Attack Characterization and Intrusion Detection using an Ensemble of Self-Organizing Maps

Lori L. DeLooze, Member, IEEE

Abstract—Self-Organized Maps (SOM) use an unsupervised learning technique to independently organize a set of input patterns into various classes. In this paper, we use an ensemble of SOMs to identify computer attacks and characterize them appropriately using the major classes of computer attacks (Denial of Service, Probe, User-to-Root and Remote-to-Local). The procedure produces a set of confidence levels for each connection as a way to describe the connection’s behavior.

I. INTRODUCTION

As computer technology evolves and the threat of computer crimes increases, the apprehension and preemption of such violations become more and more difficult and challenging. Most system security mechanisms are designed to prevent unauthorized access to system resources and data. To date, it appears that completely preventing breaches of security is unrealistic. Therefore, we must try to detect these intrusions as they occur so that actions may be taken to repair the damage and prevent further harm. Over the years, intrusion detection has become a major area of research in computer science and various innovative methods have been applied to these systems.

II. INTRUSION DETECTION SYSTEMS

Intrusion Detection Systems (IDS) are designed to monitor network traffic to determine if an intrusion has occurred. The two basic methods of detection are signature based and anomaly based [1]. The signature-based method, also known as misuse detection, looks for a specific signature to match, signaling an intrusion. Network traffic is scanned as it passes by for specific features that might indicate an attack or an intrusion. This means that these systems are not unlike virus detection systems -- they can detect many or all known attack patterns, but they are of little use for as yet unknown attack methods. Most popular intrusion detection systems fall into this category. A misuse-based IDS uses a database of traffic or activity patterns related to known attacks to identify and categorize malicious activity on the network.

Another approach to intrusion detection is called anomaly detection. Anomaly-based systems attempt to map events to the point where they “learn” what is normal and then detect an anomaly that might indicate an intrusion. Anomaly detection techniques assume that all intrusive activities are necessarily anomalous. This means that if we could establish a normal activity profile for a system, we should, in theory, flag all system states that vary from the established profile by statistically significant amounts as intrusion attempts. The main issue in anomaly detection systems thus becomes the selection of threshold levels so that the system does not flag anomalous activities that are non-intrusive nor fail to flag intrusive activities that are not anomalous. Anomaly detection systems are computationally expensive because of the overhead of keeping track of, and possibly updating, several system profile metrics.

A. Evaluation of intrusion detection systems

In 1998, the Defense Advanced Research Projects Agency (DARPA) intrusion detection evaluation created the first standard corpus for evaluating intrusion detection systems. The 1998 off-line intrusion detection evaluation was the first in a planned series of annual evaluations conducted by the Massachusetts Institute of Technology (MIT) Lincoln Laboratories under DARPA sponsorship. The corpus was designed to evaluate both false alarm rates and detection rates of intrusion detection systems using many types of both known and new attacks embedded in a large amount of normal background traffic [2]. Detection rate is computed as the ratio between the number of correctly detected attacks and the number of attacks, while the false alarm rate is computed as the ratio between the number of normal connections that are incorrectly misclassified as attacks (false alarms) and the total number of connections. Over 300 attacks were included in the 9 weeks of data collected for the evaluation. These 300 attacks were drawn from 32 different attack types and 7 different attack scenarios.

The attacks developed for the evaluation were developed to provide a reasonable amount of variance in attack methods. Some attacks occurred in a single session with all actions occurring in the clear, while others were spread out over several different sessions and clearly employed methods to evade detection. The attack scenarios also included diversity in the intent of the exploitation. Some attacks were just for fun while others were for the expressed purpose of collecting confidential information or causing damage. The corpus was collected from a simulation network that was used to automatically generate realistic traffic – including the attacks cited above. Training data was labeled with attacks and provided to participants to train and tune their intrusion detection systems [3]. Unlabeled test data was later provided for blind evaluation.

Initial observations of the evaluation results for the 1998 competition concluded that most IDSs can easily identify older, known attacks with a low false-alarm rate, but do not perform as well when identifying novel or new attacks [4]. The best system detected about 75% of the attacks in the test...
data with fewer than two false alarms per day. This misuse intrusion detection system used extensive raw packet data and hand-created attack signatures generated using the training data. It, however, missed many of the new attacks. The next-best system, which used anomaly detection based on data mining, was able to detect 64% of the attacks with 20 false alarms per day. Under these same circumstances, the rule-based system detected about 45% of the attacks with a false alarm rate of 46 false alarms per day. These results suggest that either the rule-based system or anomaly-based data mining system can provide good performance on previously seen attacks, but neither approach, however, is capable of detecting new attacks with high accuracy.

