Dynamic Intrusion Detection Using Self-Organizing Maps

Peter Lichodzijewski
piotr@cs.dal.ca

A.Nur Zincir-Heywood
zincir@cs.dal.ca

Malcolm I. Heywood
mheywood@cs.dal.ca

Faculty of Comp. Science
Dalhousie University
Halifax, NS

Abstract – *A system is described for applying hierarchical unsupervised neural networks (self organizing feature maps) to the intruder detection problem. Specific emphasis is given to the representation of time and the incremental development of a hierarchy. Preliminary results are given for the DARPA 1998 Intrusion Detection Problem.*

I. Introduction

Defensive information operations and computer intrusion detection systems are primarily designed to protect the availability, confidentiality, and integrity of critical networked information systems. The two main classes of intrusion detection systems are those that analyze network traffic and those that analyze operating system audit trails. In all of these approaches, however, the amount of monitoring data generated is extensive, thus incurring large processing overheads. For instance, general rule-based systems aim to search/match for any "known abnormal behaviour" within the monitored data. Such systems will not be able to identify any "new abnormal behaviour". On the other hand, a statistical anomaly detection approach aims to identify the "normal behaviour" by mining the monitored behaviour of each user. Unfortunately, these systems further increase the processing overheads. A balance therefore exists between the use of resources and the accuracy and timeliness of intrusion detection information. The objective of the research presented in this paper is to construct an anomaly detection system that will highlight "abnormal behaviour" without incurring extensive computational overheads. To achieve this, hierarchical self-organizing maps (SOMs) are applied to the problem of host-based intrusion detection on computer networks. Originally, the proposed system was demonstrated on "real-time session information" of a host to detect potential intruders or abusers among the "common users" of the system [5]. In this framework is demonstrated on the 1998 DARPA Intrusion Detection data set. Specific recommendations are made regarding the representation of time, network parameters and SOM architecture.

2.Methodology

In this work, we aim to investigate the applicability of an entirely data driven machine learning paradigm of unsupervised, hierarchical, neural networks to the intrusion detection problem. In order to develop such a system, we first try to identify the "characteristics of the normal connection" to the target host. This information is then used to raise a flag for any connection identified as having a "different characteristic". To achieve this, the framework of figure 1 is followed, in which the core of the approach is to automate the identification of typical connections. The first problem is to establish the nature of initial information on which the rest of the system is based. For benchmarking purposes use is made of the DARPA 1998 Intrusion Detection Evaluation data set [2]. This represents TCP dump data generated over nine weeks of simulated network traffic in a hypothetical military local area network. This data was processed into some 7 million TCP connection records for use in the 3rd International Knowledge Discovery and Data Mining Tools Competition in 1999 [2].

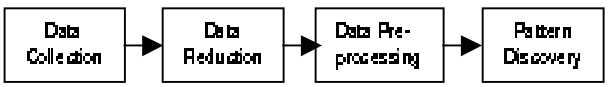

Figure 1. System flowchart.

Steps to achieve Data Reduction and Pre-processing are driven by the needs of the Pattern Discovery component. In this case, Pattern Discovery employs an unsupervised learning system – Self Organizing Maps (SOM) – to detect and visualize the characteristics of a common connection. SOMs represent an unsupervised learning technique for data analysis and visualization. They are of particular interest here on account of their efficient update scheme and ability to express topological relationships. This property of an SOM makes it very convenient for expressing relationships between different groups of connections. The hypothesis is that typical connection characteristics will be emphasized – densely populated regions of the map – whereas atypical activities will appear in sparse regions of the topology.

A. Data Set

The DARPA 1998 Intrusion Detection Evaluation data set consists of about 5 million connections of labeled training data and 2 million connections of test data [2]. Note however that the labels are only used to filter the data utilized during training (unsupervised learning does not require a label) and aid the interpretation of the trained SOM. Each connection is detailed in terms of 41 features, categorized as follows: Basic TCP features, Content features, Time-based traffic features; and Host-based traffic features [4].

