A Computationally Intelligent Approach to the Detection of Wormhole Attacks in Wireless Sensor Networks

By

Mohammad Nurul Afsar Shaon

A thesis is submitted to the faculty of graduate studies of

The University of Manitoba

In the fulfillment of the requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering

University of Manitoba

Winnipeg, Manitoba

Abstract

This thesis proposes an innovative wormhole detection scheme to detect wormhole attacks using computational intelligence and an artificial neural network (ANN). The aim of the proposed research is to develop a detection scheme that can detect wormhole attacks (In-band, out of band, hidden wormhole attack, active wormhole attack) in both uniformly and non-uniformly distributed sensor networks. Furthermore, the proposed research does not require any special hardware and causes no significant network overhead throughout the network. Most importantly, the probable location of the wormhole nodes can be tracked down by the proposed ANN-based detection scheme.

We evaluate the efficacy of the proposed detection scheme in terms of detection accuracy, false positive rate, and false negative rate. The performance of the proposed model is also compared with other machine learning techniques (i.e. SVM and regularized nonlinear logistic regression (LR) based detection models) based detection schemes. The simulation results show that proposed ANN-based detection model outperforms the SVM and LR based detection schemes in terms of detection accuracy, false positive rate, and false negative rates.

Acknowledgments

First, I would like to express my gratitude to Prof. Dr. Ken Ferens for his supervision in my research on cyber security and his guidance during my M.Sc. Study at the University of Manitoba. I would like to thank him from the core of my heart for being a great advisor and an awesome person to work with. Thank you very much once again for tolerating my naive questions so long.

I want to thank the Government of Manitoba and University of Manitoba for supporting my research on cyber security and higher study in Canada.

I would like to thank my beloved friends Kawser Been Amir and Kamrul Hasan Shajib for inspiring me to do M.Sc. abroad and teaching how to keep faith on me.

Finally, I would like to thank Almighty ALLAH for my life, unconditional love and, granting me to do M.Sc. at the University of Manitoba.

Dedicated to

I dedicate this work to my parents and adorable sisters for their unconditional love and bias support. Their inspirations to work hard and, to do better are the motivating factors and driving force of my life.

Table of Content

List of Figures

List of Tables

Chapter 1

1. Introduction

Wireless sensor networks (WSNs) consist of self-directed devices (i.e. sensor nodes), which are used in a collective manner monitoring physical or environmental phenomena in a remote and/or hostile environment. Spatially distributed autonomous sensor nodes can communicate amongst themselves in order to forward sensed data to the base station. The WSN is an especial type of ad-hoc network that has gained popularity for its versatile application in military and civil domains such as battlefield monitoring, tracking objects, healthcare, and home automation.

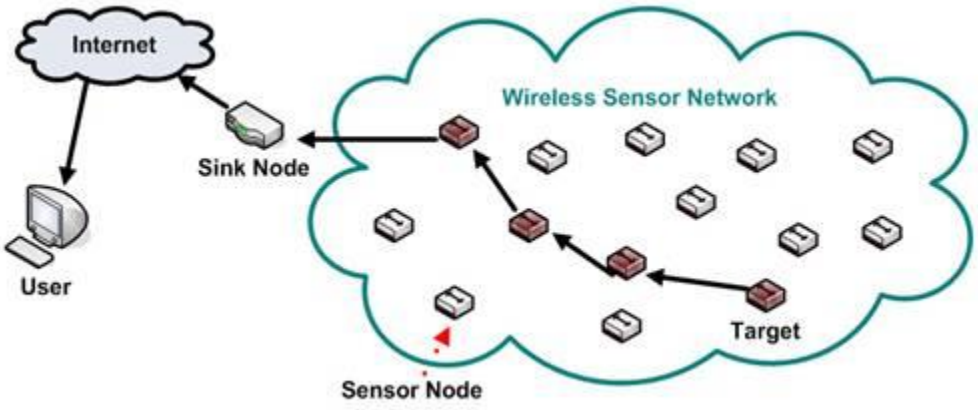

Figure 1.1 Wireless sensor network [Printed without permission, from 60].

In WSN, a larger number of sensor nodes perform an assigned task in a hostile environment without any human intervention. Since sensor nodes use a known in-band radio channel for communication and are usually deployed in a hostile or remote environment, therefore, WSNs are prone to various security threats like a sinkhole attack, sybil attack, and wormhole attack. WSN has several vulnerabilities that an attacker can exploit to obtain access to the network. Implementing security measures like data encryption is not practical for most WSN since the

firewalls become very complicated and challenging task due to the limited communication and energy resources and distributed nature of the nodes. However, research on security measures has made certain progress in secure localization algorithm, lite weight routing protocols, lite weight encryption scheme, and secure data aggregation scheme. However, those security mechanisms don't provide any protection against any attack from a legitimate node. If an attacker gains control over a few legitimate nodes, full access would be gained to the data traveling through these compromised nodes. The attacker may achieve the capability to modify the contents of the control packets later by extracting the cryptographic contents from the compromised node.

Wormhole attack is recognized as one of the most detrimental security threats for any routing protocol of WSNs. The wormhole attack can be easily launched by taking over at least two legitimate nodes from the two distant parts of the network or deploying two nodes with superior capabilities (e.g. directional antenna, larger radio range) in two distant places of the sensor field. Wormhole nodes are connected through a virtual tunnel which can be implemented in numerous ways (e.g. high-quality channel, packet encapsulation or packet relay and high powered transmission) [1]. This direct low latency link is known as a *wormhole link* [2]. A *wormhole link* creates an illusion in the network that these two colluding nodes are located within their communication range, but, in fact, their physical locations are very far apart. By creating this unauthorized link, wormhole nodes gain the ability to circulate false route information into the network that they are few hops away from the base stations. This illusion drives other sensor nodes to transmit collected data to the base station through the wormhole nodes. Wormhole attack disrupts the existing network data flow in order to monitor and capture the data packets passing through it.

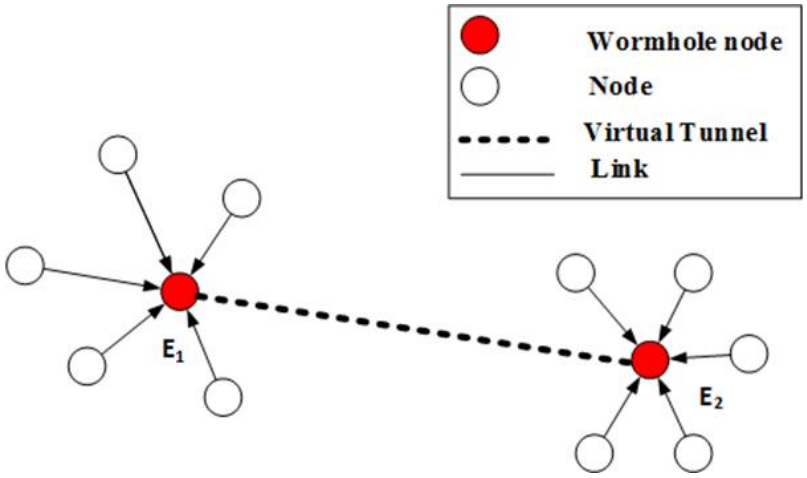

Figure 1.2 The depiction of the structure of a wormhole attack.

As shown in Figure 1.2, the E_1 and E_2 wormhole nodes, connected by a *wormhole link*, capture the data packets from one terminal of the virtual tunnel and retransmit them to another terminal of that link.

Subsequently, this wormhole attack becomes so severe that it might destroy the network or hamper the usual operation of the network by the selective dropping of packets, manipulation of traffic, or modify data packets without revealing their identities. Therefore, detection of wormhole nodes is an essential task for ensuring the security of wireless sensor networks. It is a very simple task to implement wormhole attack, but a very difficult task to detect an infected network since wormhole nodes retransmit valid packets into the network. Most of the existing countermeasures use the distance bounding technique, direction, and location abnormality among claimed neighbor nodes as detection attributes to fight against wormhole attack. To gain a certain level of accuracy, some existing schemes uses complex and highly advanced devices such as directional antenna[3], GPS[4], or ultrasound for distance measurement [5]. However, incorporation of this special hardware to each node makes the scheme more costly and impractical for the deployment. A few statistical wormhole detection schemes based on hop count [6], node connectivity [2], or

neighborhood count [7][8][9] are proposed that do not need any special hardware. However, they usually include a hardware supported scheme as a secondary approach. Furthermore, centralized statistical wormhole detection [7] may cause significant network and communication overhead in contrast to a distributed statistical approach [8]. In the network connectivity based wormhole attack detection schemes [2][10][11], the positions of neighboring nodes are estimated from the received signal strength (RSSI) by each node, which sends this information to the base station. By doing this, the network layout is determined by the base station and compared with the given network layout. This approach also causes a significant amount of control packets flow to the base station. Moreover, it is prone to distance estimation errors. Furthermore, neighborhood-based wormhole detection schemes [9][7] may not detect wormhole attack if the wormhole nodes are located in a sparsely populated area and caused the significant flow of packets to the base station. In addition, their performance in non-uniform sensor distribution is in question.

In recent years, network anomaly detection schemes have been increasingly using artificial intelligence to improve detection accuracy. An artificial neural network (ANN) is a very simplified information processing model that aims to grossly imitate the human brain function. An ANN consists of interconnected processing units and works in a parallel fashion to find a solution to a non-linear problem. The adaptive and self-learning ability of an ANN help to increase the competence of an anomaly detection model [12]. Moreover, ANNs have been widely deployed to deal with pattern recognition and classification problems [13].

