Performance Analysis of Artificial Neural Network Intrusion Detection Systems

Mohamed Abdel-Azim[1], A.I. Abdel-Fatah[1] & Mohamed Awad[2]

[1]Faculty of Engineering, Mansoura University, Mansoura, Egypt: mazim12@yahoo.com, abdelfatah@yahoo.com
[2] SCADA & Telecommunication Dept. Head, GASCO, Egypt: m_awad_m@yahoo.com

Abstract

Intrusion detection is the art of detecting computer abuse and any attempt to break into networks. As a field of research, it must continuously change and evolve to keep up with new types of attacks or adversaries and the ever-changing environment of the Internet. To render networks more secure, intrusion-detection systems (IDSs) aim to recognize attacks within constraints of two major performance considerations: high detection and low false-alarm rates. It is also not enough to detect already-known intrusions, yet-unseen attacks or variations of those known present a real challenge in the design of these systems. IDSs are firmly entrenched on the security front but the exact role they can play and what their deployment entails must be clear to planners of security. Nine artificial neural networks (ANN) based IDS were implemented and tested with three experiments with three topologies. The results showed that: (i) in average the modular neural network (MNN) provided the best results in experiment#3; about 99.60%; (ii) in average multilayer perceptron (MLP) provided the best results in experiment#2; 74.71%; (iii) in experiment#1; the MNN provided the best results.

1. Introduction

Recently, the size of Internet and the volume of traffic have grown steadily. This expansion and increase in computerization generally have also seen a rise in computer misuse and attacks on networks. Prevention of such crime is impossible and so, monitoring and detection are resorted as the best alternative line of defense; the implementation of this process, called intrusion detection. It is performed with the aid of dedicated software and hardware systems operating on security logs, audit data or behavior observations. IDS also needs to process very large amounts of audit data and are mostly based on hand-crafted attack patterns developed by manual encoding of expert knowledge [1].

1.1. What is Intrusion Detection?

With the increase of attacks on computers and networks in recent years, improved and essentially automated surveillance has become a necessary addition to information technology (IT) security. Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions [1]. Intrusions are attempts to compromise the confidentiality, integrity and availability of a computer or network or to bypass its security mechanisms. They are caused by attackers accessing a system from the Internet, by authorized users of the systems who attempt to gain additional privileges for which they are not authorized, and by authorized users who misuse the privileges given to them.

1.2. Main Benefits and Characteristics

The main benefits of IDS include: (i) detecting attacks and other security violations, which have not been prevented by other primary protection techniques; (ii) preventing problem-behaviors by increasing the perceived risk of discovery and punishment for those who would attack or otherwise abuse the system; (iii) presenting traces of intrusions, allowing improved diagnosis, recovery and corrective measures after an attack; (iv) documenting the existing threat from inside and outside a system, permitting security management to realistically assess risk and adapt its security strategy in response, and (v) acting as quality control for security design and implementation (highlighting some deficiencies or errors, before serious incidents occur) [2].

Much work has been done to implement these features, so that now over 150 commercial, freeware and shareware IDS are available. To facilitate evaluation of these solutions, Purdue University IDS research project put a list of characteristics for good systems: (i) it must run continually without human supervision. The system must be reliable enough to allow it to run in the background of the system being observed. That is, its internal workings should be examinable from outside; (ii) it must be fault tolerant in the sense that it must survive a system crash and not lose its knowledge-base at restart; (iii) it must resist subversion; (iv) the system can monitor itself to ensure that it has not been subverted; (v) it must impose minimal overhead on the system; (vi) a system that slows a computer to a crawl will simply not be used; (vii) it must observe deviations from normal behavior; (viii) it must be easily tailored to the system in question. Every system has a different usage pattern, and the defense mechanism should adapt easily to these patterns, and (ix) it must cope with changing system behavior over time as new applications are being added. The system profile will change over time; it must be adaptable [3].

2. Overview of Detection Techniques

In general IDSs may be analyzed as misuse/anomaly detection and network-based/host-based systems.

2.1. Misuse Detection

Misuse detection depends on the prior representation of specific patterns for intrusions, allowing any matches to them in current activity to be reported. Patterns corresponding to known

attacks are called signatures, also giving rise to the term signature-based detection. These systems are unlike virus-detection systems; they can detect many known attack patterns and even variations; thereof but are likely to miss new attacks. Regular updates with previously unseen attack signatures are necessary [4].

