Faculty of Science
Mathematics and Computer Science Department

Port Said University

Security in Mobile Ad-hoc Networks using Neutrosophic Technique

A Thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Computer Science

BY

Haitham Samy Mohammed Elwahsh

M. Sc. **(Computer Science)**
Assistant Lecturer in Computer science,
Faculty of Computers and Information,
Kafrelsheikh University

Supervised By

Prof. Dr. Ibrahim Mahmoud ELhenawy
Professor of Computer Science Dept.
Faculty of Computers and Information
Zagazig University

Prof. Dr.Ahmed Abdel-Khalek Salama
Professor of Mathematics Dept.
Faculty of Science
Port Said University

Prof. Dr. Magdy El-Banna
Dean of Faculty of Science

Dr. Wael Abdel-Kader
Head of Mathematics Dept

2018

ACKNOWLEDGMENTS

First of all, I’d like to thank ALLAH for his non-countable blessings. GOD is the only one deserves the appreciation for doing any good deed in life.

Second, to give rights to their owners, I’d gratefully thank my mentors, guiders and supervisors Prof. Dr. **Ibrahim Mahmoud ELhenawy**, Dr. **Ahmed Abdel-Khalek Salama**. For their remarkable efforts to illuminate me through the way for creating this work. I ask ALLAH to give them so many rewards since I can’t reward them enough.

Third, I’d like to thank **my mother** and **my father**, the endless river of love and giving. Without your inspiration, drive, and support that you have given me, I might not be the person I am today. Thanks for all of you and wish you health and wellness.

My wife who provided a lot of helping and providing a convenient environment for my research. Also this work is also dedicated for my brothers and sister.

My father-in-law, Eng. Fathy Emara, for his support and helping by his advice me for facing the life problems.

Finally I’d dedicate this work for my kids the handsome princes, “Samy and Yassin”, I ask ALLAH to see them a better than I am.

Haitham Samy ELwahsh

Abstract

Network security is a major research area for both scientists and business. Intrusion Detection System (IDS) is one of the most challenging problems in Mobile Ad Hoc Networks (MANETs). The main reason resides behind the changing and uncertain nature of MANETs networks. Hence, a compensate evolving in the IDS would be converting the whole system to rely on uncertainty and indeterminacy concepts.

These concepts are the main issues in the fuzzy system and consequently in neutrosophic system. In neutrosophic system, each attack is determined by MEMEBERSHIP, INDTERMINACY and NONMEMEBERSHIP degrees. The main obstacle is that most data available are regular values which are not suitable for neutrosophic calculation.

Therefore, the preprocessing phase of the neutrosophic knowledge discovery system is essential. Converting the regular data to neutrosophic sets is a problem of generating the MEMEBERSHIP, NONMEMEBERSHIP and INDTERMINACY functions for each variable in the system. Self-Organized Feature Maps (SOFM) are unsupervised artificial neural networks that were used to build fuzzy MEMEBERSHIP function, hence they could be utilized to define the neutrosophic variable as well.

SOFMs capabilities to cluster inputs using self-adoption techniques have been utilized in generating neutrosophic functions for the subsets of the variables. The SOFM are used to define the MEMEBERSHIP, NONMEMEBERSHIP functions of the KDD network attacks data available in the UCI machine learning repository for further processing in knowledge discovery.

Afterwards the preprocessing module generates the INDTERMINACY function from both of the MEMEBERSHIP, NONMEMEBERSHIP functions basing on the neutrosophic set definitions.

The thesis proposes a MANETs attack inference by a hybrid framework of Self Organized Features Maps (SOFM) and the Genetic Algorithms (GA). The hybrid utilizes the unsupervised learning capabilities of the SOFM to define the MANETs neutrosophic conditional variables.

The neutrosophic variables along with the training data set are fed into the Genetic Algorithms to find the most fit neutrosophic rule set. The neutrosophic correlation coefficient is selected as the fitness function to help in finding the rule set from a number of initial sub attacks where the propositions and consequences are highly correlated.

These neutrosophic classification rules are designed to detect unknown attacks in MANETs. The simulation and experimental results are conducted on the KDD-99 network attacks data available in the UCI machine-learning repository for further processing in knowledge discovery.

The proposed system proved its ability to detect attacks in MANETs environment in reasonable accuracy and lower false alarm rates in comparison with other IDS found in literature like C4.5, Support Vector Machine, Ant Colony Optimization and Particle Swarm Optimization. Hence, applying the neutrosophic concepts to the IDS enhances classification accuracy in MANETs significantly.

Publications

Publications

[1] Haitham Elwahsh, Mona Gamal, A. A. Salama, and I. M. El-Henawy Modeling Neutrosophic Data by Self-Organizing Feature Map: MANETs Data Case Study, Procedia Computer Science, 2017, Volume 121, pp 152-159, DOI.org/10.1016/j.procs.2017.11.021.

[2] Haitham Elwahsh, Mona Gamal, A. A. Salama, and I. M. El-Henawy A Novel approach for classify MANETs attacks with a neutrosophic intelligent system based on genetic algorithm, , Security and Communication Networks, 2018, Volume 2018, Article ID 5828517, 10 pages.

[3] Haitham Elwahsh, Mona Gamal, A. A. Salama, and I. M. El-Henawy Intrusion Detection System and Neutrosophic theory for MANETs: A Comparative study, neutrosophic sets and systems (NSS), Submitted.

Table of Contents

Chapter 1

Thesis Introduction

Chapter 2

MANETs overview and theoretical overview of methodologies used

Chapter 3

Preprocessing Phase of the Neutrosophic Knowledge discovery system

Chapter 4

Proposes a MANETs attack inference by a hybrid framework of Self Organized Features Maps (SOFM) and the Genetic Algorithms (GA)

Chapter 5

5. Conclusion and Future Work

List of Figures

List of Tables

List of Abbreviations

ID	Abbreviation	Full Name
1	**MANETs**	Mobile Ad Hoc Networks
2	**IDS**	Intrusion Detection System
3	**CA**	Certification Authority
4	**NS**	Neutrosophic Set
5	**FS**	Fuzzy Set
6	**IFS**	Intuitionistic Fuzzy Set
7	**T**	Truth-MEMEBERSHIP
8	**I**	INDETERMINACY
9	**F**	Falsity-MEMEBERSHIP
10	**SOFM**	Organized Feature Maps
11	**GA**	Genetic Algorithm
12	**NR**	Neutrosophic rule
13	**ALOHA**	Areal area of Hazardous Atmospheres
14	**CSMA**	Carrier Sense Multiple Access
15	**SURAN**	Survivable Adaptive Radio Network
16	**DOD**	Department of Defense
17	**GloMo**	Globe Mobile Information System
18	**NTDR**	Near Term Digital Radio
19	**IETF**	Internet Engineering Task Force
20	**DoS**	Denial of Service
21	**TTP**	Trusted Third Parties
22	**SCIE**	Science Citation Index Expanded
23	**ESCI**	Emerging Sources Citation Index
24	**NP**	Nondeterministic Polynomial time
25	**DMZ**	Demilitarized zone
26	**NIDS**	Network IDS
27	**HIDS**	Host-based intrusion detection systems
28	**CRTDH**	Chinese Remainder Theorem and Diffie-Hellman
29	**RPS**	Random Pre-distribution Scheme
30	**LM**	Leighton and Micali Scheme
31	**TA**	Trusted Authority
32	**VN**	Verifying Node
33	**GPS**	Global positioning systems
34	**ECC**	Elliptic Curve Cryptography
35	**IKM**	ID-based Key Management
36	**AC**	Authentication code
37	**ACS**	Address-based Cryptography Scheme
38	**IDPKC**	Identity Based Public Key Cryptography
39	**BDHP**	Bilinear Diffie-Hellman Problem
40	**TCP**	Transmission Control Protocol
41	**UDP**	User datagram protocol
42	**ICMP**	Internet Control Message Protocol
43	**HTTP**	Hypertext Transfer Protocol

44	**FTP**	File Transfer Protocol
45	**SMTP**	Simple Mail Transfer Protocol
46	**NON_MEM**	NONMEMEBERSHIP
47	**MEM**	MEMEBERSHIP
48	**WEKA**	Waikato Environment for Knowledge Analysis
49	**SVM**	Support Vector Machine
50	**GP**	Genetic Programming
51	**GPSVM**	Genetic Programming Support Vector Machine
52	**HG - GA**	Hypergraph based Genetic algorithm
53	**FAR**	False Alarm Rate

CHAPTER 1

INTRODUCTION

1.1 Problem description

A Mobile Ad Hoc Network (MANET) [1] is a self-organizing, infrastructure less, multi-hop network. The wireless and distributed nature of MANETs poses a great challenge to system security designers. MANETs are a group of accumulations, self-sorted out, wireless end-user terminals, independent of any settled infrastructure. The arbitrary topology of MANETs presents constraints in communication since it depends on efficient and admission node participating with the end goal to execute routing protocols (RP). These constraints in communication give a fertile ground for assailants [1].

MANETs are defenseless against packet dropping, packet modification, packet misrouting, selfish node behavior, DOS attack, etc. hence providing security ensures is rather a complicated defy. In MANETs, every node acts as a router, which can forward/receive packets to/from its neighbors. MANETs can work in both isolation or in coordination with a wired infrastructure.

MANETs are progressively used in numerous different applications in areas, for example, intelligent transportation systems and fault-tolerant mobile sensor grids. Adaptability, self-configurability are making these systems fundamental component in future mobile and wireless network models. Despite that security issues in MANETs have attached much consideration over the most recent couple of years, most research endeavors have been centered around particular security territories.

The lake of infrastructure poses huge number of challenges in MANET through the perspective of network configuration for example [2]:

1. **Channel vulnerability** – broadcast wireless channels allow message eavesdropping and injection easily.
2. **Node vulnerability** – nodes don't dwell in physically secured places, therefore effectively fall under assault.
3. **Absence of infrastructure** –certification/ authentication authorities are missing.
4. **Dynamically changing network topology** puts security of routing protocols under attacks.
5. **Power and computational limitations** prevent the utilization of complex encryption algorithms.

The idea of MANETs gives awesome amount of difficulties with system security specialist because of the below causes:

- **Firstly**: the MANETs is more vulnerable to assaults running from passive eavesdropping to active interfering;
- **Secondly**, the absence of an online certification authority (CA) or Trusted Third Party increase the trouble to announce security techniques;
- **Thirdly**: mobile devices have a tendency to have constrained power consumption and computation abilities which make it more helpless against Denial of Service assaults and unable to perform computation-heavy algorithms like public key algorithms;

- **Fourthly,** in MANETs, there are more eventuality for confided client being compromised and then being utilized by foe to dispatch assaults on system, also, taking into account both internal assaults and external assaults in MANETs, in which internal assaults are more hard to manage;
- **Finally,** the mobility of the network enforces reconfiguration which makes more possibilities for assaults, for instance, it is hard to recognize stale/forged routing information.

1.2 Neutrosophic Intrusion detection system:

Intrusion detection is a network security technology initially worked for worked for distinguishing defenselessness abuses against a target application or PC. The mechanism that is achieved is named an IDS. An IDS gathers activity data and then analyses it to determine whether or not there are any actions that assaulted the Protection rules. Once an IDS decides that an abnormal behavior or an action which recognized to be an attack occurs, it then makes an alarm to warn the security administrator.

Additionally, IDS also can start a correct response to the malicious activity. Although there are many IDS techniques built for wired networks these days, they're not appropriate for wireless networks because of the variations in their characteristics. Therefore, those techniques should be changed or new techniques should be developed to form intrusion detection work effectively in MANETs. Because of the variations in the MANETs characteristics, traditional IDS are unsuitable. This thesis tends to use neutrosophy theory as a novel solution for the indeterminacy in MANETs.

Smarandache [3] presented the principle of Neutrosophic Set (NS) and mathematical Theory, to define any situation by a ternary crisp build (MEMBERSHIP-INDETERMENACY - NONMEMEBRSHIP). Salama et al. Work [4, 5, 6, 7, 8, 9, 10 and 11] formulated a beginning to new fields of neutrosophic theory in computer discipline.

The neutrosophic indeterminacy assumption is very significant in many of circumstances such as information fusion (collecting data from various sensors). Also, NS is a conceivable set that generalize the principle of the traditional set, Fuzzy Set (FS) [12] and Intuitionistic Fuzzy Set (IFS)) [12], etc. NS ‘A’ determined on universe U. $x = x(\mathrm{T}, \mathrm{I}, \mathrm{F}) \in \mathrm{A}$ with T, I and F are defined over the interval $]0^{-}, 1^{+}[$. T is the truth-MEMEBERSHIP , I is the INDETERMINACY and F is the falsity-MEMEBERSHIP degrees on the setA.

1.3 Thesis Motivation:

Designing a neutrosophic IDS is a proper solution in handling vague circumstances. The neutrosophic IDS is formed of two sub phases: the preprocessing stage and the network attacks classification stage. The preprocessing stage is concerned by formulating the network features in a neutrosophic format appropriate for the classification under the umbrella of the neutrosophy theory.

The KDD network dataset [13] is reformatted into neutrosophic form $(\mathrm{x}, \mu_{\mathrm{A}}(\mathrm{x}), \sigma_{\mathrm{A}}(\mathrm{x}), \nu_{\mathrm{A}}(\mathrm{x}))$ where x is the value of feature, $\mu_{\mathrm{A}}(\mathrm{x})$ is the

MEMEBERSHIP (MEM), $\sigma_A(x)$ is the INDETERMINACY (I) and $\nu_A(x)$ is the NONMEMEBERSHIP (NON_MEM) degrees of the x in the feature space. This phase is implemented based on the unsupervised clustering capabilities of the self-Organized Feature Maps (SOFM), will be discussed in details in chapter 4.

Afterwards, a novel neutrosophic intelligent system based on Genetic Algorithm (GA) is proposed as an IDS. The novelty is accomplished via adding the third dimension of indeterminacy supported by the neutrosophy theory. All pervious techniques used only two dimensions (MEMBERSHIP- NONMEMEBRSHIP). The IDS makes use of the search capabilities of the GA to find the most correlated rules in KDD networks data taking the neutrosophic correlation coefficient as a fitness function [14].

1.4 Research Contribution

The contributions delivered by this study is divided into 2 directions:

I. **The preprocessing phase of the neutrosophic knowledge discovery system.**

A pre-processing phase in an intelligent system for detecting threats in MANET networks. The main issue in the pre-processing phase is concerting the regular data found in the KDD data from UCI machine learning repository into neutrosophic data. Converting the regular data to neutrosophic values is a problem of generating the MEMEBERSHIP, NONMEMEBERSHIP and INDETERMINACY functions for each variable in the system. For automating this step, SOFM are used. SOFMs are unsupervised artificial neural networks that were used to build fuzzy MEMEBERSHIP function, hence they were utilized to define the neutrosophic variable membership and non-membership functions. Then, the indeterminacy function could be calculated based on the NS definition see chapter 2 section 2.6.1 [15].

II. A Novel approach for classify MANETs attacks with a neutrosophic intelligent system based on genetic algorithm.

Next direction is to build a classification pattern for the neutrosophic variables to detect threats in the MANET network. This phase implemented by an artificial intelligent algorithm (Genetic Algorithm). Design an efficient set of neutrosophic rules used for MANETs assaults deduction by a hybrid model of SOFMs and the GA.

The hybrid uses the unsupervised learning abilities of the SOFM to characterize the MANETs neutrosophic conditional variables. The neutrosophic factors alongside with the training data set are feed into the GA to find the most fit neutrosophic rule (NR) set from a various starting sub assaults as indicated by the fitness function. This technique is intended to identify obscure assaults in MANETs [16].

1.5 Thesis Objective

- Provide proper understanding of security in MANETs as it is a very important and tricky issue in the field of MANETs.
- Implementing a secured MANET in which data can be sent and received taking into consideration the security challenges in MANETs.

- Introducing the Self-Organized Feature Maps (SOFM) as unsupervised artificial neural networks that were used to build fuzzy MEMEBERSHIP function, hence they could be utilized to define the neutrosophic variable as well.
- Build a neutrosophic IDS based on classification pattern for the neutrosophic variables to detect threats in the MANET network.
- Introducing a Novel ways to get the most fit NR set from a several of initial sub assaults according to the neutrosophic correlation coefficient as a fitness function within the genetic algorithm.

The thesis is organized as follow:

Chapter 2, gives MANETs Overview, Security, Challenges and attacks. and theoretical overview of methodologies used like Self Organize Feature Map (SOM), Genetic Algorithm (GA), Neutrosophic theory and the KDD CUP 99 Data set, and study related overview.

Chapter 3, Introduces Modeling Neutrosophic Data by Self-Organizing Feature Map: MANETs Data Case Study.

Chapter 4, illustrated the Novel approach for neutrosophic intelligent system based on genetic algorithm and the comparative study.

Chapter 5, provides conclusions and Future work.

Chapter 2

MANETs Overview and theoretical overview of methodologies used

2.1 What is a MANET?

A Mobile Ad-hoc Networks (MANETs) is an assortment of self-configuring nodes that can communicate with each other by initializing a multi-hop radio decentralized network. In MANETs every node has the functionality of routers and terminals and can communicate with another network device that in same radio range or one that's outside their radio range not relying on access point [17].

In the infrastructure wireless networks each user communicates directly with a base station or access point, in contrast, MANET is wireless ad hoc network of mobile routers that don't need access point for communication rather than these mobiles connected directly by wireless links (i.e. MANET is infrastructure less wireless network) and these devices are absolve to move suddenly and organize themselves indiscriminately. Thus, the communication between devices in MANET is occur by using multi-hop paths.

Nodes in the MANET share the wireless medium and the topology of the network changes sporadically and dynamically. In MANET, breaking of communication connect is accustomed, as nodes are free to move to anywhere. The concentration of nodes and the number of nodes are relies upon the applications in which we are utilizing MANET [18].

MANETs are widely utilized in military and other scientific areas. Different kind of applications are based on the concept of MANET for communication like remote Sensor Network, Device Networks, Data Networks, etc. With numerous applications there are still some outline issues and challenges to conquer.

There are a security challenges and troublesome in MANET results from the arbitrarily move of network devices and ability of any device to connect to any other device freely [19], it is unthinkable for Ad hoc network to own a fixed infrastructure.

MANETs are self-organized networks which comprise of an arrangement of wireless nodes. The nodes can move in a subjective way and work as its own think. They may join or leave the network without limitations. In this manner, MANETs topologies are dynamic and expensive to keep up. Besides, wireless channels make the routing and message transmission considerably more difficult [20]. In these networks the nodes can work as routers that mange the routing path between other nodes in the network as well as end-users. They'll depend alternate nodes to transfer the messages, which are exposed in an open dangerous scenario for any intermediate node to be capable of destroying the integrity or choose as their like to manage the messages. To wrap things up, nodes in MANETs have only limited resource, i.e. Battery power, bandwidth and CPU power. They are normally embedded systems which are produced for certain fixed tasks [20].

The fundamental qualities of MANETs can be summarized as the follow:

- Dynamic topology: the nodes of the network can move discretionarily, the topology of the network likewise changes.
- Bandwidth Constraints: the data transfer capacity of the connection is compelled and the limit of the network is additionally factor tremendously. Due to the dynamic topology, the yield of each transfer node will differ with the time and after that the connection limit will change with the connection change.

- Power Constraints: it is a genuine factor. Because of the mobility qualities of the network, devices utilize battery as their power supply. Accordingly, the advanced power conservation procedures are exceptionally essential in planning a system.
- Security Constraints: The security is restricted in physical perspective. The mobile network is easier to be assaulted than the settled network. Defeating the shortcoming in security and the new security inconvenience in wireless network is on demand Figure 2.1 shows the general form of cellular networks vs. MANETs.

A reaction of the adaptability is the straightforwardness with which a node can join or leave a MANET. Absence of any settled physical and, sometimes, authoritative infrastructure in these networks makes the task of securing these networks extremely challenging [21].

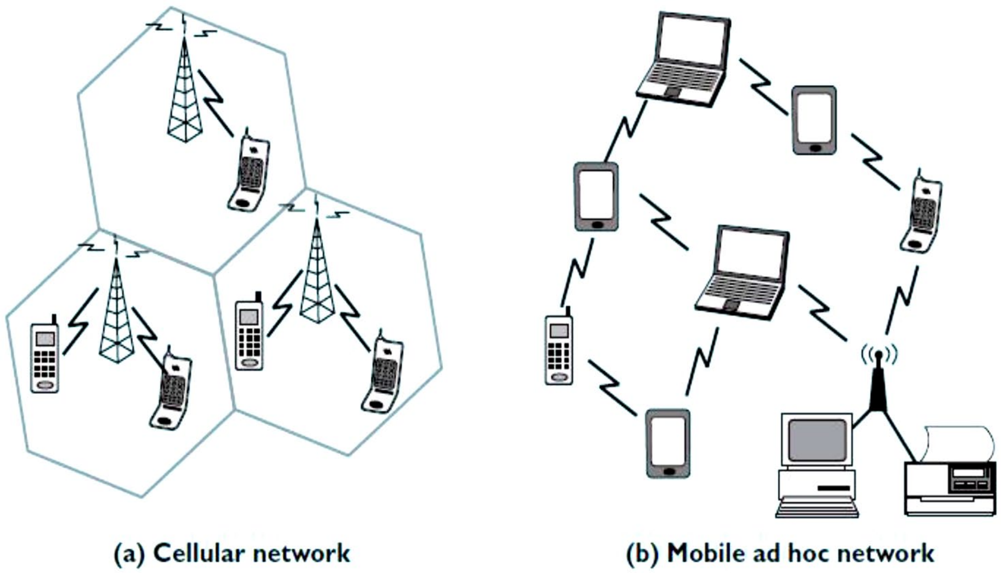


Figure 2.1: Cellular network (a) VS. (b) MANETs

2.2. Security in MANETs.

One of the challenges in MANET is security, in view of its intrinsic vulnerabilities. These vulnerabilities are nature of MANET structure that can't be expelled [22]. So that, attacks with noxious expectation have been and will be contrived to misuse these vulnerabilities and to injure MANET activities. In comparison between traditional networks and MANET, the MANET network is more vulnerable to be exploited by attackers for malicious activities.

First of all, the utilization of wireless connections renders the network susceptible to attacks ranging from passive eavesdropping to active interfering. Unlike wired networks where an enemy must increase physical access to the network wires or go through a few lines of defense at firewalls and gateways, attacks on a wireless network can originate from all directions and focus at any node. Damages can include leaking secret information, message contamination, and node impersonation. All these imply that a wireless ad-hoc network won't have a reasonable line of defense, and each node must be set up for encounters with an enemy straightforward or indirectly [23].

Second, mobile nodes are self-ruling units that are equipped for wandering freely. This implies nodes with insufficient physical security are responsive to being caught, traded off, and seized. Since finding a specific mobile node in an expensive scale ad hoc network may not be effectively done, attacks by a traded off node from inside the network are far more damaging and much harder to recognize. Along these lines, mobile nodes and the infrastructure must be set up to work in a mode that trusts no associate.

Third, decision-making in mobile computing environment is sometimes decentralized and some wireless network algorithms depend on the agreeable cooperation of all nodes and the infrastructure. The absence of centralized authority implies that the enemies can misuse this helplessness for new kinds of attacks intended to break the cooperative algorithms. A mobile ad-hoc network is extremely responsive to security attacks because of its open medium, dynamically changing network topology, cooperative algorithms, absence of centralized monitoring and administration point, and absence of an unmistakable line of safeguard as figures (1.2, 1.33) shows. These vulnerabilities are nature of MANET structure that can't be expelled. Accordingly, attacks with malignant goal have been and will be formulated to abuse these vulnerabilities and to injure MANET tasks.