1.1. Thesis statements

In this thesis, we introduce a novel detection scheme based on an ANN using '*neighborhood count*' and '*average residual energy pattern of neighbors (AREPN)*' as detection attributes. The

proposed detection scheme can detect wormhole attacks in both uniform and non-uniform sensor distributions and does not need any special hardware. Here, we have introduced a mobile node, called as detector node (D_N) that visits randomly chosen locations within the region of interest and collects two featured data samples, along with coordinate at each site visited. When the detector node D_N moves into a wormhole infected zone, this paper *theorizes* that the collected number of neighbors increases *abnormally* (uniform network scenario) or slightly *abnormally* (non-uniform network scenario), compared to a non-infected zone, in which the counts change *normally*. This thesis paper also introduces a new detection attribute, named *AREPN* that significantly low at the wormhole infected zone compare to non-infected zone. D_N captures these attributes as the evidence of the existence of a wormhole node in the network. In this detection scheme, the gathered dataset is used to train and test the learning performance of the ANN. After the training phase, the test data samples are fed into the ANN. Based on the network output, the scheme decides if there is any data sample that represent wormhole attack in the network.

1.2. Contributions of the Thesis

a) Machine learning algorithms like ANN, SVM or Logistic regression (LR) are widely used to trace the threats or anomalous behavior from the internet traffic. In this thesis, the ANN is introduced as a mean of analysis tool to identify the existence of wormhole in the network. In this study, The ANN is applied on the two featured simulated data set to detect wormhole attack from the infected WSNs.

b) In this thesis, our proposed detection model can identify the presence of any malicious node from both uniform and non-uniform sensor distributions. This detection scheme doesn't need any special hardware such as a directional antenna, sonar, GPS, and

ultrasound to detect wormhole attack. That makes the detection scheme inexpensive and practical.

c) The excessive network overhead flow is one of the major causes of energy dissipation of the sensor node. In this thesis, we have introduced a mobile node, also known as detector node, to collect the two featured data samples from the infected network and offload those collected samples to the base station once it is within the communication range of the base station. By introducing this node, the network overhead throughout the network is significantly reduced. That would be helpful to increase the lifetime of the network significantly.

d) Neighborhood count is one of the important attributes to identify the presence of the wormhole node in the network. If the malicious nodes place themselves in the sparsely populated area of the network, then it will be difficult for the detection scheme to identify wormhole nodes in the network. In this thesis, we introduce a new detection attribute called *AREPN* that significantly enhanced the performance of the proposed detection scheme.

e) We have studied in detail and compared the efficacy of the proposed algorithm with SVM and LR based detection schemes through simulations. The simulation results confirm that an ANN based wormhole detector can detect wormhole attacks with higher precision and accuracy as compared to the SVM and non-linear logistic classification based detection models.

1.3. Outline of the thesis

The organization of this thesis paper is arranged as follows: chapter 2 presents a literature survey of wormhole attack detection schemes and their countermeasures for WSNs. Chapter 3 provides the brief description on wormhole attacks, their classification and the impact of the

wormhole attacks on infected wormhole attacks. Chapter 4 describes the background of some machine learning algorithms such as ANN, SVM, and LR. This is chapter provides an in-depth detail of the ANN. Chapter 5 presents the proposed ANN-based detection scheme. This chapter explains how the detector node collects the two featured data samples from the infected WSN, how this ANN based detector identifies both the presence of wormhole and the probable location of the malicious node. Chapters 6 gives the results from the experimental works. It explains simulation setup, analyzes the outcome of the experiments and compares the results of different machine learning-based detection scheme (ANN, SVM, LR). Chapter 7 provides the conclusion of the thesis and discusses the scope of works that can be done to enhance the performance of the ANN-based detection scheme in future.

Chapter 2

2. Related Works

As the wormhole attack can be launched from legitimate nodes (compromised) and relays the valid packets throughout the network; therefore, it is reasonably difficult to detect it from the infected network. Furthermore, lightweight cryptographic solutions are incorporated into the most of the routing protocol to prevent injection of fictitious data packets into the network. As the legitimate nodes are compromised by the adversary and the wormhole node doesn't change the packet's content at the initial stage, so the wormhole node easily passes the cryptographic test. The wormhole attack is a simple task to launch, but very difficult to identify. Many researchers have been working on this field to develop efficient wormhole detection schemes based on the geographical locations, transmission time, connectivity graph, neighborhood counts and radio fingerprint.

2.1. Distance consistency based approach

Most of the researchers in this field try to detect the wormhole attack by distance bounding techniques. In these approaches, two communicating nodes are allowed to determine the distance between them; based on message traveling time information, transmission time information, and geographical information. Sometimes sensor nodes are equipped with specialized hardware like the directional antennas, GPS, ultrasound[5] to measure the distance between two adjacent sensor nodes. However, these schemes are considered impractical due to the addition of the special hardware and their performance in a sparse sensor network. These schemes may not perform well in the presence of *'out of band'* wormhole nodes.

2.2. Time information based approach

The most popular detection model of wormhole attacks uses the packet traveling time between two consecutive sensor nodes as a detection attribute. Most of the cases, the data packets traveling time is estimated from the measured round trip time (RTT). In [4][14][15][16][17][18][19], the authors proposed RTT based solutions to confront wormhole attacks. In [14], delay per hop is determined by measured RTT whereas, for each successive hop, RTT is measured to detect wormhole attack [15]. In these schemes, the distance between two adjacent nodes is measured using RTT and determine if the two communicating sensor nodes are apart by the valid possible distance. If the measured RTT surpasses a defined threshold value, the presence of wormhole node will be declared. However, the RTT based solutions require the collaboration of the sensor nodes on the path and don't work properly on the dynamic source routing (DSR) and destination sequenced direct vector routing (DSDV) routing protocols [20][21]. These RTT based solutions may not perform well if the wormhole is created by protocol deviation. The wormhole nodes forward the packets without waiting for the certain time in order to reach the destination earlier than legitimate multiple hop path. Beside this, in the *'out of band'* wormhole attack, the wormhole nodes use the high-speed low latency link for transferring packets between them. In this type of attack, data packets reach to another wormhole node more quickly than *'in-band'* wormhole attack. In fact, the time required to forward the packets to another wormhole node is significantly reduced in '*out of band'* wormhole attack. Therefore, the performance of RTT based detection scheme is in question with the presence of *'out of band'* wormhole attack. Some literature [22] also suggested that the RTT based detection scheme cannot detect *'active'* mode of wormhole attack and may not detect wormhole attack in the sparse sensor network.

In [19], the authors proposed a detection scheme based on neighborhood count and RTT. The authors considered the two facts. First, as the malicious nodes establish the new link in the network,

so the neighbors around the malicious node increases significantly. Second, the measured RTT between two malicious nodes increases significantly in a similar fashion. However, this proposed solution may not identify *'out of band'* wormhole attack from the infected network.

In [22], the authors proposed a detection scheme based on the data transmission time of each successive hop in a route assuming that the wormhole nodes are in hidden mode. In this scheme, the data transmission time of each successive hop is calculated from the measured RTT. All sensor nodes in the route transmit the recorded transmission time to the source. The source is solely responsible for checking if there is any wormhole node exist in the route. This scheme may perform poorly in sparse sensor networks. In fact, the performance of this scheme may be degraded when malicious nodes use high-speed communication channel.

2.3. Special hardware based schemes

The directional antennas are incorporated with sensor nodes for adding access restriction [3][23][24][25] and finding legitimate neighboring nodes in the sensor network. The entire communication region of a sensor node is divided into several zones. Moreover, each zone is defined by a directional antenna. When a sensor node captures the signal first time from its peering sensor node, then the probable location of the sender in terms of the zone is determined by the directional antennas. According to the authors, if a sensor node sends a packet from a particular zone, the recipient will get the signal from the opposite direction (zone). After that, the recipient sensor node verifies the legitimacy of the sender through receiving direction of the packet from unknown sensor nodes. The recipient node also cooperates with its neighbors to find out whether this node is known to other legitimate neighboring nodes or not. The incorporation of directional antennas to each sensor node makes this scheme expensive and inappropriate for practical deployment.

Another protocol named SECTOR was introduced in [17], that mostly depends on special hardware. The main concept of this detection scheme is to measure the distance between two communicating nodes based on the data transmission speed. The proposed model does not need any clock synchronization and geographical information of communicating nodes. In this algorithm, the mutual authentication with distance bounding (MADB) protocol [26] was implemented to measure the distance between nodes at the time of the encounter. According to the authors in [26], the proposed scheme permits one node to measure the mutual distance between two nodes and compares with maximum possible upper bounding distance. In this scheme, each node is incorporated with the special device that can give feedback to the sender without any delay. Initially, a node sends the one-bit challenge to its neighbors before sending any packets. Then recipient node responds back through a special device to the sender at the time it receives the one-bit challenge. When the sender transmits the one-bit challenge to a node, it turns on the clock and measures the time till it gets the response back from that node. After that, the sender estimates the mutual distance based on the measured time considering the data transfer speed as same as the light propagation speed. However, in the [18], the authors slightly modify the schemes described in [26]. In [18], both parties are allowed to measure the mutual distance and validated authenticity of the adjacent nodes around their communication range. The addition of the special device to each sensor node makes this scheme expensive and inappropriate for practical deployment.

