Hybrid Approach Using Intrusion Detection System

Tariq Ahamad[1], Abdullah Aljumah[2]
College of Computer Engineering & Sciences
Salman Bin Abdulaziz University, KSA

***Abstract*— The rapid growth of the computers that are interconnected, the crime rate has also increased and the ways to mitigate those crimes has become the important problem now. In the entire globe, organizations, higher learning institutions and governments are completely dependent on the computer networks which plays a major role in their daily operations. Hence the necessity for protecting those networked systems has also increased. An intrusion detection system (IDS) inspects all inbound and outbound network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system. In this research article, we will try to analyse different intrusion detection approaches SVM, ANN, SOM , Fuzzy Logic. In this research article, we have proposed a new technique that will tackle with all these different intrusion attacks. We propose a hybrid kind of approach that might be useful while facing these vicious network intrusion attacks.**

Key Words:IDS, SOM, ANN, SVM,

I. INTRODUCTION

Currently, Internet information resources are actively growing, penetrating many spheres of social life. Information technologies are being introduced not only into private enterprises, but also in the provision of public services. With each passing day, more and more confidential transactions are carried out via the Internet. In connection with these trends, the question of computer networks security is starkly raised. Attackers have developed and actively use many types of network intrusion , most of which can be prevented by standard methods of protection.

Intrusion detection is defined as the processes to identify the internal or external users who intend to do something unauthorized against the computer system [I]. Intrusion detection also identifies the legal connected users who intend to misuse their privileges. Intrusion detection systems (IDS) are based on the principle that malicious behaviours on computer or network systems will be noticeably different from normal behaviours. The IDS receives and analyses many data sources from computer systems or networks to detect abnormal patterns generated by the intruders who intend to attack or penetrate the computer and network system[2]. The general IDSs should have the ability to detect unauthorized access/modification of system or user information/files, network component information and unauthorized use of system resources.

Network-based attack detection routines, meanwhile, usually use network traffic data from a network packet sniffer (e.g., tcpdump). Many computer networks, including the commonly accepted Ethernet (IEEE 802.3) network, use a shared medium for communication. Therefore, the packet sniffer only needs to be on the same shared subnet as the monitored machines.
We have used the following four approaches:

1. ANN or Artificial Neural Network, artificial neural networks are computational models inspired by animals' central nervous systems (in particular the brain) that are capable of machine learning and pattern recognition. They are usually presented as systems of interconnected "neurons" that can compute values from inputs by feeding information through the network. ANN is one of the oldest systems that have been used for Intrusion Detection System (IDS), which presents supervised learning methods.
2. SOM Self Organizing Map, A self-organizing map (SOM) or self-organizing feature map (SOFM) is a type of artificial neural network (ANN) that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discredited representation of the input space of the training samples, called a map. Self-organizing maps are different from other artificial neural networks in the sense that they use a neighbourhood function to preserve the topological properties of the input space which is an ANN-based system, but applies unsupervised methods.
3. Fuzzy Logic (IDS-based), which also applies unsupervised learning methods.
4. SVMs, Support Vector Machines (also support vector networks) are supervised learning models with associated learning algorithms that analyse data and recognize patterns, used for classification and analysis. we will look at the SVM system or Support Vector Machine for IDS.

A. Artificial Neural Network ANN-IDS

One type of network sees the nodes as ‘artificial neurons’. These are called artificial neural networks (ANNs). An artificial neuron is a computational model inspired in the natural neurons. Natural neurons receive signals through synapses located on the dendrites or membrane of the

neuron[3]. When the signals received are strong enough (surpass a certain threshold), the neuron is activated and emits a signal though the axon. This signal might be
sent to another synapse, and might activate other neurons.

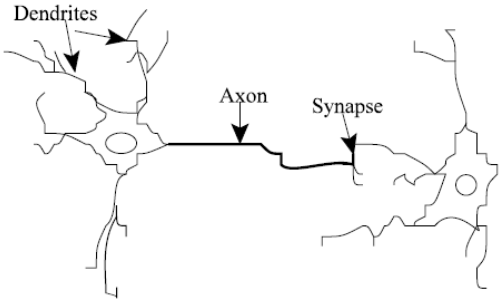


Figure 1. Natural neurons (artist's conception).

