Anomaly Based Intrusion Detection for a Biometric Identification System using Neural Networks

Tinny Mgabile, Ishmael S. Msiza, and Erick Dube

***Abstract*—**This manuscript presents a supervised machine learning approach in the identification of network attacks on a fingerprint biometric system. To reduce the problem of malicious acts on a biometric system, this manuscript proposes an intrusion detection technique that analyses the fingerprint biometric network traffic for evidence of intrusion. The neural network algorithm that imitates the way a human brain works is used in this study to classify normal traffic and learn the correct traffic pattern on a fingerprint biometric system. The aim of the study is to observe the ability of the neural network in the detection of known and unknown attacks without using a vast amount of training data. The results of the neural network model had a classification rate of 98 %, which translates to a false positive rate of 2%.

***Keywords*—**biometric system, neural network, machine learning, intrusion detection.

I. Introduction

BIOMETRIC technology provides a reliable authentication method over traditional security methods that are based on personal identification numbers (PIN) or passwords. Biometric modalities – such as fingerprints, iris, and voice – that are used to gain access through biometric authentication systems are unique for each individual and somewhat difficult to forge. Traditional security methods like PIN or passwords are considered as unreliable authentication methods as they cannot differentiate between fake and genuine users because they can easily be misplaced, shared, and/or stolen. The use of biometric technology is becoming popular due to its ease in both identification and verification, and is considered the most effective and safe method because it establishes the identity of a subject using their physical or behavioral characteristics [1].

Biometric reference data may be stored locally on the data acquisition device, locally on a personal computer, or it can be stored on a remote server depending on the requirements of an organization. Storing biometric reference data locally offers some benefits, including the reduction of data transmission risk. Storing biometric reference data on a remote server also has its own benefits. For example, the server can accommodate large numbers of biometric reference data and, in addition, users can get enrolled at one location in the system and be verified from any other location. While it has its own benefits, storing the biometric reference data on a remote server also creates opportunities for data transmission risk, as the amount of information theft and unauthorized access increases on networked systems. Researchers have identified all the vulnerabilities within a networked biometric system and, where possible, intrusions are more likely to happen, as illustrated in Fig.1 below.

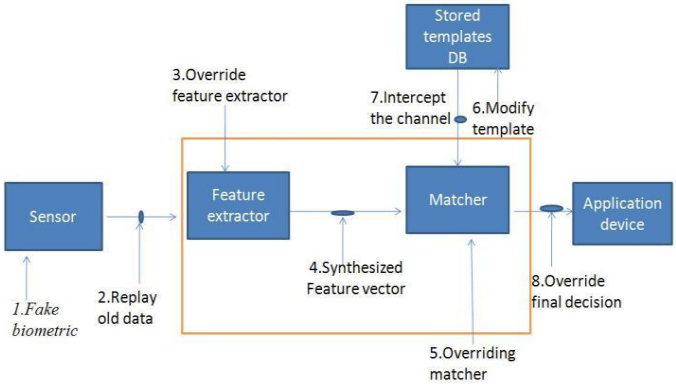

Fig. 1 Attacked points on a networked biometric system [2]

Ratha *et al*. [2] confirm the vulnerability concern on biometric systems, and this is mostly due to the fact that the system is heavily reliant on the communication network. Because of this fact, the system becomes exposed to an increasing number of network intrusions. If protection is not in place between the sub modules in a biometric architecture, the system might be vulnerable to attacks such as:

- eavesdropping,
- replay attacks,
- spoofing, and
- denial of service (DoS).

These types of attacks bypass the sensor module by injecting or replaying a digital signal that has already been

Tinny Mgabile is with the CSIR Modeling & Digital Science, Pretoria, South Africa (phone: +27 12 841 3387; fax: +27 12 841 2456; e-mail: TMgabile@csir.co.za) and the Faculty of Engineering & the Built Environment, University of Johannesburg, South Africa (phone: +27 11 559 2114; fax: +27 11 559 2054)

Ishmael S. Msiza is with the CSIR Modeling & Digital Science, Pretoria, South Africa (e-mail: IMsiza@csir.co.za)

Erick Dube is with the CSIR Modeling & Digital Science, Pretoria, South Africa (e-mail: EDube1@csir.co.za)

recognized as authentic. This act enrolls a malicious user with an authentic user's biometric identity. With this, and other similar reasons, it is necessary to have a defense mechanism in place. This study aims to experiment with biometric network traffic as parameters in an anomaly based intrusion detection system using artificial neural networks.

II. Background

Transmitting data in a networked system creates multiple points of attack where fraudulent activities can occur, as data passes between network sub modules, as illustrated in Fig.1. Different defense methods such as firewalls, patches, encryption, and intrusion detection have been used to secure network infrastructure from malicious activities. Intrusion detection, however, still maintains an important technique in providing secure network architecture. An intrusion detection system (IDS) is a program or software that can detect intrusion and notify the Network Administrator about an attack or misuse [3].