2.4. Geographical information based solution

In [27] and [4], the authors proposed a model that assigns a maximum data traveling period to each data packet. The authenticity of a data packet is ensured by using the concepts of geographical and temporal packet leashes that would help to minimize the effectiveness of the wormhole in the network. In the geographical leash (GL), when a sender sends packets to any

sensor node, the sender node incorporates sending time and its own location information to the packet. When the receiver node captures the packet, it will estimate the packet traveling based on the geographical leashes. The temporal leash includes the maximum lifetime to each packet. In temporal leash (TL), a sender adds either packet transmitting time or expired time so that the recipient can verify if the packet has traveled valid distance based on maximum transmission speed and time. A predetermined time threshold is set between any two neighbors based on their position in the network. For a specific two neighboring nodes, the data packet would be discarded if it violates the time boundary set by these specific nodes. This scheme can perform better if strict time synchronization is maintained and the additional GPS hardware is incorporated to each node. Beside this, this proposed scheme may perform poorly when the wormhole node is in active mode.

2.5. Trust based solution

Trust information of neighboring nodes is used as an attribute to detect wormhole attacks in the WSNs. Each sensor node monitors the data packet forwarding pattern of its neighbors and rates them accordingly. In the trust base scheme, a sensor node selects the most trustworthy path to reach the destination based on the trustworthiness of its neighbors. In this scheme, the researchers consider the fact that the wormhole node drops all the received packets coming from its adjacent nodes. It is expected that the system would rate the least trust level to the malicious nodes. By using this trust level scheme, the wormhole node can be avoided during the selection of a path to the destination. This also helps to reduce the effectiveness of the wormhole nodes in the infected network. The source node would forward the packets to the neighbor which possess maximum trustworthiness.

In [28], a new detection scheme was proposed to identify wormhole in the network based on both time and trust. In this scheme, both data travel time and trustworthiness of the sensor are used

as potential attributes to detect the malicious nodes from the infected network. These two modules of this proposed scheme run simultaneously. Here, the time-based module works in three stages. Firstly, each sensor node identifies its neighbors and updates the neighbor list accordingly. Secondly, each node discovers the best path towards the base station based on the trustworthiness of neighbors. Finally, the proposed algorithm determines the presence of the wormhole nodes in the selected path. According to the authors, the malicious nodes deceive the time-based solutions in many ways. Hence, the trust-based model is incorporated with the time base scheme. In this scheme, a sensor node monitors the neighboring node continuously and give them ratings. Based on the given ratings each node selects the best route to the destination.

Another trust based wormhole attack detection scheme was proposed in [29].To execute the model, the deployed sensor node must operate in a particular mode, named as 'promiscuous mode'. This trust-based scheme is applied to the DSR protocol and inherent features of DSR routing protocol are used to measure the trust level of the neighboring nodes. Here, the algorithm must be executed in each sensor node and each node must estimate the trustworthiness of its neighboring nodes by monitoring the packet transmission pattern stated by the system. The source node verifies in several stages whether the forwarding node passes the data packets or not through a series of integrity checks. If the neighboring node for a source forwards all the packets, the trust level of the node would be increased. Similarly, if the opposite happens, the trust level of that node would be decremented.

The success of this module lies on the packet dropping criteria of the malicious nodes. However, the wormhole nodes do not drop packets in the hidden mode of this attack. Hence, the trust base detection model is not capable of detecting the hidden wormhole attack. On the other hand, each node monitors the packets forwarding pattern of its neighbors. As we know, sensor

nodes have some constraints on power and energy resources. It would be a burden for the system which leads to excessive energy dissipation of the nodes.

2.6. Graph-based solution

Multi-dimensional scaling-visualization of the wormhole (MD-VOW) [30], was proposed based on the graph theory. The multi-dimensional scaling analysis of the constructed connectivity graph was used as an analysis tool to identify the presence of malevolent nodes in the network. For the static sensor network, the connectivity graph is not supposed to change frequently. Hence, the authors have considered the fact that the presence of malicious nodes introduces anomalies in the connectivity graph. In the network connectivity based wormhole attacks detection scheme, the positions of the neighboring nodes are estimated from the received signal strength (RSSI) by each node. After that, each node transmits this information to the base station for further analysis. By compiling the received information, the base station determines the network layout and compares it with the current connectivity graph. Then the presence of the wormhole node can be detected if any forbidden structure is found in the constructed network layout. However, it is prone to the distance estimation errors, especially for the sparse network. The surface smoothing technique is applied to the constructed network layout graph to compensate distance error. This approach also causes a significant amount of control packets flow to the base station. Similarly, if the wormhole node is located in the sparsely populated area, then the wormhole attack may not be identified by this network visualization based algorithm.

2.7. Radio fingerprinting based scheme

In [31], the authors explored the potentiality of the fingerprinting device as a tool to validate the legitimacy of the node in a wireless sensor network. The most important goal of this research

is to extract the features from the radio signals, radiating from the nearby sensor nodes; by which the legitimacy of a node can be evaluated. In this research, each node must be equipped with the radio fingerprinting device. The radio fingerprinting device captures the radiated radio signal from the nearby sensor nodes. After that, the fingerprinting device converts the radio signals into digital format. The transient part of the signal is located and the important features are extracted. After that, those extracted features are taken into account as fingerprints so that recipient sensor node can identify the neighboring node. The authors also implemented the incorporation of the radio fingerprinting device with a sensor node [31]. In this research, the authors also showed that the fingerprinting approach identified the nearby node while the message contents and the identification of the nearby devices were hidden. There are a few issues required to be investigated in this approach. The radio fingerprinting devices' performance in a noisy environment, and in the mobile platform is in question. By using this detection method, only the *'out of band'* wormhole attack can be detected. This proposed model may find difficulties to detect '*in-band*' wormhole attack as it can be launched from the legitimate node.

2.8. Neighborhood Count based Solution

The number of neighbors is used in [7][8][9] as detection attributes in the neighbor based detection scheme. In the centralized method, each sensor node finds the number of neighbors within its communication region and sends this information to the base station. As the distribution of the sensor node is known, the base station computes the hypothetical distribution of neighborhood counts along with the true distribution of the neighborhood counts. This process also creates a significant amount of control data packet flow throughout the network and leads to the unexpected energy dissipation of sensor node. This process is also used as secondary approach

with the distance-based scheme. In another neighborhood count based approach [9], detector node takes the count of neighbors at the each site it visited. In this approach, ANN-based detection scheme may not detect the wormhole attacks if the wormhole nodes are located in the sparsely populated area. Another important factor is that, by this scheme the probable location of the wormhole node cannot be identified.

Chapter 3

3. The Wormhole Attack

A wormhole can be a severe attack against any packet routing protocol, especially in ad-hoc and wireless sensor networks. It is very difficult to detect and to take preventive measures against wormhole attack since the malicious nodes behave as legitimate nodes and initially do not perform any illegal activity in the network. The word ‘wormhole’ means the creation of any shortcut path between two far apart points in the space-time [32]. Thus, the concept of ‘wormhole’ is used as a tool to launch this attack aiming to spoil the existing routing protocol.

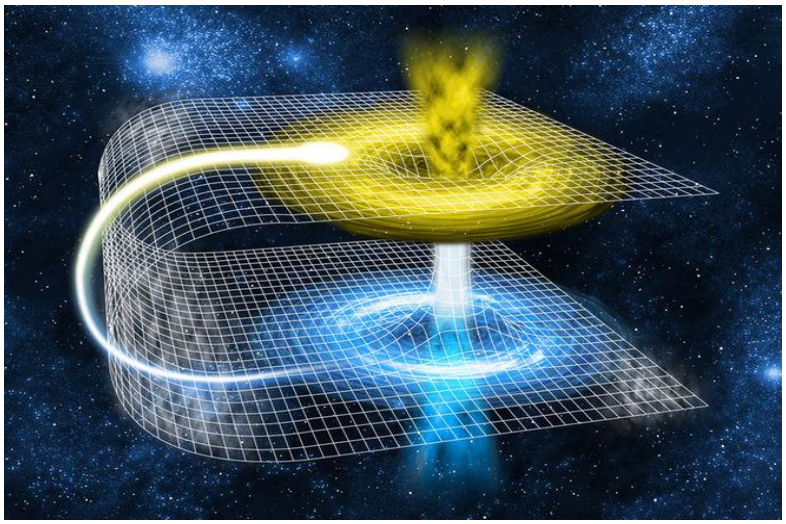

Figure 3.1 Creation of Wormhole in space [Printed without permission, from 59].

The wormhole attack starts by compromising at least two nodes from the sensor network by hacking or, deploying two nodes with superior capabilities (e.g. directional antenna, larger radio range) in two distant places of the network [33]. This attack would be more devastating if the aggressor launches the attack with multiple nodes. However, wormhole attack can be launched

with the single node by broadcasting received packets with high power level [34]. Those malicious nodes are known as '*wormhole node'*. Furthermore, to launch this attack, the wormhole nodes form an unauthorized low latency communication link for their own usage. This link is called the '*wormhole link*'. In some literature, this link is also named as *wormhole tunnel* [8]. The wormhole nodes gain unprecedented access to the network by forming this low latency link. This wormhole link can be formed in numerous ways, such as packet encapsulation, wired link and out of band radio link [1]. Packet encapsulation is the most prominent way to establish wormhole link in the network where smallest hop count uses to select the best route towards any destination.

