Hamming Net and LVQ Neural Networks for Classification of Computer Network Attacks: A Comparative Analysis

Lília de Sá Silva
Divisão de Desenvolvimento de Sistemas Solo
Instituto Nacional de Pesquisas Espaciais
São José dos Campos - SP
lilia@dss.inpe.br

Adriana C. Ferrari dos Santos
Laboratório de Computação Aplicada
Instituto Nacional de Pesquisas Espaciais
São José dos Campos - SP
adriana.ferrari@lac.inpe.br

Antonio Montes
Centro de Pesquisas Renato Archer - CenPRA
Campinas - SP
antonio.montes@cenpra.gov.br

José Demísio da Silva Simões
Laboratório de Computação Aplicada
Instituto Nacional de Pesquisas Espaciais
São José dos Campos - SP
demisio@lac.inpe.br

Abstract

This paper presents a comparative analysis of results obtained when applying Hamming Net and LVQ (Learning Vector Quantization) classifiers neural networks to recognize attack signatures in datasets. Strings similar to those located on payload field in computer networks packets are inserted in these neural networks for pattern classification. Since 2004, when it was presented for the first time, ANNIDA system (Artificial Neural Network for Intrusion Detection Application) has been improved. Although the very sufficient results presented by the application of Hamming Net neural network in this system, researches have continued to find other classification and data modeling methods in order to compare new results with those obtained from Hamming Net usage. As the LVQ neural network also uses based-competition techniques and presents architecture more simple than the Hamming Net architecture, it was decided to implement the LVQ to do the comparative tests. Tests results and analysis are presented in this paper, as well some proposals for future researches.

1. Introduction

Success of the attack techniques is very dependant on the search of information on the target machine and network activities, such as: used operational system, opened service ports, installed vulnerable software and user accounts, in special, with the access password. By means of the target network recognition, named scanning, using specific software or social engineering techniques, it is possible to explore the vulnerabilities of the operational systems or communication network protocols [7].

The exploration of systems and network resources vulnerabilities can be identified through different techniques, such as monitoring of system event logs, and investigation of payload data carried by the network packets.

In the network packet payload can reside attack information represented by malicious strings named "attack signatures" [5].

Detecting attack signatures is objective of most of the current commercial intrusion detection systems (IDS – Intrusion Detection System) [7]. These systems contain a database of string sequences that make up an attack information and, in general, are built based on rules and filters [15].

Other IDS signature-based have been implemented using neural networks techniques to search more efficient identification of attack patterns in large databases like those of computer network traffic [13][16][17]. Nowadays several studies are being accomplished in order to use neural networks to detect attacks to computer networks [11][12][14][17][18].

Neural networks present high processing power due to its distributed parallel structure [2], being useful to classify data in large datasets of computer network traffic.

Recent studies have been conducted at Instituto Nacional de Pesquisas Espaciais (INPE) in São José dos Campos, São Paulo, Brazil, to improve the ANNIDA application which prototype was built in 2004 [8] with the scope to find attack patterns in simulated datasets of network packets. By the end of 2005 several tests were made in this system to classify data, based on Hamming Net application. The basic characteristics of ANNIDA can be found in [8][9][10].

Currently, ANNIDA has been enhanced in several aspects, mainly, employing more efficient data classification neural network techniques and using data structures more appropriated to represent the involved knowledge and data storing.

In this paper, the purpose is to present a comparative analysis between results obtained by using LVQ and Hamming Net [1] neural networks.

Some basic necessary concepts related to computer network security are presented in section 2.

Data modeling to input into LVQ e Hamming Net networks is described in section 3, as well the challenges faced to build the models. In section 4, is presented the data classification process of ANNIDA.

ANNIDA system test results using Hamming Net and LVQ to classify attack signatures are discussed in section 5. Finally, this paper is concluded in section 6.

2. Basic concepts

Network packet payload, Snort signatures and signatures contents are terms used in computer network security area that are described in the next paragraphs.

