Likelihood Ratios and Recurrent Random Neural Networks in Detection of Denial of Service Attacks

Georgios Loukas and Gülay Öke
Imperial College London
{**georgios.loukas,g.oke**}**@imperial.ac.uk**

Keywords: Denial of Service, Recurrent Random Neural Networks, Network Security, Intrusion Detection, Bayesian Decision Taking

Abstract

In a world that is becoming increasingly dependent on Internet communication, Denial of Service (DoS) attacks have evolved into a major security threat which is easy to launch but difficult to defend against. In order for DoS countermeasures to be effective, the attack must be detected early and accurately. In this paper we propose a DoS detection technique based on observation of the incoming traffic and a combination of traditional likelihood estimation with a recurrent random neural network (r-RNN) structure. We select input features that describe essential information on the incoming traffic and evaluate the likelihood ratios for each input, to fuse them with a r-RNN. We evaluate the performance of our method in terms of false alarm and correct detection rates with experiments on a large networking testbed, for a variety of input traffic.

1. INTRODUCTION

Denial of Service (DoS) attacks have existed since the early 1990s, but they were not considered a major threat as they were not involved in high-profile incidents. Things have changed the last seven years mainly due to the significant financial damages inflicted on several organisations with online presence. DoS attacks are disproportionately easy to launch, since the necessary tools are readily available on the web, while defending against them takes more than installing some software. A typical DoS attack is distributed; an attacker acquires control of a number of relatively vulnerable computers, such as those without firewall and up to date antivirus software, and orders them to simultaneously attack a specific target by sending vast volumes of meaningless traffic. Over the years there have been a number of smart methods proposed to defend against DoS attacks, but they become obsolete to an extent after a while, as the attack types evolve to counter the latest defence trends.

In the most general sense, one can identify three stages that should comprise a complete DoS defence system:

- **Detection**. Usually a system running on the victim machine identifies in real-time the existence of an attack and triggers the initiation of the next two stages. The detection phase consists of looking either for anomalies in the incoming traffic or for signature characteristics of known attack types.
- **Classification**. Then, the victim or the rest of the nodes of the same network monitor the incoming traffic and attempt to distinguish between normal traffic (sent by legitimate users) and attack traffic (sent by nodes controlled by the attacker).
- **Response**. Common approaches for the response phase include dropping the traffic that was identified as attack traffic during the classification phase, and redirecting it to a trap where it can be analysed.

This three-phase paradigm does not mean that detection, classification and response have to be separate entities or that they have to be performed in strict sequence. For example, detection and classification may be achieved by observing the same traffic characteristics.

In this paper, we concentrate on the first of the three phases. In the literature, DoS detection is often considered as a pattern recognition problem where the aim is to observe and analyse the incoming traffic, and for this reason various machine learning techniques have been utilised. For example, the authors of [1] have designed a Statistical Pre-Processor and Unsupervised Neural Net based Intrusion Detector (SPUNNID), in which the statistical pre-processor is used to extract features from packets, and the feature vector is changed to numerical form and fed to an unsupervised Adaptive Resonance Theory net (ART). Neural networks for DoS detection are also used in [2], which follows a data mining approach. In [3], a scheme is presented which comprises a collector of the appropriate data fields from the incoming packets, a feature estimator that evaluates the frequencies for the encoded data, and a radial basis function neural network detector to characterise the incoming traffic as normal or DoS. In [4], an Adaptive Neuro-Fuzzy Inference System is used together with a Fuzzy C-Means Clustering Algorithm to detect DoS attacks, and in [5] another three computational intelligence techniques for DoS detection are compared, namely support vector machines, multivariate adaptive regression splines and linear genetic programs. In general, machine learning techniques have

been preferred by several researchers for their increased accuracy However, their success depends largely on how relevant their input features are to the existence or not of an attack.