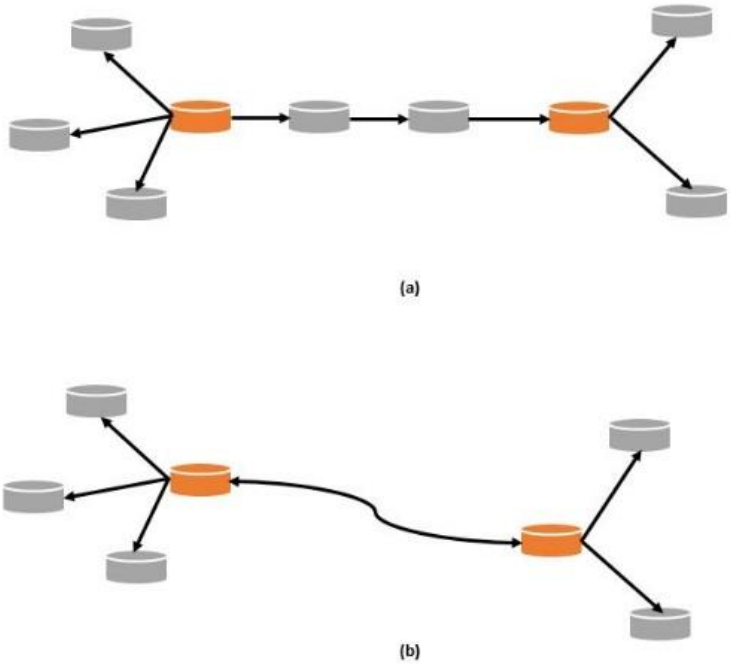

Figure 3.2 The depiction of (a) 'In-band' and (b) 'Out-band' wormhole attack.

The wormhole attack can be categorized into two classes based on the wormhole link formation. In the '*out of band'* wormhole [35], two external nodes (modified) are deployed with

higher communications and computational power than deployed sensor nodes. This kind of attack is hard to establish due to the requirement of the special hardware. In this case, the adversary connects two separate zones of the network using a wired link or a directional antenna. Similarly, in the *'in-band'* wormhole, the adversary takes over at least two legitimate nodes from different zones of the network. One of these is usually located close to the base station so that it could disguise its adjacent nodes by advertising fabricated routing information. In contrast to '*out of band'* wormhole attack, the aggressor uses packet encapsulation technique to create wormhole node rather than a wired link or directional antenna [9]. In this circumstances, two compromised nodes are connected through several legitimate nodes between them.

Once the wormhole link is functional, one of the colluding nodes transmits the packets, collected from an area of the network, towards another malicious node. The other malicious node broadcasts those received packets into its radio range [36]. The wormhole node influences those nodes, who are normally multiple hops away from it, to send data packets via wormhole by convincing them that they are few nodes away from the base station [37]. In the other words, due to the high-speed wormhole link, the received data packets (by wormhole nodes) would travel faster from one part of the network to another part of the network than a usual multi-hop route. This illusion would disrupt existing packet routing mechanisms.

At the initial stage, wormhole node eavesdrops or captures the packets passing through it for further analysis and retransmits them to another wormhole node. When the wormhole attack begins, malicious nodes do not know about the cryptographic keys are being used in the network. If this malicious node starts dropping the packets without knowing the content of the packets, the target of compromising integrity and confidentially would not be achieved. On the other hand, the dropping of the packets might rise suspicion of those nodes who have relayed the packet through

wormhole node. Therefore, there is a chance of being detected as a malicious node in the network. In fact, Wormhole node tries to place itself in the most of the route without revealing its identity. However, the initial phase of wormhole attack is called '*hidden wormhole attack' [34]*. In the '*hidden mode of attack'*, the wormhole does not appear in the routing table. Still, wormhole node is able to establish the denial of service (DOS) [38] attack by dropping packets in the '*hidden mode of attack'*.

Once the adversary figures out the encryption keys, is being used in the network, by accessing program memory and storage of the compromised node[39]; then the aggressor takes the attack to the new level. In the '*active mode of wormhole attack'*, the malicious node takes part in the routing mechanism as legitimate node [34] and starts modifying or dropping the packets passing through it [1]. In some cases, wormhole node drops the selected or critical packets to interrupt the usual operation of the network [35].

3.1. Modes of wormhole attacks

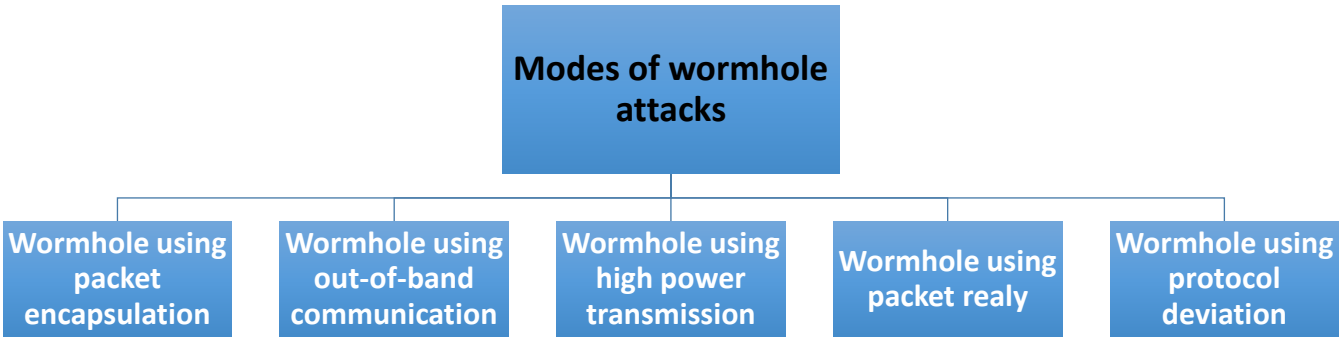

Figure 3.3 Modes of wormhole attacks.

3.2. Wormhole attack using packet encapsulation

Packet encapsulation is one of the prominent methods of creating a wormhole in the network. In this type of attack, one wormhole node captures route request packets (RREQ) from other sensor nodes and sends those RREQ packets to another malicious node through several legitimate nodes, which lie in between them. Since the malicious node forwards the encapsulated data packets, the hop count does not change during the transmission of packets [40]. Thus, the data packets are transferred from a source to the base station through two malicious nodes. In fact, this encapsulation technique prevents sensor nodes from discovering a legitimate route that is multiple hops away (from source to destination).

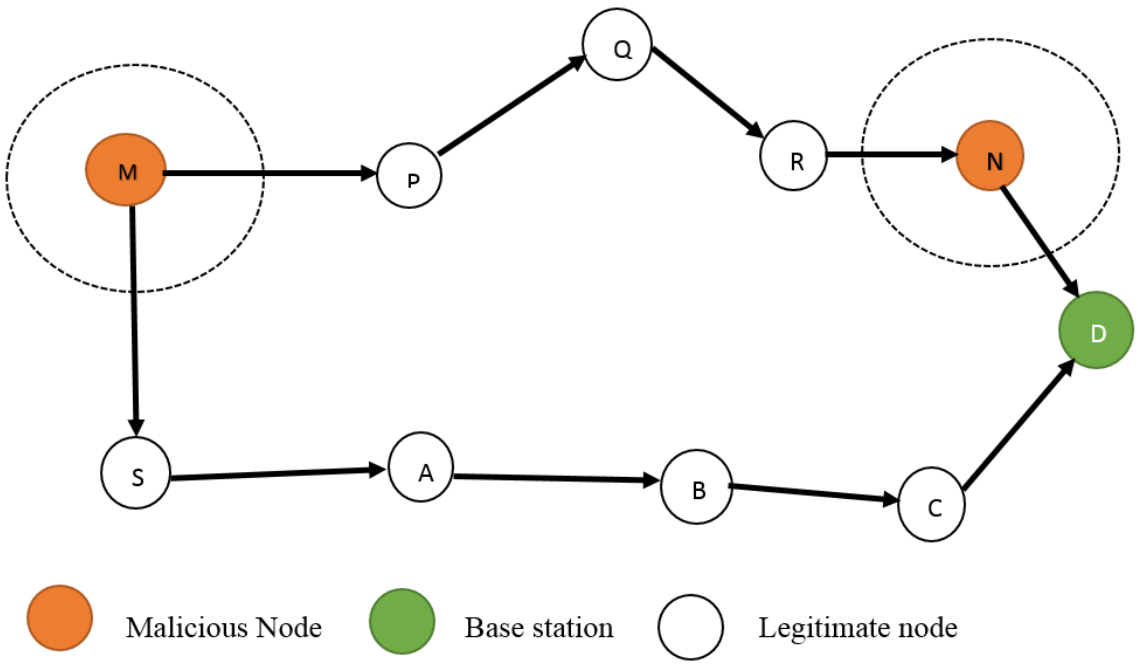

Figure 3.4 Wormhole attack using packet encapsulation.

Consider that node S wants to send some collected data to the base station, D. Hence, the node S tries to discover the shortest path in between base station D to node S with the existence of two malicious node M and N. The node S initiates the broadcast of RREQ and this broadcasted RREQ message is received by one of the malicious nodes M and legitimate node. M encapsulates the received RREQ, which is destined to another colluding node, N, through the existing path $(P - Q - R)$ between M and N. In this process, the hop count doesn't increase during the packet traversal through $(P - Q - R)$ due to the encapsulation method. After this, another malicious node, N receives the RREQ message sent by M and rebroadcasts it within its radio range. As the base station D is a neighboring node of N, D hears the RREQ broadcasted by N. Simultaneously, another copy of RREQ also travels through the path including $(A - B - C)$ to the base station D. The first path is two hops long, but in reality, it is five (05) hops long. The second route is apparently three (03) hops long. Since the first route appears as the shortest path; therefore, the base station D will choose the first route over the second legitimate route.