The complexity of real neurons is highly abstracted when modelling artificial neurons. These basically consist of inputs (like synapses), which are multiplied by weights (strength of the respective signals), and then computed by a mathematical function which determines the activation of the neuron[4]. Another function (which may be the identity) computes the output of the artificial neuron (sometimes in dependance of a certain threshold). ANNs combine artificial neurons in order to process information.
The higher a weight of an artificial neuron is, the stronger the input which is multiplied by it will be. Weights can also be negative, so we can say that the signal is inhibited by the negative weight. Depending on the weights, the computation of the neuron will be different. By adjusting the weights of an artificial neuron we can obtain the output we want for specific inputs. But when we have an ANN of hundreds or thousands of neurons, it would be quite complicated to find by hand all the necessary weights. But we can find algorithms which can adjust the weights of the ANN in order to obtain the desired output from the network. This process of adjusting the weights is called learning or training.

An Artificial Neural Network (ANN) is comprised of a collection of processing elements that are highly interconnected, and convert a set of inputs to a set of desired outputs. The outcome of the transformation is determined by the traits or characteristics of the elements, and the weights associated with the interconnections among them[5]. By altering the connections between the nodes, the network is able to adapt to the desired outputs.

Unlike expert systems, this can provide the user with a definitive answer if the characteristics, which are reviewed, perfectly match those which have been coded in the rule base. Neural network performs an analysis of the information, and presents a probability estimate that the data matches the characteristics, which it has been trained to recognize[6]. While the possibility of a match established by a neural network can be 100%, the precision or accuracy of its decisions entirely depends on the experience the system gains in analyzing examples of the stated problem.

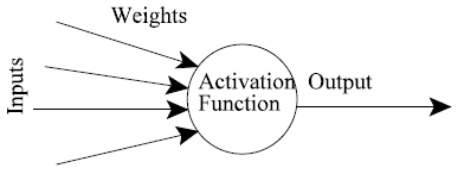
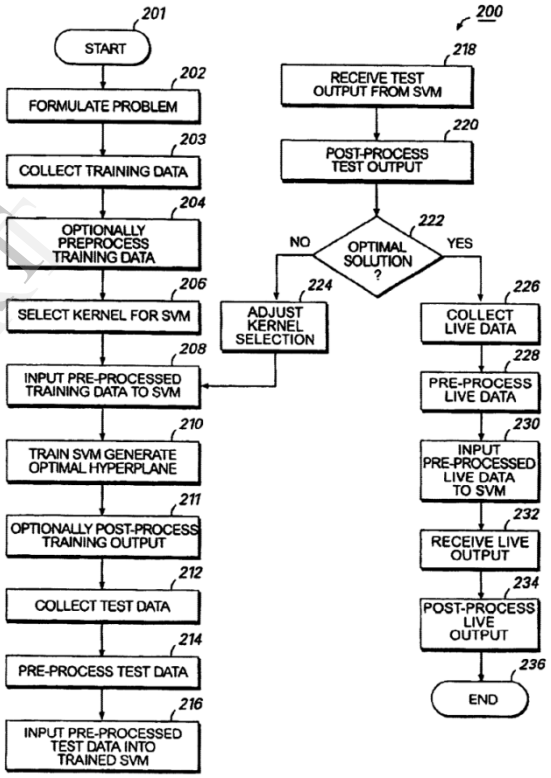


Figure 2. An artificial neuron

Initially, the neural network obtains the experience by training the system to accurately identify preselected examples of the problem. The feedback of the neural network is then assessed and the configuration of the system is improved and perfected until the neural network's analysis of the training data attains a satisfactory level[7]. Apart from the initial training period, the neural network also gains experience over time as it carries out analyses on data related to the problem.