Of these four sets of features only the "Basic TCP features" were employed. The three other sets of features are all derived using *a priori* knowledge regarding useful entities on which to build data-mining solutions [2]. As indicated above, in this work we are interested in identifying just how much can be achieved using an entirely data driven, unsupervised learning approach. Six features comprise the Basic TCP information, as follows,

1. Duration – the length (in seconds) of the connection;
2. Protocol type – the protocol type of the connections such as TCP or UDP;
3. Service – the service accessed by the connection such as HTTP or Telnet;
4. Flag – the status flag of the connection;
5. Destination Bytes – the amount of data sent by the destination of the connection;
6. Source Bytes – the amount of data sent by the source of the connection.

Training data either represented a normal connection or one of 24 different attack types. Test data was augmented with an additional 14 unseen attack types. All the forms of attack fell into one of three categories: Remote-to-Local (R2L); User-to-Root (U2R); Denial-of-service (DOS); or Probing [4].

B. SOM Architecture

Previous work [5] identified the appropriateness of a hierarchical unsupervised neural network architecture based on Kohonen's Self Organizing Map (SOM) [1] and Potential function clustering [6]. This architecture basically consists of two levels. Level one consists of feature specific SOMs that is for the six basic TCP features individual SOM's are developed to act as feature detectors over a fixed temporal horizon. The second layer consists of an integrating SOM, which is responsible for combining features detected by the six first level maps into a single 'view'. Between the two layers, Potential function clustering is employed to quantize the number of inputs 'seen' by the second layer. Based on the organization of the second layer map network administrators make decisions regarding the connection behaviors.

1) ***Data Pre-processing and Training of the First Level SOMs***

Three basic forms of pre-processing are performed before the first level SOMs receive data. Firstly, the attack connections are removed from the training data, taking care to preserve order. This means that the following SOMs spread across the 'domain' of normal behaviors. Previous work indicated that nodes of the SOM in sparse regions of the SOM topology then correspond to unusual behaviors [5]. The second pre-processing step is to separate the six basic TCP features and enumerate their values. That is, the discrete values of the basic TCP features in the training set are mapped to an integer. In the case of a discrete value which was not previously seen (in the test set), we default to mapping to the next lowest available integer. The result of the first two pre-prossessing stages is therefore six separate sequences of numbers, one for each basic TCP feature, with the nth entry of a sequence corresponding to the nth connection.

The third and final pre-processing operation is performed to provide the concept of time. This is particularly important, as SOMs have no implicit ability to recall temporal information. We note however, that we are not

interested in specific time stamp information, but the relative order of arrival. To this end a first-in first-out (FIFO) buffer is employed [5]. Such a FIFO consists of a series of inter-stage delay and 'tap' from each stage. Features propagate left to right through the FIFO. The SOM receives as input the feature currently at each 'tap' position; as in a shift register structure. The inter-stage delay is of 4 samples, where there are a total of 20 'taps'. As new connections are made the current contents of the FIFO shift to the right one location, the value of the last 'tap' being pushed off the end of the FIFO. This means that the SOM has a 'sequence horizon' spanning the last 80 connections, sampled at every fourth connection. Using such a scheme, the SOM is able to detect patterns over sequences of connections. The overall result of level one pre-processing is therefore six sets of 20 dimensional patterns, one for each basic TCP feature.

Each of the six first level SOMs consist of 36 nodes and are trained on approximately 25000 of the connection patterns from the training set, where this is roughly 2% of the original DARPA training data. A training cycle would consist of 4000 epochs, extending the training time beyond this did not appear to significantly improve the organization of the maps. All training was performed under the Matlab computing environment, using the SOM Toolbox developed at the Helsinki University of Technology [3].

