neural Network (RNN), based on the grouping of features, is used for misuse detection. Due to size reduction in RNN, training speed and convergence are improved. Thus, a fast IDS is reached which is effective in terms of Detection Rate (DR) and Cost Per Example (CPE).

International Knowledge Discovery and Data mining group (KDD) data set [31] is used for training and test of the proposed model in this study. Each connection in KDD is characterized by 41 features and a label which specifies the status of connection records (normal or a specific attack type). These features are used as the inputs of RNN and grouped into four categories: basic features (B-F), content features (C-F), time-based traffic features (TT-F), and host-based traffic features (HT-F). The RNN has five outputs, one of which indicates normal class (no attack). The other four outputs of RNN represent the type of detected attacks: Denial-of-Service (DoS), Probe, Remote-to-Local (R2L), and User-to-Root (U2R). To reduce the size and computational complexity of RNN-based IDS, the nodes of layers are partially connected based on the mentioned four feature categories.

Experimental results show that the proposed model is able to improve classification rate, particularly in R2L attacks. This method also offers better DR and CPE when compared to similar related works and also the simulated Multi-Layer Perceptron (MLP) and Elman-based IDS with the same number of hidden layer nodes. On the other hand, False Alarm Rate (FAR) of the proposed model is not degraded significantly when compared to some recent machine learning methods.

The remainder of this paper is organized as follows. Section 2 provides the KDD data set details. The architecture of the proposed model is introduced in Sect. 3. Simulations and experimental results are reported in Sect. 4. Conclusions are discussed in Sect. 5.

2 KDD intrusion data

In 1999, recorded network traffic from the Defence Advanced Research Project Agency (DARPA) data set was summarized into network connections with 41 features per connection [31]. This data set formed the benchmark provided by KDD. There are four main categories of attacks given in the KDD: DoS, Probe, R2L, and U2R. The KDD data set consists of three components: "10% KDD", "Corrected KDD", and "Whole KDD" [31] (Table 1).

As is common in literature, the analysis in this paper is performed on the "10% KDD" data set [21]. Each connection in KDD is characterized by 41 features (listed in Table 2). As mentioned earlier, these features are grouped into four categories: basic features, content features, time-based traffic features, and host-based traffic features.

Table 1 Number of samples in KDD data set

KDD dataset	Normal	DoS	Probe	U2R	R2L
10%	97,277	391,458	4,107	52	1,126
Corrected	60,593	229,853	4,166	70	16,347
Whole	972,780	3,883,370	41,102	52	1,126

Basic features can be derived from packet headers without inspecting the payload. In the content features, domain knowledge is used to assess the payload of the original Transmission Control Protocol (TCP) packets. Time-based traffic features are designed to capture properties that mature over a two-second temporal window. Host-based traffic features utilize a historical window estimated over the number of connections, instead of time. Therefore, they are designed to assess attacks that span in intervals longer than 2 s.

3 The proposed model

As mentioned before, a partially connected RNN with two hidden layers is used as misuse-based IDS in this work (Fig. 1). The categorized features defined in Sect. 2 are used as the inputs of RNN. As shown in Fig. 1, the connections between 41 input nodes and first hidden layer nodes are based on the categorization of features. The connections between the nodes of two hidden layers are considered partial. The RNN has five output neurons (representing the normal class and four attack types).

The features in the KDD data sets have different forms (discrete, continuous, and symbolic) with significantly varying resolution and ranges. Most pattern classification methods are not able to process data in such a format. Therefore, some preprocessing is required.

Symbolic-valued features, such as protocol_type (with 3 different symbols), service (with 70 different symbols), and flag (with 11 different symbols) are mapped to integer values ranging from 0 to $N-1$, where N is the number of symbols. Continuous features having smaller integer value ranges like wrong_fragment [0,3], urgent [0,14], hot [0,101], num_failed_logins [0,5], num_compromised [0,9], num_root [0,7468], num_file_creations [0,100], num_shells [0,5], num_access files [0,9], count [0,511], srv_count [0,511], dst_host_count [0,255], dst_host_srv_count [0,255] are also scaled linearly to the [0,1] range.