II. SUPPORT VECTOR MACHINE SVM-IDS

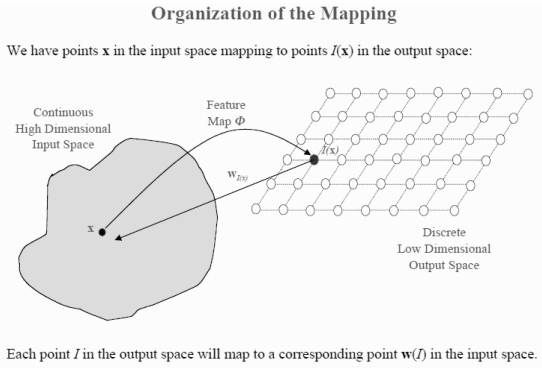
Support Vector Machines (SVM's) are a relatively new learning method used for binary classification. The basic idea is to find a hyperplane which separates the d-dimensional data perfectly into its two classes. However, since example data is often not linearly separable, SVM's introduce the notion of a \kernel induced feature space" which casts the data into a higher dimensional space where the data is separable [8]. Typically, casting into such a space would cause problems computationally, and with overfitting. The key insight used in

SVM's is that the higher-dimensional space doesn't need to be dealt with directly (as it turns out, only the formula for the dot-product in that space is needed), which eliminates the above concerns[9]. Furthermore, the VC-dimension (a measure of a system's likelihood to perform well on unseen data) of SVM's can be explicitly calculated, unlike other learning methods like neural networks, for which there is no measure. Overall, SVM's are intuitive, theoretically well- founded, and have shown to be practically successful. SVM's have also been extended to solve regression tasks (where the system is trained to output a numerical value, rather than \yes/no" classification). Support Vector Machines were introduced by Vladimir Vapnik and colleagues. The earliest mention was in (Vapnik, 1979), but the first main paper seems to be (Vapnik, 1995).

Support Vector Machines , or SVMs, are learning machines that plot the training vectors in high dimensional feature space, labelling each vector by its class. SVMs look at the classification problem as a quadratic optimization problem[10]. They combine generalization control with a method to prevent the "curse of dimensionality" by placing an upper bound on the margin between the different classes, making it a practical tool for large and dynamic data sets. The categorization of data by SVMs is done by determining a

set of support vectors, which are members of the set of training inputs that outline a hyper plane in feature space.

There are two main reasons for our experimentation with SVMs for intrusion detection. The first is speed because real time performance is of key importance to intrusion detection systems, and any classifier that can potentially outrun neural networks is worth considering. The second reason is scalability: SVMs are relatively insensitive to the number of data points and the classification complexity does not depend on the dimensionality of the feature space



III. SELF ORGANISING MAP SOM-IDS

So far we have looked at networks with supervised training techniques, in which there is a target output for each input pattern, and the network learns to produce the required outputs. We now turn to unsupervised training, in which the networks learn to form their own classifications of the training data without external help. To do this we have to assume that class membership is broadly defined by the input patterns sharing common features, and that the network will be able to identify those features across the range of input patterns.

One particularly interesting class of unsupervised system is based on competitive learning, in which the output neurons compete amongst themselves to be activated, with the result that only one is activated at any one time. This activated neuron is called a winner-takes all neuron or simply the winning neuron[11]. Such competition can be induced/implemented by having lateral inhibition connections (negative feedback paths) between the neurons. The result is that the neurons are forced to organise themselves. For obvious reasons, such a network is called a Self Organizing Map (SOM).

The self-organization map process involves four major components:

Initialization: All the connection weights are initialized with small random values.
Competition: For each input pattern, the neurons compute their respective values of a discriminant function which provides the basis for competition. The particular neuron with the smallest value of the discriminant function is declared the winner.
Cooperation: The winning neuron determines the spatial location of a topological neighbourhood of excited neurons, thereby providing the basis for cooperation among neighbouring neurons.
Adaptation: The excited neurons decrease their individual values of the discriminant function in relation to the input pattern through suitable adjustment of the associated connection weights, such that the response of the winning neuron to the subsequent application of a similar input pattern is enhanced.

Unsupervised learning methods using SOM provide a simple and efficient way to classify data sets. To process real-time data for classification, we consider SOMs to be best suited due to their high speed and fast conversion rates, as compared with other learning techniques. In addition to this, SOMs also preserve topological mappings between representations, a feature which is preferred when categorizing normal vs. intrusive behavior for network data. That is, the relationships between senders, obtained sample results statically by collecting different sample network traffic representing normal as well as DoS attack .