2.2. Anomaly Detection

Anomaly detection identifies abnormal behavior. It requires the prior construction of profiles for normal behavior of users, hosts or networks; therefore, historical data are collected over a period of normal operation. IDSs monitor current event data and use a variety of measures to distinguish between abnormal and normal activities. These systems are prone to false alarms, since user's behavior may be inconsistent and threshold levels will remain difficult to fine tune. Maintenance of profiles is also a significant overhead but these systems are potentially able to detect novel attacks without specific knowledge of details. It is essential that normal data used for characterization are free from attacks [4].

2.3. Network-Based IDS Systems

Network-based IDS monitors traffic by capturing and analyzing network packets. Advantages of network-based IDSs are: (i) the deployment of these systems has little impact on the existing network; (ii) little effect on the normal network operation and are relatively easy to upgrade, and (iii) robust in the face of attacks and can be made invisible to attackers. On the other hand, the disadvantages are: (i) during peak-traffic periods some packets may go unprocessed and attacks undetected; (ii) encrypted information cannot be analyzed; (iii) attack attempts may be detected but hosts must usually then be investigated manually to determine whether or not they were penetrated and damage caused, and (iv) attacks involving fragmentation of packets can cause these IDS to crash [5].

2.4. Host-Based IDS Systems

Host-based IDS monitors network traffic of a particular host and some system events on the host itself. One may be installed on each host or simply on some chosen critical ones within a network. Advantages of host-based IDSs are: (i) some local events on hosts can only be detected; (ii) raw data are available for analysis in non-encrypted form, and (iii) software integrity checks can be used in the detection of certain types of attack (e.g. Trojan horse). In addition, it has the following disadvantages: (i) more complex to manage; (ii) may be disabled if host is attacked and compromised; (iii) not suitable for network attacks involving distributed scans and probes; (iv) can be disabled by overload attacks (e.g. denial of service); (v) for large amounts of information to be processed, local storage may be necessary, and (vi) use host's own computing resources at a cost to performance [5].

3. Performance Indices

Important measures of efficiency of IDSs are false-alarm rates; the percentage of time-consuming false positives registered-normal data detected falsely as an intrusion and the percentage of more dangerous false negatives; intrusions falsely classified as normal data. Such measurements do not indicate the human workload required in analyzing false alarms generated by normal background traffic. Low false-alarm rates combined with high detection rates mean; the detection outputs can be trusted [6].

4. Data Collection

The Defense Advanced Research Projects Agency (DARPA) intrusion-detection evaluation datasets were the original source of data most directly relevant to this work. For the 1998 DARPA datasets, 7-weeks (about 4 GBytes of compressed binary *tcpdump* data) of training data were accumulated from the multi-system testbed, to represent basically normal operation spiced with a series of automatically or manually launched attacks. Further 2-weeks of test data were collected containing additional new and novel intrusions [7].

The Knowledge Discovery and Data Mining (KDD) Cup 1999 are the datasets, which were issued for use in the KDD '99 Classifier-Learning Competition [8]. This was preprocessed with the feature-construction framework MADAM-ID, to produce about 5×10^6 connection records. A connection is defined to be a sequence of TCP packets starting and ending at some well-defined times, between which data flow to and from a source IP address to a destination IP address under some well-defined protocol. Each connection is labeled as either normal or with the name of its specific attack. A connection record consists of about 100 bytes. A 10% of the complementary 2-weeks of test data were, likewise, preprocessed to yield a further less than half-a-million connection records. It was stressed that these test data were not from the same probability distribution as the training data and that they included specific attack types not found in the training data. A total of 24 attack types were included in the training data [8, 9].

5. Attack Categorization

Simulated attacks were classified, according to actions and goals of the attacker. Each attack falls into one of the following: (i) *Denial-of-service (DoS)* have the goal of limiting/denying services provided to a user, computer or network; a common tactic is to severely overload the targeted system (e.g. SYN flood); (ii) *Probing* have the goal of gaining knowledge of existence or configuration of computer system or network; port scans/sweeping of a given IP-address range are typically used in this category (e.g. IPsweep); (iii) *Remote-to-Local (R2L)* have the goal of gaining local access to a computer or network to which the attacker previously only had remote access; e.g. attempts to gain control of a user account, and (iv) *User-to-Root (U2R)* have the goal of gaining root/super-user access on a particular computer/system on which the attacker previously had user level access; attempts by a non-privileged user to gain administrative privileges (e.g. Eject).