This mode of wormhole attack is very easy to establish since two malicious nodes don't need to know the cryptographic keys are being used or to have any special hardware such as directional antennas, high-quality communication resources. The probable solution for this mode of attack is to use the fastest time to reach RREQ message to the destination as metric for selecting best route [4].

3.3. Wormhole attack using out of band channel

In this type of wormhole attack, two malevolent nodes are connected through a single hop high-quality communication channel. This single hop channel can be established by direct high-speed wired link or long range wireless link. This type of attack is very difficult to launch since

two malicious nodes need to have special hardware than another mode of wormhole attacks. Let's assume the scenario, depicted bellow.

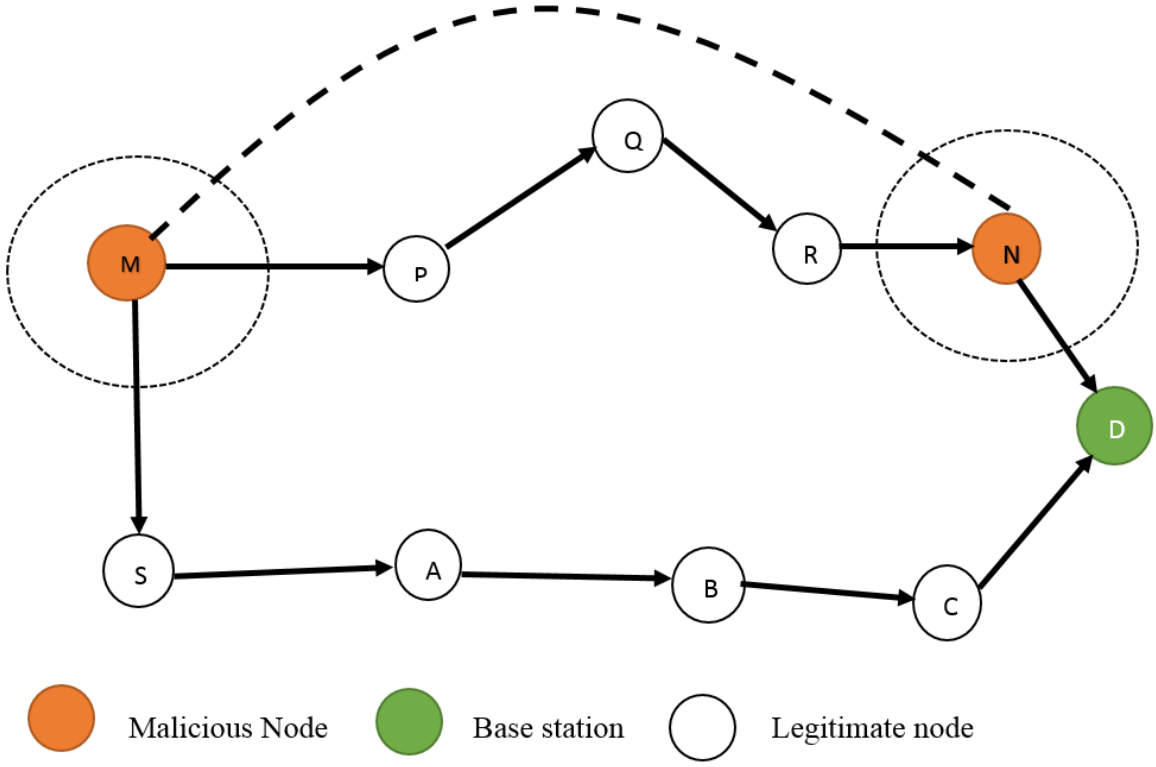

Figure 3.5 Wormhole attack using out of band channel.

Here, two malicious nodes M and N are connected through a directional wireless link. The node S wants to send some collected data packets to the base station D. The node S broadcasts RREQ and the one of the malicious node M within its neighborhood receives it. After that, M forwards RREQ message to another malicious node N through a different radio channel. M rebroadcasts it again within its communication range. Due to this broadcast, the base station D hears the RREQ message coming from the node M via malicious node N. Concurrently, another copy of the RREQ reaches to the base station D through the path $A - B - C$. Since the RREQ

message received to D (via $M - N$) faster than the second route and it appears to be the shortest path, thus the base station D will choose the first route via $M - N$ over the second route.

3.4. Wormhole attack using high power transmission

In this mode, one of the malicious nodes has the capability to communicate with other normal nodes over long-distance using high-power transmission. When the malicious node gets RREQ, it transmits the received message at high-power level. Assume the fact that no other sensor node has the capability to send the packet at high-power level except that malicious node. The sensor node usually rebroadcasts the RREQ, if the destined sensor node is not located within its communication range. Thus, the nodes that hear the broadcasted RREQ, rebroadcasts it again to reach the destined node. By doing this, the malicious node wants to increase the probability of placing itself in the route towards any destination especially sink node even without taking help from another colluding node.

3.5. Wormhole attack using packet relay

In this mode of attack, the malevolent node transmits data traffic between two non-neighboring sensor nodes by convincing them that they are direct neighbors. Packet relay based wormhole attacks can be initiated by single wormhole node. To launch this attack, the intruder must have a larger radio range compare to another legitimate node. In some literature, this attack is also mentioned as '*replay-based attack'* [48].

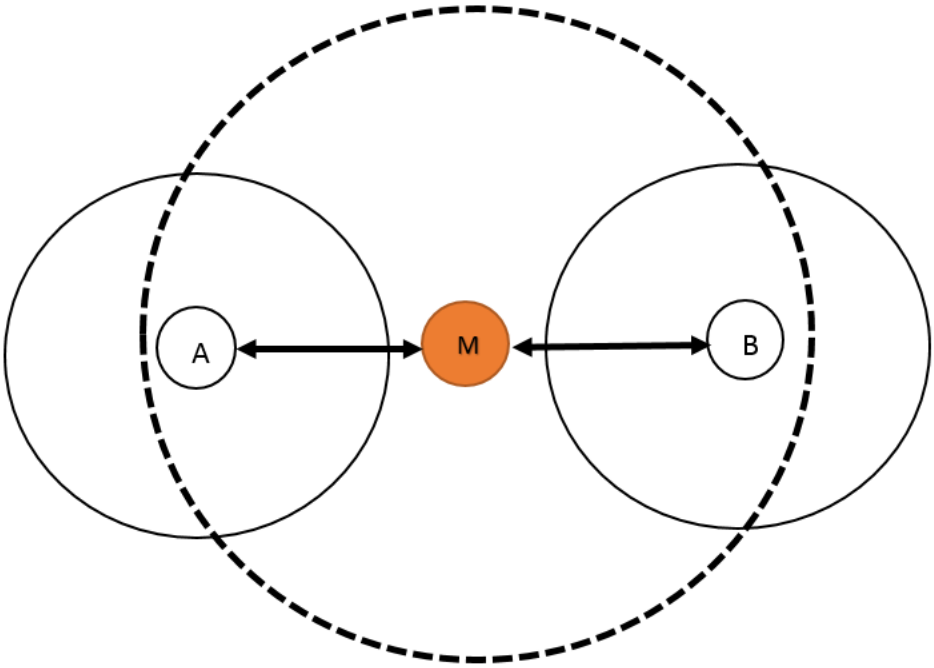

Figure 3.6 Wormhole attack using packet relay.

The malicious node M has the superior radio range than the other two sensor nodes (A and B). Let's consider that, node A wants to send packets to the node B. The malicious node receives the packet coming from A and sends it to B without mentioning its ID in the packet header. Hence, the malicious node M becomes virtually invisible to both node A and node B. The node B doesn't know fact that the packet is relayed by the node M. Thus, both nodes are forced to believe that they are direct neighbors. In this case, the malicious node M controls the link between legitimate node A and node B. At any time of operation, the malicious node M can drop packets coming through it or break the link between node A and node B.

3.6. Wormhole attack using protocol deviation

The wormhole attack can be launched by not complying with the routing protocol. When a sensor node receives the packet from other sensor nodes, it has to be backed off for a certain amount of time before transmitting packets in order to avoid MAC layer collision. The malevolent node violates this rule and sends the packets without waiting for certain time. The main aim of forwarding the RREQ packets without backing off is, to reach the destination first. In the literature, this attack is also named as '*rushing attack'*. In many situations, the intruder initially launches this mode of attack in order to mount DOS attack, which is a severe threat to the whole network.

3.7. Variant of wormhole attack

There are several attacks exist in the WSNs which are similar to the wormhole attacks. The most common and known attack for WSN is spoofing attacks. In spoofing attacks, the identity of a legitimate node is stolen by the intruder and hence, all the data packets heading to the victimized node are captured by the aggressor [41]. The spoofing attack is similar to the hidden wormhole attack. At the same time, there are many types of spoofing attacks, such as invisible node attack [42], stolen identity [43] and Sybil attack [44].

