International Journal of Computer Science and Security (IJCSS)

ISSN : 1985-1533

VOLUME 2, ISSUE 1

PUBLICATION FREQUENCY: 6 ISSUES PER YEAR

Editor in Chief Dr. Haralambos Mouratidis

International Journal of Computer Science and Security (IJCSS)

Book: 2008 Volume 2, Issue 1
Publishing Date: 31-02-2008
Proceedings
ISSN (Online): 1985-1553

IJCSS Journal is a part of CSC Publishers
http://www.cscjournals.org

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

CSC Publishers

Table of Contents

Volume 2, Issue 1, Febuary 2008.

Pages

1 -11 A Survey on MANET Intrusion Detection
Satria Mandala, Md. Asri Ngadi, A.Hanan Abdullah.

12 - 22 Implementation of Radial Basis Function Neural Network for Image Steganalysis
Sambasiva Rao Baragada, S. Ramakrishna, M.S. Rao, S. Purushothaman.

23 - 33 Segmentation of Malay Syllables in Connected Digit Speech Using Statistical Approach.
M-S Salam, Dzulkifli Mohamad, S-H Salleh.

34 - 48 GPS-less Localization Protocol for Underwater Acoustic Networks
A.K. Othman.

49 - 58 Detecting and preventing attacks using network intrusion detection systems
Meera Gandhi, S.K.Srivatsa.

59 - 72 RB-GDM: A Role-Based Grid Delegation Model.
G Geethakumari, Atul Negi, V N Sastry.

73 - 86 Implementation of Echostate Network in NIDS.
Meera Gandhi, S.K. Srivatsa.

A Survey on MANET Intrusion Detection

Satria Mandala satriamandala@hotmail.com
Faculty of Science & Technology
Department of Informatics Engineering
State Islamic University of Malang
Jl. Gajayana 50 Malang, Indonesia

Md. Asri Ngadi dr.asri@utm.my
Faculty of Computer Science & Information System,
Department of Computer System & Communication
Universiti Teknologi Malaysia (UTM)
Skudai - Johor, 81310, Malaysia

A. Hanan Abdullah hanan@utm.my
Professor, Faculty of Computer Science & Information System,
Department of Computer System & Communication
Universiti Teknologi Malaysia (UTM)
Skudai - Johor, 81310, Malaysia

Abstract

In recent years, the security issues on MANET have become one of the primary concerns. The MANET is more vulnerable to be attacked than wired network. These vulnerabilities are nature of the MANET structure that cannot be removed. As a result, attacks with malicious intent have been and will be devised *to exploit* these vulnerabilities and *to cripple* the MANET operation. Attack prevention measures, such as authentication and encryption, can be used as the first line of defense for reducing the possibilities of attacks. However, these techniques have a limitation on the effects of prevention techniques in general and they are designed for a set of known attacks. They are unlikely to prevent newer attacks that are designed for circumventing the existing security measures. For this reason, there is a need of second mechanism to "detect and response" these newer attacks, i.e. "*intrusion detection*". This paper aims *to explore* and *to classify* current techniques of Intrusion Detection System (IDS) aware MANET. To support these ideas, a discussion regarding attacks, IDS architectures, and researches achievement on MANET are presented inclusively, and then the comparison among several researches achievement will be evaluated based on these parameters. By this way, several existing security problems on MANET can be probed quickly for future researches.

Keywords: Intrusion Detection System (IDS), MANET, Survey, Wireless Ad hoc Network

1. INTRODUCTION

In MANET, a set of interacting nodes should cooperatively implement routing functions to enable end-to-end communication along dynamic paths composed by multi-hop wireless links. Several multi-hop routing protocols have been proposed for MANET, and most popular ones include:

Dynamic Source Routing (DSR) [1], Optimized Link-State Routing (OLSR) [2], Destination-Sequenced Distance-Vector (DSDV) [3] and Ad Hoc On-Demand Distance Vector (AODV) [4]. Most these protocols rely on the assumption of a trustworthy cooperation among all participating devices; unfortunately, this may not be a realistic assumption in real systems. Malicious nodes could exploit the weakness of MANET to launch various kinds of attacks.

Node mobility on MANET cannot be restricted. As results, many IDS solutions have been proposed for wired network, which they are defined on strategic points such as switches, gateways, and routers, can not be implemented on the MANET. *Thus, the wired network IDS characteristics must be modified prior to be implemented in the MANET*.

The rest of this paper will be structured as follows. Section 2 describes background of the IDS. The Intrusion detection on MANET is presented on section 3. In section 4, we present a discussion regarding the IDS classification. Finally, the conclusions and future research are shown in section 5.

2. IDS BACKGROUND

An intrusion-detection system (IDS) can be defined as the tools, methods, and resources to help identify, assess, and report unauthorized or unapproved network activity. Intrusion detection is typically one part of an overall protection system that is installed around a system or device—it is not a stand-alone protection measure.

Intrusion detection has a bit more history behind it. Endorf [5] stated that the intrusion detection was introduced as a formal research when James Anderson wrote a technical report [6] for the U.S. Air Force. Thus, it has been followed by Denning [7], Heberlein [8], and many researchers until present day.

Depending on the detection techniques used, IDS can be classified into three main categories [9] as follows: 1) signature or misuse based IDS), 2) anomaly based IDS, 3) specification based IDS, which it is a hybrid both of the signature and the anomaly based IDS.

- *The signature-based IDS* uses pre-known attack scenarios (or signatures) and compare them with incoming packets traffic. There are several approaches in the signature detection, which they differ in representation and matching algorithm employed to detect the intrusion patterns. The detection approaches, such as expert system [10], pattern recognition [11], colored petri nets [12], and state transition analysis [13] are grouped on the misuse.
- Meanwhile, *the anomaly-based IDS* attempts to detect activities that differ from the normal expected system behavior. This detection has several techniques, i.e.: statistics [14], neural networks [15], and other techniques such as immunology [16], data mining [[18[, [19]], and Chi-square test utilization [17]. Moreover, a good taxonomy of wired IDSes was presented by Debar [20].
- *The specification-based* IDS monitors current behavior of systems according to specifications that describe desired functionality for security-critical entities [48]. A mismatch between current behavior and the specifications will be reported as an attack.

3. MANET INTRUSION DETECTION

There are three focuses in this section: attacks, IDS architectures grouping, and researches achievement. The “researches achievement review” uses several parameters such as the IDS architectures, the detection techniques (see section 2), the resistance to several attacks type, and the routing protocols (see section 1).

3.1 ATTACKS

The MANET is susceptible to passive and active attacks [21]. The Passive attacks typically involve only eavesdropping of data, whereas the active attacks involve actions performed by adversaries such as replication, modification and deletion of exchanged data. In particular, attacks in MANET can cause congestion, propagate incorrect routing information, prevent services from working properly or shutdown them completely [[22],[25],[26],[23],[24],[27]].

Nodes that perform the active attacks are considered to be malicious, and referred to as *compromised*, while nodes that just drop the packets they receive with the aim of saving battery life are considered to be *selfish* [[28],[26]]. A selfish node affects the normal operation of the network by not participating in the routing protocols or by not forwarding packets. In addition, a compromised node may use *the routing protocol* to advertise itself as having the shortest path to the node whose packets it wants to intercept as in the so called *black hole* attack [[29], [30]].

Spoofing is a special case of integrity attacks whereby a compromised node impersonates a legitimate one due to the lack of authentication in the current ad hoc routing protocols [[35],[36]]. The main result of the spoofing attack is the misrepresentation of the network topology that may cause network loops or partitioning. Lack of integrity and authentication in routing protocols creates *fabrication attacks* [[37],[4],[38]] that result in erroneous and bogus routing messages.

Denial of service (DoS) is another type of attack, where the attacker injects a large amount of junk packets into the network. These packets overspend a significant portion of network resources, and introduce wireless channel contention and network contention in the MANET [[39],[40]]. A *routing table overflow attack* and *sleep deprivation attack* are two other types of the DoS attacks [41]. In the routing table *overflow attack*, an attacker attempts to create routes to non-existent nodes. Meanwhile the *sleep deprivation attack* aims to consume the batteries of a victim node.

There are also more sophisticated routing attacks. Compared to the simple attacks described above, these sophisticated attacks are much harder to detect and to prevent, i.e.: *wormhole attacks* (two compromised nodes create a tunnel that is linked through a private connection and thus they by-pass the network [[31],[32]]), *rushing attacks* [33] and *sybil attacks* [34].

3.2 IDS ARCHITECTURES

Based on the network infrastructures, the MANET can be configured to either flat or multi-layer. The optimal IDS architecture for the MANET may depend on the network infrastructure itself. There are four main architectures on the network [43], as follows: 1) Standalone IDS, 2) Distributed and Collaborative IDS, 3) Hierarchical IDS, and 4) Mobile Agent for Intrusion Detection Systems.

- *In the standalone architecture*, the IDS runs on each node to determine intrusions independently. There is no cooperation and no data exchanged among the IDSes on the network. This architecture is also more suitable for flat network infrastructure than for multi-layered network infrastructure

- *The distributed and collaborative architecture* has a rule that every node in the MANET must participate in intrusion detection and response by having an IDS agent running on them. The IDS agent is responsible for detecting and collecting local events and data to identify possible intrusions, as well as initiating a response independently.

- *The hierarchical architecture* is an extended version of the distributed and collaborative IDS architecture. This architecture proposes using multi-layered network infrastructures where the network is divided into clusters. The architecture has cluster heads, in some sense, act as control points which are similar to switches, routers, or gate ways in wired networks.

- *The mobile agent for IDS architecture* uses mobile agents to perform specific task on a nodes behalf the owner of the agents. This architecture allows the distribution of the intrusion detection tasks. There are several advantages using mobile agents [[21], [42]], for intrusion detection.

3.3 RESEARCHES ACHIEVEMENT

Many researchers have proposed several IDS especially for the MANET, some of them will be reviewed in the following paragraph.

Since the nature of MANET node is *distributed* and *requires cooperation* to other nodes, ***Zhang, Lee, and Huang [[30], [24]]*** proposed "intrusion detection (ID) and response system" should follow both the natures. In this proposed architecture model, each node is responsible for detecting signs of intrusion locally and independently, but neighboring nodes can collaboratively investigate in a broader range. Individual IDS agents are placed on each and every node. Each the IDS agent runs independently and monitors local activities (user and systems activities, and communication activities within the radio range). The agent detects intrusion from local traces and initiates response. If anomaly is detected in the local data, or if the evidence is inconclusive and a broader search is warranted, neighboring IDS agents will cooperatively participate in global intrusion detection actions. These individual IDS agents collectively form the IDS system to defend the wireless ad-hoc network.

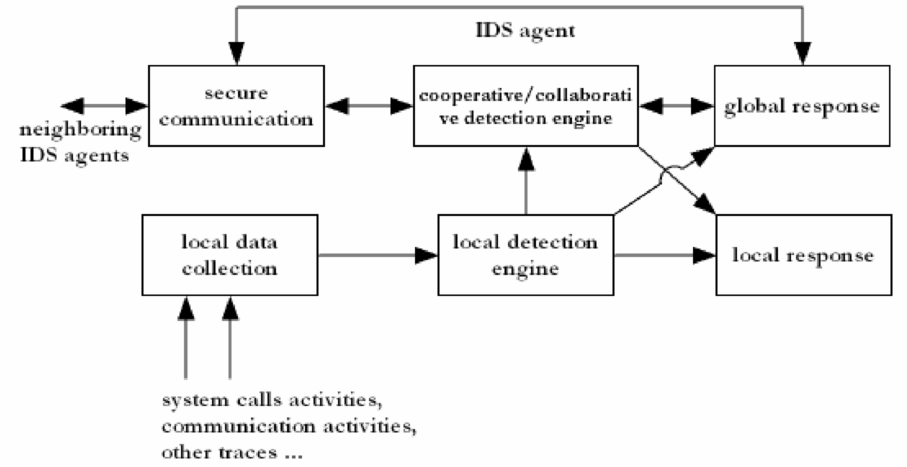


FIGURE 1: IDS agent model

Albers et al. [44] proposed a distributed and collaborative architecture of IDS by *using mobile agents*. A Local Intrusion Detection System (LIDS) is implemented on every node for local concern, which can be extended for global concern by cooperating with other LIDS. Two types of data are exchanged among LIDS: security data (to obtain complementary information from collaborating nodes) and intrusion alerts (to inform others of locally detected intrusion). In order to analyze the possible intrusion, data must be obtained from what the LIDS detects on, along with additional information from other nodes. Other LIDS might be run on different operating systems or use data from different activities such as system, application, or network activities; therefore, the format of this raw data might be different, which makes it hard for LIDS to analyze. However, such difficulties can be solved by using Simple Network Management Protocol (SNMP) data located in Management Information Base (MIBs) as an audit data source. Such a data source not only eliminates those difficulties, but also reduces the increase in using additional resources to collect audit data if an SNMP agent is already run on each node. For the methodology of detection, Local IDS Agent can use either anomaly or misuse detection. However, the combination of two mechanisms will offer the better model. Once the local intrusion is detected, the LIDS initiates a response and informs the other nodes in the network. Upon receiving an alert, the LIDS can protect itself against the intrusion.

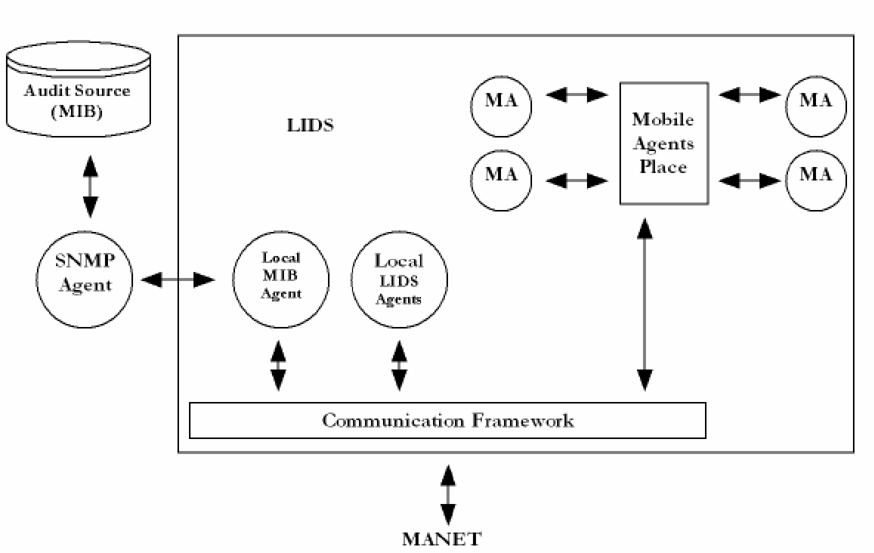


FIGURE 2: LIDS Architecture in a Mobile Node

Kachirski and Guha [45] proposed a multi-sensor intrusion detection system based on mobile agent technology. The system can be divided into three main modules, each of which represents a mobile agent with certain functionality, i.e.: monitoring, decision-making and initiating a response.

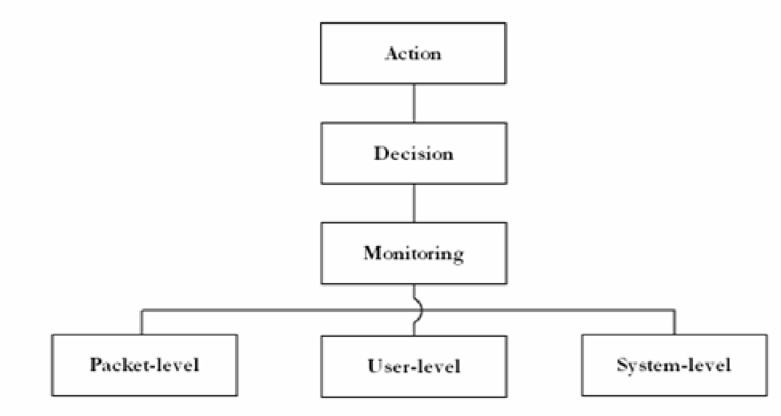


FIGURE 3: Layered Mobile Agent Architecture

- Monitoring agent: Two functions are carried out at this class of agent: network monitoring and host monitoring.

- Action agent: Every node also hosts this action agent. The action agent can initiate a response, such as terminating the process or blocking the node from the network, if it meets intrusion activities where it lives.

- Decision agent: The decision agent is run only on certain nodes, mostly at the nodes that run network monitoring agents. If the local detection agent cannot make a decision on its own due to insufficient evidence of an intrusion, it will report to this decision agent in order to investigate deeply on the suspected node

Since nodes move arbitrarily across the network, a static hierarchy is not suitable for such dynamic network topology.

Sterne et al. [46] proposed a dynamic intrusion detection hierarchy that is potentially scalable to large networks use clustering. This method is similar with Kachirski and Guha [45], but it can be structured in more than two levels. Thus, nodes on first level are cluster heads, while nodes on the second level are *leaf nodes*. In this model, every node has the task to monitor, log, analyze, respond, and alert or report to cluster heads. The Cluster heads, in addition, must also perform: 1) Data fusion/integration and data filtering, 2) Computations of intrusion, and 3) Security Management.

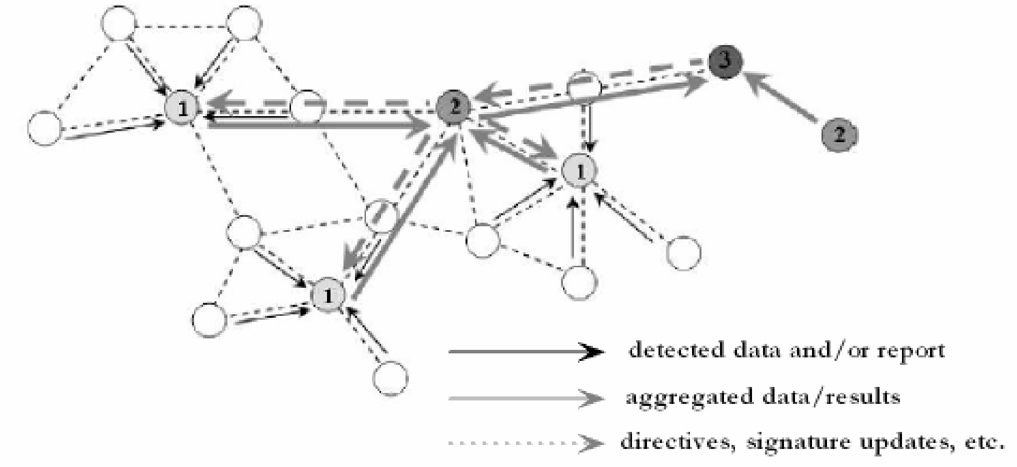

1
2
3
2
1
1
detected data and/or report
aggregated data/results
directives, signature updates, etc.

FIGURE 4: Dynamic Intrusion Detection Hierarchy

B.Sun [47] proposed Zone Based IDS (ZBIDS). In the system, the MANET is spitted into non-overlapping zones (zone A to zone I). The nodes can be categorized into two types: the intra-zone node and the inter-zone node (or a gateway node). Each node has an IDS agent run on it. This agent is similar to the IDS agent proposed by Zhang and Lee. Others components on the system are data collection module and detection engine, local aggregation and correlation (LACE) and global aggregation and correlation (GACE). The data collection and the detection engine are responsible for collecting local audit data (for instance, system call activities, and system log files) and analyzing collected data for any sign of intrusion respectively. The remainder, LACE module is responsible for combining the results of these local detection engines and generating alerts if any abnormal behavior is detected. These alerts are broadcasted to other nodes within the same zone. However, for the GACE, its functionality depends on the type of the node. If the node is an intra-zone node, it only sends the generated alerts to the inter-zone nodes. Thus, if the node is an inter-zone node, it receives alerts from other intra-zone nodes, aggregates and correlates those alerts with its own alerts, and then generates alarms. The intrusion response module is responsible for handling the alarms generated from the GACE.

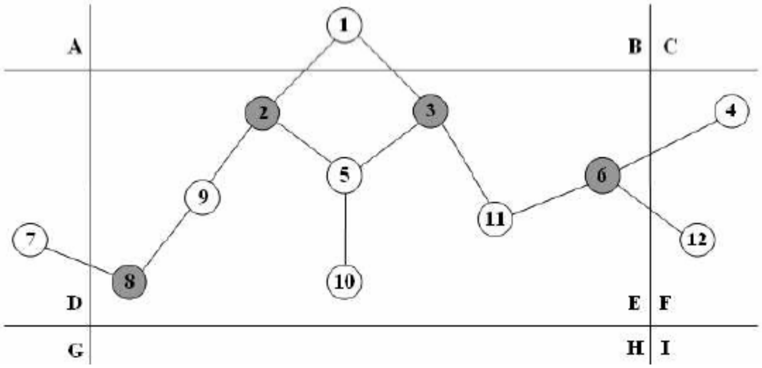


FIGURE 5a: ZBIDS for MANETs

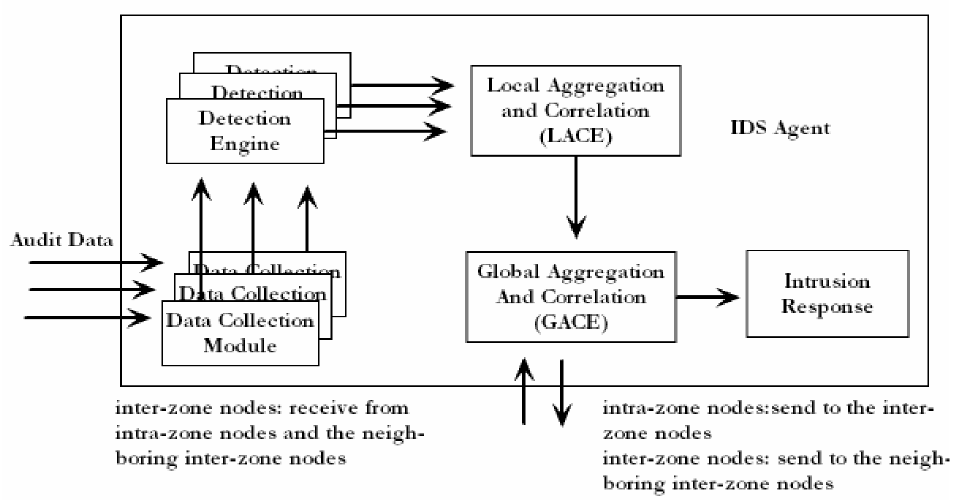

Detection
Detection
Detection Engine
Local Aggregation and Correlation (LACE)
IDS Agent
Audit Data
Data Collection
Data Collection
Data Collection Module
Global Aggregation And Correlation (GACE)
Intrusion Response
inter-zone nodes: receive from intra-zone nodes and the neighboring inter-zone nodes
intra-zone nodes:send to the inter-zone nodes
inter-zone nodes: send to the neighboring inter-zone nodes

FIGURE 5b: An IDS agent in ZBIDS

4. DISCUSSION AND SUMMARY

The classification among the proposed IDS of MANET can be composed using the parameters discussed in the previous sections, i.e.: *architecture, attacks,* and *IDS detection techniques*. Most the MANET IDSes tend to have the distributed architectures and their variants. The IDS architecture may depend on the network infrastructure (see section 3.2). But the most important thing is the reasons the architecture to be configured in distributed manner. As the nature of MANET is so open, attacks source can be generated from any nodes within the MANET itself or nodes of neighboring networks. Unfortunately, this network lacks in central administration. It is difficult for implementing firewall or the IDS on the strategic points. Moreover, each node can work as client, server or router. Delivery packets need collaboration work among the nodes participant network. For these reasons, the IDS of MANET should have characteristics that follow these natures, *distributed and collaborative*. Zhang [30], Albers [44], and Sun [47] follow this idea. Meanwhile, Kachirski [45] and Sterne [46] use the variant of the distributed and collaborative. Advantage using distributed architecture is the security accident can be detected earlier. However, this architecture needs huge resources, which is difficult to be implemented in small wireless device as PDA.

All attacks type of wired networks is possible in MANET. MANET has also several typical of attacks, which are not available in the traditional wired network, such as selfish attack, black hole attack, sleep deprivation attack and others type of attacks (see section 3.1). These attacks occur because of MANET has vulnerable in the *use of wireless link, auto-configuration mechanisms, and its routing protocol*. The existing MANET IDSes have various methods to detect and to response regarding these attacks. Zhang [30] and Sun [47] proposed the IDSes which were designed for detecting the intrusion activities on the routing protocol of MANET. Albers [44] tried to extend the traditional IDS on MANET to detect incoming telnet connections and reacted if they originated from outside community's network. Sterne [46] presented a cooperative and distributed IDS that covered conventional attacks. Table 1 shows the summary of the classification of these MANET IDS.

Author(s)	Name Specific	Architecture	Addressed Attacks type			Data Source	Technique detection	Routing protocol	Environments	Contribution
			Authentication	Routing (black hole, etc)	Selfish					
Zhang and Lee, Y. Huang [30], [24]	None	Distributed and collaborative	No	Yes (misrouting, packet dropping)	No	Audit trail (event log processing)	Anomaly	AODV, DSR, DSDV	Simulation	IDS agent for collaboration detection
P. Albers, O. Camp [44]	LIDS	Distributed and collaborative	No	No	No	Audit trail (event log processing)	Misuse, anomaly	Not identified	Simulation	Local IDS mobile agent for intrusion detection model
Kachirski and Guha [45]	None	Hierarchical architecture	No	No	No	Audit trail (event log processing)	Anomaly	Not identified	Simulation	Hierarchical IDS using mobile agent
Sterne et al. [46]	None	Hierarchical architecture	No	No	No	Audit trail (event log processing)	Misuse, Anomaly	Not identified	Simulation	Dynamic intrusion detection hierarchy model
B. Sun, K.Wu, and U. W. Pooch [47]	ZBIDS	Distributed and collaborative	No	Yes (Disruption attacks)	No	Audit trail (event log processing)	Anomaly	DSR	Simulation	Routing protocol protection from disruption

TABLE 1: Comparison researches achievement on the MANET IDS.

