Intelligent Adaptive Intrusion Detection Systems Using Neural Networks
(Comparative study)

1Aida O. Ali, 2Ahmed I. saleh and 3Tamer R. Badawy
1,2 Dep. of Computer and system engineering Faculty of engineering Mansoura University Al Mansoura, Egypt
3Dep. of Computer and system engineering Misr Engineering and Technology Institute Al Mansoura, Egypt
tam22004@hotmail.com

Abstract— Intrusion Detection Systems (IDSs) provide an important layer of security for computer systems and networks, and are becoming more and more necessary as reliance on Internet services increases and systems with sensitive data are more commonly open to Internet access. An IDS’s responsibility is to detect suspicious or unacceptable system and network activity and to alert a systems administrator to this activity. Classification algorithms are used to discriminate between normal and different types of attacks. In this paper, a comparative study between the performances of recent nine artificial neural networks (ANNs) based classifiers is evaluated, based on a selected set of features. The results showed that; the Multilayer perceptrons (MLPS) based classifier provides the best results; about 99.63% true positive attacks are detected.

Index Terms— component; Intrusion detection system; artificial neural networks; Multilayer perceptrons.

I. INTRODUCTION

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions defined as attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a computer or a network.

Given the level and nature of modern network security threats, the question for security professionals should not be whether to use intrusion detection, but which intrusion detection features and capabilities to use.

IDSs have gained acceptance as a necessary addition to every organization’s security infrastructure despite the documented contributions intrusion detection technologies make to system security, in many organizations one must still justify the acquisition of IDSs.

We may use IDSs to prevent problem behaviors by increasing the perceived risk of discovery of those who would attack or abuse the system.

There are two general categories of attacks which intrusion detection technologies attempt to identify - anomaly detection and misuse detection. Anomaly detection identifies activities that vary from established patterns for users, or groups of users. Anomaly detection typically involves the creation of knowledge bases that contain the profiles of the monitored activities.

The second general approach to intrusion detection is misuse detection. This technique involves the comparison of a user's activities with the known behaviors of attackers attempting to penetrate a system. While anomaly detection typically utilizes threshold monitoring to indicate when a certain established metric has been reached, misuse detection techniques frequently utilize a rule-based approach. When applied to misuse detection, the rules become scenarios for network attacks. The intrusion detection mechanism identifies a potential attack if a user's activities are found to be consistent with the established rules. The use of comprehensive rules is critical in the application of expert systems for intrusion detection.

The paper is structured as follows. Section 2 we discuss related work. Section 3 is the use of neural networks in intrusion detection. Section 4 introduces our proposed framework. Section 5 introduces the nine algorithms of neural network and their results. The paper is ended with a conclusion.

II. RELATED WORK

Most current approaches to the process of detecting intrusions utilize some form of rule-based analysis. Rule-Based analysis relies on sets of predefined rules that are provided by an administrator, automatically created by the system, or both. Expert systems are the most common form of rule-based intrusion detection approaches. The early intrusion detection research efforts realized the inefficiency of any approach that required a manual review of a system audit trail. While the information necessary to identify attacks was believed to be present within the voluminous audit data, an effective review of the material required the use of an automated system. The use of expert system techniques in intrusion detection mechanisms was a significant milestone in the development of effective and practical detection-based information security systems.

An expert system consists of a set of rules that encode the knowledge of a human “expert”. These rules are used by the system to make conclusions about the security-related data from the intrusion detection system. Expert systems permit the incorporation of an extensive amount of human experience into a computer application that then utilizes that knowledge to identify activities that match the defined characteristics of misuse and attack.

Unfortunately, expert systems require frequent updates to remain current. While expert systems offer an enhanced
ability to review audit data, the required updates may be ignored or performed infrequently by the administrator. At a minimum, this leads to an expert system with reduced capabilities. At worst, this lack of maintenance will degrade the security of the entire system by causing the system's users to be misled into believing that the system is secure, even as one of the key components becomes increasingly ineffective over time.

Rule-based systems suffer from an inability to detect attacks scenarios that may occur over an extended period of time. While the individual instances of suspicious activity may be detected by the system, they may not be reported if they appear to occur in isolation. Intrusion scenarios in which multiple attackers operate in concert are also difficult for these methods to detect because they do not focus on the state transitions in an attack, but instead concentrate on the occurrence of individual elements. Any division of an attack either over time or among several seemingly unrelated attackers is difficult for these methods to detect.