Data packets traveling in the network carry useful information among the interconnected computers. In the figure 1 is presented an example of TCP/IP network packet encapsulating the TCP protocol. It is made up of three parts [5] [6] [19]:

- IP (Internet Protocol) header;
- TCP (Transfer Control Protocol), UDP (User Datagram Protocol) or ICMP (Information Control Message Protocol) header, depending on the network service in usage, and
- TCP (Transfer Control Protocol), UDP (User Datagram Protocol) or ICMP (Information Control Message Protocol) payload.

For example, HTTP application uses TCP header and payload, while, Telnet network service contains UDP header and payload.

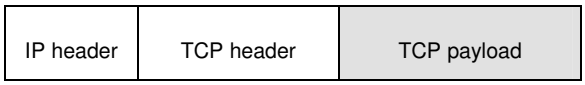

Figure 1. TCP/IP packet format

IP and TCP (UDP or ICMP) header fields contain information such as source and destination IP addresses to deliver the packet, network service type in usage, packet size in bytes, delivery data packet examination and others.

In TCP/IP packet payload field are located payload data that are transferred from source to destination computer. In the payload dataset are found several strings in ASCII or hexadecimal or in both formats. There are attack patterns that cannot be full or partially represented by characters ASCII, so they are written in hexa format. Among these strings, may exist normal or malicious strings, that is, known attack patterns used by attackers to launch effective or attempted attacks and to discover machine vulnerabilities.

These malicious strings are known as attack signatures [3][5] and are found stored in intrusion detection systems signature-based, such as the popular Snort [3][4], which maintains its own signature base (Snort signatures) in permanent and reliable updating.

Snort signatures have structure similar to that shown in figure 2.

Sample rule:

```
alert udp $EXTERNAL_NET any -> $HOME_NET 53 \
(msg: "EXPLOIT BIND_tsig Overflow Attempt"; \
content: "|00 FA 00 FF|"; content: "/bin/sh";);
```

Sample output:

```
02/22-15:33:19.472301 ATTACKER:1024 -> VICTIM:53
UDP TTL:64 TOS:0x0 ID:6755 IpLen:20 DgmLen:538
Len: 518

<lines omitted to condense output>

00 3F 90 E8 72 FF FF FF 2F 62 69 6E 2F 73 68 00 .?..r.../bin/sh.
0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D ................
1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D .. !"#$%&'()*+,-
2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B 3C EB ./0123456789:;<.
07 C0 00 00 00 00 00 3F 00 01 02 03 04 05 06 07 .......?........
08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 ................
18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 ........ !"#$%&'
28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37 ()*+,-./01234567
38 39 3A 3B 3C EB 07 C0 00 00 00 00 00 3F 00 01 89:;<........?..
02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 ................
D8 FA FF BF D8 F7 FF BF D0 7C 0D 0B 04 F7 10 40 .........|.....@
22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 "#$%&'()*+,-./01
32 33 34 35 36 37 38 39 3A 3B 3C EB 07 C0 00 00 23456789:;<.....
00 00 00 3F 00 01 02 03 04 05 06 07 08 09 0A 0B ...?............
0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B ................
1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B .... !"#$%&'()*+
2C 2D 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B ,-./0123456789:;
3C EB 07 C0 00 00 00 00 00 00 00 FA 00 FF <.............
```

Figure 2. Example of signature snort with two '*content*' fields

The main Snort signature information used in this work is the field named 'content', that alone or associated with other 'content' fields when existing, make up a single signature attack. These fields are named in this paper "signature contents".

In ANNIDA system the known signatures contents (also called attack patterns) are the patterns to be searched in the simulated network packet payload datasets. The simulated data are built from Snort signature contents strings added with noises.