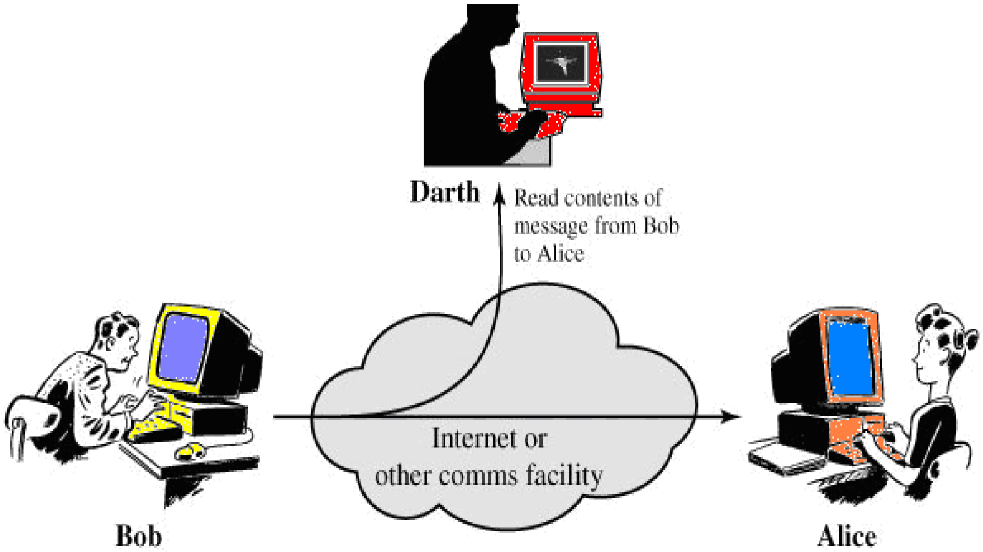

Darth
Read contents of message from Bob to Alice
Internet or other comms facility
Bob
Alice

Figure 2.2 the Release of Message Contents

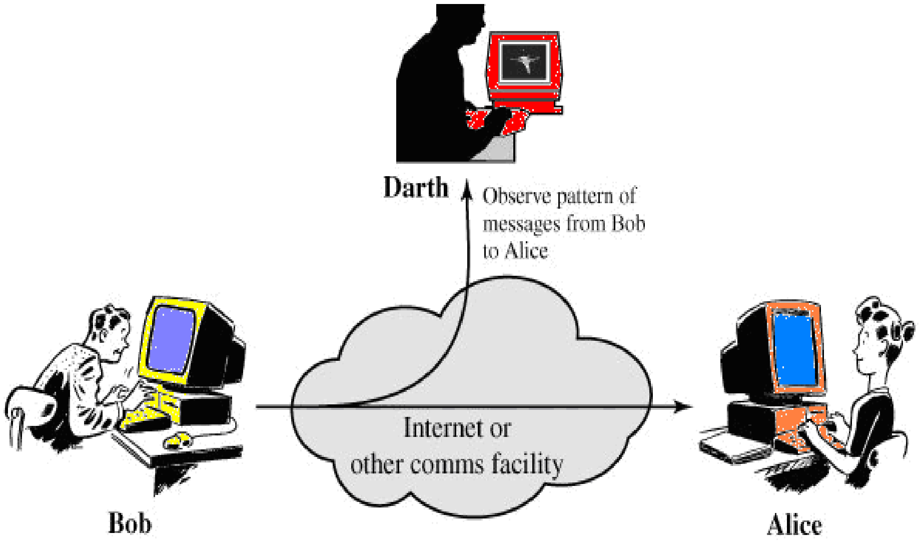

Darth
Observe pattern of messages from Bob to Alice
Internet or other comms facility
Bob
Alice

Figure 2.3 the Traffic Analysis

2.3 Security Requirements in MANETs

The nodes associated with one another in the same range thru wireless connections. Some nodes act as routers used to transfer data to extra inaccessible nodes. The topology of MANETs isn't settled [24]. The change appears when these node move in and out of each other's communication area. This make MANETs very exceptionally defenseless against assaults and the security issues turn out to be extremely intricate.

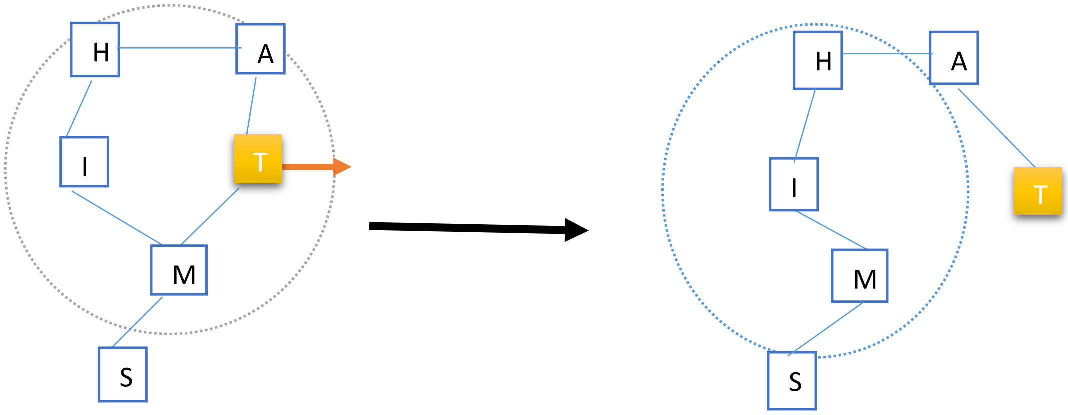


Figure 2.4 Topology Change in Ad-Hoc Networks of Nodes

MANETs are another worldview of wireless communication for mobile hosts. Node mobility causes frequent changes in topology Figure 2.4 shows such a model: nodes H, A, I, T, M, and S comprise an ad-hoc network. The circle represents the radio area of node M. At first nodes M and T have an immediate connection between them. At the point when T moves out of M's radio area, the connection is broken. However, the network is as yet connected, because M can connect T through I, H, and A. The security services of MANETs are the same in other network must meet. As follow we depict the prerequisites for MANETs.

2.3.1 Availability

Availability ensures that the coveted network services are accessible whenever they are required. As every node in the network rely upon one another to hand-off data, denial of service (DoS) assaults are simple to commit. Such as, a pernicious client might endeavor to stick or generally attempt to meddle with the stream of data. Otherwise, the routing protocol ought to have the capacity to deal with both the changing topology of the network and assaults from the noxious nodes by encouraging the system with precise data [25, 26]. Another defenseless point, is the restricted battery power of MANETs node. Typically, these clients attempt to spare vitality with power sparing plans, so that when the node isn't in active usage, energy isn't used. Assaults that intend battery depletion, so a pernicious client can reason advanced power consumption from different nodes battery, making these nodes to expire early [27].

2.3.2 Non-Repudiation

Non-repudiation guarantees that the starting point of a message can't deny having sent the message. Once a client H gets an incorrect data from a client T, non-repudiation permits H to denounce T utilizing this data and to persuade different clients that T is attacker.

2.3.3 Confidentiality and Integrity

Data confidentiality is an essential safety basic for MANETs. It guarantees that the data sent can't be realized by anybody other than the approved specialist. With wireless connections, anybody can see the data sent and the absence of encryption the data is effectively accessible. *Data integrity* indicates the perfection of message sent from one client to another. That is, it guarantees that a data sent from client M to client S was not altered amid transmission by a malignant client I. in the event that a vigorous privacy component is utilized, guaranteeing data integrity might be as straightforward as adding one-way hashes to encrypted messages.

2.4 Security Solutions Constraints

For history, network security staff have received a central, generally defensive worldview to fulfill previously mentioned necessities. Once, joining the network the node work in an open design distribute delicate documents, permitting approaching network connections since it is certainly ensured that any pernicious client from external won't be permitted get to. In spite of the fact these solutions have been viewed as from the get-go in the development of MANETs, endeavors to adjust comparable client-server keys to a decentralized domain have to a great extent been inadequate. To be well appropriate, security solutions for MANETs should preferably get the next attributes:

- **Lightweight**: reduce the measure of calculation and connection mandatory to guarantee the security facilities to suit the restricted power and computational assets of MANETs.
- **Decentralized**: efforts to protect nodes necessity be ad-hoc, they should build protection without refer to centralized. Rather, security perfect models must necessitate the collaboration of every reliable client in the system.
- **Reactive**: MANETs are self-motivated. Clients reliable and noxious may join and exit the system precipitously and unannounced. Security ideal necessity respond to variations in system behaviour; they necessity try to discover bargains and weaknesses.
- **Fault-Tolerant**: clients are probably going to leave or be endangered suddenly without any notification. So the necessities of security solutions ought to be planned with such mistakes in attention.

2.4.1 Challenges

The behaviour of MANETs make it liable to assaults running from stealthily spying to active impersonation. Spying may provide an aggressor right to use secret data, accordingly violating secrecy. Active assaults might run from data replay or erasure, infusing wrong data, impersonating a client, and so on.

Consequently, security arrangements need to consider vindictive assaults from outside as well as from inside the network. Then, the certainty connections among individual clients can alteration, particularly when a few clients are observed in danger. In this way, security technique should be self-motivated, and ought to be suitably adaptable.

2.4.1.1 Secure Routing

The present routing protocols intended for MANETs adapt well to dynamically evolving topology, yet are not intended to give protection against malignant assailants. In MANETs, with respect to aggressors, we can order them into outer and interior. External aggressors may infuse mistaken routing info, resend old routing info so as to segment or excess the system with re-transmissions and wasteful routing. Solutions must beat these potential issues and utilize a few attribute of MANETs to encourage secure routing. Once the abnormal clients have been known, if there is adequate number of conceivably disconnect and usable paths, the routing protocol ought to be capable of have the capacity to sidestep the compromised nodes by utilizing alternate routes.

2.4.1.2 Key Management

In a public key (PK) infrastructure, every client has a public/private key pair. A client disseminates its PK unreservedly to the alternate client in the network; anyway it keeps its private key (PvK) to just itself. A Certification Authority (CA) is utilized for key management and has its own (P/Pv) key pair. The CA's PK is known to each network client. The confided in CA is dependable to sign certificates, restricting PKs to clients, and needs to remain online to confirm the present ties.

The PK of a node ought to be disavowed if this client is never again trusted or leaves the network. Likewise, if a Certification Authority is imperiled, the assailant can sign incorrect certificates utilizing the database of the PvKs. Guileless repetition of Certifications Authority can make the system further helpless, then upnormal of any single imitation can make the network to fizzle. Henceforth, it might be more reasonable to share the confidence to an arrangement of clients by giving these clients a chance to distribute the key management responsibility.

2.4.2 Authentication

Authentication indicates the exact, outright recognizable proof of clients who desire to join the network. Truly, authentication has been proficient by a recognized central verification server.

2.4.3 Trusted Third Parties (TTP)

A standout amongst the primeval ways to deal with authentication in MANETs utilizes a TTP. Each client that desires to participate in MANETs acquires an authentication from a universally TTP. When two clients desire to connect, they

initially verify whether the other client has a legal certificate. The TTP approach is weighed down with imperfections. It presumably isn't sensible to require all MANETs-enabled devices to have an authentication. **Secondly**, every client desires to have a unique name.

2.4.4 Chain of Trust

Interestingly, the chain of trust paradigm depends on any client in the system to make validation. Namely, if a client desires to go into a network area, it might ask for any of the current clients for validation. This model be unsuccessful if there are abnormal clients inside the network or the approaching clients can't be authenticated at all.

2.4.5 Location-Limited Authentication (LLA)

LLA imposes on the way that two clients are near each other and most MANETs be existent in a little zone. Bluetooth and infrared are two of the most broadly utilized protocols for this type of validation. In spite of the fact that it may not appear glaringly evident, LLA is possibly exceptionally secure.

The security is acquired from physical affirmation and alter recognition. That is, the authenticating client can be sensibly sure that the client it supposes is being verified is the client it is really verifying by physical signs the exchange light on the asking for client is flickering, the individual working the device is physically existing, so on.

2.5 Theoretical Overview & Methodologies used

In a years ago, MANETs have been expansion speedily and are progressively being utilized in several applications, starting from military to regular citizen and business utilizes, since forming such networks can be stayed away from the assistance of any infrastructure or connection with a human. A few models are: data accumulation, and virtual classrooms and meetings where laptops, or other cell phones share wireless medium and connect to each other.

As MANETs turn out to be broadly spread, the security problem has turn out to be one of the fundamental attention. For instance, a lot of the routing protocols designed for MANETs suppose that each node inside the network is agreeable and not pernicious [28]. Along these lines, just one client can reason the disappointment of the whole network. There are both passive and active assaults in MANETs. For passive assaults, packets containing mystery data may be eavesdropped, that violates confidentiality. Active attacks, including sending data to incorrect clients into the network, removing data, change the contents of data, and impersonating other clients violate availability, integrity, authentication, and non-repudiation.

Proactive approaches such as cryptography and authentication [29, 30, 31 and 32] were first brought into thought, and a bunches of procedures have been proposed and executed. These applications don't seem to be sufficient. If the flexibility to discover the assaults once it comes into the network is got, these attacks can be stopped from doing any harm to the system or any data. Here is the place the intrusion detection system comes in.

Intrusion detection as announced is a strategy of observing activities in a network or a computer. The technique that is accomplished is named an IDs. An IDs gather action data and afterward analyses it to decide if there are any actions that assaulted the Protection rules. Once an IDs decide that an abnormal behavior or an action which recognize to be an assaults happens, it at that point makes an alert to warn the security manager. Additionally, IDs also can start a correct reaction to the malignant movement.

Despite the fact that there are many IDs methods built for wired networks these days, they're not appropriate for wireless networks because of the variations in their qualities. In this way, those methods should be changed or new strategies should be created to form IDs work successfully in MANETs.

2.5.1 History of Neutrosophic theory and its Applications.

Zadeh presented the degree of membership/truth (t) in 1965 and characterized the fuzzy set [33]. Atanassov presented the degree of nonmembership/falsehood (f) in 1986 and characterized the intuitionistic fuzzy set [34]. Smarandache presented the degree of indeterminacy/neutrality (i) as independent component in 1995 (published in 1998) and defined the neutrosophic set on three components (t, i, f) = (truth, indeterminacy, falsehood).

Historical, The words "neutrosophy" and "neutrosophic" were devised/created by Smarandache in his book [35] 1998. Neutrosophy thinks a proposition, theory, event, concept, or entity, "H" in relation to its opposite, "Anti-H" and that which isn't H, "Non-H. Signified by "Neut-H". Neutrosophy is the premise of neutrosophic probability, neutrosophic logic (NL), neutrosophic statistics, and neutrosophic set (NS).

Neutrosophic Logic is an overall system for union of several present logics, for example, intuitionistic fuzzy logic, paraconsistent logic, intuitionistic logic, etc. Each dimension of the space represents respectively the truth (T), the falsehood (F), and the indeterminacy (I) of the statement under thought, where T, I, F are standard or non-standard real subsets of]-0, 1+[with not really any association between them.

For programming designing proposition the established unit interval [0, 1] might be used. For single valued neutrosophic logic, the sum of the components is: $0 \leq t+i+f \leq 3$ once all three components are independent; $0 \leq t+i+f \leq 2$ when two components are dependent, whereas the third one is independent from them; $0 \leq t+i+f \leq 1$ when all three components are dependent. Whenever three or two of the components T, I, F are independent, one leaves room for incomplete information (sum < 1), paraconsistent and contradictory information (sum > 1), or complete information (sum = 1).

If all three components T, I, F are dependent, then similarly one leaves room for incomplete information (sum < 1), or complete information (sum = 1). By and large, the sum of two components A and B that differ in the unitary interval [0, 1] is: $0 \leq A + B \leq 2 - d°(A, B)$, where d°(A, B) is the degree of dependence between A and B, while d°(A, B) is the degree of independence between A and B.

In 2013 Smarandache developed the NS to n components: (T1, T2, ...; I1, I2, ...; F1, F2, ...). In a brief timeframe. Neutrosophic set has been an essential important tool in all various areas of data mining, decision making, e-learning, engineering, medicine, social science, and some more [7, 8, 9, 10, 11 and 12].

2.5.2 Genetic Algorithms (GAs).

GA is a search-based optimization mechanism. It is habitually utilized to discover optimum or close- optimum answers for troublesome issues which generally could take a time to illuminate. Optimization is the procedure of **making something better**. In some procedure, we have an arrangement of yields as appeared in the following figure 2.5.

Set of inputs
PROCESS
Set of outputs

Figure 2.5 Optimization process

Optimization denotes to get the values of entered data after some process to get the "best" outcomes. Nature has dependably been an extraordinary wellspring of motivation to all humankind.

GAs are adequately randomized in nature, achieving much better than random local search (in which we simply attempt different irregular solutions, monitoring the best up until this point), as they abuse recorded data also. GAs have different benefits which have made them widely used:

- Doesn't necessitate any derivative information (which might not be accessible for some true issues).
- Is quicker and more effective as matched to the conventional techniques.

- Has great parallel abilities.
- Always finds a solution to the issue, which shows signs of improvement over the time.
- Useful when the pursuit space is huge and there are countless parameters.

Similar to any system, GAs additionally hurt from a little restrictions:

- GAs aren't convenient for all issues, particularly issues which are simple and for which subordinate data is accessible.
- Fitness value is figured over and over which may be computationally costly for a few issues.
- There are no assurances on the optimality or the quality of the solution.
- If not executed duly, the GA may not meet the ideal solution.

2.5.3 Self Organize Feature Map (SOFM)

SOFM are a kind of neural network [36, 37]. The scientists were created in 1982. SOMs are suitably called, "Self-Organizing" for the reason that supervision isn't mandatory, through unsupervised competitive learning. "Maps" is because they try to graph their weights to stratify to the specified input data [38]. In unsupervised training, the systems knows how to frame their own groupings of the preparation information without outside help.

For this we need to accept that class membership is comprehensively characterizes by the input patterns sharing common features, and that the network will have the capacity to distinguish those features over the scope of input patterns.

One especially interesting class of unsupervised system is depends on competitive learning, in which the output neurons contend among themselves to be activated, with the outcome that just a single is activated at any one time. This activated neuron is known as a winner-takes all neuron or essentially the winning neuron.

Such competition can be prompted/executed by having lateral inhibition connections (negative feedback ways) between the neurons. The outcome is that the neurons are compelled to sort out themselves. The self-organization procedure includes four noteworthy parts:

- **Initialization**: the connection weights are initialized with little random values.
- **Competition**: For each input pattern, the neurons process their values of a discriminant function which gives the premise for competition. The specific neuron with the smallest value of the discriminant function is confirmed the winner.
- **Cooperation**: The winning neuron decides the spatial area of a topological neighbourhood of excited neurons, in this way giving the premise to participation among neighboring neurons.
- **Adaptation**: The excited neurons diminish their individual values of the discriminant function in relation to the input pattern through appropriate modification of the related association connection weights, such that the response of the winning neuron to the subsequent application of a similar input pattern is enhanced.

The phases of the SOM algorithm can be outlined as pursues:

1. **Initialization** – Pick random values for the initial weight vectors w_j.
2. **Sampling** – Draw a sample training input vector x from the input space.
3. **Matching** – Find the winning neuron **q_j** with weight vector closest to input vector.
4. **Updating** – Apply the weight update equation $w_j[t+1] = w_j[t] + \eta_{qj}[t](x_n[t] - w_j[t])$
5. **Continuation** – keep returning to step 2 until the feature map stops changing.

2.5.4 KDD99 dataset

Lee and Stolfo [39], one of the contributing groups of the DARPA occasion, gave their feature extracted and preprocessed data to Knowledge Discovery and Data Mining (KDD) yearly rivalry [40]. Pfahringer [41] won KDD 99 rivalry utilizing blend of stowing and boosting. Most articles compare their results with winner's result [41].

KDD99 dataset, made in 1999, is extremely ancient for IDS researches [42]. It has been utilized as a reference in numerous researches in the last sixteen years, in February 2016, [39] has been refered eight hundred seventy three times indicated by Google Scholar.

Besides, one hundred forty nine research papers were issued in Science Citation Index Expanded and Emerging Sources Citation Index journals between 2010 and 2015, as indicated in graph 2.6. With respect to graph 2.7, in view of the one hundred forty nine published papers, one hundred forty two, of them has been applied in either in IDS or in machine learning (ML), and one hundred eighteen indexed papers utilize two domains in the same study. These statistics demonstrate that the principle joint of ML, IDS, and data security is KDD99.

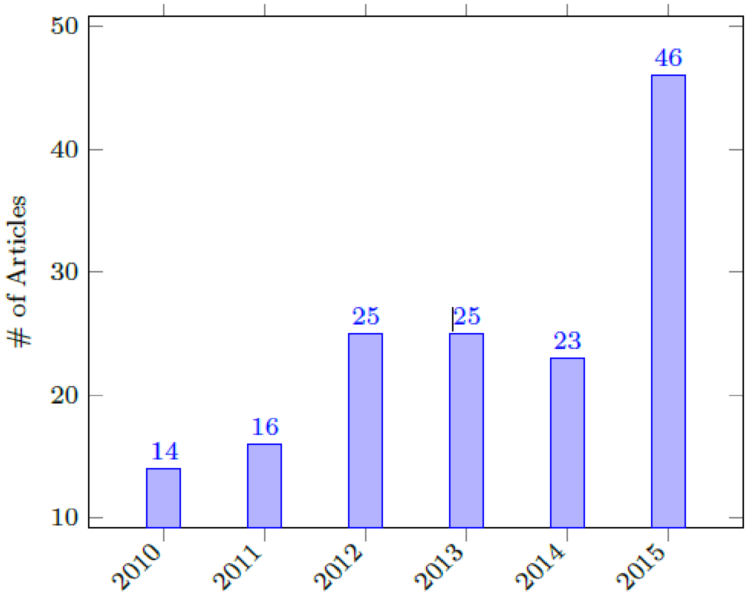


Figure 2.6 KDD99 Dataset Usage by Years.

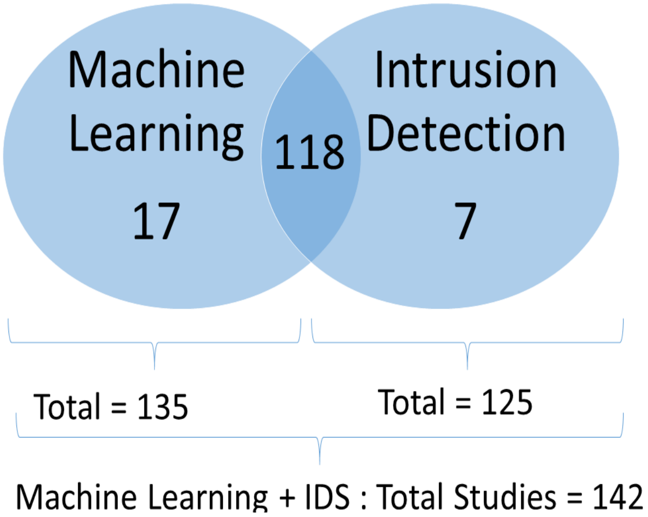


Figure 2.7 Machine Learning and IDS Usage in this Review.

2.5.5 Intrusion detection system (IDS)

ID is a set of strategies and techniques that are utilized to identify suspicious movement in the network and host level. IDS consist of two essential classes: signature-based IDS and anomaly detection systems. Attackers have signatures, similar to computer viruses that can be identified utilizing software. The aim to discover data that have any intrusion-related signatures or particularities identified with Internet protocols. Depended on a set of rules and signatures, the detection system can discover and log suspicious activity and produce cautions. Normally an IDS catches data from the network and applies its rules to that data and detects abnormalities in it.

2.5.5.1 Intrusion Detection in MANETs

IDS are programming or equipment devices (even a blend of both) that consequently scan and monitor occasions in a PC or network, scanning for meddling proof [43]. When planning an IDS to be used in MANET, some thoughts ought to be taken under consideration. There are numerous varieties in the strategy the detection engine ought to carry on concerning a wired network IDS.

In [42] a somewhat total study with respect to this subject, wherever Anjum et al. present the most difficulties to secure MANETs. Sen and Clark [44] have introduced a review in regards to existing ID methodologies for MANETs. Traditional anomaly-based IDS utilize predefined ‘‘normality’’ models to find irregularities inside the system. This can be a methodology that can’t be essentially sent in MANETs, the flexibility of nodes prompts changes of the system topology, expanding the difficulty of the detection strategy.

What's more, since the MANET nodes haven't any settled area, there's no focal management and/or potentially checking point wherever an IDS may be set. This indicates that the detection technique could likewise be conveyed into numerous nodes, and in addition the gathering and investigation of data. Thus, IDS are categorized into cooperative or independent (non-collaborative) [44].

Independent IDS are comprise of IDS agents setted in the nodes of the network and be responsible for watching all nodes inside the network and sending alerts whenever they discover any suspect activity. The fundamental inconvenience of this design is deciding the place of the IDs agents, since nodes are moveable, and several domains of the system might not be checked. Additional downside is that some resources for example bandwidth, central processing unit and/or power are rare in these situations. Augmenting the discovery rate subject to asset restriction is a nondeterministic polynomial time (NP) complete problem and a few algorithms are planned to estimate the solution [42]. A few IDS structured have been suggested to be utilized in MANETs.

2.5.5.2 The Place that the IDS ought to be set in Network Topology

Putting IDS at one or more places depending on the network topology. It likewise relies on what sort of intrusion activities you need to recognize: inner, outer or both. For instance, the need is to recognize just outer intrusion activities, and have one router joining to the Internet, the best location for the IDS might be simply inside the router or a firewall.

On the off chance that you have numerous routes to Internet, you might need to put one intrusion detection systems at every access socket. Be that as it may in the event that you need to identify internal threats too, you might need to put IDS in each network fragment. Be clear that a lot of IDS mean extra effort and extra maintenance charges. So the choice truly relies on the security rule, which characterizes what you truly need to defend from attackers. Graph 2.8 indicate common areas where you can put an IDS.

