Network Security Combining Cryptography and Steganography Using a Multi-Layer Perceptron Artificial Neural Network

Prof. Haresh R. Parmar
Assistant Professor in Computer Engineering Department @
Silver Oak College of Engineering & Technology

I. Introduction

Intrusion detection can be used to guard a host
Computer system should provide confidentiality, computer or network against being a source or a integrity and assurance against Denial of Service victim of an attack. Intrusion detection system is
(DoS). Due to increased connectivity, and the vast software that automates the intrusion detection spectrum of financial possibilities that are opening process. IDS has become increasingly vital over the up, more and more systems are subject to attack by last decade as network information systems have intruders. Any system connected to internet cannot grown into the daily life of most businesses, provide security without additional provision of government agencies and private citizens [13].
intrusion detection elimination software's [14].

Because of the increasing dependence in which
Every organization of even small size is connected companies and government agencies have their to the Internet. Due to functional requirements and own computer networks and the importance of cost factors, employees work from their home by protecting these systems from attack is critical. A connecting their systems with the main office. single intrusion of a computer network can result in
Employees exchange data in the form of revision, the loss of unauthorized utilization or modification completion of the work assigned to them. Financial of large amounts of data and cause users to question organization, automatic teller machines, landline the reliability of all of the information on the telephones, cellular phones, wireless networks network. IDSs can be categorized into three types provide internet facilities. The equipment which namely, network-based intrusion detection, routerrely upon main database stored in severs should not based intrusion detection, and host-based intrusion be damaged due to software threats in the form of detection [16]. Network based intrusion detection, intrusion. Military bases, nuclear research centers, operate at the gateway of a network and examines organization with top level information should not all the incoming packets. Router-based intrusion be damaged in the form of alteration, corruption of detection is installed on the routers to prevent information by any unknown activities entertained intruders from entering into the network. Finally, through the internet facilities by any one. Intrusion the host based intrusion detection receives the detection is the process of monitoring the events necessary audit data from the hosts operating occurring in a computer system or network and system and analyzes the generated events to keep analyzing them for signs of possible incidents the local system secure. A centralized scheme is which are violations of computer security policies. proposed to schedule authentication and intrusion

detection, which needed a centralized controller [12]. This is more opted for a single system rather than a network with distributed systems with random mobility. There are numerous methods of responding to a network intrusion [6], but they all require the accurate and timely identification of the attack. IDS detect DoS attacks either by using a priori knowledge of the types of known attacks or by recognizing deviations from normal system behaviors. DoS attacks aim at denying or degrading legitimate users access to a service or network resource, or at bringing down the servers offering such services.

Related Work

From Active Networking to Future Internet Research. Already, in the early 1990s, as the Internet rapidly grew and became ever more popular, researchers investigated dynamic network architectures. Tennenhouse and Wetherill proposed *Active Networks*, in which users could inject custom code into the network [7]. This code was associated with a set of packets that traversed the network from the source over several routers to the destination, and which was executed on intermediate nodes and modified the packets on the fly as desired. Tennenhouse andWetherall suggested four different possibilities for active packet processing:

(i) network operators inject code on the intermediate nodes (ii) every packet contains the program code to be executed (iii) packets can put code into a node and other packets could use that code (iv) packets contain a reference to code on an external server and the routers download the code from that server and store it in a local cache. Based on those four possibilities, many researchers worked on specific Active Networking architectures in the following ten years. This work is summarized by several survey papers [8–10]. However, none of the Active Network architectures found its way into the commercial Internet, some because of performance issues, others because of security concerns, and still others because no hardware supported them. Still, research on dynamic network architectures continued, driven on the one hand by Internet issues such as poor scalability, extensibility, security, and reliability, and on the other hand by a change from a static, provider-centric network to a mobile and user-centric network. For each of these issues, there exist efforts that tackle them. For example, scalability can be tackled with IPv6 or Network Address Translation (NAT), security problems can sometimes be mitigated by Virtual Private Networks (VPNs), and there is a large number of new routing or transmission protocols for mobile networks, such as Mobile IP or hop by hop transmission. The umbrella term for this research is *Future Internet*. Under this term, researchers investigate not only additions and patches to the Internet architectures, but also *clean slate architectures*. These architectures follow from asking the question “given what we know today, how would we have designed the Internet if we had to do it all over again?”. Our self-aware network architecture is also a clean-slate Future Internet architecture. *2.2. FPGAs in Active Networking.* The original work in Active

