Detecting Denial of Service Attacks with Bayesian Classifiers and the Random Neural Network

Gülay Öke, George Loukas, Erol Gelenbe

Abstract— **Denial of Service (DoS) is a prevalent threat in today's networks. While such an attack is not difficult to launch, defending a network resource against it is disproportionately difficult, and despite the extensive research in recent years, DoS attacks continue to harm. The first goal of any protection scheme against DoS is the detection of its existence, ideally long before the destructive traffic build-up. In this paper we propose a generic approach which uses multiple Bayesian classifiers, and we present and compare four different implementations of it, combining likelihood estimation and the Random Neural Network (RNN). The RNNs are biologically inspired structures which represent the true functioning of a biophysical neural network, where the signals travel as spikes rather than analog signals. We use such an RNN structure to fuse real-time networking statistical data and distinguish between normal and attack traffic during a DoS attack. We present experimental results obtained for different traffic data in a large networking testbed.**

I. Introduction

A denial-of-service attack (DoS attack) is an attempt to make a network resource unavailable to its legitimate users. Such attacks are generally distributed; the attacker typically acquires control of a large number of hosts, which are unaware that their machines are compromised, and orders them to simultaneously target the victim. Since the early 1990s and particularly the last six years, the majority of organisations with online presence have been victims of DoS attacks, with the repercussions ranging from simple nuisance to significant financial losses [27], endangerment of human life [28] and compromising of national security [29].

The extreme diversity of DoS attacks has produced similarly diverse protection proposals in the three aspects of DoS defence, namely the detection of the existence of an attack, the classification of the incoming flows as normal and DoS [20], and the corresponding response [15]. In this paper we concentrate on the problem of DoS detection. To provide a network with an effective system of protection against DoS attacks, one must first employ a method to detect such an attack. This would not be needed in the case of an ideal response architecture with proactive qualities that would render a DoS attack impossible, but such a system has not been built, and proactive solutions are usually too expensive resource-wise to operate in the absence of an attack.

A detection mechanism should monitor the traffic continuously and signal any developing attacks in the network. This should then trigger a response mechanism which will attempt to protect the network resources and maintain a satisfactory level of quality of service for the legitimate users. The success of a detection mechanism is determined by its probability of correct detection, false alarm, and missed detection, and on its ability to reach detection decisions quickly in real-time and consume minimal processing resources. The following is a summary of the existing proposals in the literature.

Intelligent learning techniques comprise a significant part of the current research on DoS detection. Neural networks [9], radial basis functions [16], support vector machines [10], fuzzy classifiers [21], and Adaptive Neuro-Fuzzy Inference Systems (ANFIS) [17] have all been applied for detection.

Normal internet traffic is known to be long-range dependent (LRD) and self-similar, while a DoS attack usually results in deviations from these statistical properties, which can be used for its detection if evaluated in real-time. Thus, various statistical techniques have been employed in DoS detection. In [12] the incoming traffic is classified as normal or attack by calculating its autocorrelation function. Xiang et al compute the variance of the Hurst parameter in consecutive time intervals to evaluate the change of self-similarity of the traffic [13]. In a similar approach, Feinstein et al [6] measure the entropy and chi-square to detect the existence of an attack.

Additionally, while the energy distribution of normal traffic is known to be relatively stationary, an attack usually results in changes in the energy distribution variance. Li and Lee [19] use wavelets to compute the variations in the energy distribution in the incoming traffic and compare them with a threshold, and Yang et al [11] determine flat bursts in the traffic with the continuous wavelet transform.

In this paper we present and compare four different implementations of multiple Bayesian classifiers combined with the Random Neural Network (RNN). Bayesian classifiers have been used before for DoS detection by Noh et al [7], but applied only on the rate of appearance of specific flags in the packets' headers, and by Chen et al [23], who used hypothesis testing on the spectral analysis of bitrate to detect only one very specific type of attack. In our work we present a more general approach which aggregates likelihood estimation of heterogeneous statistical features and combine them in a neural network structure. The RNN, proposed by Gelenbe is a computational paradigm, inspired by the random spiking behaviour of the biological neurons. The RNNs are computationally efficient structures and they represent a better approximation of the true functioning of a biophysical neural network, where the signals travel as spikes