An important section of DoS detection research is directed towards observing and analysing some statistical properties and the energy content of normal and attack traffic. Normal Internet traffic is known to be long-range dependent (LRD) and self-similar, but in the case of a DoS attack there are usually important deviations for these properties [6]. For example, in [7] the incoming traffic is characterised as normal or DoS based solely on its autocorrelation function. In [8], the self-similarity property of Internet traffic is used to identify DoS attacks. The authors use the packet number or packet size as the input feature and evaluate the Hurst parameter H by statistical techniques. In their approach, the variance of H in consecutive time intervals is calculated and if there is a doubling of the variance, it is decided that a DoS attack is in progress. In [9] the entropy is computed as a measure of randomness, and the chi-square statistic as a measure of statistical significance and estimate of confidence, to detect the existence of an attack. Also, since the energy distribution of normal traffic is known to be relatively stationary, while an attack usually results in changes in the energy distribution variance, in [10] wavelets are used for computing the variations in the energy distribution in the incoming traffic. In another study that exploits energy content [11] flat energy bursts in the traffic are determined with the continuous wavelet transform.

Here, we attempt to bridge these two general directions of DoS detection by building a system which uses several statistical features deemed in the literature as most significant for a DoS attack, and combines the individual decisions in a machine learning fashion. We use Bayesian classifiers to assess the likelihood of the existence of an attack and a Recurrent Random Neural Network (r-RNN) to fuse all information into an overall detection decision. Bayesian classifiers have been used before for DoS detection in [12], but applied only on the rate of appearance of specific flags in the packets' headers, and by Chen et al. [13], 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 combines them with a recursive structure of the RNN. The latter is a neural network model introduced by Gelenbe in [14], which is based on the spiking behaviour of the biological neuron instead of the classical approaches which assume analog transmission of signals. In this work we exploit the capability of the RNN to model the excitatory and inhibitory interactions among its inputs for the case of malicious incoming traffic in a network. We evaluate our detection technique for different traffic data in a large networking testbed.

The rest of this paper is structured as follows. In section 2 we present a description of our detection mechanism detailing input feature selection, statistical information gathering and real-time decision taking. We continue in section 3 with a description of the experiments we conducted and summarise our results. In section 4 we conclude and suggest future research directions of our work and DoS research in general.

2. DETECTION MECHANISM

DoS detection mechanisms in the literature have often suffered from a lack of plurality and diversity in the selection of input features, which in turn results in overspecialisation. In our work we address this weakness by selecting a variety of different input features and following a two-stage decision taking process. In the first stage, the likelihood of the existence of an attack is computed according to each input feature. These likelihood values are then fused using a recurrent random neural network to reach an overall detection decision.

2.1. Input Feature Selection

In our detection mechanism we use both instantaneous and statistical characteristics of the incoming traffic, which exhibit distinctly different behaviour in case of normal and DoS traffic. Since the goal of the attacker is to deny or degrade the service for legitimate users by overwhelming either the processing or the networking resources of a victim network, a DoS detection mechanism should not further aggravate the situation. Thus, the input features must be easily measurable to conform with the requirement of timely detection without heavy processing load.

- ***Bitrate***. During the majority of DoS attacks, the bitrate of the incoming traffic exhibits a ramp-up behaviour, a sharp increase at the beginning of the attack which reaches a peak when all attack flows have arrived. However, while an increasing bitrate could be a convincing confirmation of a DoS attack, it cannot be an absolute proof, since such ramp up behavior can also be observed in flash crowds, which are sharp increases in legitimate connections due to some significant event.

- ***Increase in Bitrate***. Increase in bitrate is often accompanied by increase in its rate of change. This can be observed in flooding attacks which start with a long period of increasing bitrate, and in pulsing attacks, which exhibit consecutive periods of increasing and decreasing bitrate. Determining the rate of change of bitrate provides additional information for DoS detection.

