Available online at www.sciencedirect.com

ScienceDirect

Engineering Applications of Artificial Intelligence 20 (2007) 439–451

Engineering Applications of
ARTIFICIAL INTELLIGENCE

www.elsevier.com/locate/engappai

A hierarchical SOM-based intrusion detection system

H. Gunes Kayacik, A. Nur Zincir-Heywood, Malcolm I. Heywood*

Dalhousie University, Faculty of Computer Science, 6050 University Avenue, Halifax, NS, Canada B3H 1W5

Received 25 May 2005; received in revised form 7 September 2006; accepted 19 September 2006
Available online 13 November 2006

Abstract

Purely based on a hierarchy of self-organizing feature maps (SOMs), an approach to network intrusion detection is investigated. Our principle interest is to establish just how far such an approach can be taken in practice. To do so, the KDD benchmark data set from the International Knowledge Discovery and Data Mining Tools Competition is employed. Extensive analysis is conducted in order to assess the significance of the features employed, the partitioning of training data and the complexity of the architecture. Contributions that follow from such a holistic evaluation of the SOM include recognizing that (1) best performance is achieved using a two-layer SOM hierarchy, based on all 41-features from the KDD data set. (2) Only 40% of the original training data is sufficient for training purposes. (3) The 'Protocol' feature provides the basis for a switching parameter, thus supporting modular solutions to the detection problem. The ensuing detector provides false positive and detection rates of 1.38% and 90.4% under test conditions; where this represents the best performance to date of a detector based on an unsupervised learning algorithm.
© 2006 Elsevier Ltd. All rights reserved.

Keywords: Intrusion detection systems; Self-organizing map; Unsupervised learning; Benchmarking; Hierarchical neural network

1. Introduction

The Internet, as well as representing a revolution in the ability to exchange and communicate information, has also provided greater opportunity for disruption and sabotage of data previously considered secure. The study of systems able to detect network borne intrusions provides many challenges. Classical network-based approaches to this problem often rely on either rule-based misuse detection or anomaly detection (Bass, 2000). Rule-based misuse detection systems attempt to recognize specific behaviors that represent known forms of abuse or intrusion. On the other hand, anomaly detection attempts to recognize abnormal user behavior. Both approaches have their respective advantages and disadvantages. Rule-based systems typically require an exhaustive list of templates characterizing each attack instance; there is no concept of similarity to a currently listed attack instance. The anomaly detection approach will actually identify "normal" behaviors by mining the monitored behavior of each user so that "abnormal" behaviors can be characterized. Clear distinctions between normal and abnormal, however, are difficult to achieve in practice.

Given the significance of the intrusion detection problem, there have been various initiatives that attempt to quantify the current state of the art. In particular, the International Knowledge Discovery and Data Mining Tools Competition (Hettich and Bay, 1999) provided the KDD-99 data set for assessing different AI approaches to the problem. Although not without its drawbacks (McHugh, 2001), this benchmark provides the only *labeled* data set for comparing IDS systems, which the authors are aware of, and is therefore utilized in this work. This enables the establishment of a clear comparison with alternative solutions trained on the KDD data set.

The motivation in this work lies in the desire to provide a thorough evaluation of an unsupervised learning approach to the IDS problem. To do so, a self-organizing feature map (SOM) hierarchy is utilized, where the appropriateness of such a scheme in the analysis of large data sets has already been demonstrated (e.g. WEBSOM for large document collections (Kohonen et al., 2000)). Moreover, the initial application of SOM architectures to IDS problems has also already been reported (Sarasamma et al., 2005; Ramadas et al., 2003; Kayacik et al., 2003;

*Corresponding author. Tel.: +1 902 4942951.
E-mail address: mheywood@cs.dal.ca (M.I. Heywood).

0952-1976/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engappai.2006.09.005

Litchodzijewski et al., 2002; Rhodes et al., 2000). Contributions made by this work therefore do not focus on whether the SOM model may provide the basis for an IDS detector, but instead focus on establishing how practitioners answer key design issues of specific influence to an SOM-based detector. The issues addressed by this work take the following form:

1. how should the training data be partitioned;
2. may detectors based on 'basic' features compete with those of features constructed with a priori knowledge;
3. which of the 'basic features' in the KDD data set are the most significant;
4. what complexity of SOM is necessary to be competitive with the best results achieved on the KDD data set.

In seeking answers to these questions, two systems are considered: an SOM architecture in which minimum domain knowledge is available to select applicable features and a second in which all the features supplied in the KDD benchmark are utilized. Specific contributions include (a) *Domain knowledge*—6-basic features are sufficient for recognizing a wide range of denial of service attacks. In total, 41-features are necessary to minimize False Positive rates. (b) *Architectural complexity*—2 SOM layers are sufficient when using all 41-features, whereas 3 layers are necessary when using the 6 'basic' KDD features. (c) *Feature significance*—Protocol and service appear to be the most significant of the 6 basic features, where this is (currently) indicative of the service—protocol specific nature of attacks. (d) *New detector accuracies*—Overall performance on the "Corrected (test) KDD" test set were 4.6% False positive (89% detection) rate with a detector based on 6-features and 1.38% false positive (90.4% detection) rate when basing detectors on all 41-features. This represents the best performance to date of a detector based on an unsupervised learning algorithm.

The remainder of the paper is organized as follows. Section 2 provides the background and methodology of the work. Details of each learning algorithm comprising the system are given in Section 3. Results are reported in Section 4 and Conclusions drawn in Section 5.

2. Methodology

As indicated in Section 1, the basic objective of this work is to assess just how far a machine learning approach based on unsupervised learning may be taken. To this end, an approach based on Kohonen's topological feature maps is employed. This assumes that given sufficient resolution in the maps, it is possible to associate different behaviors with different nodes in the feature map. Moreover, the significance of domain knowledge in the selection of appropriate features is also of interest. A previous work established that by utilizing a shift register to embed the temporal relationship between incoming connections, described in terms of session information, a simple two layer SOM hierarchy was sufficient to distinguish between different behaviors using only 6-features (Litchodzijewski et al., 2002). However, the data set used in that scheme only consisted of seven attacks. In this work, a thorough study is performed of the hierarchical SOM methodology on the KDD-99 benchmark. To this end, the characteristics of the data set are first described and then the SOM architecture utilized defined.

2.1. KDD data set

The KDD-99 data set is based on the 1998 DARPA initiative to provide designers of intrusion detection systems (IDS) with a benchmark on which to evaluate different methodologies (MIT Lincoln Lab, 1998). To do so, a simulation is made of a factitious military network consisting of three 'target' machines running various operating systems and services. An additional three machines are then used to spoof different IP addresses, thus generating traffic between different IP addresses. Finally, there is a 'sniffer' that records all network traffic using the *tcpdump* format. The total simulated period is seven weeks. Normal connections are created to profile that expected in a military network and attacks fall into one of five categories: user to root; remote to local; denial of service; data; and probe.

- *Denial of service* (*dos*): Attacker tries to prevent legitimate users from using a service.
- *Remote to local* (*r2l*): Attacker does not have an account on the victim machine, hence tries to gain access.
- *User to root* (*u2r*): Attacker has local access to the victim machine and tries to gain super user privileges.
- *Probe*: Attacker tries to gain information about the target host.

Note that user to root and remote to local can represent content-based attacks, and may therefore only be detected indirectly when the detector is limited to 6 "basic features" (e.g. guessing passwords often manifests itself as multiple attempted login's between the same source destination pair).