For the three features that span over a very large integer range, logarithmic scaling (base 10) is applied. The mentioned features are duration [0,58329], src_bytes

Table 2 Description and category of 41 features in KDD data set

Feature	Description	Category
duration	Duration of the connection (in s)	Basic
protocol_type	Type of the connection protocol	
service	Service on the destination	
flag	Status flag of the connection	
src_bytes	Number of bytes sent from source to destination	
dst_bytes	Number of bytes sent from destination to source	
land	1 if connection is from/to the same-host/port; 0 otherwise	
wrong_fragment	Number of wrong fragments	
urgent	Number of urgent packets	
hot	Number of "hot" indicators	Content
num_failed_logins	Number of failed logins	
logged_in	1 if successfully logged in; 0 otherwise	
num_compromised	Number of "compromised" conditions	
root_shell	1 if root shell is obtained; 0 otherwise	
su_attempted	1 if "su root" command attempted; 0 otherwise	
num_root	Number of "root" accesses	
num_file_creations	Number of file creation operations	
num_shells	Number of shell prompts	
num_access_files	Number of operations on access control files	
num_outbound_cmds	Number of outbound commands in a FTP session	
is_host_login	1 if the login belongs to the "hot" list; 0 otherwise	
is_guest_login	1 if the login is a "guest" login; 0 otherwise	
count	Number of connections to the same host as the current connection in the past 2 s	Time-based Traffic
srv_count	Number of connections to the same service as the current connection in the past 2 s	
serror_rate	Percent of connections that have "SYN" errors (same-host connections)	
srv_serror_rate	Percent of connections that have "SYN" errors (same-service connections)	
rerror_rate	Percent of connections that have "REJ" errors (same-host connections)	
srv_rerror_rate	Percent of connections that have "REJ" errors (same-service connections)	
same_srv_rate	Percent of connections to the same service	
diff_srv_rate	Percent of connections to different services	
srv_diff_host_rate	Percent of connections to different hosts	
dst_host_count	Number of connections having the same destination host	Host-based Traffic
dst_host_srv_count	Number of connections having the same destination host and using the same service	
dst_host_same_srv_rate	Percent of connections having the same destination host and using the same service	
dst_host_diff_srv_rate	Percent of different services on the current host	
dst_host_same_src_port_rate	Percent of connections to the current host having the same src port	
dst_host_srv_diff_host_rate	Percent of connections to the same service coming from different hosts	
dst_host_serror_rate	Percent of connections to the current host that have an S0 error	
dst_host_srv_serror_rate	Percent of connections to the current host and specified service that have an S0 error	
dst_host_rerror_rate	Percent of connections to the current host that have an RST error	
dst_host_srv_rerror_rate	Percent of connections to the current host and specified service that have an RST error	

[0,1.3 billion], and dst_bytes [0,1.3 billion], the spans of which have been reduced to [0,4.77] and [0,9.11], respectively. Other features are either Boolean, like logged_in, or continuous, like diff_srv_rate, in the range of [0,1]. No scaling is needed for these features. So, each of the mapped features are linearly scaled to the [0,1] range.

4 Experimental results

In this work, 49,402 records from "10% KDD" data set and 31,104 records from "Corrected KDD" data set are used as training and test sets, respectively (Table 3). Except for U2R test samples, the remaining sets have the