Due to the relative success of the Data Mining approach during the DARPA Intrusion Detection System Evaluation and the significant challenge of identifying new attacks, the organizational committee for the 1999 Knowledge Discovery and Data Mining (KDD) Conference suggested an intrusion detection problem augment the existing 1999 KDD learning challenge. The KDD competition aims at showcasing the best methods for discovering higher level knowledge from data and closing the gap between research and industry, thereby stimulating further KDD research and development. The KDD’99 Cup competition used a subset of the preprocessed DARPA training and test data supplied by Professors Sal Solvo and Wenke Lee [5], the principal researchers for the Data Mining entry to the DARPA evaluation.

The raw training data was about four gigabytes of compressed binary tcpdump data from seven weeks of network traffic. This was processed into about five million connection records. Similarly, the two weeks of test data yielded around two million connection records. Scoring focused on the systems ability to detect novel attacks in the test data that was a variant of a known attack labeled in the training data. The KDD ’99 training datasets contained a total of 24 training attack types, with an additional 14 attack types in the test data only.

Participants were given a list of high-level features that could be used to distinguish normal connections from attacks. A connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows to and from a source IP address to a target IP address under some well defined protocol. Each connection is labeled as either normal, or as an attack, with exactly one specific attack type.

B. Unsupervised Learning

The Kohonen Map [6], or Self-Organizing Map (SOM) uses an unsupervised learning technique to independently organize a set of input patterns into various classes. The model generally consists of a two-dimensional neuron configuration (map), as shown in figure 1, though topologies of higher dimensions are also conceivable.

Each neuron is assigned a representative set of M features, called a vector. In the course of the training process, the feature weights corresponding to each neuron are adapted in accordance with the input signal and their positions are shifted in the input space in the direction of the input vector. As a result of the algorithm, an organized network develops where similar input patterns is transformed into a degree of proximity between the locations of excited neurons. The neurons are arranged in accordance with the input patterns by means of neighborhoods. That is, the neurons not are adapted individually, but in conjunction with neighboring neurons.

A SOM does not need a target output to be specified unlike many other types of network. Instead, where the node weights match the input vector, that area of the lattice is selectively optimized to more closely resemble the data for the class the input vector is a member of. From an initial distribution of random weights, and over many iterations, the SOM eventually settles into a map of stable zones. Each zone is effectively a feature classifier, so you can think of the graphical output as a type of feature map of the input space. Any new, previously unseen input vectors presented to the network will stimulate nodes in the zone with similar weight vectors.

Several variations of the Kohonen algorithm exist. The algorithm used for the SOMs in this research is as follows:

1) Each node's weights is initialized.
2) A vector is chosen at random from the set of training data and presented to the lattice.
3) Every node is examined to calculate which one's weights are most like the input vector. The winning node is commonly known as the Best Matching Unit (BMU).
4) The radius of the neighborhood of the BMU is now calculated. This is a value that starts large, typically set to the ‘radius’ of the lattice, but diminishes each time-step. Any nodes found within this radius are deemed to be inside the BMU's neighborhood.
5) Each neighboring node's (the nodes found in step 4) weights are adjusted to make them more like the input vector. The closer a node is to the BMU, the more its weights get altered.
6) Repeat step 2 for N iterations.
To determine the best matching unit, one method is to iterate through all the nodes and calculate the Euclidean distance between each node’s weight vector, \( W_i \), and the current input vector, \( V_i \). The node with a weight vector closest to the input vector is tagged as the BMU.

The Euclidean distance is given as:

\[
\text{dist} = \sqrt{\sum_{i=0}^{\text{gen}} (V_i - W_i)^2} \tag{1}
\]

A unique feature of the Kohonen learning algorithm is that the area of the neighborhood shrinks over time. This is accomplished by making the radius of the neighborhood shrink over time. To do this, we use the exponential decay function:

\[
\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\lambda}\right) \quad t = 1,2,3 \ldots \tag{2}
\]

where the Greek letter sigma, \( \sigma_0 \), denotes the width of the lattice at time \( t = 0 \), and the Greek letter lambda, \( \lambda \), denotes a time constant and \( t \) is the current time-step (iteration of the loop). Over time the neighborhood will shrink to the size of just one node - the BMU.