6. KDD Features

In the KDD'99 data [8], the initial features extracted for a connection record include the basic features of an individual TCP connection, such as: its duration, protocol type, number of bytes transferred and the flag indicating normal or error status of a connection. These intrinsic features provide information for general network-traffic analysis purposes. Since most DoS and Probe attacks involve sending a lot of connections to the host(s) at

the same time, they can have frequent sequential patterns, which are different to the normal traffic. For these patterns, same host feature examines all other connections in the previous 2-secs, which had the same destination as the current connection. Similarly, same service feature examines all other connections in the previous 2-secs, which had the same service as the current connection. Temporal and statistical characteristics are referred to as time-based traffic features; there are several Probe attacks which use a much longer interval than 2-secs (e.g., one minute) when scanning hosts or ports; mirror set of host-based traffic features were constructed based on a connection window of 100 connections. The R2L and U2R attacks are embedded in the data portions of the TCP packets and may involve only a single connection. To detect these, connection features of an individual connection were constructed using domain knowledge [10]. These features suggest whether the data contains suspicious behavior, such as: number of failed logins, successfully logged in or not, whether logged in as root, whether a root shell is obtained, etc. In general, there are 42 features (including the attack type) in each connection record, with most of them taking on continuous values.

7. IDS Classification Techniques

A brief description of ANN-based classifiers will be presented in the following subsections. ANNs are uniquely powerful tool in multiple class classification, especially when used in such applications where formal analysis would be very difficult. The accuracy however of such classifications depends on a variety of parameters, ranging from the architecture of the actual neural network to the training algorithm of choice [11-15].

7.1. Multilayer Perceptron (MLP)

MLP is a layered feed forward networks typically trained with static back propagation. 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) [12].

7.2. Generalized Feed-Forward (GFF)

GFF networks are a generalization of the MLP such that connections can jump over one or more layers. In theory, a MLP can solve any problem that a generalized feed-forward network can solve [12]. In practice, however, GFF networks often solve the problem much more efficiently. It suffices to say that a standard MLP requires hundreds of times more training epochs than the GFF network containing the same number of processing elements.

7.3. Modular Neural Network (MNN)

MNN is a special class of MLP; which processes inputs using several parallel MLPs, and then recombine the results. This tends to create some structures within the topology, which will foster specialization of function in each sub-module. In contrast to MLP, MNNs don't have full interconnectivity between their layers. Therefore, a smaller number of weights are required for the same size network; which tends to speed up training times and reduces 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. Four modular topologies were implemented in simulation; but the best one is tabulated [12].

7.4. Jordan/Elman network (JEN)

JEN extends MLP with context units, which are processing elements that remember past activity. Context units provide the network with the ability to extract temporal information from data. In JEN; the activity of the first hidden elements are copied to the context units. Networks which feed the input and the last hidden layer to the context units are also available. Four modular topologies were realized; but the best one is tabulated [13].

7.5. Principal Component Analysis (PCA) Network

PCA networks combine unsupervised and supervised learning in the same topology. PCA is an unsupervised linear procedure that finds a set of uncorrelated features principal components (PCs) from input data [14]. MLP is supervised network to perform nonlinear classification from these components. The number of PCs selected will be a compromise between training efficiency (few components) and accurate results (too many components).

7.6. Radial Basis Function (RBF) Networks

RBF networks are nonlinear hybrid networks typically containing a single hidden layer of processing elements. The RBF layer uses Gaussian transfer functions, rather than the standard sigmoidal functions. 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 MLP. For standard RBF's, the supervised segment of the network only needs to produce a linear combination of the output at the unsupervised layer [15].

7.7. Self Organized Maps (SOM)

SOM network transforms the input of arbitrary dimension into one or two dimensional discrete map. The feature maps are computed using Kohonen unsupervised learning. The SOM output can be used as input to a supervised network; the key advantage of SOM is the clustering produced; which reduces the input space into representative features using a self-organizing process [18].

7.8. Time Lagged Recurrent Networks (TLRNs)

TLRNs are MLPs extended with short term memory structures. Most real-world data contains information in its time structure. Yet, most ANNs are purely static classifiers. TLRNs are the state of the art in nonlinear timeseries prediction system identification and temporal pattern classification; focused topology includes the memory kernels connected to the input layer [13].