Rule-based systems also lack flexibility in the rule-to-audit record representation. Slight variations in an attack sequence can affect the activity-rule comparison to a degree that the intrusion is not detected by the intrusion detection mechanism. While increasing the level of abstraction of the rule-base does provide a partial solution to this weakness, it also reduces the granularity of the intrusion detection device.

An artificial neural network consists of a collection of processing elements that are highly interconnected and transform a set of inputs to a set of desired outputs. The result of the transformation is determined by the characteristics of the elements and the weights associated with the interconnections among them. By modifying the connections between the nodes the network is able to adapt to the desired outputs.

Unlike expert systems, which can provide the user with a definitive answer if the characteristics which are reviewed exactly match those which have been coded in the rule base, a neural network conducts an analysis of the information and provides a probability estimate that the data matches the characteristics which it has been trained to recognize. While the probability of a match determined by a neural network can be 100%, the accuracy of its decisions relies totally on the experience the system gains in analyzing examples of the stated problem.

The neural network gains the experience initially by training the system to correctly identify pre-selected examples of the problem. The response of the neural network is reviewed and the configuration of the system is refined until the neural network's analysis of the training data reaches a satisfactory level. In addition to the initial training period, the neural network also gains experience over time as it conducts analyses on data related to the problem.

III. NEURAL NETWORK INTRUSION DETECTION SYSTEMS

A limited amount of research has been conducted on the application of neural networks to detecting computer intrusions. Artificial neural networks offer the potential to resolve a number of the problems encountered by the other current approaches to intrusion detection. Artificial neural networks have been proposed as alternatives to the statistical analysis component of anomaly detection systems. Statistical Analysis involves statistical comparison of current events to a predetermined set of baseline criteria. The technique is most often employed in the detection of deviations from typical behavior and determination of the similarity of events to those which are indicative of an attack. Neural networks were specifically proposed to identify the typical characteristics of system users and identify statistically significant variations from the user's established behavior.

Artificial neural networks have also been proposed for use in the detection of computer viruses.

Neural networks were proposed as statistical analysis approaches in the detection of viruses and malicious software in computer networks. The neural network architecture may be a self-organizing feature map which uses a single layer of neurons to represent knowledge from a particular domain in the form of a geometrically organized feature map. The proposed network was designed to learn the characteristics of normal system activity and identify statistical variations from the norm that may be an indication of a virus.

While there is an increasing need for a system capable of accurately identifying instances of misuse on a network there is currently no applied alternative to rule-based intrusion detection systems. This method has been demonstrated to be relatively effective if the exact characteristics of the attack are known. However, network intrusions are constantly changing because of individual approaches taken by the attackers and regular changes in the software and hardware of the targeted systems. Because of the infinite variety of attacks and attackers even a dedicated effort to constantly update the rule base of an expert system can never hope to accurately identify the variety of intrusions.

The constantly changing nature of network attacks requires a flexible defensive system that is capable of analyzing the enormous amount of network traffic in a manner which is less structured than rule-based systems. A neural network-based misuse detection system could potentially address many of the problems that are found in rule-based systems.

The aim of this work is to establish a framework that can detect the known and the unknown events of attacks and to choose the best algorithm between nine algorithms which provides minimum errors.

IV. PROPOSED FRAMEWORK

Fig. 1, illustrates the proposed framework. The proposed framework is described in terms of four phases; the first phase is the network sensor in this phase we analyze the input packets to obtain the packet parameters and then filtering these parameters to obtain the needed parameters for intrusion detection, the second phase is the event manager which processes the filtered parameters and then compare these parameters with known attacks for determining attacks signatures and also compare these parameters with normal events then we go to the third phase which is the response manager which respond to the attack and normal events in a suitable manner. The fourth phase is the learning model in this
phase we use a mixed database of normal and attack events then sending these events to learning model of neural network. After obtaining a learning module, the unlabeled events could be classified as a normal or attack events.

Fig. 1. Proposed framework for the learning algorithm

Applying this framework we can obtain a database of normal and attack events then we can use this database for applying our algorithms, we have a database of 145587 of normal and attack events we use this database for the comparative study.

After randomizing the events database, 70% of these events are used for the training phase and the other 30% for the test phase. An institutional goal of classification is to discriminate between normal and attack events, while a more ambitious goal may be to classify different attack types. There is a large number of ANNs based classifiers. The performance of nine of them will be evaluated and assessed.