3. Data modeling

On previous ANNIDA implementations [8][9][10], using Hamming Net, the exemplars and input patterns used to data classification were modeled and processed according to the usage of:

- only two neural network layers: input and output;
- fixed weights with w=d/2 value, where d corresponding to exemplar patterns (Snort signature contents);
- input pattern formed by a unique element in 14 bit-bipolar format;
- set of exemplar patterns formed by elements in 14-bit bipolar format;
- calculus of similarity measure between input and exemplar patterns using the Hamming Distance;
- exemplar patterns (multiple contents strings) extracted from Snort signature files;
- character array with 72 positions to store network packet simulated data (exemplar patterns);
- classification rule: if it occurs 100% of similarity between input and exemplar pattern, then to alert "it was detected one known attack string ";
- threshold to establish the similarity level desired in the search, allowing to discard matches with low similarity and to consider close matches. Thus, it is possible to identify not only well-known malicious strings but strings which similarity is near to those known (in the attack variation).

The challenges to model the data using Hamming Net were to:

- model and treat the data correctly, for insertion and processing in the neural network. It was modeled the neural network input in 14-bit bipolar format, in order to represent in bipolar a number of 4 digits generated by the folding-shift hash algorithm. The 4-digit size was considered sufficient to represent the strings analyzed. The hash algorithm was used in order to represent any size of string in a unique way, standardizing the input size to the neural network;
- process the neural network in several steps or level, as illustrated in figure 3, where each level represents a data column in the attack signature content set to be compared with each input pattern;

The LVQ neural network application in the ANNIDA required a more simplified data modeling consisting of:

- only two neural network layers: input and output;
- fixed weights with w=c value, where c corresponding to exemplar patterns (Snort signature contents);
- input pattern formed by a unique element in 4 numbers in decimal format;
- set of exemplar patterns formed by elements in 4 numbers in decimal format;
- calculus of similarity measure between input and exemplar patterns using the Euclidian Distance;
- exemplar patterns (multiple contents strings) extracted from Snort signature files;
- characters array with 72 positions to store network packet simulated data (exemplar patterns);
- classification rule: if it occurs 100% of similarity between input and exemplar pattern, then to alert "it was detected one known attack string ".

4. Data classification

ANNIDA data classification process is, basically, to find the exemplar class more similar to a given input, where the exemplar classes are the data in signature content files named s1, s2, and so on, according to the content amount of the larger attack signature considered.

The s1 signature file contains the first *content* of all lines in the Snort signature file processed, the contents located in the second position in the Snort signature file are arranged in s2 and so on, in order to have a set of associated files containing each one a column of *content* fields, where data in the corresponding lines in these files represents one attack signature.

All *contents* located on the same line in the associated signatures files represent a known attack signature, formed by a single or multiple *content*. For instance, if two *contents* compound a Snort signature, two *content* associated files (named levels) have to be processed to search malicious strings. The neural network acts finding malicious strings in each level of contents, but the sequence of strings for level, which represents an attack signature, is previously mapped with basis on the string sequences just as found in the Snort signature database. The idea is to look for patterns (strings) associated to an attack inside the packet's content by means of reading string by string.

In Figure 3, two strings "SITE" and "Cl3A5Cl" were classified by Hamming Net and LVQ neural networks in different test stages, according to the attack classes specified by the specialists in computer network security based on Snort signature files, and, the output is one attack signature with multiple *content* fields ("SITE Cl3A5Cl") was found.

