Detecting Distributed Denial of Service Attacks: A Neural Network Approach

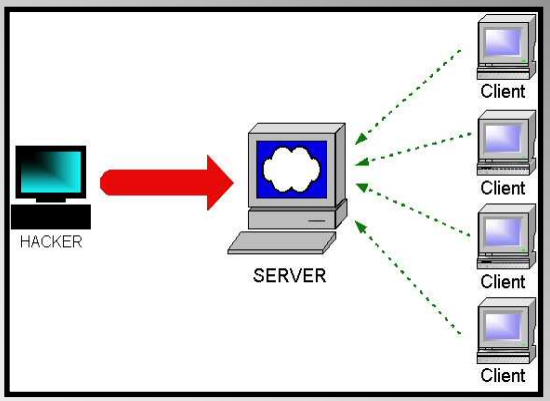
Gulay Oke
(g.oke@imperial.ac.uk)
Intelligent Systems and Networks Group
Dept. of Electrical and Electronic Engineering
Imperial College London

Multi-Service Networks'07
12-13 July 2007

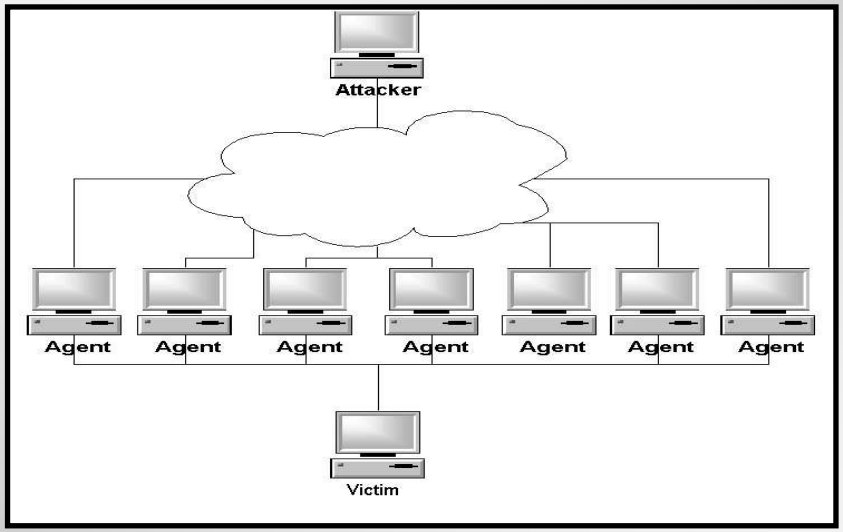
Contents

What is a DoS Attack?

An attack with the purpose of preventing legitimate users from using a specific network resource



Distributed DoS



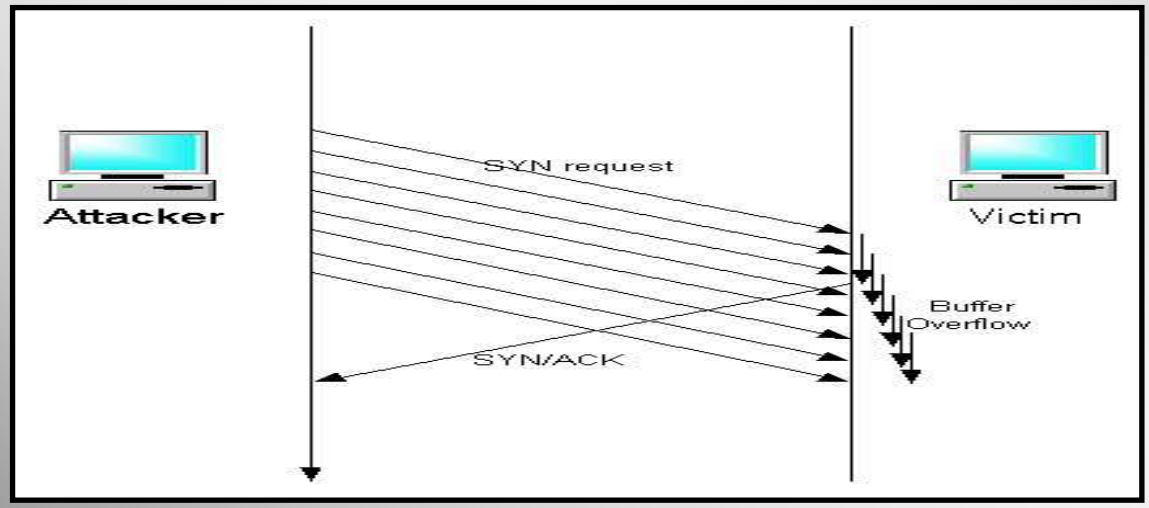
Intelligent Systems & Networks

Imperial College London

1985, R.T. Morris writes:

“The weakness in the Internet Protocol is that the source host **itself** fills in the IP source host id, and there is no provision in TCP/IP to discover the true origin of a packet” .. IP Spoofing

SYN Flood Attack



Why is Detection Necessary?

A combination of detection and response mechanisms are used to defend against such attacks.

Detection would not be necessary in the ideal case of a response architecture with proactive qualities that would render impossible any DoS attack. However:

- No response system is perfect to date.

- Denial of Service attacks against one's network do not happen very often and at least resource-wise a proactive protection system is usually too expensive to operate in the absence of an attack.

Therefore, a detection mechanism can trigger the response procedure to overcome the weaknesses stated above.

Detection of DoS Attacks

1. Methods Based on Identification of the Source Address
2. Methods Based on Analysis of Traffic

A robust DoS detection scheme must satisfy the following:

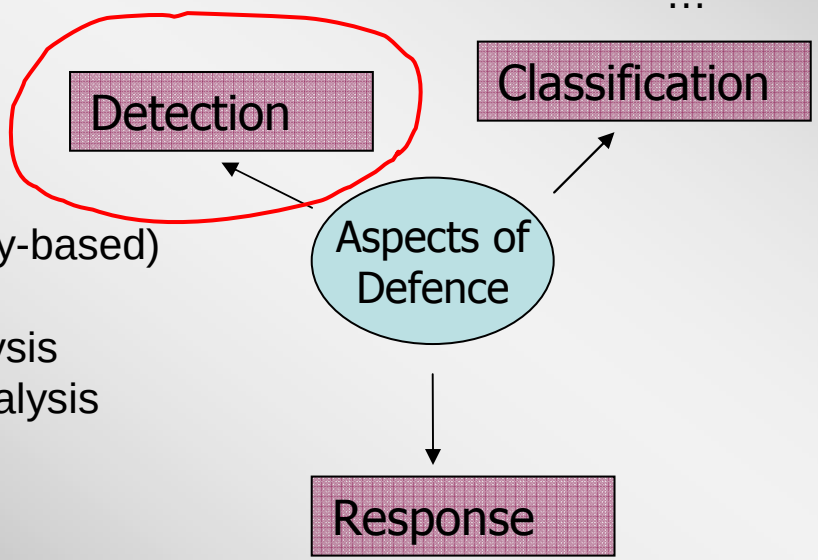
- ✓ High detection rates
- ✓ Minimal false alarm rates
- ✓ Real-time detection with low memory and CPU-time requirements
- ✓ Invariance in evolutionary trends in DoS attacks

Defence techniques

Passive tests
- Loyal clients (beyond suspicion)
- Hop-count filtering (check the TTL)

Active tests
- CAPTCHAs
- Cryptographic puzzles

…



(signatured- & anomaly-based)
Learning techniques
Statistical signal analysis
Wavelet transform analysis
Multiple agents
Fuzzy
…

Proactive server roaming
Pushback
Secure overlay tunneling
Dynamic resource pricing
…

Intelligent Systems & Networks

Imperial College London

Detection Using Bayesian Classifiers

Select the Input Features

- Total incoming bit rate
- Change in total incoming bit rate (acceleration)
- Entropy
- Hurst Parameter
- Delay
- Delay Rate