Networking exploited the flexibility of softwareonly systems. But already in 1998 the first extensions appeared that used FPGAs as hardware accelerators, even though, at that time, FPGAs were small and could only be reconfigured as a whole. For example, Hadis and Smith introduced the Programmable Protocol Processing Pipeline (P4) architecture [11]. It uses several FPGAs and a switching array that decides which packet will be processed by which FPGA. They can add new functions to the system by reconfiguring an FPGA, and they implement different protocol stacks by changing the path of a packet through the FPGAs. At the same time, Decapper et al. introduced the Active Network Node (ANN) [12]. The central component of their system is a switch which has, for each input port, a CPU and an FPGA.

Performance-critical functions are executed on the FPGA, which can be reconfigured by the CPU. However, they do not discuss how they decide when to reconfigure the FPGA. The Plato architecture is an FPGA-only architecture for Active Networks [13]. They also aim at reconfiguring the FPGA for flexibility but do not discuss how a reconfiguration could be started. Fragkiadakis et al. suggest an architecture with one CPU and one FPGA [14]. They allow only one active application to run at a time, and this application determines the functionality implemented on the FPGA. The currently active application depends on the received packet. There is no discussion on reconfiguration overhead of the FPGA, which seems surprising since the active application could change with every packet. To summarize, these architectures have realized the potential of FPGAs for active networking but have failed to convincingly address all aspects of using FPGAs. Either they failed to describe the circumstances under which an adaptation is triggered or they failed to discuss adaptation overhead and hence also the maximum adaptation frequency that still leads to an overall performance benefit.

Methodology

There have been many different encryption algorithms and public key cryptographic methods are being proposed to provide security to such data. All of these algorithms depend upon a user's key which he uses as the key for encryption. But these keys may be hacked by hacker, hence the only feature or data of a person that hackers cannot hack is their biometric features, hence this proposed project consider IRIS image of a user to generate secrete key for encryption. For security, only encryption may not be enough, hence proposed project include combination of both cryptography and Steganography. The encrypted data hide into the image and then image is transmitted in the network. There is some weakness in hiding information in images; that is adversary could easily detect the confidential message, by noticing the noise and clarity of the image's pixels, also by observing the difference between the embedded image and the original one if it is known to him. In the proposed project, here use Iris images instead of images that contain faces or natural scenes, because the only feature or data of a person that hackers cannot hack is their biometric features.

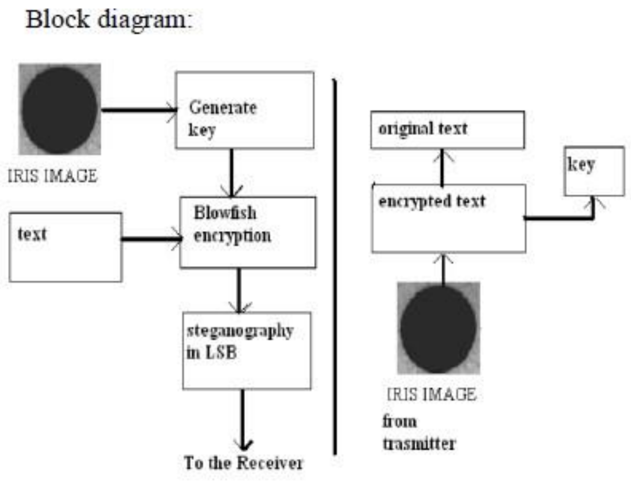


Figure 1. functional block diagram

Overview of algorithm

Image Definition

To a computer, an image is a collection of numbers that constitute different light intensities in different areas of the image [9]. This numeric representation forms a grid and the individual points are referred to as pixels. Most images on the internet consists of a rectangular map of the image’s pixels (represented as bits) where each pixel is located and its color [10]. These pixels are displayed horizontally row by row. The number of bits in a color scheme, called the bit depth, refers to the number of bits used for each pixel [11]. The smallest bit depth in current color schemes is 8, meaning that there are 8 bits used to describe the color of each pixel [11].