5. CONSLUSION & FUTURE WORK

With the nature of mobile ad hoc networks, almost all of the intrusion detection systems (IDSs) are structured to be distributed and have a cooperative architecture (see table 1). Refer to the table 1, mostly the proposed research prefers using anomaly detection approach. An intrusion detection system aims to detect attacks on mobile nodes or intrusions into the networks. However, attackers may try to attack the IDS system itself. Accordingly, the study of the defense to such attacks should be explored as well.

6. ACKNOLEDGEMENTS

Authors would like to thank to the Ministry of Science, Technology and Innovation of Malaysia (the MOSTI) and Research Management Centre of UTM (RMC-UTM) – This work has been sponsored by the e-science Fund, Vot number 79027.

7. REFERENCES

1. D.B. Johnson, D.A. Maltz, et.al. *“The dynamic Source Routing Protocol for Mobile Ad hoc Networks (DSR)”*. Internet Draft, draft-ietf-manet-dsr-07.txt, work in progress, 2002

2. T. Clausen, P. Jaquet, et.al. *“Optimized link state routing protocol”*. Internet Draft, draft-ietf-manet-olsr-06.txt, work in progress, 2001

3. C.E. Perkins, P. Bhagwat. *"Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers"*. SIGCOMM 94 Conference on Communications Architectures, Protocols and Applications, 1994

4. C.E Perkins, E. Belding-Royer. "*Ad hoc On-demand Distance Vector (AODV)"*, Request For Comments (RFC) 3561, 2003

5. C. Endorf, E. Schultz and J. Mellander, *"Intrusion Detection & Prevention"*, McGraw-Hill, ISBN: 0072229543 (2004)

6. J. P. Anderson. *"Computer Security Threat Monitoring and Surveillance"*. Technical Report, James P. Anderson Co., Fort Washington, PA, 1980

7. D.E. Denning, *"An Intrusion-Detection Model"*. IEEE Transactions on Software Engineering, pp. 222- 231, 1987

8. L. Heberlein, G. Dias, et.al. *"A network security monitor"*. In Proceedings of the IEEE Symposium on Security and Privacy, pp. 296-304, 1990

9. A. Hijazi and N. Nasser. *"Using Mobile Agents for Intrusion Detection in Wireless Ad Hoc Networks"*. In Wireless and Optical Communications Networks (WOCN), 2005

10. T. F. Lunt, R. Jagannathan, et al. *"IDES: The Enhanced Prototype C a Realtime Intrusion-Detection Expert System"*. Technical Report SRI-CSL-88-12, SRI International, Menlo Park, CA, 1988

11. M. Esposito, C. Mazzariello, et.al. *"Evaluating Pattern Recognition Techniques in Intrusion Detection Systems"*. The 7th International Workshop on Pattern Recognition in Information Systems, pp. 144-153, 2005

12. S. Kumar and E. Spafford, "*A Pattern Matching Model for Misuse Intrusion Detection"*. The 17th National Computer Security Conference, pp. 11-21, 1994

13. P.A. Porras and R. Kemmerer, *"Penetration State Transition Analysis C a Rule-Based Intrusion Detection Approach"*. The 8th Annual Computer Security Application Conference, pp. 220-229, 1992

14. P. Porras and A. Valdes, *"Live Traffic Analysis of TCP/IP Gateways"*. ISOC Symposium on Network and Distributed System Security, San Diego, CA, 1998

15. H. Debar, M. Becker and D. Siboni. *"A Neural Network Component for an Intrusion Detection System"*. Proceedings of IEEE Symposium on Research in Security and Privacy, Oakland, CA, pp. 240-250, 1992

16. S. Forrest, S.A. Hofmeyr, and A. Somayaji*. "Computer Immunology".* Communications of the ACM, pp. 88-96, 1997

17. N. Ye, X. Li, et.al. *"Probabilistic Techniques for Intrusion Detection Based on Computer Audit Data"*. IEEE Transactions on Systems, Man, and Cybernetics, pp. 266-274, 2001

18. W. Lee, S.J. Stolfo, K.W. Mok. *"A Data Mining Framework for Building Intrusion Detection Models"*. IEEE Symposium on Security and Privacy (Oakland, California), 1999

19. G. Florez, S.M. Bridges, and R.B. Vaughn, *"An Improved Algorithm for Fuzzy Data Mining for Intrusion Detection"*. The North American Fuzzy Information Processing Society Conference, New Orleans, LA, 2002

20. H. Debar, M. Dacier, and A.Wespi, *“A Revised Taxonomy for Intrusion-Detection Systems”*. Annales des Telecommunications, pp. 361-378, 2000

21. A.J. Menezes, S.A. Vanstone, P.C. Van Oorschot, *“Handbook of Applied Cryptography”.* CRC Press, Inc., USA (2001)

22. A. Mishra, K. Nadkarni, and A. Patcha. *“Intrusion Detection in Wireless Ad Hoc Networks”.* IEEE Wireless Communications, Vol. 11, Issue 1, pp. 48-60, 2004

23. L. Zhou and Z. J. Haas. *“Securing ad hoc networks”*. IEEE Network Magazine , 1999

24. Y. Zhang, W. Lee, and Y. Huang. *“Intrusion Detection Techniques for Mobile Wireless Networks”.* Wireless Networks Journal (ACM WINET), 9(5): 545-556, 2003.

25. E.C.H. Ngai, M.R. Lyu, R.T. Chin. *“An authentication service against dishonest users in mobile ad hoc networks”*, IEEE Proceedings on Aerospace Conference, vol. 2, pp. 1275–1285 2004.

26. L. Blazevic et al. *“Self-organization in mobile ad-hoc networks: the approach of terminodes”*, IEEE Communications Magazine , pp. 166–173, 2001

27. W. Zhang, R. Rao, et. al. *“Secure routing in ad hoc networks and a related intrusion detection problem”*, IEEE Military Communications Conference (MILCOM), vol. 2, 13–16 p. 735– 740, 2003

28. J. Kong et al. *“Adaptive security for multi-layer ad-hoc networks”*. Special Issue of Wireless Communications and Mobile Computing, John Wiley Inter Science Press (2002)

29. P. Kyasanur, N. Vaidya. *“Detection and handling of MAC layer misbehavior in wireless networks”*. International Conference on Dependable Systems and Networks. pp. 173–182, 2003

30. Y. Zhang, W. Lee, *“Intrusion detection in wireless ad-hoc networks”*, The 6th Annual International Conference on Mobile Computing and Networking, pp. 275–283, 2000

31. Y. Hu, A. Perrig, and D. Johnson. *“Packet leashes: A defense against wormhole attacks in wireless ad hoc networks”*. In Proceedings of IEEE INFOCOM’03, 2003

32. Y. Hu, A. Perrig, D. Johnson, *“Ariadne: a secure on-demand routing protocol for ad hoc networks”*. ACM MOBICOM, 2002

33. Y. Hu, A. Perrig, and D. Johnson. *“Rushing attacks and defense in wireless ad hoc network routing protocols”*. In Proceedings of ACM MobiCom Workshop - WiSe’03, 2003

34. J. R. Douceur. *“The sybil attack”*. The 1st International Workshop on Peer-to-Peer Systems pp. 251–260, 2002.

35. J. Hubaux, L. Buttya´n, S. Capkun, “*The quest for security in mobile ad hoc networks.”* The 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2001

36. P. Papadimitratos, Z.J. Haas, E.G. Sirer, *“Path set selection in mobile ad hoc networks”*, The Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 1–11, 2002

37. B. DeCleene et al. *“Secure group communications for wireless networks”.* IEEE Military Communications Conference, 2001.

38. S. Bo, W. Kui, U.W. Pooch. *“Towards adaptive intrusion detection in mobile ad hoc networks”*. IEEE Global Telecommunications Conference, pp. 3551–3555, 2004

39. C. Douligeris, A. Mitrokosta, *“DDoS attacks and defense mechanisms: classification and state-of-the-art”*. Computer Networks: The International Journal of Computer and Telecommunications Networking 44 (5):643–666, 2004

40. C.M. Chlamtac, J.J.-N. Liu, *“Mobile ad hoc networking: imperatives and challenges”*, Ad Hoc Networks 1, 2003

41. H. Yang, H.Y. Luo, et.al. *“Security in Mobile Ad Hoc networks: challenges and solutions”.* IEEE Wireless Communications, pp.38–47, 2004.

42. C. Krugel and T. Toth. *“Applying mobile agent technology to intrusion detection”.* In ICSE Workshop on Software Engineering and Mobility, 2001.

43. T. Anantvalee and J. Wu. *“A Survey on Intrusion Detection in Mobile Ad Hoc Networks”*, Book Series Wireless Network Security, Springer, pp. 170 – 196, ISBN: 978-0-387-28040-0 (2007)

44. P. Albers, O. Camp, et al. *“Security in Ad Hoc Networks: a General Intrusion Detection Architecture Enhancing Trust Based Approaches”.* Proceedings of the 1st International Workshop on Wireless Information Systems (WIS-2002), pp. 1-12, April 2002

45. O. Kachirski, R. Guha*. “Effective Intrusion Detection Using Multiple Sensors in Wireless Ad Hoc Networks.”* Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS’03), IEEE, 2003

46. D. Sterne, P. Balasubramanyam, et al. *“A General Cooperative Intrusion Detection Architecture for MANETs”*. In Proceedings of the 3rd IEEE International Workshop on Information Assurance (IWIA'05), pp. 57-70, 2005

47. B. Sun, K.Wu, and U. W. Pooch. “*Alert Aggregation in Mobile Ad Hoc Networks”*. The 2003 ACM Workshop on Wireless Security in conjuction with the 9th Annual International Conference on Mobile Computing and Networking (MobiCom'03), pp. 69-78, 2003

48. C. Ko, J. Rowe, P. Brutch, K. Levitt, *“System Health and Intrusion Monitoring Using a hierarchy of Constraints”.* In Proceedings of 4th International Symposium, RAID, 2001

Implementation of Radial Basis Function Neural Network for Image Steganalysis

Sambasiva Rao Baragada mrsambasivarao@yahoo.com
Department of Mathematics & Computer Science
Sri Venkateswara University
Tirupati, 517502, India

S. Ramakrishna drsramakrishna@yahoo.com
Department of Mathematics & Computer Science
Sri Venkateswara University
Tirupati, 517502, India

M. S. Rao cfs_dfs@dfs.gov.in
Director cum Chief Forensic Scientist
Directorate of Forensic Science
Ministry of Home Affairs
New Delhi, 110003, India

S. Purushothaman dr.s.purushothaman@gmail.com
Faculty of Engineering
Asia-Pacific Institute of Information Technology
KaulaLumpur, 57000, Malaysia

Abstract

Steganographic tools and techniques are becoming more potential and widespread. Illegal use of steganography poses serious challenges to the law enforcement agencies. Limited work has been carried out on supervised steganalysis using neural network as a classifier. We present a combined method of identifying the presence of covert information in a carrier image using fisher's linear discriminant (FLD) function followed by the radial basis function (RBF). Experiments show promising results when compared to the existing supervised steganalysis methods, but arranging the retrieved information is still a challenging problem.

Keywords: Steganography, carrier image, covert image**.**

1. INTRODUCTION

Steganography is a type of hidden communication that literally means "covered writing". The message is out in the open, often for all to see, but goes undetected because the very existence of the message is secret [12, 20, 21]. Steganalysis could be described as a method to prevent steganography. There are other attacks on steganography. Attacking the end hosts of the steganography algorithm by searching for security credentials is not steganalysis. Digital forensics encompasses more methods than solely steganalysis to attack steganography. The target for digital forensics is detection of steganography. The objective of steganalysis is “detecting messages hidden using steganography”. Steganalysis is about separating cover-messages from stego-messages. In this work, passive steganalysis is focused.

Most of the present literature on steganalysis follows either a parametric model [28, 24, 26] or a blind model [32, 27, 22, 23, 35, 33]. A general steganalysis method that can attack steganography blindly, detect hidden data without knowing embedding methods, will be more useful in practical applications. A framework for steganalysis based on supervised learning has been done in [34]. The framework was further developed and tested. Limited work has been carried out on supervised steganalysis, using neural networks as a classifier [29, 30]. Fishers’ linear discriminant function (FLD) as a classifier show impressive results in [31]. The present neural network based steganalytic work is implemented by combining the radial basis function neural network with fishers’ linear discriminant function.

2. METHODOLOGY

Machine learning theory based steganalysis assume no statistical information about the stego image, host image and the secret message. This work falls under the category of supervised learning employing two phase strategies: a) training phase and b) testing phase. In training phase, original carriers are supplied to neural classifier to learn the nature of the images. RBF takes the role of neural classifier in this work. By training the classifier for a specific embedding algorithm a reasonably accurate detection can be achieved. RBF neural classifier in this work learns a model by averaging over the multiple examples which include both stego and non-stego images. In testing phase, unknown images are supplied to the trained classifier to decide whether secret information is present or not. The flowcharts of both the phases are given below in figure 1:

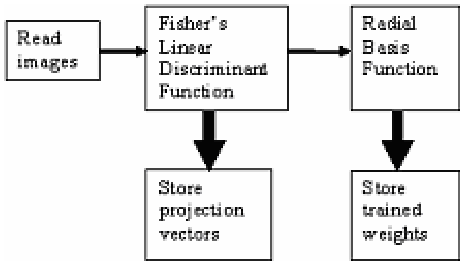


FIGURE 1a: Training Phase

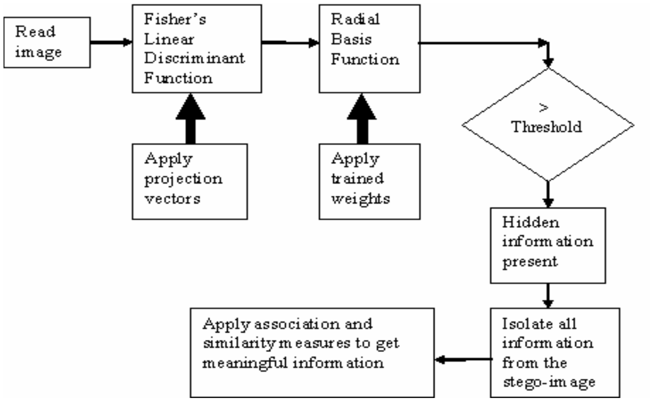

Read image
Fisher's Linear Discriminant Function
Radial Basis Function
> Threshold
Apply projection vectors
Apply trained weights
Hidden information present
Isolate all information from the stego-image
Apply association and similarity measures to get meaningful information

FIGURE 1b: Testing phase

2.1 Fisher’s Linear Discriminant Function

The process of changing the dimensions of a vector is called transformation. The transformation of a set of n-dimensional real vectors onto a plane is called a mapping operation. The result of this operation is a planar display. The main advantage of the planar display is that the distribution of the original patterns of higher dimensions (more than two dimensions) can be seen on a two dimensional graph. The mapping operation can be linear or non-linear. R.A. Fisher developed a linear classification algorithm [1] and a method for constructing a classifier on the optimal discriminant plane, with minimum distance criterion for multi-class classification with small number of patterns [16]. The method of considering the number of patterns and feature size [4], and the relations between discriminant analysis and multilayer perceptrons [17] has been addressed earlier. A linear mapping is used to map an n-dimensional vector space $\Re^n$ onto a two dimensional space. Some of the linear mapping algorithms are principal component mapping [5], generalized declustering mapping [2, 3, 8, 9], least squared error mapping [11] and projection pursuit mapping [6]. In this work, the generalized declustering optimal discriminant plane is used. The mapping of the original pattern 'X' onto a new vector 'Y' on a plane is done by a matrix transformation, which is given by

$$Y=AX \tag{1}$$

where

$$A = \begin{bmatrix} \varphi_1 \\ \varphi_2 \end{bmatrix} \tag{2}$$

and φ1 and φ2 are the discriminant vectors (also called projection vectors).

An overview of different mapping techniques [14, 15] is addressed earlier. The vectors φ1 and φ2 are obtained by optimizing a given criterion. The plane formed by the discriminant vectors is the optimal vectors which are the optimal discriminant planes. This plane gives the highest possible classification for the new patterns.

The steps involved in the linear mappings are:

Step 1: Computation of the discriminant vectors φ1 and φ2: this is specific for a particular linear mapping algorithm.

Step 2: Computation of the planar images of the original data points: this is for all linear mapping algorithms.

1) Computation of discriminant vectors φ_1 and φ_2

The criterion to evaluate the classification performance is given by:

$$J(\varphi) = \frac{\varphi^{T} S_b \varphi}{\varphi^{T} S_W \varphi} \tag{3}$$

Where

S_b the between class matrix, and

S_w the within class matrix which is non-singular.

$$S_b = \sum p(\omega_i)(m_i - m_o)(m_i - m_o)^T \tag{4}$$

$$S_w = \sum p(\omega_i) E\left[X_i - m_o)(X_i - m_i)^T \omega_i\right] \tag{5}$$

where

$P(\omega_i)$ a priori the probability of the i^{th} pattern, generally, $p(\omega_i) = 1/m$

m_i the mean of each feature of the i^{th} class patterns, (i=1.2…,m),

m_o the global mean of a feature of all the patterns in all the classes,

X {xi, l=1, 2,…L} the n-dimensional patterns of each class,

L the total number of patterns.

Eq.(3) states that the distance between the class centers should be maximum. The discriminant vector φ_1 that maximizes 'J' in Eq. (3) is found as a solution of the eigenvalue problem given by:

$$S_b \varphi_1 = \lambda_{ml} S_w \varphi_1 \tag{6}$$

where

λ_{ml} the greatest non-zero eigenvalue of $(S_b S_w^{-1})$

φ_1 eigenvalue corresponding to λ_{ml}

The reason for choosing the eigenvector with maximum eigenvalue is that the Euclidean distance of this vector will be the maximum, when compared with that of the other eigenvectors of Eq.(6). Another discriminant vector φ_2 is obtained, by using the same criterion of Eq.(3). The discriminant vector φ_2 should also satisfy the condition given by:

$$\varphi^T{}_2\,\varphi_1=0 \tag{7}$$

Eq.(7) indicates that the solution obtained is geometrically independent and the vectors φ_1 and φ_2 are perpendicular to each other. Whenever the patterns are perpendicular to each other, it means, that there is absolutely no redundancy, or repetition of a pattern. The discriminant vector φ_2 is found as a solution of the eigenvalue problem, which is given by:

$$Q_p\ S_b\ \varphi_2 = \lambda_{m2}\ S_w\ \varphi_2 \tag{8}$$

where
λ_{m2} the greatest non-zero eigen value of $Q_p\ S_b\ S_w^{-1}$, and
Q_p the projection matrix which is given by

$$Q_p \quad = \quad I - \frac{\varphi_1\ \varphi_1^T\ S_W^{-1}}{\varphi_1^T\ S_W^{-1}\ \varphi_1} \tag{9}$$

where
I an identity matrix

The eigenvector corresponding to the maximum eigenvalue of Eq. (8) is the discriminant vector φ_2. In Eq.(6) and Eq. (8), S_W should be non-singular. The S_W matrix should be non-singular, even for a more general discriminating analysis and multi-orthonormal vectors [7, 18, 19]. If the determinant of S_W is zero, then singular value decomposition (SVD) on S_W has to be done. On using SVD [10, 13], S_W is decomposed into three matrices U, W and V. The matrices U and W are unitary matrices, and V is a diagonal matrix with non-negative diagonal elements arranged in the decreasing order. A small value of 10^{-5} to 10^{-8} is to be added to the diagonal elements of V matrix, whose value is zero. This process is called perturbation. After perturbing the V matrix, the matrix S_w^1 is calculated by:

$$S_w^{\ 1} = U * W * V^T \tag{10}$$

where
S_W^1 the non-singular matrix which has to be considered in the place of S_w.
Minimum perturbed value should be considered, which is just sufficient to make S_w^1 non-singular. As per Eq.(7), when the values of φ_1 and φ_2 are innerproducted, the resultant value should be zero. In reality, the innerproducted value will not be zero. This is due to floating point operations.

2) Computation of two-dimensional vector from the original n-dimensional input patterns

The two-dimensional vector set y_i is obtained by:

$$y_i = (u_i\ ,\ v_i\ ) = (X_i^T\ \varphi_1\ ,\ X_i^T\ \varphi_2) \tag{11}$$

The vector set y_i is obtained by projecting the original pattern 'X' onto the space, spanned by φ_1 and φ_2 by using Eq.(11). The values of u_i and v_i can be plotted in a two-dimensional graph, to know the distribution of the original patterns.

2.2 Radial Basis Function

A radial basis function (RBF) is a real-valued function whose value depends only on the distance from the origin. If a function ‘h’ satisfies the property $h(\boldsymbol{x})=h(||\boldsymbol{x}||)$, then it is a radial function. Their characteristic feature is that their response decreases (or increases) monotonically with distance from a central point. The centre, the distance scale, and the precise shape of the radial function are parameters of the model, all fixed if it is linear [25].

A typical radial function is the Gaussian which, in the case of a scalar input, is

$$h(x)=exp((-(x-c)^2)/(r^2)) \tag{12}$$

Its parameters are its centre *c* and its radius *r*.

A Gaussian RBF monotonically decreases with distance from the centre. In contrast, a multiquadric RBF which, in the case of scalar input, monotonically increases with distance from the centre. Gaussian-like RBFs are local (give a significant response only in a neighbourhood near the centre) and are more commonly used than multiquadric-type RBFs which have a global response. Radial functions are simply a class of functions. In principle, they could be employed in any sort of model (linear or nonlinear) and any sort of network (single-layer or multi-layer). RBF networks have traditionally been associated with radial functions in a single-layer network. In the Figure 2, the input layer carries the outputs of FLD function. The distance between these values and centre values are found and summed to form linear combination before the neurons of the hidden layer. These neurons are said to contain the radial basis function with exponential form. The outputs of the RBF activation function is further processed according to specific requirements.

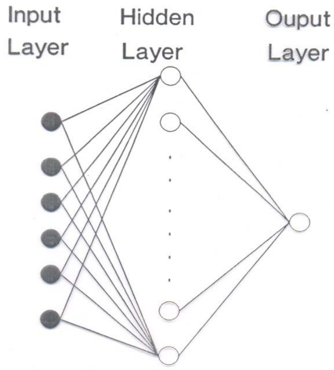


FIGURE 2: Radial Basis Function Network

3. IMPLEMENTATION

a) Training

1. Decide number of cover images.
2. Read each Image.
3. Calculate the principal component vector by

$$Z=Z * Z^{T}$$

where

Z denotes the intensities of image

4. Find eigenvector of the Z matrix by applying eigen process.
5. Calculate the φ_1 and φ_2 vectors.

φ_1 = eigenvector ($S_b * S_w^{-1}$)

$S_b = \sum (PCV_i – M_0) (PCV_i – M_0)^T / N$

where:

PCV_i (i = 1,2,3)

```
            $PCV_{1,}$ Principal component vector1
            $PCV_{2,}$ Principal component vector2
            $PCV_{3,}$ Principal component vector3
              $M_0$ = Average of ($PCV_1 + PCV_2 + PCV_3$)
               $S_w = (\sum ( PCV_i - M_i ) ( PCV_i - M_i )^T ) / N$
    where:
           $M_{i\,(}\, i = 1, 2, 3)$
            $M_{1,}$ average of $PCV_1$
           $M_{2,}$ average of $PCV_2$
                     $M_{3,}$ average of $PCV_3$
     6. Calculate $\varphi_2$ vector.
              $\varphi_2$ = eigenvector ($Q\, S_b\, S_w^{-1}$ )
               $Q = I - ((\varphi_1 * \varphi_1^{-1} * S_w^{-1}) / (\varphi_1^t * S_w^{-1} * Phi\_\varphi_1) )$
     7. Transfer for N dimensional vector into 2 dimensional vector.
                $U = \varphi_1 * PCV_{i\ (1,2,3)}$
               $V = \varphi_2 * PCV_{i\ (1,2,3)}$
     8. Apply RBF.
               No. of Input = 2
               No. of Patterns = 15
               No. of Centre = 2
            Calculate RBF as
              RBF = exp (-X)
           Calculate Matrix as
             G = RBF
             $A = G^T * G$
          Calculate
             $B = A^{-1}$
          Calculate
             $E = B * G^T$
     9. Calculate the final weight.
             F = E * D
     10.  Store the final weights in a File.
```

b) Testing

```
     1. Read steganographed image.
     2. Calculate the principal component vector.
        $Z=Z * Z^T$
     3.  Find eigenvector of the Z matrix by applying eigen process.
     4. Calculate RBF as.
             RBF = exp ( -X )
            G = RBF
            $A = G^T * G$
           $B = A^{-1}$
           $E = B * G^T$
     5. Calculate.
           F = E * D
     6. Classify the pixel as containing information or not.
```

4. RESULTS AND DISCUSSION

The simulation of steganalysis has been implemented using MATLAB 7®. Sample sets of images considered are gray and true color images. The different sets of cover images considered in the simulation are given in Figure 3. The information image is given in Figure 4. Encryption technique has not been considered during the simulation. The different ways the secret information scattered in the cover images are given in Figure 5.