Gather statistical information on DoS and normal traffic

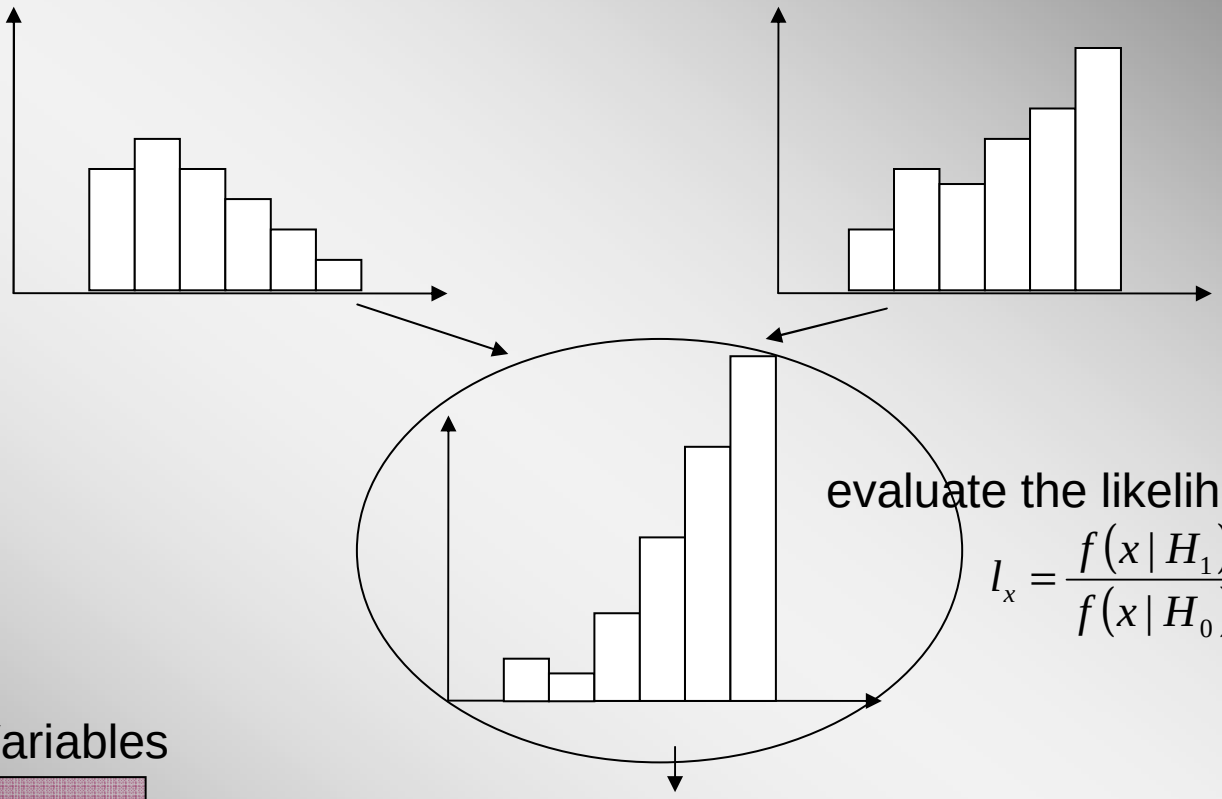
- Obtain histograms
- Evaluate likelihood ratios
- Set thresholds

Real-time decision taking

- Measure the real-time values of the input features from the actual traffic
- 1st level decision (evaluate likelihood ratios)
- 2nd level decision (Average Likelihood or RNN)

Compute the histogram
f(x|H0) of normal traffic

Compute the histogram
f(x|H1) of attack traffic



evaluate the likelihood ratios:

$$l_x = \frac{f(x \mid H_1)}{f(x \mid H_0)}$$

Decision Variables

- Bit rate
- Change in bit rate (acc)
- Entropy
- Self-similarity (Hurst)
- Delay
- Delay Rate

$l_{Bitrate}, l_{Acc},$
$l_{Entropy}, l_{Hurst},$
$l_{Delay}, l_{DelayRate}$

Averaging likelihood
OR
RNN

Intelligent Systems & Networks

Imperial College London

Randomness

Entropy $S = -\sum_{i=1}^{n} f_i \log_2 f_i$

Self-Similarity

The Hurst Parameter represents the degree of self-similarity.