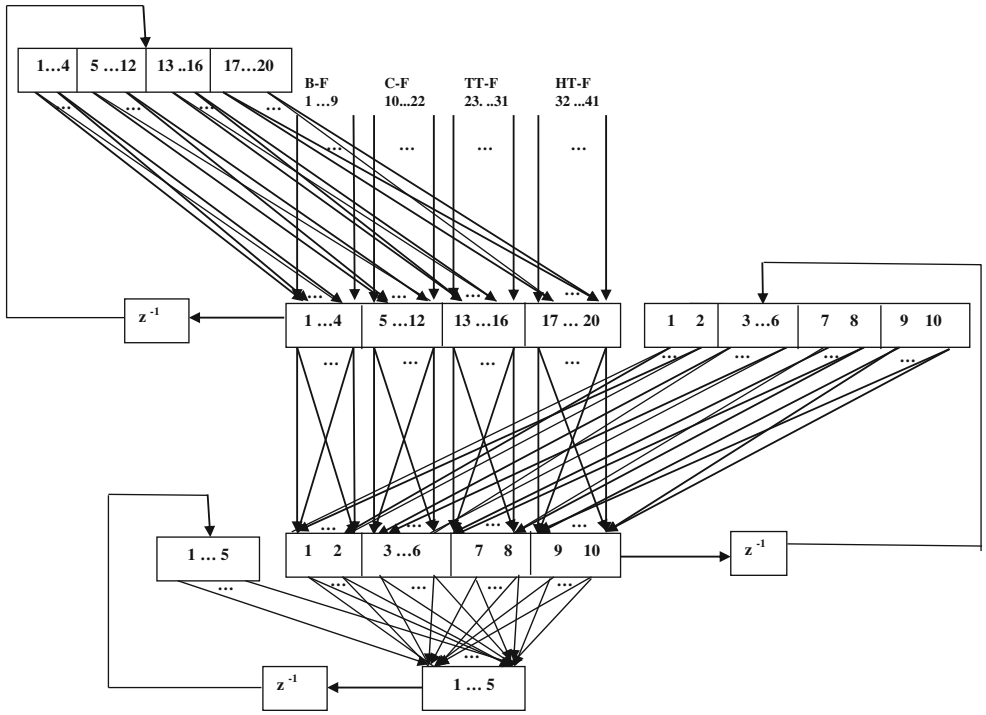

Fig. 1 Partially connected RNN-based IDS

Table 3 Size of the training and test sets

Class	Number of training samples	Number of test samples
Normal	9,727	6,059
DoS	39,145	22,985
Probe	411	417
U2R	6	24
R2L	113	1,619

Table 4 Cost matrix values for KDD

Actual	Predicted				
	Normal	DoS	Probe	U2R	R2L
Normal	0	2	1	2	2
DoS	2	0	1	2	2
Probe	1	2	0	2	2
U2R	3	2	2	0	2
R2L	4	2	2	2	0

same distribution, as different categories of attacks corresponding to KDD data sets.

The standard metrics that have been developed for evaluating IDS are DR and FAR as the two most common metrics. DR is computed as the ratio between the number of correctly detected attacks and the total number of attacks, while FAR is computed as the ratio between the number of normal connections that is incorrectly misclassified as attacks and the total number of normal connections. For the purpose of classifier algorithm evaluation,

Table 5 Confusion matrix and training time of proposed RNN model in comparison with MLP and Elman classifiers

Classifier	Actual	Predicted				
		Normal	DoS	Probe	U2R	R2L
Proposed RNN model	Normal	6,036	9	11	0	3
	DoS	95	22,882	8	0	0
	Probe	133	32	249	0	3
	U2R	16	7	1	0	0
	R2L	1,166	0	9	0	444
Training time = 1,383 s						
MLP	Normal	6,042	9	8	0	0
	DoS	3,366	19,615	0	0	4
	Probe	92	26	298	0	1
	U2R	17	1	3	0	3
	R2L	1,498	0	1	0	120
Training time = 1,905 s						
Elman	Normal	5,964	88	7	0	0
	DoS	1,251	21,734	0	0	0
	Probe	157	18	240	0	2
	U2R	23	0	0	0	1
	R2L	1,571	0	3	0	45
Training time = 723 s						