Least Significant Bit Algorithm

Least significant bit (LSB) insertion is a common, simple approach to embedding information in a cover image [3]. The least significant bit in other words, the 8th bit of some or all of the bytes inside an image is changed to a bit of the secret message. When using a 24-bit image, a bit of each of the red, green and blue color components can be used, since they are each represented by a byte. In other words, one can store 3 bits in each pixel. An 800 × 600 pixel image, can thus store a total amount of 1,440,000 bits or 180,000 bytes of embedded data [7]. For example a grid for 3 pixels of a 24-bit image can be as follows: When the number 200, which binary representation is 11001000, is embedded into the least significant bits of this part of the image, the resulting grid is as follows: ()()()0010110000011101101110101001111000100

000110110100110101100110001**100000011**

()()()001011000011101
110111001010011111000101

000011001101001101011100 01100011**100**

Although the number was embedded into the first 8 bytes of the grid, only the 3 underlined bits needed to be changed according to the embedded message. On average, only half of the bits in an image will need to be modified to hide a secret message using the maximum cover size [7]. Since there are 256 possible intensities of each primary color, changing the LSB of a pixel results in small changes in the intensity of the colors. These changes cannot be perceived by the human eye - thus the message is successfully hidden. With a well-chosen image, one can even hide the message in the least as well as second to least significant bit and still not see the difference [3].

Blowfish

Blowfish is a symmetric encryption algorithm, meaning that it uses the same secret key to both encrypt and decrypt messages. A graphical representation of the Blowfish algorithm appears in Figure 2. In this description, a 64-bit plaintext message is first divided into 32 bits. The "left" 32 bits are XORed with the first element of a P-array to create a value I'll call P', run through a transformation function called F, then XORed with the "right" 32 bits of the message to produce a new value I'll call F'. F' then replaces the "left" half of the message and P' replaces the "right" half, and the process is repeated 15 more times with successive members of the P-array. The resulting P' and F' are then XORed with the last two entries in the P-array (entries 17 and 18), and recombined to produce the 64-bit cipher text.

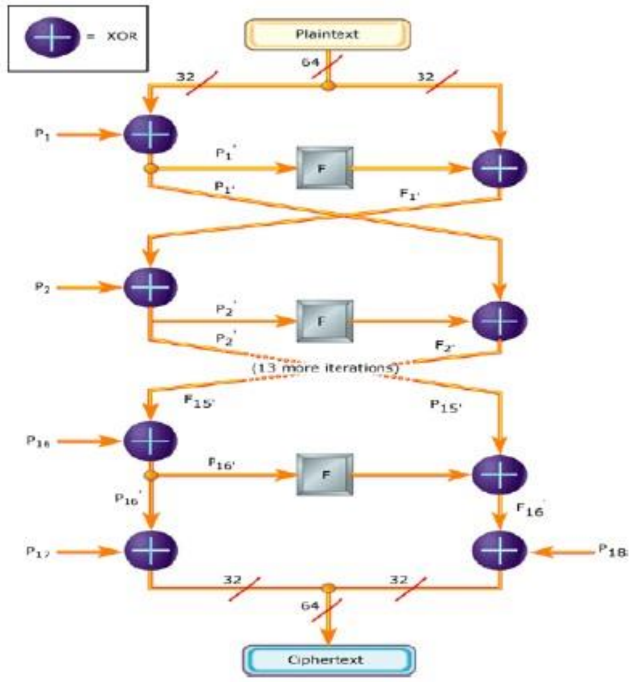


Figure 2. Blowfish algorithm

A graphical representation of F appears in Figure 2. The function divides a 32-bit input into four bytes and uses those as indices into an S-array. The lookup results are then added and XORed together to produce the output. The P-array and S-array values used by Blowfish are precompiled based on the user's key. In effect, the user's key is transformed into the P-array and S-array; the key itself may be discarded after the transformation.