- ***Entropy***. Entropy is an indication of the degree of uncertainty associated with the underlying probability description of the data. For example, a probability distribution expanding over a wide range of values would yield

high entropy. It has been established in the technical literature that the entropy content in normal Internet traffic and traffic under DoS attack differ significantly and thus it can be used as a discriminator in attack detection. For example, in [9] the entropy of the amount of source IP addresses to detect attacks is computed as a detection criterion. In our detection mechanism, we calculate the entropy associated with the incoming traffic as given by [15]:

$$E = -\sum_i f_i log f_i \quad (1)$$

where f_i are the histogram values obtained for the bitrate.

- ***Hurst Parameter***. Normal Internet traffic is known to be self-similar or equivalently long-range dependent, a measure of which is the Hurst parameter. Xiang et al. [8] use the variations of the Hurst parameter of the number and the size of packets to detect attacks in consecutive time intervals. We use the actual value of the Hurst parameter for the incoming bitrate in our study and we compute it using the (R/S) analysis, as described in [16]. Below, x is the bitrate of the incoming traffic, n is the observation time, and N is the total number of observation points, the (R/S) is given by:

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

  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***. A natural consequence of high bitrate and building up of congestion is the increase in the packet delays. Still, to our knowledge it has not been used before 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. We are not aware of existing work using the change of the delay as a detection feature, but we consider it a natural next step.

2.2. Statistical Information Gathering

In order to collect statistical information about the traffic, the node that we are defending monitors the incoming flows and accumulates data. This phase comprises two steps, obtaining the probability density functions (pdf) and computation of the likelihoods. A probabilistic description of the network traffic is derived by estimating the pdfs for both normal and attack traffic in the form of histograms, for all the input features that we described in section 2.1.. The pdfs are indicated by $f_{feature}(x|w_N)$ and $f_{feature}(x|w_A)$, where x is the measured value of the feature from the available traffic data, w_N denotes the normal traffic, w_A the attack traffic and *feature* refers to bitrate, bit acceleration (increase in bitrate), entropy, Hurst parameter, delay and delay rate respectively.

After obtaining the pdfs, the next step of information gathering entails calculation of the likelihoods, $l_{feature}$ for each feature: $\frac{f_{feature}(x|w_A)}{f_{feature}(x|w_N)}$, which will then be used in the Bayesian decision taking mechanism. The Bayesian decision theory aims to minimise the risks encountered by the decision taking process by evaluating the various tradeoffs between decisions [17]. Detection of DoS attacks is a two-category pattern recognition problem, where the discriminator observes traffic values and decides whether the incoming traffic is normal or part of an attack. For a classification problem with two categories (w_1 and w_2), Bayesian classifiers are used by 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 [18]. This step is performed in real-time detection phase, which is discussed in detail next.

2.3. Decision Taking

The victim node monitors the incoming traffic continuously and measures network parameters to determine the current values of the input features. According to the measured values, likelihoods ratios are computed by referring to the likelihood values already stored during the statistical information gathering phase. These likelihood ratios indicate a measure of ongoing attack possibility with respect to each input feature. These individual likelihood ratios, which can also be interpreted as first order decisions, are then combined in an information fusion phase, by r-RNNs. By combining decisions established by different features describing various charateristics of the traffic, we aim to decrease false alarm rate and increase the correct decision rate, since erroneous decisions resulting in learning and generalisation steps can now be compensated.

The RNN is a biologically inspired architecture, applied successfully in various areas, including image processing [19], pattern recognition [20], and optimisation [21]. RNNs

model the signals propagating between neurons as spikes rather than analog signals and therefore provide a more realistic approximation to real biological neurons than artificial neural networks. They also carry a strong analogy with queuing systems, which is why they have also been used as the basis of a networking protocol [22]. In the RNNs positive and negative impulse signals, with unit amplitude, which represent excitation and inhibition respectively are accumulated in neurons. Positive signals are cancelled by negative signals and neurons may fire if their potential is positive.