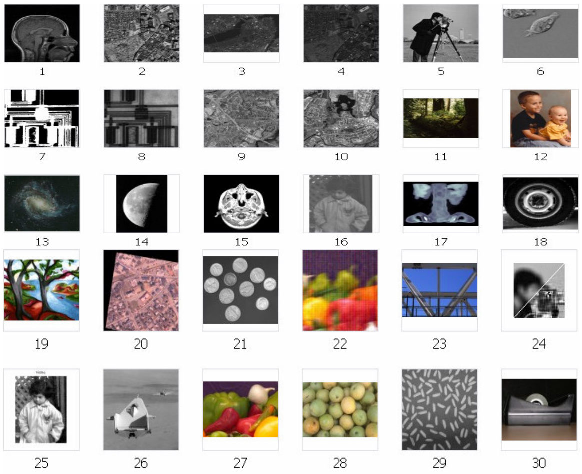


FIGURE 3: Cover images

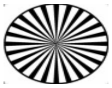


FIGURE 4: Information Image

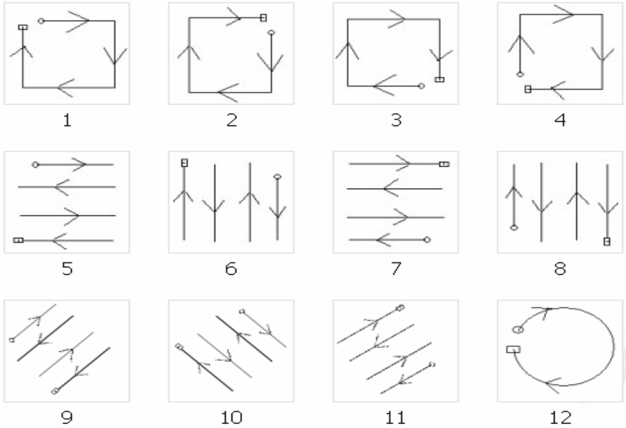
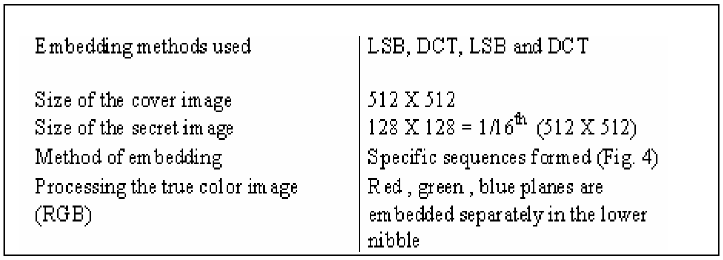


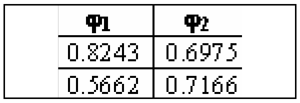
FIGURE 5: Distribution of information image in cover image

In this simulation, the information is embedded using least significant bit (LSB), discrete cosine transformation (DCT) separately. In certain cases, 50% of the information image is embedded using LSB and remaining 50% of the information is embedded using DCT. In the entire simulation, the size of the information image is considered to 1/8 size of the original image (Table I). The outputs of the FLD (Figure 6), RBF (Figure 7), and combined method FLD with RBF (Figure 8) are shown. The projection vectors are given in Table II.



Embedding methods used	LSB, DCT, LSB and DCT
Size of the cover image	512 X 512
Size of the secret image	128 X 128 = 1/16th (512 X 512)
Method of embedding	Specific sequences formed (Fig. 4)
Processing the true color image (RGB)	Red , green , blue planes are embedded separately in the lower nibble

Table I: SIMULATION ENVIRONMENT USED



φ1	φ2
0.8243	0.6975
0.5662	0.7166

Table II: PROJECTION VECTORS

These vectors obtained after finding out the S_w and S_b matrices considering 30 steganographed images using the images given in Figure 3 and Figure 4. Figure 7 and figure 8 are obtained by setting a detection threshold value of 2. Any output greater than a threshold is considered as the pixel containing the information. The threshold value is different for different method.

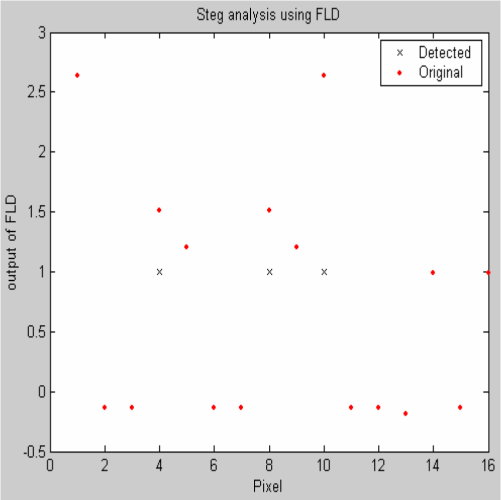


FIGURE 6: Steganalysis using FLD

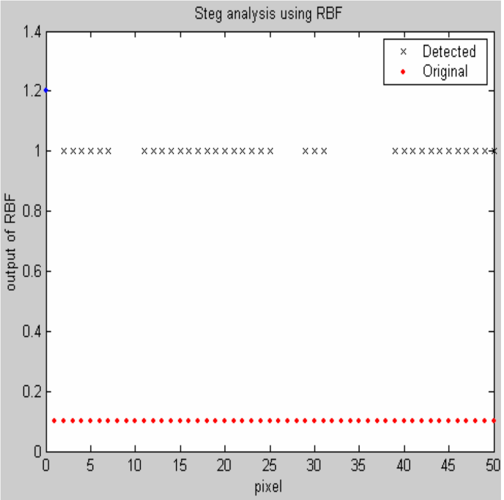


FIGURE 7: Steganalysis using RBF

5. CONCLUSION

Steganalysis has been implemented using FLD, RBF and combination of FLD and RBF algorithms. The outputs of the algorithms for one steganographed image have been presented. Secret information is getting retrieved by the proposed algorithms with various degrees of accuracies. It can be noticed that the combined method FLDRBF is much promising in detecting the presence of hidden information. The cover images chosen for the simulation are standard images. The percentage of identifying the hidden information is more than 95%, but arranging the retrieved information is still a challenging problem. The information can be well arranged in a meaningful way by using a set of association rules.

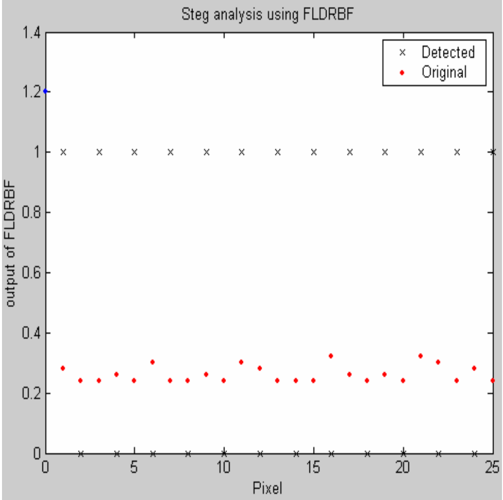


FIGURE 8: Steganalysis using FLDRBF

6. ACNOWLEDGEMENT

We express our sincere thanks to Mr. Ch Gandhi, GEQD, Hyderabad for his valuable support in improving the quality of the paper. We also express our heartful thanks to Mr. P. Krishna Sastry and Mr. M. Krishna for their technical support in doing experiments at computer forensic division, Hyderabad. Appreciation also goes to Ms. S. Rajeswari, Ms. G. Sarada, Ms. Uma Devi, and Ms. Chennamma and for their help in literature survey

7. REFERENCES

1. R.A.Fisher, "*The Use of Multiple Measurements in Taxonomic Problems*," Ann. of Eugenics, vol. 7, pp. 178 – 188, 1936.
2. J.W.Sammon, "*An Optimal Discriminant Plane*," IEEE Trans. On Comp, vol. 19, no. 9, pp. 826 – 829, 1970.
3. J.W Sammon., "*Interactive Pattern Analysis and Classification"*, *IEEE* Trans. on Comp., vol. 19, no. 7, pp. 594–616, 1970.
4. D.H.Foley, "*Consideration of Sample and Feature Size*," IEEE Trans., on Info. Theory, vol.18, no. 5, pp. 626 – 681, September 1972.
5. J.Kittler, P.C.Young P.C, "*Approach to Feature Selection Based on the Karhunen-Loeve Expansion*," Pattern Recognition, Vol. 5, No 5, pp. 335 – 352, 1973.
6. H.Friedman, J.W Turkey, "*A Projection Pursuit Algorithm for Exploratory Data Analysis*," IEEE Trans. on Comp., vol. 23, no. 9, pp. 881–890, 1974.

7. D.H Foley, J.E.Sammon, “*An Optimal Set of Discriminant Vectors*,” IEEE Trans. on Comp., vol. 24, no. 3, pp. 281 – 289, 1975.
8. J. Fehlauer., B.A Eisenstein., “*A Declustering Criterion for Feature Extraction in Pattern Recognition*, IEEE Trans. on Comp., vol. 27, no. 3, pp. 261 – 266, 1978.
9. E.Gelsema, R.Eden, “*Mapping Algorithms in ISPAHAN*,” pattern Recognition, vol. 12, no. 3, pp.127 – 136, 1980.
10. V.C.Klema, A.J.Laub, “*The Singular Value Decomposition: Its Computation and Some Applications*,” IEEE Trans. on Automatic Control, vol. 25, no.2, pp. 164 – 176, 1980.
11. D.F.Mix, R.A.Jones, “*A Dimensionality Reduction Techniques Based on a Least Squared Error Criterion*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 4, no. 1, pp. 537 – 544, 1982.
12. Simmons, G. J., “*The Prisoners’ Problem and the Subliminal Channel*,” CRYPTO83, Advances in Cryptology, August 22-24, pp. 51 – 67, 1984.
13. B.J.Sullivan, B.Liu, “*On the Use of Singular Value Decomposition and Decimation in Discrete-Time Band-Limited Signal Extrapolation*,” IEEE Trans. on Acoustics, Speech and Signal Processing, vol. 32, no. 6, pp. 1201 – 1212, 1984.
14. W.Siedlecki, K.Siedlecka, J. Skalansky, “*An Overview of Mapping Techniques for Exploratory Data Analysis*,” Pattern Recognition, vol. 21, no. 5, pp. 411 – 429, 1988.
15. W.Siedlecki, K.Siedlecka., J.Skalancky, “*Experiments on Mapping Techniques for Exploratory Pattern Analysis*,” Pattern Recognition, vol. 21, no.5, pp. 431 – 438, 1988.
16. Z.Q.Hong, Y.J.Yang, *“Optimal Discriminate Plane for a Small Number of Samples and Design Method of Classifier on the Plane*,” pattern recognition, vol. 24, pp. 317 – 324, 1991.
17. P.Gallinari, S. Thiria., F.Badran., F Fogelman-Soulie F., “*On The Relations Between Discriminant Analysis and Multilayer Perceptrons*,” Neural Networks, vol. 4, no.3, pp.349 – 360, 1991.
18. K.Liu, Y.Q.Cheng, J.Y.Yang J.Y., “*A Generalized Optimal Set of Discriminant Vectors*,” Pattern Recognition, vol. 25, no. 7, pp. 731 – 739, 1992.
19. Y.Q.Cheng, Y.M.Zhuang, J.Y.Yang, “*Optimal Fisher Discriminant Analysis Using the Rank Decomposition*,” Pattern Recognition, vol. 25, no. 1, pp. 101 – 111, 1992.
20. R. J. Anderson and F.A.P Petitcolas, “*On the limits of steganography*,” IEEE Journal on Selected Areas in Communications, vol. 16 no. 4, pp. 474 -484, 1998.
21. N.Johnson and J. Sklansky, “*Exploring steganography: seeing the unseen*,” IEEE Computer, pp. 26 – 34, 1998.
22. N. Provos and P. Honeyman, “*Detecting steganographic content on the internet*,” CITI Technical Report 01-11, August, 2001.
23. A. Westfeld and A. Pfitzmann, “*Attacks on steganographic systems*,” Third Information Hiding Workshop, September. 1999.
24. J. Fridrich, R. Du, and M. Long, “*Steganalysis of lsb encoding in color images*,” IEEE ICME, vol. 3, pp. 1279 – 1282, March 2000.
25. Meng Joo Er,.Shiqian Wu, Juwei Lu and Hock Lye Toh, “*Face Recognition with Radial Basis Function(RBF) Neural Networks*,” IEEE Trans. on Neural Networks, vol. 13, no.3, pp. 697 – 910, May 2002.
26. R. Chandramouli, “*A mathematical framework for active steganalysis*,” ACM Multimedia Systems, vol, 9, no.3, pp.303 – 311, September 2003.
27. J. Harmsen and W. Pearlman, “*Steganalysis of additive noise modelable information hiding*,” Proc. SPIE Electronic Imaging, 2003.
28. S. Lyu and H. Farid, “*Steganalysis Using Color Wavelet Statistics and One-Class Support Vector Machines*,” SPIE Symposium on Electronic Imaging, San Jose, CA, 2004.
29. Shi, Y.Q, Guorong Xuan, Zou, D, Jianjiong Gao, Chengyun Yang, Zhenping Zhang, Peiqi Chai, Chen, W, Chen C, “*Image steganalysis based on moments of characteristic functions using wavelet decomposition, prediction-error image, and neural network*,” IEEE International Conference on Multimedia and Expo, ICME , July 2005.
30. Ryan Benton and Henry Chu**, “***Soft Computing Approach to Steganalysis of LSB Embedding*

in Digital Images," Third International Conference on Information Technology: Research and Education, ITRE, pp. 105 – 109, June 2005.

31. Ming Jiang, Wong, E.K. Memon, N. Xiaolin Wu, "*Steganalysis of halftone images*," Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP, vol. 2, pp. 793 – 796, March 2005.
32. Shalin Trivedi and R. Chandramouli*,* "Secret *Key Estimation in Sequential Steganography*," *IEEE Trans. on Signal Proc*., vol. 53, no. 2, pp. 746 - 757, February 2005.
33. Liang Sun, Chong-Zhao Han, Ning Dai, Jian-Jing Shen, "*Feature Selection Based on Bhattacharyya Distance: A Generalized Rough Set Method*," Sixth World Congress on Intelligent Control and Automation, WCICA, vol. 2, pp. 101-105, June, 2006.
34. H. Farid, "*Detecting hidden messages using higher-order statistical models*," Proc. IEEE Int. Conf. Image Processing, New York, pp.905- 908, Sep. 2002.
35. Zugen Liu, Xuezeng Pan, Lie Shi, Jimin Wang, Lingdi Ping, "*Effective steganalysis based on statistical moments of differential characteristic function*," International Conference on Computational Intelligence and Security, vol. 2, pp. 1195 – 1198, November 2006.

Segmentation of Malay Syllables in Connected Digit Speech Using Statistical Approach

M-S Salam sah@utm.my
Faculty of Computer Science & Information System
University Technology Malaysia
Skudai, 81310 Johor Bahru, Malaysia

Dzulkifli Mohamad dzul@utm.my
Faculty of Computer Science & Information System
University Technology Malaysia
Skudai, 81310 Johor Bahru, Malaysia

S-H Salleh hussain@fke.utm.my
Faculty of Biomedical Engineering & Health Science
University Technology Malaysia
Skudai, 81310 Johor Bahru, Malaysia

Abstract

This study present segmentation of syllables in Malay connected digit speech. Segmentation was done in time domain signal using statistical approaches namely the Brandt's Generalized Likelihood Ratio (GLR) algorithm and Divergence algorithm. These approaches basically detect abrupt changes of energy signal in order to determine the segmentation points. Patterns used in this experiment are connected digits of 11 speakers spoken in read mode in lab environment and spontaneous mode in classroom environment. The aim of this experiment is to get close match between reference points and automatic segmentation points. Experiments were conducted to see the effect of number of the auto regressive model order *p* and sliding window length *L* in Brandt's algorithm and Divergence algorithm in giving better match of the segmentation points. This paper reports the finding of segmentation experiment using four criterions ie. the insertion, omissions, accuracy and segmentation match between the algorithms. The result shows that divergence algorithm performed only slightly better and has opposite effect of the testing parameter *p* and *L* compared to Brandt's GLR. Read mode in comparison to spontaneous mode has better match and less omission but less accuracy and more insertion.

Keywords: Speech Segmentation, Divergence Algorithm, Brandt's Algorithm

1. INTRODUCTION

Malay language is an agglutinative language. It is a language of derivative, which allows addition of prefix and suffix to the base word to form new word(s) [1]. Most of Malay words can be considered as consist of combination of syllables where syllables can be comprised of a vowel, or a vowel with a consonant or a vowel with several consonants [2] Several studies and experiments show that syllable unit size is remarkably salient and may exhibit specific acoustic characteristic [3]. Being able to segment the syllables correctly will make recognition a much easy work. Previous experiment on isolated Malay digit syllables where the segmentation was done manually reach recognition up to 80% [4]. However, automatic syllable segmentation from connected digit is a taunting task as syllables signal in connected speech is highly complex with no fixed property and significant acoustic cues exists in between syllables.

Time domain segmentations with non-fixed overlapping segment window size proved to give a good segmentation result with less omission [5]. Among these non-fixed overlapping segment window size segmentation, two algorithms usually applied are the Brandt's GLR algorithm and the Divergence algorithm. Brandt's GLR algorithm and divergence algorithm detect segment points by identifying discontinuities of speech signal without any further knowledge upon the phonetic sequence [6]. In another words they are linguistically unconstrained and are therefore expected to make insertions and omissions. Nevertheless, an “ideal” Brandt's GLR which disregards omission and insertions yields better word segmentation accuracy compare to HMM in experiment done in [6]. On the other hand, experiment on segmentation of music found that divergence algorithm performed better than Brandt's algorithm [7]. Similar conclusion is yielded for experiment on word in continuous speech in [8].

With respect to the foregoing, this paper report works in syllable segmentation from a sequence of Malay connected digits speech signal using both Brandt's GLR and divergence algorithms with the objective to find the best match between automatic segmentation and reference segmentation points. The aim is to apply the points from automatic segmentation of syllables in recognition of connected digit. That work however, is beyond of the scope of this paper.

Four evaluation criterions are subjects of this paper interest which are the omissions, insertion, accuracy and match based on given time tolerance. The experiment conducted on different value of auto regressive model order p and sliding window length L is to analyze the effect of these parameters upon those criterions and speech utterance mode. This report is outline as follow. Next section describes human perception of word, and then the data used in this experiment which is Malay Connected Digits is explained in the next section. The following section after that is on approaches applied in the experiment. The result is reported in the following section with discussion and conclusion at end of the report.

2. HUMAN PERCEPTION OF WORD

Word pronunciation and perception are common task for human. In daily communication syllables in word are not pronounced clearly and at equal phase which lead to lack of acoustic information of the word. Human however can easily anticipate the incomplete information at perception level. This is so because in real time human communication, human does not listen speech utterances in complete but anticipate them by comparing some existing sound model in their brain [9][10]. Human perception is not only accurate but also rapid [11]. When a word is said, human will listen and able to segment chunk by chunk of acoustic information before making perception of the word. In most cases, a native speaker of a language would already know what is going to be said prior to end of the word uttered in that language.

As an example, the word *senaman* can be percept by human as stages at Table 1. When the sound /s/ is heard, native listener would already have in his brain a list of possible word starting with sound /s/. The list reduced as the acoustics information of the sound become clearer. In this straight forward example, the listener already percept the word at stage 5 before the word is fully uttered.

Stage	Sound heard	Optimization of the word
1	/s/	**s**atu, **s**aya, **s**emut, **s**enapang, **s**enak, **s**eniman, **s**enada, **s**enaman **s**ilat, **s**ikap, **s**opan, etc… (any words in memory that start with /s/)
2	/se/	**se**mut, **se**napang, **se**nak, **se**niman, **se**nada, **se**naman (only word start with /se/ remain)
3	/sen/	**sen**apang, **sen**ak, **sen**iman, **sen**ada, **sen**aman (reduced to fewer words)
4	/se/ +/na/	**sena**pang, **sena**da, **sena**man (fewer possible words)
5	/se/ + /nam/	**senam**an (already anticipate the word)
6	/se/+/nam/+ /a/	**senama**n
7	/se/+ /nam/ + /an/	**senaman**

TABLE 1: Stages in Human Perception of The Word *senaman*

Some researcher works on phoneme as the basic chunk of acoustic information deriving the word [12][13]. Phoneme based however, is too fragile as to have only very small interval. Thus is not suitable for integration of spectral and temporal dependencies [14]. Furthermore, phoneme segmentation is much more difficult compare to syllables due to the same reasons. This work emphasizes on syllables as the basic acoustic chunk for segmentation and perception as it is salient and may exhibit specific acoustic characteristic especially in Malay language.

3. THE SYLLABLES IN MALAY DIGIT

There are 20 syllables that consist in Malay digit from 0-9. The syllables are { ko\, song\, sa\ , tu\ ,du\, a\, ti\, ga\, em\, pat\, li\, ma\, e\, nam\, juh\, la\, pan\, sem\, bi\, lan\ }. Table 2 shows the combination of phonetic alphabets in the 20 syllables. In general there are combinations of four types of consonant alphabets which are plosive, nasal, fricative and lateral approximant consonants and two types of vowels which are front and back vowels.

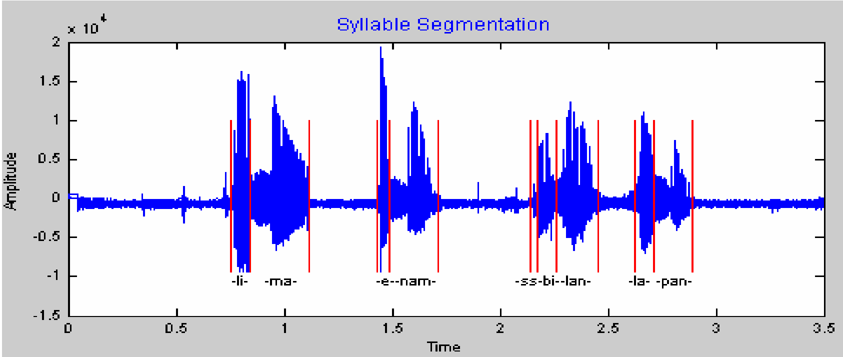


FIGURE 1: Connected Digit and its Syllable Manual Segmentation.

These syllables are visually and audibly distinguishable when pronounced in clear read mode. In spontaneous mode on the other hand, the cues are not as clear. Therefore, even for manual segmentation by human the task is not easy. Figure. 1 shows example of connected digit and its manual segmentation for digit ”lima-enam-sembilan-lapan” (5698).

Most of the syllables have two significant energy clusters where the start and end of the syllables is visually noticeable based on the abrupt changes of the energy. However, when pronounced connectedly and closely together, the correct segmentation point do not significantly visual and false abrupt changes exist in other places depends on the speaker's utterance style. The syllable's sound and the signal pattern visually are also influenced by the preceding and following syllables. This effect is known as co-articulation effect.

No.	Syllables	Descriptions
1	/a/ and /e/	A front vowel syllable
2	/bi/, /ga/ and /ti/	A plosive consonant with a front vowel
3	/ko/, /du/ and /tu/	A plosive consonant with a back vowel
4	/ma/	A nasal consonant with a front vowel
5	/pan/	A plosive consonant with a front vowel ending with a nasal consonant
6	/nam/	A nasal consonant with a front vowel ending with a nasal consonant
7	/sem/	A fricative consonant with a front vowel ending with a nasal consonant
8	/pat/	A plosive consonant with a front vowel ending with plosive
9	/la/	A lateral approximant consonant with a front vowel
10	/sa/	A fricative consonant with a front vowel
11	/em/	A front vowel with a nasal consonant
12	/li/	A lateral approximant consonant with a front vowel
13	/lan/	A lateral approximant consonant with a front vowel ending with a nasal consonant
14	/juh/	An lateral approximant consonant with a back vowel ending with a fricative consonant
15	/song/	A fricative consonant with a back vowel ending with a nasal and plosive consonants

TABLE 2: Phonetic Attributes of The 20 Syllables.

For read mode patterns, most of the pattern signal is quite clear as there are significant silence intervals in between words and even syllables for some cases. The only minimum noises are from nasal, mouth and lips ie. there is no background noise. However, for spontaneous mode patterns It is much complex as the position of the words does not necessary have silence interval and noises from back ground exist which leads to extra abrupt changes of the energy. It is observed that syllable with consonant like 'p','l' and 't' make short instance fluctuation in energy signal. These extra fluctuations of energy may lead to difficulties in obtaining the right point and increase number of insertion in segmentation.

4. THE STATISTICAL APPROACHES

Both Brandt's GLR algorithm and Divergence algorithm use statistical analysis in determining the segment points. The signal is assumed to be described by a string of homogeneous units, each of which is characterized by a statistical model of the form:

$$y_n = \sum_{i=1}^{p} a_i y_{n-i} + e_n \qquad \text{(i)}$$

where e_n is the excitation of the acoustic channel and is an uncorrelated zero mean Gaussian sequence with $\mathrm{var}(e_n) = \sigma_n^2$ The model is parameterized by the vector Θ defined by:

$$\begin{aligned} \Theta^T &= (\theta^T, \varphi^T) \\ \theta^T &= (a_1, ..., a_p) \end{aligned} \qquad \text{(ii)}$$

where φ is parameter vector which determines the sequence σ_n. These methods consist in performing on line a detection of changes in the parameter Θ starting from location of the previous detected. The algorithms are basically, (1). Detect when changes occurs. (2). Estimate the location of the changes. The two segmentation algorithms differ in the assumption of the excitation of the model and in the choice of the test statistics. Basically, a fixed and a growing window are used and then suitable distance estimation compares the two spectra.

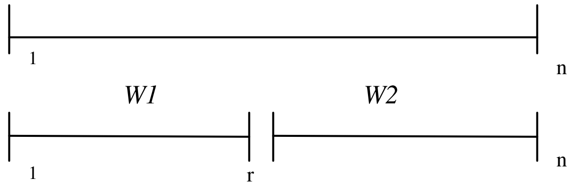


FIGURE 2: Location of the three windows in Brandt's GLR algorithm

4.1 Brandt's GLR algorithm

In Brandt's algorithm, the model assume

$$\sigma_n^2 = \sigma_n \quad \text{(i.e. } \varphi = \sigma \text{)} \qquad \text{(iii)}$$

The test is to monitor ($y_1, \ldots, y_n$), decide between the hypotheses:

- $H_o : \Theta = \Theta_0 \qquad for \quad 1 \le k \le n$
- $H_1 : \exists\, r \quad such\ that \quad \Theta = \Theta_1 \ \ for\ 1 \le k \le r$

  $and\ \ \Theta = \Theta_2 \quad for\ 1 \le k \le n$

There are 3 windows to manage as shown in Figure 2. The algorithm attempts to decide based on the likelihood between the two hypotheses, where the time instant r and the Θ_i's are replaced by their maximum likelihood estimates, so that the changes is detected if the distance

$$D_n = \max_r \ \max_{\Theta_1 \Theta_2} \ \max_{\Theta_0} \ \log\left(\frac{p\left[y_{1,\Lambda}, y_n \| H_1\right]}{p\left[y_{1,\Lambda}, y_n \| H_0\right]} \right) \ge \lambda \qquad \text{(iv)}$$

where λ is the threshold.