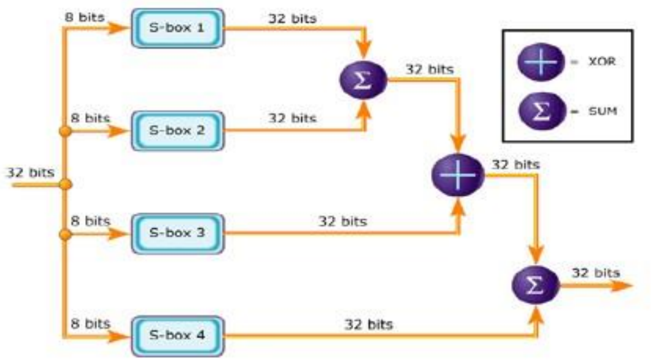


Figure 3. Graphical representation of F

Experimental Setup

PC must have Visual basic 6 software to run GUI. PC com port connected to ARM kit com Port. We used two UART port of ARM kit, one is connected to PC com port and second connected to ZigBee. So transmitter can acts as a receiver or receiver can acts as transmitter if required. As shown in Figure 4, for practical demonstration we required two PC or Laptops, two ARM kit, two ZigBee module and two serial com cables.

Finally, recombine xL and yr. to get the cipher text.

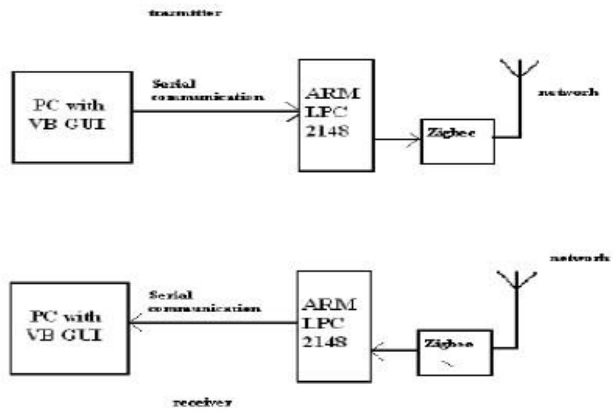


Figure 4. Experimental setup block diagram

Artificial Neural Networks

Artificial Neural Networks (ANN) has computing elements that are based on the structure and function of the biological neurons. The new algorithms are faster and give better performance. ANN consists of interconnected processing units. The general model of processing unit consists of summing part followed by an output part. The summing part receives n input values and weight values, and performs a weighted sum. The weighted sum is called the activation value. The sign of the weight for each input determines whether the input is excitatory (positive weight) or inhibitory (negative weight). The input and output could be the digital or analog data values. Several processing units are interconnected according to a selected topology of the network to achieve a pattern recognition task. The input of a processing unit may come from outputs of other processing units, and or from an external source. The output of each unit may be given to several units including it. A network can be static or dynamic; some of the static networks use the back propagation algorithm and radial basis function with multilayer perceptions. Some of the dynamical networks (recurrent networks) have output feedback, state feedback and feed forward dynamics. The learning of the network can be supervised or unsupervised. In supervised learning, both inputs and outputs are presented to the network. In the unsupervised learning (self-recognizing networks), the inputs alone are presented to the network. Some of the algorithms for unsupervised learning are adaptive resonance theory [3] self-organizing features maps [11]. One of the important applications of ANN is in pattern recognition analysis. A pattern is a set of inputs and outputs. Either supervised or unsupervised training method can be used to train an ANN, depending upon the network topology. In the supervised training, the difference between the calculated output of the network and the desired output of the pattern is minimized. To achieve the minimum difference, synaptic weights are updated.