Gülay Öke, George Loukas, and Erol Gelenbe are with the Intelligent Systems and Networks group, Dept of Electrical and Electronic Engineering, Imperial College London, SW7 2BT, UK (emails: {g.oke,georgios.loukas,e.gelenbe}@imperial.ac.uk).

rather than analog signals. The strong analogy between queuing networks and the RNN make it a powerful tool for dealing with problems where excitation and inhibition among problem inputs are prevalent. RNNs have been successfully applied in various problems, including image processing [4], pattern recognition [5], and optimisation [3]. They can be used in both feedforward and recurrent architectures. We use the feedforward version and evaluate our approach for different traffic data in a large networking testbed.

II. Multiple Bayesian Classifiers in DoS Detection

The Bayesian Decision theory is a major pattern recognition technique based on a probabilistic description of the underlying features of a problem. It aims to minimise the risks encountered by the decision taking process by evaluating the various tradeoffs between decisions [25]. For a classification problem of two categories (w_1 and w_2), the use of Bayesian classifiers entails evaluating the likelihood ratio, which is the ratio of the probability density functions $\Lambda(x) = \frac{f(x|w_1)}{f(x|w_2)}$, for the measured value x of the observation variable, and comparing it with a threshold T. Then, x is assigned to category w_1 if $\Lambda(x) > T$; otherwise it is assigned to category w_2 [24].

The task of DoS detection can be considered as a two-category classification problem, where w_1 corresponds to normal network condition and w_2 to existence of DoS attack. We have used multiple Bayesian classifiers to take individual decisions for the monitored features of the traffic and combined them in an information fusion phase to detect DoS attacks in incoming traffic. In the following sections we present our approach, including the selection of the input features, the offline statistical information gathering and decision taking.

A. Selecting the Input Features

The selection of useful and information bearing input features is vital for successful detection of DoS. Since DoS attacks aim at overwhelming the victim system's networking or processing capacity, the detection method should not further aggravate the situation by consuming too many resources. Thus, we chose the following statistical features, which represent both the instantaneous and long-term behaviour of the incoming traffic and are easy to measure:

- ***Bitrate***. An unusually high value of incoming bitrate is the most conspicuous property of flooding DoS attack. Although it is a very strong, if not the strongest indication of DoS, a similar condition is observed during flash crowds, when for some legitimate reason interest for a network resource increases dramatically. Due to its simplicity, the bitrate measurement and similar measurements such as the number of packets per flow [9] are often used in detection mechanisms.
- ***Increase in Bitrate***. Depending on its type, a DoS attack typically demonstrates sudden and sustained increases in the rate of the incoming traffic. For example, flooding attacks start with a long period of increasing bitrate, while in pulsing attacks, the incoming traffic undergoes consecutive periods of increasing and decreasing bitrate.
- ***Entropy***. The entropy is a measure of randomness or uncertainty of information. It has been reported in technical literature that the entropy of normal internet traffic and traffic under DoS attack differ significantly. Thus, Feinstein et al calculate the entropy of the amount of source IP addresses to detect attacks [6]. In our work, we compute the entropy of the value of the incoming bitrate at the nodes we monitor, as given by [1]: $E = -\sum_i f_i log f_i$, where f_i are the histogram values obtained for bitrate, as explained in Section II-C. This would yield a higher value when the probability distribution expands over a wider range of values, indicating an increase in uncertainty.
- ***Hurst Parameter***. It has been studied in detail in [22] that the self-similarity properties of normal and attack traffic are distinctively different. Since the Hurst parameter is an indicator of the self similarity of traffic, it can be used in DoS detection. Xiang et al [13] use the variations of the Hurst parameter of the number and the size of packets to detect attacks. In our approach we compute the actual value of the Hurst parameter for the incoming bitrate, for which we have used the (R/S) analysis, as described [14]. If x is the bitrate of the incoming traffic, n is the observation time, and N is the total number of observation points, then (R/S) is given by:

$$(R/S)_N = \frac{\max\limits_{1\le n\le N} \sum\limits_{n=1}^{N}(x-\bar{x}) - \min\limits_{1\le n\le N} \sum\limits_{n=1}^{N}(x-\bar{x})}{\sqrt{\dfrac{\sum\limits_{n=1}^{N}(x-\bar{x})^2}{N}}}$$