IV. FUZZY LOGIC-IDS

Fuzzy logic starts and builds on a set of user-supplied human language rules. The fuzzy systems convert these rules to their mathematical equivalents. This simplifies the job of the system designer and the computer, and results in much more accurate representations of the way systems behave in the real world[12]. Additional benefits of fuzzy logic include its simplicity and its flexibility. Fuzzy logic can handle problems with imprecise and incomplete data, and it can model nonlinear functions of arbitrary complexity. Fuzzy logic techniques have been employed in the computer security field since the early 90's (Hosmer, 1993). Its ability to model

complex systems made it a valid alternative, in the computer security field, to analyze continuous sources of data and even unknown or imprecise processes (Hosmer, 1993). Fuzzy logic has also demonstrated potential in the intrusion detection field when compared to systems using strict signature matching or classic pattern deviation detection. Bridges (Bridges and Vaughn, 2000), states the concept of security itself is fuzzy. In other words, the concept of fuzziness helps to smooth out the abrupt separation of normal behavior from abnormal behavior. That is, a given data point falling outside/inside a defined "normal interval", will be considered anomalous/normal to the same degree regardless of its distance from/within the interval[13]. Fuzzy logic has a capability to represent imprecise forms of reasoning in areas where firm decisions have to be made in indefinite environments like intrusion detection. The model suggested in (Dokas et al., 2002) building rare class prediction models for identifying known intrusions and their variations and anomaly/outlier detection schemes for detecting novel attacks whose nature is unknown. The latest in fuzzy is to use the
Markov model. As suggested in (Xu et al., 2004) a Window Markov model is proposed, the next state in the window equal evaluation to be the next state of time t, so they create Fuzzy window Markov model. As discussed, researchers propose a technique to generate fuzzy classifiers using genetic algorithms that can detect anomalies and some specific intrusions. The main idea is to evolve two rules, one for the normal class and other for the abnormal class using a profile data set with information related to the computer network during the normal behaviour and during intrusive (abnormal) behaviour.

With the fuzzy input sets defined, the next step is to write the rules to identify each type of attack. A collection of fuzzy rules with the same input and output variables is called a fuzzy system. We believe the security administrators can use their expert knowledge to help create a set of rules for each attack.

The rules are created using the fuzzy system editor contained in the Matlab Fuzzy Toolbox. This tool contains a graphical user interface that allows the rule designer to create the member functions for each input or output variable, create the inference relationships between the various member functions, and to examine the control surface for the resulting fuzzy system. It is not expected, however, that the rule designer utterly relies on intuition to create the rules[14]. Visual data mining can assist the rule designer in knowing which data features are most appropriate and relevant in detecting different kinds of attacks.

V. TYPE OF ATTACKERS

A. *Host and Port Scanning*

The Internet today is a complex entity comprised of diverse networks, users, and resources. Most of the users are oblivious to the design of the Internet and its components and only use the services provided by their operating system or applications. However, there is a small minority of advanced users who use their knowledge to explore potential system vulnerabilities. Hackers can compromise the vulnerable hosts and can either take over their resources or use them as tools for future attacks. With so many different protocols and countless implementations of each for different platforms, the launch of an effective attack often begins with a separate process of identifying potential victims.

One of the popular methods for finding susceptible hosts is port scanning. Port scanning can be defined as "hostile Internet searches for open 'doors,' or ports, through which intruders gain access to computers." This technique consist of sending a message to a port and listening for an answer. The received response indicates the port status and can be helpful in determining a host's operating system and other information relevant to launching a future attack.
Attackers often conduct host and port scans as Precursors to other attacks. An intruder will try to establish the existence of hosts on a network or whether a particular service is in use. A host scan is normally characterized by unusual number of Connections to hosts on the network from an uncommon origin. The scans may use a variety of Protocols, and may also utilize an identifier called an SDP to represent a unique link between a source, destination, and a service port.

B. *Denial of Service Detection*

DoS attacks, which come in many forms, are explicit attempts to block legitimate users' system access by reducing system availability. We could, for example, consider the intentional removal of a system's electrical power as a physical DoS attack. An attacker could also render a computing resource unavailable by modifying the system configuration (such as its static routing tables or password files). Such physical or host-based intrusions are generally addressed through hardened security policies and authentication mechanisms. Although software patching defends against some attacks, it fails to safeguard against DoS flooding attacks, which exploit the unregulated forwarding of Internet packets. A secondary defense that includes both attack detection and countermeasures is required.A common attack scenario is when an attacker overwhelms a target machine with too much data. This chokes the target and inhibits it from performing its intended role.