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 $\sum_j p^+(i,j) + p^-(i,j)) + d(i) = 1$. Positive and negative weights can be computed by:

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

where weights w represent excitatory and inhibitory signal emmission rates and $r(i)$ is a Poisson firing rate, with independent, identical exponentially distributed inter-impulse intervals:

$$r(i) = \sum_j w^+(i,j) + w^-(i,j)$$

The steady state probability that the neuron i is excited can be computed by $q_i = \frac{N(i)}{D(i)}$, 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 rates of exogenous excitory and inhibitory signal inputs into neuron i, respectively.

In order to fuse the individual decisions obtained by evaluating the likelihood ratios associated with each input feature we have designed a r-RNN consisting of two layers, an input layer with twelve nodes and an output layer with two nodes, as depicted in Figure 1. In the input layer, we have employed two nodes for each input variable; one for the excitatory signals and one for the inhibitory signals. Each node sends excitatory signals to same type of nodes and inhibitory signals to opposite types of nodes. At the output layer, excitatory signals are collected at one node and inhibitory signals are summed up at the second node. To obtain the final decision, the ratio of the output nodes is computed, a ratio with a value smaller than 1 denotes normal traffic while if it is greater than 1, an attack is signalled.

In addition to aggregating likelihoods with RNNs, we have also implemented RNNs with actual and quantised histogram values of input features. The advantage of using histogram values compared to actual values is that, in the learning phase the RNN has to learn a smaller set of values , hence the learning performance will be improved. We repeated the experiments with actual data merely for comparison reasons. To implement the RNNs, we have used the software developed in [23].

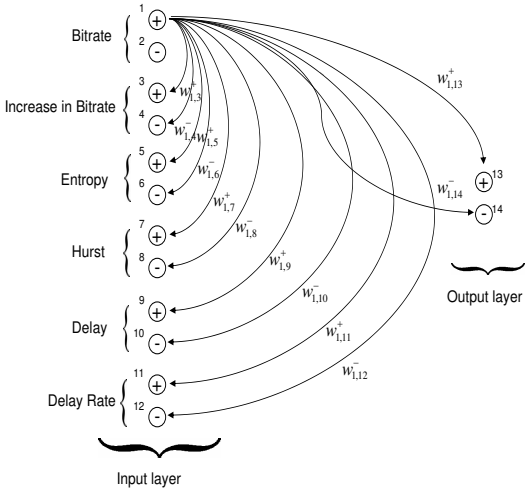

Figure 1. Random Neural Network in the recurrent architecture used in the experiments

3. PERFORMANCE EVALUATION

We have implemented our detection mechanism on a large networking testbed in our laboratory. The testbed consists of 46 nodes connected with 100 MBits/sec links and the topology of Fig. 2. Instead of following a random topology, we chose to recreate a representative academic one, which is the SwitchLAN [1] backbone network topology. We chose a specific node to play the role of the victim while the rest of the nodes send traffic to it according to a variety of datasets that we tried.

We have designed a r-RNN with three types of inputs, the likelihoods of the input features, their quantised histogram values and the actual raw data. We have used four types of datasets. Dataset0 and Dataset1 are designed by ourselves according to our experience with DoS attacks, and represent normal and attack traffic respectively. Dataset2 and Dataset3