We have used the *R/S* statistic to calculate the Hurst parameter

χ : **incoming bit rate**

$$(R/S)_N = \frac{1}{s_N}\left[\max_{1\le n\le N}\sum_{n=1}^{N}\left(x-\bar{x}\right) - \min_{1\le n\le N}\sum_{n=1}^{N}\left(x-\bar{x}\right)\right]$$

$$s_N = \left[\frac{1}{N}\sum_{n=1}^{N}\left(x-\bar{x}\right)^2\right]^{1/2}$$

$$(R/S)_N = cN^H$$

Intelligent Systems & Networks

Imperial College London

Random Neural Network (RNN)

✓**RNNs represent better approximation of the true functioning of a biophysical neural network, where the signals travel as spikes rather than analog signals**

✓**They are computationally efficient structures.**

✓**They are easy to simulate since each neuron is simply represented by a counter.**

$$\sum_j \left(p^+(i,j) + p^-(i,j)\right) + d(i) = 1$$

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

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

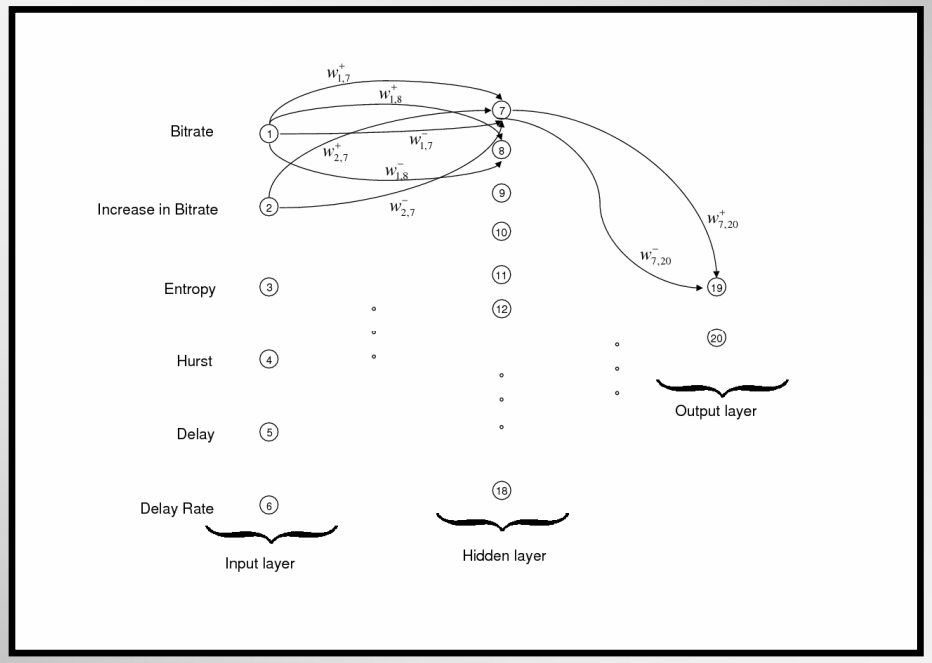
The potential for neuron *i* is:

$$q_i = \frac{N(i)}{D(i)}$$

$$N(i) = \sum_j q_j w^+(j,i) + \Lambda(i)$$

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

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



Feedforward RNN

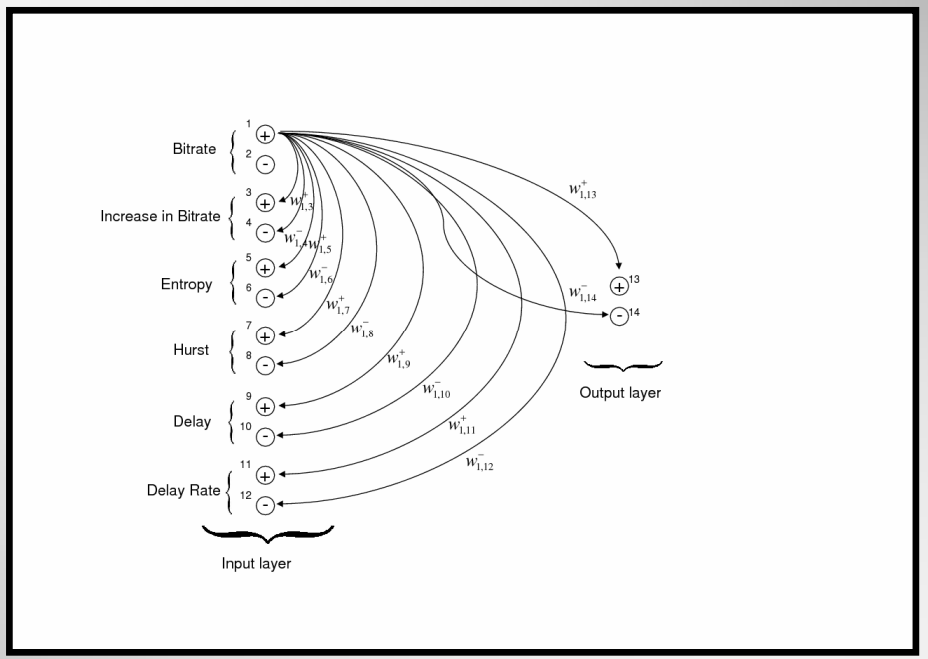
An input layer of six neurons, a hidden layer with twelve neurons and an output layer with two neurons.