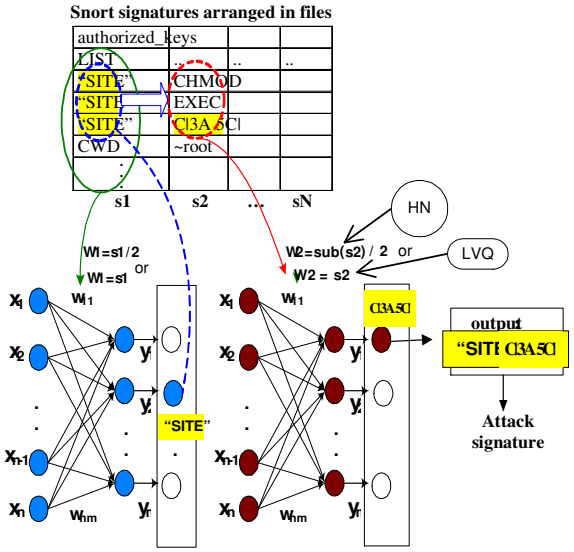

Figure 3. Processing of a signature in two levels

5. Comparative Analysis

Several tests were accomplished to classify attack signature using Hamming Net and LVQ neural networks in the ANNIDA.

Among the Snort signature classes tested some these are presented in Table 1: finger, icmp, ddos, dns, ftp, oracle e exploit.

Class	Input size (char)	Content Level	Exemplar No.	Classification (HN and LVQ)	Time (sec) HN	Time (sec) LVQ
finger	99	1	NUMEXEMP= 15 MAXLEN= 12	100%	9,80	5,01
icmp	312	1	NUMEXEMP=11 MAXLEN=66	100%	8,45	4,08
		2	NUMEXEMP=11 MAXLEN=66	100%	9,20	5,32
ddos	78	1	NUMEXEMP=227 MAXLEN= 24	100%	50,8	39,7
dns	60	1	NUMEXEMP=19 MAXLEN=108	100%	8,35	3,66
		2			8,46	3,67
ftp	200	3	NUMEXEMP= 69 MAXLEN=24	100%	25,4	17,5
oracle	67	3	NUMEXEMP=200 MAXLEN=70	90%	45,7	26,9
exploit	89	4	NUMEXEMP=28 MAXLEN=24	100%	13,6	9,59
		5	NUMEXEMP=77 MAXLEN=12	90%	19,8	9,94

Table 1. Tests results of ANNIDA

For each signature class analyzed, the following parameters were used:

- type of signature classes;
- input string sizes;
- level of signature contents;
- Amount of exemplar input and content fields;
- Hit rate percentage of classification;
- Application processing time, using both Hamming Net and LVQ.

According to the test results, it was noticed that:

- The LVQ processing time was smaller than the processing time of Hamming Net;
- For both neural networks was found the same classification hit rate;
- As bigger the number of content level, exemplars and content sizes, the ANNIDA processing time increases;
- Reduced number of classifications occurred with precision smaller than 100%;
- Collision between input values of the neural networks happened.

6. Conclusion

Analyzing the test results of the ANNIDA, the following conclusions were obtained:

- The smaller processing time of LVQ is due to the small hash key used – 4 digits (4 input units), different of the processing of the Hamming Net with a 14 bit bipolar input ;
- Also, the smaller processing time of LVQ is due to the learning with weights insertion, that is, the weights values used

are the proper reference patterns (exemplars);

- As both networks have similar classification technique, based on competition, the precision of results were equivalent;
- The classification error rate is due to the collisions generated between the hash keys. Pairs of different strings (input and exemplar) were classified as similar (positive-false), for reason of the normalization strategy to format the input and exemplar data for the neural networks. The creation of hash keys with 4 digits to the LVQ and the hash keys in 14-bit bipolar to the Hamming Net, using the simple hash algorithm named fold-shifting, did not present satisfactory results.

The next challenges to be faced in this work are to:

- improve the performance of the application by means of a method more refined to store and manipulate data, instead of store data in files;
- introduce network packet real data to be classified;
- solve the problem of data collision modeled to the neural networks, through the a more efficient method of hashing.

7. References

[1] Fausset, L., Fundamentals of Neural Networks: architectures, algorithms, and applications", Prentice Hall, New York, 1994, chapter 4, pp. 164-169 and pp.187-190.

[2] Hayking, S., Redes Neurais Princípios e Práticas, 2nd Ed, Bookman, Porto Alegre, 2001, chapter 1, pp. 27.