7.9. Recurrent Network (RN)

Fully RN feedbacks the hidden layer to itself. Partially recurrent networks start with a fully RN and add a feedforward

connection that bypasses recurrency effectively treating the recurrent part as a state memory. RNs can have an infinite memory depth and thus find relationships through time as well as through instantaneous input space [13].

8. Results

Attributes in KDD are continuous, discrete, and symbolic, with significantly varying resolution and ranges. Most classification methods are not able to process data in such a format; hence preprocessing was required. For training classifiers, duplicates were removed; the original 10% labeled training dataset is 494021 records reduced to 145587 records. Symbolic features like protocol-type (3-symbols), service (70-symbols), and flag (11-symbols) were mapped to integer values ranging from 0 to N-1; where N is the number of symbols. Attack names (e.g., Neptune, Prel, etc.) and normal labels were: (i) mapped to integer values ranging from 0 to 21 (22 attack names) and 22 for normal; (ii) mapped to 0 to 4 classes, '0' normal, '1' probe, '2' DoS, '3' U2R, and '4' R2L, and (iii) all attacks and normal are mapped to "attack" and "normal" respectively. 80% of the dataset used for training and 20% for testing. In the tables of results, of zeros classification rate; the corresponding category has been removed.

Three experiments were implemented for each ANN based classifier with three topologies. Experiments differ according to the number of outputs: (#1) 23 classes (22 attacks and normal); (#2) 5 classes (4 attacks and 1 normal), and (#3) 2 classes (attacks and normal). Topologies differ in the number of hidden layers: (1) one hidden layer (50 units); (2) two hidden layers (50×50), and (3) three hidden layers (28×23×23). It should be noted that the number of input features is fixed for all cases; 41 features.

Results of Experiment#1: Firstly, 23-classes IDS systems were implemented to classify each intrusion to one of the learned attack; results of this experiment were given in Tables.1 to 9.

- *Results of Experiment#2:* In this experiment we implemented 5-classes IDS to classify each intrusion as belonging to one of 4 known intrusion classes (probe, DoS, U2R, R2L) and normal. The results are given in Tables 10 to 18.
- *Results of Experiment#3:* Finally, we implemented a 2-classes IDS system, to classify all intrusions as belonging to attack class and normal class; results were given in; Tables 19 to 27.

9. Conclusion and Future Work

Nine types of ANN classifiers were developed to classify TCP data to recognize whether a system is under attack. Among all the classifiers tested, MNN for experiment#1 and exepriment#3 and MLP for experiment#2 delivered highly accurate results. The results showed that for all single ANN based classifiers: experiment#3 provided perfect detection results, while with experiment#1 and #2 the detection results were unsatisfactory. Therefore, these two experiments require hybrid multilayer ANN-based classifiers.

10. References

[1] R. Power, "2002 CSI/FBI computer crime and security survey," Computer Security Journal, Vol. XVIII, No.2, pp: 7-30, 20002.

[2] A. Delamer, "intrusion detection with data mining," Master's thesis, Donau University, Krems, Austria, May 2002.

[3] R. Bace and P. Mellm "NIST special publication on intrusion detection systems", Computer security resources, national institute of standards and technology, pp:1-51, 2002.

[4] T. Verwoed and R. Hunt, "intrusion detection techniques and approaches," Elsevier: computer communications, Vol.25, No.10, pp: 1356-1365, 2002.

[5] S. Chobrolu, A. Abraham, P. Johnson, "feature deduction and ensemble design of intrusion detection systems," Elsevier computers & security, Vol.24, pp: 195-307, 2005.

[6] A. Tartakovsky, et al., "detection intrusion in information system by sequential change-point methods," Elsevier, statistical methodology, Vol.3, pp: 252-293, 2006.

[7] R. Lippmann, et al., "1999-DARPA offline intrusion detection evaluation," Computer networks, Vol.34, pp: 579-595, 2000.

[8] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html/

[9] G. Mum, Y. Kim, et al., "network intrusion detection using statistical probability distribution," information systems & information technology, Vol. 3984, pp: 340-348, 2006.

[10] H. Kayacik, A. Zincir-Haywood, and M. Haywood, "selecting features for intrusion detection: a feature relevance analysis on KDD99 intrusion detection datasets," Dalhousie University, 2005.

[11] R. Beghdad, "critical study of neural networks in detecting intrusions" Computers & Security, Vol. 27, No. 5-6. Pp:168-175, 2008.