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

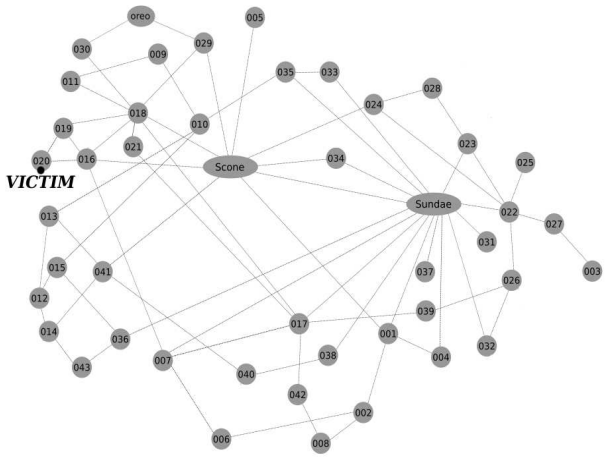

Figure 2. The network topology used in the experiments

have been extracted from data acquired from an online repository of traces [24] and represent flooding and pulsing attacks. For these we recreate the exact attack scenarios by allocating the traffic sent by each source node of the traces to a node in our topology. All our experiments last 120s and we have measured the variables with a sampling rate of 2s. In the attack cases, to illustrate the difference in the traffic and to see graphically the operation of the detection mechanism, we start with normal traffic on which we superimpose attack traffic for the time period between 50s and 100s. The last 20s the network returns to its normal operation, as the attack sources stop sending traffic to the victim.

Table 1 summarises the performance results of the detection mechanism in terms of average correct detection and false alarm rates, and figures 3-6 show the real-time detection decisions as time progresses. The y axis in these graphs is in logarithmic scale with the detection metric being the ratio of the two output RNN nodes as described in section 2.3.. The decision threshold over which an attack is signalled is the RNN output ratio of 1. The closer the detection ratio is to 1 the less certain the mechanism is of its detection decision.

The results of fig. 3 show that for normal traffic the RNNs signal correctly the absence of attack throughout the duration of the experiment, although not with the same degree of certainty. Figures 4-6 correspond to the three attack datasets that we used. All three implementations of the RNN detect the attacks quickly and have minimal missed detections and false alarms. The r-RNN with actual values had very high detection rate and no false alarms for any of the datasets, but also showed the lowest degree of certainty for its decisions by yielding values close to 1. The use of the quantised values of histogram categories appeared to improve the correct detection rate even further, but at the expense of a few false alarms, while the likelihood values performed at about the same level in both respects.

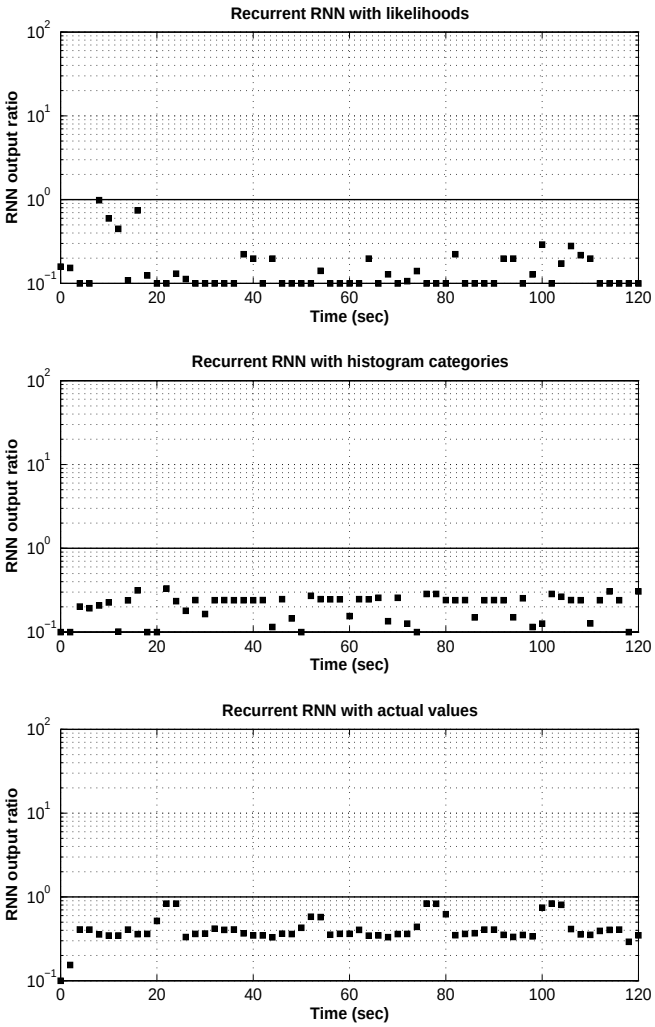

Figure 3. Detection results for Dataset0 (normal traffic)