This procedure is adopted for all the patterns.

Algorithms

Although a number of algorithms are investigated, a sample number of algorithms are presented here and their performances are discussed. A. Back Propagation Algorithm (BPA) BPA is one of the most studied and used algorithm for neural networks learning [10]. The BPA uses the Steepest Descent Method (SDM) to reach a global minimum. The SDM uses the error in the output layer of the network to update the weights of the network so as to reach the minimum of the objective function, which is defined to the summation of squared error between the desired outputs and the network outputs. The algorithm uses a learning parameter called The algorithm works on supervised learning. The number of iterations required for different values of for different range of synaptic weights for SDM, the number of iterations required for constant weights for SDM and the number of iterations required for different hidden nodes with one hidden layer for SDM were found. A comparison is made between the iterations required for one hidden layer and two hidden layers in SDM, the iterations required for the nodes in the hidden layer for different value of _ for SDM and the iterations required by the nodes in the hidden layer with and without _ in SDM were found. However, it would take enormous amount of time for the ANN to learn the patterns. Hence only 1000 patterns have been considered for training purpose. The dataset has been separated as training and testing (intrusion detection). Training indicates the formation of final weights which indicate a thorough learning of intrusion and normal packets along with corresponding labeling. The convergence rate of BPA is shown in Fig. 1. The classification performance of BPA is shown in Table 1. In Table 2, false acceptance rate and false rejection rates are shown.

ESNN [6], [7] possesses a highly interconnected and recurrent topology of nonlinear PEs that constitutes a "reservoir of rich dynamics" and contains information about the history of input and output patterns. The outputs of internal PEs (echo states) are fed to a memory less but adaptive readout network (generally linear) that produces the network output. The interesting property of ESNN shown in Fig. 2 is that only the memory less readout is trained, whereas the recurrent topology has fixed connection weights. This reduces the complexity of RNN training to simple linear regression while preserving a recurrent topology, but obviously places important constraints in the overall architecture that have not yet been fully studied. To train the ESNN, reservoirs and state matrix have to be used. The number of the iterations required for ESNN is lesser than the number of iterations required for SDM.

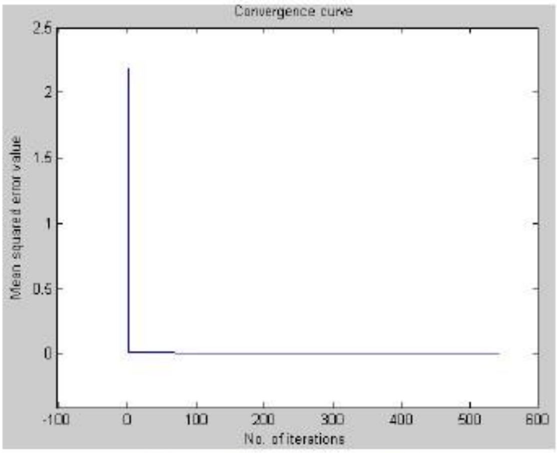


Fig. 1: Mean squared error curve

Table 1: Classification performance

Packet type	Total No. tested	No. classified	No. misclassified
Normal	363	360	3
Intrusion	637	600	37

Table 2: False acceptance / Rejection rate

Packet type	False Acceptance Rate (FAR)	False Rejection Rate (FRR)
Normal	5.8% (37/637)	0.8% (3/360)
Intrusion	10.1%(37/363)	0.4%(3/637)

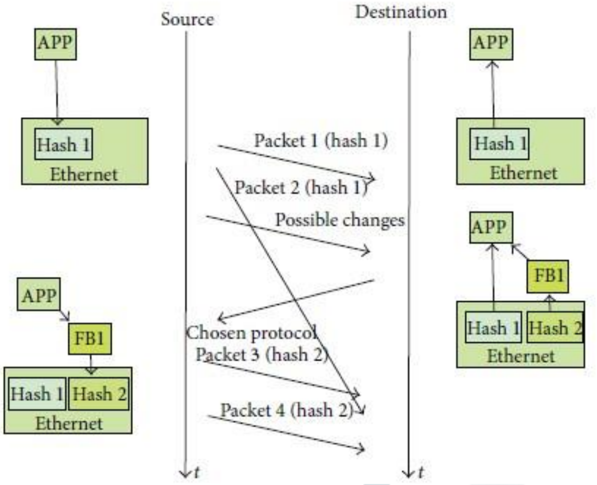
Internode Adaptation.