In 1999, the original TCP dump files were preprocessed for utilization in the intrusion detection system benchmark of the International Knowledge Discovery and Data Mining Tools Competition (Hettich and Bay, 1999). To do so, packet information in the TCP dump file are summarized into connections. Specifically, "a connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows from a source IP address to a target IP address under some well defined protocol" (DARPA Archive, 1999). This process is completed using the Bro IDS (Bro, 1999), resulting in nine "Basic features of an Individual TCP connection" (DARPA Archive, 1999); hereafter referred to as 'basic features':

- duration of the connection;
- protocol type, such as TCP, UDP or ICMP;

- service type, such as FTP, HTTP, Telnet;
- status flag, derived by Bro to describe a connection;
- total bytes sent to destination host;
- total bytes sent to source host;
- whether source and destination addresses are the same or not;
- number of wrong fragments;
- number of urgent packets.

Note that only protocol and service features are not derived i.e. they are estimated immediately as opposed to after a connection has completed. Moreover, the above 'status flag' should not be confused with the TCP/IP flag of the same name. Finally, the last three features are specific to certain attack types (no variation is observed across the normal data in the training set). In line with a minimum a priori assumption, these features are also ignored. Thus, one detector will be built using the first six basic features, the case of minimum domain knowledge.

In addition to the above nine 'basic features,' each connection in the KDD data set is also described in terms of an additional 32 *derived* features, falling into three categories:

- *Content features*: Domain knowledge is used to assess the payload of the original TCP packets. This includes features such as the number of failed login attempts.
- *Time-based traffic features*: These features are designed to capture properties that mature over a 2 s temporal window. One example of such a feature would be the number of connections to the same host over the 2 s interval.
- *Host-based traffic features*: Utilize a historical window estimated over the number of connections—in this case 100—instead of time. Host-based features are therefore designed to assess attacks, which span intervals longer than 2 s.

In this work, the above information is considered as domain knowledge and is used to support the second SOM architecture. The following study compares the case of two SOM architectures: one built from the six 'basic features' alone (6-feature architecture) and the second from the complete set of 9 basic and 32 derived features (41-feature architecture).

2.2. Hierarchical SOM

As in our earlier work, a hierarchical SOM architecture is employed (Kayacik et al., 2003; Litchodzijewski et al., 2002). Such architecture is motivated by the success of other hierarchical neural architectures, such as those used in hierarchical vector quantization (Luttrell, 1989), real-time optical character recognition (Sackinger et al., 1992) and fingerprint categorization (Halici and Ongun, 1996). Within the context of intrusion detection, the architecture is designed to correlate network behavior across multiple features of relevance to specific attack types with greater specificity as the hierarchy progresses from layers 1 to 3, Fig. 1. Moreover, such a scheme facilitates the distribution of computational overheads in constructing such a detector. At layers 1 and 2, relatively small SOMs are utilized (6×6) as training is conducted across all 'P' exemplars in the (training) data set ($P \approx 500{,}000$). SOMs in the last layer are only constructed over the subset of exemplars for which a neuron in layer 2 is the 'best matching unit (BMU)' (P', P'', etc., where all are significantly smaller than P). This means that the SOMs at the last layer can be larger (20×20) than used in layers 1 and 2, thus improving the resolution hence discriminatory capacity of the SOM, whilst still requiring less training overhead.

As indicated in the introduction, two feature scenarios are considered, that of a system based on the 6 basic features, and that utilizing domain knowledge to build a total of 41-features. The 41-feature scenario supports the characterization of network behavior over different time lines, the Traffic-based features of Section 2.1. In order to support the characterization of temporal behavior in the 6-feature case, a tapped delay line of a priori depth is utilized, Fig. 2. Thus, individual SOMs are associated with each of the six basic TCP features with the objective of correlating temporal properties specific to each feature. In the case of the 41-feature architecture, the derived features provide a description of temporal properties of the connections. This removes the requirement for the first layer of the hierarchy. Thus under the 41-feature scenario, layers 2 and 3 are sufficient.

2.3. Preprocessing and clustering

In order to build the hierarchical SOM architecture, several data normalization operations are necessary. This

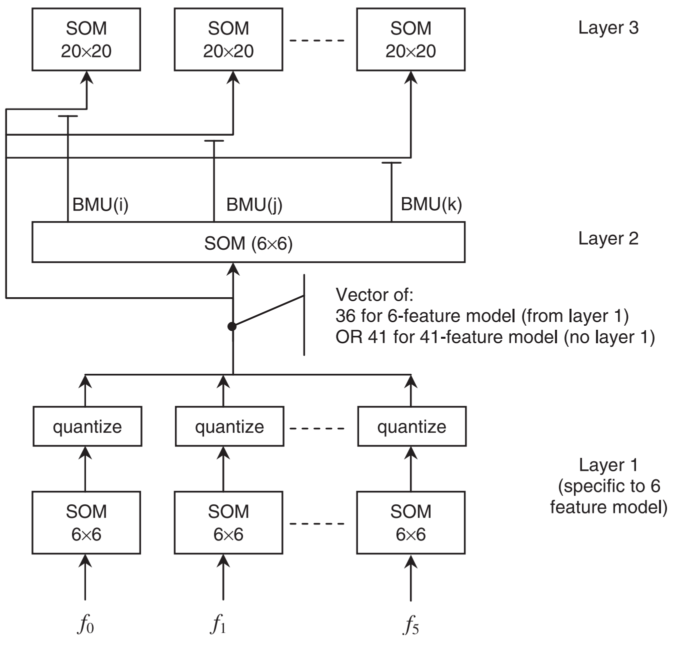

Fig. 1. Hierarchical SOM architecture: (a) architecture and (b) data partitioning.

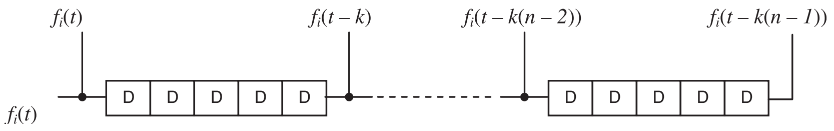

Fig. 2. Tapped delay line of 6-feature architecture. 'n' taps at modulo 'k' over tapped delay line of length 'l' layer 1 SOM input for feature 'i'.

provides for the initial temporal preprocessing and then inter-layer 'quantization' between the first and the second layer maps. Preprocessing has two basic functions, to provide a suitable representation for the KDD data and support the representation of time. In the case of the KDD data representation, three of the basic features—protocol type, service type and status flag—are alphanumeric. As the first SOM layer treats each feature independently, each instance of an alphanumeric character is merely mapped to sequential integer values. Numerical features—connection duration, total bytes sent to destination/source host—are used unchanged (such preprocessing is naturally also applicable for the 41-feature architecture).

In the case of a suitable temporal representation (6-feature architecture only), the standard SOM used here has no capacity to recall histories of patterns directly. However, a previous work identified that sequence as opposed to time stamp, is the property of most significance in the IDS problem (Litchodzijewski et al., 2002). A shift register of length '*l*' is therefore employed in which a 'tap' is taken at a predetermined repeating interval '*k*' such that $l \% k = 0$, where % is the modulus operator. The first-level SOMs only receive values from the shift register that correspond to tap locations, Fig. 2. Thus, as each new connection is encountered (enters at the left), the content of each shift register location is transferred one location (to the right), with the previous item in the *l*th location being lost.

In case of the 6-feature architecture, it is necessary to 'quantize' the number of neurons between first and second-level SOMs. Specifically, the purpose of the second-level SOM is to provide an integrated view of the input feature specific SOMs developed in the first layer, Fig. 1. There is therefore the potential for each neuron in the second layer SOM to have an input dimension defined by the total neuron count across all first layer SOM networks. This would be a brute force solution that does not scale computationally (there are half a million training set patterns). Moreover, given the topological ordering provided by the SOM, neighboring neurons will respond to similar stimuli. The topology of each first layer SOM is therefore quantized in terms of a fixed number of neurons using a clustering algorithm, Fig. 1. This reduces the number of inputs to the second layer SOM from 216 $(6 \times 6 \times 6)$ to 36 (6×6).