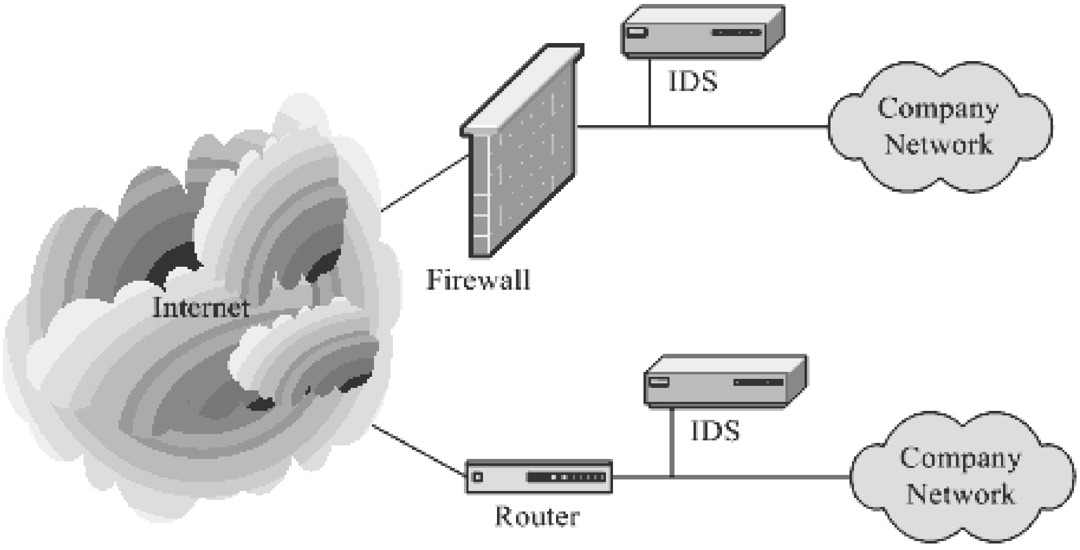


Figure 2.8 Typical locations for an IDS.

As should be obvious from graph 2.8, regularly you ought to set an IDS beside every one of the firewalls and routers.

2.5.5.3 Some Definitions

We have to take in a few definitions identified with security. An essential comprehension of these terms is important to process other confused security ideas.

- **Network IDS (NIDS)**

NIDS are IDS that collect data packets roaming on the network media and compare them to a database of signatures. According, whether a packet is matched with an intruder signature, an alarm is produced or the packet is logged to database.

- **Host IDS (HIDS)**

HIDS are installed as agents on a client. These IDS can see into system and application log files to expose any intruder action. Some of these systems are reactive, implying that they notice you just when something has occurred. Some HIDS are proactive; they can expect the network traffic coming to a specific host and alarm you continuously.

- **Signatures**

A signature is utilized to identify one or numerous kinds of assaults. Signatures might be existing in various fragments of a data packet contingent on the nature of the assaults. Such as, you can discover signatures in the IP header, TCP or UDP header. Typically Intrusion Detection System relies on signatures to get some answers concerning intruder movement. Some vendor-specific Intrusion Detection System require updates from the vendor to add new signatures when a new type of assaults is found.

- **Alerts**

Alarms are any kind of client warning of an intruder movement. At the point when an Intrusion Detection System discovers an attacker, it needs to alert the security person. Alarms might be in the form of pop-up windows, sending e-mail and so on. Alarms saved in databases or log files where the security specialists can be seen in future.

- **False Alarms**

False alarms are cautions produced because of a sign that isn’t an intruder action. For instance, misconfigured interior hosts may some cases transmit mails that reason a rule, causing a false alarm. Now and again you may need to change and tune distinctive or cripple some of the rules to stay away from false cautions.

2.6 Related Work

Security was an incredible test in the specific start of the research in the wireless connections, such a large numbers of scientists have been endeavoring to set up an anchored structure for safe connection in MANETs. We found a numerous studies in this field as a consuming subject in everyday life. Below is a review on them with demonstrating important points.

- **Donald al. [45] in 2003**

Passive eavesdropping, active eavesdropping, and Traffic analysis are three grouped assaults that damage privacy or confidentiality of the session. The scientists reviewed over fluctuated wireless security attacks and their counter estimates cryptographic methods. Then they planned to locate an assimilated secure structure having appropriate authentication technique close to a solid and secure encryption algorithm utilizing block cipher.

- **Ravi al. [46] in 2005**

The scientists prescribed an efficient key agreement structure especially Diffie-Hellman and Chinese Remainder Theorem (DHCRT), which, there is no pre-shared mystery between the individuals and furthermore the service of a confided authority or a gathering controller isn't requisite. DHCRT utilizes the DH key exchange and furthermore the CR Theorem for effective key agreement of Symmetric Encryption. In any case, DHCRT suffers from man-in-the-middle assault [47].

- **Tina al. [48] in 2005**

The scientists focused on authentication and identification in MANETs. For alleviating the identity assaults, the scientists planned to relate the message sender with an area and utilize this area data to discover identity. As indicated by this technique, a Verifying Node (VN) verifies the area of sending nodes utilizing a mix of signal properties, global positioning systems (GPS), and trusted-peer coordinated effort for identification purposes. They see identification depended on triangular situating framework where the three key focuses are the trusted peer, VN, and sender. At that point, the three functions are utilized to calculate the sender's area. Besides, calculation depended on relative position isn't a proficient methodology for recognizing a lying node.

- **Shichun al. [49] in 2006**

The researchers presented an optimized improved to State Space Search Algorithm (SSS). The security of this model is depends on Elliptic Curve Cryptography (ECC) [50]. The key of elliptic bend cryptography has the size which is significantly less than RSA cryptography. Subsequently, the model ought to be so vital in applications with restricted memory and computing power.

- **Wei al. [51] in 2006**

The researchers attracted our consideration to ID-based Key Management (IKM) cryptography. IKM as a certificate-less solution helps public keys of mobile nodes to be specifically logical utilizing their known IDS and some normal data. Thus, it expels the need of certificate-based authenticated public-key distribution, which is essential in conventional public-key management schemes.

- **PI Jian al. [52] in 2006**

The scientists contemplated a plan to dispose of the customary identity authentication mechanism for PKI and identity. The authentication code (AC) stays unaltered in all sessions and the session key is distinctive in each session. Their performance investigation demonstrates that, existential imitation assaults can be averted by their plan. Despite the simulation outcomes of this technique obtain high efficiency versus Sybil assault, the plan is helpless to the key imitating assault.

- **Chris al. [53] in 2006**

The scientists recommended an application space particular detection technique for MANETs. The plan is depends on the way that Sybil nodes in MANET normally transfer in groups. The scientists demonstrated that the assessment of topographical area examples of groups of identifiers, which are moving together, can conceivably show the nearness of a node propelling a Sybil assaults.

- **Marianne A. et al. [54] in 2007**

The researchers contemplated diverse limit cryptography systems and State Space Search (SSS). They enrolled numerous difficulties alongside research alternatives in the field of limit authentication and cryptography. The real difficulties of SSS are legitimacy time of the partial key or mystery sharing, choices of the ideal edge level, dynamic alteration of the partial key legitimacy time and conduct of adulterated nodes utilizing erroneous partial keys.

- **Pierre al. [55] in 2007**

The paradigm gives secure common authentication and explicit key formation over an insecure system. The researchers guaranteed that this paradigm is more protected than any other present basic Key Agreement (SKA) protocols.

- **A. Macedo [56] in 2008**

The researchers presented Address-based Cryptography Scheme (ACS) as a security show for MANETs. ACS is a blend of public key cryptography and Ad hoc node address. ACS nodes are straightforward deliverable from their known Ad hoc node address and some normal data, for example quantity of nodes in the system, entrance time and departure time. The scientists requested that in this technique MANETs are saved from the eavesdropping and masquerading. ACS broadcasts scrambled message including its own private key which expands security dangers for MANETs.

- **Mengbo al. [57] in 2009**

The researchers suggested that ID based cryptography plot is powerless to the key imitating assault, where an active foe can block and suitably adjust the messages traded between two entities, and force the two entities to admit the similar session key.

- **Yuguang al. [58] ~~in 2009~~**

The researchers presented distinctive cryptographic systems for MANETs. The scientists proposed to utilize Identity Based Cryptography [59], in view of the restricted asset requirements of MANETs.

Chapter Summary

This chapter provides brief overview to MANETs and the history of this special type of networks and theoretical overview of methodologies used like Self Organize Feature Map (SOM), Genetic Algorithm (GA), Neutrosophic theory and the KDD CUP 99 Data set, are described to get proper understandings of field related backgrounds.

Chapter 3

Preprocessing phase of the neutrosophic knowledge discovery system

3.1 Introduction

Building a neutrosophic IDS is a feasible solution in dealing with ambiguity circumstances. The neutrosophic IDS is composed of two main sub modules: the preprocessing phase and the network attacks classification phase. The preprocessing phase concentrates on preparing the network data in a format suitable for the classification module.

This chapter is concerned in reformatting the regular data in the KDD data set [13] into neutrosophic format $(x , \mu_A(x), \sigma_A(x), \nu_A(x))$ where x is the value of attribute data, $\mu_A(x)$ is the membership (MEM) value, $\sigma_A(x)$ is the indeterminacy value and $\nu_A(x)$ is the non-membership (NON-MEM) value of the x in the data space. Although, manual procedures of human interfering could help in reformatting the neutrosophic network data, the huge amount of data would be an obstacle for this kind of help. Semi manual techniques could be utilized here. Human expert could help in preparing a subset of common values in the network data. A machine learning technique could be utilized to learn from this subset and complete the process of reformatting.

SOFM [36] is a neural network with unsupervised learning capabilities Figure 3.1. SOFM would be utilized to learn the $\mu_A(x), \sigma_A(x), \nu_A(x)$ functions from the input subset provided by the human expert in the network field. The SOFM was used for generating membership functions for fuzzy variables by Chih-Chung Yang and N.K. Bose [60]. Their research was an inspiration for this thesis to generate the neutrosophic functions in the same methodology.

The SOFM are used to define the membership, nonmembership and indeterminacy functions of the KDD network attacks data [13] available in the UCI machine learning repository for further processing in IDS and knowledge discovery. The focus of this Chapter is on how SOFM could be utilize to deal with neutrosophic information. In this way, the information being related is all neutrosophic variables.

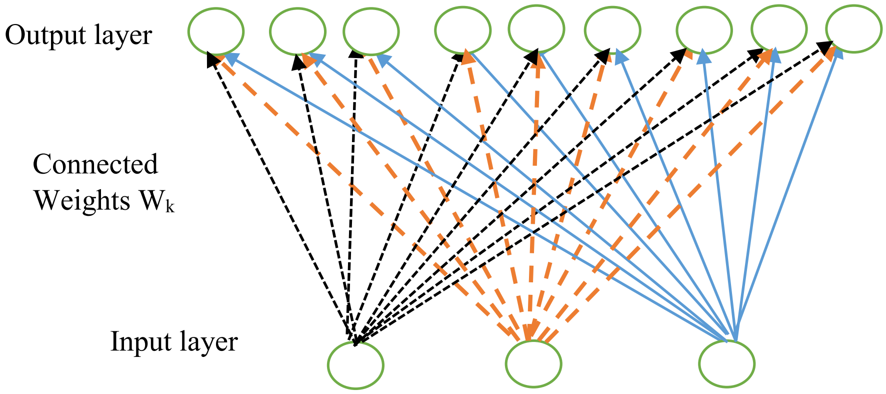


Figure 3.1. The self-Organizing (Kohonen) map

3.2 DESIGNING SOFM for Modeling Neutrosophic Variables

A neutrosophic set is an object having the form $\boldsymbol{A} = (\boldsymbol{x}, \boldsymbol{\mu_A}(\boldsymbol{x}), \boldsymbol{\sigma_A}(\boldsymbol{x}), \boldsymbol{\nu_A}(\boldsymbol{x}))$ where $\boldsymbol{\mu_A}(\boldsymbol{x})$, $\boldsymbol{\sigma_A}(\boldsymbol{x})$, and $\boldsymbol{\nu_A}(\boldsymbol{x})$which represent the degree of MEMEBERSHIP function (namely), the degree of indeterminacy (namely), and the degree of NON-MEMEBERSHIP (namely $\boldsymbol{\mu_A}(\boldsymbol{x})$ the degree of indeterminacy namely ($\boldsymbol{\sigma_A}(\boldsymbol{x})$) and the degree of non-member ship (namely $\boldsymbol{\nu_A}(\boldsymbol{x})$) respectively of each element $\boldsymbol{x} \,\boldsymbol{\epsilon}\, \mathbf{X}$ to the set $\boldsymbol{A}$ where

$$\mathbf{0}^- \le \left(\boldsymbol{\mu}_\mathbf{A}(\mathbf{x}), \boldsymbol{\sigma}_\mathbf{A}(\mathbf{x}), \boldsymbol{\nu}_\mathbf{A}(\mathbf{x})\right) \le \mathbf{1}^+ \text{ and } \quad (1)$$

$$\mathbf{0}^- \le \left(\boldsymbol{\mu}_\mathbf{A}(\mathbf{x}) + \boldsymbol{\sigma}_\mathbf{A}(\mathbf{x}) + \boldsymbol{\nu}_\mathbf{A}(\mathbf{x})\right) \le \mathbf{3}^+ \quad (2)$$

Smarandache introduced the following: Let T, I, F be real standard or nonstandard subsets of $]\,\mathbf{0}^-, \mathbf{1}^+[$, with Sup_T=t_sup, inf_T=t_inf; Sup_I=i_sup,inf_I=i_inf; Sup_F=f_sup, inf_F=f_inf ; n-sup=t_sup+i_sup+f_sup n-inf=t_inf+i_inf+f_inf, ; $\mathbf{T}, \mathbf{I}, \mathbf{F}$ are called neutrosophic components.

Fuzzy system [61] depends on the fuzzy variable to build the fuzzy rules and equations. SOFM was utilized in generating fuzzy MEMEBERSHIP function [62]. Because neutrosophic system depends partially on MEMEBERSHIP function, SOFM could be used to define neutrosophic MEMEBERSHIP function for the variables. Furthermore, SOFM could be utilized to define NONMEMEBERSHIP. Then, the INDETERMINACY function can be calculated via equation 2 to complete the neutrosophic variable modeling.

“Self-Organizing” is because no supervision is required. SOFMs learn on their own through unsupervised competitive learning. “Maps” is because they try to chart their weights to stratify to the given input data [38]. The nodes in a SOFM network attempt to become like the inputs presented to them. Assuming an input vector of $z_n = [x_n y_n]^T$, the SOFM will go through two main phases. The first is the training phase which accept the input data and apply the calculations in equations 3, 4 and 5 to find the winning neuron and update the network connections. The second is the retrieving phase which retrieves the weight connections associated with the winning neuron for each new instance of input data.

$$q(\mathbf{x_n}) = \min_{\forall j} \left\| \mathbf{x_n} - \mathbf{w_j} \right\| ; \qquad (3)$$

$$\boldsymbol{\eta_{qj}[t]} = \begin{cases} \boldsymbol{\mu[t] j \epsilon N_q} \\ \boldsymbol{0 j \notin N_q} \end{cases} \qquad (4)$$

$$\mathbf{w_j[t+1] = w_j[t] +} \boldsymbol{\eta_{qj}}\mathbf{[t]\left(x_n[t] - w_j[t]\right)} \qquad (5)$$

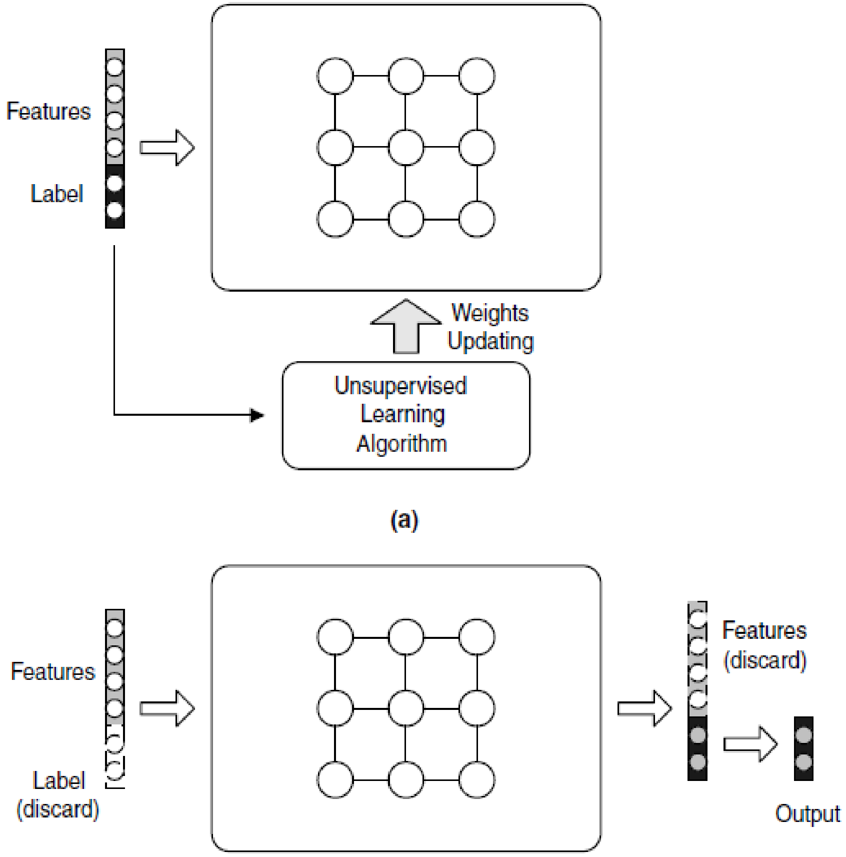
With, in this case, the weight vector$w_j = \left[w_{j1} w_{j2} \ldots \ldots \ldots . . w_{jd} x_{j(d+c)}\right]^T$ $=\left[w_{jd} w_{jc}\right]$,$j = 1, \ldots, J$.After the learning phase, the SOFM can be considered as a MEMEBERSHIP generation network just like its counterpart, the feed forward multilayer neural network trained with a supervised learning algorithm [63]. However, in the retrieving phase, it is not as straightforward as in the case of the feed forward multilayer neural network and some modification, described next, is required.

In the retrieving phase, the input feature vector is only $\mathrm{x_n}$. Therefore, the input feature vector will find the best matching neuron q by considering only the weight sub vector $\mathrm{w_{jd}} = [\mathrm{w_{j1}} \ \dots\dots \mathrm{w_{jd}}]^{\mathrm{T}}$ related to input features, which is Eq. 3. Subsequent getting the triumphant neuron q, the output of SOFM is the weight sub vector $\mathrm{w_{qc}} = [\mathrm{w_{q(d+1)}} \ \dots\dots\dots\dots \mathrm{w_{q(d+c)}}]^{\mathrm{T}}$ related with the labeling information. Likewise, it is the neutrosophic MEMEBERSHIP created by SOFM.

The SOFM used to identify the MEMEBERSHIP, NONMEMEBERSHIP for the network features. Then those two function would be used to define the INDETERMINACY function basing on the neutrosophic set definitions Eq. 1 and 2.

The neurons in the SOFM emulate the inputs applied to them to achieve the learning process. The topological relationships between input data are conserved when mapped to a SOFM network. This is a very important capability when inspect complex data. The age of neutrosophic MEMEBERSHIP function through SOFM has, so far been a two-phase execution [64].

The initial step creates the correct groups. At that point, the neutrosophic MEMEBERSHIP function is created as per to the groups in the first stage. Be that as it may, it is conceivable to combine the two-stage process and produce the neutrosophic MEMEBERSHIP function straightforwardly amid the learning stage, the proposed procedure is explained in Figure 3.2. The principle thought is to expand the input feature vector with the group labeling information.



(a)

(b)

Figure 3.2 (a) learning phase and (b) retrieving phase

For the neutrosophic MEMEBERSHIP function, a key advance in the introduced mechanism is to consolidate the input feature vector $x_n = [x_{n1}x_{n2} \ldots \ldots \ldots . . x_{nd}]^T$ with the vector$y_n = [y_{n1}y_{n2} \ldots \ldots \ldots . . y_{nc}]^T$ coding the subset labeling information of truth degrees. The dimensions of x_n and y_n are respectively, the quantity of input features d and the quantity of subset labels c. That is, a new vector z_n of dimension $c + d$ is built as per to $z_n = [x_n y_n]^T$=$[x_n \quad 0]^T + [0 \quad y_n]^T$. In the learning phase, the newly constructed z_n will be the Input feature vector to SOFM. The weight updating as indicated by Eq.5. The procedure is the same for the neutrosophic NONMEMEBERSHIP function. Instead of the degree of truth, the z_n vector will hold the degree of falsity for different labels of the variable values.

According to the neutrosophic set definition, the MEMEBERSHIP, NONMEMEBERSHIP and INDETERMINACY functions are independent but with one condition which is provided in Eq. 2. The summation of the three values for a neutrosophic label should not exceed 3. Hence, the indeterminacy function could be defined by knowing the MEMEBERSHIP and NONMEMEBERSHIP function as follows.

$$0^{-} - \left(\mu_A(x) + \sigma_A(x)\right) \leq \nu_A(x) \leq 3^{+} - (\mu_A(x) + \sigma_A(x)) \qquad (6)$$

Then the resulted INDETERMINACY function should be normalized in Eq.7 to fit in the [0, 1] interval according to the second condition in Eq. 1. The algorithm and flowchart of the proposed technique is illustrated in figure 3.3 and 3.4 respectively.

$$Z_i = \frac{X_i - Min\ (X_i)}{Max(X_i) - Min(X_{i)}} \qquad (7)$$

Where $x = (x_1 ,……..,x_n)$ $x=(x_1,...,x_n)$ and Z_i is normalized data

```
Input: input_data vectors(Trainig_data set), Input_dim, output_dim,
Output: neutrosophic variable MEMEBERSHIP , NONMEMEBERSHIP and indeterminacy functions
//MEMEBERSHIP  function generation
      1. Trainig_Data←Read_data(MEMEBERSHIP _data)
      2. MEMEBERSHIP _data← SOFM(Trainig_data, Input_dim, output_dim)
      3. Draw (MEMEBERSHIP _data)
      4. Trainig_Data←Read_data(NON_MEMEBERSHIP _data)
      5. NON_Membership_data← SOFM(Trainig_data, Input_dim, output_dim)
      6. Draw (NON_MEMEBERSHIP _data)
      7. Indeterminancy← Calculate_ind(MEMEBERSHIP_data, NON_MEMEBERSHIP _data)
      8. Draw (Indeterminancy)
End
  Function SOFM
  Input: Trainig_data, Input_dim, output_dim
  Output: Output _Function
     Initialize_SOFM (input_neurons, output_neurons)
     Randomly_Initialize_SOFM_Weights ()
     While Error>threshold Do
     Foreach Record in Trainig_data
     Input_Record ();
     Winning_neuron$\mathbf{q_j}$=$\mathbf{q}(\boldsymbol{x_n}) = \min_{\forall j} \|\boldsymbol{x_n} - \boldsymbol{w_j}\|$;
     Update_weights (Winning_neuron$\mathbf{q_j}$);
     Endforeach
     Error =Calculate_ErrorRate ();
     End while
     Retrieving_phase ();
     Output Function←Network_Weights)
     End fun
```

```
**Update_weights**
**Input**: Winning_neuron$q_j$
**Output**: Update_weights
    1. Find (Winning_neuron$q_j$)
    2. $\eta_{qj}[t] = \begin{cases} \mu\,[t] j\epsilon N_q \\ 0 \quad j \notin N_q \end{cases}$
    3. $w_j[t+1] = w_j[t] + \eta_{qj}[t]\big(x_n[t] - w_j[t]\big)$
    4. Output (Update_weights)
    5. End fun
```