In the second part of the protocol stack setup phase, the local node negotiates a particular protocol stack with the destination node or nodes. First, the local node computes identifiers for all the protocol stacks computed in the first part and sends them to the destination node. The destination node then decides which protocol stack to use, sets up this protocol stack, and sends the identifier of the chosen stack back to the local node. If the local node never receives a reply from the destination, which could happen on a loss link, the source resends the configuration message and waits for the confirmation. After the completion of the negotiation phase, actual data transmission starts. It is important to note that all self-aware network nodes use the same method to compute the identifier of a given protocol stack. In order to distinguish between data messages and control messages (for the stack negotiation) a one byte header is introduced. There remains the problem of what protocol to choose for the protocol negotiation phase itself, for which we simply assume that all nodes in a given network segment use the Ethernet protocol. Similarly, if a connection to a node in another segment should be established, the intermediate nodes must use the same internetworking protocol, for example, IPv4 or IPv6.

Dynamic Hardware/Software Mapping.

In order to improve the performance of the system as compared to a software only system, functional blocks can be implemented in hardware accelerators. While for performance reasons it would be desirable to implement as many functional blocks in hardware as possible, we are constrained by the FPGA area and by the requirement to include novel protocols that are unknown at design time. Therefore, a system is required that dynamically decides which functional block should be implemented in hardware and which in software. Figure 3 shows our approach for self-aware hardware/ software mapping. It is a simplified version of the general self-aware node architecture described in Figure 1. The mapping algorithm obtains information from three different sources: *goals*, *sensors*, and *models*. The goals are specified by the user and might be "no packet loss,"

"minimize CPU load caused by network traffic," and so forth. The sensors collect statistical information such as "packets per second per flow" or "CPU load." The models describe the overall system and can either be known or learned at run time. Examples of models are "a packet is processed faster in hardware than in software," or "packets per second per flow does not change between two measurement intervals." Based on this input, the self-aware scheduler determines the hardware software mapping and also initiates the reconfiguration of the hardware, should that be required.



SCAN Xmas TREE

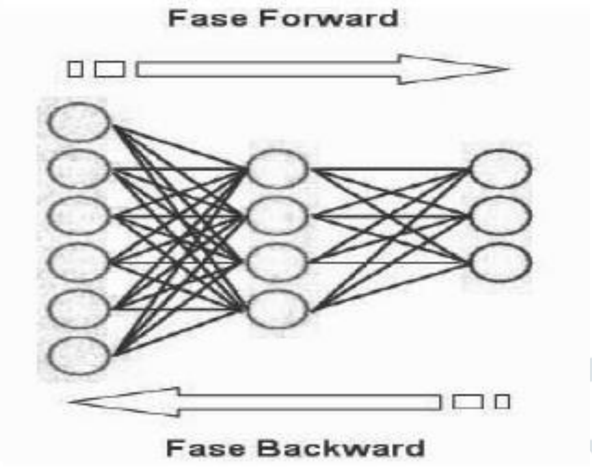
A common first step is recognizing the remote operating system. A basic approach to this endeavor uses a port scanner to send malformed packets to target machines. There are various types of scanning, including TCP Scanning, UDP Scanning, ACK Scanning, and e-Window

Scanning. According to the answers received, the attacker draws their own conclusion about open ports, kernel versions, operating systems, and other Information that is necessary to initiate an attack.