3. Learning algorithms

Two learning algorithms are used to build the hierarchical SOM architectures. The first is used to train each SOM in the hierarchy (Kohonen, 2000), Section 3.1. The second is a clustering algorithm that is used to quantize the number of SOM neurons 'perceived' by the second layer for the 6-feature architecture alone, Fig. 1, Section 3.2. In the latter case, the potential function algorithm is employed (Chiu, 1994), although no special significance is attributed to this particular method. Finally, in order to build SOMs associated with the last layer of the hierarchy, a decision rule is necessary for defining under what conditions this takes place. The scheme used in this work is summarized in Section 3.3.

3.1. Self-organizing feature map

Kohonen's SOM algorithm is an unsupervised learning algorithm in which an initially 'soft' competition takes place between neurons to provide a topological arrangement between neurons at convergence (Kohonen, 2000). The learning process is summarized as follows:

1. Assign random values to the network weights, w_{ij}.
2. Present an input pattern, $\boldsymbol{x}$. In the case of the 6-feature architecture this is the series of 20 previous connection features provided by the current shift register taps. In the case of the 41-feature architecture this is a vector composed from the 9 'basic' and 32 derived KDD features.
3. Calculate the distance between pattern, $\boldsymbol{x}$, and each neuron weight $\boldsymbol{w}_j$, and therefore identify the winning neuron, or
$$d = \min\{\|x - w_j\|\}, \tag{1}$$
where $\|\cdot\|$ is the Euclidean norm and $\boldsymbol{w}_j$ is the weight vector of neuron *j*.
4. Adjust all weights in the neighborhood of the winning neuron, or
$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)K(j,t)\{x_i - w_{ij}(t)\}, \tag{2}$$
where $\eta(t)$ is the learning rate at epoch *t*; and $K(j, t)$ is a suitable neighborhood function.
5. Repeat steps (2)–(4) until the convergence criterion is satisfied.

Following convergence, presentation of an input vector, $\boldsymbol{x}$, results in a corresponding output vector, $\boldsymbol{d}$, the Euclidian distance between each neuron and input. The neuron with the smallest distance represents the winning or BMU, step (3). The best matching neuron also defines the relative position from which the neighborhood of neurons, $K(j, t)$, is defined. Once the maps are trained, it is this concept of a

BMU that is used to facilitate the labeling of the second- and third-level maps.

3.2. Potential function clustering

The potential function clustering algorithm consists of four steps (Chiu, 1994):

1. Identify the potential of each data point relative to all other data points. All data points represent candidate cluster centers.
2. Select the data point with largest potential and label as a cluster center.
3. Subtract the potential of the data point identified at step (2) from all others and remove this point from the list of candidate cluster centers.
4. Repeat on step (2) until the end criterion is satisfied.

In this application, the set of data points correspond to the set of neurons in each (first layer) 6-feature architecture SOM, where the weights of each neuron describe a neuron position in terms of the original input space. Step 1 characterizes neurons in terms of how close they are to others. A neuron with many local neighbors should have a high 'potential' as expressed by a suitable cost function, or

$$P_t(w(j)) = \sum_{i=1}^{M} \exp\left(-\alpha\left\|w(i) - w(j)\right\|^2\right),$$

where $w(j)$ is the 'j'th SOM neuron, $P_t(w(j))$ is the potential for such neuron at iteration t, M is the number of data points (in this case SOM neurons), and α is the cluster radii.

Step 2 identifies a candidate cluster center (SOM neuron) by choosing the point with largest potential $P_t(x^*)$. Step 3 removes the influence of the chosen neuron from the remaining (unselected) set of SOM neurons. That is, the remaining neurons have their respective potentials decreased by a factor proportional to the distance from the current cluster center, or

$$P_{t+1}(w(j)) = P_t(w(j)) - P_t(w^*)\exp\left(-\beta\left\|w(i) - w(j)\right\|^2\right),$$

where $t+1$ is the index of the updated potential at iteration t; w^* is the data point associated with the current cluster center, and β is the cluster radii ($\alpha < \beta$).

The result of step 3 is the labeling of a specific SOM neuron as a cluster center. Step 4 iterates the process in conjunction with some suitable stop criterion. In this case, stopping when six cluster centers are identified, where the alpha and beta values are set accordingly. That is to say, further cluster centers correspond to points with a potential value less than 10% of the first potential located. The net effect of this process is therefore that each of the six first layer SOMs are characterized in terms of 6 SOM nodes, resulting in a total of 36 inputs to the second-level SOM.

Once the 6 cluster centers are identified for each SOM, representing the 'quantized' SOM output, and normalizing as follows:

$$y = \frac{1}{1 + \|w - x\|},$$

where w is the cluster center and $\boldsymbol{x}$ is the original input.

The second layer SOM now receives a vector, $\boldsymbol{y}$, of the form

$$y = [y_{1,1,\ldots,}\, y_{1,6,}\, y_{2,1,}\, \ldots,\, y_{i,j}],$$

where i is the SOM index and j is the cluster (neuron) index, $i, j \in [1,\ldots,6]$.

3.3. Decision rule for building last layer SOMs

Neurons in the second layer will ideally act as the BMU for exemplars with the same class label (normal, DoS, probe, U2R, R2L), thus maximizing the detection rate and minimizing the number of false positives. However, there is no guarantee that this will be the case. In order to resolve this, second layer SOM neurons that act as a BMU for exemplars from more than one class are used to partition the data, Fig. 1. Thus, third layer SOMs are trained on subsets of the original (training) data. This enables the size of the third layer SOMs to increase, thus improving the class specificity, whilst still presenting a reasonable computational cost as they are only trained on subsets of the original training data. Once training is complete, second layer BMUs act to identify which exemplars are forwarded to corresponding third layer SOMs on unseen data (test set).

A decision rule is required to determine under what conditions the classification performance of a BMU at the second layer SOM is judged sufficiently poor for association with a third layer SOM. There are several aspects that require consideration, including:

- What is the local misclassification rate of the second layer BMU—relative to the number of exemplars labeled at the second layer BMU, what is the minimum acceptable misclassification rate.
- How many (training) exemplars does a candidate second layer BMU represent—if there are only 10 exemplars for which the second layer neuron is the BMU, association with a third layer SOM would not be appropriate, even if the misclassification rate of the BMU was high.

To this end the following decision rule is used in this work—IF ((min(#*normal*, #*attack*) > 1,500) AND (*misclassification* > 3.5%)) THEN (build third layer SOM).

Where '#normal' ('#attack') are the count of exemplars associated with the (same) BMU and labeled as normal (attack $\in$ {DoS $\cup$ Probe $\cup$ U2R $\cup$ R2L}; and misclassification is the percentage of exemplars miss classified relative to the major class label of the exemplars associated with the BMU.

The basic implication of this rule is that there must be at least 1500 normal and attack connections associated with a second layer BMU for training of a corresponding third layer SOM, and the rate of misclassification over exemplars associated with the second layer BMU exceeds 3.5%.

4. Evaluation methodology

In all cases, the SOM Toolbox and SOM-PAC were employed for the design of each SOM comprising the SOM hierarchy for either architecture (SOM-PAC). In the following, the data set, training procedure and evaluation of the hierarchical SOM architecture are described.

4.1. KDD-99 data set

The KDD-99 data consists of several components, Table 1. As in the case of the International Knowledge Discovery and Data Mining Tools Competition, only the '10% KDD' data is employed for the purposes of training (Hettich and Bay, 1999). This contains 24 attack types and is essentially a more concise version of the 'Whole KDD' data set. One side effect of this is that it actually contains more examples of attacks than normal connections. Moreover, the attack types are not represented equally, with denial-of-service attack types—by the very nature of the attack type—accounting for the majority of the attack instances. However, the so-called 'Corrected (Test)' data set provides a data set with a significantly different statistical distribution than either '10% KDD' or 'Whole KDD' and in addition includes 14 attack types which are not encountered during training.