Figure 3.3. The Algorithm

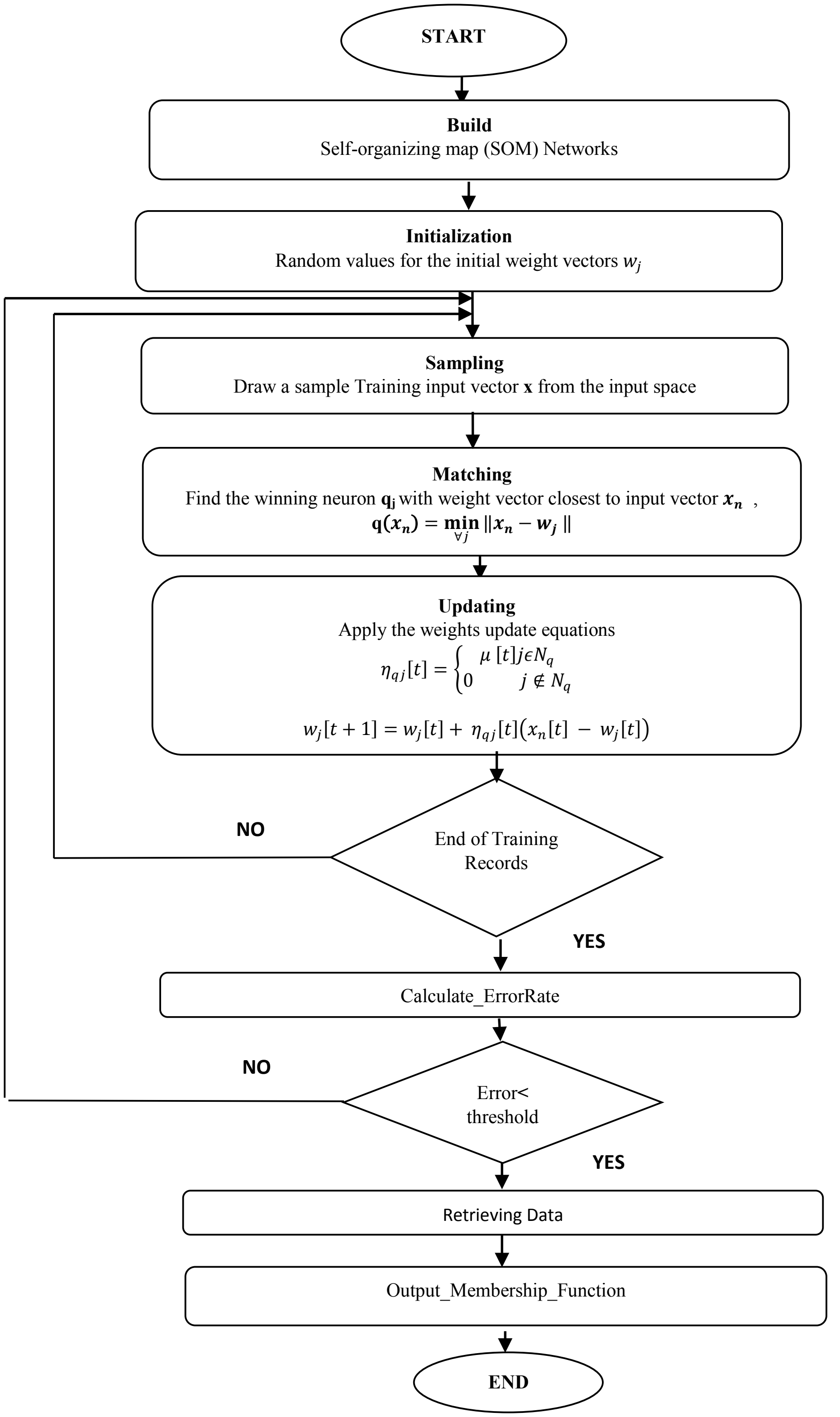

START
Build
Self-organizing map (SOM) Networks
Initialization
Random values for the initial weight vectors w_j
Sampling
Draw a sample Training input vector **x** from the input space
Matching
Find the winning neuron $\mathbf{q_j}$ with weight vector closest to input vector $\boldsymbol{x_n}$,
$\mathbf{q}(\boldsymbol{x_n}) = \min_{\forall j} \|\boldsymbol{x_n} - \boldsymbol{w_j}\|$
Updating
Apply the weights update equations
$\eta_{qj}[t] = \begin{cases} \mu\,[t] j \epsilon N_q \\ 0 \quad j \notin N_q \end{cases}$
$w_j[t+1] = w_j[t] + \eta_{qj}[t](x_n[t] - w_j[t])$
NO
End of Training Records
YES
Calculate_ErrorRate
NO
Error< threshold
YES
Retrieving Data
Output_Membership_Function
END

Figure 3.4. Algorithm Flowchart

3.3 Experimental Results

3.3.1 The MANET network data

The KDD-99 dataset's [13] connections are represented by 41 features; the features in Columns 2, 3, and 4 are the protocol type, the service type, and the flag. The value of the protocol type may be TCP, UDP, or ICMP; the service type could be one of the 65 different network services such as HTTP and PRIVATE; and the flag has 9 possible values such as SF or S0. After reducing KDD-99 features from each record, pre-processing will be done by reverse each feature from text or symbolic into numerical form. So for each text or symbolic an Integer code is assigned. As follow:

Table 1. Numeric Values of KDD Features

PROTOCOL TYPE	FEATURE VALUE	SERVICE	FEATURE VALUE	FLAG	FEATURE VALUE
TCP	1	HTTP	1	SF	1
UDP	2	PRIVATE	2	S0	2
	3	FTP_DATA	3	REJ	3
		SMTP	4	RSTR	4
		.	.	.	.
		.	.	.	.
		POP_2	65	S2	9

3.3.2 SOFM for modeling neutrosophic variable

The proposed technique is concerned by the pre-processing phase of the neutrosophic knowledge discovery system. Self-Organized Feature Maps (SOFM) are unsupervised artificial neural networks that were used to build fuzzy MEMEBERSHIP function, hence they could be utilized to define the neutrosophic variable as well. SOFMs capabilities to cluster inputs using self-adoption techniques have been utilized in generating neutrosophic functions for the subsets of the variables.

The SOFM are used to define the MEMEBERSHIP, NONMEMEBERSHIP and INDETERMINACY functions of the KDD network attacks data available in the UCI machine learning repository for further processing in knowledge discovery. Our experimental Results Shows the features and their corresponding functions.

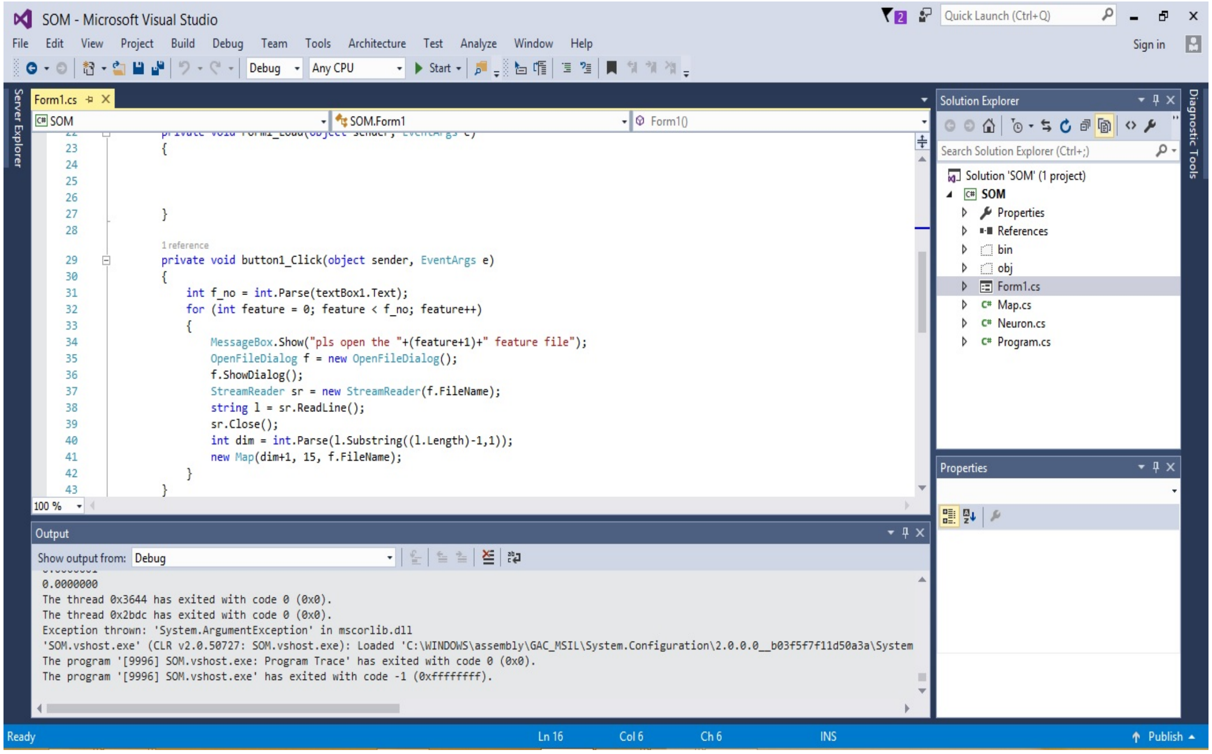
SOM parameters

- Dimensions :
  Number of input dimensions: **3 inputs [value, Normal, Up-normal]**
- Outputs :Neuron: 225 neuron
- Error rate threshold: **0.15**

System features:

- Processor: Intel(R) Core (TM) i3-3227U CPU @ 1.90GHZ 1.90 GHZ
- Memory (RAM): 4.00 GB (3.87 GB usable)
- System type: 64-bit operating system, x64- based processor
- Windows edition: windows 8.1

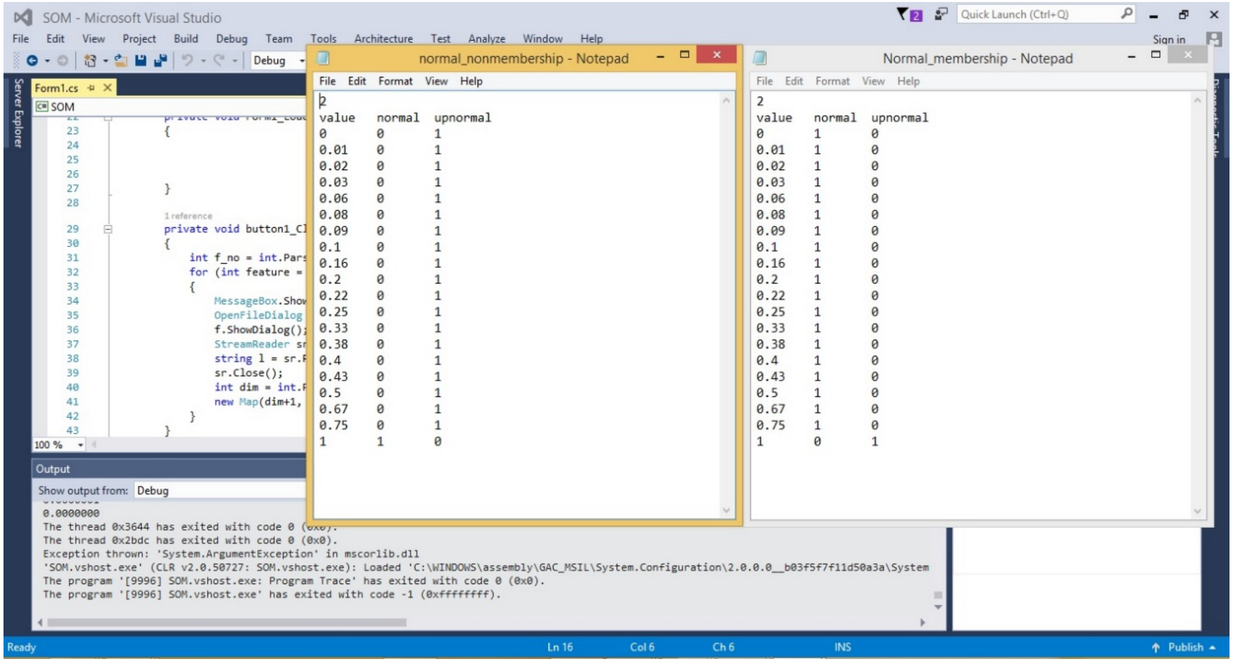
SOFM program simulation is implemented by C# figure 3.5. Each feature from the KDD data will have two different files for MEMEBERSHIP, NONMEMEBERSHIP assumption values figure 3.6. These files are provided to the SOFM program to build these two functions. The generated output file contains the values of elected neuron figure 3.7 and figure 3.8.



```
        private void Form1_Load(object sender, EventArgs e)
        {



        }

        private void button1_Click(object sender, EventArgs e)
        {
            int f_no = int.Parse(textBox1.Text);
            for (int feature = 0; feature < f_no; feature++)
            {
                MessageBox.Show("pls open the "+(feature+1)+" feature file");
                OpenFileDialog f = new OpenFileDialog();
                f.ShowDialog();
                StreamReader sr = new StreamReader(f.FileName);
                string l = sr.ReadLine();
                sr.Close();
                int dim = int.Parse(l.Substring((l.Length)-1,1));
                new Map(dim+1, 15, f.FileName);
            }
        }
```

```
0.0000000
The thread 0x3644 has exited with code 0 (0x0).
The thread 0x2bdc has exited with code 0 (0x0).
Exception thrown: 'System.ArgumentException' in mscorlib.dll
'SOM.vshost.exe' (CLR v2.0.50727: SOM.vshost.exe): Loaded 'C:\WINDOWS\assembly\GAC_MSIL\System.Configuration\2.0.0.0__b03f5f7f11d50a3a\System
The program '[9996] SOM.vshost.exe: Program Trace' has exited with code 0 (0x0).
The program '[9996] SOM.vshost.exe' has exited with code -1 (0xffffffff).
```

Figure 3.5 SOFM program simulation is implemented by C#.



```
            {



            }

            1 reference
            private void button1_Cl
            {
                int f_no = int.Pars
                for (int feature =
                {
                    MessageBox.Show
                    OpenFileDialog
                    f.ShowDialog();
                    StreamReader sr
                    string l = sr.R
                    sr.Close();
                    int dim = int.P
                    new Map(dim+1,
                }
            }
```

```
0.0000000
The thread 0x3644 has exited with code 0 (0x0).
The thread 0x2bdc has exited with code 0 (0x0).
Exception thrown: 'System.ArgumentException' in mscorlib.dll
'SOM.vshost.exe' (CLR v2.0.50727: SOM.vshost.exe): Loaded 'C:\WINDOWS\assembly\GAC_MSIL\System.Configuration\2.0.0.0__b03f5f7f11d50a3a\System
The program '[9996] SOM.vshost.exe: Program Trace' has exited with code 0 (0x0).
The program '[9996] SOM.vshost.exe' has exited with code -1 (0xffffffff).
```

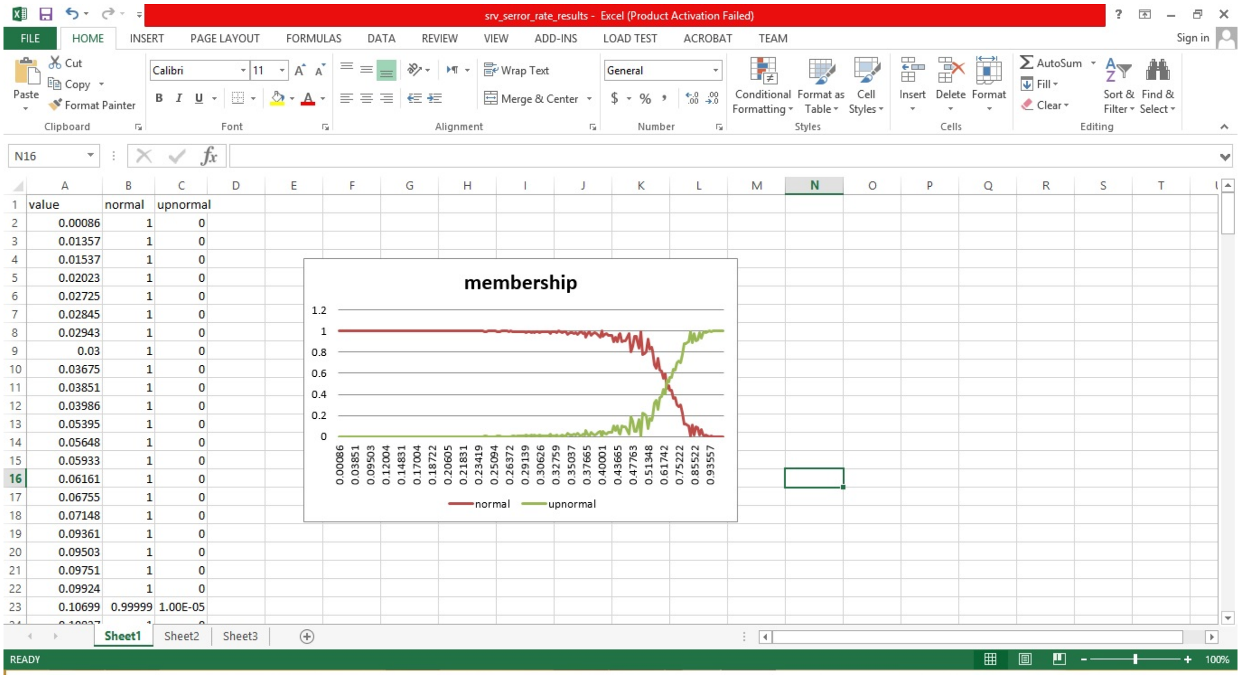
normal_nonmembership - Notepad

```
2
value   normal  upnormal
0       0       1
0.01    0       1
0.02    0       1
0.03    0       1
0.06    0       1
0.08    0       1
0.09    0       1
0.1     0       1
0.16    0       1
0.2     0       1
0.22    0       1
0.25    0       1
0.33    0       1
0.38    0       1
0.4     0       1
0.43    0       1
0.5     0       1
0.67    0       1
0.75    0       1
1       1       0
```

Normal_membership - Notepad

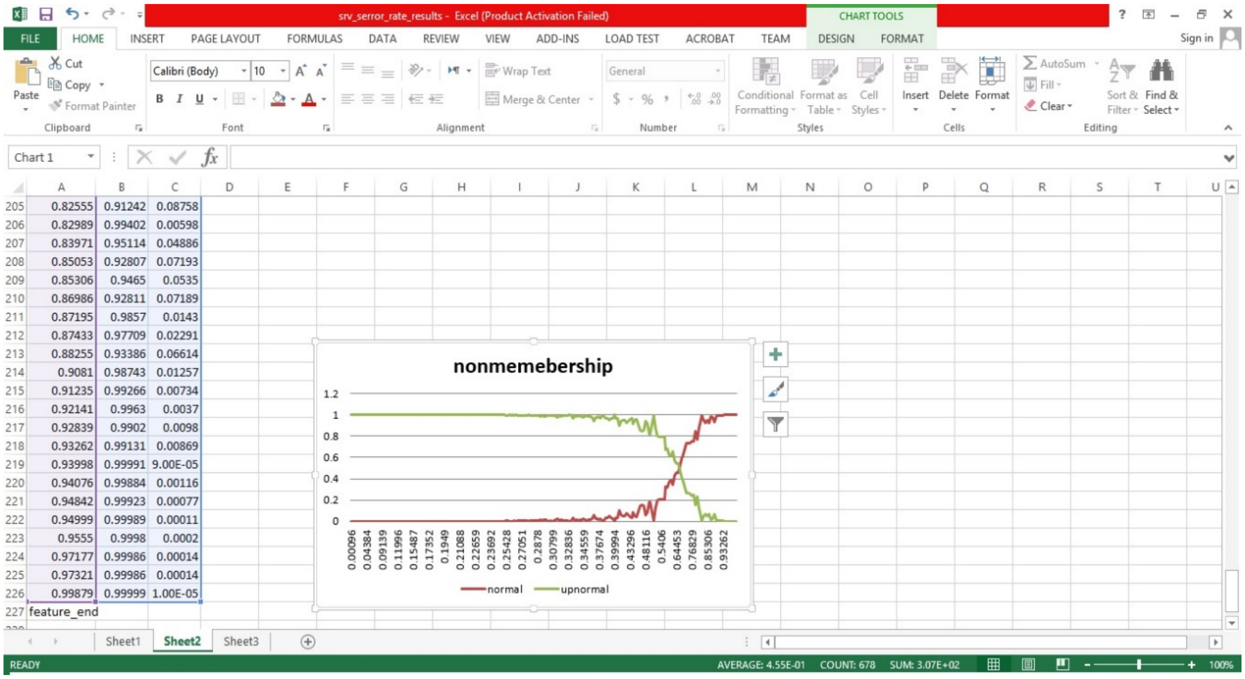
```
2
value   normal  upnormal
0       1       0
0.01    1       0
0.02    1       0
0.03    1       0
0.06    1       0
0.08    1       0
0.09    1       0
0.1     1       0
0.16    1       0
0.2     1       0
0.22    1       0
0.25    1       0
0.33    1       0
0.38    1       0
0.4     1       0
0.43    1       0
0.5     1       0
0.67    1       0
0.75    1       0
1       0       1
```

Figure 3.6 Different files for MEMEBERSHIP, NONMEMEBERSHIP assumption values



	A	B	C
1	value	normal	upnormal
2	0.00086	1	0
3	0.01357	1	0
4	0.01537	1	0
5	0.02023	1	0
6	0.02725	1	0
7	0.02845	1	0
8	0.02943	1	0
9	0.03	1	0
10	0.03675	1	0
11	0.03851	1	0
12	0.03986	1	0
13	0.05395	1	0
14	0.05648	1	0
15	0.05933	1	0
16	0.06161	1	0
17	0.06755	1	0
18	0.07148	1	0
19	0.09361	1	0
20	0.09503	1	0
21	0.09751	1	0
22	0.09924	1	0
23	0.10699	0.99999	1.00E-05

Figure 3.7 the generated output file contains the values of elected neuron, MEMEBERSHIP

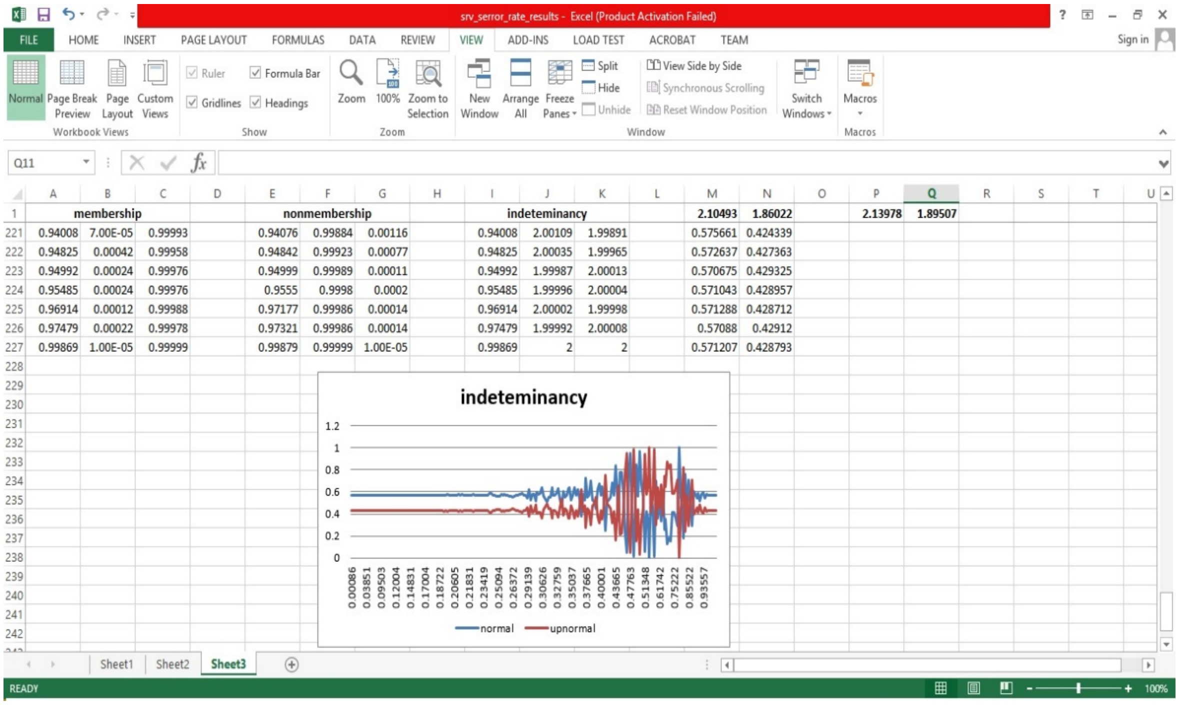


	A	B	C
205	0.82555	0.91242	0.08758
206	0.82989	0.99402	0.00598
207	0.83971	0.95114	0.04886
208	0.85053	0.92807	0.07193
209	0.85306	0.9465	0.0535
210	0.86986	0.92811	0.07189
211	0.87195	0.9857	0.0143
212	0.87433	0.97709	0.02291
213	0.88255	0.93386	0.06614
214	0.9081	0.98743	0.01257
215	0.91235	0.99266	0.00734
216	0.92141	0.9963	0.0037
217	0.92839	0.9902	0.0098
218	0.93262	0.99131	0.00869
219	0.93998	0.99991	9.00E-05
220	0.94076	0.99884	0.00116
221	0.94842	0.99923	0.00077
222	0.94999	0.99989	0.00011
223	0.9555	0.9998	0.0002
224	0.97177	0.99986	0.00014
225	0.97321	0.99986	0.00014
226	0.99879	0.99999	1.00E-05
227	feature_end		

Figure 3.8 the generated output file contains the values of elected neuron, NONMEMEBERSHIP

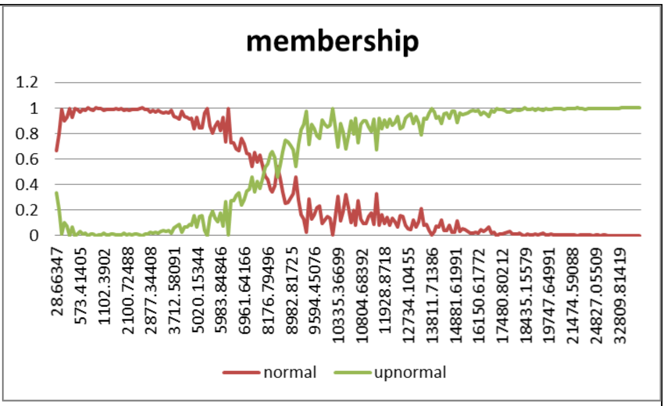
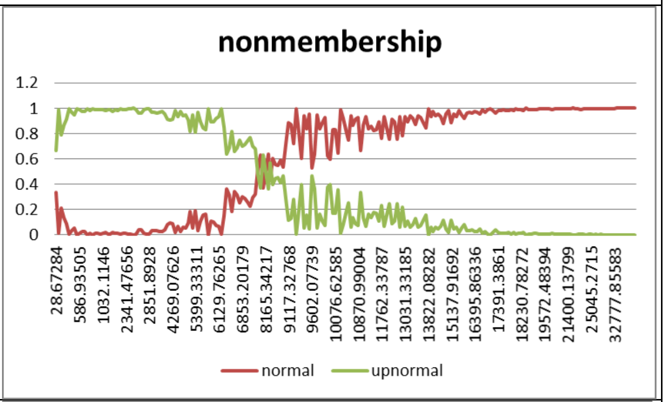
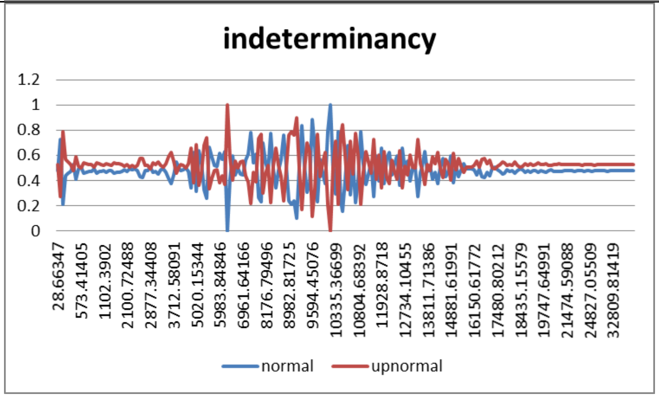
Figure 3.9 The INDETERMINACY function is calculated from the MEMEBERSHIP and NONMEMEBERSHIP data according to Eq. 6. The INDETERMINACY function is further normalized to fit within the interval $]0^{-}, 1^{+}[$ the results graphs shown below figure 3.10 (a, b, c, d and e), which indicate the relation of MEMEBERSHIP, NON-MEMEBERSHIP and INDETERMINACY functions.