Then the estimate of the change $\hat{r}$ is the argument of the maximum in the relation (iv). The maximum likelihood estimates of the Θ's are given by the formulae:

$$\hat{\theta}(W_j) = \arg\min_{\theta} \sum_{k \in W_j} \left(Y_k - \sum_{i=1}^{p} a_i y_{k-i} \right)^2 \qquad \text{(v)}$$

$$\hat{\sigma}^2 = \min_{\theta} \frac{1}{card(W_j)} \sum_{k \in W_j} \left(Y_k - \sum_{i=1}^{p} a_i y_{k-i} \right)^2 \qquad \text{(vi)}$$

Where *W* denotes one of the three windows depicted in Figure 2. This finally yields the following formula for *D_n*.

$$D_n = \max_r D_n(r) \qquad \text{(vii)}$$

$$D_n = n \log \hat{\sigma}_0 - r \log \hat{\sigma}_1 - (n-r) \log \hat{\sigma}_2 \qquad \text{(viii)}$$

To avoid high computational cost in detection-estimation of the above formula, the different parameter Θ_0, Θ_1 and Θ_2 are identified by *Durbin-Levinson algorithm*.

4.2 Divergence Algorithm

The model set for divergence is similar as in Brandt's BLR algorithm. Equations (i), (ii) and (iii) are applied. In Divergence algorithm, the test is based on the monitoring of a suitable distance measure between the two models Θ_0 and Θ_1 located as shown in Figure3

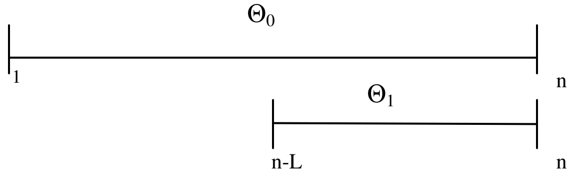


FIGURE 3: Location of The Two Models for The Divergence Algorithm

This distance is derived from the cross entropy between the conditional distribution of these two models. Consider

$\overset{T}{y}_m = (y_{1,\mathrm{K}}, y_m)$ and denote by

$$\vartheta_0(y_m \,\|\, y_{m-1}) \;\; and \;\; \vartheta_1(y_m \,\|\, y_{m-1})$$

the two conditional densities corresponding to the models of Figure 3. Introduce the cross entropy between the two models, ϑ_0 and ϑ_1:

$$w_m = \int \vartheta_0(y \,\|\, y_{m-1}) \log \frac{\vartheta_1(y \,\|\, y_{m-1})}{\vartheta_0(y \,\|\, y_{m-1})} dy - \log \frac{\vartheta_1(y \,\|\, y_{m-1})}{\vartheta_0(y \,\|\, y_{m-1})}$$

Which introduce the cumulative sum $W_n = \sum_{m=1}^{n} w_m$

It can be shown under hypothesis *H_0*: $\Theta = \Theta_0$ and under hypothesis *H_1*: $\Theta = \Theta_1$.

A change detection occur when the long term model disagree with the short term model in the sense of cumulative sum statistics. Detection is done by comparing the cumulative sum with threshold value as follow

$$\max_{1 \le r \le n} \hat{W}_r - \hat{W}_r > \lambda \text{ and}$$

$$W_n = \sum_{m=1}^{n} w_m + \delta$$

where δ is a bias value and λ is a threshold.

5. EXPERIMENTAL PROCEDURE

Our purpose is to see the affect of number of the auto regressive model order *p* and sliding window length *L* in Brandt's algorithm and Divergence algorithm to the four measurement criterions which are the omission, insertion match and accuracy. These criterions are evaluated by comparing the points gotten from automatic segmentation using the methods with a manual procedure. Hereon it will be known as referenced points. Points by manual procedure is considered the best as it used humanly ability consist of visual and audio intelligent that is through viewing the waveform pattern abrupt changes and verified through listening. Figure 4 shows an example of automatic and manual segmented pattern and the insertions and omission points. The graph is plotted using SFSWIN ver 1.5 from University College of London.

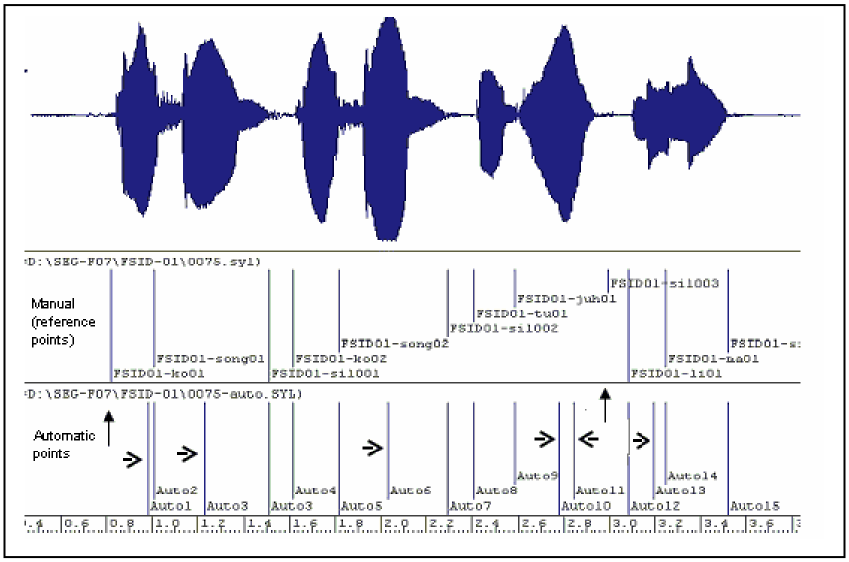


FIGURE 4: Example of comparison between reference (above) and automatic (below) segmentation points and its corresponding omission and insertion points. ↑ indicates omission points while → and ← shows insertion points.

The measurement criterions is defined as below adapted from [6],
Let *U* = *{U*$_1$*, U*$_2$*, …. U*$_n$*}* and *V* = *{V*$_1$*, V*$_2$*, …. V*$_p$*}* be the points in second of the segmentation marks obtained respectively by an automatic algorithm and by manual procedure which acts as the reference segmentation points. For each U_j, a correspondence is done with the reference segmentation by determining the time instant V_{kj} which is closest to U_j. A sequence Vu =*{* V_{k1}, V_{k2}, …. V_{kn}*}* is built in order to compare both segmentations.

Thus, omission is evaluated as points in *Vu* that is not in U_j and insertion is defined as extra points in U_j that is not in *Vu*. Match is calculated as number of similar points in U_j and *Vu* say, *m* divide by number of points in *V*, *p*. Thus, it can be defined as, match = *(m/p * 100*) and accuracy = *((m/p+n) * 100*) where accuracy will be influenced by number of insertion occurrences. Performance of the methods is evaluated better if has less omission and insertion and high match and accuracy. The value for threshold, λ for Divergence and Brandt were delivery set low as to be able to avoid omission and thus get better match however, it may lead to high insertion occurrences. Nevertheless, insertion is not our main concern in this work.

6. EXPERIMENTAL RESULT

The results of the experiments are presented in two sections. The first is on comparison between Brandt's and divergence algorithm and secondly on comparison between read mode and spontaneous mode. Comparisons are made in term of the performance criterion stated earlier in the effect of the changes made on two experimental variables which are the auto regression model order *p* and sliding window length, *L*.

6.1 Comparison between Divergence and Brandt's GLR algorithms

The result shows that Brandt's has opposite effect to divergence algorithms in experiment on sliding window size *L* for all four criterion that are match, accuracy, omission and insertion. Incrementing the size *L* from 300 to 500 sample lead to better match, better accuracy and less omission but greater insertion for Brandt's algorithm. On the hand, for divergence algorithm the effect would be fewer matches, less accuracy, greater omission and lesser insertion.
Similar effects is observed in the experiment on the value for auto regression model order, *p*. Incrementing the value *p* increase accuracy and match, lessen omission but increase the insertion for Brandt's. In contrast for divergence, it leads to decreasing accuracy and match, increase the omission and decrease the insertion.

In general, there are only slightly different of better match observed in divergence algorithm compare to Brandt's in all experimental parameters. The increment or decrement probability different are very small around 0.005 to 0.20 percent for both algorithms. Figure 5 and Figure 6 show the graphs comparing the best match for both algorithms on spontaneous and read mode speech for experimental on *p* and *L*. The figures shows divergence algorithm perform better match with parameter *p*=2 and *L*=300 and Brandt with *p=2* at *L*=500.

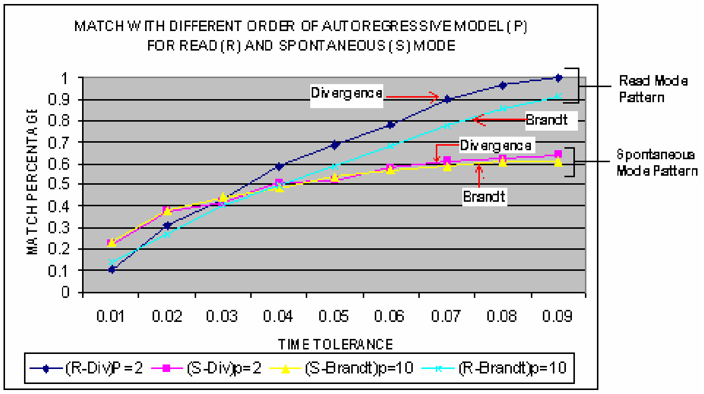


FIGURE 5: Match comparison between the best value for *p* for Divergence and Brandt on spontaneous and read mode patterns.

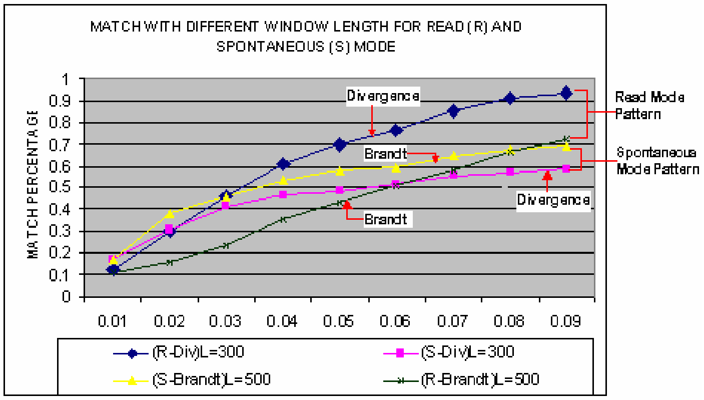


FIGURE 6: Match comparison between the best value for *L* for Divergence and Brandt on spontaneous and read mode patterns.

6.2 Comparison between Read Mode and Spontaneous Mode

It is expected that read mode segmentation would be easier and thus perform better than spontaneous mode. However, the experiment conducted observed that for certain criterion spontaneous mode has better performance than read mode. Number of insertion in spontaneous mode is less compare to read mode which lead to accuracy calculation for spontaneous mode better than read mode. On the other hand omission and match of spontaneous mode are not really good. The best match for spontaneous mode is 70% for L=500 and p=2 at time tolerance 0.09 second while for read mode it is 100% for p=2 and L=300 at time tolerance 0.09 using divergence algorithm. Similar to insertion with accuracy, omission goes less when match is high. The best accuracy for both modes is obtained using divergence algorithm is 44% for spontaneous mode and 42% for read mode. It can be noticed that accuracy criteria for read mode drop significantly compare to its match criteria due to accuracy calculation influenced by number of insertion occurrences. Figure 7, Figure 8 and Figure 9 show the best experimental result of accuracy, insertion and omission respectively for both read mode and spontaneous mode.

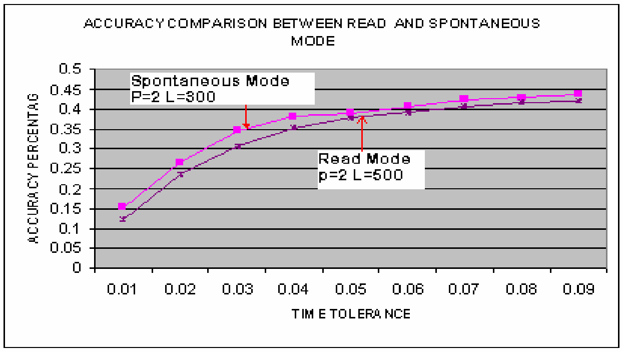


FIGURE 7: Accuracy comparison between spontaneous and read mode patterns.

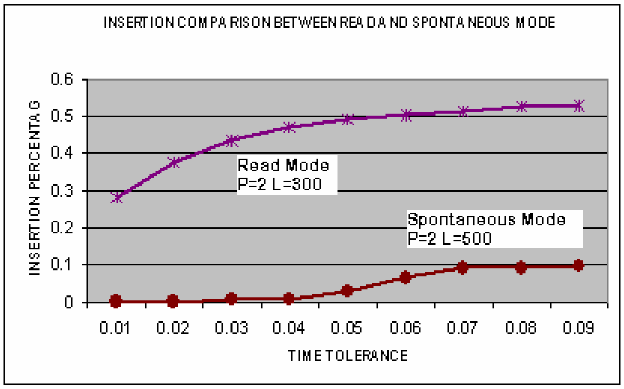


FIGURE 8: Insertion comparison between spontaneous and read mode patterns.

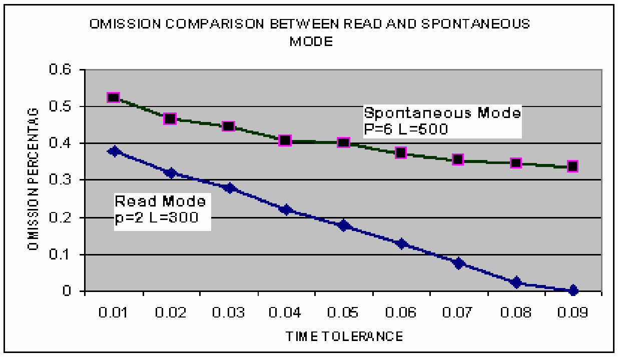


FIGURE 9: Omission comparison between spontaneous and read mode patterns.

7. CONCLUSION & FUTURE WORK

Segmentation is an inherently extremely a difficult problem [15]. The statistical approaches used in this experiment do not use any acoustic information in determining the segmentation point. Furthermore, the threshold value is set to low as not to miss any match in reference pattern. Thus, insertion is expected to occur. Nevertheless, the objective of getting no omission is achieved as segmentation match reach up to 100% using divergence algorithm.

Malay is an agglutinative language where words forming are combination of syllables or phoneme. Phoneme based modeling is too fragile as to have very small interval thus not suitable for integration of spectral and temporal dependency [14]. Whereas, syllable is able represent specific acoustic characteristic thus maybe most suitable as the basis in forming Malay words. Previous works in Malay isolated syllable recognition able to reach more than 80% recognition. However, no works has done to segment connected syllables in connected words. This work is the initial step to segment syllables as the basis for recognition of continuous Malay words. It is our future plan to develop an intelligent algorithm that can guess missing syllables with embedded language knowledge in the system. Our test using simulated data on the prototype system shows a promising result [16]. However, the result from this works indicates that insertion may become a drawback for real speech data. On going work is done to eliminate the insertion using Neural Network.

8. REFERENCES

1. H.N Ting, Y. Jasmy and S.H Salleh. "*Malay Syllable Recognition Using Neural Network".* In Proceeding of the International Student Conference on Research and Development, SCOReD, Kuala Lumpur, 2006
2. Abas Lutfi. *"Linguistik Deskriptif Nahu". Dewan Bahasa dan Pustaka*, Kuala Lumpur: pp. 10 - 20 (1971)
3. J.L Rouas, J. Farinas, F. Pellegrino and R.A Obrecht, "*Rhythmic Unit Extraction and Modeling for Automatic Language Identification*". Speech Communication 47: 436-456. 2005
4. Md Sah Hj Salam and Mohamad Nasir Said Ibrahin. "*An Initial Experiment on Syllable Based Approach For Malay Digit Recognition".* In Proceeding of Advance Technology Congress. ATC2003, Putrajaya, Malaysia 2003.
5. B. Michele and V.N. Igor, "*Detection of Abrupt Changes: Theory and Application*", *Prentice Hall, INC*. USA 1993

6. S. Jarifi, D. Pastor and O. Rosec,. *"Brandt's GLR Method & Refined HMM Segmentation for TTS Synthesis Application"*. In Proceeding of European Signal Processing Conference, EUSIPCO'2005. Antalya,Turkey. 2005
7. T. Jehan T. *"Musical Signal Parameter Estimation"*. Master Thesis, University of Rennes, France. 1997
8. R.A. Obrecht, "*Automatic Segmentation of Continuous Speech Signal"*, IEEE Trans. Acoustic, Speech and Signal Processing, vol ASSP-36(1). pp 29-40, 1988
9. K. Kohler *"Segmental reduction in connected speech in German: Phonological facts and phonetics explaination"*. Speech Production and Speech Modelling, Kluwer, Dordrecht. pp.69-92. 1990
10. O. Engstrand, *"Sytematicity of phonetic variation in natural discourse"*. Speech Communication 11, pp. 337-346. 1992
11. *Language Production and Perception*, online : http://www.ling.upenn.edu/courses/Fall_1998/ling001/production_perception.html. pp. 1 – 11.
12. P. Cosi, J.P. Hosom, and F. Tesser. "*High performance Italian continuous digit recognition*", In Proceedings of International Conference on Spoken Language Processing, Beijing, China, ICSLP 2000.
13. W. Wei .and and S.V. Vuuren. *"Improved neural Network Training of Inter-Word Context Units for Connected Digit recognition"*, In Proceeding of IEEE International Conf. on Acoustics, Speech & Signal Processing, Seattle, ICASSP 1998
14. T. Nuttakorn and K. Boonserm. " *A syllable - based connected Thai digit speech recognition using neural network and duration modeling"*. In Proceeding of *The 1999 IEEE International Symposium on Intelligent Signal Processing and Communication. Pp 785-788. 1999.*
15. L.R Rabiner and M.R Sambur. "*Some Preliminary Experiments in the Recognition of Connected Digits".* IEEE Trans. Acoustic, Speech and Signal Processing, vol ASSP-24. pp 170-182 April 1976
16. Md Sah Hj Salam , Dzulkifli Mohamad dan S-H Salleh." *Speech Anticipation via Genetic Optimization: An Experiment on Simulated Data",* In Proceeding of International .Conference on Artificial Intelligence in Engineering and Technology, ICAIET '06, Kota Kinabalu, Sabah, Malaysia.2006.

GPS-less Localization Protocol for Underwater Acoustic Networks

Al-Khalid Othman okhalid@feng.unimas.my
Faculty of Engineering
Universiti Malaysia Sarawak
Kota Samarahan, 94300, Sarawak, Malaysia

Abstract

The problem of underwater positioning is increasingly crucial due to the emerging importance of sub-sea activities. Knowledge of node location is essential for many applications for which sensor networks can be used. At the surface, positioning problems have been resolved by the extended use of GPS, which is straightforward and effective. Unfortunately, using GPS in the sub-sea environment is impossible and positioning requires the use of special systems. One of the major challenges in the underwater acoustic networks (UANs) area of research is the development of a networking protocol that can cope with the management of a dynamic sub-sea network. We propose a scheme to perform node discovery, using only one seed node (primary seed) in a known position. The discovery protocol can be divided into two parts: First, building up the relative co-ordinate system. Second, involving more remote nodes becoming seed nodes for further discoveries. Four different algorithms have been investigated; (i) Farthest/Farthest Algorithm, (ii) Farthest/Nearest Algorithm, (iii) Nearest/Farthest Algorithm and (iv) Nearest/Nearest Algorithm. We investigated the performances of random and fixed (grid) network topologies. Different locations of primary seed node were exercised and statistics for node discovery will be reported.

Keywords: Underwater Acoustic Network, Protocol, Localization, Network Discovery, Network Scenarios.

1. INTRODUCTION

Underwater acoustic networks can be formed by acoustically connected anchored nodes, autonomous underwater vehicles (AUVs), and it is possible to have a surface link that serves as a gateway to provide a communication link to an onshore station. Figure 1 shows a generic underwater acoustic network.

An underwater network has several limitations compared to radio networks, most importantly the propagation delays which are very long with limited bandwidth. Another restriction that needs to be considered in UANs is the incapability of modems to transmit and receive signals at the same time (the near-far effect). To prevent the near-far effect which causes loss of data, scheduled transmission is required. The technique of node discovery must minimize the exchange of data in order to keep network management overheads to a minimum. Furthermore, in underwater acoustic networks, node connectivity is unpredictable. This connectivity depends upon several factors such as relative node orientation, noise level, propagation losses and fading. The connectivity is further affected by relative movement of the nodes, node and link failures and the addition of new nodes. Consequently, a very important characteristic of an underwater communication network is the ability to deal with changing topology.

To achieve full network functionality, nodes need to self-organize in an autonomous network which can adapt to the characteristics of the ocean environment. This paper addresses the following problem: Given a set of nodes with unknown position co-ordinates, determine the relative co-ordinates of nodes.

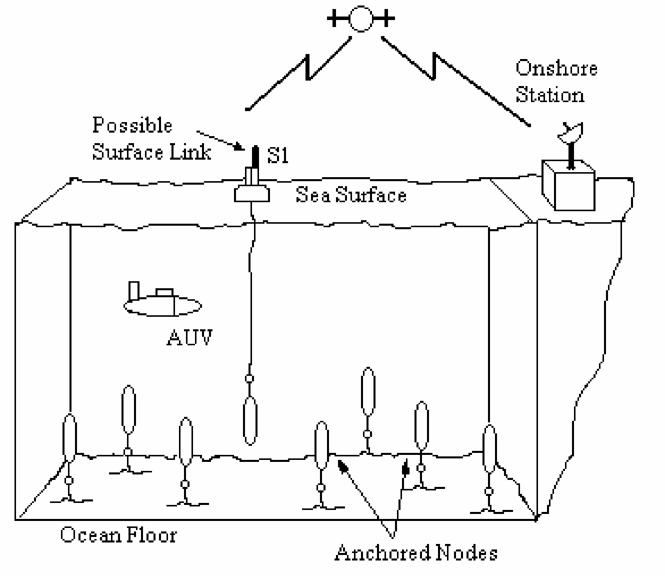


FIGURE 1: Underwater acoustic network

2. RELATED WORK

In a localization system, several capabilities are necessary. First, the measurement techniques used to gain the information such as distance and other information. Second, the network discovery protocol which concerns the communication between nodes. Finally, techniques of deployment either using the anchor or beacon (nodes with known co-ordinate) or anchor-free bases.

The most popular measurement type is ranging. There are two methods used to obtain range measurements; timing and signal strength. Ranging is usually provided by estimating the distance to a neighbour by measuring the received signal strength (RSS) [1-3] from that neighbour, by time of arrival (ToA) [4] or by time difference of arrival (TDoA) [5].

In the ToA approach, the distance between a remote node and the beacon is measured by finding the one way propagation time between that node and the beacon. Geometrically, this provides a circle, centred on the beacon, on which the remote node must lie. By using at least three beacons to resolve ambiguities, the remote node's position is given by the intersection of the circles. In the TDoA approach, the time difference of transmission and reception at the beacons is used. By using this approach, the time synchronization can be eliminated [5]. Time of arrival range measurement can be implemented using inquiry-response protocol [6, 7]. Another measurement method for node localization is Angle of Arrival (AoA) [8] where the node estimates the direction from which a neighbour is sending a signal. It can be implemented either using an antenna array, or a combination of radio and ultrasound receivers. In this method, triangulation is used for the localization.

A localization system can be implemented that is based on RSS, ToA, TDoA or AoA, or a combination of these. However, due to a non-uniform signal propagation environment, especially in underwater acoustic networks, RSS methods are not very reliable and accurate. With antenna array is needed in the AoA method; it is impractical to employ in large networks because it is very costly. Furthermore, in this method, nodes may require additional hardware such as a digital compass to provide more information about the node's orientation. Even though ToA or TDoA may require additional hardware at the sensor nodes to receive a signal [9] these methods have better accuracy and are most suitable to be implemented in an underwater environment.

Another requirement for a localization system is the network discovery protocol. There have been many investigations in the radio network field into neighbourhood node and topology discovery [10–12]. In these protocols each node broadcasts a message to gain information of the network. Protocols, such as Bluetooth [13], propose and analyze symmetric protocols for 2-node link formation, which is based on a random schedule. Law et al. [14] and Birthday protocol [11] propose a probabilistic protocol for node discovery; a node decides, with a probability *p,* to start discovering other nodes, or, with probability 1-*p*, to listen until it discovered by another node. A node gives up, either if it does not discover another node or does not hear from any other node within a defined period of time. However, these protocols aim at establishing one-to-one connections.

The discovery protocol discussed above may require explicit exchanges of messages containing the node address/ID and, sometimes, the node co-ordinates. Furthermore, the nodes do not share their discoveries with other nodes in the region. This typically requires some form of reliable broadcast system which makes these schemes very expensive in terms of energy consumption and convergence time, matters of high priority in underwater networks.

Previous research has addressed two deployment techniques for localization in ad hoc networks. These are known as anchor-based and anchor-free. Localization algorithms that rely on anchor nodes [15-23] assume that a certain minimum number, or fraction, of the nodes know their position by structured placement or by using some other location mechanism. The advantage of having anchor nodes which are spatially distributed throughout the network region is that they let devices compute their location in a scalable, decentralized manner. For such mechanisms, questions arise as to the number and the sophistication of placements of anchor nodes. Doherty [15] has proposed a convex optimization technique with the anchor nodes to be placed on the outer boundary, preferably at the corners of the deployment area to work well. The advantage of this approach is that it requires very few anchors (3 or 4) since all system constraints are solved globally. However, this algorithm is not very robust to failures when there are ambiguities in measurements. The Cricket Location Support System [16], Active Badge [17], the Bat System [18] and HiBall Tracker [19] use proximity based techniques and propose guidelines for the deployment of anchor nodes based on practical considerations (influenced by environment conditions and application requirements). The anchor nodes are located in an unobtrusive location like a ceiling or wall. Another approach to addressing the deployment problem of anchor nodes is using optimal placement algorithms including Pursuit-Evasion [20] and Facility Location [21, 22].

In contrast, the anchor-free method [23], uses local distance information to attempt to determine node co-ordinates. In this method every node in the network performs discoveries and shares the information with neighbouring nodes and, thus, defines the local co-ordinate system and finally the network co-ordinate system.