4.2. Training

Learning parameters for the SOMs are summarized in Table 2, where this process is repeated for each SOM comprising the hierarchy. In each case, training is completed in two stages, the first providing for the general organization of the SOM and the second for the fine-tuning of neurons. Table 3 summarizes the additional parameters utilized by the shift register and Potential Function clustering algorithm (6-feature architecture only). In each case, Potential Function parameters are adjusted until exactly 6 clusters are provided for each layer one feature; Section 3.2. All 6-feature architectures utilize a shift register of 96 elements and 20 taps ($l = 5$).

In the case of SOM hierarchies built from the 6 basic features a hierarchy consists of 6 SOM networks in the first layer (temporal encoding), each consisting of 6×6 grid and 20 inputs. On completion of training at the second layer, labeling takes place. That is, for each connection in the training set, the corresponding label is given to the BMU in the second layer. A count is retained for the number of normal and attack connections each BMU receives. Third layer SOMs are built for second layer SOM neurons that demonstrate significant counts for both attack and normal connections, as per the decision rule of Section 3.3. This results in 6 third layer SOMs being built on top of specific second layer neurons. In each case third layer SOMs consist of 20×20 neurons, where a larger neuron count is utilized in the third layer in order to increase the likelihood of separation between the two connection types. Moreover, only connections for which the corresponding second layer SOM is the best-matching unit are used to train third layer SOMs. This facilitates the use of larger SOMs at the third layer without experiencing a high computational overhead. Moreover, in each case, the inputs to third-level SOMs correspond to the 36-element vector of 'quantized' first layer outputs.

In Section 5.2, SOM hierarchies are also built from all 41 KDD-99 features. In this case, the first SOM layer is

Table 2
SOM training parameters

Parameter	Rough training	Fine tuning
Initial η	0.5	0.05
η decay scheme	$f(\text{epoch}^{-1})$	
Epoch limit	4000	
Neighborhood parameters		
Initial size	2	1
Function	Gaussian	
Relation	Hexagonal	

Table 3
Per feature potential function parameters

10% KDD	
α	1×10^{-6}, 1×10^{-3}, 2×10^{-7}, 1×10^{-6}, 16×10^{-6}, 0.1599015,
β	2×10^{-2}, 2×10^{-2}, 1×10^{-2}, 4×10^{-2}, 1×10^{-1}, 1×10^{-2}
Normal only	
α	5×10^{-7}, 1×10^{-5}, 1×10^{-5}, 1×10^{-5}, 33×10^{-7}, 6×10^{-6}
β	9×10^{-1}, 13×10^{-2}, 7×10^{-2}, 13×10^{-2}, 1×10^{-2}, 1×10^{-2}
50/50	
α	9×10^{-6}, 1×10^{-5}, 1×10^{-7}, 1×10^{-5}, 175×10^{-7}, 7×10^{-7}
β	2×10^{-1}, 15×10^{-2}, 7×10^{-2}, 13×10^{-2}, 1×10^{-1}, 1×10^{-2}

Table 1
Basic characteristics of the KDD data set

Data set label	Total	Normal (%)	DOS (%)	Probe (%)	U2R (%)	R2L (%)
10% KDD	494,020	19.79	79.2	0.8	0.01	0.2
Corrected (test)	311,029	19.58	73.9	1.3	0.02	5.2
Whole KDD	4,898,430	19.8	79.3	0.84	0.001	0.02

dispensed with, as temporal features are explicitly supported in the feature set, Section 2.1. Thus, a single SOM trained on all 41-features is equivalent to the second layer of the SOM trained on 6 basic features alone.

4.3. Performance metric

Performance of the classifier is evaluated in terms of the false positive and detection rates, estimated as follows:

$$\text{False positive rate} = \frac{\text{Number of false positives}}{\text{Total number of normal connections}},$$

$$\text{Detection rate} = 1 - \frac{\text{Number of false negatives}}{\text{Total number of attack connections}},$$

where false positive (negative) rate is the number of normal (attack) connections labeled as attack (normal). In addition the classification accuracy on categories of attack and attack types for the more difficult corrected (test) KDD data set are considered.

5. Results

The performance of the hierarchical SOM model will be reviewed from two perspectives: features, and hierarchy complexity. As such the study commences with the case of 6-features and a hierarchy consisting of layers 1 and 2 alone, Section 5.1. The case is then made for selectively building layer 3 SOMs, relative to the performance of layer 2 BMUs that 'win' for multiple classes, Section 5.1.2. Section 5.1.3 conducts a study on the sensitivity of the SOM detector to each of the 6 features, making recommendations on how further modularization of the detector would be possible. Findings on the 6-feature scenario are then summarized in Section 5.1.4.

Performance is then evaluated under the case of 41-features, Section 5.2, first using a single SOM, and then using a two-layer hierarchy, Sections 5.2.1 and 5.2.2 (layer 1 of the hierarchy is redundant). Finally, a comparison is made against previous machine learning results on the KDD'99 data set, for both solutions based on supervised and unsupervised learning algorithms, Section 5.3.

5.1. Architectures based on 6 basic features

5.1.1. Six-features, two layer SOM hierarchy, 3 data partitions

The 10% KDD data set was prepared by the organizers of the original competition to provide a concise version of the original data set (whole KDD), Section 2.1. However, the resulting data set is rather unbalanced, Table 1, with the majority of the data representing attack connections. Three partitions are therefore considered in this work: training on 10% KDD; training on the 97,277 'normal' connections alone, as taken from the 10% KDD data set, Table 1 (i.e. an anomaly detection system), or training on all 97,277 'normal' connections and the first 97,277 attack connections. These are hereafter referred to as '10% KDD', 'Normal', and 50/50, respectively.

Fig. 3 summarizes the distribution of attack and normal connections in the second-level SOM for the case of training on 10% KDD; where proportionally larger counts result a greater area of the hexagon being colored black. It is apparent that nodes 1, 32 and 36 account for most of the attack connections and neuron 19 most of the normal connections. It is also apparent that several neurons also respond to both normal and attack connections, where the worst cases are highlighted in gray, Fig. 3. Neuron labels are assigned on the basis of the majority class count at each neuron on training data (10% KDD in all cases irrespective of the partition used to train the architecture). Neurons responding to more than one class naturally lead to misclassification. Disambiguation of such cases will be considered in Section 5.2 when third layer SOM networks are selectively built for second layer neurons that respond to multiple classes. The overall result of the labeling activity is therefore that each second layer SOM node is labeled as either attack or normal as determined by the BMU count on a post training run through 10% KDD.

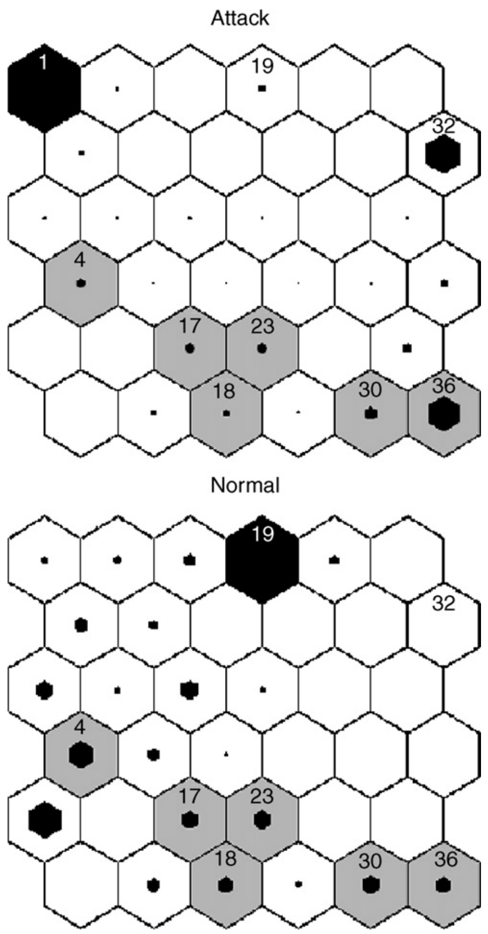

Fig. 3. Hit histogram of the second-level map (training data)—SOM hierarchy trained with 10% KDD data set.