As the name suggests, this security software does not prevent the intruders from getting into the system, it simply detects an intruder within the system and provides the Network Administrator with relevant information necessary to pick up the presence of an intruder. In addition, it is considered to be the first line of defense in many organizations in providing security to network and computer systems [4].

Literature reveals two main categories of IDS: the network based intrusion detection system (NIDS), and host based intrusion detection system (HIDS). HIDS detects possible attacks on an individual computer that has an intrusion detection system is installed. It monitors and analyses the internals of a computer system rather than the network packets on its external interfaces [5]. NIDS, on the other hand, uses network traffic data from a network packet sniffer. It listens to traffic on a given network and, by doing so, is able to detect both legitimate and attack traffic as it passes through [5].

IDS differ in ways they detect attacks and threats. The most common approaches used to detect attacks include anomaly intrusion detection, misuse intrusion detection, and hybrid intrusion detection.

The anomaly based intrusion detection approach responds to deviation from normal behavior, which typically involves the creation of a knowledge base that contains the profile of monitored activities. Examples might be an unusually high number of network connections within an interval of time, unusually high CPU activity, or use of peripheral devices not usually used. Anomaly detection systems are good enough to find known attacks as well as unknown attacks but, performance wise, these systems are not good, because they produce false positives at a higher rate [6].

Misuse based intrusion detection approaches aim to match the user's activity with known behavior of attacks attempting to penetrate the system. For example, exploitation of the finger and send mail bugs used in the internet worm attack would be in this category. Misuse detection systems are good enough to find all known attacks and are efficient, performance wise. It means that they may produce a minimum number of false positives or almost zero, but are unable to find new intrusions or attacks [6].

Each of these approaches – misuse and anomaly intrusion detection – presents some benefits and some drawbacks. Studies have been conducted regarding the combining of these approaches into one single hybrid system to achieve the goal of high accuracy rate and low false positive rate [7].

Hybrid intrusion detection takes advantage of both approaches and combines the output of the two intrusion detection approaches. The hybrid IDS is more powerful than anomaly based detection on its own because it uses the advantage of the misuse approach for detecting attacks.

Researchers have identified the limitations concerning the network intrusion detection system. Either anomaly or misuse based; they have in common a drawback mechanism of false alarm and their performance deteriorates with increasing network traffic [8]. False alarm is a state where a network intrusion detection system sensor would classify legitimate traffic as a possible intrusion. These false alarms can be due to the fact that the system is lacking background knowledge, hence causing the system to ignore roughly half of the network traffic, that is, the system detects traffic which can be either legitimate or malicious activity and it cannot reliably decide whether it is an attack or not. A high rate of false alarm can cause an overflow for the administrator and can also jeopardize the security of the system.

III. Supervised Machine Learning

This study employs a machine learning (ML) technique to intrusion detection systems, to overcome the limitations experienced by the current intrusion detection systems as highlighted in section II. ML is an area of artificial intelligence that deals with the study of computer algorithms that allow the computer to learn and adapt new techniques [9].

Machine learning techniques have been applied to many areas of research and have shown tremendous results in learning the behavior of various systems. Peters, a Ph.D. student at the University of Southern California applied this technique of learning through experience to the motor skills in robotics [10]. This is but one example of many applications of machine learning techniques.

Today, research in machine learning techniques has grown significantly. Researchers have realized the fact that, with machine learning, it is possible to achieve the best performance in computer systems because of its ability to learn through experience. In 2008, Sewell [11] published a manuscript highlighting the three types of machine learning techniques:

- supervised learning,
- unsupervised learning, and
- reinforcement learning.

In this study, a supervised learning technique is used in the process of training an artificial neural network. The benefit of using a supervised model is that it allows the neural network's weights to be adjusted between the input data to obtain the desired output.

IV. Artificial Neural Networks

This technique simulates the learning process of the human brain. A biological study of the human brain shows that the brain is made up of a highly interconnected system of neurons, which are connected to each other by means of axons, which end in synapses. These synapses send signals along the stream called the dendrite into the neurons which make it powerful in forming and remembering patterns and also keeping these patterns intact. The brain has continued to be a mystery to many scientists like Warren McCulloch, a neuro physiologist, and Walter Pitts, a young mathematical genius who developed the concept of neural network by imitating the workings of the brain [12]. This highly interconnected system of neurons is called a biological neural network. Artificial neural networks work in a similar manner as biological neural networks. They connect processing elements to produce results from an input signal and depend on many interconnected variables, as illustrated in Fig. 2 below.