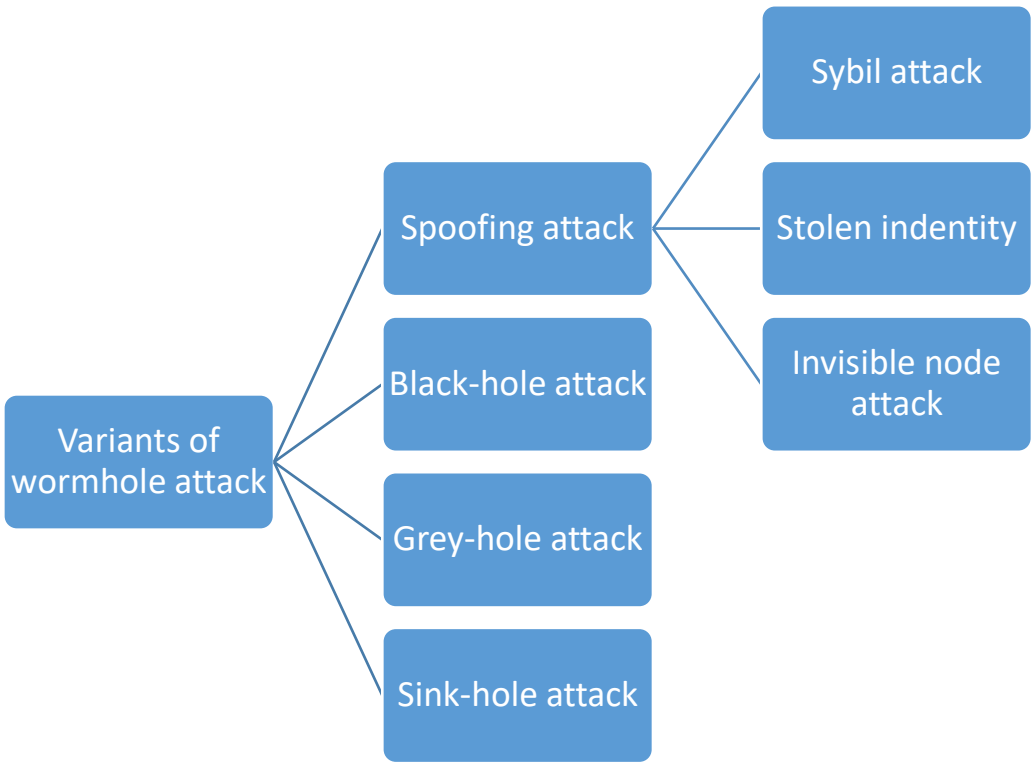

Figure 3.7 Variant of wormhole attack.

However, there are also other types of attacks, that are related to the wormhole attack, namely black hole attack, grey hole attack and sinkhole attack. The main objectives of these attacks are either gain access of the collected data that is sent toward the base station or interrupt the data packet flow towards the base station. In the black-hole attack, the malicious nodes drop all the received packets coming from their neighboring nodes to degrade the performance of the network. Thus, this phenomenon also increases the probability to get caught by the existing detection scheme. To evade the intrusion detection scheme, the attacker might introduce a more sophisticated attack, known as a grey-hole attack. In this grey hole attack, the attacker drops the

critical data packets selectively so that it could cause a major interruption to the operation of the system.

In the sinkhole attack, one of the malicious nodes places itself in the network as a neighboring node of the base station or creates a high-quality single hop link with the base station through different radio channel [33]. The malicious node usually has higher capabilities than deployed legitimate nodes in terms of communication and computational resources. The intruder tries to the put itself in most of the routing paths toward the base station. Sometimes, this attack targets any potential node in the network instead of sink node. In this case, the sinkhole attack hampers availability and usability of this targeted sensor node. After gaining the access to the data packets directed to the base station, the malicious node alters the data packets flow or drops the packets in order to suppress the network performance.

3.8. The impact of the wormhole attack

Wormhole attack is considered as severe threats for routing protocols of WSNs. This attack usually occurs in the network layer and is immune to encryption techniques. The wormhole attack is able to degrade the efficacy of the existing routing protocol and compromises the integrity and confidentiality of the data packets traveling throughout network [45]. Once wormhole nodes get the access to the network, the adversary can drop the packet selectively or delay the transmission of critical packets for the system in order to destabilize system performance [35]. The aggressor tries to establish the denial of service attack (DOS) and attempts to compromise the integrity and confidentiality of the network.

During the active mode of this attack, the malicious nodes become the sinkholes [1]. However, other nodes around the wormhole send data packets without knowing the fact that they are the

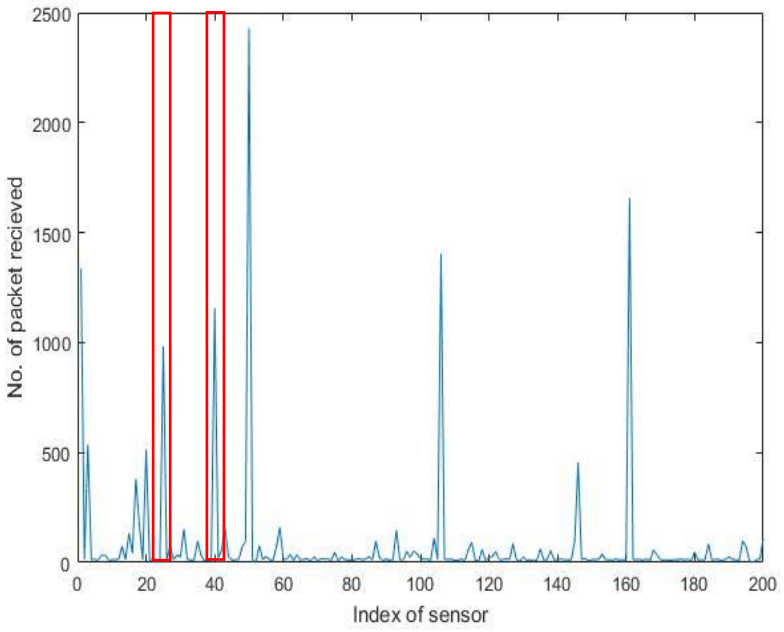

Figure 3.8 Wormhole Packet reception pattern

victims of the wormhole attacks. Since the major share of data packets traverses through the malicious nodes, the attacker may control and monitor the packet without having multiple observation points in the network. Some literature also suggests that wormhole node also lessens the throughput of the network by the selective dropping of the packets [1]. Furthermore, wormhole node also can turn on and off the wormhole link randomly [1]. This event creates instability in the routing service and causes a significant amount of control packets flow throughout the sensor network suddenly.

In general, the routing protocol of the wireless sensor network can be categorized into two classes, such as ‘pro-active’ routing protocol and ‘on-demand’ routing protocol. Routing updates

are transmitted periodically in the pro-active routing protocol, whereas on-demand routing protocol searches the route to a specific destination when it is necessary. However, the wormhole attack is successful to invade the network accessibility for both classes of wormhole attack [46]. Some literature on wormhole attack mentions that two wormhole nodes are able to attract more than 50% data traffic towards them directed to the base station [3], [47].

Chapter 4

4. Background on Machine learning algorithms

This chapter describes the background on the machine learning algorithms such as ANN, SVM, and LR in detail. The first section explains the structure of the ANN and how the ANN defines a non-linear relationship of given data samples and actual outputs. The following section explains the SVM and LR in detail.

4.1. Artificial neural networks (ANN)

The Artificial Neural Network (ANN) is a network based stochastic learning model that has evolved from the study, characteristics, organization, and decision-making ability of the unit cell of a human brain called a neuron [48]. In other words, it aims to imitate the most simplified and basic function of the human brain. Analogous to the unit cell of a human brain, an ANN contains several interconnected information processing units, called a neuron, learning the underlying process of the data samples presented to the network. The significant phenomena of ANN are the ability to estimate the non-linear complex relationship between inputs and outputs without any prior knowledge of dataset like a black box. ANN is usually observed as a model of interconnected neurons that maps the input and outputs through information exchange among neurons.

4.1.1. Natural neuron and Analogy to the ANN model

The fundamental building block of the human neural network is a neuron. A typical human neuron has four parts- a cell body (Soma) which is responsible to perform non-linear complex operation, an array of input paths or spiked (myriad) extension of soma to receive input signals from the adjacent neurons (Dendrites), a relatively long output path that carries the output signal from the soma away from the neuron, and multiple axon terminals (synapses) that connect the

neurons with other neurons. The dendrites capture the transmitted signal from the adjacent neurons and forward them toward the cell body [49]. The synaptic terminals are connected with dendrites of the other adjacent neurons or the effector cells in the muscles or glands.

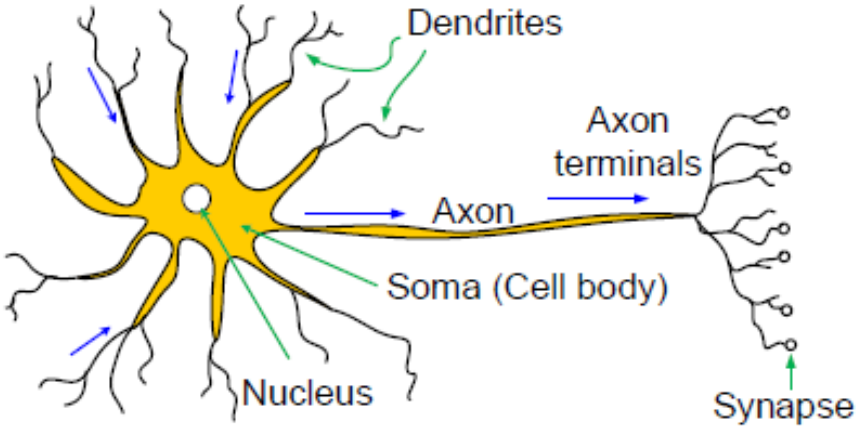

Figure 4.1 Drawing of natural neuron [Printed without permission, from 58].