The above process was repeated for the case of SOM hierarchies trained under the two alternative training set partitions: normal connections only (i.e. an anomaly detector) and 50/50. In each case, labeling of the alternative architectures was naturally conducted (post training) using all the 10% KDD training data. Table 4 summarizes performance of each architecture at the second layer on both whole KDD and corrected (test) KDD data sets. Table 5 provides an attack specific breakdown of corrected test set performance over the three training partitions. It is now apparent that the Normal partition results in better detection rates for probe and R2L whereas 50/50 is more reliable at U2R. The 10% KDD training partition records the best overall FP rate on account of a better performance on normal connections. Moreover, it is also apparent that 20–30% of U2R and 10–20% of R2L attacks are correctly identified. This is interesting as no direct support for content-based features is provided in the 6 basic features (U2R and R2L are content based), implying that indirect detection of content based attacks is possible. Finally, this should not be taken as indicating that random chance would provide a better detector, as these results need to be considered within the context of the overall detection and false positive rates, Table 4.

As indicated in Section 4.1, the corrected (test) KDD data set used to evaluate the performance of detectors contains an additional 14 attack types not included in the training data (10% KDD). Tables 6 and 7 detail the ability of the second layer of the three respective systems to detect attack types available and unavailable in the training data set. Moreover, Tables 6 and 7 detail the total number of instances of each attack type. In the following observations, the contribution from attacks with less than a 100 exemplars is therefore discounted i.e., no information regarding risk (significance) is available to bias the development of the detector towards the identification of such infrequent exemplars (as in the case of sufficient a priori knowledge to design a suitable weighted cost function).

Table 4
Performance of 2nd layer SOM under different training data partitions for case of 6 basic features—total FP and detection rate

Partition	FP rate (%)	Detection rate (%)
Whole KDD		
10% KDD	10.6	99.8
Normal only	15.7	99.9
50/50	8.25	99.8
Corrected KDD		
10% KDD	7.6	90.6
Normal only	14.5	91.5
50/50	14.3	91.3

Table 5
Performance of 2nd layer SOM under different training data partitions for case of 6 basic features—category specific detection rates under corrected KDD test data

Partition	Normal	DOS	Probe	U2R	R2L
10% KDD	92.4	96.5	72.8	22.9	11.3
Normal	85.5	96.5	91	22.9	20.5
50/50	85.7	96.7	79.7	30	18.4

Table 6
Attack specific detection rates at the 2nd layer under each training data partition—previously encountered attack types; corrected KDD data set

Attack name (# exemplars)	Category	10% KDD (%)	Normal (%)	50/50 (%)
Back (1098)	Dos	50.9	7.2	52.0
Land (9)	Dos	100.0	100.0	100.0
Neptune (58001)	Dos	96.4	97.4	96.9
Pod (87)	Dos	6.9	60.9	4.6
Smurf (164091)	Dos	99.9	100.0	99.8
Teardrop (12)	Dos	16.7	16.7	25.0
Ipsweep (306)	probe	25.5	83.0	25.2
Nmap (84)	probe	40.5	95.2	85.7
Portsweep (354)	probe	52.0	93.8	84.7
Satan (1633)	probe	73.7	88.0	76.6
ftp_write (3)	R2l	0.0	0.0	33.3
guess_passwd (4367)	R2l	7.9	16.7	13.4
Imap (1)	R2l	0.0	100.0	100.0
Multihop (18)	R2l	0.0	16.7	0.0
Phf (2)	R2l	0.0	50.0	50.0
Snmpguess (2406)	R2l	3.2	5.5	7.5
Warezmaster (1602)	R2l	27.3	34.1	34.5
Worm (2)	R2l	50.0	50.0	50.0
buffer_overflow (22)	U2r	13.6	18.2	22.7
Loadmodule (2)	U2r	0.0	0.0	0.0
Perl (2)	U2r	0.0	0.0	50.0
Rootkit (13)	U2r	69.2	53.8	61.5
Sqlattack (2)	U2r	50.0	50.0	50.0

Table 7
Attack specific detection rates at the 2nd layer under each training data partition—not previously encountered attack types; corrected KDD data set

Attack name (# exemplars)	Category	10% KDD (%)	Normal (%)	50/50 (%)
Apache2 (794)	dos	90.3	29.2	96.0
Mailbomb (5000)	dos	7.8	8.0	8.0
Processtable (759)	dos	59.4	77.2	71.9
Udpstorm (2)	dos	0.0	0.0	0.0
Mscan (1053)	probe	90.2	92.3	91.7
Saint (736)	probe	79.1	97.4	89.1
Httptunnel (158)	r2l	58.9	88.6	71.5
Named (17)	r2l	23.5	41.2	35.3
Sendmail (17)	r2l	5.9	29.4	11.8
Snmpgetattack (7718)	r2l	11.5	23.0	20.1
Xlock (9)	r2l	0.0	11.1	11.1
Xsnoop (4)	r2l	0.0	0.0	0.0
Ps (16)	u2r	0.0	0.0	6.3
Xterm (13)	u2r	23.1	30.8	38.5

It is first noted that the attacks with the largest number of exemplars are consistently identified, independent of training partition e.g. Neptune, Smurf (DoS). However, it is interesting to note that the SOM trained with the complete 10% KDD partition does not necessarily result in the best detector for attack types with hundreds to thousands of exemplars (e.g. Ipsweep, Portsweep or Processtable). Moreover, in the case of the probe attack category the SOM trained as an anomaly detector (normal) consistently performs better than either of the other detectors on both seen or unseen attacks. For R2L the detector trained with 10%KDD is consistently the worst (on either seen or unseen attack types), whereas under U2R the low number of exemplars results in all systems performing equally bad.

5.1.2. *Six basic features, three layer SOM hierarchy*

Partitioning of the training data clearly has an impact on the quality of the detector, where this is not a particularly desirable result, given the complexity of the ensuing combined detector and the uncertainly in a priori selection of the best data partition. In the following, the case of adding a third layer to the SOM hierarchy is considered whilst training on the 50/50 and 10% KDD training data partitions (best overall FP and detection rates on whole KDD and corrected KDD, respectively).

From Fig. 3, it is apparent that nodes 1, 32 and 36 of the second layer SOM (10% KDD training data partition) account for most of the attack connections and neuron 19 most of the normal connections. It is also apparent that several neurons respond to both normal and attack connections. To this end neurons 4, 17, 18, 23, 30 and 36 are selected for association with third-level SOMs, one for each second layer neuron, as per the decision rule of Section 3.3. On completion of training at the third-level maps, a clear separation between normal and attack was achieved. Fig. 4 illustrates this property for the case of neuron 36. In this case, it is apparent that the normal connections all reside in the top left corner of the map, whereas the attack connections populate the remainder.

Table 8 details test set performance for the case of the 6-feature three layer SOM hierarchies on 'corrected (test)' KDD and 'whole KDD' data sets. Moreover, Table 9 details the performance of the three-layer hierarchies on 'corrected (test) KDD' set for different attack categories, whereas the performance on new attacks in 'corrected (test) KDD' is detailed in Table 10. Corresponding results for the two-layer architectures are given in Tables 4, 5 and 7. The common thread through these results is an underlying improvement to the false positive rate at the expense of a decrease in the detection rate, Tables 8 and 4. Specifically, the addition of the third-level results in an improvement or no change to the most frequent exemplar classes—normal and DoS—at the expense of the remaining attack categories and unseen attacks; compare Tables 9 and 5 or 10 and 7. Moreover, the improved results reported by the three-layer architecture on "Whole KDD" relative to those for "Corrected KDD" appear to indicate that the detector is becoming too specific, Tables 8 and 4.