After determining the radius, we iterate through all the nodes in the lattice to determine if they lie within the radius and adjust the weights accordingly. Every node within the BMU’s neighborhood (including the BMU) has its weight vector adjusted according to the following equation:

\[
W(t+1) = W(t) + \Theta(t)L(t)(V(t) - W(t)) \tag{3}
\]

where \( t \) represents the time-step and \( L \) is a small variable called the learning rate, which decreases with time.

The decay of the learning rate is calculated each iteration using the following equation:

\[
L(t) = L_0 \exp\left(-\frac{t}{\lambda}\right) \quad t = 1,2,3 \ldots \tag{4}
\]

The learning rate at the start of training is set to some constant and then gradually decays over time so that during the last few iterations it is close to zero. The effect of the learning should decrease proportionally according to the distance of the node from the BMU. In fact, the edges of the BMU’s neighborhood should have barely any effect at all. Ideally, the amount of learning should fade over distance according to the Gaussian decay shown below.

To achieve this, all it takes is a slight adjustment to the equation above.

\[
W(t+1) = W(t) + \Theta(t)L(t)(V(t) - W(t)) \tag{5}
\]

where \( \Theta(t) \) to represents the amount of influence a node’s distance from the BMU has on its learning. \( \Theta(t) \) is given by

\[
\Theta(t) = \exp\left(-\frac{\text{dist}^2}{2\sigma^2(t)}\right) \quad t = 1,2,3 \ldots \tag{6}
\]

where \( \text{dist} \) is the distance a node is from the BMU and \( \sigma \) is the width of the neighborhood function. Note that \( \Theta \) also decays over time.

### III. ENSEMBLE OF SOMS

Self-Organizing Maps have been applied to the domain of Intrusion Detection as a postmortem, or off-line analysis tool. The computational complexity of unsupervised learning algorithms scale up steeply with the size of the considered data. Other attempts at applying SOMs have limited the problem to only certain attack types [7], to only host-based data [8] or to user-generated proprietary data [9].

We have created a generalized, multi-SOM approach that used existing pre-processed data from the KDD ’99 dataset.

We initially followed the feature selection technique proposed by the Minnesota Intrusion Detection System, MINDS [10], whereby features are extracted based on content, time and connection. Content features include number of total packets, acknowledgement packets, data bytes, retransmitted packets, pushed packets, SYN and FIN packets flowing to or from the source and destination. We will also tracked the status (ie. completed, not completed or reset) for each connection. Time-based features included the number of connections and type of services to or from the source and destination within the last 5 seconds. Because this time-based approach will not be able to detect slow and stealthy attacks, we will also used extracted connection-based features, such as the duration of the connection, the service requested and protocol used.

Giorgio Giacinto and Fabio Roli from the University of Cagliari [11] have shown the effectiveness of this ensemble approach to intrusion detection using multi-layer perceptrons. They first created an overall classifier that used three fully-connected multi-layer perceptrons with three layers of neurons. Each network trained using distinct feature representations: 4 intrinsic features, 7 content features, and 19 traffic features and 30 overall features.

During the same month, Srinivas Mukkamala and Andrew H. Sung of New Mexico Tech [12] presented their research on creating an ensemble of Support Vector Machines (SVMs) that can be used for Intrusion Detection. Their goal was to determine which of the 41 features listed above were important, unimportant or in-between (ie. secondary features). The authors stated that they used only a subset of the available data and their paper implies that they used only those attacks detected by a Support Vector Machine using all
41 features. Because their research examined the impact of removing each of them from the set, this was sound for their goals of determining the importance of each feature. This approach, however, ignores the effect of a combination of features on the classification problem.

We initially explored the possibility of using an ensemble of three SOMs for attack identification. Our training set consisted of 503 normal connections and 24 attack connections extracted from the available data set. The test set consisted of the entire 148,404 connections in the test set. Of these, 93,290 connections were known attacks and had associated training records in the training set. The KDD '99 test set also contained 18,336 unknown attack connections. These were brand new attack types that were not related to any attacks in the training set. Each feature set used a different combination of features to describe the attack.