4. CONCLUDING REMARKS

In this paper we have purposefully chosen not to discuss about the special case of flash-crowd, where there is an unusually sharp increase in the number of legitimate visitors to a website due to some significant event. Although all traffic may be legitimate, the consequences are very similar to those of DoS attacks, such as network outages and dramatically reduced quality of service. In our opinion, flash-crowds could be handled as a third category in our Bayesian classification problem, but there has not been enough work in the literature on the existence of unique statistical properties that characterise these phenomena.

Also, there are two major weaknesses in current DoS research, namely the lack of standards of evaluation for the detection and defence methods and the scarce information on modern types of attacks. Launching real attacks against real networks with real legitimate users, is impractical, and

	False Alarm			Correct Detection		
Detection method	Dataset1	Dataset2	Dataset3	Dataset1	Dataset2	Dataset3
r-RNN with likelihood values	0.06	0.11	0.03	0.96	0.96	0.80
r-RNN with histogram categories	0.06	0.06	0.06	0.96	1	0.88
r-RNN with actual values	0	0	0	0.92	1	0.84

Table 1. Comparison of the three detection implementations, in terms of false alarm and correct detection rates

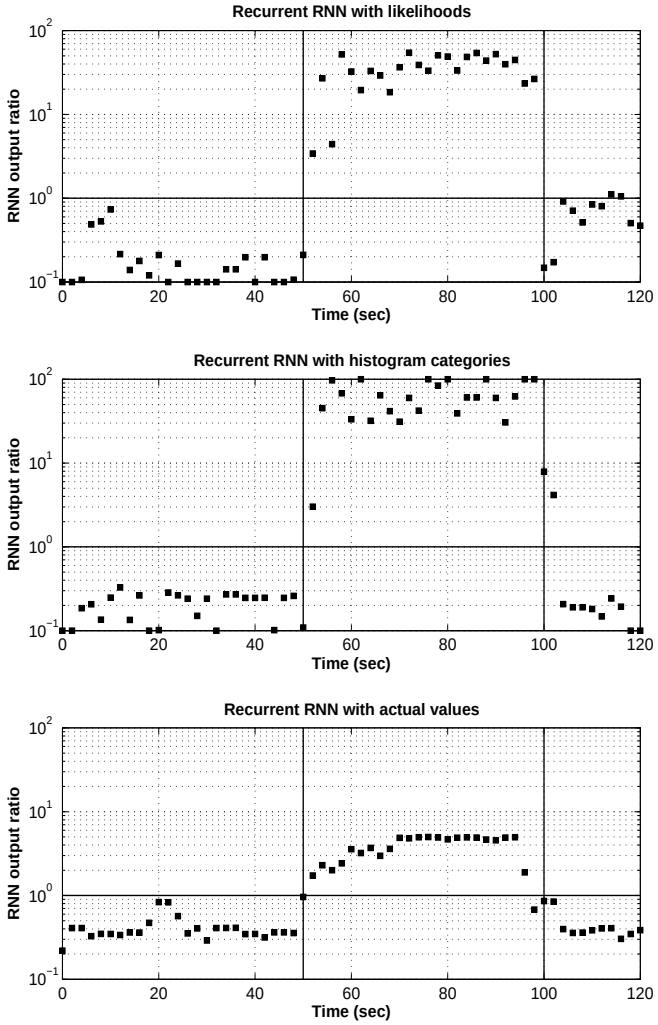

Figure 4. Detection results for Dataset1 (attack traffic)