  The Hurst parameter and $(R/S)_N$ are related by $(R/S)_N = cN^H$, which for $c = 1$ becomes $H = log_N((R/S)_N)$.
- ***Delay***. Although a DoS attack is also expected to increase the packet delays as congestion builds up, to our knowledge it has not been used as an attack indicator. For the fastest and least invasive way to detect changes in the delays, the node we monitor sends constantly a small number of packets to all its direct neighbours. By measuring the average round trip time (RTT) for the acknowledgments to return, we have a clear indication of the congestion near the node.
- ***Delay Rate***. As with bitrate, depending on the type of the attack and for its whole duration, the packet delays are expected to undergo significant changes. Although we are not aware of an existing work using the change of the delay as a detection feature, we consider it as a natural next step.

B. Offline Statistical Information Gathering

The Statistical information gathering phase in our detection scheme consists of two steps: We first obtain the probability density function (pdf) values for normal and attack traffic and then evaluate the likelihood ratios. This information is collected offline at each victim candidate of the network, from available traffic data, known to belon to normal or DoS traffic. For each of the input feature of Section II-A, estimates of probability density function for both normal and attack traffic are obtained. We have t compute $f_{feature}(x|w_N)$ and $f_{feature}(x|w_A)$, where *featur* can be bitrate, bit acceleration, entropy, Hurst parameter delay and delay rate, x is the measured value of the featur from the available traffic data, w_N denotes the normal traffi and w_A the attack traffic. We have used the histogram method to calculate the estimates of the probability densit functions. With this method the range of observable value for a variable is divided into a number of intervals. Ther for each interval, we compute the ratio of the number o data points that fall into it to the total number of data point available [25].

After obtaining the probability density function estimate for each input for both traffic types, we compute the likeli hood ratios $l_{feature}$ for each feature: $\frac{f_{feature}(x|w_A)}{f_{feature}(x|w_N)}$, which will then be used in the decision taking mechanism (Section II-C). Actual values and likelihood ratios of the features are used also in the training of RNN.

C. Decision Taking Methods

We have designed the following four implementations of the decision taking process:

1) Average likelihood estimation: The actual values of the input features of section II-A are measured in real-time at each of the DoS victim candidates that we monitor.

For each feature, a likelihood ratio is obtained by resorting to the likelihood functions computed in II-B. The information collected from all of these features must be aggregated in a higher level decision taking step where a compensation is provided for possible errors, so that a low level of false alarms and missed detections are observed at the final decision.

The first approach we pursue to combine individual features is to compute the likelihood of the existence of a DoS attack by averaging the likelihood of attack for each feature:

$$l_{final} = \frac{l_{bit} + l_{acc} + l_{entr} + l_{Hurst} + l_{delay} + l_{delrate}}{\textit{total number of features}}$$

l_{final} has a value between 0 and 1. The decision on whether the incoming traffic is normal or DoS is then taken by comparing this value to a specified threshold, which may or may not be dependent on the impact that the DoS attack is expected to have on the victim.

2) RNN with likelihood values: In the second variation of the detection mechanism, the computed likelihoods are used as input in a Random Neural Network (RNN) [2]. In the specific work we have used a feedforward RNN structure with six inputs, twelve neurons in one hidden layer and two outputs (Fig. 1). The inputs receive the values of the likelihood ratios for the six input features and the output nodes correspond to normal and attack traffic. For the fusion of the data we utilised the RNN implementation described in [30].