The graphical representation shows of the duration, dst_bytes, hot, count and Srv_serror_rate variables from the KDD data set. The first figure for each variable represent the MEMBERSHIP function while the second figure represent the NON-MEMBERSHIP function. It is clear that the two figures are the complement of each other. The third figure is the INDTERMINACY function that fills the unknown gap left by the MEM and NON-MEM functions.

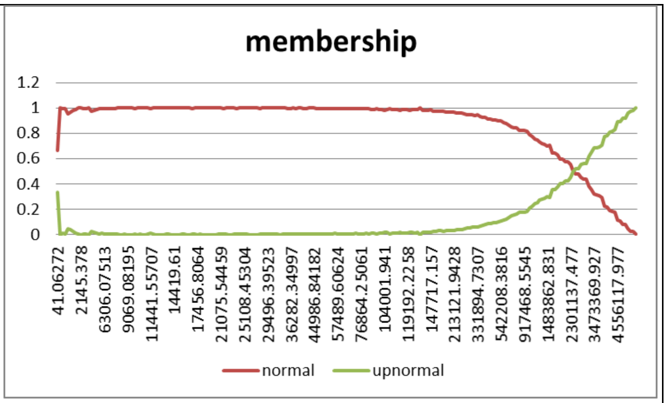
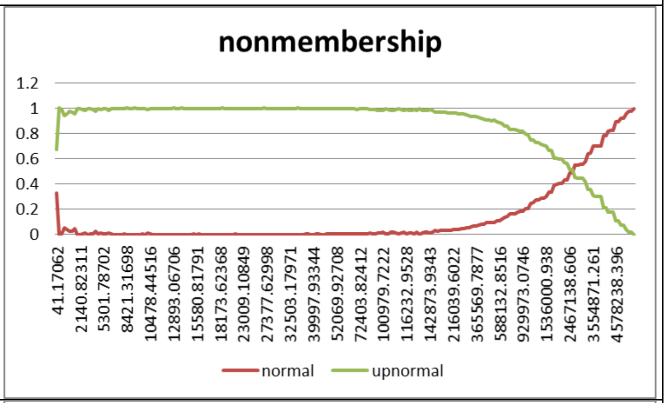
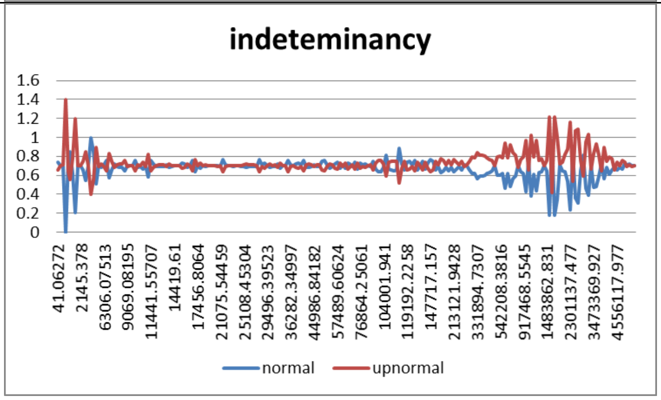


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	membership				nonmembership				indeteminancy				2.10493	1.86022		2.13978	1.89507
221	0.94008	7.00E-05	0.99993		0.94076	0.99884	0.00116		0.94008	2.00109	1.99891		0.575661	0.424339			
222	0.94825	0.00042	0.99958		0.94842	0.99923	0.00077		0.94825	2.00035	1.99965		0.572637	0.427363			
223	0.94992	0.00024	0.99976		0.94999	0.99989	0.00011		0.94992	1.99987	2.00013		0.570675	0.429325			
224	0.95485	0.00024	0.99976		0.9555	0.9998	0.0002		0.95485	1.99996	2.00004		0.571043	0.428957			
225	0.96914	0.00012	0.99988		0.97177	0.99986	0.00014		0.96914	2.00002	1.99998		0.571288	0.428712			
226	0.97479	0.00022	0.99978		0.97321	0.99986	0.00014		0.97479	1.99992	2.00008		0.57088	0.42912			
227	0.99869	1.00E-05	0.99999		0.99879	0.99999	1.00E-05		0.99869	2	2		0.571207	0.428793			

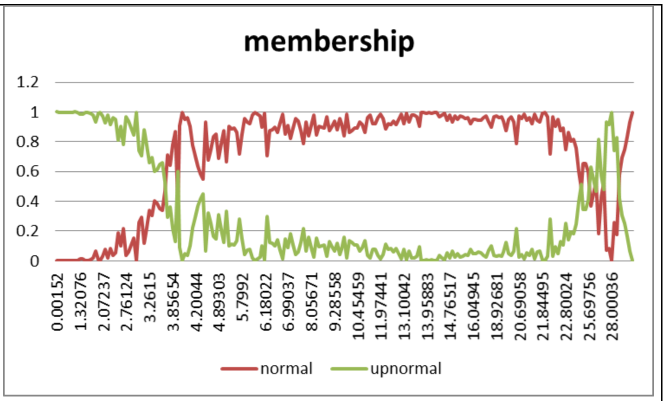
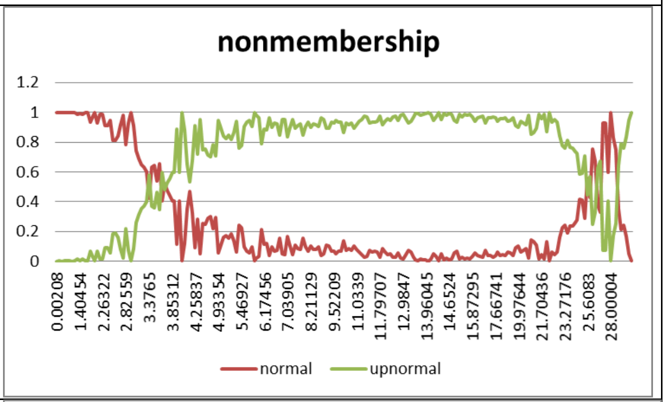
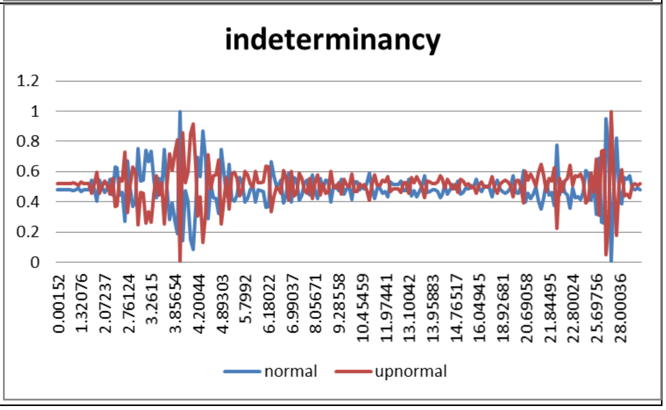
Figure 3.9. The INDETERMINACY function is calculated from the MEMEBERSHIP and NONMEMEBERSHIP.

MEMEBERSHIP	
NONMEMBERSHIP	
INDETEMINANCY	

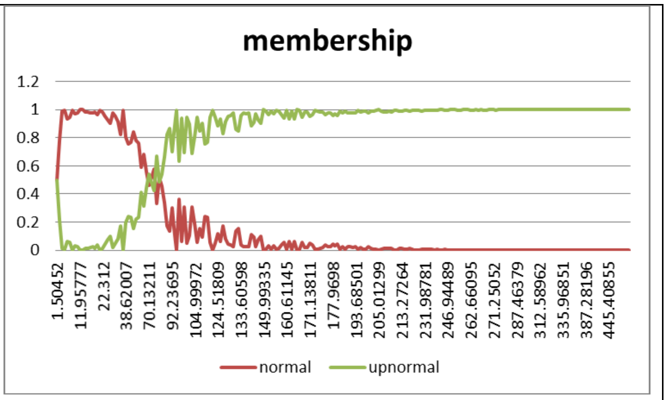
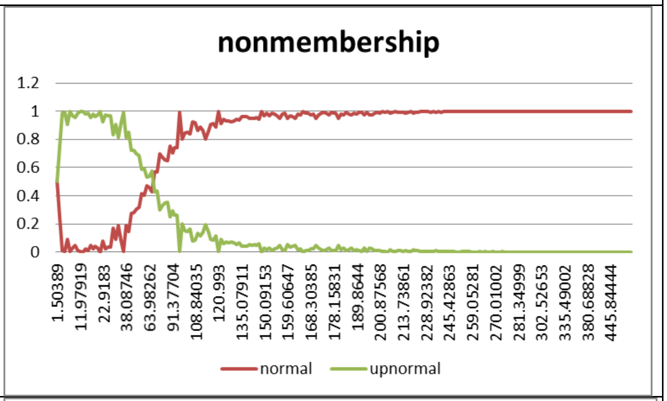
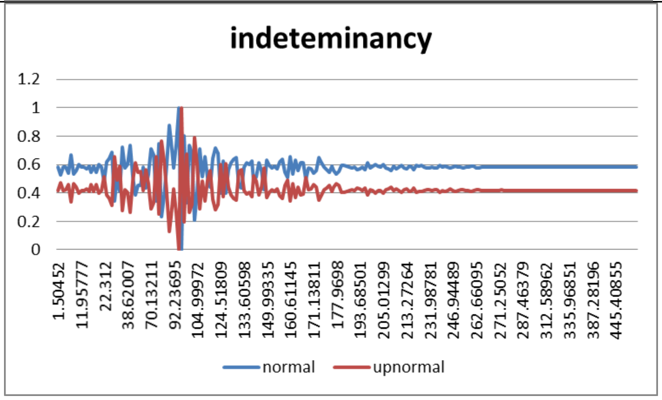
a: Attribute name: Duration

MEMEBERSHIP	
NONMEMBERSHIP	
INDETEMINANCY	

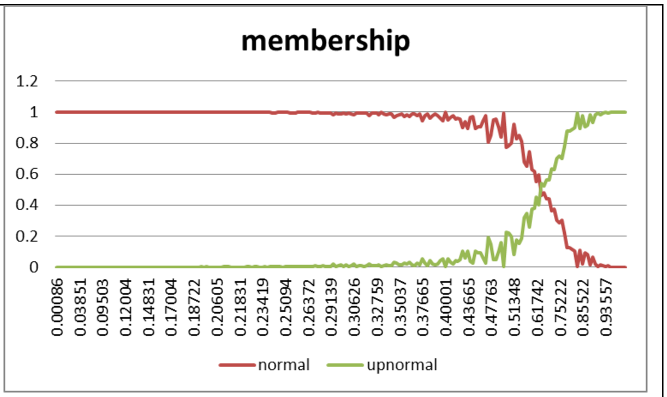
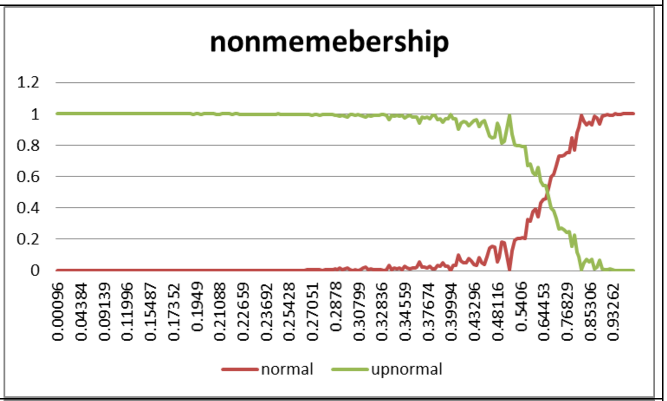
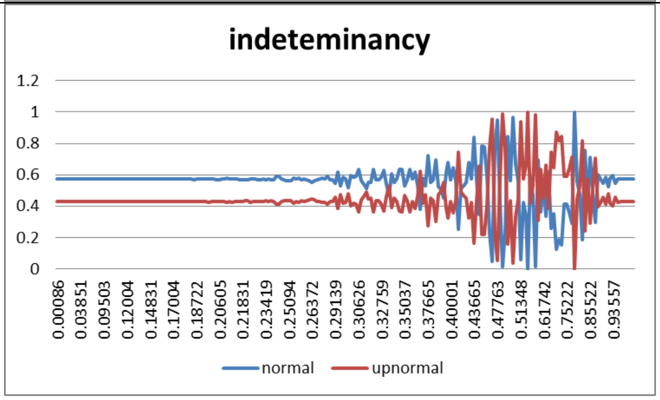
b: Attribute name: dst_bytes

MEMEBERSHIP	
NONMEMBERSHIP	
INDETEMINANCY	

c: Attribute name : Hot

MEMEBERSHIP	
NONMEMBERSHIP	
INDETEMINANCY	

d: Attribute name : Count

MEMEBERSHIP	
NONMEMBERSHIP	
INDETEMINANCY	

e: Attribute name : Srv_serror_rate

Figure 3.10 (a, b, c, d and e). The relation of membership, non-membership and neutrosophic membership

Chapter Summary

This chapter is concerned by the preprocessing phase of the neutrosophic knowledge discovery system. Self-Organized Feature Maps (SOFM) are unsupervised artificial neural networks that were used to build fuzzy MEMEBERSHIP function, hence they could be utilized to define the neutrosophic variable as well, also presented the algorithm and flowchart of the proposed technique. Finally the experimental results showed the KDD'99 data set features in the neutrosophic format (i.e. the MEMBERSHIP, NONMEMBERSHIP AND INDTERMINACY function are calculated and plotted). Having the neutrosophic format for the KDD'99 data set is the first step toward building a classification model for MANET attacks. Producing the neutrosophic IDS basing on the historical data in the KDD'99 data set is the main concern of chapter 4.

Chapter 4

Proposes a MANETs attack inference by a hybrid framework of Self Organized Features Maps (SOFM) and the Genetic Algorithms (GA)

4.1 Introduction

Intrusion Detection System (IDS) is an essential security part for any online network nowadays. An intrusion is "a collection of actions that try to comprehend the privacy, integrity or availability for various resources". Intrusion can likewise be characterized as "a collection of actions imagine to get unapproved assets, abuse rights, cause finish frameworks and systems smashed, diminish running intensity, or refuse any assistance". In this manner, IDS might be a framework to monitor events in PCs or systems and examinations and checking the frameworks uprightness and privacy.

The neutrosophic IDS is formed of two sub phases: the preprocessing stage and the network attacks classification stage. The preprocessing stage is concerned by formulating the network features in a format appropriate for the classification. The KDD network data [13] is reformatted into neutrosophic form $(\mathbf{x}\,, \boldsymbol{\mu}_{\mathbf{A}}(\mathbf{x}), \boldsymbol{\sigma}_{\mathbf{A}}(\mathbf{x}), \boldsymbol{\nu}_{\mathbf{A}}(\mathbf{x}))$ where x is the value of feature, $\boldsymbol{\mu}_{\mathbf{A}}(\mathbf{x})$ is the MEMEBERSHIP (MEM), $\boldsymbol{\sigma}_{\mathbf{A}}(\mathbf{x})$ is the INDETERMINACY (I) and $\boldsymbol{\nu}_{\mathbf{A}}(\mathbf{x})$ is the NONMEMEBERSHIP (NON_MEM) degrees of the x in the feature space.

The Self Organized Features Maps (SOFMs) [14] , machine learning technique, was used to prepare the neutrosophic KDD through learning the MEM and, NON_MEM functions of the KDD network attacks data [13] downloaded from the UCI repository for further processing in knowledge discovery. Then, the MEM and NON_MEM functions are used to calculate the I function according to neutrosophic set definitions. Having the converted neutrosophic KDD dataset which is explained in details in chapter 3, the second phase will be the main interest of this chapter.

The Genetic algorithms (GAs) [33] searching mechanism is utilized in finding a set of neutrosophic (if –then) rules to classify MANETs attacks. The GA initial population is a set of randomly generated individuals. Each individual represents a structure of a neutrosophic (if-then) classification rule. The neutrosophic structure comes from the fact that both the propositions and consequences of the if-then rule are neutrosophic variables. During the GA iterations, the selection, crossover and mutation processes are applied on the populations for generating new fit offsprings. The fitness of the offsprings is quantified by the concept of neutrosophic correlation co-efficient introduced by Salama et al. in [14].

The final population will serve as the neutrosophic rule set for the neutrosophic IDS for the KDD data set. The testing procedure apply new instances of KDD instances (not used during training) to measure the accuracy levels of the obtained neutrosophic IDS. . The comparative study shows that the introduced neutrosophic IDS is competing with other systems found in literature working on the KDD data set. Examples of these techniques are C4.5 [65], Support Vector Machine (SVM) [66], C4.5 +Ant Colony Optimization (ACO) [67], SVM + ACO, EDADT, SVM + Particle Swarm Optimization and C4.5 + PSO algorithms [68].

In light of values got, the precisions of the previous algorithms are 93.23%, 87.18%, 95.06%, 90.82%, 98.12%, 91.57%, 95.37% respectively. The proposed neutrosophic IDS gives an accuracy of %99.3608 with a false alarm rate of 0.089. Compared to IDS in literature, the proposed system gives the highest accuracy with minimal false alarm rates.

4.2 DESIGNING THE PROPOSED NEUTROSOPHIC IDS

IDS could be figured as an arrangement of if-then rules that depict the potential intrusions of the network or systems. Finding the ideal arrangement of these rules is a vital search problem. These algorithms switch the problem in a particular space into a model by utilizing a chromosome-like data structure and develop the chromosomes using selection, recombination, and mutation operators.

GA uses a critical solving strategy and gives ideal solution of the problem. GA works on the Darwinian principle of reproduction. GA uses set of initial individual objects, which each correlating fitness value into new generation of population. Afterwards, it applies crossover and mutation function to generate new offsprings. The algorithm iterates several time to reach the most fit individuals (if-then rules) in the populations.

The proposed hybrid combines SOFM and GA algorithms to produce the neutrosophic rules in two phases. The first phase sets the neutrosophic variables by creating the membership, non-membership and indeterminacy functions for the neutrosophic subsets of the variables. The implementation for the first stage is done by using SOFM from a prior research [15]. The outcome of this stage is passed to GAs [16] along with the training data to first randomly generate initial population, thereafter the neutrosophic correlation coefficient is used as a fitness function to pick out the most fit rules with regard to the training data. Afterwards, the test data is utilized to check for the precision of the rules created.

4.2.1 Formatting Neutrosophic KDD features using SOFM

As declared in chapter 3, in order to build a neutrosophic IDS, the system should be based on neutrosophic variables. The regular features in the KDD data set cannot be used in neutrosophic processing. Hence, reformatting the KDD features into neutrosophic ones is a preprocessing step in the intrusion detection system. Self-Organized Feature Maps (SOFM) are unsupervised artificial neural networks that were used to define the neutrosophic variables [15].

SOFMs capabilities cluster inputs using self-adoption techniques. These capabilities were utilized in generating neutrosophic functions for the subsets of the variables. The SOFM are used to define the membership, non-membership and indeterminacy functions for the KDD data set features. The algorithm for generating the neutrosophic features definitions is cleared in section 3.1.1.

These definitions are used during the GA search in fitness calculation (the neutrosophic correlation co-efficient). The procedure of generating the neutrosophic IDS classification rules is introduced in the next section.

4.2.2 Creating neutrosophic rules utilizing Genetic Algorithms

The neutrosophic knowledge based system is composed of a set of neutrosophic rules (Figure 4.1). This procedure is in charge of designing the neutrosophic conditional rules via applying the processing power of GAs at random initial population. After that, utilizing the neutrosophic correlation co-efficient as a fitness function, it picks the most proper performers from the population. At the last stage, the population becomes the set of neutrosophic rules required for the neutrosophic inference engine of the IDS.

$$\mathbf{R1}\quad \overset{S_1,}{\underbrace{(\mu_A(x),\sigma_A(x),\nu_A(x))}_{A_1}}\quad \overset{S_2,}{\underbrace{(\mu_A(x),\sigma_A(x),\nu_A(x))}_{A_2}}\quad \overset{S_3,}{\underbrace{(\mu_A(x),\sigma_A(x),\nu_A(x))}_{A_3}}\quad \ldots\ldots\quad \overset{d_1,\ MIN_i}{\underbrace{(\mu_A(x),\sigma_A(x),\nu_A(x))}}$$

$$\mathbf{R2}\quad \overset{S_1}{\underbrace{(\mu_A(x),\sigma_A(x),\nu_A(x))}_{A_1}}\quad \overset{S_2}{\underbrace{(\mu_A(x),\sigma_A(x),\nu_A(x))}_{A_2}}\quad \overset{S_3}{\underbrace{(\mu_A(x),\sigma_A(x),\nu_A(x))}_{A_3}}\quad \ldots\quad \ldots\ldots\quad \overset{d_2,\ MIN_i}{\underbrace{(\mu_A(x),\sigma_A(x),\nu_A(x))}}$$

Figure 4.1. Neutrosophic rules of the knowledge based

An individual represents the possible solution or the possible neutrosophic rule which is composed of a set of conditional propositions (neutrosophic attributes) and the consequence decision attribute. Each neutrosophic attribute and the decision attribute will occupy one gene within the individual.

To demonstrate the neutrosophic format, each gene will be represented by $\boldsymbol{A} \in \mathrm{S}$, $(\boldsymbol{\mu}_{\mathbf{A}}(\mathbf{x}), \boldsymbol{\sigma}_{\mathbf{A}}(\mathbf{x}), \boldsymbol{\nu}_{\mathbf{A}}(\mathbf{x}))$ where A is the neutrosophic feature that belongs to the subset S with degrees of membership $\boldsymbol{\mu}_{\mathbf{A}}(\mathbf{x})$, non- membership $\boldsymbol{\nu}_{\mathbf{A}}(\mathbf{x})$ and indeterminacy $\boldsymbol{\sigma}_{\mathbf{A}}(\mathbf{x})$. The GA individual is presented in Figure 4.2.

$\boldsymbol{A}_{\mathbf{1}} \in \mathrm{S}_2$, $(\boldsymbol{\mu}_{\mathbf{A}}(\mathbf{x}), \boldsymbol{\sigma}_{\mathbf{A}}(\mathbf{x}), \boldsymbol{\nu}_{\mathbf{A}}(\mathbf{x}))$	$\boldsymbol{A}_{\mathbf{2}} \in \mathrm{S}_3$, $(\boldsymbol{\mu}_{\mathbf{A}}(\mathbf{x}). \boldsymbol{\sigma}_{\mathbf{A}}(\mathbf{x}). \boldsymbol{\nu}_{\mathbf{A}}(\mathbf{x}))$	$\boldsymbol{A}_{\mathbf{3}} \in \mathrm{S}_2$, $(\boldsymbol{\mu}_{\mathbf{A}}(\mathbf{x}), \boldsymbol{\sigma}_{\mathbf{A}}(\mathbf{x}), \boldsymbol{\nu}_{\mathbf{A}}(\mathbf{x}))$		$\boldsymbol{A}_{\mathbf{5}} \in \mathrm{S}_1$, $(\boldsymbol{\mu}_{\mathbf{A}}(\mathbf{x}), \boldsymbol{\sigma}_{\mathbf{A}}(\mathbf{x}), \boldsymbol{\nu}_{\mathbf{A}}(\mathbf{x}))$	$\boldsymbol{d}_{\mathbf{1}} \in$ S_3

$$\textbf{If} \; (A_1 \in s_2 \wedge A_2 \in s_3 \wedge A_3 \in s_2 \wedge A_5 \in s_1) \; \textbf{then} \; d_1 \in s_3$$

Figure 4.2: The GA individual neutrosophic rules of a system has 5 feature and a dependent class

Note that A1, A2, A3, A5 are the neutrosophic attributes and s2, s3, s2, s1 are the neutrosophic subsets of the attributes. ^ is the logical and operator. Moreover, the previous neutrosophic conditional rule does not rely on A4 as a proposition and produces d1 as a decision within the neutrosophic subset s3.