[12] Y. Yu, Y. Wei, et al., "anomaly intrusion detection approach using hybrid MLP/CNN neural network," intelligent systems design & applications. ISDA 6th int. conference, Vol.2, issue.16-18, pp: 1095-1102, October, 2006.

[13] J. Skaruz, "recurrent neural networks on duty of anomaly detection in databases," proceedings of 4th international symposium on neural networks: advances in neural networks part III, pp: 85-94, 2007.

[14] W. Wang and R. Battiti, "identifying intrusions in computer networks with principle component analysis," ARES2006 IEEE computer society, pp. 270-279.

[15] W. Dongli, Z. Yan, and H. Xiaoyang, "RBF neural network-based model predictive control for freeway traffic systems," International Journal of Intelligent Systems Technologies and Applications, Vol. 2, No.4, pp. 370 - 388, 2007.

[18] L. Vokorokos, A. Baláž, M. Chovanec, "intrusion detection system using self organizing map," Acta Electrotechnica et Informatica No. 1, Vol. 6, pp: 16, 2006

Table.1 Results of MLP using experiment #1

Classes	Topology-1	Topology-2	Topology-3
Normal	99.99%	99.99%	99.99%
Neptune	100.00%	99.99%	100.00%
Teardrop	99.41%	99.99%	2.30%
Portsweep	87.62%	71.32%	89.71%
Satan	90.12%	79.00%	00.00%
Warezclient	00.00%	29.82%	00.00%

Table.2 Results of GFF using experiment #1

Classes	Topology-1	Topology-2	Topology-3
Normal	99.87%	99.92%	99.94%
Neptune	99.99%	99.99%	100.00%
Teardrop	00.00%	99.27%	99.14%
Portsweep	00.00%	00.00%	83.42%
Satan	83.54%	00.00%	00.00%
Nmap	10.13%	00.00%	00.00%
Warezclient	53.53%	32.03%	7.39%

Table.3 Results of PCA using experiment #1

Classes	Topology-1	Topology-2	Topology-3
Normal	100.00%	100.00%	99.99%
Neptune	00.00%	85.81%	99.98%

Table.4 Results of RBF using experiment #1

Classes	Topology-1	Topology-2	Topology-3
Normal	99.94%	99.94%	99.92%
Neptune	100.00%	100.00%	100.00%

Table.5 Results of SOM using experiment #1

Classes	Topology-1	Topology-2	Topology-3
Normal	100.00%	100.00%	100.00%
Neptune	99.95%	99.96%	99.96%
Satan	29.28%	04.75%	00.00%

Table.6 Results of TLRN using experiment #1

Classes	Topology-1	Topology-2	Topology-3
Normal	99.94%	99.95%	99.97%
Neptune	99.93%	99.94%	99.96%
Smurf	93.64%	00.00%	00.00%
Teardrop	96.09%	69.19%	00.00%
Portsweep	00.00%	26.32%	00.00%
Satan	00.00%	76.24%	00.00%
Warezclient	00.00%	34.71%	00.00%

Table.7 Results of RN using experiment #1

Classes	Topology-1	Topology-2	Topology-3
Normal	99.92%	99.96%	99.99%
Neptune	99.98%	99.93%	99.98%
Portsweep	00.00%	45.00%	00.00%
Ipsweep	87.69%	10.04%	00.00%
Satan	00.00%	79.56%	00.50%

Table.8 Results of TLRN using experiment #1

Classes	Topology-1	Topology-2	Topology-3
Normal	99.96%	99.83%	99.93%
Neptune	99.99%	100.0%	99.98%
Smurf	79.82%	81.41%	89.25%
Teardrop	00.00%	99.27%	99.27%
Portsweep	86.32%	86.05%	18.16%
Ipsweep	00.00%	80.49%	78.98%
Imap	00.00%	00.00%	00.00%
Satan	00.00%	66.96%	00.00%
Warezclient	00.00%	90.03%	00.00%

Table.9 Results of JAN using experiment #1

Classes	Topology-1	Topology-2	Topology-3
Normal	99.91%	99.97%	99.97%
Neptune	99.99%	100.0%	99.99%
Smurf	98.46%	76.10%	00.00%
Pod	00.00%	00.00%	00.00%
Teardrop	00.00%	98.41%	85.09%
Portsweep	88.16%	79.21%	00.00%
Ipsweep	00.00%	78.03%	88.07%
Satan	82.10%	79.00%	00.00%
Warezclient	87.68%	31.69%	06.61%