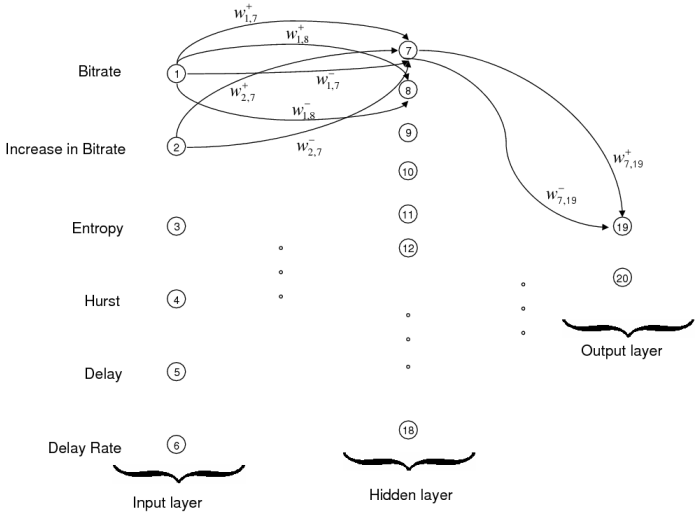

Fig. 1. Random Neural Network in feedforward architecture used in the experiments

In the RNN, neurons exchange positive and negative impulse signals, which represent excitation and inhibition respectively. Neurons accumulate signals as they arrive and positive signals are cancelled by negative signals. Neurons may fire if their potential is positive, to send signals either to other neurons or outside the network. In a RNN, a signal may leave neuron i for neuron j as a positive signal with probability $p^+(i,j)$, as a negative signal with probability $p^-(i,j)$, or may depart from the network with probability $d(i)$, where $p(i,j) = p^+(i,j) + p^-(i,j)$ and $\sum_j p(i,j) + d(i) = 1$. Positive and negative weights are computed with:

$$w^+(j,i) = r(i)p^+(i,j) \geq 0$$
$$w^-(j,i) = r(i)p^-(i,j) \geq 0$$

where $r(i)$ is the firing rate. The potential for the neuron i is $q_i = \frac{N(i)}{D(i)}$, where

$$N(i) = \sum_j q_j w^+(j,i) + \Lambda(i)$$
$$D(i) = r(i) + \sum_j q_j w^-(j,i) + \lambda(i)$$

with $\Lambda(i)$ and $\lambda(i)$ denoting the external inputs into neuron i. The firing rate $r(i)$ is then computed as the sum: $r(i) = \sum_j w^+(i,j) + w^-(i,j)$.

3) RNN with histogram categories: To observe the performance of the RNN when actual values of features were presented, we carried out another implementation of RNN

where we used the histogram values of each of the features as inputs. The advantage of using histogram values instead of actual values is to achieve better learning performance for the RNN since the range of values that it has to learn is quantised.

4) RNN with actual values: For the sake of comparison we have also implemented the detection mechanism consisting only of the RNN module and using as input the raw values for the six input features that we measured during the experiments.

III. Experimental Evaluation

One of the most important issues in DoS research is the lack of common datasets and network topologies on which researchers can evaluate and compare their methods [1]. This is not the result of a lack of consensus, but a known aspect of the nature of DoS attacks. Realistic datasets can be acquired only from real traffic data of networks under real attacks, but then determining the point in time that the attacks started and stopped is in itself an important problem of DoS detection. As a result, for this work we have used traffic traces of DoS attacks designed in our laboratory or by other academic sources, as explained later in this section. In terms of the network topology used in our experiments, instead of applying an ideal theoretical one, we chose to recreate a representative academic network, the SwitchLAN backbone topology [2], which consists of 46 nodes connected with 100 MBits/sec links, as seen in Fig. 2.

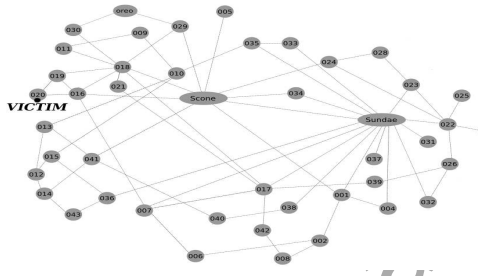

Fig. 2. The network topology used in the experiments

We used four sets of input traffic data and a real 46-nodes networking testbed to evaluate the detection mechanisms that we described in Section II. All experiments lasted 120 sec and, for the sake of comparison, in all cases the victim was one specific node in the topology. Some representative results of the detection results as time progresses can be seen in Figures 3-6. Fig. 3 shows the detection decisions when we have introduced only normal traffic in the network. The figures for the RNN methods are in logarithmic scale since the detection result of these methods is the ratio of the two output nodes. If the ratio is over 1 then the detector decides that the node it monitors is under attack. The closer the ratio is to 1 the less certain the mechanism is of the detection decision.

