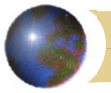
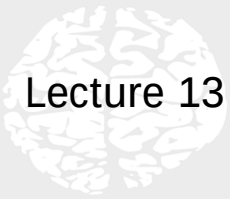


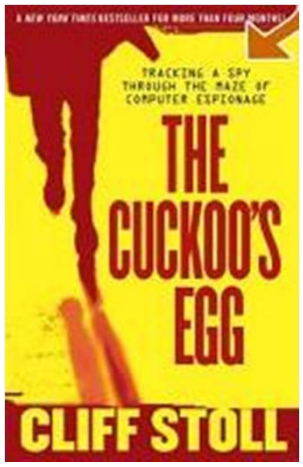
Self-Organizing Maps - Applications



1

Intrusion Detection using Self-Organizing Maps

2



http://projects.autonomy.net.au/ai/chrome/site/resource/ebooks-security/cuckoo_egg.pdf

3

Real-Time Network-Based Intrusion Detection using Self-Organizing Maps

Khaled Labib
Dept. of Applied Science
U.C. , Davis

4

Introduction

- Intrusion Detection Systems (IDS)
  - With the growing rate of interconnections among computer systems, network security is becoming a real challenge.
  - IDS are designed to protect availability, confidentiality and integrity of critical networked information systems.

5

Types of IDS

- Early research into IDS suggested two major detection principles:
  - Anomaly Detection
    - Attempts to quantify the usual or acceptable behavior and flags other irregular behavior as potentially intrusive.
  - Signature Detection
    - Attempts to flag behavior that is close to some previously defined pattern signature of a know intrusion.

6

NSOM

- We created a prototype system, NSOM, to classify network traffic in real-time.
  - Implemented as a combination of C and TCL/TK
  - NSOM Operation overview
    - Continually collect network data from network port
    - Process the data and select features suitable for classification
    - Begin classification process, a chunk of packets at a time
    - Send the resulting classification to a graphical tool

7

NSOM

- Hypothesis
  - Our hypothesis is that routine traffic that represents normal behavior for a given host would be clustered around one or more cluster centers. Any irregular traffic representing abnormal and possibly suspicious behavior would be clustered outside of the normal clustering.
  - Our clustering technique preserves topological mapping between its inputs and outputs.

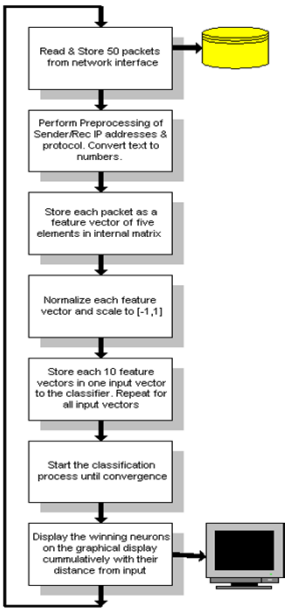
8

Why Self-Organizing Maps ?

- Unsupervised learning using SOM provide a simple and efficient way of classifying data sets. This is needed for Real-time performance.
- SOM are best suited due to their high speed and fast conversion rates as compared with other learning techniques.
- SOM preserve topological mapping between representations

9

Block Diagram of NSOM



10

Data Collection & Preprocessing

- Used a host PC running Linux as our primary test bed with an Ethernet controller connected to a sub net that has tens of other hosts.
- Used tcpdump to filter and collect all network traffic to or from our host. It runs as a background process and dumps every 50 packets into a file continuously.

11

Data Collection & Preprocessing

- Feature Selection
  - We select the following features from every packet for further preprocessing
    - IP address of destination (least significant 2 bytes)
    - IP address of source (least significant 2 bytes)
    - Protocol type

12

Data Collection & Preprocessing

- Packet preprocessing
  - A feature vector representing a packet consists of five values
  - Due to the large variations of these numbers we normalize each vector to be in the range [0,1]
  - We further scale the vectors to be in the range [-1,1]. This provides better performance of the SOM classifier.