2) ***Data Pre-processing for Training Second Level SOMs***

As indicated above, the second level SOM acts as an integration stage for providing a unified view of the network connection condition. To do so, each SOM node in the first layer represents a potential input to the second stage, or a total input vector length of 216 to the second layer SOM. To this end the Potential Function clustering method is employed to reduce the dimension of each layer one SOM from 36 to 6. That is to say, in the case of each layer one SOM, the distance, d, to each of the 6 cluster centers is calculated and then normalized to the unit interval,

$$d' = \frac{1}{1+d}$$

Thus SOM activities close to the center of one of the clusters tend to a normalized distance of unity, whereas SOM activities distant from a center tend to zero. The additional effect of this process is that each of the six layer one SOMs provide input to the second layer SOM using the same range of activation, avoiding domination of the second level SOM by level one SOMs with large dynamic range of activation.

3) ***Training Second Level SOM***

No additional representation of time is performed between first and second level SOM. The resulting second level SOM has 36 nodes. As in the case of the first level maps, 10000 epochs was sufficient to train the second level map. Figure 2 summarizes the distance between adjacent nodes of the layer two SOM. Of interest are the light regions of the map, or nodes, which are distant with respect to their neighbors. Thus nodes 31, 32 and 33 are the most distant with nodes 20, 26 and 27 forming a second set of isolated nodes.

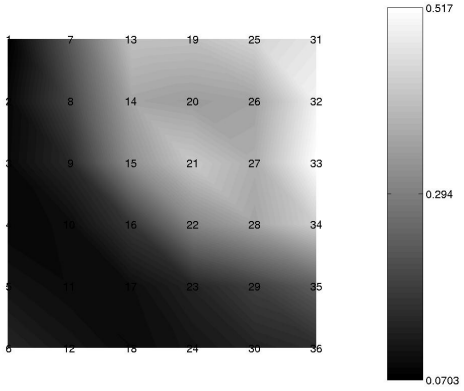

Figure 2. Unified matrix of second level SOM.

3.Results

Figure 3 summarizes the number of time that each node of the second level SOM represented the winning or best matching unit (BMU) under training data without attack connections. None of the low frequency BMUs lie in the dark regions of the adjacency distance plot, figure 2. Figure 4 plots the BMUs for the training data in which only patterns (each pattern consists of values from several connections) with at least one component of attack were present. This specifically identifies nodes 32 and 33 as synonymous with attack connections, where these are also two of the three most distant level two SOM nodes, figure 2.

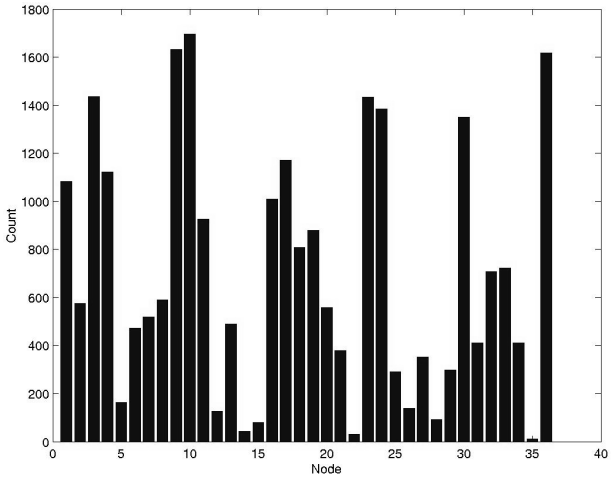

Figure 3. BMU frequency on normal training data (no attack connections).

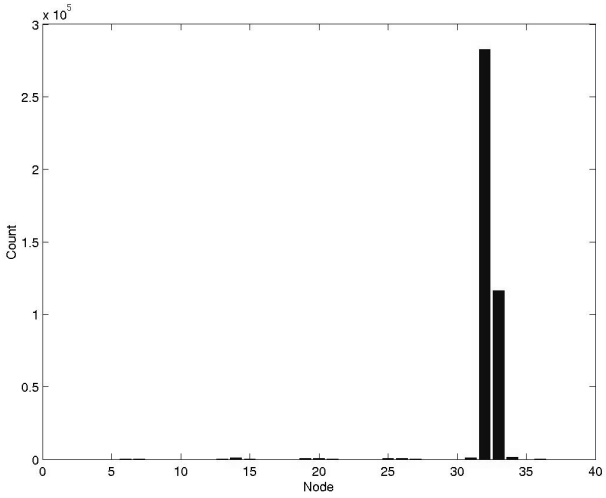

Figure 4. BMU frequency on attack training data (no normal connections).