Each respective feature set creates a representative vector for one of the three SOMs. In other words, one SOM will try to find anomalous behavior based on content features (Figure 2), another based on time features, and the third based on connection features. We also created a SOM trained on all features in the three independent feature sets.

The extraction of suitable features representing network connections is based on expert knowledge about the characteristics that distinguish attacks from normal connections. Different attacks will be more visible under different feature sets. Each feature space is used independently to perform attack detection. Then the evidences are combined in order to produce the final decision. This process reflects the human analyst perspective that usually looks at different traffic statistics in order to produce reliable attack signatures. In addition, the generalization capabilities of pattern recognition algorithms allow for the detection of novel attacks that are not provided by rule-based signatures.

Each vector was mapped onto a Self-Organizing Map where the most similar attacks were associated with adjacent or proximal neurons. Figure 3 shows the resulting SOM for the connection feature set. Note that all attacks are not clustered together, but are grouped into subgroups of neurons that correspond to common feature sets. Normal neurons are interspersed among the sets of attack neurons. This shows how similar an attack can be to a normal connection.

We separated out the results by known attacks versus unknown attacks. The original KDD 1999 competition [5] stated that most systems performed very well on known attacks, but varied greatly with unknown attacks. The detection rate is computed as the ratio between the number of correctly detected attacks and the number of attacks, while the false alarm rate is computed as the ratio between the number of normal connections that are incorrectly misclassified as attacks (false alarms) and the total number of normal connections.

Table I reports the performances of the four SOMs on known attacks included in the training set and unknown attacks, ie the attack samples related to attack types not included in the training set. Detection rates for the Connection and Time feature sets were relatively successful for known attacks and somewhat successful for unknown attacks. Surprisingly, the detection rates for time features was higher than that for the SOM trained on all features. This confirms that, even in isolation, SOMs trained on distinct feature sets may be better at classification than a SOM trained on all features together.

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>Detection Rate</th>
<th>False Alarms</th>
<th>Known Attacks</th>
<th>Unknown Attacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>73.34%</td>
<td>0.033</td>
<td>77.96%</td>
<td>31.10%</td>
</tr>
<tr>
<td>Content</td>
<td>0.02%</td>
<td>0.0</td>
<td>0.01%</td>
<td>0.06%</td>
</tr>
<tr>
<td>Time</td>
<td>87.25%</td>
<td>0.009</td>
<td>87.60%</td>
<td>23.87%</td>
</tr>
<tr>
<td>All Features</td>
<td>81.85%</td>
<td>0.003</td>
<td>80.20%</td>
<td>12.56%</td>
</tr>
</tbody>
</table>

Once training is conducted and the resulting SOM is created, classification is quick. The process takes less than 30 seconds for each classification run. As with most intrusion detection systems, an ensemble SOM would probably have to be used postmortem, or off-line as an analysis tool. The speed and ease of classification, however, can support its use in a near real-time system.

Table II reports the overall detection rates when the results of the individual SOMs are combined. The Majority Ensemble reports an attack if two of the three SOMs report an attack on a particular connection. Due to the very poor performance of the content SOM, we can assume that these were attacks that were reported in both the Connection SOM and the Time SOM. The Belief Ensemble reports an attack if any of the three SOMs reports an attack for a particular connection. The results for the belief SOM are excellent.

![Figure 3. Self-Organizing Map for Connection Features](image-url)
because the connections that fail detection using Connection Features will likely be detected using Time Features, and vice versa.

### IV. FEATURE SET COMPOSITION

Based on the performance superiority of the selective time-based feature set over the comprehensive combined feature set, we designed an experiment to find a unique set of features that can best detect each type of attack family in the test data. The test data contains attacks that fall into four main categories [13]:

**Denial of Service (DOS):** A denial of service attack is an attack in which the attacker makes some computing or memory resource too busy or too full to handle legitimate requests, or denies legitimate users access to a machine. (99,275 attacks in test data)

**Probe:** Network probes map the machines and services that are available on a network and can be used to locate weak points. (3,736 attacks in test data)

**Remote-to-Local (R2L):** A Remote to User attack occurs when an attacker who has the ability to send packets to a machine over a network, but who does not have an account on that machine, exploits some vulnerability to gain local access as a user of that machine. (62 attacks in test data)

**User-to-Root (U2R):** User to Root attacks occur when an attacker, who starts out with access to a normal user account on the system (which may have been gained by a previous attack), is able to exploit some vulnerability to gain root access to the system. (32 attacks in test data)

Because each type of attack family has a different attack signature, we expect that each has a unique feature set that is best suited for classifying attacks of that type. Using a brute force method would be time prohibitive, so we used a genetic algorithm to find the best possible combination of features to classify each attack class.