We use four different measures to evaluate the performance of the artificial neural network based classification techniques: (i) mean-square-error (MSE); (ii) normalized mean-square-error (NMSE); (iii) correlation coefficient (r), and (iv) error percentage (%error). The mean squared error of an individual case (i) is evaluated by the equation:

\[
MSE = \frac{1}{n} \sum_{j=1}^{n} (P_{ij} - T_j)^2
\]

where \(P_{ij}\) is the value predicted by the individual case i for fitness case j (out of n fitness cases or sample cases); and \(T_j\) is the target value for fitness case j.

The normalized mean square error is an estimator of the overall deviations between predicted and measured values. It is defined as:

\[
MSE = \frac{1}{n} \sum_{j=1}^{n} \left( \frac{P_{ij} - T_j}{T_j} \right)^2 \cdot \frac{1}{n \cdot P \cdot T}
\]

where:

- \(P_{ij}\) is the predicted value for the jth fitness case by the i-th model.
- \(T_j\) is the target value for the jth fitness case.
- \(n\) is the number of fitness cases.
- \(P\) is the number of models.
- \(T\) is the total number of fitness cases.

\[1\]

\[2\]
The correlation coefficient \( r \) is a quantity that gives the quality of a least squares fitting to the original image. For two data sets \( x, y \); the auto correlation is given by:

\[
r = \frac{\text{cov}(x, y)}{\sigma_x \times \sigma_y}
\]

where \( \sigma_x \) and \( \sigma_y \) are the standard deviation of image \( x \) and \( y \). Finally; the error percentage is calculated as the percentage difference between the measured value and the accepted value.

A. **Multilayer perceptrons (MLPs)**

Multilayer perceptrons (MLPs) are layered feedforward networks typically trained with static backpropagation. These networks have found their way into countless applications requiring static pattern classification. Their main advantage is that they are easy to use, and that they can approximate any input/output map. The key disadvantages are that they train slowly, and require lots of training data (typically three times more training samples than network weights).

B. **Generalized feedforward networks (GFF)**

Generalized feedforward networks are a generalization of the Multilayer perceptrons (MLP) such that connections can jump over one or more layers. In theory, a MLP can solve any problem that a generalized feedforward network can solve. In practice, however, generalized feedforward networks often solve the problem much more efficiently. A classic example of this is the two spiral problem. Without describing the problem, it suffices to say that a standard MLP requires hundreds of times more training epochs than the generalized feedforward network containing the same number of processing elements.

C. **Modular feedforward networks (MFN)**

Modular feedforward networks are a special class of MLP. These networks process their input using several parallel MLPs, and then recombine the results. This tends to create some structure within the topology, which will foster specialization of function in each sub-module. In contrast to the MLP, modular networks do not have full interconnectivity between their layers. Therefore, a smaller number of weights are required for the same size network (i.e. the same number of PEs). This tends to speed up training times and reduce the number of required training exemplars. There are many ways to segment a MLP into modules. It is unclear how to best design the modular topology based on the data. There are no guarantees that each module is specializing its training on a unique portion of the data.

D. **Jordan and Elman networks**

Jordan and Elman networks extend the multilayer perceptron with context units, which are processing elements (PEs) that remember past activity. Context units provide the network with the ability to extract temporal information from the data. In the Elman network, the activity of the first hidden PEs are copied to the context units, while the Jordan network copies the output of the network. Networks which feed the input and the last hidden layer to the context units are also available.

E. **Principal component analysis networks (PCAs)**

Principal component analysis networks (PCAs) combine unsupervised and supervised learning in the same topology. Principal component analysis is an unsupervised linear procedure that finds a set of uncorrelated features, principal components, from the input. A MLP is supervised to perform the nonlinear classification from these components.

F. **Radial basis function (RBF) networks**

Radial basis function (RBF) networks are nonlinear hybrid networks typically containing a single hidden layer of processing elements (PEs). This layer uses gaussians are set by unsupervised learning rules, and functions employed by MLPs. The centers and widths of the gaussians are set by unsupervised learning rules, and supervised learning is applied to the output layer. These networks tend to learn much faster than MLPs. If a generalized regression (GRNN) / probabilistic (PNN) net is chosen, all the weights of the network can be calculated analytically. In this case, the number of cluster centers is by definition equal to the number of exemplars, and they are all set to the same variance. Use this type of RBF only when the number of exemplars is so small (<100) or so dispersed that clustering is ill-defined.