The possible answers include:

- Open or Accepted: The host sent a reply Indicating that a service is listening on the port. The ANN can be used in interconnected layers, and one of its more important properties is its intrinsic learning capacity. During supervised learning, examples of input-output pairs are presented to the network that adapts its internal weights to approximate the desired mapping. The Multi-Layer

Perceptron has at least one 'hidden layer' of neurons, whose outputs are kept isolated and not fed into other neurons. These hidden neurons do not have, in principle a 'desired output' from examples, but rather utilize a well-known optimization technique known as the backpropagation rule, which allows weight adjustments to occur. No reliable rules exist to determine the number and size of the hidden layers; this is usually accomplished by trial-and-error, with the user trying to find the smallest network that provides optimal performance. - Closed or Denied, or Not Listening: The host sent a reply indicating that connections will be denied to the port. - Filtered, Dropped or Blocked: There was no reply from the host. The Xmas Tree Scan was first used in 1999, when the computers could not handle packets with flags FIN, URG and PUSH sets which caused Windows Operating Systems to crash. Presentation of examples and weight correction measures through the back-propagation algorithms continue until a stop criterion is attained. Stop criterion usually employ a target value for the mean-square error over the training data. To train our neural network, the first task is to collect data. An important concern in this phase is to present unbiased data to the network. Data capture was done using dump, a packet decoder installed in most UNIX and Linux systems. The normalization of these data is necessary to input data to the ANN presented herein, and was accomplished with a dins that formats the fields shown in Table 1. During normalization, the data are collected and we try to input parameters as varied as possible so as not to incur the mistake of letting our network tendentious, and consequently harm the proof concept.



The data were arranged in two sets, as is usual in ANN training. These sets included the training set, containing 7,000 examples of traffic variables, both for normal traffic and during an attack. Each example is composed of seven variables (Table 1) and a desired output (0 for normal traffic, 1 for attack). A second set (the test set) is composed of 5,000 other traffic variable examples that will not be used to adapt the ANN weights, but only to evaluate its performance and generalization capacity. Initially, the archives were preprocessed for the backpropagation algorithm used in the MLANN. In this way, were created four archives in the Excel format:

- Input Matrix - Training: 7,000 x 7 dimension
- Input Matrix - Validation: 5,000 x 7 dimension

Each input matrix has a desired output (class), And the preparation of these matrices is described Hereinafter:

- Output Matrix - Training: the output layer will Consist of one node, which is responsible for warning of a network attack (output 1), or if the network traffic is normal (output 0). Thus, the output dimension will be 7,000 x 1.

Conclusion

The need for continuous improvements in device security motivates new alternatives for identifying computer network attacks. The use of ANNs for implementing network security is an attractive alternative to other common and less effective antiattack methods. Based on the signatures of Snort [2], we established malicious packages for assembly of the ANN training files. Our results indicated a 99% success rate for recognizing potential attack code. Commercial applications require supplementary input parameters, such as a package timestamp and package payload, among others. The Diversification of the examples is another factor to consider to improve the training of the ANN. Nonetheless, we consider our 99% success rate to be a highly promising framework for developing future ANNs against malicious traffic.

References

[1] Noticia20070704035010

[2] http://www.snort.org

[3] http://www.rfc.org

[4] B. Saha and A. Gairola. "Botnet: An Overivew". CERT-In White Paper, CIWP-2005-05, June 2005

[5] Haykin S. - "Redes Neurais Princípios e Práticas", Editora Bookman, 2001.

[6] Braga, A. P., Carvalho A. C. P. L. F. e udemir T. B. - "Redes Neurais Artificiais", Editora LTC, 2000.

[7] ROCHA, D. L. - "Utilização de um ambiente de honeynet no treinamento de redes neurais artificiais para detecção de intrusão", ENE-FT UnB, 2006.

[8] http://www.vmware.com

[9] http://www.insecure.org/nmap.

C. Hewitt, ORGs for scalable, robust, privacyfriendly client cloud computing, *IEEE Internet Computing*, vol.12, no.5, pp.96-99, 2008.