Table 6 Performance comparison of models in intrusion detection

Model	DR	FAR	CPE
Winner of KDD in 2000 [25]	91.8	0.60	0.2331
PNrule [32]	91.1	0.40	0.2371
Self organizing feature map (SOFM) ANN [33]	71.6	28.37	NR*
Jordan ANN [33]	62.9	37.09	NR*
RNN [33]	73.1	26.85	NR*
Data mining [15]	70–90	2	NR*
Clustering [15]	93	10	NR*
K-nearest neighbor [15]	91	8	NR*
Support vector machine (SVM) [15]	98	10	NR*
Hierarchical self organizing map (H-SOM) ANN [15]	90–91.5	7.6–14.5	NR*
Fuzzy association rules [29]	91	3.34	NR*
Proposed reduced-size RNN	94.1	0.38	0.1666
MLP (simulated in this work)	80.0	0.28	0.4170
Elman (simulated in this work)	87.9	1.57	0.2972

* not reported

another comparative measure is defined which is Cost Per Example (CPE) [32]. CPE is calculated using the following relation:

$$\mathrm{CPE} = \frac{1}{T}\sum_{i=1}^{m}\sum_{j=1}^{m}\mathrm{CM}(i,j)\cdot C(i,j) \quad (1)$$

where CM and C are confusion matrix and cost matrix, respectively. T represents the total number of test instances and m is the number of classes in classification. CM is a square matrix in which each column corresponds to the predicted class, while rows correspond to the actual classes. An entry at row i and column j, CM(i,j), represents the number of misclassified instances that originally belong to class i, although incorrectly identified as a member of class j. The entries of the primary diagonal, CM(i,i), stand for the number of properly detected instances. Cost matrix is similarly defined, that is to say entry $C(i,j)$ represents the cost penalty for misclassifying an instance belonging to class i into class j. Cost matrix values employed for the KDD classifier learning contest are shown in Table 4 [31].

The confusion matrix and training time of the proposed RNN model are reported in Table 5. The confusion matrices and training times of MLP and Elman-based neural classifiers, with the same number of nodes in hidden layers, are also reported in Table 5.

The performance of the proposed model has been compared to some other machine learning methods, in terms of DR, FAR, and CPE as well (Table 6). As shown in Table 6, the proposed RNN model performs better in terms of DR and CPE. FAR of the proposed IDS is not degraded significantly when compared to some recent machine learning methods.

5 Conclusions

In this paper, a partially connected RNN model with four groups of input features has been proposed as misuse-based IDS. Experimental results have shown that the reduced-size neural classifier has improved classification rates, especially for R2L attack category, when compared to other classifiers. The proposed model shows better performance in terms of DR and CPE when compared to some recent related works. The FAR metric has been improved in comparison with some recent machine learning methods, as well.

References

1. Sabhnani M, Serpen G (2004) Why machine learning algorithms fail in misuse detection on KDD intrusion detection data set. J Intelli Data Anal 6:1–13
2. Shon T, Moon J (2007) A hybrid machine learning approach to network anomaly detection. J Infor Sci 177:3799–3821
3. Chen Y, Abraham A, Yang B (2007) Hybrid flexible neural-tree-based intrusion detection systems. Int J Intell Syst 22:337–352
4. Ye N, Emran SM, Chen Q, Vilbert S (2002) Multivariate statistical analysis of audit Trials for host-based intrusion detection. IEEE Trans Comput 51:810–820
5. Garcia-Teodoro P, Diaz-Verdejo J, Macia-Fernandez G, Vazquez E (2009) Anomaly-base network intrusion detection: techniques, systems and challenges. J Comput Secur 28:18–28
6. Kruegel C, Mutz D, Robertson W, Valeur F (2003) Bayesian event classification for intrusion detection. In: The proceedings of the annual computer security applications conference, pp 14–23
7. Yeung DY, Ding Y (2003) Host-based intrusion detection using dynamic and static behavioral models. J Pattern Recognit 36:229–243
8. Cansian AM, Moreira E, Carvalho A, Bonifacio JM (1997) Network intrusion detection using neural networks. In: The