Nevertheless, the techniques discussed in the deployment system above are (a) not scalable to large sensor networks, and (b) not suitable for rapid deployment. In addition, with the limitations in such underwater acoustic networks as mentioned earlier, it is impossible to employ anchor nodes that infer their position through GPS. In our method, we do not use any anchor nodes in the network except the primary seed node (node with known co-ordinate). Information received during discovery is shared with neighbouring nodes and the information is then used to determine second order seed nodes.

3. DISCOVERY PROTOCOL AND LOCALIZATION ALGORITHM

To establish the relative co-ordinate system for the network, the protocol proposed in this paper uses various commands for peer to peer communication. Table 1 presents these commands.

Command	Description
001 (DISC_COMM)	Discovery Command – enables neighbours to establish distances from the sender
100 (NOT_RESPONSE)	Not Response Command – enables the node not to respond for any command
010 (MORE_DISC)	More Discovery Command – enables the node to become a seed node for further discovery
011 (RESPONSE)	Response Command – enables the node to respond again for any command received

TABLE 1: Command and Description during Node Discovery

Discovery and localization protocol can be divided into two parts:

Stage 1: Building up the relative co-ordinate system using the information gained from the first three seed node discoveries.
Stage 2: Further node discovery by selected seed nodes.

Assume that S_1 is the first seed node and there are remote nodes available in its region of communication. Following node deployment, seed node S_1 will broadcast a DISC_COMM packet. It will await replies from nodes within its range. When replies are received, information such as node ID and distance are retained in the seed node memory. In this first discovery, the seed node only discovers the node IDs and their distances but not their location. The next stage is to set a second seed node for further discovery. We propose that the second seed node selected will be the farthest node from S_1. The advantage of choosing the farthest node as the second seed node, S_2, is that a larger area can be covered more quickly. Assume that A_i is the information set of a discovery sequence, it contains the distance measurement and node ID of those nodes replied. S_1 will broadcast A_1 and MORE_DISC to its neighbouring nodes. At this point, each node in the S_1 region has the information of A_1. If a node in the S_1 region receives this command and the ID is equal to the node ID of the next seed node, then this node will recognise that it is to become the second seed node, S_2. S_2 proceeds with the same manner of discovery; it will then broadcast the newly discovered information, A_2, back to its neighbours. The neighbouring nodes that receive this information will store the new information in their memory. Assuming that there is no data loss during broadcasting, after receiving information from S_2, S_1 will then update its own neighbours by re-broadcasting the A_2 data. At this point each node in the S_1 and S_2 regions has the information of A_1 and A_2. At this juncture, the locations of any overlap nodes from S_1 and S_2 are ambiguous. In order to solve this ambiguity, we introduce a third seed node, S_3. S_3 is chosen from those nodes that lie in both the S_1 and the S_2 regions and have the maximum summation distance from S_1 and S_2. After selecting S_3, S_1 will send another MORE_DISC command to define S_3. S_3 will then start a new discovery process by broadcasting a DISC_COMM command. After it receives replies from neighbourhood nodes, it rebroadcasts the information, A_3, back to its neighbours. Since S_1 and S_2 are in the region of S_3, when they receive the new information from

S_3 they immediately broadcast the information to their own neighbours. Figure 2 illustrates the discovery process made by the primary seed node for building up the relative co-ordinate system.

Figure 3 shows the regions of two and three distance measures after discovery by the first three seed node. The grey area in this figure shows the area that has knowledge of three distance measures of S_1, S_2 and S_3. Consider that S_1 has absolute knowledge of its own coordinate defined here as 0, 0. S_2 will be assumed to be at d_{12}, 0 coordinate, where d_{12} is the distance of the farthest replying node from S_1. With S_1 being the origin of the relative coordinate system, S_2 is defined to lie on the positive x axis. S_3 is now assumed to have a positive y component to define the y axis. With the assumptions made and information received, nodes in the overlap region are able to calculate their own coordinates and the coordinates of other nodes using the triangulation technique. Table II shows the summaries this approach made.

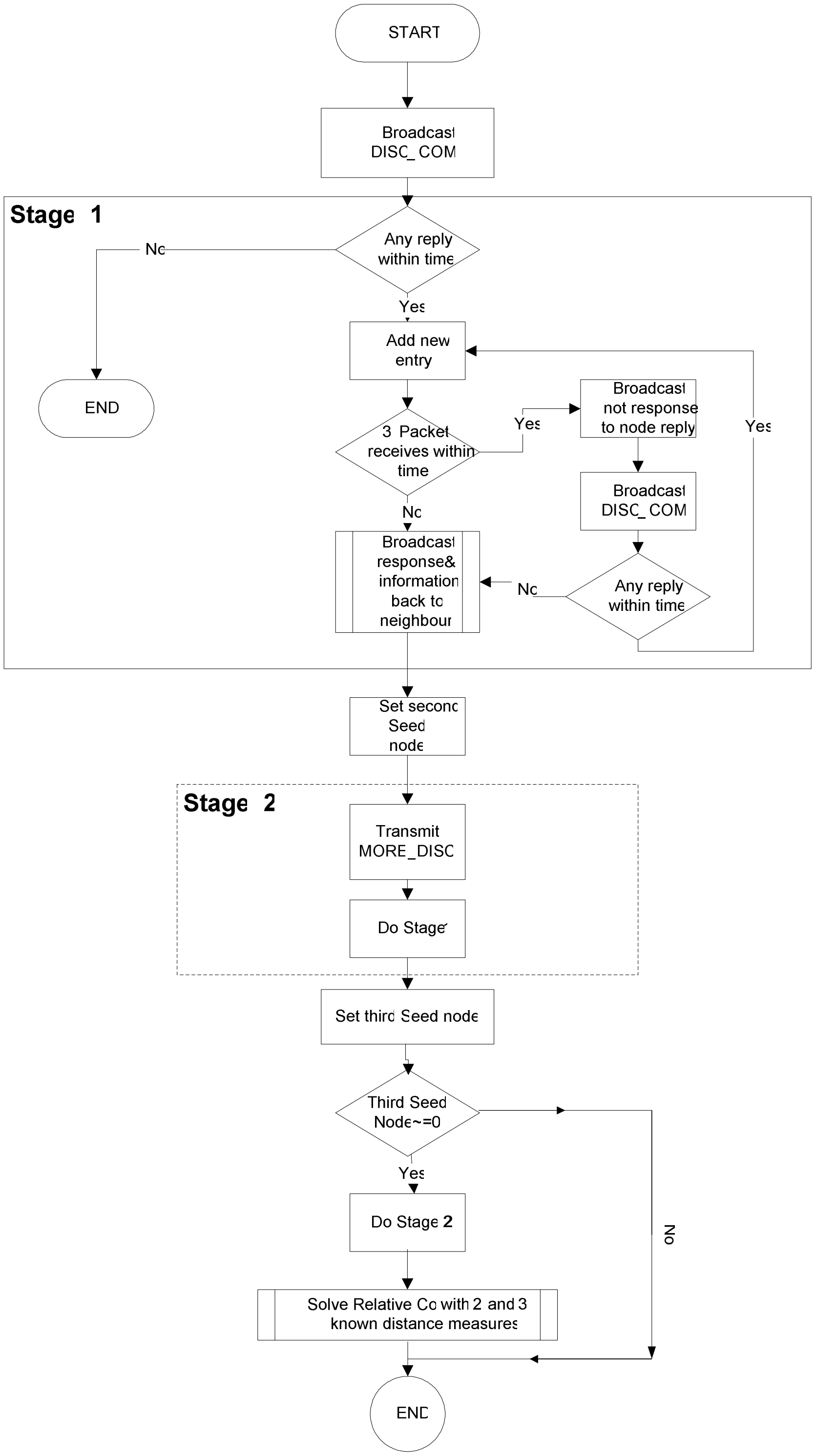


FIGURE 2: Discovery process for build-up the relative co-ordinate system

The cross-hatched region in figure 3 shows the area where only two known distance measures from their seed nodes are certain. As there are two solutions from this method, it is essential to know on which side of the line the nodes lie. This is the drawback with only two distance measurements, where we have the ambiguity of node placement. This may be resolved using the method described below. The computation of the coordinates will be done locally at each node.

Seed Node	x-Co-ordinate	y-Co-ordinate
First, S_1	0	0
Second, S_2	d_{12}	0
Third, S_3	$d_{13}\cos\theta$	$d_{13}\sin\theta$

TABLE 2: Relative Co-ordinate System

$$\theta = \cos^{-1}\left(\frac{d_{12}{}^{2} + d_{13}{}^{2} - d_{23}{}^{2}}{2 \times d_{12} \times d_{13}}\right)$$

d_{12} – distance between first and second seed node
d_{13} - distance between first and third seed node
d_{23} – distance between second and third seed node

Assuming that:

```
$M_d$ = maximum Distance Broadcast

$NR_1$ = Node Reply from $S_1$; $NR_2$ = Node Reply from $S_2$;
$NR_3$ = Node Reply from $S_3$
and
$NR_{12} = NR_1 \cap NR_2 \in NR_3$; $NR_{13} = NR_1 \cap NR_3 \in NR_2$;
$NR_{23} = NR_2 \cap NR_3 \in NR_1$

Algorithm for 2 Known Coordinates with $S_1$
and $S_2$ as Reference Nodes

If $NR_{12}$ ~= $\Phi$
   Compute Possible Locations of $NR_{12}$
   $Z_{1,i} = X_{1,i};Y_{1,i}$ and
   $Z_{1,j} = X_{1,j};Y_{1,j}$

   Compute Distance from $Z_{1,i}$ and $Z_{1,j}$ to $S_3$;
   $d_{1,i}$ and $d_{1,j}$ respectively

   If $d_{1,i} > M_d$ & $d_{1,j} < M_d$
      $NR_{12} = Z_{1,i}$
   end
   If $d_{1,i} < M_d$ & $d_{1,j} > M_d$
      $NR_{12} = Z_{1,j}$
   end
   If $d_{1,i} > M_d$ & $d_{1,j} > M_d$
      %No Possible Coordinate can be Calculated
      $NR_{12} = \Phi$;
   end
   If $d_{1,i} < M_d$ & $d_{1,j} < M_d$
      %No Possible Coordinate can be Calculated
      $NR_{12} = \Phi$;
   end
else, end
```

Similar algorithms can be applied to NR_{13} and NR_{23} to gain the relative location for the nodes in their region.

4. ALGORITHMS FOR SELECTING FURTHER SEED NODES

A. Farthest/Farthest Algorithm

The Farthest/Farthest algorithm uses the farthest undefined node from a previous seed node, and the node with the maximum summation distance from this node and the previous seed node.

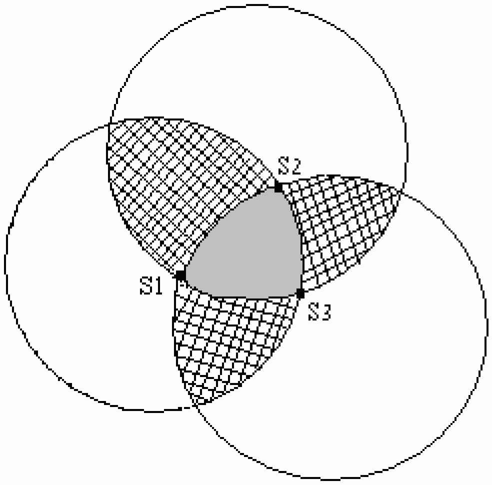


FIGURE 3: Region of two and three distance measurements

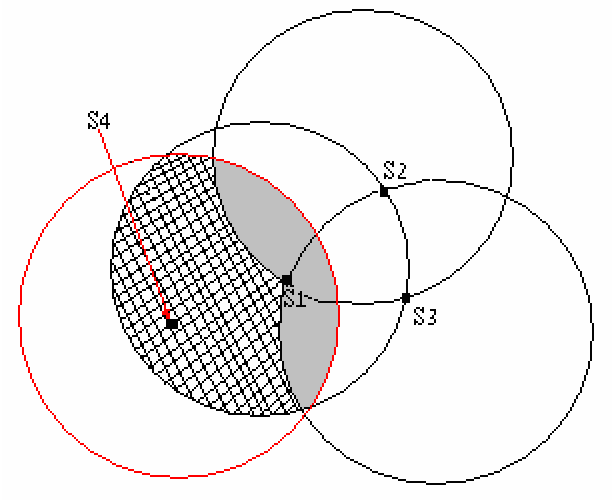


FIGURE 4: Area of nodes with known co-ordinate by S_4 discovery and potential area of the S'_4

Each remote node in the seed nodes region of the first stage of discovery will independently compute the relative location of all other nodes. Because of a lack of sufficient data, some nodes will be unable to fully define their location. Therefore, more information, such as distances from nodes with known co-ordinates, is needed for them to gain their relative co-ordinates.

In this Farthest/Farthest algorithm, first, each node in the seed nodes region will identify the undefined node in their dataset and find the farthest node from their seed node. If a node determines that it is the farthest undefined node from their seed node, it will automatically set itself as a new seed node and carry out a discovery process. When it receives replies from its neighbouring nodes, it will define its own relative co-ordinates and re-broadcast the information back to its neighbours. At this stage, the positions of overlap nodes between the new seed nodes and their first stage seed nodes are ambiguous. Therefore, another seed node is needed in order to solve the ambiguity. This seed node can be defined as the maximum summation distance of undefined node between the two seed nodes. The process will end when the seed node receives replies from all nodes with coordinates in its region, or the seed node cannot find its own coordinates where only one distance measurement of a node with known coordinates replies during the discovery.

Consider figure 4 as an example. S_4 is assumed to be the farthest node from S_1, therefore S_4 becomes the next seed node. S_4 precedes the same procedure of discovery by broadcasting DISC_COMM and waiting for reply from other nodes in its region. When it receives all the replies from the nodes, it will re-broadcast the discovery information back to its neighbours and use the discovery information to determine its own coordinates. If the seed cannot define its own relative co-ordinate then the next farthest node of the undefined node in the S_1 region is used as the new seed node. The discovery process will carry on until the new seed node resolves its own coordinates. The remote nodes in a region discovered by S_4 may contain one (from S_4) or two known distance measures (from S_1 and $S_{4,}$ say). With this information, all the nodes in the cross-hatched area shown in figure 3 still do not have sufficient data to solve their location, since there is ambiguity of the nodes' position. Following this problem, another seed node is needed. The next chosen seed node, S'_4, will be the undefined node with maximum summation distance from S_1 and $S_{4.}$

B. Farthest/Nearest Algorithm

A different approach can be taken in order to gain the relative coordinates of nodes. The Farthest/Nearest algorithm

uses the farthest undefined node from a previous seed node and the node with minimum summation distance from this node and the previous seed node.

C. Nearest/Farthest Algorithm

Alternatively, the Nearest/Farthest algorithm can use the nearest undefined node from a previous seed node and the node with maximum summation distance from this node and the previous seed node.

D. Nearest/Nearest Algorithm

The Nearest/Nearest algorithm uses the nearest undefined node from a previous seed node and the node with minimum summation distance from this node and the previous seed node.

5. SIMULATION SET-UP AND PERFORMANCE RESULTS

In this set of experiments, we generated a set of 30 to 100 nodes randomly in a $10 \times 10\ km$ area. The distances between nodes are set not less than $100\ m$ apart. At the initial stage of discovery nodes have no knowledge of location with respect to the other nodes and number of remote nodes in the network. We generated 100 samples and we used the same network topologies for all four algorithms in selecting the next seed node, as described in section 4.

Figure 5 shows the average of network set-up times for the four algorithms with different numbers of node deployment. It is clear that the network set-up time achieved by all the algorithms increases linearly with the number of nodes in the network. The figure suggests that, with lower numbers of node deployment, the Nearest/* algorithms ([*]) uses less time for network set-up compared to the Farthest/* algorithms. As expected the Farthest/* algorithms have least performance with the low numbers of node deployment. However, with high numbers of node

[*] indicates both Farthest and Nearest

deployment, the Farthest/* algorithms have a better performance (about 4%─13% less network set time) compared to the Nearest/* algorithms. Figure 6 shows the average number of seed nodes for four algorithms with different numbers of node deployment. The figure suggests that, with lower numbers of node deployment, the Farthest/* algorithms use more nodes to become seed nodes for further discovery compared to the Nearest/* algorithms. As expected the Farthest/* algorithms have least performance with the low number of node deployment. However, with high numbers of node deployment, it is obvious that the Farthest/Farthest algorithm has a better performance compared to the other algorithms. Figure 7 shows the average number of undefined nodes for four algorithms with different numbers of node deployment. The figure suggests that the average number of undefined nodes increases with the number of node deployment. As expected the Farthest/* algorithms gained better performance compared to the Nearest/* algorithms.

Also investigated were the performances of the algorithms with different locations of a primary seed node. Figure 8 shows the average of network set-up times for the four algorithms with different numbers of node deployment with primary seed node located at 1000, 5000. The figure suggests that, with lower numbers of node deployment, the Nearest/* algorithms use less time for network set-up compared to the Farthest/* algorithms. As expected the Farthest/* algorithms have least performance with the low number of node deployment. However, with high numbers of node deployment, the Farthest/Farthest algorithm has a better performance (3%─12% less network set up time) compared to the other algorithms. Figure 9 shows the average number of seed nodes for four algorithms with different numbers of node deployment and with primary seed node located at 1000, 5000. The figure suggests that the Farthest/Nearest algorithms use more nodes to become seed nodes for further discovery compared to the other algorithms. Figure 10 shows the average number of undefined nodes for four algorithms with different numbers of node deployment with primary seed node located at 1000, 5000. The figure suggests that the average number of undefined nodes increases with the number of node deployment. As expected the Farthest/* algorithms gained better performance compared to the Nearest/* algorithms.

Our first experiment compares the four algorithms in different performance matrices and studies the impact of different locations of primary seed node in a random topology. The experiment results suggested that the distribution of nodes in the area affects the performance of the algorithms. For larger numbers of node deployment, the Farthest/Farthest algorithms took less time for the network set-up, used fewer seed nodes for discovery and resulted in fewer numbers of undefined nodes compared to the Nearest/* algorithms. It also shows that the performance results vary with different locations of the primary seed node.

We also investigated the performances of two grid network topologies with 30 and 90 nodes deployed in a 10x10 km square with different locations of primary seed node. For each topology, we generated 100 samples with each node scattered 0─100 m around its position. We used the same network topology for all four algorithms for selecting the next seed node as described in section IV.

Figure 11 shows the average network set-up time for the 30 and 90 nodes in different locations of the primary seed node. As expected, different locations of the primary seed in the deployment area gave different performance results. A primary seed located at the centre (5000, 5000) of the deployment area gained better performances compared to a primary seed node located at 1000, 5000. This figure also shows that the Farthest/Farthest algorithm has better performances compared to the other algorithms. Figure 12 shows the average number of seed nodes with different locations of primary seed node. It shows that the Farthest/Farthest algorithm uses a smaller number of nodes to become seed nodes for the discovery compared to the other algorithms with primary seed located at 5000, 5000. Figure 13 shows the average number of undefined nodes with different locations of primary seed node. The figure suggested that in a 30-node topology, the Farthest/* algorithms have a smaller number of undefined nodes with primary seed node located at 1000, 5000 compared to primary seed node located at 5000, 5000. This result is reversed in the 90-node topology. In this 90-node topology, the Nearest/ Nearest algorithm gains fewer undefined nodes compared to the other algorithms when the primary seed node is located at 1000, 5000.

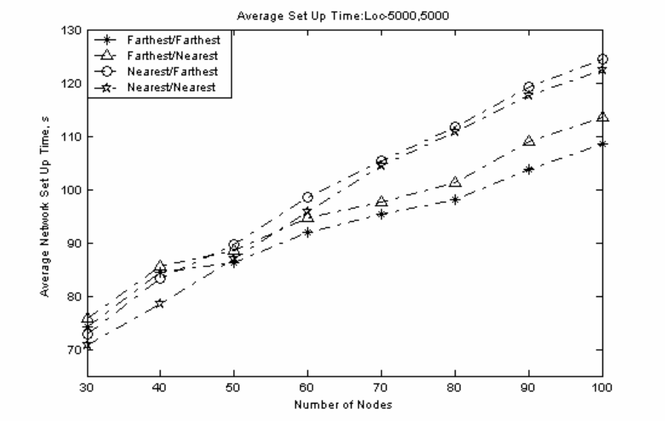


FIGURE 5: Average network set up time for random topology with primary seed coordinated at 5000, 5000

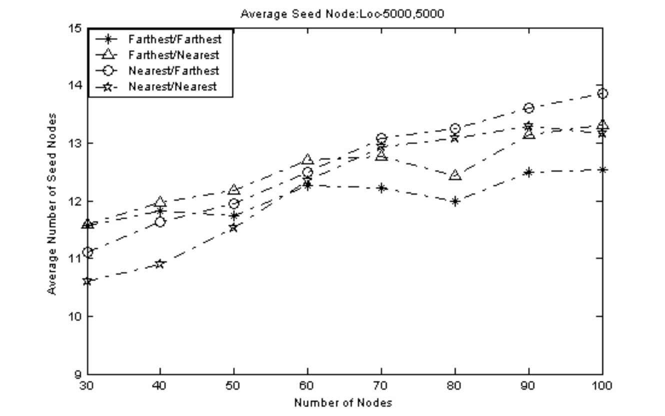


FIGURE 6: Average number of seed nodes for random topology with primary seed coordinated at 5000, 5000

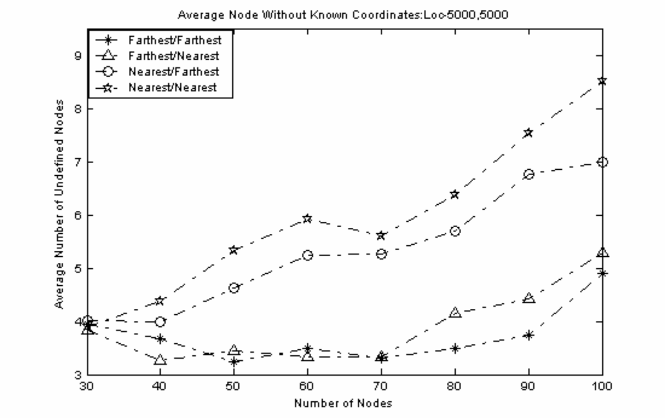


FIGURE 7: Average number of undefined nodes for random topology with primary seed coordinated at 5000, 5000

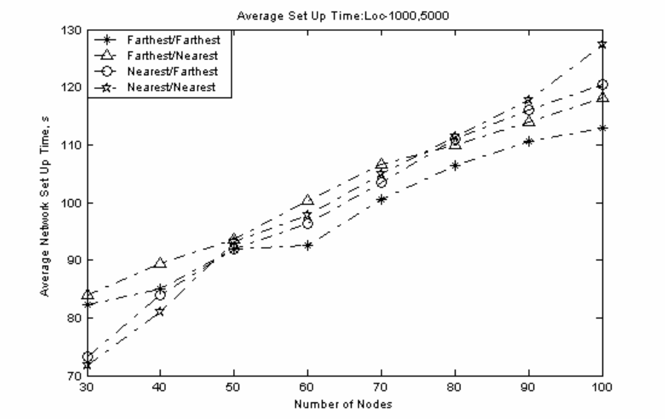


FIGURE 8: Average network set up time for random topology with primary seed coordinated at 1000, 5000

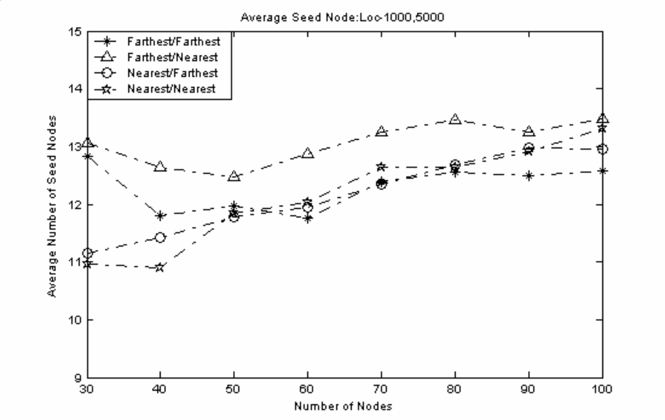


FIGURE 9: Average number of seed nodes for random topology with primary seed coordinated at 1000, 5000

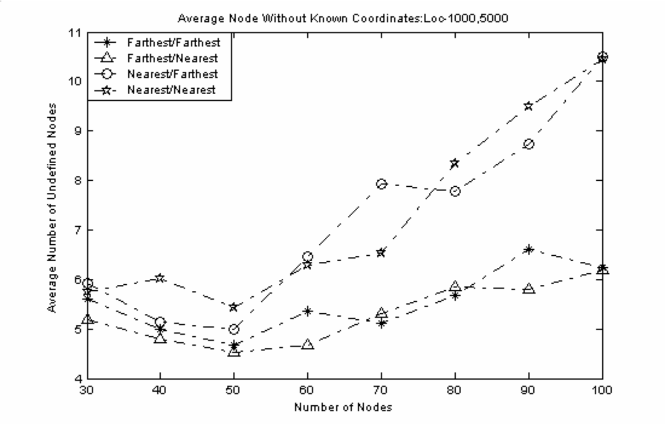


FIGURE 10: Average number of undefined nodes for random topology with primary seed coordinated at 1000, 5000

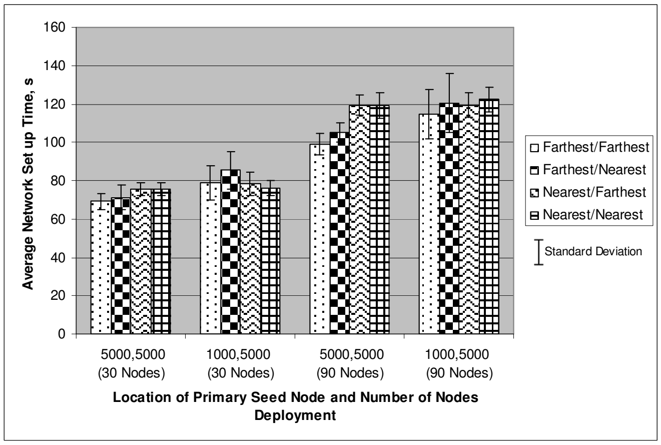


FIGURE 11: Average network set up time for 30 and 90 nodes in grid topology with primary seed coordinated at 5000,5000 and 1000, 5000

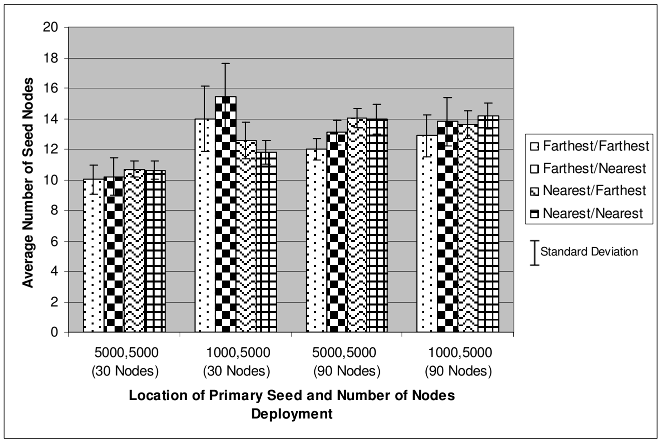


FIGURE 12: Average number of seed nodes for 30 and 90 nodes in grid topology with primary seed coordinated at 5000,5000 and 1000, 5000

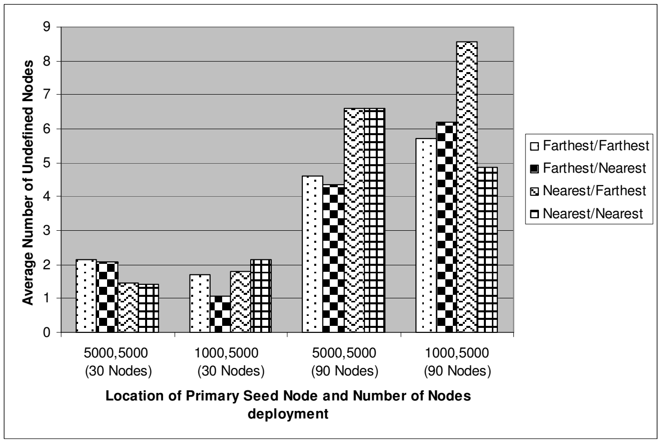


FIGURE 13: Average number of undefined nodes for 30 and 90 nodes in grid topology with primary seed coordinated at 5000, 5000 and 1000, 5000

6. CONCLUSIONS

We have presented a node discovery protocol and localization for UANs. The discovery protocol and localization algorithms proposed here form one of the possible approaches to collaborative location discovery. What is unique in our protocol is that we do not use any anchor node except the primary seed node and use the information gained during the discovery to select the next seed node. Furthermore, in this proposed protocol it is only the seed node that attempts the discovery and the information received is shared among the neighbourhood. However, the proposed protocol and algorithms show that the nodes only know their relative co-ordinates from the primary seed node. We conclude that the Farthest/Farthest algorithm is suggested as having better performances compared to the other algorithms. In addition, the location of the primary seed node can affect the performances of the algorithms. We suggest that the primary seed node located at the centre of the network achieves better performances.