[2] R. C. Chen and C. M. Lin, Adaptive end-to-end delay equalizations for TCP virtual path transmissions in Internet environments, *International Journal of Innovative Computing, Information and Control*, vol.6, no.3(A), pp.1069-1078, 2010. [3] J. F. Lin, Performance analysis and discussion on a heuristic approach for scheduling multiprocessor tasks in a grid computing environment, *International Journal of Innovative Computing, Information and Control*, vol.6, no.12, pp.5451-5462, 2010.

[4] J. F. Lin, Scheduling parallel tasks with intra communication overhead in a grid computing environment, *International Journal of Innovative Computing, Information and Control*, vol.7, no.2, pp.881-896, 2011.

[5] RR Nadikattu, 2016 THE EMERGING ROLE OF ARTIFICIAL INTELLIGENCE IN MODERN SOCIETY. International Journal of Creative Research Thoughts. 4, 4 ,906-911.

[6] C. C. Jane, J. S. Lin and J. Yuan, Reliability evaluation of a limited-ow network in terms of minimal cutsets, *IEEE Transactions on Reliability*, vol.42, no.3, pp.354-361, 1993.

[7] Y. K. Lin, Reliability evaluation for an information network with node failure under cost constraint, *IEEE Transactions on Systems, Man and Cybernetics { Part A: Systems and Humans*, vol.37, no.2, pp.180-188, 2007.

[8] Sikender Mohsienuddin Mohammad, **"DEVOPS AUTOMATION AND AGILE METHODOLOGY "**, International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.5, Issue 3, pp.946-949, August-2017, Available at :http://www.ijcrt.org/papers/IJCRT1133441.pdf

[9] Y. K. Lin and C. T. Yeh, Evaluation of optimal network reliability under components assignments subject to a transmission budget, *IEEE Transactions on Reliability*, vol.59, no.3, pp.539550, 2010.

[10] Y. K. Lin and C. T. Yeh, Optimal resources assignment to maximize multistate network reliability, *Computers and Operations Research*, vol.37, no.12, pp.2229-2238, 2010.

[11] R.R. Nadikattu. 2017. ARTIFICIAL INTELLIGENCE IN CARDIAC MANAGEMENT. International Journal of

Creative Research Thoughts, Volume 5, Issue 3, 930-938.

[12] Y. K. Lin and C. T. Yeh, Optimal carrier selection based on network reliability criterion for stochastic logistics networks, *International Journal of Production Economics*, vol.128, no.2, pp.510-517, 2010.

[13] J. Xue, On multistate system analysis, *IEEE Transactions on Reliability*, vol.34, no.4, pp.329337, 1985.

[14] W. C. Yeh, Multistate network reliability evaluation under the maintenance cost constraint, *International Journal of Production Economics*, vol.88, no.1, pp.73-83, 2004.

[15] Sikender Mohsienuddin Mohammad, **"CONTINUOUS INTEGRATION AND AUTOMATION"**, International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.4, Issue 3, pp.938-945, July 2016, Available at :http://www.ijcrt.org/papers/IJCRT1133440.pdf

[16] L. D. Bodin, B. L. Golden, A. A. Assad and M. O. Ball, Routing and scheduling of vehicles and crews: The state of the art, *Computers and Operations Research*, vol.10, no.2, pp.63-212, 1983.

[17] Johnson, Neil F. And Sushil Jajodia. “Exploring steganography: seeing the unseen.” IEEE computer, 32:2. 26-34. 1998.

[18] Proves, N. And Honeyman, P. “Hide and Seek: An Introduction to steganography.”,IEEE security &privacy, (2003).

[19] Menezes, A., Van Oorschot, P., and Vanstone, S. “Handbook of applied cryptography.” CRC Press, (1996).

[20] Hassan Mathkour, Batool AL-sadoon, ameur touir ” a new image steganography technique”.

[21] Sim hiew moi, nazeema binti abdul rahim,puteh saad, pang li sim, zalmiyah zakaria,