[3] Caswell, B., J. Beale, J. C. Foster, and J. Posluns, Snort 2 - Sistema de Detecção de Intruso Open Source, Editora Alta Books, Rio de Janeiro, 2003, chapter 1, pp. 23.

[4] SNORT - http://www.snort.org/, page accessed on jan 2006.

[5] Northcutt, S., and J. Novak, Network Intrusion Detection, Third Ed., New Riders, 2002, chapter 13, pp. 237-243, chapter 12, pp. 234.

[6] Stevens, W.R, TCP/IP Illustrated Vol. 1 – Addison Wesley, 2002, chapter 17, pp. 225.

[7] Carmona, T., Segredos da Espionagem Digital, Digerati Books, São Paulo, 2005, chapter 1, pp.11-12.

[8] L.S. Silva, A.C.F. Santos, J.D.S. Silva, and A. Montes, "A Neural Network Application for Attack Detection in Computer Networks", IJCNN2004 International Joint Conference in Neural Networks, Budapest, Hungria, 2004.

[9] L.S. Silva, A.C.F. Santos, J.D.S. Silva, and A. Montes, "ANNIDA: Artificial Neural Network for Intrusion Detection Application – Aplicação da Hamming Net para Detecção por Assinatura", CBRN2005 VII Congresso Brasileiro de Redes Neurais, Natal, RN, Brasil, 2005. (in Portuguese)

[10] L.S. Silva, A.C.F. Santos, J.D.S. Silva, and A. Montes, "Estudo do uso da Hamming Net para Detecção de Intrusão", SSI2005 VII Simpósio de Segurança em Informática, Instituto Tecnológico de Aeronáutica (ITA), São José dos Campos, SP, 2005. (in Portuguese)

[11] R.M. Silva, and M.A.G.M. Maia, "Redes Neurais Artificiais Aplicadas a Detecção de Intrusos em Redes TCP/IP", SSI 2004 Simpósio de Segurança da Informação, Instituto Tecnológico de Aeronáutica (ITA), São José dos Campos, SP, 2004. (in Portuguese)

[12] S.C. Lee, and D.V. Heinbuch, "Training a neural-network based intrusion detector to recognize novel attacks Systems", Man and Cybernetics, Part A, IEEE Transactions on Vol. 31, Issue 4, Jul 2001, pp. 294 – 299.

[13] C. OZ, "Signature recognition and verification with artificial neural network using moment invariant method", Lecture Notes In Computer Science, 2005, Vol. 3497, pp. 195-202.

[14] C.L. Zhang, J. Jiang, and M. Kamel, "Intrusion detection using hierarchical neural networks", Pattern Recognition Letters, may 2005, v. 26 (6), pp. 779-791.

[15] S.J. Han., and S.B. Cho, "Detecting intrusion with rule-based integration of multiple models", Computers & Security, 2003, v. 22 (7), pp. 613-623.

[16] J. Cannady, and R.C. Garcia., "The application of fuzzy ARTMAP in the detection of computer network attacks", ICANN 2001 Artificial Neural Networks, Proceedings Lecture Notes In Computer Science, 2001, v. 2130, pp. 225-230.

[17] H. Lingxuan, and H. Zhijun, "Neural network-based intrusion detection systems", Proceedings of the Sixth International Conference for You Computer Scientist: in Computer Science and Technology in New Century, 2001, pp. 296-298.

[18] C.L. Zhang, J. Jiang, M. Kamel, "Comparison of BPL and RBF network in intrusion detection system", Computing Lecture Notes In Artificial Intelligence, 2003, v. 2639, pp. 466-470.

[19] Chaves, M.H.P, “Analise de Estado de Trafego de Redes TCP/IP para Aplicação em Detecção de intrusão”, Dissertação de Mestrado em Computação Aplicada, Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, SP, 2002. (in Portuguese)