7. REFERENCES

[1] W. Figel, N. Shepherd, and W. Trammell, "Vehicle Location by a Signal Attenuation Method," *IEEE Trans. Vehic. Tech.*, vol. VT-18, pp. 105–110, Nov. 1969.

[2] M. Hata and T. Nagatsu, "Mobile Location Using Signal Strength Measurements in a Cellular System," *IEEE Trans. Vehic. Tech.*, vol. VT-29, pp. 245–51, May 1980.

[3] J. Hightower, R. Want, and G. Borriello, "SpotON: An indoor3D location sensing technology based on RF signal strength," UW CSE2000-02-02, University of Washington, Seattle, WA, USA, February 2000.

[4] Bernhard Hofmann-Wellenhof, Herbert Lichtenegger, and James Collins Global positioning system: Theory and practice, 2nd. Springer -Verlag, 1992.

[5] J. Werb and C. Lanzl, "Designing a positioning system for finding things and people indoors," *IEEE Spectrum*, **35**(9), pp. 71–78, September 1998.

[6] R. Fleming and C. Kushner, "Low-power, miniature, distributed position location and communication devices using ultra-wideband, non-sinusoidal communication technology," *Aetherwire Inc., Semi-Annual Tech. Rep.*, ARPA Contract J-FBI-94-058, July 1995.

[7] D. D. McCrady, L. Doyle, H. Forstrom, T. Dempsy, and M. Martorana, "Mobile ranging with low accuracy clocks," *IEEE Trans. Microwave Theory Tech.*, vol. 48, pp. 951–957, June 2000.

[8] D. Niculescu, and B. Nath, " Ad Hoc Positioning System (APS) Using AOA," in *Proc. of IEEE INFOCOM* (Salt Lake City, UT), pp. 2037–2040, April 2003.

[9] M. Hata and T. Nagatsu, "Mobile Location Using Signal Strength Measurements in a Cellular System," *IEEE Trans. Vehic. Tech.*, vol. VT-29, pp. 245–51, May 1980.

[10] N. Bulusu., J. Heidemann, D. Estrin, and T. Tran, "Self-configuring localization systems: Design and experimental evaluation," *Trans. On Embedded Computing* Sys. 3(1), pp. 24-60, 2004.

[11] M. J. Mc Glynn, and S. A Borbash, "Birthday Protocols for Low Energy Deployment and Flexible Neighbour Discovery in Ad Hoc Wireless Network," in *Proceeding of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing*, 2001.

[12] Y. Xu, J. Heidemann and D. Estrin, "Geography-informed Energy Conservation for Adhoc Routing," in *Proceeding of the 7th Annual International Conference on Mobile Computing and Networking*, 2001.

[13] T. Salonidis, P. Bhagwat, and L. Tassiulas, "Proximity Awareness and Fast Connection Establishment in Bluetooth," *The 1st ACM Annual Workshop on Mobile Ad Hoc Networking and Computing* (MobiHoc 2000), August 2000.

[14] C. Law, A. K. Metha, and K.-Y. Siu, "Performance of a Bluetooth Scatternet Formation Protocol," *The 2nd ACM Annual Workshop on Mobile Ad Hoc networking and Computing* (MobiHoc 2001), October 2001.

[15] L. Doherty, K. Pister, and L. Ghaoui, "Convex position estimation in wireless sensor networks," in *Proc. IEEE INFOCOM*, April 2001.

[16] N. B. Priynatha, A. Chakraborty, and H. Balakrisnan, "The cricket Location-Support System," in *6th ACM International Conference on Mobile Computing and Networking* (ACM MOBICOM), August 2000.

[17] R. Want, A. Hopper, V. Falcao, and J. Gibbsons, “The Active Badge location System,” *ACM Transactions on Information System 10*, pp. 91-102, January 1992.

[18] A. Harter, and A. Hopper, “A New Location technique for the Active Office,” IEEE Personal Communication 4(5), pp. 42-47, October 1997.

[19] G. Welch, G. Bishop, L. Vicci, S. Brumback, K. Kelel, and D. Colluci, “The HiBall Tracker: High-Performance Wide-Area Tracking for Virtual and Augmented Environments,” *Symposium on Virtual Reality and Technology*, 1999.

[20] L. Guibas, D. Lin, J. C. Latombe, S LaVella, and R. Motwani, “Visibility-based pursuit evasion in a polynomial environment,” *International Journal of Computational Geometry Application*, 9(5), pp. 471-494, October 1999.

[21] M. Charikar, S. Guha, D. Shmoys, and E. Tardos, “ A constant factor approximation algorithm for the k median,” in *Proceeding of 31st Annual ACM Symposium on Theory of Computing* (STOC), pp. 1-10, May 1999.

[22] D. Shmoys, and F. A. Chudak, “Improved approximation algorithms for capacitated facility location problems,” in *Proceedings of 5th Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA), pp. S875-S876, 1999.

[23] S. Capkun, M. Hamdi, and J. Hubaux, “GPS-free positioning in mobile ad hoc networks,” in *Int. Conf. on System Sciences* (HICSS-34) pp 3481-3490, Maui, Hawaii, January 2001.

Detecting and preventing attacks using network intrusion detection systems

MeeraGandhi meera.gandhi@gmail.com
Department of Computer Science and Engg.,
ResearchScholar,
SathyabamaUniversity,

S.K.Srivatsa profsks@hotmail.com
Professor, ICE, St.Joseph's College of Engg., Chennai,

Abstract

Intrusion detection is an important technology in business sector as well as an active area of research. It is an important tool for information security. A Network Intrusion Detection System is used to monitor networks for attacks or intrusions and report these intrusions to the administrator in order to take evasive action. Today computers are part of networked; distributed systems that may span multiple buildings sometimes located thousands of miles apart. The network of such a system is a pathway for communication between the computers in the distributed system. The network is also a pathway for intrusion. This system is designed to detect and combat some common attacks on network systems. It follows the signature based IDs methodology for ascertaining attacks. A signature based IDS will monitor packets on the network and compare them against a database of signatures or attributes from known malicious threats. It has been implemented in VC++. In this system the attack log displays the list of attacks to the administrator for evasive action. This system works as an alert device in the event of attacks directed towards an entire network.

Key words: intruders, information security, real time IDS, attacks, signature

1. INTRODUCTION

With the development of network technologies and applications, network attacks are greatly increasing both in number and severity. As a key technique in network security domain, Intrusion Detection System (IDS) plays vital role of detecting various kinds of attacks and secures the networks. Main purpose of IDS is to find out intrusions among normal audit data and this can be considered as classification problem. Intrusion detection systems (IDS) are an effective security technology, which can detect, prevent and possibly react to the attack. It performs monitoring of target sources of activities, such as audit and network traffic data in computer or network systems, requiring security measures, and employs various techniques for providing security services. With the tremendous growth of network-based services and sensitive information on networks, network security is becoming more and more important than ever before.

Symantec in a recent report [1] uncovered that the number of fishing attacks targeted at stealing confidential information such as credit card numbers, passwords, and other financial information are on the rise, going from 9 million attacks in June2004 to over 33 millions in less than a year .One solution to this is the use of network intrusion detection systems (NIDS) [2], that detect attacks by observing various network activities. It is therefore crucial that such systems are accurate in identifying attacks, quick to train and generate as few false positives as possible.

This paper presents the scope and status of our research in misuse detection [2, 3]. Experimental results have demonstrated that this model is much more efficient in the detection of network intrusions, compared with network based techniques. Section 2 describes an overview of frequently occurring network attacks and discusses related research done so far, also presents the experimental results. Finally, section 3 provides the concluding remarks and future scope of the work. Section 4 briefs the references.

2. NETWORKING ATTACKS

A Network Intrusion Detection System is used to monitor networks for attacks or intrusions[5,6] and report these intrusions to the administrator in order to take evasive action. A large NIDS server can be set up on a backbone network, to monitor all traffic; or smaller systems can be set up to monitor traffic for a particular server, switch, gateway, or router. It has been shown in fig. 1.

Intrusion detection is needed in today's computing environment because it is impossible to keep pace with the current and potential threats and vulnerabilities in our computing systems. The environment is constantly evolving and changing field by new technology and the Internet. Intrusion detection products are tools to assist in managing threats and vulnerabilities in this changing environment. Threats are people or groups who have the potential to compromise your computer system. These may be a curious teenager, a disgruntled employee, or espionage from a rival company or a foreign government [4].

Attacks on network computer system could be devastating and affect networks and corporate establishments. We need to curb these attacks and Intrusion Detection System helps to identify the intrusions. Without an NIDS, to monitor any network activity, possibly resulting in irreparable damage to an organization's network

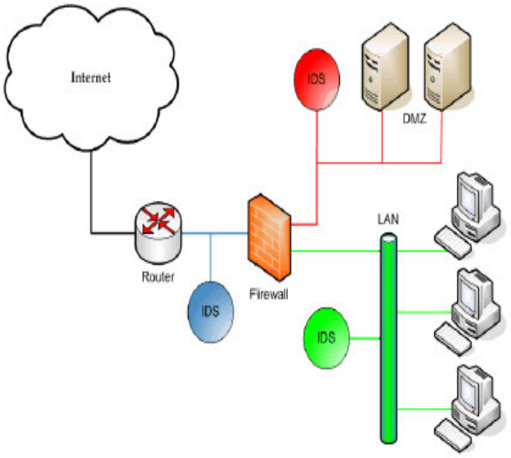


FIGURE 1: Computer network with Intrusion Detection Systems

Intrusion attacks [7, 8, and 9] are those in which an attacker enters your network to read, damage, and/or steal your data. These attacks can be divided into two subcategories: *pre intrusion activities and intrusions.*

2.1 Pre intrusion activities

Pre intrusion activities are used to prepare for intruding into a network. These include port scanning to find a way to get into the network and IP spoofing to disguise the identity of the attacker or intruder.

- **Port scans**: A program used by hackers to probe a system remotely and determine what TCP/UPD ports are open (and vulnerable to attack) is called a scanner. A scanner can find a vulnerable computer on the Internet, discover what services are running on the machine, and then find the weaknesses in those services. There are 65,535 TCP ports and an equal number of UDP ports. Stealth scanners use what is called an IP half scan, sending only initial or final packets instead of establishing a connection, to avoid detection.
- **IP spoofing**: This is a means of changing the information in the headers of a packet to forge the source IP address. Spoofing is used to impersonate a different machine from the one that actually sent the data. This can be done to avoid detection and/or to target the machine to which the spoofed address belongs. By spoofing an address that is a trusted port, the attacker can get packets through a firewall.

Various *intrusions* into the network are given as follows:

- **Source routing attack**: This is a protocol exploit that is used by hackers to reach private IP addresses on an internal network by routing traffic through another machine that can be reached from both the Internet and the local network [7, 8]. TCP/IP to allow those sending network data to route the packets through a specific network point for better performance supports source routing. Administrators to map their networks or to troubleshoot routing problems also use it.

- **Trojan attacks**: Trojans are programs that masquerade as something else and allow hackers to take control of your machine, browse your drives, upload or download data, etc. For example, in 1999, a Trojan program file called Picture.exe was designed to collect personal data from the hard disk of an infiltrated computer and send it to a specific e-mail address. So-called Trojan ports are popular avenues of attack for these programs.
- **Registry attack**: In this type of attack, a remote user connects to a Windows machine's registry and changes the registry settings. To prevent such an attack, configure permissions so that the every one group does not have access.
- **Password hijacking attacks**: The easiest way to gain unauthorized access to a protected system is to find a legitimate password. This can be done via social engineering (getting authorized users to divulge their passwords via persuasion, intimidation, or trickery) or using brute force method.

2.2 System Description

2.2.1 Packet Sniffer

This module involves capturing all traffic passing through the network. The sniffer will be installed on the end system in a network on which the traffic has to be captured. The sniffer[10] captures all network traffic by operating the network adapter in promiscuous mode.

2.2.2 Determination of attack signatures

Attack Signatures [13, 14] refers to the pattern of attack traffic. Signatures are modeled based on the packet header pattern a particular attack follows. It involves a count of packets from a particular target or a particular source or destination port or it may even be modeled with the help of other details in the packet such as header size, Time to Live (TTL), flag bits, protocol.

2.2.3 Identification of attacks

This involves extracting useful information from captured local traffic such as source and destination IP addresses, protocol type, header length, source and destination ports etc and compare these details with modeled attack signatures to determine if an attack has occurred.

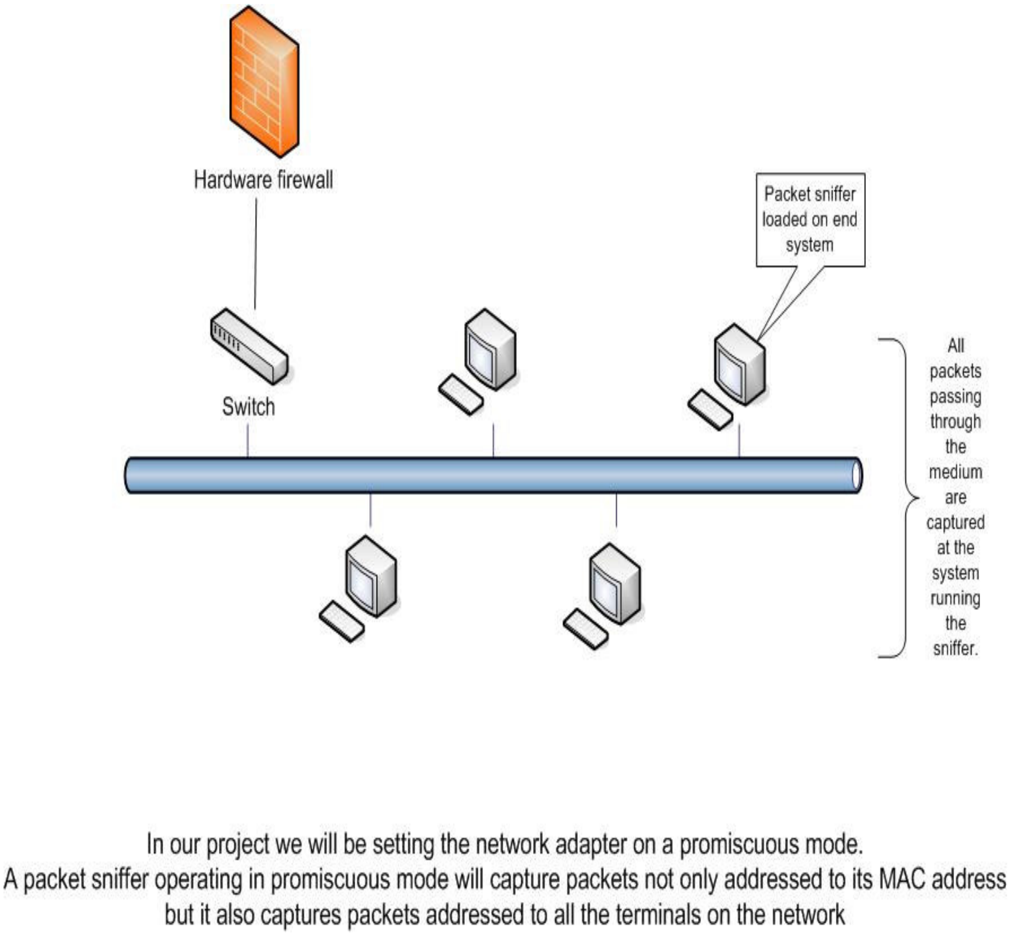
2.2.4 Reporting attack details

This involves reporting the attack to the administrator so that he may take evasive action. Reporting involves specifying attack details such as source and victim IP addresses, time stamp of attack and more importantly the type of attack.

2.3 Experimental Results

2.3.1 Signature based intrusion detection

Signature-based IDSs operate analogously to virus scanners, i.e. by searching a database of signatures for a known identity – or signature – for each specific intrusion event. In signature-based IDSs, monitored events are matched against a database of attack signatures to detect intrusions.


Hardware firewall
Packet sniffer loaded on end system
Switch
All packets passing through the medium are captured at the system running the sniffer.

In our project we will be setting the network adapter on a promiscuous mode.
A packet sniffer operating in promiscuous mode will capture packets not only addressed to its MAC address
but it also captures packets addressed to all the terminals on the network

FIGURE 2: IDS in Promiscuous mode

Signature-based IDS [15] are unable to detect unknown and emerging attacks since signature database has to be manually revised for each new type of intrusion that is discovered.

In addition, once a new attack is discovered and its signature is developed, often there is a substantial latency in its deployment across networks [13]. The most well known signature-based

IDS include SNORT [14], Network Flight Recorder [16], NetRanger [17], RealSecure [18], Computer Misuse Detection System (CMDS™) [20], NetProwler [21], Haystack [22] and MuSig (Misuse Signatures) [23].

This system follows the signature based IDs methodology for ascertaining attacks. A signature based IDS will monitor packets on the network and compare them against a database of signatures [19] or attributes from known malicious threats.

Most intrusion IDS are signature based. This means that they operate in much the same way as a virus scanner, by searching for a known attack or signature for each specific intrusion event. And, while signature-based IDS is very efficient at sniffing out known attack, it does, like anti-virus software, depend on receiving regular signature updates, to keep in touch with variations in hacker technique.

Because signature based IDS can only ever be as good as the extent of the signature database, two further problems immediately arise. Firstly, it is easy to fool signature-based solutions by changing the ways in which an attack is made. This technique simply skirts around the signature database stored in the IDS, giving the hacker an ideal opportunity to gain access to the network. This can be overcome by using defense in depth technique.

Secondly, the more advanced the signature database, the higher the CPU load for the system charged with analyzing each signature. Inevitably, this means that beyond the maximum bandwidth packets may be dropped. We have overcome these problems in our IDS system by using capture drivers that support network of up to 1 GBPS (Giga bits per second).

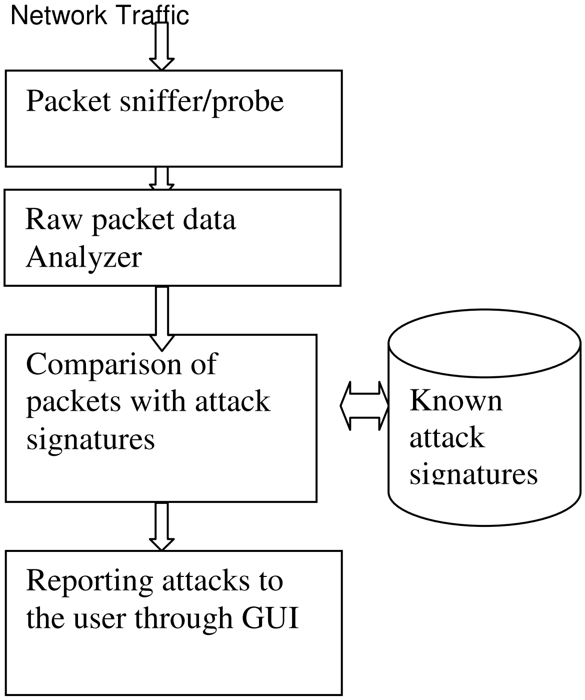

Network Traffic
Packet sniffer/probe
Raw packet data Analyzer
Comparison of packets with attack signatures
Known attack signatures
Reporting attacks to the user through GUI

FIGURE 3. – Implementation Architecture

2.3.2 Packet sniffing and promiscuous mode

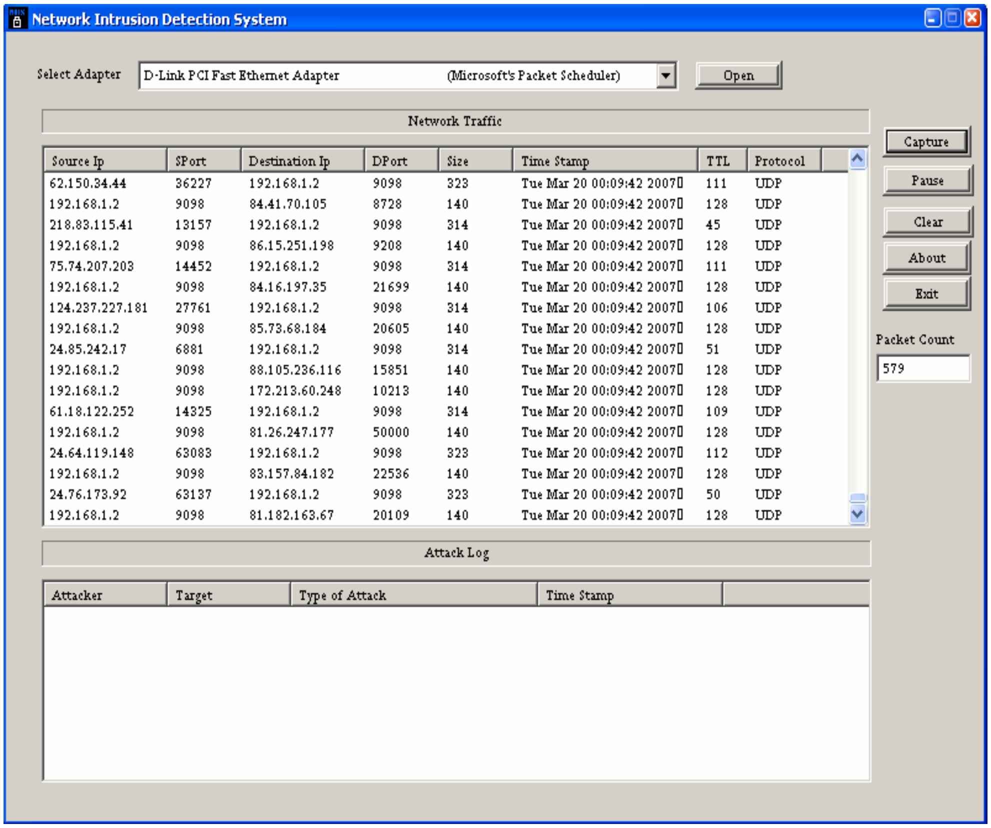
Packet sniffers generally require that a network interface is in promiscuous mode. The packet sniffer normally requires administrative privileges on the machine being used as a packet sniffer, so that the hardware of the network card can be manipulated to be in promiscuous mode is given in Figure 2.

This system uses a network probe to capture raw packet data and then we use this raw packet data to retrieve packet information such as source and destination IP address, source and destination ports, flags, header length, checksum, Time to Live (TTL) and protocol type. We then use this data and compare it with known attack signatures to identify threats to the network, shown in figure 3.The experimental results have been shown through screen shots in the figure 4 and 5

2.3.3 Attacks captured by software

IGMP KOD

An IGMP based denial-of-service attack that depletes the stack's large envelopes and also has source IP address spoofing. KOD (Kiss of Death) is a denial-of-service attack, which results in "Blue Screen" error message (so called "blue screen of death") or instantaneous reboot of computer. KOD send to victim's computer malformed IGMP (Internet Group Management Protocol) packets causing TCP/IP stacks to fail.