Each output neuron stands for a decision; attack or not.

The final decision is determined according to the ratio of the two output neurons.

Intelligent Systems & Networks

Imperial College London



Recurrent RNN

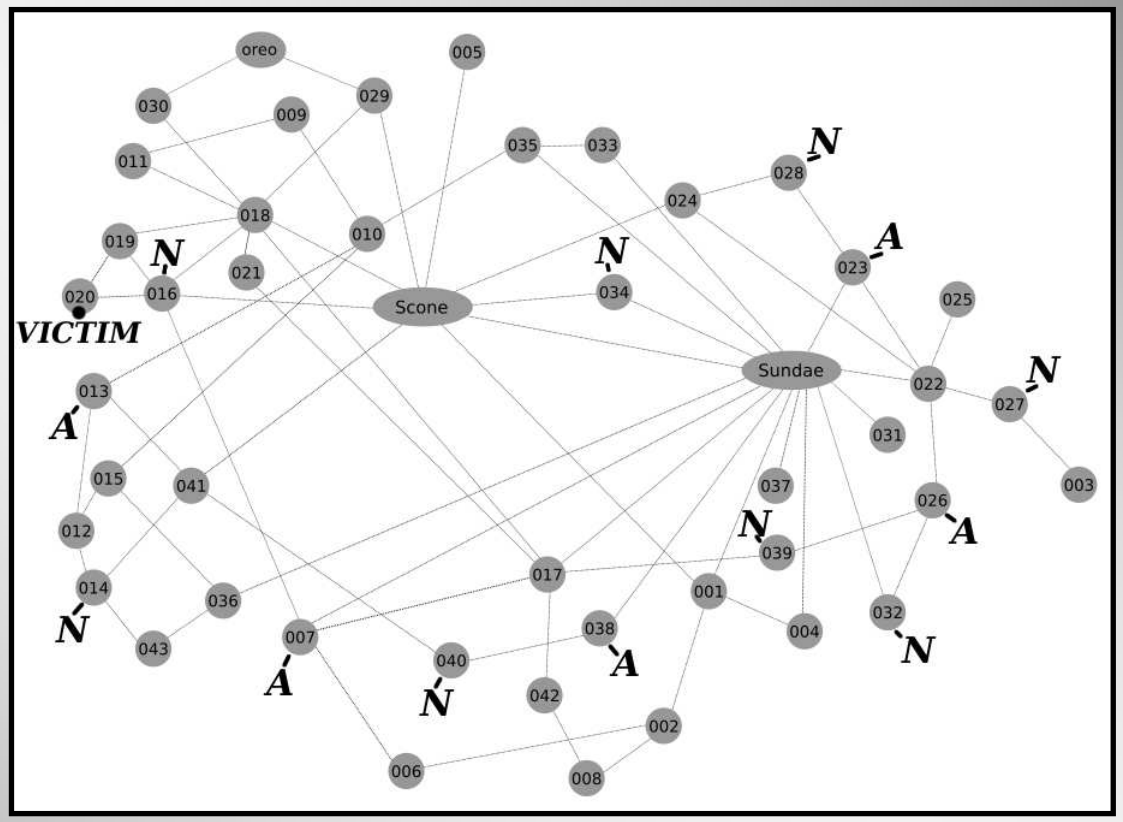
It consists of an input layer with twelve neurons and an output layer with two neurons.

In the input layer, there are two neurons for each input variable; one for the excitatory signals and one for the inhibitory signals.

Each neuron sends excitatory signals to same type of neuron and inhibitory signals to opposite type of neuron.

At the output layer, excitatory signals are collected at one neuron and inhibitory signals are summed up at the second neuron.