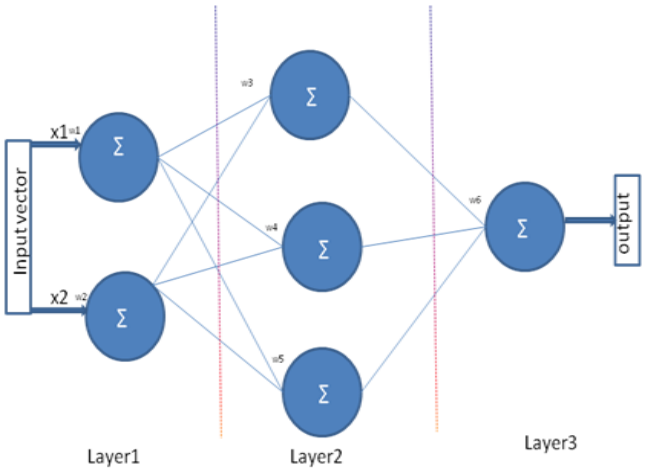

Fig. 2 Structure of the Neural Network

An artificial neural network consists of nodes connected together with links between them, where there is a directed flow of data. These connections have numeric weights to determine how much one node will affect the other. The numeric weight can be manipulated until desired outputs are attained. The network architecture adopted in this study is the multi layer perception (MLP).

The advantage of using an artificial neural network is that it is an inherent parallel processor and it is also adaptive, meaning that it can be trained to take decisions on its own. The capability of an artificial neural network to learn and characterize data and recognize instances that are unlikely is one of its important features.

V. Dataset

The datasets for training and testing were both obtained from a simulation of a fingerprint biometric system. The simulation consists of a fingerprint scanner and two personal computers (PC's), both running on a Linux environment. One PC acts as a client machine which connects to the fingerprint scanner. The other PC is the server machine which performs the authentication process and also contains the database that stores the fingerprint templates. . Additionally, a third machine was used to act as a man in the middle (MITM) to sniff and record the traffic as it moves between the client and the server machines. Finally, all data from the dataset was collected using a TCPDUMP sniffer program readily available from the Linux operating system and services. The collected dataset was divided into two groups: a group of network packets sent during the authentic biometric identification and a group of network packets collected during a replay session. These packets are authentic identification packets sent earlier which are now replayed.

A. Data Pre Processing

Before the data pre processing stage, the dataset collected from a packet sniffer contained packet header information, namely:

- Source addresses, destination addresses, and ports.
- Flags like SYN, FIN, PUSH, and RST (TCP only).
- Data sequence number of the packet (TCP only).
- Data sequence number of the data expected in return (TCP only).
- Number of bytes of receive buffer space available (TCP only).
- Indication of whether or not the data is urgent (TCP only).
- Packet length (TCP only), etc.

The packet header information was pre-processed into the format that the neural network finds it easy to identify during the training session. Generally, the input data to the neural network is the output data of the pre-processing stage. During the pre-processing stage, the training data was divided into two pairs: the input data and the output data. The input data consisted of 15 network parameters to be fed into the neural network as training data. The second part of the training pair was a two bit output vector. This tells the neural network whether the presented session is an attack or normal traffic. If it is classified as an attack the neural network would output binary 0, while if it is classified as normal the neural network would output binary 1.

B. Sorting Data

For training and testing of the neural network a total number of 400 instances was used, with normal and attack files equally divided. This means that 200 instances represent normal traffic, whereas the other 200 instances represent a replayed

traffic. The dataset is separated into two equal sets of 200 instances, namely, the training dataset and testing dataset. Both datasets contain 100 instances of normal and 100 instances of replayed traffic, as illustrated in TABLE I.

TABLE I
TRAINING DATASET

	Training Set	Testing Set
Normal	100	100
Attack	100	100
Total	**200**	**200**

VI. Neural Network Training Model

The neural network architecture used in this study is a three layer feed forward neural network with backpropagation as a learning method. The architecture consists of an input layer (with 15 input parameters), a hidden layer, and an output layer. A similar model had been used in previous studies to solve a classification problem in an intrusion detection system [13].

In this study, the number of nodes in the hidden layer complies with Kolmogorov's Theorem which states that, during the process of training a multi layer neural network model, the number of hidden units should be twice the number of input parameters [14].

For this study this simply implies the number of nodes in the hidden layer is 2 times 15 = 32. This, therefore, leads to the construction of a {15:32:1} neural network architecture.
The said model is constructed through the use of the MATLAB neural network toolbox [15]. MATLAB consist of suitable functions for creating a neural network simulation. The network is trained using a Logarithmic Sigmoid (Logsig) transfer function in, both, the input layer and the hidden layer; and a Pure Linear (purelin) transfer function in the output layer. The logsig is a continuous sigmoid function with its entire value lying between 0 and 1. Lastly, the purelin transfer function is used in the study as linear approximations, which has its values lie between 1 and -1.