proceedings of the international conference on computational intelligence and multimedia applications, pp 276–280
9. Ramadas M, Ostermann S, Tjaden B (2003) Detecting anomalous network traffic with self-organizing maps. Recent advances in intrusion detection, RAID, Lecture notes in computer science (LNCS) 2820:36–54
10. Dickerson JE (2000) Fuzzy network profiling for intrusion detection. In: The proceedings of the North American fuzzy information processing society (NAFIPS) international conference, pp 301–306
11. Gomez J, Dasgupta D (2002) Evolving fuzzy classifiers for intrusion detection. In: The proceedings of the IEEE workshop on information assurance, pp 68–75
12. Song D, Heywood MI, Zincir-Heywood AN (2005) Training genetic programming on half a million patterns: an example from anomaly detection. IEEE Trans Evol Comput 9:225–239
13. Sequeira K, Zaki M (2002) ADMIT: anomaly-based data mining for intrusions. In: The proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining, pp 386–395
14. Biermann E, Cloeteand E, Venter LM (2001) A comparison of intrusion detection systems. J Comput Secur 20:676–683
15. Han SJ, Cho SB (2003) Detecting intrusion with rule-based integration of multiple models. J Comput Secur 22:613–623
16. Novikov D, Yampolskiy RV, Reznik L (2006) Artificial intelligence approaches for intrusion detection. In: The proceedings of the IEEE conference on systems, applications and technology, pp 1–8
17. Joshi MV, Agrawal RC, Kumar V (2001) Mining needless in a haystack: classifying rare classes via two-phase rule induction. In: The proceedings of the ACM SIGMOD conference on management of data, pp 91–102
18. Debar H, Dorizzi B (1992) An application of recurrent network to an intrusion detection system. In: The proceedings of the international joint conference on neural networks, pp 478–483
19. Kayacik G, Zincir-Heywood N, Heywood M (2003) On the capability of an SOM-based intrusion detection system. In: The proceedings of the international joint conference on neural networks, pp 1808–1813
20. Golovko V, Vaitsekhovich L, Kochurko P, Rubanau U (2007) Dimensionality reduction and attack recognition using neural network approaches. In: The proceedings of the international joint conference on neural networks, pp 2734–2739
21. Beghdad R (2008) Critical study of neural networks in detecting intrusions. J Comput Secur 27:168–175
22. Sheikhan M, Sha'bani AA (2009) Fast neural intrusion detection system based on hidden weight optimization algorithm and feature selection. World Appl Sci J 7(Special Issue of Computer & IT):45–53
23. Lin Y, Chen K, Liao X (2004) A genetic clustering method for intrusion detection. J Pattern Recognit 37:924–927
24. Denning DE (1987) An intrusion-detection model. IEEE Trans Softw Eng 13:222–232
25. Pfahringer B (2000) Winning the KDD 99 classification cup: bagged boosting. J SIGKDD Explor 1:65–66
26. Levin I (2000) KDD classifier learning contest: LLSoft's results overview. J SIGKDD Explor 1:67–75
27. Mukkamala S, Janoski G, Sung AH (2002) Intrusion detection using neural networks and support vector machines. In: The proceedings of the international joint conference on neural networks, pp 1702–1707
28. Abadeh MS, Habibi J, Lucas C (2005) Intrusion detection using a fuzzy genetic–based learning algorithm. J Netw Comput Appl 30:414–428
29. Tajbakhsh A, Rahmati M, Mirzaei A (2009) Intrusion detection using fuzzy association rules. J Appl Soft Comput 9:462–469
30. Sheikhan M, Jadidi Z (2009) Misuse detection using hybrid of association rule mining and connectionist modelling. World Appl Sci J 7(Special Issue of Computer & IT):31–37
31. KDD Cup 1999 Data. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. Accessed July 2008
32. Agrawal R, Joshi MV (2000) PNrule: a new framework for learning classifier models in data mining (a case-study in network intrusion detection). IBM research division, report no. RC-21719
33. Beghdad R (2007) Training all the KDD data set to classify and detect attacks. Neural Netw World 17:81–91