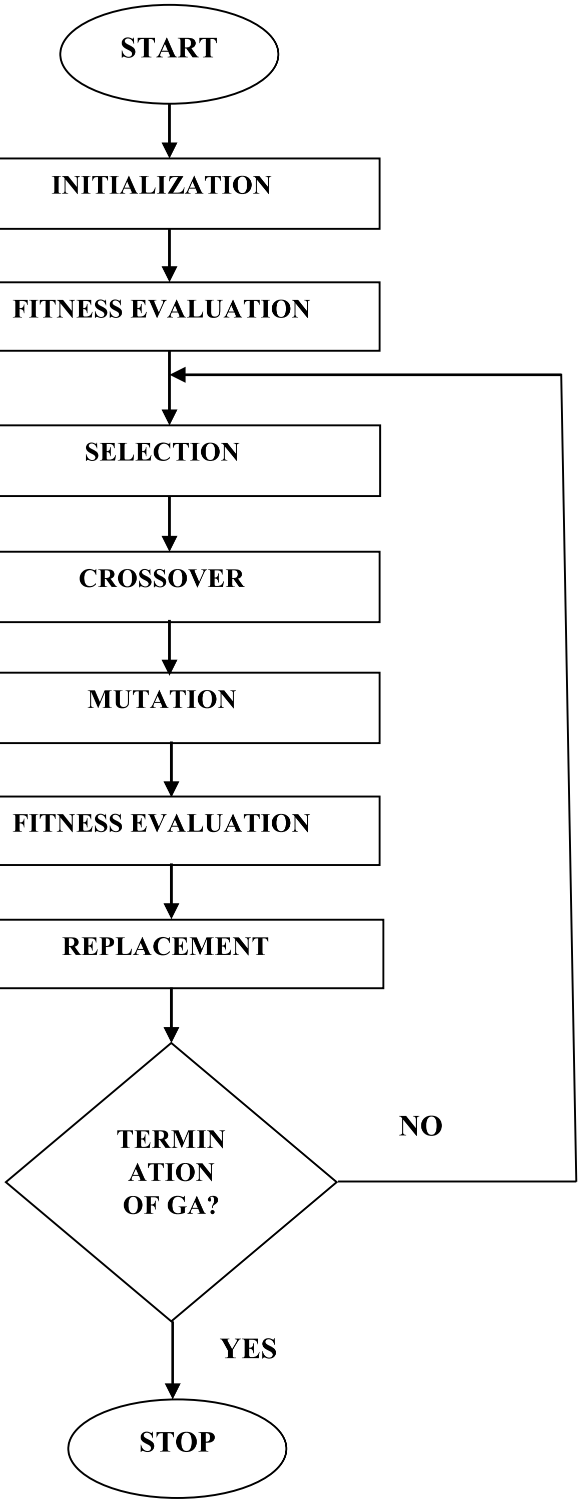


Figure 4.3: Genetic Algorithm Process

The flow chart for the generating neutrosophic rules procedure is provided in figure 4.3. This procedure is composed of **7** phases with an iterated loop until the predefined number of iterations is performed.

The first phase generates a collection of random neutrosophic rules according to the previous format (figure 4.2) which forms the initial population.

The second and third phases are the fitness evaluation and the selection process that determines the most fit individuals in the pool according to the neutrosophic correlation coefficient as a fitness function.

The fourth phase is the crossover which produces new offspring from the fit individuals selected during the selection phase.

The fifth phase is the mutation which switches one of the genes randomly to help increase the performance but this is accomplished under very rare circumstances.

The sixth and seventh phases recalculate the fitness of the new offspring's and replace the children in place of their parents. These phases are repeated for a number of iterations from the third to the seventh.

After completion, the final fit generation will form the set of neutrosophic conditional rules for the neutrosophic IDS. The GA procedure is straight forward and mostly used in the same way for most of classification applications. The main module that differentiate one application from the other is the fitness function calculation.

In order to integrate the neutrosophic concepts within the GA, the neutrosophic correlation co-efficient is used to find the rules with the highest dependency between the neutrosophic conditional features and the decision attribute. The co-efficient equations are illustrated in the next section.

4.3 Fitness function

The neutrosophic correlation coefficient is used to measure the fit rules. The neutrosophic correlation coefficient measures the degree of relation between propositions and the decision attribute. The generated neutrosophic rules that maximizes this correlation will be the most fit rules in the population and will be selected and passed to the next generation during the GA process. Some operations on neutrosophic sets introduced and studied by Salama et al in 2012 [14]. For S and Y are two neutrosophic sets in a finite space x={x1, x2,……….., xn } , the correlation of neutrosophic sets S and Y is defined as follows:

$$\boldsymbol{C}\ (\boldsymbol{S},\boldsymbol{Y}) = \sum_{i=1}^{n} \left\lfloor \left(\mu_S(x_i).\, \mu_Y(x_i) + \sigma_S(x_i).\ \sigma_Y(x_i) + \nu_S(x_i).\, \nu_Y(x_i)\right) \right\rfloor \quad \text{(Eq.1)}$$

and the correlation coefficient of S and Y given by

$$\boldsymbol{R}(\boldsymbol{S},\boldsymbol{Y}) = \frac{\boldsymbol{C}(\boldsymbol{S},\boldsymbol{Y})}{(\boldsymbol{T}(\boldsymbol{S}).\boldsymbol{T}(\boldsymbol{Y}))½} \qquad \text{(Eq.2)}$$

where $\boldsymbol{T}\ (\boldsymbol{S}) = \sum_{i=1}^{n} \left\lfloor (\mu^2{}_S(x_i) + \sigma^2{}_S(x_i) + \nu^2{}_S(x_i)) \right\rfloor$, $\boldsymbol{T}\ (\boldsymbol{Y}) = \sum_{i=1}^{n} \left\lfloor (\mu^2{}_Y(x_i) + \sigma^2{}_Y(x_i) + \nu^2{}_Y(x_i)) \right\rfloor$; $|\boldsymbol{R}(\boldsymbol{S},\boldsymbol{Y})| \leq \boldsymbol{1}$

‘S’ refers to the conditional propositions in a neutrosophic if-then rule like srv_serror_rate type, the Srv_serror_rate type, and the flag [15] where ‘Y’ refers to the class attributes like there is an attack or not.

4.4 EXPERMINTAL RESULTS

The proposed hybrid aims to identify the attacks that take place in the network to classify them correctly and increase the detection rate of attacks, figure 4.4 indicate the whole system. The hybrid consists of a preprocessing step and two major phases.

The preprocessing step utilizes the Waikato Environment for Knowledge Analysis (WEKA) [69] data mining tool to get the most important attributes from the KDD-99 data set.

The first phase is the neutrosophic variables definition which converts the normal data into neutrosophic variables utilizing the Self Organized features maps (SOFM) [15] declared in chapter 3. The neutrosophic definition of the variables along with the training KDD-99 data are fed into the classification process.

The second phase is the neutrosophic IDS building. The proposed system utilizes the evolutionary capabilities of The Genetic Algorithms (GA) to find the appropriate classification (if-then) rules.

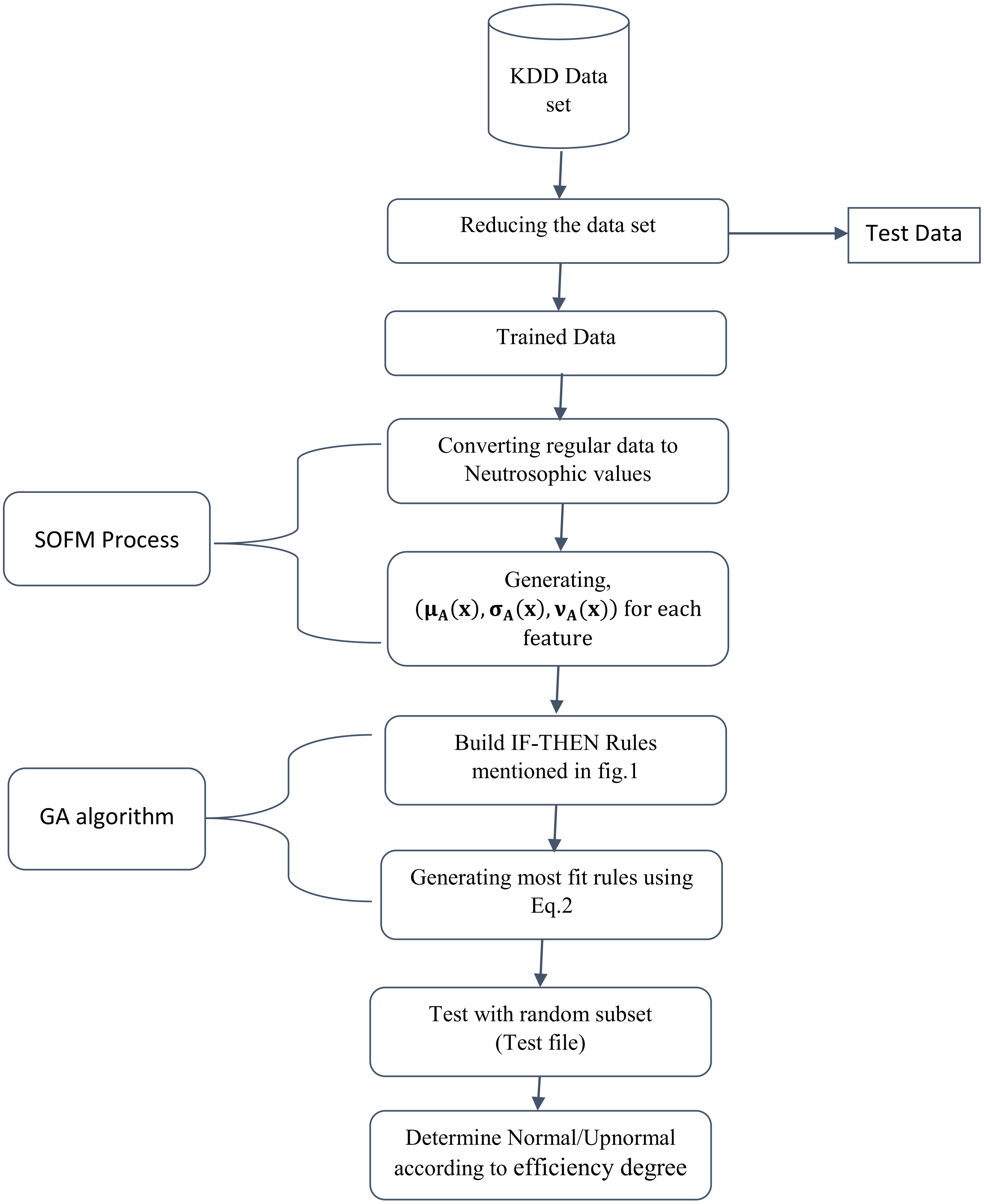
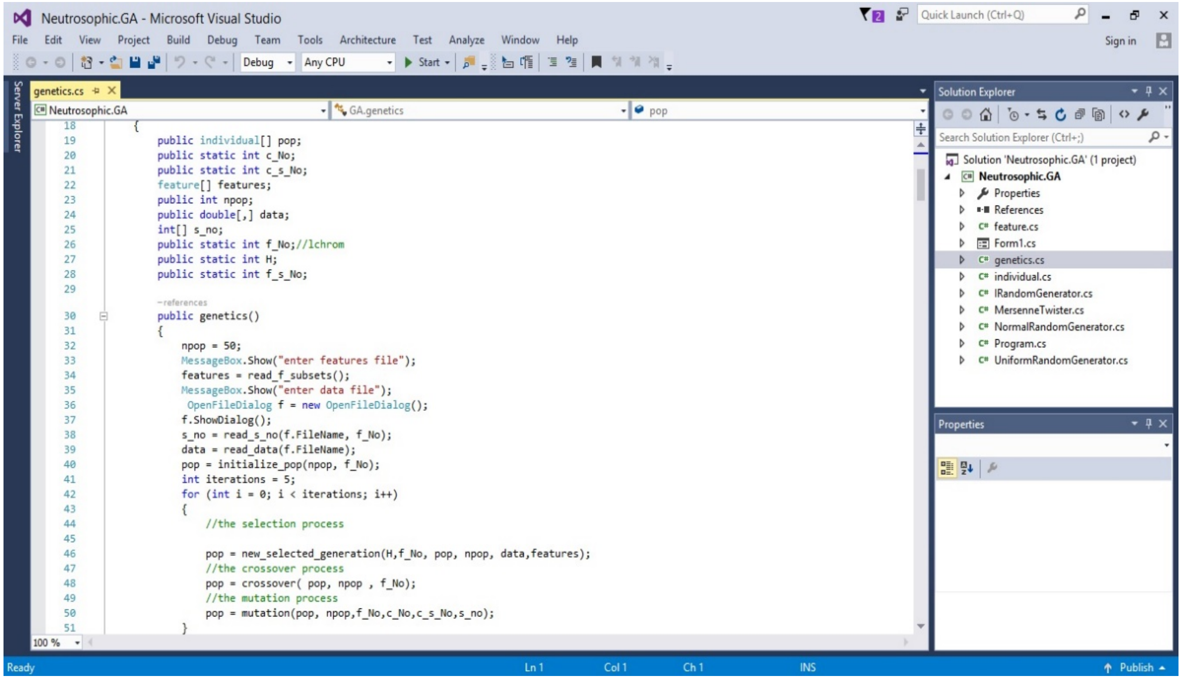

KDD Data set
Reducing the data set
Test Data
Trained Data
Converting regular data to Neutrosophic values
SOFM Process
Generating, $(\mu_A(x), \sigma_A(x), \nu_A(x))$ for each feature
Build IF-THEN Rules mentioned in fig.1
GA algorithm
Generating most fit rules using Eq.2
Test with random subset (Test file)
Determine Normal/Upnormal according to efficiency degree

Figure 4.4: A frame work design for the whole system

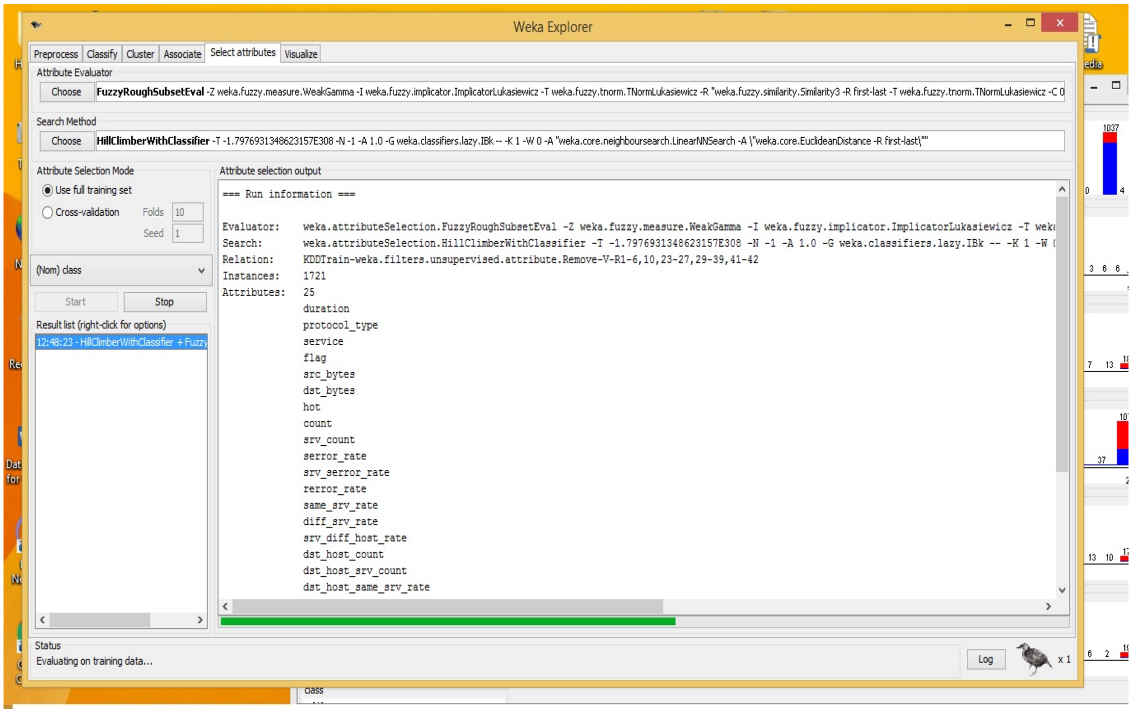
Simulation of the proposed system is implemented by C# figure 4.5 environment on Dell Inspiron 15.6" Laptop - Intel Core i5, Memory (RAM): 8.00 GB, System type: 64-bit operating system and Windows edition: windows 10.



```
    {
        public individual[] pop;
        public static int c_No;
        public static int c_s_No;
        feature[] features;
        public int npop;
        public double[,] data;
        int[] s_no;
        public static int f_No;//lchrom
        public static int H;
        public static int f_s_No;

        -references
        public genetics()
        {
            npop = 50;
            MessageBox.Show("enter features file");
            features = read_f_subsets();
            MessageBox.Show("enter data file");
             OpenFileDialog f = new OpenFileDialog();
            f.ShowDialog();
            s_no = read_s_no(f.FileName, f_No);
            data = read_data(f.FileName);
            pop = initialize_pop(npop, f_No);
            int iterations = 5;
            for (int i = 0; i < iterations; i++)
            {
                //the selection process

                pop = new_selected_generation(H,f_No, pop, npop, data,features);
                //the crossover process
                pop = crossover( pop, npop , f_No);
                //the mutation process
                pop = mutation(pop, npop,f_No,c_No,c_s_No,s_no);
            }
```

Figure 4.5. The proposed system is implemented by C#.

The original data set is composed of 42 attributes; hence a reduction preprocessing step is required. The preprocess stage is implemented using (WEKA) [69] figure 4.6. The reduction algorithm used is the Attribute Evaluator ‘FuzzyRoughSubsetEval’ and search method ‘HillClimberWithClassifier’. After the reduction process, KDD-99 data file contain 25 features and 1721 instance as in figure 4.7 which red color abnormal (attacks) and the blue is normal.



Attribute Evaluator

FuzzyRoughSubsetEval -Z weka.fuzzy.measure.WeakGamma -I weka.fuzzy.implicator.ImplicatorLukasiewicz -T weka.fuzzy.tnorm.TNormLukasiewicz -R "weka.fuzzy.similarity.Similarity3 -R first-last -T weka.fuzzy.tnorm.TNormLukasiewicz -C 0

Search Method

HillClimberWithClassifier -T -1.7976931348623157E308 -N -1 -A 1.0 -G weka.classifiers.lazy.IBk -- -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

Attribute Selection Mode

● Use full training set

○ Cross-validation Folds 10 Seed 1

(Nom) class

Result list (right-click for options)

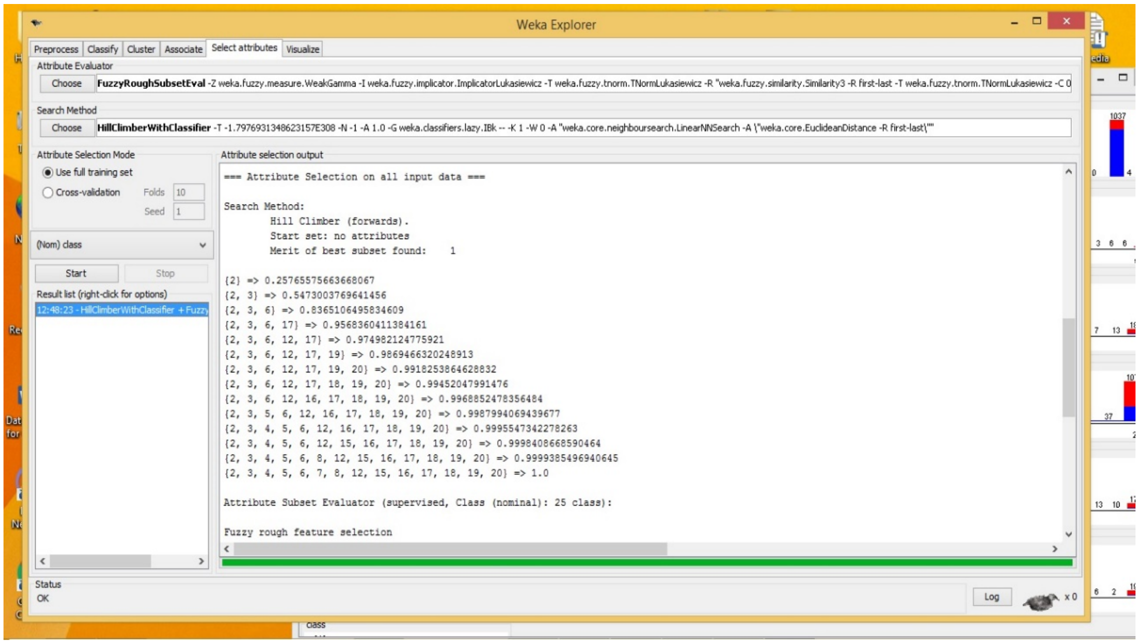
12:48:23 - HillClimberWithClassifier + Fuzzy

Attribute selection output

```
=== Run information ===

Evaluator:    weka.attributeSelection.FuzzyRoughSubsetEval -Z weka.fuzzy.measure.WeakGamma -I weka.fuzzy.implicator.ImplicatorLukasiewicz -T weka
Search:       weka.attributeSelection.HillClimberWithClassifier -T -1.7976931348623157E308 -N -1 -A 1.0 -G weka.classifiers.lazy.IBk -- -K 1 -W (
Relation:     KDDTrain-weka.filters.unsupervised.attribute.Remove-V-R1-6,10,23-27,29-39,41-42
Instances:    1721
Attributes:   25
              duration
              protocol_type
              service
              flag
              src_bytes
              dst_bytes
              hot
              count
              srv_count
              serror_rate
              srv_serror_rate
              rerror_rate
              same_srv_rate
              diff_srv_rate
              srv_diff_host_rate
              dst_host_count
              dst_host_srv_count
              dst_host_same_srv_rate
```

Status

Evaluating on training data...