A. Training

During the training phase, the neural network was presented with the training dataset illustrated in TABLE I. The network was trained using the *trainlm* function, which takes as arguments the network to be trained, the input data and the output data. Before the network was trained *init* function was used to initialize the weights. After learning is complete, the matrices containing the model's weights and biases are saved and later used to evaluate the model's ability to classify between normal and attack data.

B. Testing

The reliability of the neural network model is examined during the testing stage, where the model is tested on an unknown dataset – the testing dataset. When modelling the test results, it is important to evaluate a model's performance, such as measuring the accuracy of the model. Classification accuracy in the testing dataset is the only measure that gives usable information about the model's classification ability. Mathematically, this classification accuracy is presented as:

$$Accuracy = \frac{FP + TN}{TP + FP + TN + FN} \quad (1)$$

In this equation; TP represents a true positive, where the model classifies an intrusion as an intrusion; FP represents a false positive, where the model incorrectly classifies a normal data as an intrusion; TN represents a true negative, where the model correctly classifies normal data as normal; and FN represents an incorrect classification of an intrusion as normal data.

The concept of a confusion matrix serves as a visual representation of the accuracy measure, where the numerator of equation (1) is the sum of the diagonal entries of TABLE II, and the denominator is the sum of all the entries in this table.

TABLE II
ACCURACY RESULTS

	As	
Actual	Attack	Normal
Attack	100	00
Normal	02	98
Accuracy	98.0%	

VII. Conclusion

Machine learning techniques, like artificial neural networks, have shown excellent results in solving the accuracy concern and lowering the false positives rate on intrusion detection systems. With the evidence presented in this study, the neural networks proved to have more capabilities in improving accuracy and identifying patterns from the presented data. They also show an excellent ability to classify between normal and replayed attacks.

References

[1] K. Anil. K. Nandakumar, A. Nagar, "Biometric Template Security", *EURASIP Journal on Advances in Signal Processing. Hindawi Publishing Corporation.*, Article ID 579416. Doi:10.1155/2008/579416. 2008.

[2] N.K. Ratha, J.H. Connell, R.M. Bolle, " An analysis of minutiae matching strength. ," *Third International Conference on Audio- and Video-Based Biometric Person Authentication.*, pp. 223-228, 2001.

[3] G.V.Victor, R.M.Sreenivasa , V. CHVenkaiah, Intrusion Detection Systems – Analysis and Containment of False Positives Alert. *International Journal of Computer Applications*. Vol. 5, No. 8, Aug 2010.

[4] R. Raghudharan, "Intrusion Detection Systems: Beyond the first line of defense" [Online]. HYPERLINK "www.networkmagazineindia.com/200109/security1.htm"

[5] X. Wang, L. Hongxia , " Improvement and Implementation of Network Intrusion Detection System. ,"*Journal of Communication and Computer*. *USA.*, vol.3, no.1 (Serial No.14), Jan. 2006.

[6] A. Murali, M. Roa, "A survey on intrusion detection approaches," *Information and Communication Technologies, 2005. ICICT 2005. First International Conference on.*, pp. 233-240, Aug 2005.

[7] H.Kai, C. Min, et al, "Hybrid intrusion detection with weighted signature generation over anomalous internet episodes*"., IEEE*

Treansactions on dependable and secure computing.pp. 41-55, 2007.

[8] S. Kumar, "Survey of Current Network Intrusion Detection Techniques," 2007, pp. 1-18.

[9] Bishop,C.M, "Pattern recognition and machine learning".*Information Science and Statistics. New Yolk:Springer.*

[10] J. Peters, "Machine learning for motor skills in Robotics. ,"*Ph.D. Thesis at University of Southern California (USA).*

[11] M. Sewell, "Machine learning," Department *of Computer Science, University College London*., pp1-5, 2008.

[12] W,S. McCulloch, W.H. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biology*., vol.5, pp115-133. 1943.

[13] M. Pradhan, S.K. Pradhan, S.K. Sahu, "Anomaly detection using artificial neural network ., International Journal of Engineering Sciences & Emerging Technologies. Vol 2, issue 1, pp. 29-36, April 2012.

[14] I.S. Msiza, M. Szewczyk, A. Halinka, J-H.C. Pretorius, P. Sowa, and T. Marwala; "Neural Networks on Transformer Fault Detection: Evaluating the Relevance of the Input Space Parameters".; *IEEE Power Systems Conference and Exposition (PSCE); Phoenix, Arizona*.; ISBN: 978-1-61284-787-0, pp. 01-06, Mar. 2011.

[15] *Matlab and Simulink for Technical Computing*, Matlab 7.0 Manual Relaese 14: Mathworks, June 2004