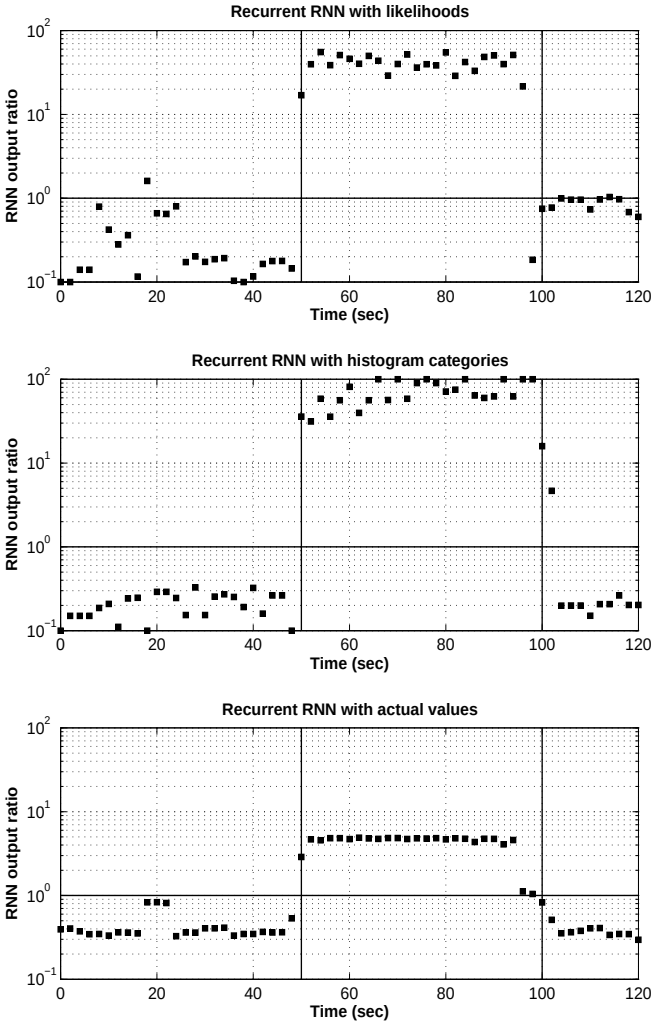

Figure 5. Detection results for Dataset2 (attack traffic)

this leaves the researchers with the option of less dependable datasets, e.g. simulated or acquired from outdated traffic traces. A pragmatic solution to these problems consists in organising a close cooperation of the research community with organisations which are frequently under attack, such as e-commerce and online betting websites. Accurate and up-to-date datasets will help distinguish the best defence approaches in an unbiased manner and will prompt further research and improvements in detection and defence mechanisms. Until these goals are achieved, however, researchers will have to either use the existing obsolete datasets or create their own. In this work we chose to do both, but for the reasons we explained we cannot argue on how these datasets compare against others and how realistic they are. We can, however, argue that our investigation method, using a real large networking testbed, instead of simulation, should provide a significant degree of realism. The experiments we conducted showed that our mechanism is able to detect DoS threats in a timely fashion and can even very quickly identify the end of an attack, at least for the range of attacks that