The main objective of a neuron is to capture the incoming signals from the other neurons and determine whether to send or not to send the processed electrical signal to the other neurons, muscles, or glands. The neuron usually receives signals from the external environments or from the other adjacent neurons. In the soma, all the received information is integrated and summed up. Based on the strength of the summed input, the neuron determines the transmission of the output from the cell body. The most noticeable aspect of the neuron is it can transmit and receive both electrical and chemical signals simultaneously. Furthermore, neuron sends the action potential (normalized summed inputs) to the end terminal of the neuron (Synapses) through the axon. When the signal reaches at the end terminal of the neuron, it forces the synaptic terminals to trigger a

release of the chemically encoded message. That is why the synaptic terminals are also called neurotransmitters [50]. These chemically encoded signals relay across the synapses to the next neuron or the effector cell. The magnitude, density of the chemical release is not well understood or defined. However, the response of the receiver can be either excitatory or inhibitory, depending on the characteristics of the receptor. If the received signal surpasses the certain threshold of recipient neuron, it would activate the recipient neuron and forces it to send an excitatory signal along the axon to the synaptic terminal.

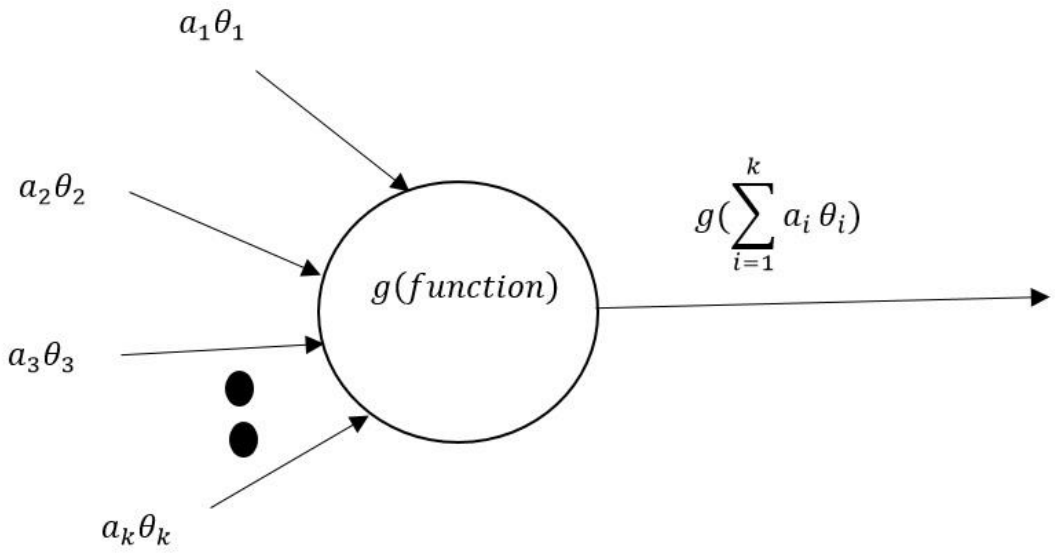

Figure 4.2 The basic operation of the natural neuron.

For example, let's say n neurotransmitters send the electrical activation levels (a_k) to the specific neuron q through and along the axon (shown in Figure 4.2). At the synaptic terminal of the k^{th} neuron, the activation level (a_k) is multiplicatively amplified before transferring to the dendrites of the adjacent neuron q. The amplification process can be understood through weight

value (θ_k). Therefore, the k^{th} neuron sends out the activation level $a_k\theta_k$. Dendrites of the neuron q receive those activation levels and transmit to the cell body. In the cell body of the neuron q, all the received activation levels are summed up and normalized by the activation function. If the sum of the activation levels is strong enough to surpass the predefined threshold, then the neuron q sends the excitatory signal along the axon to the synaptic terminal. Otherwise it transmits the inhibitory signal to the synaptic terminals. The threshold value of the neuron can be known easily, though we can model it to be learned as the weight value θ_0.

The exact behaviors of the neuron are still a mystery for the researcher, though some aspect of the neurons is known. There is the diverse class of neurons in the human brain whose exact mechanism are completely different from each other. The myriad dendrites and synapses of the neurons perform the non-linear complex computation that cannot be modeled till now. Hence, drawing a serious analogy between neural network models and the human brain is not suited well. The artificial neural networks try to imitate the basic functions of the most complex, diverged and potent building block of the human brain, the neuron. The artificial neural networks have been evolved or modified to solve the computational problems which cannot be solved by the conventional methods. It is just a new way to solve problems.

4.1.2. Structure of the ANN

Furthermore, in a multi-layer perceptron (MLP), there is an input layer, followed by one or more hidden layers, and an output layer [51]. In each layer, several neurons are employed, which are fully connected with other neurons of an adjacent layer, and they are associated with different random weight values [52]. In other words, neurons are fully attached to the neurons of the

following layer; but in the same layer, neurons are not connected with each other. The number of neurons in the output layer depends on the type of the problem that we want to solve by the ANN.

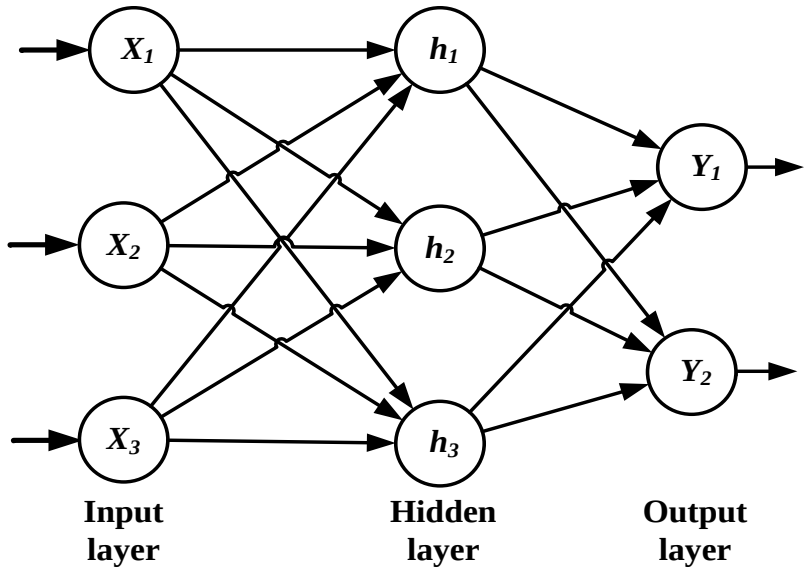

Figure 4.3 The structure of the ANN.

4.1.3. Forward propagation

Input features from the input layer are shared with an adjacent hidden layer through unidirectional branches [53]. Those input features are multiplied by random weights associated with the unidirectional branches; summed up, and passed through the activation function of the neuron (e.g. sigmoid function). The bias term is also added to each layer except the output layer to activate the artificial neurons. This bias term is also connected to the neurons of the adjacent layer with unidirectional branches associated with the weight values ($b_i^{(l)}$) so that the summed inputs exceed the predefined threshold. In the forward propagation, the output of each neuron of

the prior layer is considered as the input to all neurons of the following layer. As shown in the Figure 4.4, the three layers relate to the weight value $\theta_{ij}^{(l)}$. $\theta_{ij}^{(l)}$ represents the weight value going to the unit i in layer $(l+1)$ and coming from the unit j in layer l.

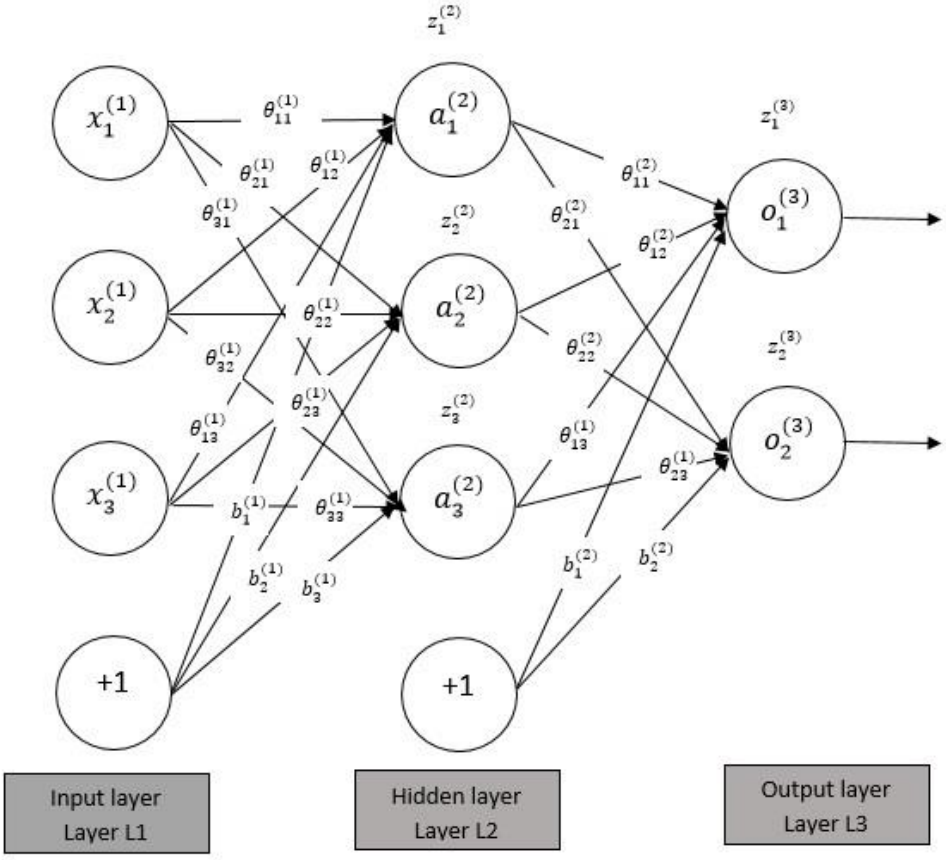

Figure 4.4 Forward propagation of the ANN.