For example, one of the top results for the first generation DOS SOM was combination 5, which consisted of the feature set 3, 4, 6, 8, 9, 11, 12, 14, 16, 19, 20, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34, 36, 37, 38, 40, 41. This combination of features, along with other high-performing chromosomes, were manipulated to create the next generation. After five generations for each attack family, we determined the best feature set for each and calculated the detection rates and corresponding false detection rates for each resulting SOM.

This is our first attempt at classifying each attack by family. While the rather low detection rates of 71.88% for U2R attacks and 43.54% for R2L attacks may seem disappointing, it is important to compare them against the initial detection rates from the KDD ’99 IDS evaluation [14]. Note that only two of the four groups even attempted to detect U2R attacks and only three of the four attempted to detect R2L attacks.

Table 1: Results for Initial Attack SOMs

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>Detection Rate</th>
<th>False Alarms</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOS</td>
<td>97.59%</td>
<td>0.019</td>
</tr>
<tr>
<td>Probe</td>
<td>89.11%</td>
<td>0.025</td>
</tr>
<tr>
<td>R2L</td>
<td>43.54%</td>
<td>0.025</td>
</tr>
<tr>
<td>U2R</td>
<td>71.88%</td>
<td>0.010</td>
</tr>
</tbody>
</table>

Table 2: Results for Related SOMs

<table>
<thead>
<tr>
<th>Combination</th>
<th>Technique</th>
<th>Detection Rate</th>
<th>False Alarms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Majority</td>
<td>Ensemble</td>
<td>63.02%</td>
<td>0.003</td>
</tr>
<tr>
<td>Belief</td>
<td>Ensemble</td>
<td>87.29%</td>
<td>0.013</td>
</tr>
</tbody>
</table>

Intrusion Detection Systems are evaluated using a graph called a Receiver Operating Characteristic curve (or ROC curve.) It is a plot of the true positive rate against the false positive rate for the different possible cutpoints of a diagnostic test.

Figure 4. ROC Curve for Initial DOS SOM

Figure 5. ROC Curve for Initial Probe SOM

Figure 6. ROC Curve for Initial R2L SOM
Using the specific feature sets, we were able to significantly improve the detection rate. The overall detection rate increased from 91.01% to 94.21% for all attacks, from 38.46% to 79.51% for unknown attacks and from 96.81% to 99.50% for known attacks while also providing the additional information about the attack type.

V. EXPERIMENT AND RESULTS

We propose an improvement to this process to further characterize the connection by using the unique Self-Organizing Map property of adjusting neurons throughout the training process to create an organized network, i.e. signal similarity of the input patterns is transformed into a degree of proximity between locations of excited neurons. Using this property, we should be able to describe the degree of “attackness” of a connection based on its proximity to attack neurons. We should also be able to create a profile of the attack based on the position of the connection in each one of the four SOMs.

The original SOM for each attack type was used simply to identify if an attack was detected. In an attempt to classify these attacks by type, we relabelled the neurons to better identify attacks of that type. Only the labels were changed - no additional training or modifications of any other kind were made.

Neurons in close proximity to the attack neurons should be treated differently than those further away. In Figure 10, for example, neurons 1, 2 and 5 are labelled as one class and 3 and 4 are labelled as another, yet 2 and 5 that are close neighbours of the 3 and 4 and, therefore, should be treated differently than neuron 1 which is a more distant neighbour.

The combination of confidence levels for each connection in each of the four SOMs (DOS, Probe, U2R, R2L) should give an analyst enough information to identify the attack and, therefore, aid in its mitigation. Each connection will be given a confidence level in each of the four areas. A completely normal connection will have a confidence level of 0.0 for each type of attack, while a DOS attack will have a 1.0 confidence level for DOS and other, perhaps non-zero, confidence levels for the other types of attacks. This is due to the mapping of the connection in each of the SOMs.