G. **Self-organizing feature maps (SOFMs)**

Self-organizing feature maps (SOFMs) transform the input of arbitrary dimension into a one or two dimensional discrete map subject to a topological (neighborhood preserving) constraint. The feature maps are computed using Kohonen unsupervised learning. The output of the SOFM can be used as input to a supervised classification neural network such as the MLP. This network's key advantage is the clustering produced by the SOFM which reduces the input space into representative features using a self-organizing process. Hence the underlying structure of the input space is kept, while the dimensionality of the space is reduced.

H. **Time lagged recurrent networks (TLRNs)**

Time lagged recurrent networks (TLRNs) are MLPs extended with short term memory structures. Most real-world data contains information in its time structure, i.e. how the data changes with time. Yet, most neural networks are purely static classifiers. TLRNs are the state of the art in nonlinear time series prediction, system identification and temporal pattern classification.
1. **Recurrent networks**

   Fully recurrent networks feedback the hidden layer to itself. Partially recurrent networks start with a fully recurrent net and add a feedforward connection that bypasses the recurrences, effectively treating the recurrent part as a state memory. These recurrent networks can have an infinite memory depth and thus find relationships through time as well as through the instantaneous input space. Most real-world data contains information in its time structure. Recurrent networks are the state of the art in nonlinear time series prediction, system identification, and temporal pattern classification.

   The learning curves of the nine discussed ANN classifiers are illustrated from Fig.2 to Fig.10 within 1000 epochs.

---

**Fig. 2.** Learning curve of MLPS classifier after 1000 epochs

**Fig. 3.** Learning curve of GFN classifier after 1000 epochs

**Fig. 4.** Learning curve of MFN classifier after 1000 epochs

**Fig. 5.** Learning curve of JEN classifier after 1000 epochs

**Fig. 6.** Learning curve of PCAS classifier after 1000 epochs

**Fig. 7.** Learning curve of RBF classifier after 1000 epochs
The performances of all tested ANN-based classifiers were evaluated through four performance indices; MSE, NMSE, r, and %Error. Table I through Table IX illustrates the results of the nine discussed ANN classifiers. All these classifiers are trained with the following parameters: (i) 10 processing elements; (ii) one hidden layer, and (iii) 1000 epochs.

**Table I**

<table>
<thead>
<tr>
<th>Results of MLP Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
</tr>
<tr>
<td>NMSE</td>
</tr>
<tr>
<td>r</td>
</tr>
<tr>
<td>% Error</td>
</tr>
</tbody>
</table>

**Table II**

<table>
<thead>
<tr>
<th>Results of GFN Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
</tr>
<tr>
<td>NMSE</td>
</tr>
<tr>
<td>r</td>
</tr>
<tr>
<td>% Error</td>
</tr>
</tbody>
</table>

**Table III**

<table>
<thead>
<tr>
<th>Results of MFN Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
</tr>
<tr>
<td>NMSE</td>
</tr>
<tr>
<td>r</td>
</tr>
<tr>
<td>% Error</td>
</tr>
</tbody>
</table>

**Table IV**

<table>
<thead>
<tr>
<th>Results of JEN Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
</tr>
<tr>
<td>NMSE</td>
</tr>
<tr>
<td>r</td>
</tr>
<tr>
<td>% Error</td>
</tr>
</tbody>
</table>

**Table V**

<table>
<thead>
<tr>
<th>Results of PCAS Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
</tr>
<tr>
<td>NMSE</td>
</tr>
<tr>
<td>r</td>
</tr>
<tr>
<td>% Error</td>
</tr>
</tbody>
</table>

**Table VI**

<table>
<thead>
<tr>
<th>Results of RBF Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
</tr>
<tr>
<td>NMSE</td>
</tr>
<tr>
<td>r</td>
</tr>
<tr>
<td>% Error</td>
</tr>
</tbody>
</table>

**Table VII**

**Table VIII**
As a result of the comparative study it has been found that, the classifier based on Multilayer perceptrons provides the best results: MSE=0.019614889247; NMSE=0.061119179527; $r=0.571021701230$, and $\%$ Error=0.711851781093.

It should also be noted that, the Multilayer perceptrons provides the best results among all tested ANN classifiers for all groups of selected features. On the other hand, the Radial basis function based classifiers provides the worst results.

### VII. CONCLUSION

As a result of the comparative study; the Multilayer perceptrons (MLPS) based classifier provides the best results among nine other classifiers; about 99.63% true positive attacks are detected using this classifier. On the other hand, the Radial basis function based classifiers provides the worst results. And after applying the proposed framework we can obtain online intrusion detection system which is intelligent by detecting novel attacks and also adaptive by updating the attack signature database.

### REFERENCES