In the first case, we test the four mechanisms for normal traffic. While the average likelihood, the RNN with likelihoods and the RNN with histogram categories methods correctly yield low values, which indicate the absence of attack traffic, the RNN with actual values method signals an attack for the whole duration of the experiments (Fig. 3). The best of the RNN methods seems to be the one using the histogram categories, which did not signal any false alarm for the duration of the experiments, in contrast with the RNN with likelihoods which did signal a few false alarms.

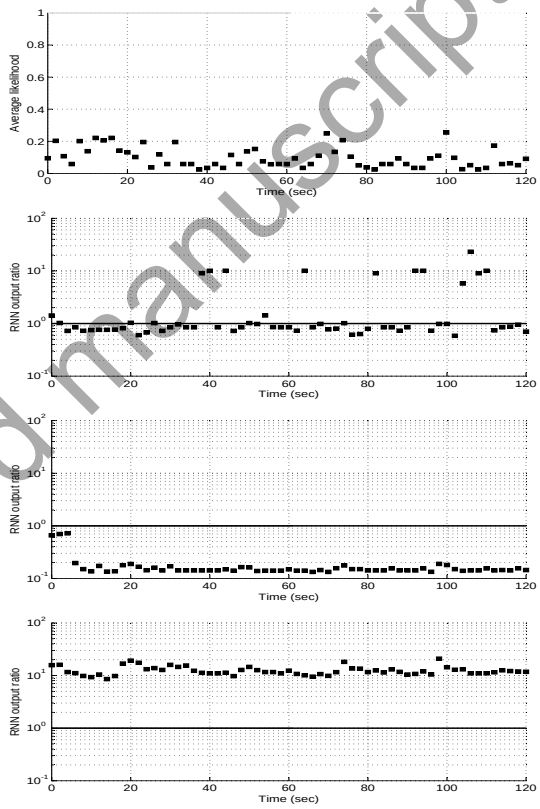

Fig. 3. Detection results for normal traffic (from the top: average likelihood, RNN with likelihoods, RNN with histogram categories, RNN with actual values)

In the second case, in addition to the normal traffic used above, for the time period between 50 and 100 sec, we introduce attack traffic that we have created in our lab. As seen in Fig. 4, all four methods detect a clear difference for the duration of the attack, with the RNN with histogram categories method being the most certain among them of the existence of an attack.

In the third and fourth cases we present in this paper, instead of using our own attack traffic, we test the detection mechanisms with attack traffic extracted from traces downloaded from an online repository of traces [26]. We attempt to recreate the exact attack scenarios by allocating the traffic sent by each source node of the traces to a node in our topology. Again we introduce the attack traffic on top of the

[1]There is the exception of the DARPA-98 to 2000 datasets for DoS detection [31], which, however, are severely outdated; the types of attack that they represent were significant in 2000, but are now easily detected with simple rule-based mechanisms

[2]The SwitchLAN network provides service in Switzerland to all universities, two federal institutes of technology and the major research institutes.

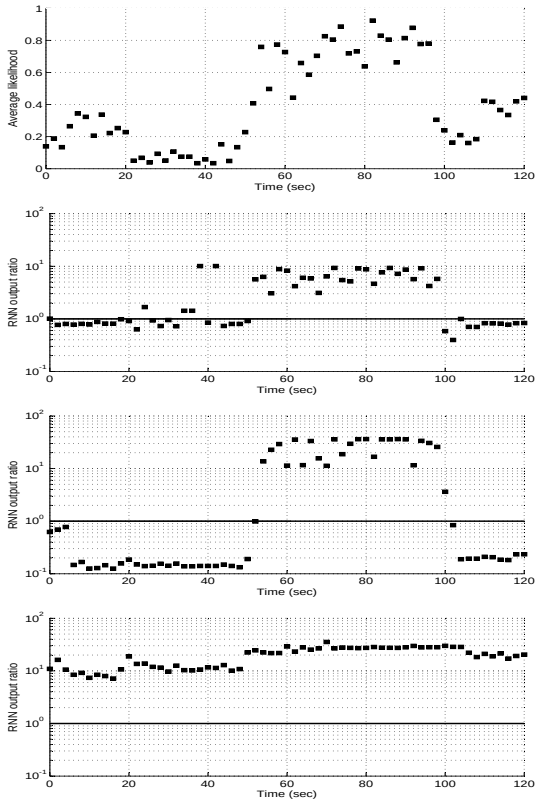

Fig. 4. Detection results for attack traffic (from the top: average likelihood, RNN with likelihoods, RNN with histogram categories, RNN with actual values)