Experimental Results



Topology of the test-bed used in the experiments (Node 20 is the victim)

Intelligent Systems & Networks

Imperial College London

We have used four data sets:

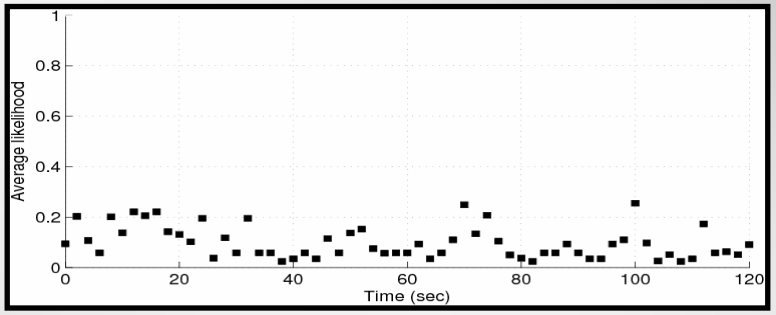
1) Normal traffic we have designed

2) Attack traffic we have designed

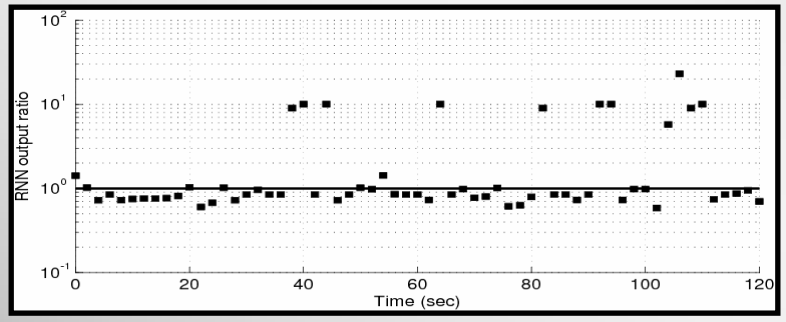
3) Attack traffic extracted from traces downloaded from an online repository of traces (trace1)

4) Attack traffic extracted from traces downloaded from an online repository of traces (trace2)