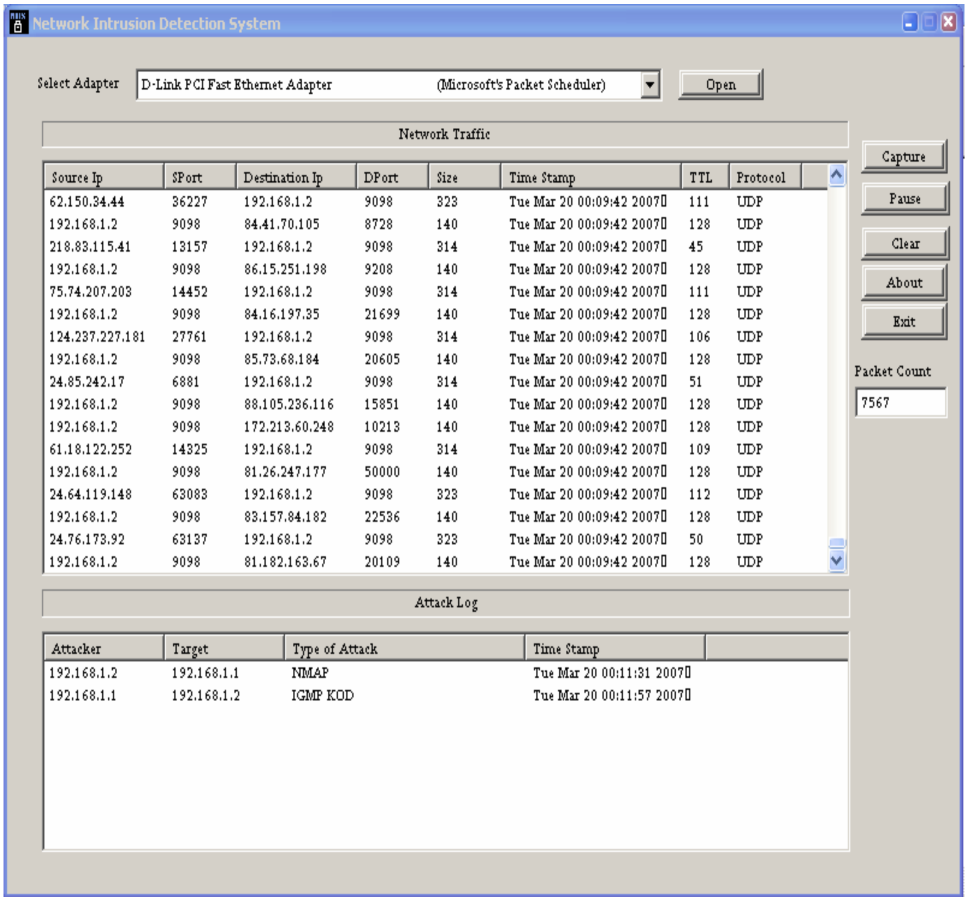
Select Adapter: D-Link PCI Fast Ethernet Adapter (Microsoft's Packet Scheduler)

Source Ip	SPort	Destination Ip	DPort	Size	Time Stamp	TTL	Protocol
62.150.34.44	36227	192.168.1.2	9098	323	Tue Mar 20 00:09:42 2007▯	111	UDP
192.168.1.2	9098	84.41.70.105	8728	140	Tue Mar 20 00:09:42 2007▯	128	UDP
218.83.115.41	13157	192.168.1.2	9098	314	Tue Mar 20 00:09:42 2007▯	45	UDP
192.168.1.2	9098	86.15.251.198	9208	140	Tue Mar 20 00:09:42 2007▯	128	UDP
75.74.207.203	14452	192.168.1.2	9098	314	Tue Mar 20 00:09:42 2007▯	111	UDP
192.168.1.2	9098	84.16.197.35	21699	140	Tue Mar 20 00:09:42 2007▯	128	UDP
124.237.227.181	27761	192.168.1.2	9098	314	Tue Mar 20 00:09:42 2007▯	106	UDP
192.168.1.2	9098	85.73.68.184	20605	140	Tue Mar 20 00:09:42 2007▯	128	UDP
24.85.242.17	6881	192.168.1.2	9098	314	Tue Mar 20 00:09:42 2007▯	51	UDP
192.168.1.2	9098	88.105.236.116	15851	140	Tue Mar 20 00:09:42 2007▯	128	UDP
192.168.1.2	9098	172.213.60.248	10213	140	Tue Mar 20 00:09:42 2007▯	128	UDP
61.18.122.252	14325	192.168.1.2	9098	314	Tue Mar 20 00:09:42 2007▯	109	UDP
192.168.1.2	9098	81.26.247.177	50000	140	Tue Mar 20 00:09:42 2007▯	128	UDP
24.64.119.148	63083	192.168.1.2	9098	323	Tue Mar 20 00:09:42 2007▯	112	UDP
192.168.1.2	9098	83.157.84.182	22536	140	Tue Mar 20 00:09:42 2007▯	128	UDP
24.76.173.92	63137	192.168.1.2	9098	323	Tue Mar 20 00:09:42 2007▯	50	UDP
192.168.1.2	9098	81.182.163.67	20109	140	Tue Mar 20 00:09:42 2007▯	128	UDP

Packet Count: 579

Attacker	Target	Type of Attack	Time Stamp

FIGURE 4: Screen shots 1



Select Adapter: D-Link PCI Fast Ethernet Adapter (Microsoft's Packet Scheduler)

Packet Count: 7567

Network Traffic

Source Ip	SPort	Destination Ip	DPort	Size	Time Stamp	TTL	Protocol
62.150.34.44	36227	192.168.1.2	9098	323	Tue Mar 20 00:09:42 2007▯	111	UDP
192.168.1.2	9098	84.41.70.105	8728	140	Tue Mar 20 00:09:42 2007▯	128	UDP
218.83.115.41	13157	192.168.1.2	9098	314	Tue Mar 20 00:09:42 2007▯	45	UDP
192.168.1.2	9098	86.15.251.198	9208	140	Tue Mar 20 00:09:42 2007▯	128	UDP
75.74.207.203	14452	192.168.1.2	9098	314	Tue Mar 20 00:09:42 2007▯	111	UDP
192.168.1.2	9098	84.16.197.35	21699	140	Tue Mar 20 00:09:42 2007▯	128	UDP
124.237.227.181	27761	192.168.1.2	9098	314	Tue Mar 20 00:09:42 2007▯	106	UDP
192.168.1.2	9098	85.73.68.184	20605	140	Tue Mar 20 00:09:42 2007▯	128	UDP
24.85.242.17	6881	192.168.1.2	9098	314	Tue Mar 20 00:09:42 2007▯	51	UDP
192.168.1.2	9098	88.105.236.116	15851	140	Tue Mar 20 00:09:42 2007▯	128	UDP
192.168.1.2	9098	172.213.60.248	10213	140	Tue Mar 20 00:09:42 2007▯	128	UDP
61.18.122.252	14325	192.168.1.2	9098	314	Tue Mar 20 00:09:42 2007▯	109	UDP
192.168.1.2	9098	81.26.247.177	50000	140	Tue Mar 20 00:09:42 2007▯	128	UDP
24.64.119.148	63083	192.168.1.2	9098	323	Tue Mar 20 00:09:42 2007▯	112	UDP
192.168.1.2	9098	83.157.84.182	22536	140	Tue Mar 20 00:09:42 2007▯	128	UDP
24.76.173.92	63137	192.168.1.2	9098	323	Tue Mar 20 00:09:42 2007▯	50	UDP
192.168.1.2	9098	81.182.163.67	20109	140	Tue Mar 20 00:09:42 2007▯	128	UDP

Attack Log

Attacker	Target	Type of Attack	Time Stamp
192.168.1.2	192.168.1.1	NMAP	Tue Mar 20 00:11:31 2007▯
192.168.1.1	192.168.1.2	IGMP KOD	Tue Mar 20 00:11:57 2007▯

FIGURE 5: Screen shots 2.

DOS attack

In computer security, a denial-of-service attack (DOS) is an attempt to make a computer resource unavailable to its intended users. Typically the targets are high-profile web servers, and the attack attempts to make the hosted web pages unavailable on the internet. It is a computer crime that violates the Internet proper use policy as indicated by the internet Architecture Board (IAB).

DOS attacks have two general forms:

i) Force the victim computer(s) to reset or consume its resources such that it can no longer provide its intended service.
ii) Obstruct the communication media between the intended users and the victim so that they can no longer communicate adequately.

A denial-of-service attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service. Examples include:

- flooding a network, thereby preventing legitimate network traffic;
- Disrupting service to a specific system or person.

- Attacks can be directed at any network device, including attacks on routing devices and web, electronic mail, or Domain Name System Servers.
- Consumption of computational resources, such as bandwidth, disk space, or CPU time

DOS conseal

Vulnerability exists in the conseal firewall product that causes the vulnerable system to reboot or lock up when a large number of spoofed UDP packets are received by the firewall. The way this attack kills the machine happens in 2 ways

- If Conseal is set for "learning" mode the flooding packets from all the different IPs and ports will cause the program to continuously attempt to write more and more new rules. This eventually uses up all the resources and results in a freeze and eventually a reboot.
- If Conseal is set to log attacks, once again because of the number of packets the system resources are eaten up and the machine dies.

DOS bloop

It is a denial Of Service attack that sends random spoofed ICMP packets. ICMP flooding is probably the most common type of Denial of Service attack, since nearly all websites reply to ICMP packets, its easy to use ICMP flooding to shut them down. The result of the attack is freezes the users machine or a CPU usage will rise to extreme lag potential.

ICMP flooding works by sending a lot of ICMP packets to the target machine, for each packet sent the remote computer has to reply to each one, meaning it would exhaust the machines bandwidth so a legitimate user could not access the server. ICMP packets are better known as “Pings”, they are used to see if a remote computer is online.

NMAP

NMAP was the source of strange new scan patterns started being detected by the SHADOW ID Systems located throughout the Internet. This scan’s signature is characterized by SYN packets sent to apparently random destination ports over some discreet range of values. At the end of these scans we typically see several packets to high numbered TCP and UDP ports, followed by a small number of packets to a common destination port. The two basic scan types used most in NMAP [8,9] are TCP connect () scanning and SYN scanning also known as half-open, or stealth scanning.

DNS solinger

Berkeley Internet Name Domain (BIND) is an implementation of the Domain Name System (DNS) protocols that provides an openly re-distributable reference implementation of the major components of the Domain Name System. BIND SOLINGER vulnerability could allow remote attackers to hang the service for periods up to 120 seconds by initiating abnormal TCP connections to the daemon. In some systems, it is possible to set the system wide solinger timeout to a lower value, however this may have unexpected consequences with other applications.

2.4 Testing tool

We have used Karalon traffic IQ professional [11, 24] for testing our software with intrusion attacks. Traffic IQ Professional provides a unique industry approved software solution for auditing and testing the recognition and response capabilities of Intrusion detection systems.
Features include

- Traffic Replay
- Traffic scan list
- Reporting
- Traffic file editor
- Command prompt

- Traffic library

3. CONCLUSION

We have successfully created a network based intrusion detection system with signature IDS methodology. It successfully captures packets transmitted over the entire network by promiscuous mode of operation and compares the traffic with crafted attack signatures. The attack log displays the list of attacks to the administrator for evasive action. This system works as an alert device in the event of attacks directed towards an entire network It has functionality to run in the background and monitor the network.

It also incorporates functionality to detect installed adapters on the system, selecting adapter for capture, pause capture and clearing captured data is shown in *the screen shots*. It may be incorporated with further signatures for attacks. This system could be used as a stand alone for providing attack alerts to the administrator or it can be used as a base system for developing a network intrusion prevention system. The types of attacks share the characteristic that upon their initiation and while they are in progress, Global attack and of distributed intrusion detection processes produce sufficient network traffic (e.g. port scanning) so that local detectors can find sufficient evidence of the attack and report the attacks.

4. REFERENCES

[1] "Symantec-Internet Security threat report highlights (Symantec.com)", http://www.prdomain.com/companies/Symantec/newreleases/Symantec_internet_205032.htm

[2] Symantec Security Response, W32.ExploreZip.L.Worm,

http://securityresponse.symantec.com/avcenter/venc/data/w32.explorezip.l.worm.html , January 2003.

[3] Komninos T., Spirakis P.: Dare the Intruders, Ellinika Grammata and CTI Press (2003).

[4] E. Biermann, E.Cloete, L.M. Venter, A comparison of Intrusion detection systems, *Computers and* Security, 20(2001)8, 676–683.

[5] P. Ning and D. Xu. Hypothesizing and reasoning about attacks missed by intrusion detection systems. ACM Transactions on Information and System Security, 7(4):591– 627, November 2004

[6] Herringshaw, C. (1997) 'Detecting attacks on networks', IEEE Computer Society Vol.30, pp.16 – 17.

[7] International Standard IS0 7498.2, Information processing system - Open system interconnection – Basic reference model, PaR *2:* Security architecture, 1989.

[8] D. Oollmann, *Cornpuler Security,* John Wiley & Sons, 1999.

[9] R.G. Bace, *Intrusion* Detection. Macmillan Technical Publishing, 2000

[10] http://www.winpcap.org/ - Obtained drivers for packet capture with wpcap.dll and packet.dll driver.

[11] http://www.karalon.com - Obtained Karalon IQ professional tool for testing our network intrusion
detection system.

[12] http://www.securityfocus.com – White papers for intrusion detection techniques and methodologies.

[13] R. Lippmann, The Role of Network Intrusion Detection, In *Proceedings of the Workshop on Network*
Intrusion Detection, H.E.A.T. Center, Aberdeen, MD, March 19-20, 2002.

[14] SNORT Intrusion Detection System, www.snort.org, 2004.

[15] Snort-Wireless Intrusion Detection, http://snort-wireless.org, 2003.]

[16] NFR Network Intrusion Detection, http://www.nfr.com/products/NID/, 2001.

[17] Cisco Systems, Inc., NetRanger-Enterprise-scale, Real-time, Network Intrusion Detection System,
http://www.cisco.com/univercd/cc/td/doc/product/iaabu/netrangr/, 1998.

[18] Internet Security Systems, Inc., RealSecure, http://www.iss.net/prod/rsds.html, 1997.

[19] Intrusion.com, Intrusion SecureHost, white paper available at: www.intrusion.com/products/hids.asp ,
2003.

[20] J. Van Ryan, SAIC's Center for Information Security, Technology Releases CMDS Version 3.5,

http://www.saic.com/news/may98/news05-15-98.html, 1998.

[21] N. Weaver, V. Paxson, S. Staniford and R. Cunningham, A Taxonomy of Computer Worms, In
Proceedings of the The Workshop on Rapid Malcode (WORM 2003), held in conjunction with the
10th ACM Conference on Computer and Communications Security, Washington, DC, October
27, 2003.

[22] Wheel Group Corporation, Cisco Secure Intrusion Detection System,

http://www.cisco.com/univercd/cc/td/doc/product/iaabu/csids/index.htm , 2004

[23] Patwardhan, A. Parker, J., Joshi,A., Karygiannis, A., and Iorga,M. “Secure Routing and Intrusion
Detection in Ad Hoc Networks”, *Third IEEE International Conference on Pervasive Computing and*
Communications, Kauai Island, Hawaii, 2005.

[24] Komninos T, Spirakis P., Stamatiou et.al..: A Software Tool for Distributed Intrusion Detection in Computer Networks (Helena) (Best Poster presentation in PODC 2004).

IMPLEMENTATION OF ECHOSTATE NETWORK IN NIDS

[1]**Meera Gandhi,**
Research scholar,
Department of Computer science and engg.
Sathyabama University,
Sholinganallur ,
Chennai-600100 meera.gandhi@gmail.com

[2]**S.K.Srivatsa,**
Professor,
ICE, St.Joseph's College of Engg.,
Chennai – 600119 profsks@hotmail.com

Abstract

Identifying instances of network attacks by comparing current activity against the expected actions of an intruder has become an important. Most current approaches to misuse detection involve the use of rule-based expert systems to identify indications of known attacks. Artificial neural networks provide the potential to identify and classify network activity based on limited, incomplete, and nonlinear data sources. Transmission of data over the internet keeps on increasing, which needs to protect connected systems also increasing. Intrusion Detection Systems (IDSs) are the latest technology used for this purpose. Although the field of IDSs is still developing, the systems that do exist are still not complete, in the sense that they are not able to detect all types of intrusions. Some attacks which are detected by various tools available today cannot be detected by other products, depending on the types and methods that they are built on. In this work, an artificial neural network using echo state network algorithm has been used to implement the IDS. This paper proposes an approach to implement recurrent echo state network real time IDS. Twenty four packet information both normal and intrusion have been considered for training. Testing has been done with new sets of packet information. The result of intrusion detection (ID) is very close to 90%. The topology of the echo state network is (41 X 20 X 1). The network converged with 24 iterations. However, very huge amount of packets are to be evaluated to know the complete performance of the developed system.

Keywords: Echo state network, Intrusion detection, misuse detection, neural networks, computer security

1. INTRODUCTION

The complexity, as well as the importance, of distributed computer systems and information resources is rapidly growing. Due to this, computers and computer networks are often exposed to computer crime. Many modern systems lack properly implemented security services; they contain a variety of vulnerabilities and, therefore, can be compromised easily. As network attacks have increased in number over the past few years, the efficiency of security systems such as firewalls have declined.

It is very important that the security mechanisms of a system are designed to prevent unauthorized access to system resources and data. Building a complete secure system is impossible and the least that can be done is to detect the intrusion attempts so that action can be taken to repair the damage later. Organizations are increasingly implementing various systems that monitor IT security breaches. Intrusion detection systems (IDSs) have gained a considerable amount of interest within this area. The main task of IDS is to detect an intrusion and, if necessary or possible, to undertake some measures eliminating the intrusions.

Because most computer systems are vulnerable to attack, intrusion detection (ID) is a rapidly developing field. Intrusion Detection Systems (IDSs) detect intrusions using specific methodologies that are specific to each of them. A method describes how an IDS analyzes data to detect possible intrusions, based on the analysis approaches. The analysis approaches are anomaly detection and misuse detection. There are many methods that are used. Examples of them include statistical approaches, protocol anomaly detection, neural networks [4-6][14][18][20][24] , file checking, expert systems , rule-based measures[19], and genetic algorithms (GAs) [26][35].

2. BACKGROUND

Intruders tend to find new ways to compromise systems each day. As more intrusions occur, the weaknesses of existing technologies like firewalls are exposed. Since it is impossible to build a complete secure system, IDSs are used to detect the intrusions that occur. This is why IDSs are gaining acceptance in every organization. To understand what an IDS is, first one should know what intrusion and intruders are. Intrusion is the unauthorized attempt to access information, manipulate information, or render a system unreliable or unusable. To detect intrusions and to prevent them, one has to be aware of how an intruder can cause intrusions.

The primary ways an intruder can get into the system is through primary intrusion, system intrusion and remote intrusion. ID is the process of monitoring the events occurring in a computer system or network, and analyzing them for intrusions. The prevention of intrusions should be done through effective IDSs. An IDS is a software or hardware product that automates this monitoring and analysis process.

The types of IDSs can be described in terms of three fundamental functional components. They are the information source, analysis and response. The information source of the system mainly depends on where the IDSs are being placed, hence it is also known as the monitoring locations of the IDS. The information sources are mainly of three types: network-based IDSs, host-based IDSs and application-based IDSs [Bace, 2002]. Since the focus of this work is on network-based IDS, the other two types will not be considered here. Network-based IDSs detect attacks by capturing and analyzing network packets.

They search for attack signatures within the packets. Signatures might be based on actual packet contents, and are checked by comparing bits to known patterns of attack. If the bits are matched to known patterns of attack, then an intrusion is triggered. Once the information sources have monitored network traffic, the next step is to analyze the events to detect the intrusion. The two main techniques or approaches used to analyze events to detect attacks are misuse detection and anomaly detection. Response is the set of actions that the system takes once it detects intrusions. Some of the responses [11-13] involve reporting results and findings to a pre-specified location, while others are more actively automated responses.

Commercial IDSs support both active and passive responses, and sometimes a combination of the two. IDSs can be viewed as the second layer of protection against unauthorized access to networked information systems because despite the best access control systems, intruders are still able to enter computer networks. IDSs expand the security provided by the access control systems by providing system administrators with a warning of the intrusion.

They also provide the system administrators with necessary information about the intrusions. This assists the system administrators in controlling the intrusions that has occurred, in order to avoid them in the future or to minimize the damage that may occur due to an intrusion [15-17]. Although IDSs can be designed to verify the proper operation of access control systems by looking for the attacks that get past the access control systems, IDSs are more useful when they can detect intrusions that use methods that are different from those used by the access control systems. For this purpose, they must use more general and more powerful methods than simple database look-ups of known attack scenarios.

2.1. Signature basics

A network IDS signature is a pattern that we want to look for in traffic. Some of the methods that can be used to identify each one:

- Connection attempt from a reserved IP address. This is easily identified by checking the source address field in an IP header.

- Packet with an illegal TCP flag combination. This can be found by comparing the flags set in a TCP header against known good or bad flag combinations.

- Email containing a particular virus. The IDS can compare the subject of each email to the subject associated with the virus-laden email, or it can look for an attachment with a particular name.

- DNS buffer overflow attempt contained in the payload of a query. By parsing the DNS fields and checking the length of each of them, the IDS can identify an attempt to perform a buffer overflow using a DNS field. A different method would be to look for exploit shell code sequences in the payload.

- Denial of service attack on a POP3 server caused by issuing the same command thousands of times. One signature for this attack would be to keep track of how many times the command is issued and to alert when that number exceeds a certain threshold [21].

- File access attack on an FTP server by issuing file and directory commands to it without first logging in. A state-tracking signature could be developed which would monitor FTP traffic for a successful login and would alert if certain commands were issued before the user had authenticated properly [22-23].

2.1.1 Purpose of Signatures

Different signatures have different goals. The obvious answer is that, want to be alerted when an intrusion attempt occurs. Why we might want to write or modify a signature. Perhaps seeing some odd traffic on network and want to be alerted the next time it occurs. It has been noticed that it has unusual header characteristics, and want to write a signature that will match this known pattern. Perhaps are interested in configuring IDS to identify abnormal or suspicious traffic in general, not just attacks or probes. Some signatures may tell which specific attack is occurring or what vulnerability the attacker is trying to exploit, while other signatures may just indicate that unusual behavior is occurring, without specifying a particular attack. It will often take significantly more time and resources to identify the tool that’s causing malicious activity, but it will give more information as to why being attacked and what the intent of the attack is.

2.1.2 Header Values

Simple signature characteristic header values are presented. Some header values are clearly abnormal, so they make great candidates for signatures. A classic example of this is a TCP packet with the SYN and FIN flags set. This is a violation of request for comments (RFC 793) (which defines the TCP standard), and has been used in many tools in an attempt to circumvent firewalls, routers and intrusion detection systems. Many exploits include header values that purposely violate RFCs, because many operating systems and applications have been written on the assumption that the RFCs would not be violated and don't perform proper error handling of such traffic.

Many tools either contain coding mistakes or are incomplete, so that crafted packets produced by them contain header values that violate RFCs. Both poorly written tools and various intrusion techniques provide distinguishing characteristics that can be used for signature purposes [25]. There's a catch. Not all operating system (OS) and applications completely adhere to the RFCs. In fact, many have at least one facet of their behavior that violates an RFC. Over time, protocols may implement new features that are not included in an RFC.

New standards emerge over time, which may "legalize" values that were previously illegal; RFC 3168, for Explicit Congestion Notification (ECN), is a good example of this. So an IDS signature based strictly on an RFC may produce many false positives. Still, the RFCs make a great basis for signature development, because so much malicious activity violates RFCs. Because of RFC updates and other factors, it's important to review and update existing signatures periodically [27] [29].

2.1.3 Sample signature

- Various source IP addresses
- TCP source port 21, destination port 21
- Type of service 0
- IP identification number 39426
- SYN and FIN flags set
- Various sequence numbers set
- Various acknowledgment numbers set
- TCP window size 1028

Packet values that are completely normal don't make good signature characteristics by themselves, although they are often included to limit the amount of traffic that we study. By including the normal IP protocol value of 6 for a protocol, so that only check TCP packets. But other characteristics that are completely normal, such as the type of service set to 0, are much less likely to be helpful in signature development.

A signature based on few suspicious characteristics may be too specific [32-34]. Although it would provide much more precise information about the source of the activity, it would also be far less efficient than a signature that only checks one header value. Signature development is always a tradeoff between efficiency and accuracy. In many cases, simpler signatures are more prone to false positives than more complex signatures, because simpler signatures are much more general [36][39-40]. But more complex signatures may be more prone to false negatives than simpler signatures, because one of the characteristics of a tool or methodology may change over time.

2.1.4. Schematic flow diagram

The sequence of steps required for the IDS by implementing an ESNN has been given in Figure 1

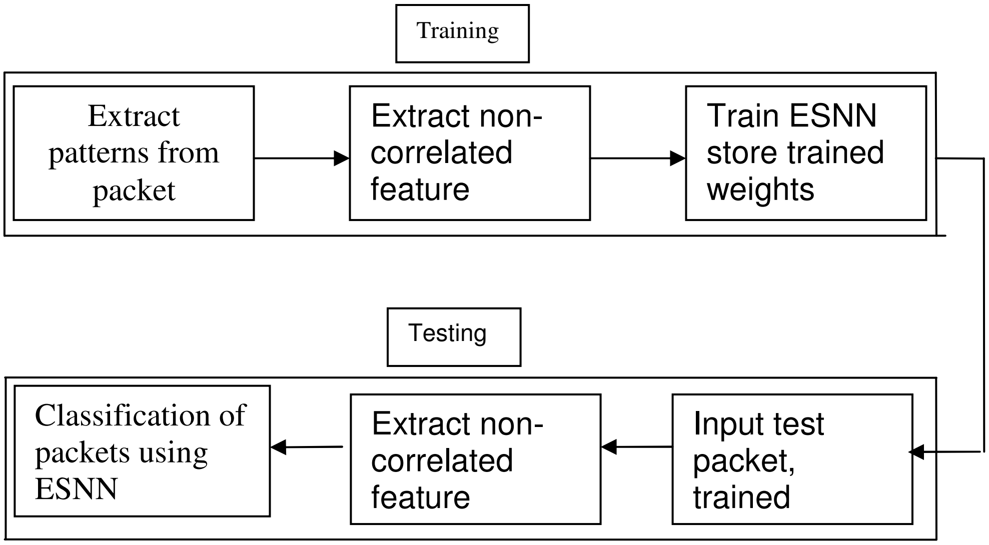

Training
Extract patterns from packet
Extract non-correlated feature
Train ESNN store trained weights
Testing
Classification of packets using ESNN
Extract non-correlated feature
Input test packet, trained

FIGURE.1 Schematic diagram of the IDS

After training the ESNN, a set of final weights are obtained and stored in a file. During the process of testing, a test packet converted into uncorrelated features followed by classification by ESNN using final weight values arrived during training phase.