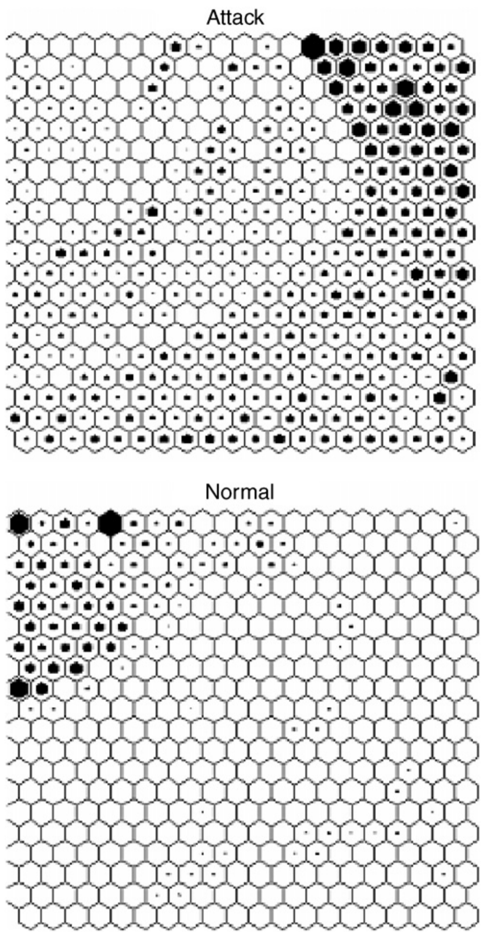

Fig. 4. Third-level map associated neuron 36 of the second-level map—10% KDD training data partition.

Table 8
Comparative test results for 3-layer hierarchy

Partition	FP rate (%)	Detection rate (%)
Whole KDD		
10% KDD	1.75	99.7
50/50	2.6	99.1
Corrected test		
10% KDD	4.6	89.0
50/50	10.9	88.6

Table 9
Category specific detection rates for 3-layer hierarchy—corrected (test) data set

Hierarchy level	Normal	DoS	Probe	U2R	R2L
10% KDD	95.4	95.1	64.3	10.0	9.9
50/50	89.1	93.4	76.7	24.6	20

Table 10
Attack specific detection rates for 3 layer hierarchies—not previously encountered attack types in corrected KDD data set

Attack	Number of exemplars	Category	10% KDD	50/50
apache2	794	Dos	90.7	91.9
Mailbomb	5000	Dos	6.8	7.8
Processtable	759	Dos	47.6	54.7
Udpstorm	2	Dos	0.0	0.0
Mscan	1053	Probe	60.9	81.4
Saint	736	Probe	78.7	86.4
Httptunnel	158	r2l	20.9	25.3
Named	17	r2l	0.0	58.8
Sendmail	17	r2l	11.8	17.6
Snmpgetattack	7741	r2l	10.3	27.7
Xlock	9	r2l	0.0	22.2
Xsnoop	4	r2l	0.0	25
Ps	16	u2r	0.0	18.75
Xterm	13	u2r	30.8	23.1

Table 11
Comparison of 2-layer 5-feature SOM detectors with 2-layer 6-feature SOM on corrected KDD test—10% KDD training data

Excluded feature	FP rate (%)	Detection rate (%)	*T*-test (%)
None (baseline)	7.6	90.6	N/a
Duration	2.3	89	0.02
Protocol	44.1	91.2	4.2
Service	0.9	87.1	0.001
Flag	2.8	88.3	0.003
Source bytes	1.5	88.3	0.0017
Destination bytes	1.4	88.3	0.0017

5.1.3. *Sensitivity analysis for the case of 6 basic features*

The 9 basic features utilized in the KDD competition are derived from the original DARPA *tcpdump* file using the 'Bro' network analyzer (Bro, 1999). Historically, very little has been done to identify the significance or contribution of these basic features to the performance of the ensuing IDSs. In this work, only 6 of the 9 features are employed, as three of the original 9 features are actually invariant over the 10% KDD training data set, Section 2.1. In order to quantify the contribution of each of the remaining six basic features, a 'leave-one-out' methodology is adopted. Specifically, 6 s layer SOMs are trained on the 10% KDD training data, each with a different combination of the five basic features i.e., 5 rather than all 6 layer 1 SOMs are utilized. Performance on the corrected KDD test set is then compared with that from the second layer detector trained with all six basic features.

Table 11 details the false positive and detection rates over the corrected KDD test set for the six versions of the 5-feature system, with the corresponding full (6-feature) detector identified in Section 5.1.3 (all systems trained on the 10% KDD training data partition). It is apparent that either the detection or false positive rates decrease whenever one of the 6 features is removed, indicating that all features contribute to detection. It is interesting to note, however, that only when the protocol feature is dropped does the false positive and detection rate increase. All other features result in a decrease to both false positive and detection rates. In addition, a student paired two-tailed *T*-test is made for the hypothesis that the classification counts for each attack remain the same, Table 11. All are independent from the 6-feature baseline at the 95% confidence interval (99% for all but protocol, where protocol resulted in the least change to detection rate).

5.1.4. *Summary of architectures utilizing 6-features*

When using 6 basic features, the resulting detector is capable of detecting over 95% of normal and DoS connections when using the standard training distribution. Detection of the probe category is better when using the 50/50 partition. This appears to be a natural reflection of the frequency with which connection types are represented in the training data. The SOM algorithm will allocate BMUs in accordance with the frequency with which behaviors appear. Reducing the frequency of the DoS class results in a better detection of probe, at the expense of the larger classes (normal and DoS).

Of the six basic features, protocol appears to be the most significant, where a natural extension might be to filter and associate connections with different detectors on the basis of this parameter. Finally, the three layer SOM appears to improve FP rate at the expense of detection, relative to the two layer SOM architecture.

5.2. *Architectures based on 41 KDD features*

5.2.1. *Forty-one features, single SOM*

In this section, SOM detectors are built when domain knowledge is used to construct relevant features. To this end, a single SOM is built in which the input is the 41 KDD features and compared against the corresponding two layer 6-feature SOM hierarchy. The 41-feature case naturally does not require potential function clustering as the new feature set makes the first layer maps redundant. The same neuron count (36) for the case of the 41-feature single layer SOM is retained.

Table 12 summarizes performance for the 41-feature SOM with a single layer (equivalent to layer 2 of the 6-feature model) on Corrected KDD and Whole KDD test data sets. Table 4 provides the corresponding results under the 6-feature hierarchy. It is immediately apparent that the false positive rates for 41-feature SOM under 10% KDD and 50/50 partitions are much tighter than that under the six-feature architecture.

In order to identify the source for this improvement, performance under different attack categories are compared, Table 13 versus Table 5 for the 6-feature architecture. The most striking difference between the two systems is the stronger performance by the 41-feature architecture on 'Normal'. However, the better performance on Normal (and DoS) is achieved at the expense of the probe category for the architecture trained on the 10% KDD partition. In

Table 12
Performance of single layer SOM under different training data partitions for case of All 41 features—total FP and detection rate

Partition	FP rate (%)	Detection rate (%)
Whole KDD		
10% KDD	0.67	99.69
Normal only	14.45	99.73
50/50	0.29	99.62
Corrected test		
10% KDD	1.53	89.92
Normal only	27.16	94.0
50/50	1.4	90.14

Table 13
Performance of single layer SOM under different training data partitions for case of 41 features—category specific detection rates under corrected test

Partition	Normal	DOS	Probe	U2R	R2L
10% KDD	98.5	96.8	63.4	0	0.15
Normal	72.8	96.6	70.6	11.4	64
50/50	98.6	96.8	75.4	0	1

Table 14
Performance of dual layer SOM under different training data partitions for case of All 41 features—total FP and detection rate

Partition	FP	Detect	Normal	DOS	Probe	U2R	R2L
10% KDD	1.57	90.6	98.4	96.9	67.6	15.7	7.3
50/50	1.38	90.4	98.6	96.9	81.3	0	1.1

addition, the 50/50 partition is now clearly much more competitive (with 10% KDD) than was the case under the 6-feature scheme. In particular, the 50/50 partition now provides the best case FP and detection rates, Table 12, where this appears to be manifest in better detection of the 'probe' attack category, Table 13. This also naturally results in the additional advantage of a faster training time; the 50/50 partition is 61% smaller than the 10% KDD partition.