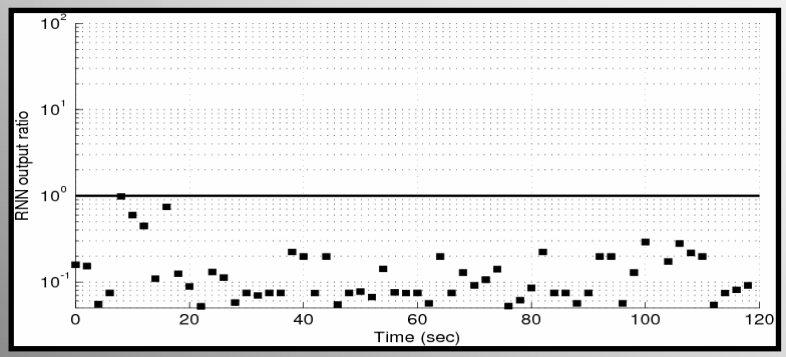
Our Dataset --- Normal Traffic



Averaged Likelihood



Feedforward RNN

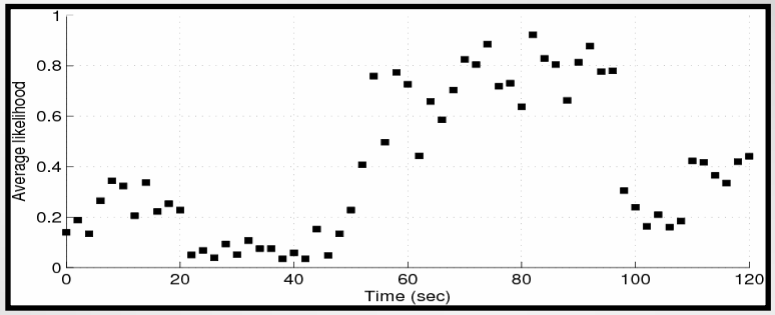


Recurrent RNN

Intelligent Systems & Networks

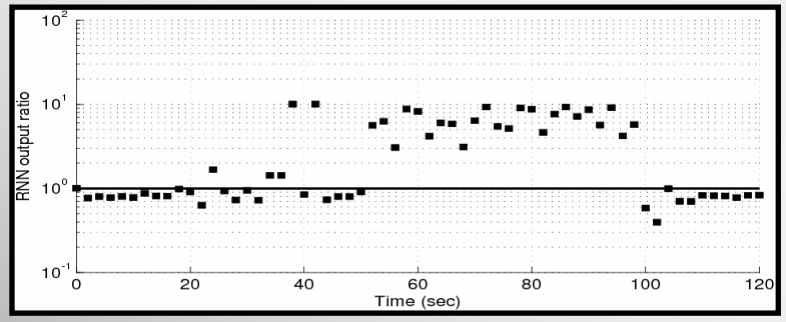
Imperial College London

Our Dataset --- Attack Traffic



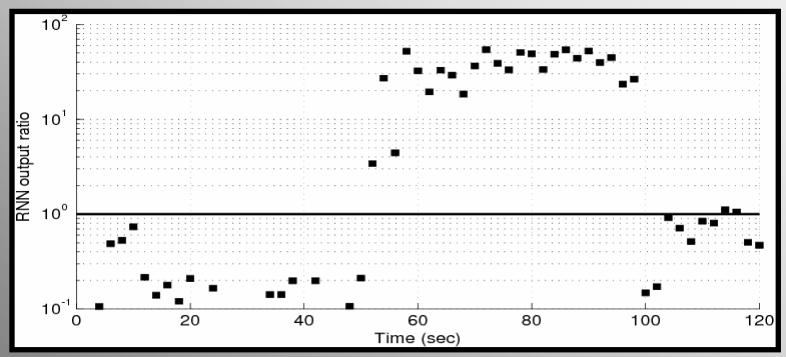
Averaged Likelihood

False Alarms: 0 %
Correct Detections: 80 %



Feedforward RNN

False Alarms: 16.7 %
Correct Detections: 96 %



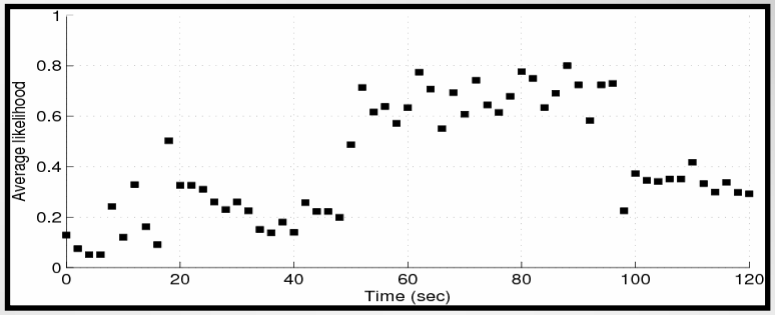
Recurrent RNN

False Alarms: 5.5 %
Correct Detections: 96 %

Intelligent Systems & Networks

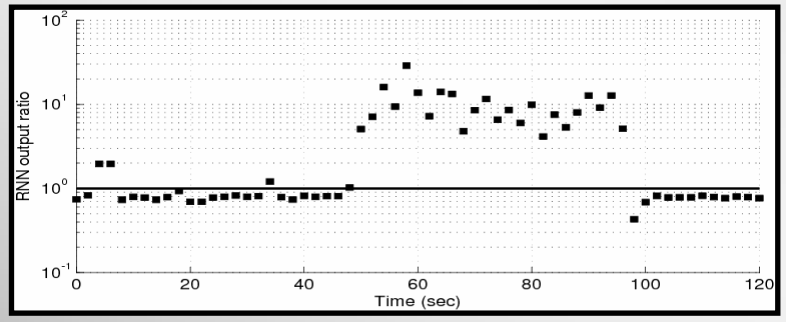
Imperial College London

Trace1 --- Attack Traffic



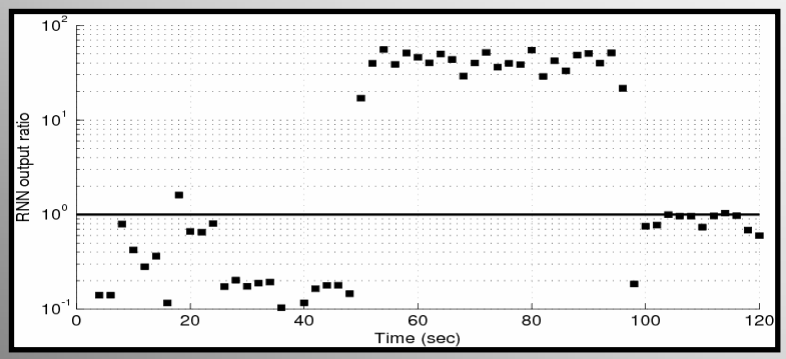
Averaged Likelihood

False Alarms: 2.8 %
Correct Detections: 88 %



Feedforward RNN

False Alarms: 11 %
Correct Detections: 96 %



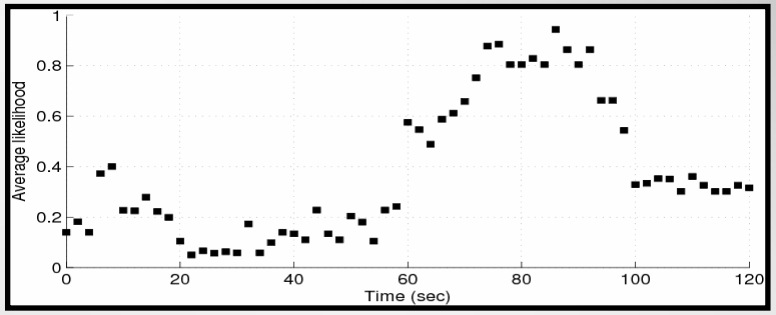
Recurrent RNN

False Alarms: 11 %
Correct Detections: 96 %