<table>
<thead>
<tr>
<th>Attack Type</th>
<th>Detection Rate</th>
<th>False Alarms</th>
<th>Known Attacks</th>
<th>Unknown Attacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOS</td>
<td>98.53%</td>
<td>0.008</td>
<td>99.90%</td>
<td>52.69%</td>
</tr>
<tr>
<td>Probe</td>
<td>98.15%</td>
<td>0.028</td>
<td>95.97%</td>
<td>99.00%</td>
</tr>
<tr>
<td>R2L</td>
<td>82.26%</td>
<td>0.026</td>
<td>72.22%</td>
<td>86.6%</td>
</tr>
<tr>
<td>U2R</td>
<td>79.31%</td>
<td>0.010</td>
<td>100.00%</td>
<td>81.25%</td>
</tr>
<tr>
<td>Overall</td>
<td>97.31%</td>
<td>0.042</td>
<td>99.90%</td>
<td>69.95%</td>
</tr>
</tbody>
</table>
This further refinement enables much better detection rates and a significantly lowers the false alarm rate. It is interesting to note the considerable differences in the performance rates for known and unknown attacks. In fact, both the Probe and Root-to-Local SOMs were better able to classify unknown attacks than known attacks.

The ROC curves for the newly relabelled SOMs are presented below. The ensemble of SOMs is able to detect more than any of the other systems except the Data Mining approach for the Probe attacks. The Probe SOM has several nodes that are in the attack zone that could have been designated as Probe attacks and therefore improved the detection rate, but the associated normal connections would then have been misclassified and increased the false alarm rate.

Table V indicates that almost all the attacks were mapped within the proximity of the attack neurons. Those that are outside of the "attack zone" would be characterized as completely normal (i.e. confidence level of 0.0 that it is associated with an attack). By creating a buffer area between the attack neurons and normal neurons, we are able to identify those attacks that are more normal-like and those normal connections that are more attack-like. While these would still be a concern to an analyst, they could be given less attention.

Although the newly labeled SOMs we were able to significantly improve the detection rate and reduce the false alarm rate, the real benefit is the ability to characterize a connection based on the confidence levels contributed by each SOM. Several representative connections are shown in the table below. Note that normal connections can still have a degree of confidence of attack greater than 0 and attacks can have confidence levels for multiple attack types. The additional information provided to the analyst enables them to more quickly begin mitigation actions. Normal connections with zero confidence levels would be completely ignored by the analyst while normal connections with some indication of an anomaly would be of more concern. Because the anomalies are associated with a particular type of attack, the connection is characterized
according to its behavior. A normal connection with some degree of behavior similar to a DOS attack may be of less concern that a normal connection with some degree of behavior similar to a User-to-Root attack.

Similarly, the association with a type of attack may be able to give the analyst some additional information about the source of the attack. The last two attacks in Table VI would be correctly identified as a User-to-Root attack. Note that the first of the two attacks has no behavior in common with Remote-to-Local attack. The analyst can assume, based on confidence levels alone, that the attack originated from the local network.

<table>
<thead>
<tr>
<th>TABLE VI</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONNECTION</td>
</tr>
<tr>
<td>Type</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>Normal</td>
</tr>
<tr>
<td>Normal</td>
</tr>
<tr>
<td>Normal</td>
</tr>
<tr>
<td>Normal</td>
</tr>
<tr>
<td>Probe</td>
</tr>
<tr>
<td>Probe</td>
</tr>
<tr>
<td>R2L</td>
</tr>
<tr>
<td>U2R</td>
</tr>
<tr>
<td>U2R</td>
</tr>
</tbody>
</table>

* False Alarm

It is not as important to classify the connection by type as it is characterize it appropriately according to its behavior. The ensemble of SOMs allows our system to provide additional valuable information to the analyst so he can perform his job more effectively.

VI. CONCLUSION

The reported results have shown the effectiveness of ensemble learning techniques for anomaly detection. This research has shown that an ensemble detection scheme using several Self-Organizing Maps using distinct feature sets for each attack type can effectively characterize a connection. The profile of the connection, in turn, can aid an analyst in the mitigation of the attack.

Future research will modify the configuration of the SOMs to increase the granularity of the nodes, from 20x20 to 30x30, and allow the analyst to vary the desired value for the Gaussian Decay around the BMU. In this way, the resulting characterization of attacks can more closely isolate “attack nodes” while simultaneously reducing the false alarm rate.

REFERENCES