Denial of service (dos) attacks can take a variety of forms, and use different types of Protocols. We developed a representative Fuzzy System for a common dos attack based on ICMP Traffic congestion, and to test the system, we launched an ICMP dos attack called ping flood against a target in a controlled environment, collected the network traces and input the resulting data to the fuzzy system.

C. *Unauthorized Servers Detection*

Another intrusion detection scenario, which is potentially more damaging than the previous two scenarios, is when an attacker invades a system and install a backdoor or Trojan horse program that can lead to further compromise. Telltale activity that can help identify such intrusions include identifying unusual service ports that are in use on the network, unusual numbers of connections from foreign or

unfamiliar hosts, and/or unusual amounts of network traffic load to from a host on the network.

VI. CONCLUSION

In this research article , we proposed two types of Artificial Intelligence system, both supervised and unsupervised. In the article, ANN and SVM represent the supervised methods, while SOM and Fuzzy Logic represent the unsupervised methods. We have proposed that hybrid-based approaches can overcome problems that appear in the prediction of the IDS and the attacks can be stopped and if not stopped we might get enough time to defend. Lot of research have been done on this and we have a lot to do yet. In future we will try to improve and give detailed and better form for our approach

REFERENCES

1. Xiapu Luo, Edmond W.W.Chan,Rocky K.C.Chang: Detecting Pulsing Denial-of-Service Attacks with Nondeterministic Attack Intervals, EURASIP Journal on Advances in Signal Processing (2009)
2. Nagesh,H.R.,Chandra Sekaran,K.: Design and Development of Proactive Models for Mitigating Denial-of-Service and Distributed Denial-of-Service Attacks, International Journal of Computer Science and Network Security, Vol. 7, No.7 (2007).
3. Nagy, H., Watanabe, K., and Hirano, M. (2002). "Prediction of Sediment Load Concentration in Rivers using Artificial Neural Network Model." *J. Hydraul. Eng.*, 128(6), 588–595.
4. ''Use of neural networks in design of coastal sewage system.'' *J.Hydraul. Eng.*, 124~5!, 457–464.
5. Grubert, J. P., ~1995!. ''Application of neural networks in stratified flow stability analysis.'' *J. Hydraul. Eng.*, 121~7!, 523–532.
6. Rashidian, V. and Hassanlourad, M. (2014). "Application of an Artificial Neural Network for Modeling the Mechanical Behavior of Carbonate Soils." *Int. J. Geomech.*, 14(1), 142–150.
7. Demuth, H. , Beale, M. , and Hagan, M. (2007). *Neural Network Toolbox 5 user's guide* , MathWorks, Natick, MA.
8. Lam, K., Lam, M., and Wang, D. (2010). "Efficacy of Using Support Vector Machine in a Contractor Prequalification Decision Model." *J. Comput. Civ. Eng.*, 24(3), 273–280.
9. Lam, K. C. , Hu, T. S. , and Ng, S. T. (2005). "Using the principal component analysis method as a tool in contractor pre-qualification." *Constr. Manage. Econom.* , **23** (**7**), 673–684
10. Cristianini, N. , and Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods* , Cambridge University Press, Cambridge, U.K
11. Chang, C. C. , and Lin, C. J. (2004). "LIBSVM: A library for support vector machines." Dept. of Computer Science and Information Engineering, National Taiwan Univ.
12. Wang, K. and Altunkaynak, A. (2012). "Comparative Case Study of Rainfall-Runoff Modeling between SWMM and Fuzzy Logic Approach." *J. Hydrol. Eng.*,
13. Altunkaynak, A. , and Şen, Z. (2007). "Fuzzy logic model of lake water level fluctuations in Lake Van, Turkey." *Theor. Appl. Climatol.* , **90** (**3–4**), 227–233.
14. Pappis, C. P. , and Mamdani, E. H. (1977). "A fuzzy logic controller for a traffic junction." *IEEE Trans. Syst. Man Cybern.* , **7** (**10**), 707–717.