5.2.2. Forty-one features, two layer SOM hierarchy

As in the case of the three layer 6-feature architecture, additional neuron specific SOMs may be added to the 41-feature case. Table 14 details the corresponding false positive and detection rates after expanding the 6 layer 1 neurons with the highest misclassification rates in the case of the two best training partitions (10% KDD and 50/50). The superiority of the 50/50 partition is emphasized, as results continue to improve whereas those for 10% KDD actually get worse in some cases. This is most probably an artifact of the 10% KDD data set being dominated by DoS attack type. In effect the SOM dedicates most resources to representing the DoS distribution, at the expense of the other connection categories.

The final comparison made is in terms of classification rates on the most frequently seen and unseen attacks for 6 and 41-feature architectures. In this case, the difference between the classification counts for the two architectures is plotted for all attack types with more than 100 exemplars (Tables 6 and 7 summarize the number of exemplars per attack type). Fig. 5 details the case for 2 layer architectures and Fig. 6 the 3 layer architecture under the 50/50 training data partition (similar observations are true for the case of the 10% KDD partition). Positive (negative) differences indicate that the 41-feature architecture has a higher classification count relative to the 6-feature (41-feature) case. It is evident that the 6-feature case tends to provide better performance over a wider range of attacks. However, the 41-feature case performs better on the majority of the higher frequency seen attack types (e.g. Neptume, Smurf and Portsweep), thus resulting in better overall false positive and detection rates.

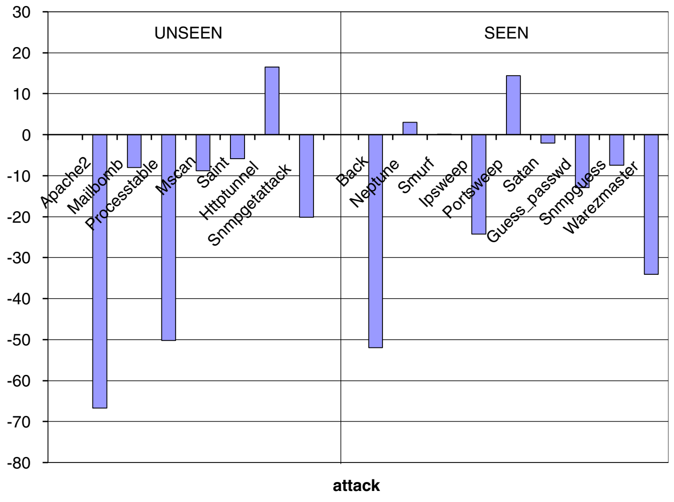

Fig. 5. Percent difference on frequent attack types—2 layer 41-feature versus 6-feature.

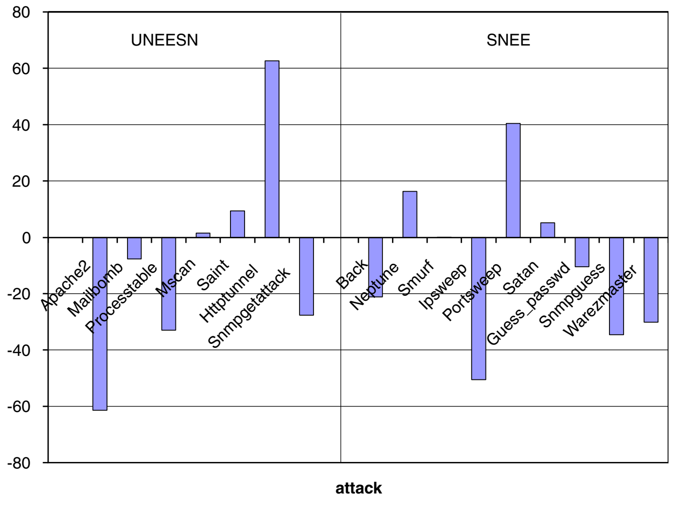

Fig. 6. Percent difference on frequent attack types—3 layer 41-feature versus 6-feature.

Moreover, the 41-feature case benefits more when additional node specific SOMs are built.

5.2.3. Summary of architectures utilizing 41-features

Utilizing all 41-features results in incremental improvements to the detection of Normal and DOS categories, whilst significantly advancing performance under Probe. However, this is only true for the 50/50 partition; training on all training data results in little benefit relative to the performance achieved using 6-features alone. Moreover, a wider range of attack types were still recognized by the 6-feature architecture, with most performance gains in the 41-feature case coming from better detection rates on the more frequently represented classes.

5.3. Previous approaches to the KDD-99 data set

Table 15 provides a summary of some recent results from alternative approaches based on *unsupervised* learning algorithms. In all cases training was conducted using the KDD-99 data set and tested using the 'corrected (test)' data (Eskin et al., 2002), (Lee et al., 1999). Results for the alternative schemes fall over the interval 2–10% for false positives and 70–98% on detection. This is competitive with the 3 layer 6-feature SOM architecture i.e. a false positive rate of 4.6% and a detection rate of 89%. However, there are several additional factors with which these results need to be interpreted. In the case of Eskin et al. (2002), figures quoted are for a mixture of specific and multiple attack types, making it difficult to determine performance over the entire data set. Moreover, use was also made of host-based information, thus providing an advantage when detecting content-based attacks (Eskin et al., 2002). None of the results approach that achieved here using a 2 layer SOM based on all 41-features.

Table 16 details the top two winning entries from the original data mining competition (Elkan, 2000). These are both trained using *supervised* learning and make use of hundreds of decision trees, utilize all 41-features, divide the training data set into multiple partitions (Levin, 2000) (or utilize bagged-boosting Pfahringer, 2000)) and require unique hand crafted optimization criteria. Performance on the most frequently occurring classes are very similar (normal and DoS are within 1% of each other); however, the supervised schemes differentiate themselves on the less frequent classes. The overall effect of this is a false positive rate which is three times smaller than that provided by the 2-layer 41-feature SOM (0.5% versus 1.4%).

Table 15
Recent results on corrected (test) KDD using alternative clustering algorithms

Technique	Detection rate (%)	FP rate (%)
Data-mining (Lee et al., 1999)	70–90	2
Clustering (Esking et al., 2002)	93	10
K-NN (Esking et al., 2002)	91	8
SVM (Esking et al., 2002)	98	10

Table 16
Winning KDD competition supervised classifier results on corrected (TEST) KDD data set

Model	Normal	DOS	Probe	U2R	R2L	Detection	FP
(Pfahringer, 2000)	99.5	97.1	83.3	13.2	8.4	90.9	0.45
(Levin, 2000)	99.4	97.47	84.5	11.54	7.32	91.5	0.58

As indicated in the introduction, several SOM-based anomaly and misuse models have been proposed. Several authors have suggested using a priori knowledge to build SOM models for specific protocol—service combinations (Ramadas et al., 2003; Rhodes et al., 2000), however, both works report results on proprietary data sets making comparison impossible. With regards to alternative SOM architectures applied to the KDD'99 data set, Sarasamma et al. design a three layer SOM hierarchy in which different feature subsets are utilized at each level in the hierarchy (Sarasamma et al., 2005). As in our work, higher layer SOMs are built selectively, depending on the consistency of performance of BMUs at lower layers. The principle difference lies in the use of a priori knowledge to specify feature subsets utilized at each layer of the hierarchy. Such a scheme returns Detection rates in the region of 90.94—93.46% and false positive rates of 2.19–3.99% (depending on the number of neurons used at each SOM layer); as compared to the work reported here in which a two (three) layer SOM hierarchy based on all 41 (6 basic) features returns a detection rate of 90.4% (89%) and false positive rate of 1.38% (4.6%). Hence, using a priori knowledge to select subsets of features over a 3 layer architecture at each layer is not able to improve on the performance of a 2 layer architecture based on all 41 features, however, such a scheme does half the false positive rate of a system limited to the 6 basic features alone.