Attribute Evaluator

FuzzyRoughSubsetEval -Z weka.fuzzy.measure.WeakGamma -I weka.fuzzy.implicator.ImplicatorLukasiewicz -T weka.fuzzy.tnorm.TNormLukasiewicz -R "weka.fuzzy.similarity.Similarity3 -R first-last -T weka.fuzzy.tnorm.TNormLukasiewicz -C 0

Search Method

HillClimberWithClassifier -T -1.7976931348623157E308 -N -1 -A 1.0 -G weka.classifiers.lazy.IBk -- -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

Attribute Selection Mode

● Use full training set

○ Cross-validation Folds 10 Seed 1

(Nom) class

Result list (right-click for options)

12:48:23 - HillClimberWithClassifier + Fuzzy

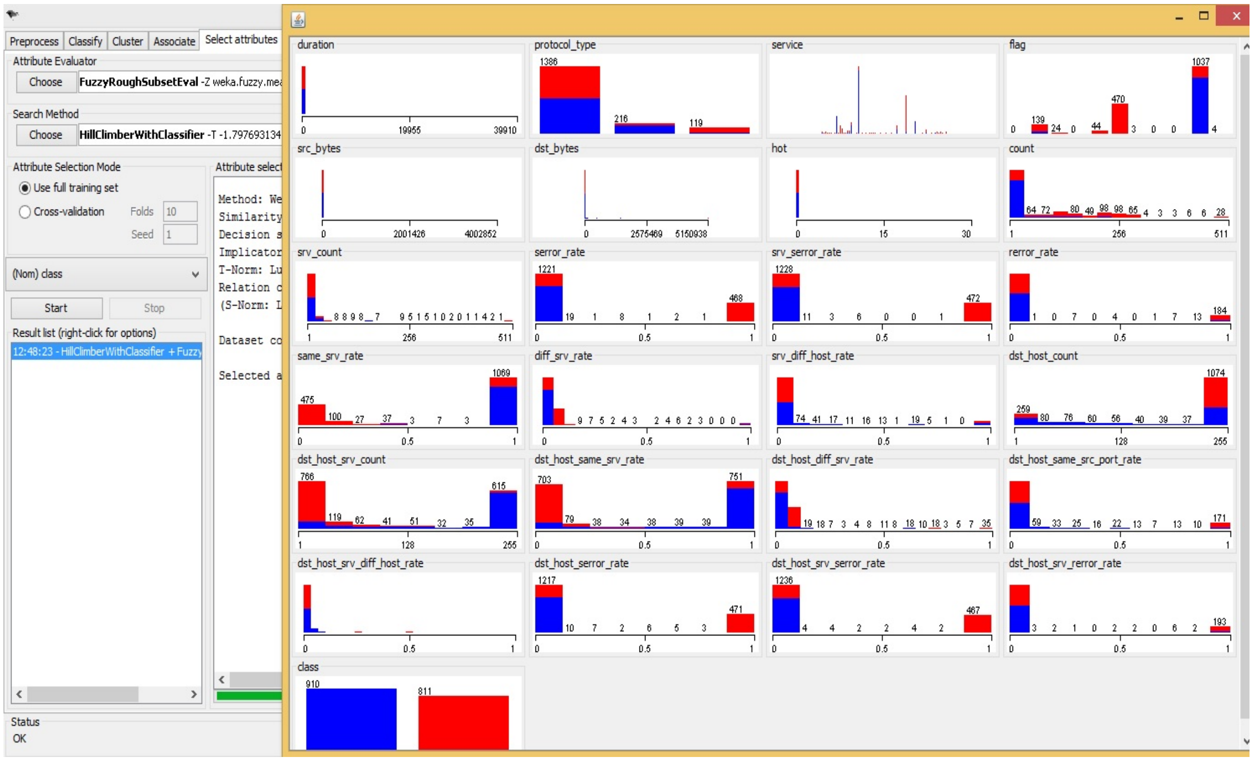
Attribute selection output

```
=== Attribute Selection on all input data ===

Search Method:
	Hill Climber (forwards).
	Start set: no attributes
	Merit of best subset found:    1

{2} => 0.25765575663668067
{2, 3} => 0.5473003769641456
{2, 3, 6} => 0.8365106495834609
{2, 3, 6, 17} => 0.9568360411384161
{2, 3, 6, 12, 17} => 0.974982124775921
{2, 3, 6, 12, 17, 19} => 0.9869466320248913
{2, 3, 6, 12, 17, 19, 20} => 0.9918253864628832
{2, 3, 6, 12, 17, 18, 19, 20} => 0.99452047991476
{2, 3, 6, 12, 16, 17, 18, 19, 20} => 0.9968852478356484
{2, 3, 5, 6, 12, 16, 17, 18, 19, 20} => 0.9987994069439677
{2, 3, 4, 5, 6, 12, 16, 17, 18, 19, 20} => 0.9995547342278263
{2, 3, 4, 5, 6, 12, 15, 16, 17, 18, 19, 20} => 0.9998408668590464
{2, 3, 4, 5, 6, 8, 12, 15, 16, 17, 18, 19, 20} => 0.9999385496940645
{2, 3, 4, 5, 6, 7, 8, 12, 15, 16, 17, 18, 19, 20} => 1.0

Attribute Subset Evaluator (supervised, Class (nominal): 25 class):

Fuzzy rough feature selection
```

Status

OK

Figure 4.6. A reduction preprocessing step using WEKA.



Attribute Evaluator

FuzzyRoughSubsetEval -Z weka.fuzzy.mea

Search Method

HillClimberWithClassifier -T -1.797693134

Attribute Selection Mode

● Use full training set

○ Cross-validation Folds 10 Seed 1

(Nom) class

Result list (right-click for options)

12:48:23 - HillClimberWithClassifier + Fuzzy

Attribute select

```
Method: We
Similarity
Decision s
Implicator
T-Norm: Lu
Relation c
(S-Norm: L

Dataset co

Selected a
```

Status

OK

Figure 4.7: 25 features red color abnormal (attacks) and the blue is normal.

Through the neutrosophic variable definition phase, The SOFM is applied to the KDD-99 data set to define the MEMEBERSHIP, INDETERMINACY and NONMEMEBERSHIP functions. The technique used for neutrosophic variables definition is illustrated in our previous work in [15] declared also in chapter 3. Figure 3.10 in chapter 3 shows the result of the neutrosophic features definition phase.

During the second phase, The IDS classification pattern which detects threats in the MANET network is implemented by an artificial intelligent algorithm (GA). Each feature from the KDD-99 data set will have three different definitions for MEMEBERSHIP, INDETERMINACY and NONMEMEBERSHIP assumption for the variable values.

These features along with a random sub set of KDD-99 (training data) are passed to GA program to build the set of if-then rules which represent the neutrosophic IDS. The generated output file contains the most fit rules according the steps in GA pseudo code. The fitness function of the GA is the neutrosophic correlation coefficient which is calculated according to Eq. 2.

The GA program simulation is implemented in 3 dimensions “MEMEBERSHIP, INDETERMINACY and NONMEMEBERSHIP” compared with other techniques or algorithms used only two dimension “MEMEBERSHIP and NONMEMEBERSHIP”. The genetic algorithm parameters like cross over rate, mutation rate, number of population and number of iterations are assigned to the values illustrated in table 2.

Table 2: GA parameters in experiments

cross over rate	0.6
mutation rate	0.90
number of population	500
number of iterations	50

At the end of the GA iterations, the final file will contain the most fit (if-then) rules. Each one will have MEMBERSHIP, INDETERMENANCY AND NONMEMEBERSHIP values for each feature. These rules will be the inference engine for the neutrosophic IDS.

The inference methodology used mimics the (min-max) mamdani inference methodology [70]. The output rules file generated have the most appropriate neutrosophic rules which indicate whether a given instance is a Normal connection or an attack. On applying a new instances (network data) to the system, the program select the MIN $(\boldsymbol{\mu}_{\mathbf{A}}(\mathbf{x}), \boldsymbol{\sigma}_{\mathbf{A}}(\mathbf{x}), \boldsymbol{\nu}_{\mathbf{A}}(\mathbf{x}))$ value from all features in each rule in assumption that all feature are interconnected by 'AND' gate. Also, assuming that all rules are connected by 'OR' gate, the program select the rule with MAX $(\boldsymbol{\mu}_{\mathbf{A}}(\mathbf{x}), \boldsymbol{\sigma}_{\mathbf{A}}(\mathbf{x}), \boldsymbol{\nu}_{\mathbf{A}}(\mathbf{x}))$ value to be the matched one. Then, the result of that matched rule will be compared to the actual KDD-99 data set to calculate the number of accurate instances percentage and the false rate percentage.

During experiments, the KDD99 data set is divided randomly into two equal data sets (training and testing). The neutrosophic IDS is built using the training data set, then its accuracy is measured by the new instances in the test data set.

The neutrosophic IDS reached an average accuracy 99.3608% which indicates that the proposed technique is more accurate than the previous algorithms used in this area [71, 72 and 73], this appears in the table 3, the figures 4.8 and 4.9 below. The results shown below in Table 3 represents the accuracy, sensitivity and specificity values for the proposed neutrosophic genetic algorithm against C4.5, SVM, C4.5 +ACO, SVM + ACO, EDADT , SVM + PSO and C4.5 + PSO algorithms [68].

In light of values got, the precision of C4.5 is 93.23%, the precision of SVM is 87.18%, the precision of C4.5 + ACO is 95.06%, the precision of SVM + ACO is 90.82%, the precision of C4.5 + PSO is 95.37%, the precision of SVM + PSO is 91.57% and the accuracy of Improved EDADT is 98.12%.

It is obvious that the neutrosophic IDS generated by GA takes highest precision percentage when compared to all seven classification based algorithms. Figure 4.8 indicate the corresponding chart for the result obtained in Table 3. Figure 4.9 shows the performance of existing and proposed neutrosophic Intrusion Detection system (IDs) algorithm based on false alarm rate (FAR). Thus the proposed neutrosophic Intrusion Detection System (IDS) Algorithm effectively detects attack with less false alarm rate.

Table3: Performance of proposed neutrosophic genetic algorithm vs. existing algorithms

Algorithms	Accuracy (%)	FAR (%)
C4.5	93.23	1.65
SVM	87.18	3.2
C4.5+ACO	95.06	0.87
SVM+ACO	90.82	2.42
C4.5+PSO	95.37	0.72
SVM+PSO	91.57	1.94
EDADT	98.12	0.18
proposed neutrosophic genetic algorithm	99.3608	0.089

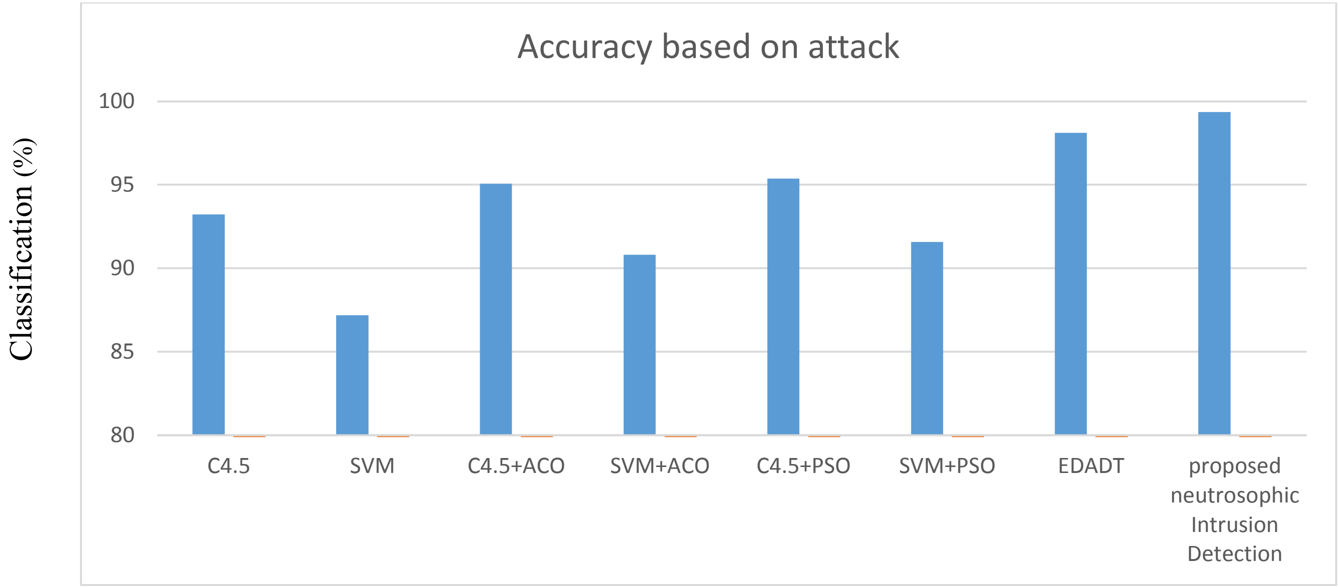


Figure 4.8: Results of neutrosophic genetic algorithm vs. existing algorithms

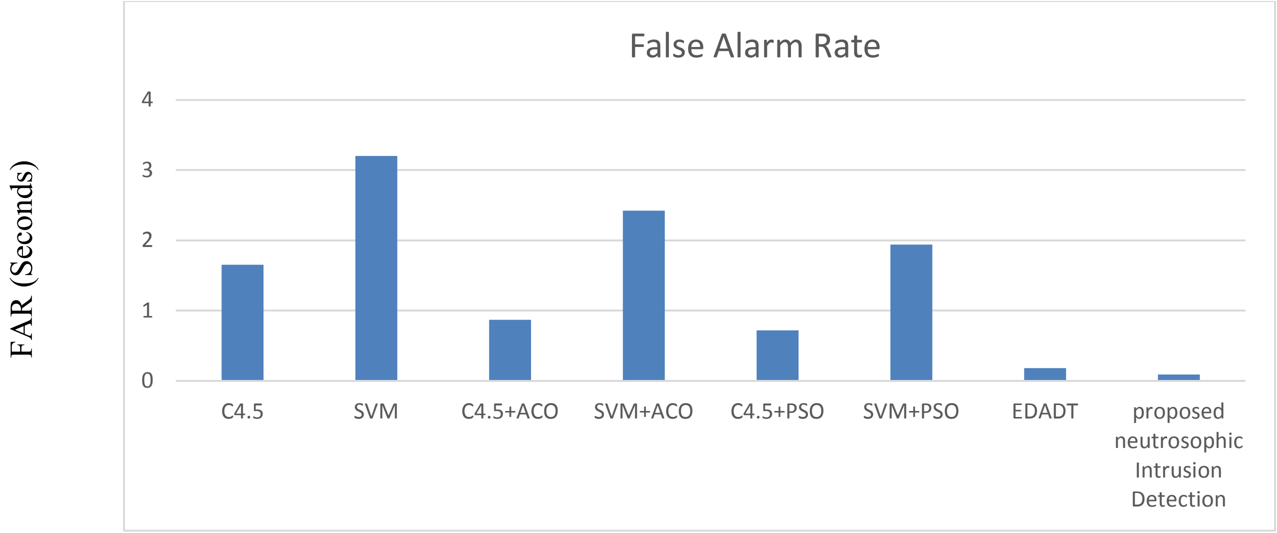


Figure 4.9. False alarm rate of proposed neutrosophic genetic algorithm vs. existing algorithms.

4.5 A Comparative study

Recently, Sen and Clark [74] have introduced a survey regarding existing intrusion detection approaches for MANETs. Traditional anomaly-based IDSs use predefined ‘‘normality’’ models to discover anomalies within the network. This approach simply cannot be deployed in MANETs for the following reasons:

- The flexibility of MANET nodes, makes building a definition of ‘‘normal’’ and ‘‘malicious’’ behaviour very hard and challenging.
- The mobility of nodes leads to changes of the network topology, increasing the complexity of the detection method.
- Since the MANET nodes haven't any fixed location, there's no central management and/or monitoring point wherever an IDS might be placed. This means that the detection method could also be distributed into many nodes, as well as the collection and analysis of data.

Consequently, IDS are categorized into cooperative or independent (non-collaborative) [74].

Independent IDs consist of IDs agents settled in the nodes of the network and be accountable for observing all nodes inside the network and sending alarms whenever they detect any suspect activity. This design has a number of drawbacks such as:

- Determining the place of the IDs agents, since nodes are moveable, and several domains of the network might not be monitored (for example, if the node hosting an IDs agent of one domain moves to another, the first remains uncovered).

- Some resources such as bandwidth, central processing unit and/or power are scarce in these environments. Therefore, nodes hosting the IDs agents ought to be those having more resources and moreover, a bigger transmission vary. Maximizing the detection rate subject to resource limitation is a nondeterministic polynomial time (NP) complete problem and a few algorithms are planned to approximate the solution [75].

Several IDs architectures have been proposed to be used in mobile networks. The most updated IDs in the last three years ago are summarized.

First, Md Nasir Sulaiman, in 2015 [66] proposed a new classifier to enhance the abnormal attacks detection rate based on support vector machine (SVM) and genetic programming (GP). Depending on the experimental results, GPSVM classifier is controlled to earn higher detection rate on the scarce abnormal attacks, without vital reduction on the general accuracy. This could be because that, GPSVM optimization mission confirms the accuracy balancing between classes without reducing the generalization property of SVM. GPSVM has an average accuracy of 88.51%.

Second, Shankar Sriram V S in 2017 [76] presented an adaptive, and a strong IDs technique using Hypergraph based Genetic algorithm (HG - GA) for parameter setting and feature selection in Support Vector Machine (SVM). Hyper – clique property of hypergraph was exploited for the generation of initial population to fasten the search for the optimum answer and to stop the trap at the local minima.

HG-GA uses a weighted objective function to keep up the trade-off between increasing the detection rate and minimizing the false alarm rate, in conjunction with the optimum range of features. The performance of HG-GA SVM was evaluated using NSL-KDD intrusion dataset under two situations (i) All features and (ii) informative features obtained from HG – GA, with the Accuracy rate is 97.14% and the false rate is 0.83%.

Third, Muder Almi'ani in 2018 [77] designed an intelligent IDs using clustered version of Self-Organized Map (SOM) network. The planned system consists of two subsequent stages. SOM network was designed, then a hierarchical agglomerative clustering using k-means was applied on SOM neurons. The proposed work in this research addressed the issues of sensitivity and time consumption for every connection record process. The proposed system was demonstrated using NSL-KDD benchmark dataset, wherever it's achieved superior sensitivity reached up to 96.66% you uninterested in less than 0.08% milliseconds per connection record.

The researches in 2017 and 2018 [15 and 16] tend to plan a novel approach for classifying MANETs attacks with a neutrosophic intelligent system based on genetic algorithm. Neutrosophic system could be a discipline that produces a mathematical formulation for the indeterminacy found in such complex situations. Neutrosophic rules are computed with symbols rather than numeric values creating a good base for symbolic reasoning. These symbols ought to be carefully designed as they form the propositions base for the neutrosophic rules (NR) in the IDs. Every attack is set by MEM, NON_MEM and I degrees in neutrosophic system. The research proposed a MANETs attack inference by a hybrid framework of Self Organized features Maps (SOFM) and the Genetic Algorithms (GA). The hybrid uses the unsupervised learning capabilities of the SOFM to reformat the MANETs neutrosophic conditional variables.

The neutrosophic variables along with the training information set are fed into the GA to find the most match neutrosophic rule set from a number of initial sub attacks according to the neutrosophic correlation coefficient as a fitness function. This technique is designed to discover unknown attacks in MANETs.

The simulation and experimental results are conducted on the KDD-99 network attacks data available in the UCI machine-learning repository for further process in knowledge discovery. The experiments cleared the feasibility of the proposed hybrid by an average accuracy of 99.3608 and false rate is 0.089.

It is clear that the neutrosophic IDs generated by GA takes highest precision percentage in comparison to all three classification based algorithms. Figure 4.10 refered to the corresponding chart for the result obtained in Table 4. Figure 4.11 shows the performance of existing and proposed neutrosophic IDs algorithm based on false alarm rate (FAR). Therefore our proposed neutrosophic IDs Algorithm [15 and 16] effectively detects attack with less false alarm rate.

Table 4: Performance of Neutrosophic genetic algorithm vs. other existing algorithms

System name	Accuracy%	false rate%
GPSVM	88.51	0.76
HG-GA SVM	97.14	0.83
Clustered SOM	96.66	0.08
neutrosophic intelligent system based on genetic algorithm	99.3608	0.089

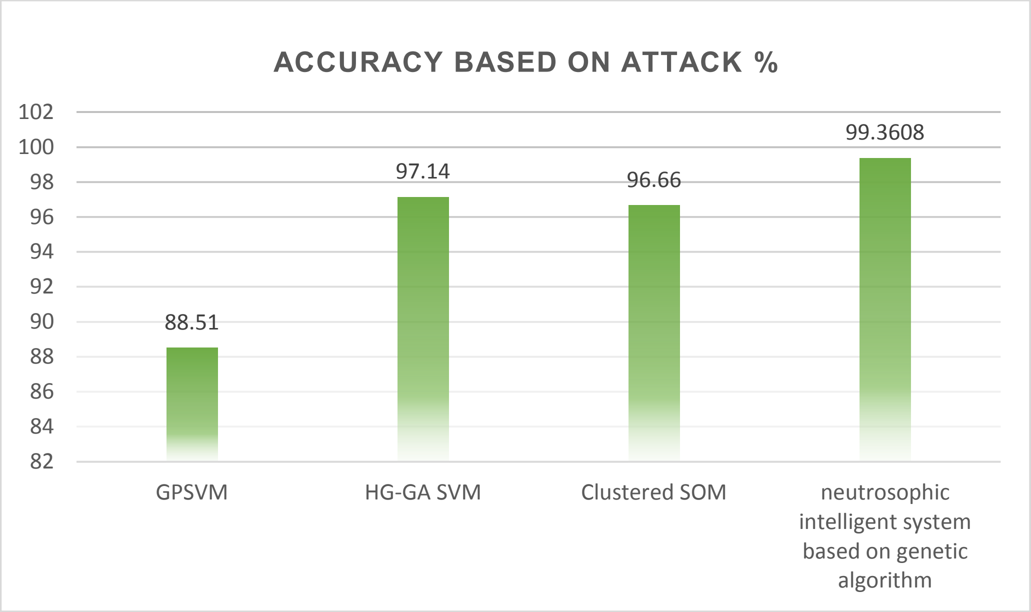


Figure 4.10: Results of neutrosophic genetic algorithm vs. existing algorithms

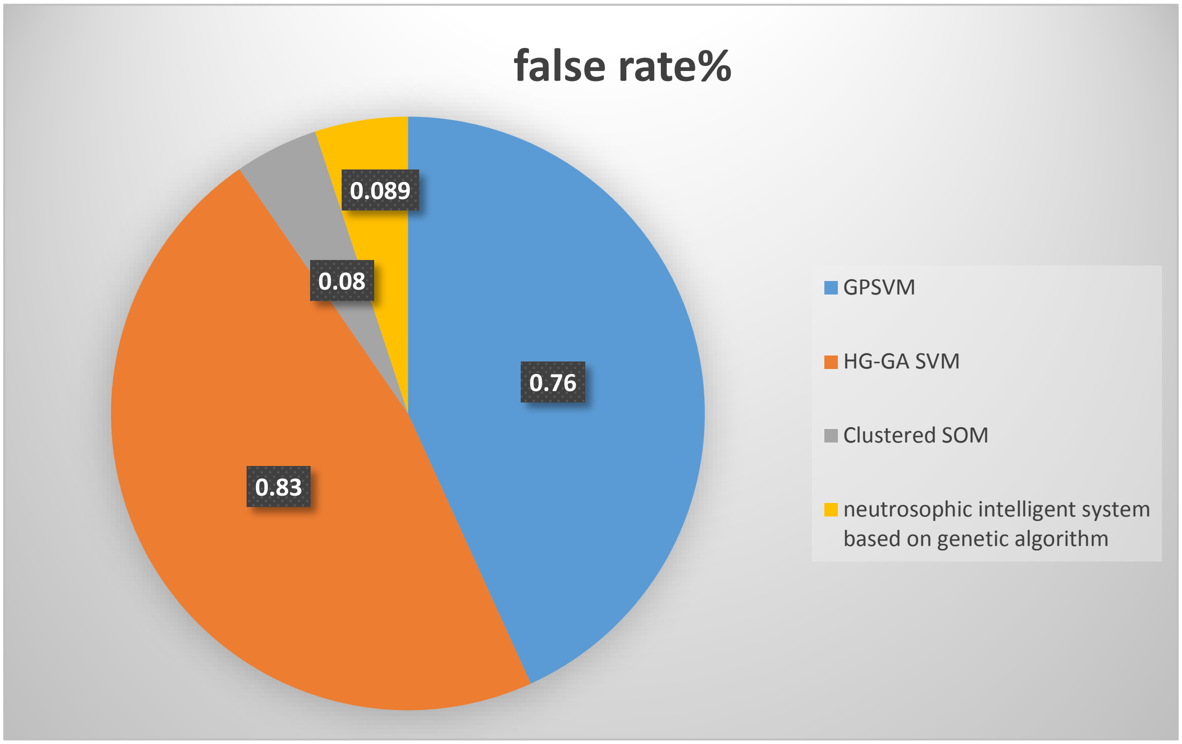


Figure 4.11. False alarm rate of proposed neutrosophic genetic algorithm vs. existing algorithms.

4.6 Chapter Summary

This chapter is concerned by the process of designing and building a neutrosophic IDS basing on the neutrosophic KDD network data set features converted in the chapter 3. The neutrosophic IDS is implemented via the searching capabilities of the GA. The process takes as an input the neutrosophic definitions of the network features along with the training data set. The GA generates an initial set of neutrosophic if-then rules and begins to match the rule set to the training data. The most fit rules are chosen upon the neutrosophic correlation coefficient. The GA continue throw the iterations applying the crossover, mutation and selection processes. At the end of the iterations, the final generation of the neutrosophic if-then rules will be the core of the neutrosophic IDS. The experiments show the proposed system accuracy level and false alarm rates using the test data set. The next chapter will present the conclusion and future work of the proposed thesis.

Chapter 5

Conclusions & Future work

5.1 Conclusions

In this chapter the analysis for the studies achieved during this research is provided as well as the conclusions from the experimental results and future work.

In this research the MANET is introduced as one of the vital kind of networks because of its several advantages including the disappearance of the regular settled infrastructure and the depressed cost requirements to work in such networks.

The nature of the Mobile Adhoc Networks (MANETs) puts it under attacks from inside and outside the network. Thus, it desperately requires security plans and techniques to remain against these attacks.

The security has two mechanisms, first prevent mechanism like cryptography and hash function, second reactive mechanism like intrusion detection system. Such systems work in an environment are permanently changing and are full of indeterminacy and uncertainty. Hence, the traditional techniques for finding the network attacks are not suitable. Neutrosophic theory handles every data object from three points of view which are the truth, falsity and indeterminacy degrees. The thesis merges the neutrosophic concepts within the classification if-then rules to produce a neutrosophic IDS which is responsible for recognizing attacks in the MANET network data. In building the neutrosophic classification model, the system depends mainly on neutrosophic variables to detect threats in MANETs.

The KDD'99 network data found on the UCI machine learning repository are numeric and traditional sets. Therefore, a mechanism for reformatting the traditional feature into neutrosophic one is required. The reformatting process is simply determining the MEMBERSHIP, NONMEMBERSHIP AND INDTERMINACY functions of each neutrosophic set defined over the network data feature.