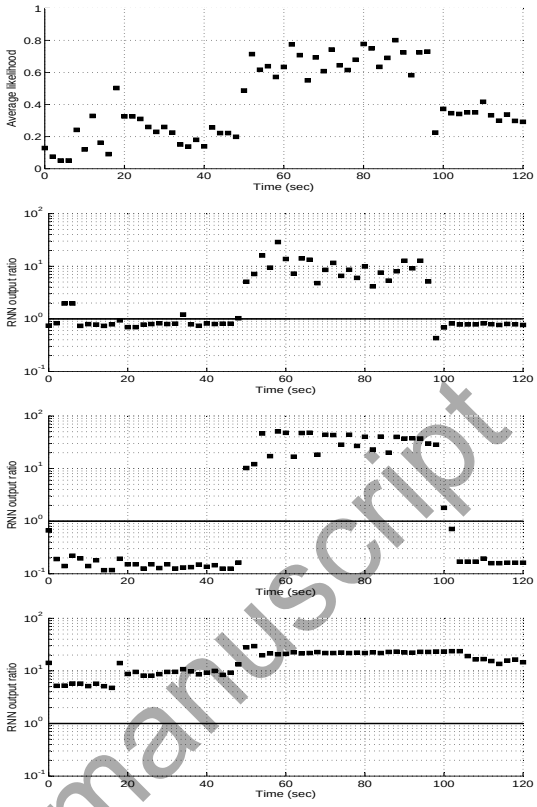

Fig. 5. Detection results for trace1 attack traffic (from the top: average likelihood, RNN with likelihoods, RNN with histogram categories, RNN with actual values)

existing normal traffic for the time period between 50 and 100 sec. The results shown in Figures 5 and 6 are similar to those of the previous case. The average likelihood method yields values around 0.8 while the attack lasts and less than 0.3 before it starts and after it ends. The RNN methods all detect the attack while it lasts, with the RNN with histogram categories method being the most confident of the detection decision and the RNN with actual values being the least successful.

As observed in Table I, and should be expected based on our discussion so far, employing RNN with histogram categories yields the lowest number of false alarms, while both histogram categories and likelihood values give fairly low values of missed detection. As for the implementation of the RNN with actual values, the lack of missed detections cannot be considered as a success, since observing also its false alarm ratio, it is obvious that it is not able to discriminate between normal and attack traffic.

IV. Conclusions

In this paper we have presented the design of a generic DoS detection scheme which uses multiple Bayesian classifiers and the biologically inspired Random Neural Network. After selecting the input features to use for the detection, we obtained estimates of probability density functions as histograms for each feature and we computed likelihood ratios. These ratios can be interpreted as first-level decisions for each of the features. In high level decision taking step, we aggregated first level decisions, by combining them with averaging or with RNNs. We also implemented RNNs with actual values and histogram categories of the features. We evaluated our approach for a variety of input traffic data in a large networking testbed and compared the four implementations in terms of correct detections, missed detections and false alarms. The strong point of our approach is that it combines the powerful discriminating capacity and approximation properties of the RNN with both instantaneous values and statistical data of the incoming traffic. In our experiments, we attempted to fuse several of the most commonly used input features together with some of our design, but we have also described how additional or in some cases more suitable input features can be used. The next step in our research is to integrate this set of detection mechanisms with the DoS classification and DoS response methods that we covered in our previous papers [8], [15], [20].

References

[1] C.E. Shannon and W. Weaver: "The Mathematical Theory of Communication", University of Illinois Press, 1963.

[2] E. Gelenbe: Learning in the Recurrent Random Neural Network, *Neural Computation*, Vol. 5, pp. 154-164, 1993.