We are now in a position to begin to suggest appropriate rules for describing attacks in terms of BMUs. To do so, the sequence of BMUs preceding and following an attack, as well as the BMUs associated with an attack are used. Tables 1 and 2 summarize the first, second and third BMUs occurring most frequently for patterns before and after node 32 or 33 are the BMU on training data, exclusive of attack connections. This provides a filter for normal behavior currently resulting in false positives. The rule for labeling an attack connection now has the formulation,

IF (node 32 or 33 is the BMU)
THEN IF (second and third BMUs do NOT match table 1 or 2)
THEN (label connection as an attack)

	Previous	Current	Next
First BMU	32	32	32
Second BMU	25	25	25
Third BMU	26	26	26

Table 1: BMU's before and after node 32

	Previous	Current	Next
First BMU	32	33	33
Second BMU	34	34	34
Third BMU	27	27	27

Table 2: BMU's before and after node 33

Naturally, the number of patterns over which the above type of rule is formulated may be extended. Table 3 summarizes the performance of the system on 10% of the test data using different pattern limits. In the case of zero patterns (node 32 and 33 BMUs alone indicate an attack), 15308 of the total 250399 attacks are missed. Moreover, in comparison to present best practice [4], the overall best false negative rates previously reported were ≈ 0.33 with a false positive rate of 0.0002. Note however, that the system returning these results utilized all four categories of information in the DARPA dataset or a total of 41 features (only 6 were used here) and was trained over the entire training data set (only ≈10% was used in this work), and was tested over the whole test data set.

# of Patterns	# of FPs	# of FNs	FP Rate	FN Rate
0	20588	15308	0.3404	0.0611
3	7674	81881	0.1269	0.3270
5	6877	82222	0.1137	0.3284
10	5197	83117	0.0859	0.3319
25	2968	84523	0.0491	0.3376
50	1543	85854	0.0255	0.3429
75	587	86837	0.0097	0.3468
100	121	87475	0.0020	0.3493

Table 3: Final Results

4. Conclusion

The work presented is naturally of a preliminary nature, but we believe sufficient to warrant continued development. In particular we have demonstrated that a hierarchically built unsupervised neural network approach is able produce encouraging results. Future work will naturally extend the nature of the tests conducted and investigate the use of more advanced SOM architectures and additional layers to the hierarchy.

References

[1] R. N. Dave and R. Krishnapuram, "Robust clustering method: a unified view," *IEEE. Trans. on Neural Networks*, 5(2): 270-293, 1997.

[2] The Third International Knowledge Discovery and Data Mining Tools Competition, http://kdd.ics.uci.edu/databases/kddcup99.kddcup99.html, May 2002.

[3] S. Kaski, "Data exploration using Self-Organizing maps," *Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series No. 82*, March 1997.

[4] W. Lee, S. J. Stolfo and K. W. Mok, "Mining in a data-flow environment: experience in network intrusion detection," in *Knowledge Discovery and Data Mining*, pp. 114-124, 1999.

[5] P. Lichodzijewski, A. n. Zincir-Heywood and M. I. Heywood, "Host-based intrusion detection using self-organizing maps," *Proceedings of the 2002 IEEE World Congress on Computational Intelligence*, 2002 (in press).

[6] S.L. Chiu, "Fuzzy Model Identification based on Cluster Estimation," *Journal of Intelligent and Fuzzy Systems*, 2: 267-278, 1994.