Table.10 Results of MLP using experiment #2

Classes	Topology-1	Topology-2	Topology-3
Normal	99.95%	98.80%	99.77%
DoS	97.78%	98.86%	97.78%
PRB	92.03%	94.42%	92.03%
R2L	00.00%	79.46%	83.97%

Table.11 Results of GFF using experiment #2

Classes	Topology-1	Topology-2	Topology-3
Normal	99.97%	98.96%	99.79%
DoS	97.66%	98.22%	97.70%
PRB	58.09%	00.00%	00.00%
R2L	00.00%	00.00%	81.76%

Table.12 Results of PCA using experiment #2

Classes	Topology-1	Topology-2	Topology-3
Normal	99.92%	99.97%	100.00%
DoS	95.91%	95.84%	89.68%
PRB	00.00%	85.39%	00.00%
R2L	51.10%	00.00%	00.00%

Table.13 Results of RBF using experiment #2

Classes	Topology-1	Topology-2	Topology-3
Normal	99.97%	99.79%	99.55%
DoS	95.61%	95.52%	95.70%

Table.14 Results of SOM using experiment #2

Classes	Topology-1	Topology-2	Topology-3
Normal	100.00%	100.00%	100.00%
DoS	94.11%	94.11%	94.09%
PRB	38.91%	40.94%	40.49%

Table.15 Results of TLRN using experiment #2

Classes	Topology-1	Topology-2	Topology-3
Normal	99.88%	99.85%	99.93%
DoS	97.66%	97.27%	97.68%
PRB	02.33%	86.61%	55.00%
R2L	79.56%	00.00%	00.00%

Table.16 Results of RN using experiment #2

Classes	Topology-1	Topology-2	Topology-3
Normal	99.96%	99.97%	99.90%
DoS	97.69%	97.43%	97.70%
PRB	04.82%	90.31%	64.79%
R2L	00.00%	29.46%	58.92%

Table.17 Results of MNN using experiment #2

Classes	Topology-1	Topology-2	Topology-3
Normal	99.93%	99.79%	99.96%
DoS	97.48%	97.71%	96.18%
PRB	84.47%	81.53%	75.49%
R2L	00.10%	81.06%	00.00%

Table.18 Results of JAN using experiment #2

Classes	Topology-1	Topology-2	Topology-3
Normal	99.71%	99.64%	99.97%
DoS	00.00%	00.00%	05.71%
PRB	97.58%	99.67%	96.45%
R2L	80.46%	79.96%	29.26%

Table.19 Results of MLP using experiment #3

Classes	Topology-1	Topology-2	Topology-3
Normal	99.82%	99.85%	99.73%
Attack	98.12%	98.97%	98.43%

Table.20 Results of GFF using experiment #3

Classes	Topology-1	Topology-2	Topology-3
Normal	99.81%	99.71%	99.73%
Attack	98.11%	98.96%	98.00%

Table.21 Results of PCA using experiment #3

Classes	Topology-1	Topology-2	Topology-3
Normal	100.00%	99.43%	99.48%
Attack	00.00%	95.46%	94.88%

Table.22 Results of RBF using experiment #3

Classes	Topology-1	Topology-2	Topology-3
Normal	98.57%	99.27%	99.07%
Attack	95.52%	94.36%	93.27%

Table.23 Results of SOM using experiment #3

Classes	Topology-1	Topology-2	Topology-3
Normal	100.00%	100.00%	99.99%
Attack	89.47%	89.51%	89.63%

Table.24 Results of TLRN using experiment #3

Classes	Topology-1	Topology-2	Topology-3
Normal	99.77%%	99.79%	99.79%
Attack	97.09%	97.24%	96.55%

Table.25 Results of RN using experiment #3

Classes	Topology-1	Topology-2	Topology-3
Normal	99.28%	99.15%	99.59%
Attack	95.42%	98.17%	97.06%

Table.26 Results of MNN using experiment #3

Classes	Topology-1	Topology-2	Topology-3
Normal	99.50%	99.70%	99.90%
Attack	99.69%	97.48%	94.71%

Table.27 Results of JAN using experiment #3

Classes	Topology-1	Topology-2	Topology-3
Normal	99.31%	99.84%	99.68%
Attack	99.08%	98.36%	97.02%