[3] E. Gelenbe, V. Koubi, and F. Pekergin: Dynamical random neural network approach to the traveling salesman problem, *in: Systems, Man and Cybernetics, 1993. 'Systems Engineering in the Service of Humans', IEEE Conference Proceedings*, Vol. 2, pp. 630-635, ISBN: 0-7803-0911-1, Oct 17-20 1993.

	False Alarms			Missed Detections		
Detection Method	Dataset1	Dataset2	Dataset3	Dataset1	Dataset2	Dataset3
RNN with likelihood values	0.167	0.111	0.083	0.04	0.04	0.16
RNN with histogram categories	0.028	0.111	0.028	0.08	0	0.2
RNN with actual values	1	1	1	0	0	0

TABLE I

FALSE ALARM AND MISSED DETECTION RATES FOR THE THREE RNN METHODS AND THREE DATASETS

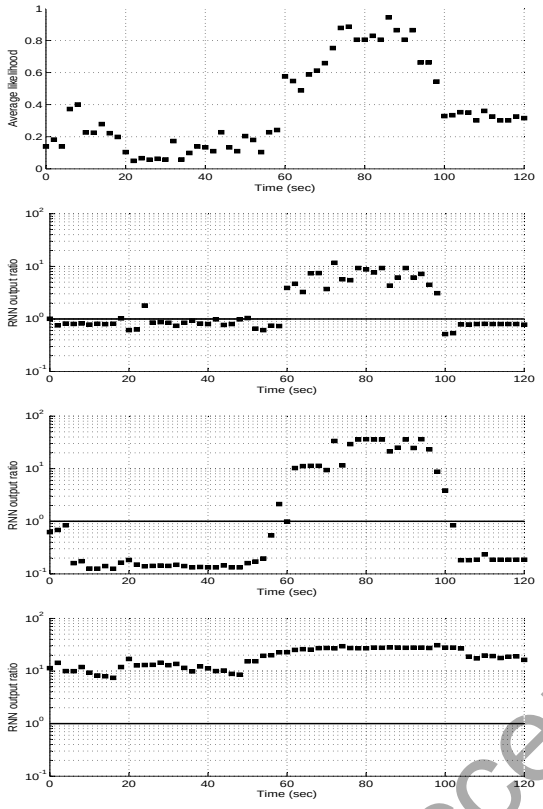

Fig. 6. Detection results for trace2 attack traffic (from the top: average likelihood, RNN with likelihoods, RNN with histogram categories, RNN with actual values)

[4] E. Gelenbe, Y. Feng, and K. Krishnan: Neural network methods for volumetric magnetic resonance imaging of the human brain, *Proceedings of the IEEE 84*, pp. 1488-1496, 1996.

[5] E. Gelenbe, T. Kocak, and L. Collins: Sensor Fusion for Mine Detection with the RNN, *Proceedings of the 7th International Conference on Artificial Neural Networks*, Vol. 1327, pp. 937-942, 1997.

[6] L. Feinstein, D. Schnackenberg, R. Balupari, and D. Kindred: "Statistical Approaches to DDoS Attack Detection and Response", *Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX'03)*, 2003.

[7] S. Noh, C. Lee, K. Choi, and G. Jung: "Detecting Distributed Denial of Service (DDoS) Attacks through Inductive Learning", *Lecture Notes in Computer Science*, Vol. 2690, pp. 286-295, 2003.

[8] E. Gelenbe, M. Gellman, and G. Loukas: "Defending networks against denial of service attacks", In E.M. Carapezza, editor, *Proceedings of the Conference on Optics/Photonics in Security and Defence (SPIE)*, Unmanned/Unattended Sensors and Sensor Networks, Vol. 5611, pp. 233-243, London, UK, October 2004.

[9] M. Kim, H. Na, K. Chae, H. Bang, and J. Na: "A Combined Data Mining Approach for DDoS Attack Detection", *Lecture Notes in Computer Science*, Vol. 3090, pp. 943-950, 2004.

[10] S. Mukkamala and A.H. Sung: "Computational Intelligent Techniques for Detecting Denial of Service Attacks", *Lecture Notes in Artificial Intelligence*, Vol. 3029, pp. 616-624, 2004.