2.2. Echo State Neural Network (ESNN)

An artificial neural network (ANN) is an abstract simulation of a real nervous system that contains a collection of neuron units, communicating with each other via axon connections. Such a model bears a strong resemblance to axons and dendrites in a nervous system. Due to this self-organizing and adaptive nature, the model offers potentially a new parallel processing paradigm. This model could be more robust and user-friendly than the traditional approaches. ANN can be viewed as computing elements, simulating the structure and function of the biological neural network. These networks are expected to solve the problems, in a manner which is different from conventional mapping. Neural networks are used to mimic the operational details of the human brain in a computer. Neural networks are made of artificial 'neurons', which are actually simplified versions of the natural neurons that occur in the human brain. A neural architecture comprises massively parallel adaptive elements with interconnection networks, which are structured hierarchically.

Artificial neural networks are computing elements which are based on the structure and function of the biological neurons [3]. These networks have nodes or neurons which are described by difference or differential equations. The nodes are interconnected layer-wise or intra-connected among themselves. Each node in the successive layer receives the inner product of synaptic weights with the outputs of the nodes in the previous layer [1], [8-9]. The inner product is called the activation value

Dynamic computational models require the ability to store and access the time history of their inputs and outputs. The most common dynamic neural architecture is the time-delay neural network that couples delay lines with a nonlinear static architecture where all the parameters (weights) are adapted with the back propagation algorithm. Recurrent neural networks (RNNs) implement a different type of embedding that is largely unexplored. One of the main practical problems with RNNs is the difficulty to adapt the system weights. Back propagation through time and real-time recurrent learning; have been proposed to train RNNs. These algorithms suffer from computational complexity, resulting in slow training, complex performance surfaces, the possibility of instability, and the decay of gradients through the topology and time. The problem of decaying gradients has been addressed with special processing elements (PEs).

ESNN [28],[30-31] 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 topology of the network is shown in Figure 2. 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 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.

The echo state condition is defined in terms of the spectral radius (the largest among the absolute values of the eigen values of a matrix, denoted by (|| . ||) of the reservoir's weight matrix (|| W || < 1). This condition states that the dynamics of the ESNN is uniquely controlled by the input, and the effect of the initial states vanishes. The current design of ESNN parameters relies on the selection of spectral radius. There are many possible weight matrices with the same spectral radius.

ESNN is composed of two parts (Jaeger, H 2002): a fixed weight (|| W || < 1) recurrent network and a linear readout. The recurrent network is a reservoir of highly interconnected dynamical components, states of which are called echo states. The memory less linear readout is trained to produce the output. The recurrent discrete-time neural network is given in with M input units, N internal PEs, and L output units.

The value of the input unit at time n is
$u(n) = [u_1(n), u_2(n), . . . , u_M(n)]^T$, (1)

The internal units are
$x(n) = [x_1(n), x_2(n), . . . , x_N(n)]^T$, and (2)

Output units are
$y(n) = [y_1(n), y_2(n), . . . , y_L (n)]^T$. (3)

The connection weights are given

- in an N x M weight matrix $W^{back} = W_{ij}^{back}$ for connections between the input and the internal PEs,
- in an N × N matrix $W^{in} = W_{ij}^{in}$ for connections between the internal PEs
- in an L × N matrix $W^{out} = W_{ij}^{out}$ for connections from PEs to the output units and
- in an N × L matrix $W^{back} = W_{ij}^{back}$ for the connections that project back from the output to the internal PEs.

Here
M is the no. of neurons in the input layer
N is the no. of neurons in the hidden layer and
L is the no. of neurons in the output layer
The activation of the internal PEs (echo state) is updated by using the relation

$$x(n + 1) = f(W^{in} u(n + 1) + Wx(n) +W^{back}y(n)),$$

(4)

where
$f = (f_1, f_2, . . . , f_N)$ are the internal PEs' activation functions.

All f$_i$'s are hyperbolic tangent function $\frac{e^x - e^{-x}}{e^x + e^{-x}}$. The output from the readout network is computed as follows

$y(n + 1) = f^{out}(W^{out}x(n + 1))$, . (5)

where

$f^{out} = (f_1^{out}, f_2^{out}, ..., f_L^{out})$ are the output unit's of nonlinear functions.

The ESNN topology specified in this work is {41 x no. of reservoirs x 1}, where two nodes are in the input layer, one in the output layer and any number of reservoirs in the hidden layer. The connections between input-hidden layers, hidden-output layer are initialized with random numbers. The training of the ESNN is done with choosing initial random weights in a range of 0.25 to 0.55. The random weights are chosen within a small range for easier quicker settlement of final weights and also to prevent the network from further oscillation.

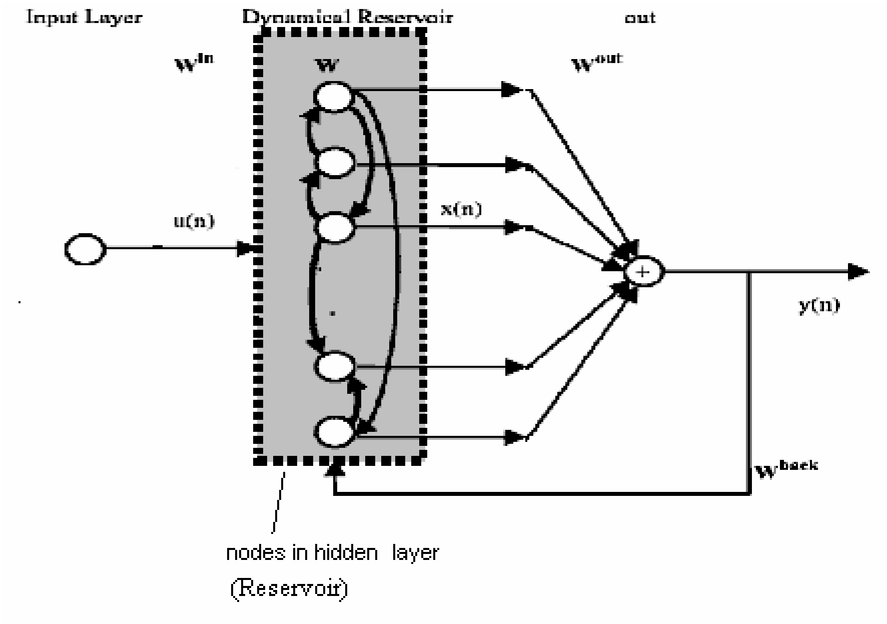


FIGURE.2: An echo state neural network (ESNN)

2.2.1 Implementation of IDS using ESNN

To obtain the final trained weights by training the ESNN

Step 1: Find the uncorrelated features of the packet
Step 2: Fix the target values (class label)
Step 3: Set the no. of inputs, no. of reservoirs, and no. of outputs
Step 4: Initialize connection matrices-using random weights for
no.of reservoirs versus no. of inputs,
no.of outputs versus no. of reservoirs,
no. of reservoirs versus no. of reservoirs
Step 5: Determine values of matrices less than a threshold for updating the weights
Step 6: Normalize the reservoir matrix by finding its eigen value.
Step 7: The initial state matrix is updated with tanh() function. The inputs for the tanh() function are
{input pattern X weights between input and hidden layer **+**
desired output X weights between output and hidden layer +

normalized reservoir matrix}
Step 8: Store the weight matrices.

2.2.2 Implementation of ESNN for IDS using the trained weights of ESNN

Step 1: The trained weights are given as inputs.
Step 2: Apply the uncorrelated features of the packet
Step 3: Process the inputs with the trained weights
Step 4: Employ transfer function to get the output of the ESNN
Step 5: Set threshold and classify intrusion or not.
Overall algorithms for training and testing

1. Obtain the uncorrelated feature of the packet
2. Train the ESNN with the inputs and target outputs. Trained weights are obtained once all the Patterns are presented to the ESNN.
3. Test the ESNN with a new packet, and classify for intrusion.

2.3 Network Intrusion Detection System (NIDS) using ANN

2.3.1 Packet Capture

Packet capture is the beginning process in NIDS. It can be implemented by setting the working mode of the network card as the promiscuous mode. The network card under common mode can only receive the packet whose destination address is the network card itself. Only those packets are not sufficient to serve for the data source of the NIDS. So it is necessary to set the network card's working mode as the promiscuous mode. Under this mode, the network card can receive not only the packets sent to itself but also the packets routed to some other hosts. Thus the NIDS can monitor the network stream of all hosts of some local area network and detect whether intrusion happens or not.

2.3.2 Feature Extraction

Feature selection and extraction [37], [38] is one of the pivotal problems in implementing the intrusion detection system. Network stream itself is not suitable directly as the input for the ESNN, so it is necessary to extract some features from the network stream. The uncorrelated features extracted from the network stream form a feature vector which serves for the description of the packet. Whether the feature vector can describe the network stream correctly and efficiently or not has a large effect on the efficiency and correctness of the NIDS.

Selecting several features such as the protocol code, the packet head length, the checksum, the port number and some TCP Flags, etc have been done. Based on these features, a vector is obtained as follows to describe an intrusion. The following representation is some of the intrusion types which contain some sequence of intrusion appearance

Attack(type)=(P-id, H-Len, C-sum ,S-port, D-port, ICMP-type ,ICMP-Code, Flag, P-Len, P-data)

Above is the general description form of an abstract attack.

Perhaps some concrete examples can explain the vector well.

Attack (CGI)=(Tcp,32, O, 2345, 80, null, null, A, 421, get CGI-bin)

Attack(FTP)=(TCP, 24, 16, 21, 21, null, null, PA, 256, ROOM)

Attack(Redirect)=(ICMP, 20, null, null, 8, 3, null, 192, la)

Attack(UDP)=(UDP, 16, 10, 138, 126, null, null, nuli,448,3c)

If the features of a packet are found as any of the above, it represents a CGI attack; a FTP attack, a REDIRECT attack, and a UDP attack respectively. The feature vector will serve for the input of the ESNN Classifier., then the ESNN Classifier will judge whether the feature vector represents an intrusion or not.

2.3.3. Experimental Setup

The simulation results were obtained from the standard KDD data set. It is a well defined as normal and with different types of attack for TCP, UDP, ICMP, etc. A set of sample data set is shown in Table 1. Each row is a pattern. The fields in each pattern describe the properties of respective packet. The various attacks considered during training are

back	dos
buffer_overflow	u2r
ftp_write	r2l
guess_passwd	r2l
imap	r2l
ipsweep	probe
land	dos
loadmodule	u2r
multihop	r2l
neptune	dos
nmap	probe
perl	u2r
phf	r2l
pod	dos
portsweep	probe
rootkit	u2r
satan	probe
smurf	dos
spy	r2l
teardrop	dos
warezclient	r2l
warezmaster	r2l

Table 1. Sample KDD dataset

S.no	Packet details
1	0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,254,1.00,0.01,0.00,0.00,0.00,0.00,0.00,0.00,normal.
2	0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,254,1.00,0.01,0.00,0.00,0.00,0.00,0.00,0.00,normal.
3	0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,254,1.00,0.01,0.00,0.00,0.00,0.00,0.00,0.00,normal.
4	0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,254,1.00,0.01,0.00,0.00,0.00,0.00,0.00,0.00,snmpgetattack.
5	0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,254,1.00,0.01,0.01,0.00,0.00,0.00,0.00,0.00,snmpgetattack.
6	0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,255,1.00,0.00,0.01,0.00,0.00,0.00,0.00,0.00,snmpgetattack.
7	0,udp,domain_u,SF,29,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,1,0.00,0.00,0.00,0.00,0.50,1.00,0.00,10,3,0.30,0.30,0.30,0.00,0.00,0.00,0.00,0.00,normal.
8	0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,253,0.99,0.01,0.00,0.00,0.00,0.00,0.00,0.00,normal.

9	0,udp,private,SF,105,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255, 254,1.00,0.01,0.00,0.00,0.00,0.00,0.00,0.00,snmpgetattack.
10	0,tcp,http,SF,223,185,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,4,4,0.00,0.00,0.00,0.00,1.00,0.00,0.00,71,255, 1.00,0.00,0.01,0.01,0.00,0.00,0.00,0.00,normal.

Instead of KDD data set, free sniffer software's like network sniffer, packet sniffer and more software's can be used to extract the values of a packet, which can be further labeled as normal or an attack to be used for training. The contents of the packet should be suitably modified into meaningful numerical values. A sample dataset used for training is shown in Table 2.

Table 2 . Sample dataset used for training

S.No	**Patterns used for training** **Input to ESNN after uncorrelating the features of patterns**	**Target outputs**
1	0 .2 .01 .1 105 146 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0.00 0.00 0.00 0.00 1.00 0.00 0.00 255 254 1.00 0.01 0.00 0.00 0.00 0.00 0.00 0.00	.1
2	0 .2 .01 .1 105 146 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0.00 0.00 0.00 0.00 1.00 0.00 0.00 255 254 1.00 0.01 0.00 0.00 0.00 0.00 0.00 0.00	.1
3	0 .2 .01 .1 105 146 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0.00 0.00 0.00 0.00 1.00 0.00 0.00 255 254 1.00 0.01 0.00 0.00 0.00 0.00 0.00 0.00	.1
4	0 .2 .01 .1 105 146 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0.00 0.00 0.00 0.00 1.00 0.00 0.00 255 254 1.00 0.01 0.00 0.00 0.00 0.00 0.00 0.00	.2
5	0 .2 .01 .1 105 146 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0.00 0.00 0.00 0.00 1.00 0.00 0.00 255 254 1.00 0.01 0.01 0.00 0.00 0.00 0.00 0.00	.2

2.3.4. Results and Discussion

The topology of ESNN used is 41 X 20 X 1; no. of nodes in the input layer is 41, no. of nodes in the hidden layer is 20 and no. of nodes in the output layer is 1. The labeling is set as 0.1 (Normal) or 0.2(attack). It is mandatory to use huge amount of patterns to be presented for training ESNN. However, it would take enormous amount of time for the ESNN to learn the patterns. Hence, only 24 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. Figure 3 shows the performance of the ESNN .The x axis represents the packets. It is a combination of accepted packets and intruding packets. The legend ⫯ indicates the desired target used for normal and intrusion. The '○' represents the output of the network. Table 3 gives number of patterns used for training and testing the performance of ESNN in classifying the intrusion packet. Table 4 gives number of patterns classified and misclassified.

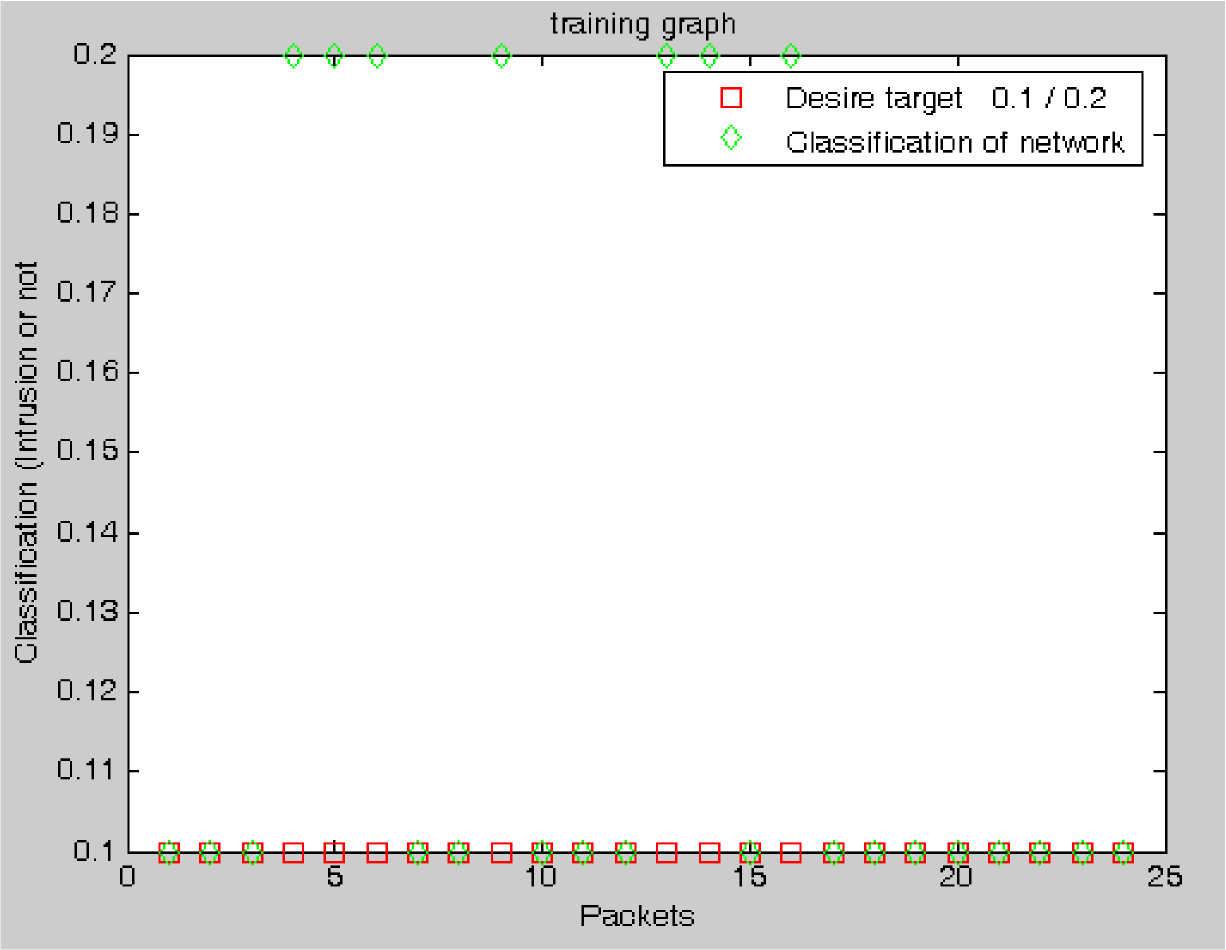


FIGURE 3. Packet classification

Table 3 : Distribution of patterns chosen for training

Packet Type	Total numbers used for training
Normal	17
Intrusion	7

Table 4 : Classification performance

Packet type	Total number tested	No. classified	No. misclassified
Normal	17	15	2
Intrusion	7	2	5

3. CONCLUSION AND FUTURE WORK

In this work, KDD dataset has been considered to experiment the performance of ESNN in classifying the LAN intrusion packets. A topology of 41 X 20 X 1 had been chosen. The future work will involve in implementing an echo state neural network for classification of intrusion packet. Future work will focus on comparison of echo state work with liquid state machine and back-propagation algorithm

4. REFERENCES

[1] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), Parallel distributed processing: Explorations in the microstructure of cognition (Vol.1, pp. 318-362).

[2] Denning, Dorothy. (February, 1987). An Intrusion-Detection Model. *IEEE Transactions on Software Engineering, Vol. SE-13, No. 2.*

[3] Lippmann.R.P,; AN Introduction to computing with neural nets;IEEE Transactions on ASSP Mag. 35,4(2) 4-22, 1987.

[4] Fox, Kevin L., Henning, Rhonda R., and Reed, Jonathan H. (1990). A Neural Network Approach Towards Intrusion Detection. *In Proceedings of the 13th National Computer Security Conference.*

[5] HIROSE,Y., YAMSHITA,K., AND HIJIYA,S., 1991, Back-propagation algorithm which varies the number of hidden units, Neural Networks, Vol.4, No.1, pp-61-66.

[6] Debar, H. & Dorizzi, B. (1992). An Application of a Recurrent Network to an Intrusion Detection System. In *Proceedings of the International Joint Conference on Neural Networks.* pp. (II)478-483.

[7] Debar, H., Becke, M., & Siboni, D. (1992). A Neural Network Component for an Intrusion Detection System. In *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy.*

[8] Hammerstrom, Dan. (June, 1993). Neural Networks At Work. *IEEE Spectrum*. pp. 26-53.

[9] BERSHAD, N.J., SHYNK, J.J., AND FEINTUCH, P.L., 1993, Statistical analysis of the single-layer-back-propagation algorithm: Part-I-Mean weight behaviour, IEEE Trans. on Acoustics, Speech and Signal Processing, Vol.41, No.2, pp.573-582.

[10] BERSHAD, N.J., SHYNK, J.J., AND FEINTUCH, P.L., 1993, Statistical analysis of the single-layer back-propagation algorithm: Part-II-NMSE and classification performance, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol.41, No.2, pp.583-591.

[11] Helman, P. and Liepins, G., (1993). Statistical foundations of audit trail analysis for the detection of computer misuse, *IEEE Trans. on Software Engineering*, 19(9):886-901.

[12] Ilgun, K. (1993). USTAT: A Real-time Intrusion Detection System for UNIX. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*. pp. 16-28.

[13] Denault, M., Gritzalis, D., Karagiannis, D., and Spirakis, P. (1994). Intrusion Detection:

Approach and Performance Issues of the SECURENET System. In *Computers and Security Vol. 13, No. 6, pp. 495-507*

[14] Frank, Jeremy. (1994). Artificial Intelligence and Intrusion Detection: Current and Future Directions. In *Proceedings of the 17th National Computer Security Conference.*

[15] Mukherjee, B., Heberlein, L.T., Levitt, K.N. (May/June, 1994). Network Intrusion Detection. *IEEE Network*. pp. 28-42.

[16] Chung, M., Puketza, N., Olsson, R.A., & Mukherjee, B. (1995) Simulating Concurrent Intrusions for Testing Intrusion Detection Systems:Parallelizing. In *Proceedings of the 18th NISSC*. pp. 173-183.

[17] Cramer, M., *et al.* (1995). New Methods of Intrusion Detection using Control-Loop Measurement. In *Proceedings of the Technology in Information Security Conference (TISC) '95.* pp. 1-10.

[18] Kohonen, T. (1995) *Self-Organizing Maps*. Berlin: Springer.

[19] Staniford-Chen, S. (1995, May 7). Using Thumbprints to Trace Intruders. UC Davis.

[20] Tan, K. (1995). The Application of Neural Networks to UNIX Computer Security. In *Proceedings of the IEEE International Conference on Neural Networks, Vol.1* pp. 476 – 481.

[21] Ghost, A.K., *et al.* (September 27, 1997). "Detecting Anomalous and Unknown Intrusions Against Programs in Real-Time". DARPA SBIR Phase I Final Report. Reliable Software Technologies.

[22] Porras, P. & Neumann, P. (1997). EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances. In *Proceedings of the 20th NISSC.*

[23] Puketza, N., Chung, M., Olsson, R.A. & Mukherjee, B. (September/October, 1997). A Software Platform for Testing Intrusion Detection Systems. *IEEE Software, Vol. 14, No. 5*

[24] Ryan, J., Lin, M., and Miikkulainen, R. (1997). Intrusion Detection with Neural Networks. *AI Approaches to Fraud Detection and Risk Management: Papers from the 1997 AAAI Workshop (Providence, Rhode Island)*, pp. 72-79. Menlo Park, CA: AAAI.

[25] Tan, K.M.C & Collie, B.S. (1997). Detection and Classification of TCP/IP Network Services. In *Proceedings of the Computer Security Applications Conference*. pp. 99-107.

[26] Sebring, M., Shellhouse, E., Hanna, M. & Whitehurst, R. (1988) Expert Systems in Intrusion Detection: A Case Study. In *Proceedings of the 11th National Computer Security Conference.*

[27] GRAHAM, R. 2000. FAQ: Network Intrusion Detection Systems, http://www.robertgraham.com

[28] Jaeger, H.; The echo state approach to analyzing and training recurrent neural networks; (Tech.Rep. No. 148). Bremen: German National Research Center for Information Technology, 2001.

[29] BACE, R. 2002. Intrusion Detection, Macmillan Technical Publishing

[30] Jaeger, H.; Tutorial on training recurrent neural networks, covering BPPT, RTRL,EKF and the "echo state network" approach (Tech. Rep. No. 159).; Bremen: German National Research Center for Information Technology, 2002.

[31] Jaeger, H;. Short term memory in echo state networks; (Tech. Rep. No. 152) Bremen: German National Research Center for Information Technology. 2002.

[32] KAZIENKO, P., AND DOROSZ, P. 2003, Intrusion Detection Systems (IDS) Part I – network intrusions; attack symptoms; IDS tasks; and IDS

[33] BACE, R AND MELL, P., 2004, NIST Special Publication on Intrusion Detection Systems, http://www.nist.gov

[34] GORDEEV, M. 2004. Intrusion Detection Techniques and Approaches, http://www.ict.tuwein.ac.a

[35] JEAN-PHILIPPE 2004, Application of Neural Networks to Intrusion Detection, http://www.sans.org

[36] Albert Mo Kim Cheng,On-Time and Scalable Intrusion Detection in Embedded Systems, Real-Time Systems Laboratory,Department of Computer Science,University of Houston, TX 77204, USA

[37] H. Güneş Kayacık, A. Nur Zincir-Heywood, Malcolm I. Heywood, Selecting Features for Intrusion Detection:A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets Dalhousie University, Faculty of Computer Science,,6050 University Avenue, Halifax, Nova Scotia. B3H 1W5

[38] R.S.Michalski, K.A.Kaufman, J.Pietrzykowski, B.Sniezynski, J.Wojtusiak, Intelligent Information Systems 2006, New Trends in Intelligent Information Processing and Web Mining Ustron, Poland, June 19-22, 2006

[39] Steven Cheung, Bruno Dutertre, Martin Fong, Ulf Lindqvist, Keith Skinner and Alfonso Valdes, Using Model-based Intrusion Detection for SCADA Networks,Computer Science Laboratory,SRI International,December 7, 2006

[40] Ajith Abraham, Ravi Jain, Johnson Thomas and Sang Yong Han, D-SCIDS: Distributed SoftComputing intrusion detection system, Journal of Network and Computer Applications 30 (2007) ,PP 81–98

COMPUTER SCIENCE JOURNALS SDN BHD
M-3-19, PLAZA DAMAS
SRI HARTAMAS
50480, KUALA LUMPUR
MALAYSIA