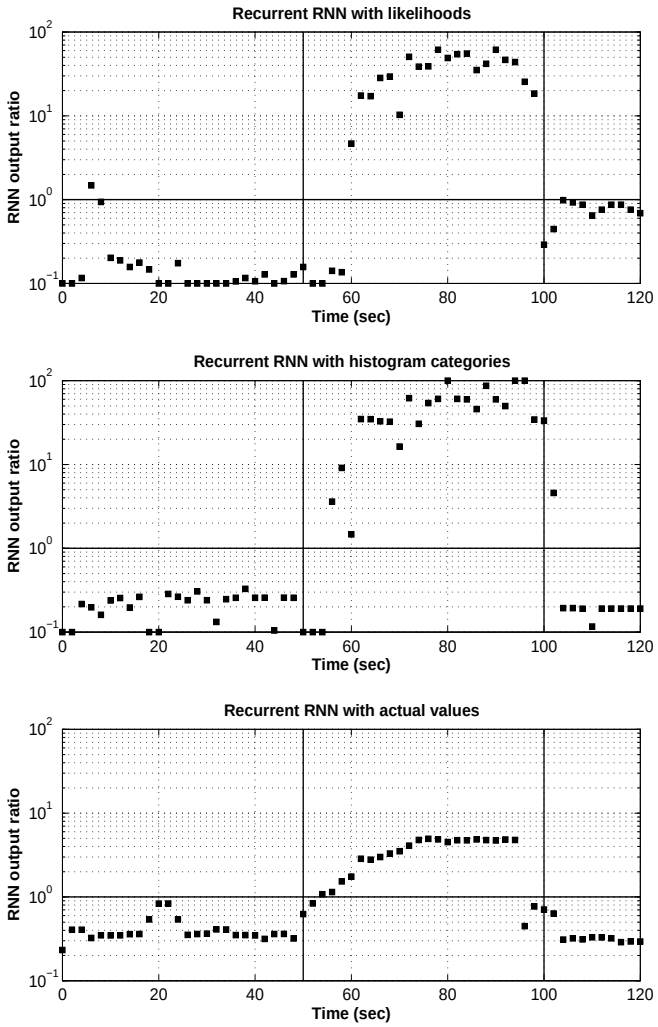

Figure 6. Detection results for Dataset3 (attack traffic)

we investigated.

This paper presents part of our ongoing work to develop a complete DoS defence architecture covering all three aspects mentioned in section 1.. In [25] and [26] we presented our approaches on classification and response, which naturally we intend to combine with the work presented here. Also, a way to improve the detection performance of our current mechanism for each individual node employing it is by introducing cooperation between the various nodes involved in the DoS defence. This we have already achieved in our work for classification and response, and would be particularly useful for accurate detection too.

REFERENCES

[1] R. Jalili, F. Imani-Mehr, M. Amini, and H.-R. Shahriari: "Detection of Distributed Denial of Service Attacks Using Statistical Pre-processor and Unsupervised Neural Networks", *Lecture Notes in Computer Science*, Vol. 3439, pp. 192-203, 2005.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

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

[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] 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.

[13] 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.

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

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

[16] 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.

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

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

[19] C. Cramer, E. Gelenbe, and H. Bakircioglu: Low bit-rate video compression with neural networks and temporal subsampling, *Proc. IEEE*, Vol. 84, No. 6, pp. 1529-1543, Oct. 1996.

[20] A. Teke and V. Atalay: Texture Classification and Retrieval Using the Random Neural Network Model, *Computational Management Science*, Vol. 3(3), pp. 193-205, 2006.

[21] E. Gelenbe and F. Batty: Minimum cost graph covering with the random neural network, *in: O. Balci, R. Sharda, S. Zenios (Eds.), Computer Science and Operations Research*, Pergamon Press, New York, pp. 139-147, 1992.

[22] E. Gelenbe, R. Lent, A. Montuori, and Z. Xu: Cognitive packet networks: QoS and performance, *Proc. MASCOTS 2002, Modeling, Analysis and Simulation of Computer and Telecommunications Systems*, pp. 3-9, ISSN: 1526-7539, ISBN: 0-7695-1840-0, 2002.

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

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

[25] E. Gelenbe, M. Gellman, G. Loukas: "An autonomic approach to denial of service defence", *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Vol. 2, pp. 537–541, Taormina, Italy, 2005.

[26] E. Gelenbe and G. Loukas: "Self-Aware Approach to Denial of Service Defence", *Computer Networks: Special Issue on Security through Self-Protecting and Self-Healing Systems*, Vol. 51(5), pp. 1299–1314, 2007.