Intelligent Systems & Networks

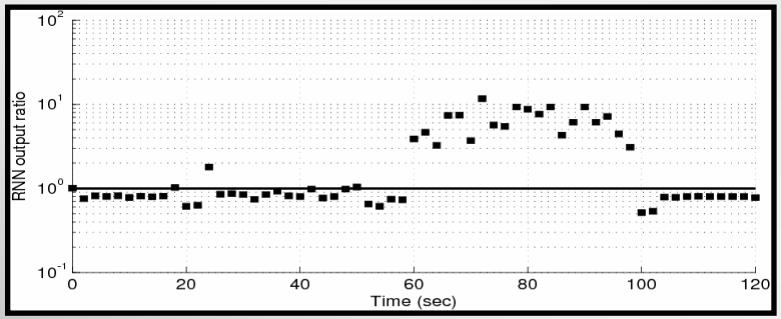
Imperial College London

Trace2 --- Attack Traffic



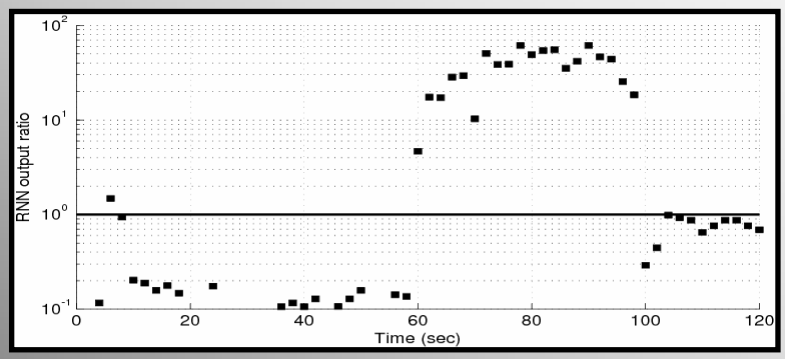
Averaged Likelihood

False Alarms: 0 %
Correct Detections: 76 %



Feedforward RNN

False Alarms: 8.3 %
Correct Detections: 84 %



Recurrent RNN

False Alarms: 2.8 %
Correct Detections: 80 %

Intelligent Systems & Networks

Imperial College London

Future Research

Currently, we are working on the combination of this detection mechanism and previously studied response approaches to build an integrated defence scheme against DoS.

The Bayesian detectors will monitor the traffic and compute a likelihood value for the possibility of an ongoing attack. Based on this value the response mechanism will take action by prioritization and rate limiting.

We are also planning to design a dynamic defence distribution scheme which allocates nodes to rate-limit or drop packets based on predetermined thresholds.

Thanks…

Questions?