13

Data Collection & Preprocessing

- Each 10 vectors are combined together as a single input to the classifier
- Several of the input vectors are presented to the classifier to form the clustering map

14

Time Representation

- We did not use explicit time of packet arrival and departure to represent time.
- We used an implicit time representation scheme
  - *n* successive packet features are gathered to form one input vector to the classifier
  - The value we chose for *n* in our experiment was 10

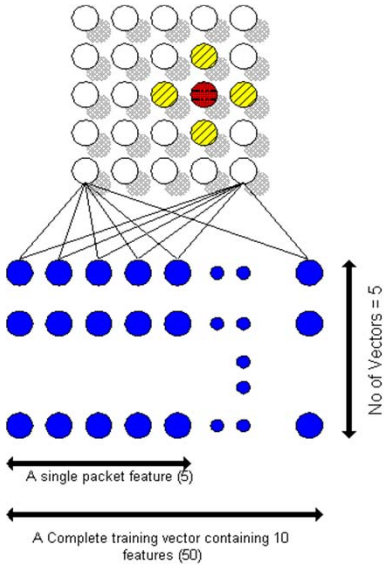
15

SOM Structure

- We experimented with two SOM structures: Linear and Diamond structures.
- Diamond structures gave better classification results
  - We updated the winning neuron along with a neighborhood distance of R = 1.
  - This updates the top, bottom, left and right neurons of the winning neuron, which resembles a diamond like structure.
  - We used 25 output neurons and η = 0.6.

16

SOM Structure



17

Classification Process

- After *m* successive input vectors are collected, normalized and scaled, the process of classification is started until we reach convergence.
- When conversion is reached, neuron values and their locations are sent to a graphical tools where they are displayed in 2 dimensional form
- The display maintains the old values as well to show the clustering and accumulation effects.

18

Results

- First obtained sample results statically by collecting different sample traffic representing normal as well as **Denial of Service (DoS)** attacks.
- Looked at the output of the classifier and noticed that all normal network traffic was clustered roughly between neurons 5 and 16
- We then subjected the classifier to various simulated DoS attacks, such as frequent SYN packets and heavy ping (ICMP req), neuron activities began to be scattered much outside the normal cluster window, roughly between neurons 0, 18 indicating a possible attack

19

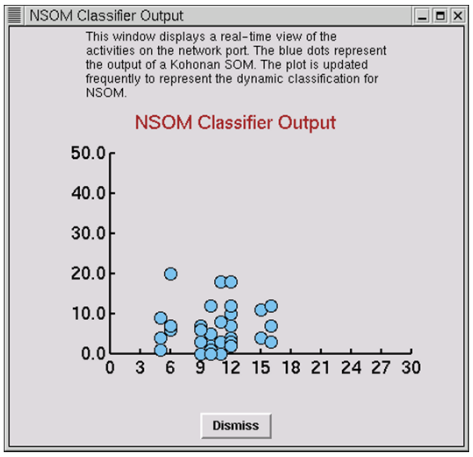
Results

- When we were more confident about the results, we tested NSOM in real-time.
- Similar behavior as with static testing was noted.

20

Results

Distance of the winning neuron with respect to the input vector

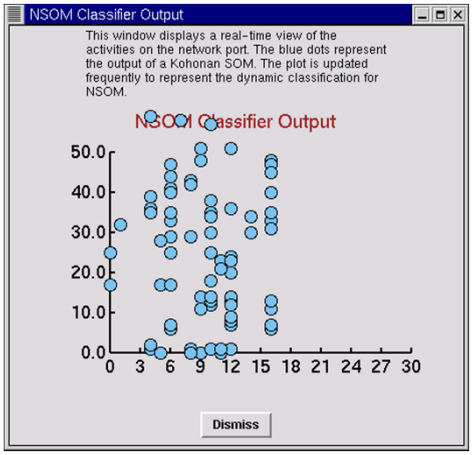


Output of classifier for normal traffic

21

Results

Distance of the winning neuron with respect to the input vector



Output of classifier for a simulated DoS attack

22

Results

- It is interesting to note that the Y values, on the graph, of the attack neurons were much higher than those for normal ones. Since Y value represents the distance of the winning neuron with respect to the input vector, we can conclude that these high Y value neurons represent uncommon and irregular behavior and therefore a possible attack.

23

Conclusion

- We described and implementation of a prototype system for classifying real-time network traffic using Self-Organizing Maps for the purpose of intrusion detection.
- We presented motives behind using unsupervised learning, our data collection and preprocessing procedures, how we represented time and displayed the results.
- We discussed the structure for our SOM.
- Results showed that we were able to classify simulated DoS attacks graphically as opposed to normal traffic, by showing different clustering of output neurons.

24

Positive Side of SOM

- Excellent for classification problems
- Greatly reduces computational complexity
- High sensitivity to frequent inputs
- New ways of associating related data
- No need of supervised learning rules

25

Negative Side of SOM

- system is a black box
- error rate may be unacceptable
- no guarantee of network convergence for higher dimension networks
- many problems can't be effectively represented by a SOFM
- a large training set may be required
- for large classification problems, training can be lengthy
- can be difficult to come up with the input vector.
- associations developed by SOFM not always easily understood by people.

26

Home Work:
Explore SOFM in Neural Networks Toolbox Environment

27

28