This is the responsibility of the preprocessing phase of the neutrosophic knowledge discovery system. Self-Organized Feature Maps (SOFM) are unsupervised artificial neural networks that were used to build fuzzy MEMEBERSHIP function basing on the clustering capability of the algorithm and the some few predetermined data from experts in the field. Hence they could be utilized to define the neutrosophic variable MEMBERSHIP and NONMEMBERSHIP functions. Afterwards, the INDTERMINACY functions could be calculated basing on the neutrosophic set definitions.

The experimental results showed the KDD'99 data set features in the neutrosophic format (i.e. the MEMBERSHIP, NONMEMBERSHIP AND INDTERMINACY function are calculated and plotted). Having the neutrosophic format for the KDD'99 data set is the first step toward building a classification model for MANET attacks.

The neutrosophic IDS is implemented via the searching capabilities of the GA. The process takes as an input the neutrosophic definitions of the network features along with the training data set. The GA generates an initial set of neutrosophic if-then rules and begins to match the rule set to the training data.

The most fit rules are chosen upon the neutrosophic correlation coefficient. The GA continue throw the iterations applying the crossover, mutation and selection processes. At the end of the iterations, the final generation of the neutrosophic if-then rules will be the core of the neutrosophic IDS.

This thesis indicates the integration of neutrosophic correlation coefficient into the genetic algorithm for upgrading an effective intrusion detection system. The proposed hybrid can increase the detection rate and reduce the false alarm rate in MANETs networks. The novelty in the proposed technique that used three dimension (MEMBERSHIP- INDETERMENANCY - NONMEMEBRSHIP) all pervious technique used only two dimension (MEMBERSHIP-NONMEMEBRSHIP).

Experimental results prove that the proposed algorithm solves and detects the attacks in an effective manner compared with other existing works. Therefore, it will pave the way for an effective means for intrusion detection with better accuracy and reduced false alarm rate within uncertain and indeterminate environments.

5.2 Future work

Envision the following situations:

- A remote work of rooftop-mounted ad hoc routers;
- An ad hoc network of autos for moment activity and other data;
- sensors and robots shaping a sight and sound system that permits remote perception and control;
- Various airborne switches (from little robots to airships) consequently giving availability and capacity where required (e.g., at a football game);
- An adhoc system of rocket around and in travel between the Earth and Mars.

All thesis imagination scenarios, requires to be secure and more confidential and trusted. Based on the work accomplished in this thesis, future work can be undertaken in the following directions:

- Applying the proposed technique over various kinds of networks other than Mobile Adhoc Networks.
- The performance of the system can be utilized in increasing the security of the system and predicting the intruders in MANETs networks by enhancing issues

such as lack of resource consumption information to achieve an automatic Intrusion Detection System.

- Integrating the system using the two techniques of the prevention and detection methods will give more security for the network.
- Defining then applying the credibility measures for both of the neutrosophic variables and the neutrosophic classification rules would be a good point for research that produces a credible neutrosophic IDS.

REFERNCES

References:

[1] CR Komala, Srinivas Shetty, S. Padmashree, E. Elevarasi , “Wireless Ad hoc Mobile Networks, National Conference on Computing, Communication and Technology, pp. 168-174, 2010.

[2] K. SIVAKUMAR, M.Ph, G. SELVARAJ.” OVERVIEW OF VARIOUS ATTACKS IN MANET AND COUNTERMEASURES FOR ATTACKS”International Journal of Computer Science and Management Research Vol 2 Issue 1 January 2013.

[3] A. A. Salama, Smarandache, Neutrosophic Crisp Set Theory, Educational, Education Publishing 1313 Chesapeake, Ohio 43212 USA, Columbus, 2015.

[4] A. A. Salama, Florentin Smarandache, S. A. Alblowi, New Neutrosophic Crisp Topological Concepts, Neutrosophic Sets and Systems, 2014.

[5] A. A. Salama, Florentin Smarandache, Hewayda ElGhawalby : Neutrosophic Approach to Grayscale Images Domain, Neutrosophic Sets and Systems, vol. 21, 2018, pp. 13-19. https://doi.org/10.5281/zenodo.1408681.

[6] A. A. Salama, Florentin Smarandache, Mohamed Eisa: Introduction to Image Processing via Neutrosophic Techniques, Neutrosophic Sets and Systems, Vol. 5, 2014, pp. 59-64.doi.org/10.5281/zenodo.571456.

[7] A. A. Salama, Mohamed Eisa, S.A.El-Hafeez, M. M. Lotfy: Review of Recommender Systems Algorithms Utilized in Social Networks based e-Learning Systems & Neutrosophic System, Neutrosophic Sets and Systems, vol. 8, 2015, pp. 32-41. doi.org/10.5281/zenodo.571583.

[8] A. Salama, Haitham A. El-Ghareeb, Ayman M. Manie, Florentin Smarandache: Introduction to Develop Some Software Programs for Dealing with Neutrosophic Sets, Neutrosophic Sets and Systems, vol. 3, 2014, pp. 51-52. doi.org/10.5281/zenodo.571453.

[9] A.A.Salama, Mohamed Eisa, Hewayda ElGhawalby, A.E. Fawzy: Neutrosophic Features for Image Retrieval,Neutrosophic Sets and Systems, vol. 13, 2016, pp. 56-61.doi.org/10.5281/zenodo.570857.

[10] A. A. Salama, Basic Structure of Some Classes of Neutrosophic Crisp Nearly Open Sets and Possible Application to GIS Topology, Neutrosophic Sets and Systems, 2015, Volume 7, pp. 18-22.

[11] Eman.M.El-Nakeeb, Hewayda ElGhawalby, A.A. Salama, S.A.El-Hafeez: Neutrosophic Crisp Mathematical Morphology, Neutrosophic Sets and Systems, Vol. 16 (2017), pp. 57-69. doi.org/10.5281/zenodo.831936

[12] K. Atanassov, Intuitionistic Fuzzy Set, Fuzzy Sets and Systems, 1986, Volume 20, pp. 87-96.

[13] KDD Cup 1999 Data, https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[14] I.M. Hanafy, A.A. Salama, K. Mahfouz, Correlation of neutrosophic Data, International Refereed Journal of Engineering and Science (IRJES),2002 Volume 1, PP.39-43.

[15] Haitham Elwahsh, Mona Gamal, A. A. Salama, and I. M. El-Henawy Modeling Neutrosophic Data by Self-Organizing Feature Map: MANETs Data Case Study, Procedia Computer Science, 2017, Volume 121, pp 152-159, DOI.org/10.1016/j.procs.2017.11.021.

[16] Haitham Elwahsh, Mona Gamal, A. A. Salama, and I. M. El-Henawy A Novel approach for classify MANETs attacks with a neutrosophic intelligent system based on genetic algorithm, , Security and Communication Networks, 2018, Volume 2018, Article ID 5828517, doi.org/10.1155/2018/5828517

[17] D.S KUTE., A.S. PATIL, N.V PARDAKHE, A.B KATHOLE. “A REVIEW: MANET ROUTING PROTOCOLS AND DIFFERENT TYPES OF ATTACKS IN MANET”. International Journal of Wireless Communication. Volume 2, Issue 1 pp.-26-28. 2012.

[18] Md Tanzilur Rahman, Kunal Gupta.” MANET: Security Aspects and Challenges”. International Journal of Computer Trends and Technology (IJCTT) – volume 4 Issue 6–June 2013.

[19] M. Refaei, L. DaSilva, M. Eltoweissy, and T. Nadeem,“Adaptation of Reputation Management Systems to Dynamic Network Conditions in Ad Hoc Networks,” IEEE Trans. Computers, vol. 59, no. 5,pp. 707-719, May 2010.

[20] A. Jaganraj, A. Yogaraj, N. Vignesh, R. V. Anuroop.” Handling MANET routing attacks using risk aware mitigation mechanism with distributed node control”. Journal Electrical and Electronic Engineering; 1(3): 61-67; 2013.

[21] Alex Hinds, Michael Ngulube, Shaoying Zhu, and Hussain Al-Aqrabi." Review of Routing Protocols for Mobile Ad-Hoc Networks (MANET)". International Journal of Information and Education Technology, Vol. 3, No.1, February 2013.

[22] A. Perrig, R. Szewczyk, V. Wen, D. E. Culler, and J. D. Tygar, "SPINS: security protocols for sensor networks", In Wireless Networking, Vol. 8, pp. 189-199, 2002.

[23] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Roye, "A Secure Routing Protocol for Ad Hoc Networks", In Proceedings of the 10th IEEE International Conference on Network Protocols, pp. 78-87, November 2002.

[24] L. Zhou and Z. J. Haas, "Securing Ad-Hoc Networks", IEEE Network: special issue on network security, Vol. 13, pp. 24–30, 1999.

[25] H. Deng, W. Li, and D. P. Agrawal, "Routing Security in Ad hoc Networks", In IEEE Communications Magazine, Vol. 40, pp 70-75, October 2002.

[26] L. Zhou and Z. Haas, "Securing Ad Hoc Networks", In IEEE Network, November/December 1999.

[27] V. Karpijoki, "Security in Ad-Hoc Networks", Helsinki University of Technology, Telecommunications Software and Multimedia Laboratory, Website,http://www.hut.fi/~vkarpijo/netsec00/netsec00_manet_sec.ps, 2000.

[28] Y. Zhang, W. Lee, and Y. Huang, "Intrusion Detection Techniques for Mobile Wireless Networks," ACM/Kluwer Wireless Networks Journal (ACM WINET), Vol. 9, No. 5, September 2003.

[29] A. Perrig, R. Canetti, D. Tygar and D. Song, "The TESLA Broadcast Authentication Protocol," RSA CryptoBytes, 5 (Summer), 2002.

[30] M. G. Zapata, "Secure Ad Hoc On-Demand Distance Vector (SAODV) Routing," ACM Mobile Computing and Communication Review (MC2R), Vol. 6, No. 3, pp. 106-107, July 2002.

[31] Y. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure On-Demand Routing Protocol for Ad hoc Networks," Proceedings of the 8th Annual International Conference on Mobile Computing and Networking (MobiCom'02), pp. 12-23, September 2002.

[32] Y. Hu, D. B. Johnson, and A. Perrig, “SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks,” Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA’02), pp. 3-13, June 2002.

[33] L. A. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338-353.

[34] K. Atanassov, neutrosophic sets, Fuzzy Sets and Systems 2087-96. 1986.

[35] 1995-1998 - introduction of neutrosophic set/logic/probability/statistics; generalization of dialectics to neutrosophy**;** http://fs.gallup.unm.edu/ebook-neutrosophics6.pdf

[36] T. Kohonen, Self-Organizing Maps. 3rd Ed., Springer-Verlag, ISBN 3-540-67921-9, 2000.

[37] Kohonen, T., 1990. The self-organizing map. Proc. IEEE 78 (9), 1464–1480.

[38] Mitra, S., Pal, S.K., 1994. Self-organizing neural network as a fuzzy classifier.IEEE Trans. Systems Man Cybernet. 24 (3), 385–399.

[39] Lee W, Stolfo SJ (2000) A framework for constructing features and models for intrusion detection systems. ACM Transactions on Information and System Security 3:227{261, DOI http://doi.acm.org/10.1145/382912. 382914, URL http://doi.acm.org/10.1145/382912.382914

[40] KDD (1999) Intrusion detector learning.

[41] Pfahringer B (2000) Winning the kdd99 classi_cation cup: bagged boost-ing. SIGKDD Explor Newsl 1:65{66, DOI http://doi.acm.org/10.1145/846183.846200, URL http://doi.acm.org/10.1145/846183.846200

[42] Brugger S (2007) Kdd cup 99 dataset (network intrusion) considered harm-Ful

[43] R. Bace, P. Mell, NIST Special Publication on Intrusion Detection Systems, Technical Report, National Institute of Standards and Technologies, 2001.

[44] S. Sen, J.A. Clark, Intrusion Detection in Mobile Ad Hoc Networks, Springer, 2008, pp. 427–454.

[45] Donald Welch and Scott Lathrop, “Wireless Security Threat Taxonomy”, IEEE Workshop on Information Assurance, pp. 76–83, 2003.

[46] Ravi K. Balachandran, Ramamurthy B., Xukai Zou and Vinodchandran N.V., CRTDH: “An Efficient Key Agreement Scheme for Secure Group Communications in Wireless Ad Hoc Networks”, IEEE International Conference on Communications, Vol. 2, pp. 1123–1127, 2005.

[47] Kamer Kaya and Ali Aydm Selcuk, “A Verifiable Secret Sharing Scheme Based on the Chinese Remainder Theorem”, Lecture Notes In Computer Science, Proceedings of the 9th International Conference on Cryptology in India: Progress in Cryptology, Vol. 5365, pp. 414–425, 2008.

[48] Tina Suen and Yasinsac A., “Ad hoc network security: peer identification and authentication using signal properties”, Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop, pp. 432–433, 2005.

[49] Shichun Pang and Shufen Liu, “An ECC based Vector Space Key Sharing Scheme”, 1st International Symposium on Pervasive Computing and Applications, pp. 524– 527, 2006.

[50] D. Hankerson,A. Menezes and S. Vanstone, “Guide to Elliptic Curve Cryptography”, Springer Verlag, 2004.

[51] Wei Liu, Yanchao Zhang, Wenjing Lou and Yuguang Fang, “Securing Mobile Ad Hoc Networks with Certificateless Public Keys”, IEEE Transactions on Dependable and Secure Computing, Vol. 3, No. 4, pp.386–399, 2006.

[52] PI Jian-yong, LIU Xin-song, WU Ai and LIU Dan, “A Novel Cryptography for Ad Hoc Network Security”, International Conference on Communications, Circuits and Systems Proceedings, Vol. 3, pp. 1448–1452, 2006.

[53] Chris Piro, Clay Shields, and Brian Neil Levine, “Detecting the Sybil Attack in Ad Hoc Networks”, Proceedings from IEEE/ACM International Conference on Security and Privacy in Communication Networks (SecureComm), pp. 1–11, 2006.

[54] Marianne A. Azer, Sherif M. El-Kassas and Magdy S. El-Soudani, “Threshold Cryptography and Authentication in Ad Hoc Networks Survey and Challenges”, Second International Conference on Systems and Networks Communications, pp. 5–11, 2007.

[55] Pierre E. Abi-Char, Abdallah Mhamed and Bachar El-Hassan, “A Secure Authenticated Key Agreement Protocol Based on Elliptic Curve Cryptography”, Proceedings of the Third International Symposium on Information Assurance and Security, pp. 89–94, 2007.

[56] A. Rex Macedo Arokiaraj and A. Shanmugam, “ACS: An efficient address based cryptography scheme for Mobile ad hoc networks security”, International Conference on Computer and Communication Engineering, pp. 52–56, 2008.

[57] Mengbo Hou and Qiuliang Xu, “Key Replicating Attack on Certificateless Authenticated Key Agreement Protocol”, Asia-Pacific Conference on Information Processing, pp. 47–50, 2009.

[58] Yuguang Fang, Xiaoyan Zhu and Yanchao Zhang, “Securing resource-constrained wireless ad hoc networks”, IEEE Wireless Communications, Vol. 16, No. 2, pp. 24–30, 2009.

[59] Joonsang Baek, Reihaneh Safavi Naini, Willy Susilo and Jan Newmarch, “A Survey of Identity-Based Cryptography”, Proceedings of Aug 2004, Identification and Authentication Issues in Computing, 2004.

[60] Chih-Chung Yang, and N.K. Bose , Generating fuzzy membership function with self-organizing feature map Pattern Recognition Letters, Volume 27, Issue 5, 1 April 2006, Pages 356–365

[61] Horikawa, S. 1997. Fuzzy classification system using self-organizing feature map.Oki Tech. Rev. 63(159), [Online]. Available from: <http://www.oki.com/en/otr/html/nf/otr-159-05.html>.

[62] I. Turksen, Interval valued fuzzy sets based on normal forms, Fuzzy Sets and Systems 20 (1986) 191-210.

[63] Takagi, H., Hayashi, I., 1991. NN-driven fuzzy reasoning. Internat.J.Approx.Reason. 5, 191–212.

[64] Haitham Elwahsh, Mohamed Hashem , Mohamed Amin, “Secure Service Discovery Protocols for Ad Hoc Networks”, Springer (LNCS) in Computer Science, Advances in Computer Science and Information Technology Communications in Computer and Information Science, 2011, Volume 131, Part 1, 147-157, DOI: 10.1007/978-3-642-17857-3_15

[65] J. Ross Quinlan, C4.5: programs for machine learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, 1993

[66] Muhammad Syafiq Mohd Pozi, Md Nasir Sulaiman, Norwati Mustapha, Thinagaran Perumal, Improving Anomalous Rare Attack Detection Rate for Intrusion Detection System Using Support Vector Machine and Genetic Programming. Neural Process Lett (2016) 44:279–290. DOI 10.1007/s11063-015-9457-y.

[67] O. Hussein, T. Saadawi, Ant Routing Algorithm for Mobile Ad-hoc networks (AMMA), 2003, Conference Proceedings of IEEE International Performance, Computing, and Communications Conference., DOI: 10.1109/PCCC.2003.1203709

[68] G. V. Nadiammai and M. Hemalatha, “Effective approach toward Intrusion Detection System using data mining techniques,” Egyptian Informatics Journal, vol. 15, no. 1, pp. 37–50, 2014.

[69] Machine Learning Group at the University of Waikato https://www.cs.waikato.ac.nz/ml/weka/.

[70] Madjid Tavana, Farshad Azizi, Farzad Azizi, Majid Behzadian, "A fuzzy inference system with application to player selection and team formation in multi-player sports, Sport Management Review, 2013, Volume 16, pp 97-110,

[71] R. Becker, S. Eick, A. Wilks, Visualizing network data, IEEE Transactions on Visualization and Computer Graphics, 1995, Volume 1, pp 16-28, DOI.10.1109/2945.468391.

[72] Ehsan Amiria, Hassan Keshavarzb, Hossein Heidaric, Esmaeil Mohamadid, Hossein Moradzadehe, Intrusion Detection Systems in MANET: A Review, Procedia - Social and Behavioral Sciences, 2014, pp. 453 – 459.

[73] R. Thanuja, A. Umamakeswari, EFFECTIVE INTRUSION DETECTION SYSTEM DESIGN USING GENETIC ALGORITHM FOR MANETs, ARPN Journal of Engineering and Applied Sciences, 2016, Volume 11.

[74] S. Sen, J.A. Clark, Intrusion Detection in Mobile Ad Hoc Networks, Springer, 2008, pp. 427–454.

[75] F. Anjum, P. Mouchtaris, Security for Wireless Ad Hoc Networks, Wiley-Interscience, 2007.

[76] M.R. Gauthama Raman , Nivethitha Somu , Kannan Kirthivasan , Ramiro Liscano , V.S. Shankar Sriram , An Effcient Intrusion Detection System based on Hypergraph - Genetic Algorithm for Parameter Optimization and Feature Selection in Support Vector Machine, Knowledge-Based Systems (2017), doi: 10.1016/j.knosys.2017.07.005.

[77] Muder Almi'ani, Alia Abu Ghazleh, Amer Al-Rahayfeh, Abdul Razaque, Intelligent Intrusion Detection System Using Clustered Self Organized Map, 2018 Fifth International Conference on Software Defined Systems (SDS), DOI: 10.1109/SDS.2018.8370435.

الملخص

يعتبر أمان ألشبكات اللاسلكية ذات العقد المتحركة MANETS مجال مهم لللباحثين الاكاديمين وكذلك المتخصصين الغير اكاديمين. كما ان تصميم نظام كشف التسلل (IDS) يعد من أصعب المشاكل في "الشبكات اللاسلكية ذات العقد المتحركة " (MANETs). يكمن السبب الرئيسي فى هذه الصعوبات الى الطبيعة المتغيرة وغير مستقرة لشبكاتMANETs. ومن ثم فإن نظام كشف التسلل (IDS) سوف يحتاج الى التطور بحيث يعتمد النظام برمته على مفاهيم عدم اليقين و الضبابية. وتعتبر هذه المفاهيم هي القضايا الرئيسية التى يهتم بمعالجتها نظام Fuzzy System، وايضاً في نظام النيتروسوفيك Neutrosophic. في تقنية النيتروسوفيك Neutrosophic، يتحدد كل هجوم (تسلل) بدرجة من المصداقية MEMEBERSHIP والخطأ NONMEMEBERSHIP وكذلك عدم التوقعindeterminacy الا ان العقبة الرئيسية هي البيانات المتاحة والتى فى معظمها قيم عادية ليست مناسبة لحسابات النيتروسوفيك Neutrosophic.

تقترح الرسالة نظام للاستدلال على الهجوم (التسلل) الى MANETs بإطار هجين من SOFM والخوارزميات الجينية (GA). يستخدم الهجين قدرات التعلم الذاتى فى ال SOFM لتعريف المتغيرات الشرطية للنيتروسوفيك فى الشبكات اللاسلكية ذات العقد المتحركة MANETs ثم تقوم الخوارزميات الجينية بايجاد مجموعة القواعد الشرطية المسئولة عن تحديد نوع المعاملات على الشبكة من حيث ما اذا كانت هجوم ام عادية وذلك كما هو موضح فى المراحل التالية.

<u>ويمكن تقسيم العمل على مرحلتين:</u>

1. **مرحلة إعداد المتغيرات :**

تعتبر المرحلة الاولية فى تصميم نظام كشف التسلل تحت مظلة تقنية النيتروسوفيك Neutrosophic. حيث يتم تحويل البيانات العادية لقيم النيتروسوفيك وحساب كلاً المصداقية MEMEBERSHIP والخطأ NONMEMEBERSHIP و عدم التوقع indeterminacy لكل متغير (variable) فى الشبكة. خلال تنفيذ

هذه المرحلة يستفيد النظام من خرائط الميزات ذاتية التنظيم (SOFM) لتقسيم مساحة المتغيرات إلي الفئات المناسبة و ذلك فى ضوء مساعدة من الخبراء فى مجال الشبكات. لقد استخدمت هذه الطريقة لتعريف دالة fuzzy MEMEBERSHIP ، ومن ثم يمكن أن تستخدم في تعريف دالتى المصداقية ودرجة الخطأ. بعد ذلك نستخدم تعريفات المجموعات داخل النيوتروسوفيك neutrosophic فى حساب دوال عدم التوقع للمتغيرات. وبذلك يتم الاستفادة من قدرات واستخدامات (SOFMs) فى تجميع المدخلات باستخدام تقنيات اعتماد ذاتي و توليد دوال النيتروسوفيك لمجموعات فرعية لمتغيرات الشبكة ومن ثم دراسة و اكتشاف الهجمات (التسللات) الموجودة بقاعدة البيانات KDD-99.

2. مرحلة تصميم نظام كشف التسلل (Neutrosophic IDS) :

فى هذه المرحلة يتم تغذية متغيرات النيتروسوفيك للشبكة جنبا إلى جنب مع مجموعة البيانات إلى الخوارزمية الجينية (GA) للعثور على القواعد الشرطية neutrosophic rule set الاكثر ملاءمة من عينة من الهجمات الأولية الفرعية وفقا لدالة الثقة fitness function. هنا يتم اعتبار دالة الثقة هى مقدار الترابط neutrosophic correlation coefficient بين مقدمات ونتائج القواعد الشرطية . يهدف هذا الأسلوب للكشف عن الهجمات غير المعروفه في الشبكات اللاسلكية ذات العقد المتحركة (MANETs). لقد تمت محاكاة النتائج التجريبية على قاعدة بيانات شبكة الهجمات (KDD-99) والمتوفرة في المستودع آلة التعلم UCI لمزيد من المعالجة في اكتشاف المعرفة. لقد اثبتت التجارب ان IDS Neutrosophicله القدرة على تحديد الهجمات بشكل ادق وبمعدل انذار خاطىء اقل من الانظمة الاخرى الموجودة فى مجال اكتشاف التسلل.

كلية العلوم
قسم الرياضيات و علوم الحاسب

جامعة بورسعيد

تأمين الشبكات اللاسلكية ذات العقد المتحركة بإستخدام تقنية النيتروسوفيك

رسالة مقدمة إلى كلية العلوم – جامعة بورسعيد
لاستيفاء الحصول على درجة الدكتوراة فى علوم الحاسب

مـــن

هيثم سامي محمد الوحش

ماجستير علوم الحاسب
مدرس مساعد علوم الحاسب-كلية الحاسبات والمعلومات
جامعة كفر الشيخ

تحت إشراف

الأستاذ الدكتور / إبراهيم محمود الحناوي
أستاذ علوم الحاسب
كلية الحاسبات والمعلومات-جامعة الزقازيق

الأستاذ الدكتور/ أحمد عبد الخالق سلامة
أستاذ الرياضيات وعلوم الحاسب
كلية العلوم – جامعة بورسعيد

الأستاذ الدكتور / مجدي البنا
عميد كلية العلوم

الدكتور / وائل عبد القادر
رئيس قسم الرياضيات وعلوم الحاسب

2018