6. Conclusion

An unsupervised learning approach to the IDS problem is investigated and demonstrated on the International Knowledge Discovery and Data Mining Tools Competition intrusion detection benchmark (Hettich and Bay, 1999). To do so a hierarchical SOM architecture is investigated under two basic feature sets, one is limited to 6 basic features whereas the other contains all '41-features'. The significance of five basic design decisions are considered and commented on as follows:

Training partition—The original partition, denoted 10% KDD, is considered alongside the case of an anomaly detector (10% KDD with all attack exemplars removed) and 50/50, in which the number of attack and normal exemplars is balanced. Given that the original '10% KDD' is highly unbalanced, it is natural that this partition also contains the most exemplars. This both increases training time and is eventually observed to bias the resulting detector towards the two largest categories of Normal and DoS.

Hierarchy—Both the 2-layer 6-feature hierarchy and 1-layer 41-feature hierarchy benefit from the provision of additional "node specific" SOMs. Such a scheme appears to result in the natural division of attack and normal exemplars. However, the 41-feature architecture appears to benefit most, where this naturally indicates that the 41-feature architecture results in a more discriminant feature space than the 36 features derived from the level one SOMs of the 6-feature architecture. Thus, in this case domain knowledge provides most advantage.

Basic feature significance—Of the 6 basic features, protocol is the only feature to result in increases of false positive and detection rates. Dropping any of the other five features resulted in detectors providing lower false positive and detection rates. This appears to indicate that detectors should actually be built with respect to protocol and service (best case false positive rate), where this reflects the a priori knowledge that attacks are (currently) specific to both. Moreover, such a scheme would result in a modular detector architecture that facilitates incremental updates much easier than the current (machine learning) norm of single detectors (Ramadas et al., 2003).

41-feature architecture—The combination of all 41-features with two SOM layers (equivalent to 3-layers of the 6-feature scheme) resulted in the best-case performance. Moreover, a definite preference for the 'balanced' 50/50 training partition was demonstrated. The resulting false positive and detection rates were 1.38% and 90.4%, respectively.

Previous KDD detectors—In comparison to past best case supervised learning solutions, the (unsupervised) solutions identified here have a false positive rate three times higher on the corrected (test) KDD data set (1.38% versus 0.45%), but similar Detection rates (90.4% versus 90.9%). We believe the principle reason for this to be in terms of the availability of suitable 'boosting' algorithms for unsupervised learning, as opposed to the supervised learning methodology itself.

It is anticipated that future work will investigate the utilization of a boosting scheme within the context of a distributed solution to the IDS problem. Moreover, several algorithms for incrementally 'growing' SOM-type architectures have been proposed and demonstrated to be appropriate for data mining type tasks (Rauber et al., 2002). It would therefore be of interest to consider such systems for IDS-type tasks.

Acknowledgments

This work was supported in part through NSERC Discovery, CFI New Opportunities, and IRIS Grants (all Canadian). The first author was also a recipient of a Killam Doctoral Scholarship.

References

Bass, T., 2000. Intrusion detection systems and multisensor data fusion. Communications of the ACM 43 (4), 99–105.

Bro, 1999. Bro user manual 〈http://www.icir.org/vern/bro-manual/index.html〉.

Chiu, S.L., 1994. Fuzzy model identification based on cluster estimation. Journal of Intelligent and Fuzzy Systems 2, 267–278.

DARPA Archive, 1999. Knowledge discovery in databases DARPA archive. Task Description. 〈http://www.kdd.ics.uci.edu/databases/kddcup99/task.html〉.

Elkan, C., 2000. Results of the KDD'99 classifier learning contest SIGKDD explorations. ACM SIGKDD 2 (1), 2–13.

Eskin, E., Arnold, A., Prerau, M., Portnoy, L., Stolfo, S., 2002. A geometric framework for unsupervised anomaly detection: detecting intrusions in unlabeled data. In: Barbara, D., Jajodia, S. (Eds.), Applications of Data Mining in Computer Security, Kluwer, Dordrecht, ISBN 1-4020-7054-3 (Chapter 4).

Kayacik, H.G., Zincir-Heywood, A.N., Heywood., M.I., 2003. On the capability of an SOM based intrusion detection system. IEEE-INNS International Joint Conference on Neural Networks 1808–1813.

Halici, U., Ongun, G., 1996. Fingerprint classification through self-organizing feature maps modified to treat uncertainties. Proceedings of the IEEE 84 (10), 1497–1512.

Hettich, S., Bay, S.D., 1999. The UCI KDD Archive. University of California, Department of Information and Computer Science, Irvine, CA 〈http://kdd.ics.uci.edu〉.

Kohonen, T., 2000. Self-Organizing Maps, third ed. Springer, Berlin, ISBN 3-540-67921-9.

Kohonen, T., Kaski, S., Lagus, K., Salojrvi, J., Honkela, J., Paatero, V., Saarela, A., 2000. Self-organization of a massive document collection. IEEE Transactions on Neural Networks 11 (3), 574–585.

Lee, W., Stolfo, S., Mok, K., 1999. A data mining framework for building intrusion detection models. Proceedings of the 1999 IEEE Symposium on Security and Privacy 120–132.

Levin, I., 2000. KDD-99 Classifier Learning Contest, LLSoft's Reslts Overview, SIGKDD Explorations. ACM SIGKDD 1 (2), 67–75.

Litchodzijewski, P., Zincir-Heywood, A.N., Heywood, M.I., 2002. Host-based intrusion detection using self-organizing maps. IEEE International Joint Conference on Neural Networks 1714–1719.

Luttrell, S.P., 1989. Hierarchical vector quantization. IEE Proceedings I—Communications, Speech and Vision 136(6), 405–413.

McHugh, J., 2001. Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. ACM Transactions on Information and System Security 3 (4), 262–294.

MIT Lincoln Lab, 1998. The 1998 intrusion detection off-line evaluation plan. MIT Lincoln Lab., Information Systems Technology Group. 〈http://www.11.mit.edu/IST/ideval/docs/1998/id98-eval-11〉. txt, 25 March.

Pfahringer, B., 2000. Winning the FDD99 classification cup: bagged-boosting, SIGKDD explorations. ACM SIGKDD 1 (2), 65–66.

Ramadas, M., Ostermann, S., Tjaden, B., 2003. Detecting anomalous network traffic with self-organizing mapps. Recent advances in intrusion detection. Lecture Notes in Computer Science 2820, 36–54.

Rauber, A., Merkl, D., Dittenbach, M., 2002. The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data. IEEE Transactions on Neural Networks 13 (6), 1331–1341.

Rhodes, B., Mahaffey, J., Cannady, J., 2000. Multiple self-organizing maps for intrusion detection. In: Proceedings of the 23 National Information Systems Security Conference.

Sackinger, E., Boser, B.E., Bromle, J., Lecun, Y., Jackel, L.D., 1992. Application of the ANNA neural network chip to high-speed character recognition. IEEE Transactions on Neural Networks 3, 498–505.

Sarasamma, S.T., Zhu, Q.A., Huff, J., 2005. Hiearchical Kohonenen net for anomaly detection in network security. IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics 35 (2), 302–312.

SOM-PAC. Software Packages from Helsinki University of Technology. Laboratory of Computer and Information Science Neural Networks Research Centre 〈http://www.cis.hut.fi/research/software.shtml〉.