[11] X. Yang, Y. Liu, M. Zeng, and Y. Shi: "A Novel DDoS Attack Detecting Algorithm Based on the Continuous Wavelet Transform", *Lecture Notes in Computer Science*, Vol. 3309, pp. 173-181, 2004.

[12] M. Li: "An Approach to Reliably Identifying Signs of DDOS Flood Attacks Based on LRD Traffic Pattern Recognition", *Computers and Security*, Vol. 23, No. 549-558, 2004.

[13] Y. Xiang, Y. Lin, W.L. Lei, and S.J. Huang: "Detecting DDOS attack based on Network Self-Similarity", *IEE Proceedings in Communication*, Vol. 151, No.3, pp. 292-295, 2004.

[14] D.O. Cajueiro and B.M. Tabak: "The Hurst Exponent over Time:Testing the Assertion That Emerging Markets Are Becoming More Efficient", *Physica A*, Vol. 336, pp. 521-537, 2004.

[15] E. Gelenbe, M. Gellman, and G. Loukas: "An autonomic approach to denial of service defence", *In Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pp. 537-541, June 2005.

[16] D. Gavrilis and E. Dermatas: "Real-time detection of distributed denial-of-service attacks using RBF networks and statistical features",*Computer Networks*, Vol. 48, pp. 235-245, 2005.

[17] H.T. He, X.N. Luo, and B.L. Liu: "Detecting Anomalous Network Traffic with Combined Fuzzy-Based Approaches", *Lecture Notes in Computer Science*, Vol. 3645, pp. 433-442, 2005.

[18] S.Y. Lee, Y.S. Kim, B.H. Lee, S.H. Kang, and C.H. Youn: "A Probe Detection Model Using the Analysis of the Fuzzy Cognitive Maps", *Lecture Notes in Computer Science*, Vol. 3480, pp. 320-328, 2005.

[19] L. Li and G. Lee: "DDoS Attack Detection and Wavelets", *Telecommunication Systems*, Vol. 28:3, No. 4, pp. 435-451, 2005.

[20] E. Gelenbe and G. Loukas: "Self-Aware Approach to Denial of Service Defence", *Accepted for publication in special edition of Elsevier Journal on Computer Networks: Security through Self-Protecting and Self-Healing Systems*, 2006.

[21] W. Wei, Y. Dong, D. Lu, and G. Jin: "Combining Cross-Correlation and Fuzzy Classification to Detect Distributed Denial-of-Service Attacks", *Lecture Notes in Computer Science*, Vol. 3994, pp. 57-64, 2006.

[22] M. Li: "Change Trend of Averaged Hurst Parameter of Traffic under DDOS Flood Attacks", *Computers and Security*, Vol. 25, pp. 213-220, 2006.

[23] Y. Chen and K. Hwang: "Collaborative detection and filtering of shrew DDoS attacks using spectral analysis", *Parallel Distrib. Comput.*, Vol. 66, pp. 1137-1151, 2006.

[24] S. Haykin: *Neural Networks A Comprehensive Foundation*,pp. 143-146, Prentice-Hall Inc., 1999.

[25] R.O. Duda, P.E. Hart, and D.G. Stork: *Pattern Classification*, pp. 20-214, John-Wiley and Sons, 2001.

[26] UCLA CSD packet traces: http://www.lasr.cs.ucla.edu/ddos/traces/public/usc/.

[27] SecurityFocus, August 2004: FBI busts alleged DDoS Mafia, http://www.securityfocus.com/news/9411.

[28] BBC, September 2001: Teenager cleared of hacking, http://news.bbc.co.uk/1/hi/england/hampshire/dorset/3197446.stm.

[29] WiredNews: Pentagon Hacker Exposed by Just. Dpt, http://www.wired.com/news/technology/0,1282,11030,00.html.

[30] H. Abdelbaki: Matlab simulator for the RNN, http://www/cs/ucf.edu/ ahossam/rnnsim.

[31] MIT Lincoln Laboratory: DARPA intrusion detection evaluation data sets. http://www.ll.mit.edu/IST/ideval/data/data_index.html