According to the Figure 4.4, the net output of i^{th} unit (including bias term) of the hidden layer ($a_i^{(l)}$) is,

$$Z_i^{(2)} = \sum_{j=1}^{n} \theta_{ij}^{(1)} x_j^{(1)} + b_i^{(1)} \quad (1)$$

$$a_i^{(2)} = g(z_i^2) = g\left(\sum_{j=1}^{n} \theta_{ij}^{(1)} x_j^{(1)} + b_i^{(1)}\right) \quad (2)$$

Similarly, the output of k^{th} unit in the output layer is,

$$o_k^{(3)} = g\left(\sum_{j=1}^{n} \theta_{kj}^{(2)} a_j^{(2)} + b_k^{(2)}\right) \quad (3)$$

The cost function,

$$J(\theta) = [\sum_{k=1}^{n} \frac{1}{2m} \sum_{i=1}^{m} (o_k^m - y_k^m)^2] \quad (4)$$

In the equation (4), $J(\theta)$ refers to the average error occurred over m training samples during the training procedure, and k defines the number of neurons in the output layer.

The transfer function of a neuron plays an important role in the training procedure. The purpose of the transfer function is to replicate the activation mechanism of the biological neuron. Usually, the net output of a unit in a layer is calculated from the net received input through the transfer function. The transfer function must be non-linear, continuous and differentiable at any point in order to apply gradient descent learning algorithm. There are several types of transfer function such as sigmoid function, hyperbolic tangent function, and rectified linear function. Some literature [50] also suggests that the rectified linear function was found to accelerate the convergence rate of stochastic gradient descent due to its non-linear and non-saturating form. However, in our research, the sigmoid function was used as the transfer function of each unit in any layer (except input layer).

$$g(z) = \frac{1}{1+e^{-z}} \quad (5)$$

$$g(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (6)$$

$$g(z) = \begin{cases} 0 & if\ z \leq 0 \\ z & if\ z > 0 \end{cases} \quad (7)$$

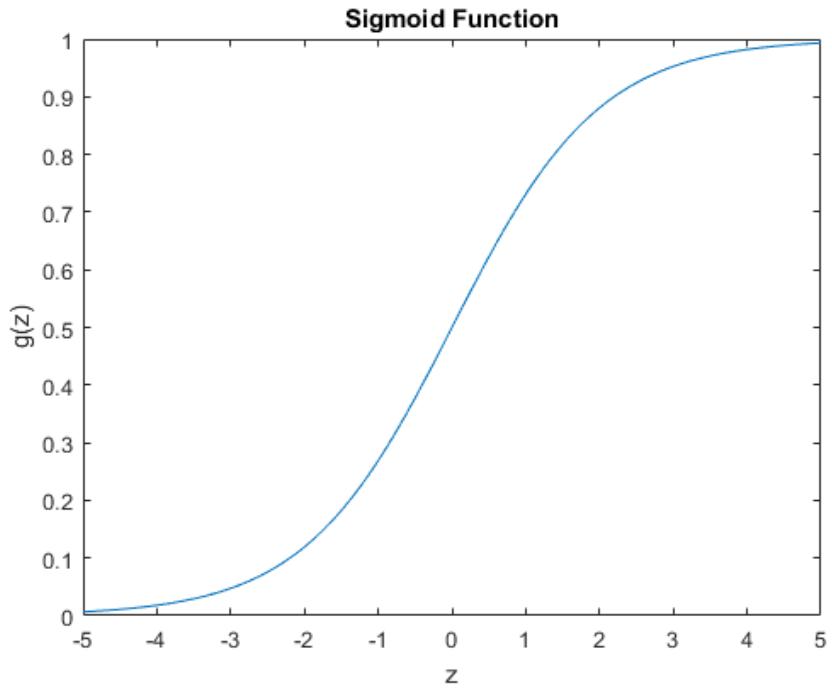

Figure 4.5 Sigmoid function.

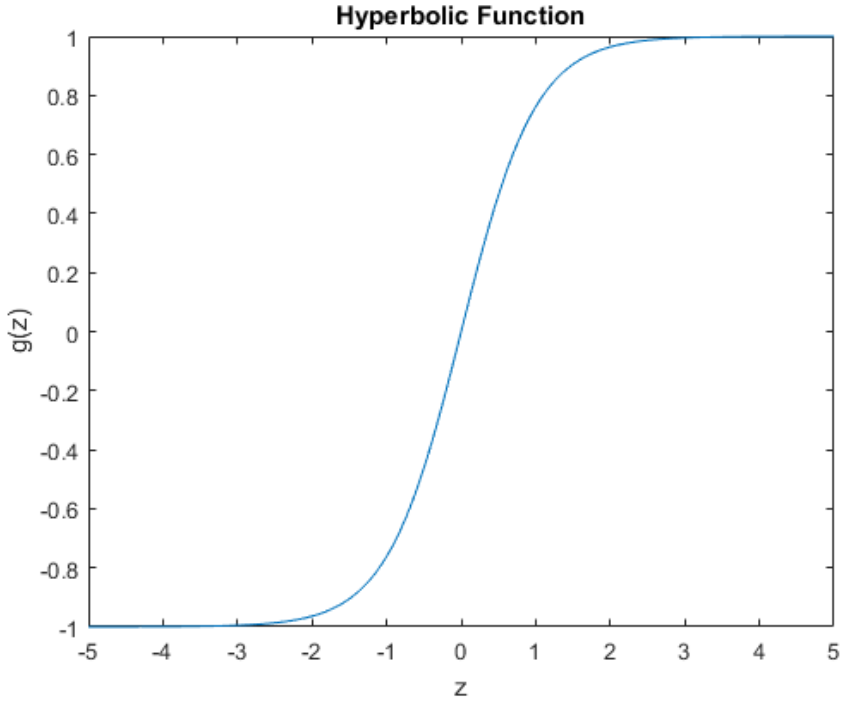

Figure 4.6 Hyperbolic function.

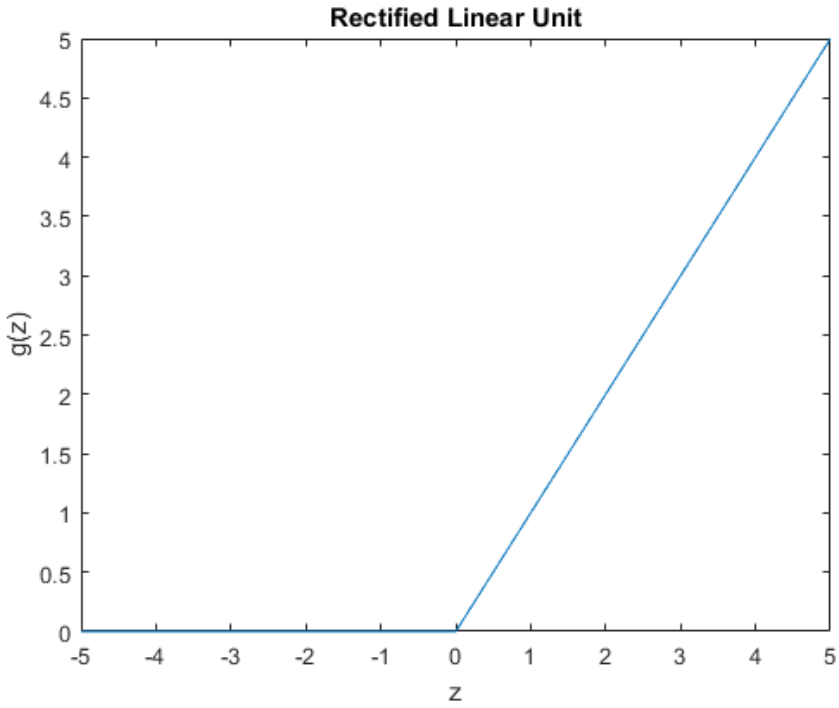

Figure 4.7 Rectified linear unit.

4.1.4. Backward Propagation

In the back propagation, the outcome of the output layer is compared with actual output. The error is measured and propagated backward to adjust the branch weights in order to minimize error that would occur due to the estimation of output. In other words, we minimize the cost or energy of the error function, $J(\theta)$, by passing back the error occurred during the training phase. In the back-propagation algorithm, the gradient descent algorithm is applied to learn the network parameters $(\theta_{ij}^{(l)}, b_{ij}^{(l)})$ from the training set $\{x_i^m, y_i^m\}$.

However, there are many ways to update the weights: batch mode, online mode. In the batch mode, each weight is updated after measuring the total error occurred in an epoch (i.e. after one

training cycle). In contrast, in the online mode, a training sample is drawn randomly from the training set and passes through the network. The network parameters are modified m times per training cycle if there are m training examples in the training set. This online mode of weight updating is also known as stochastic gradient descent.

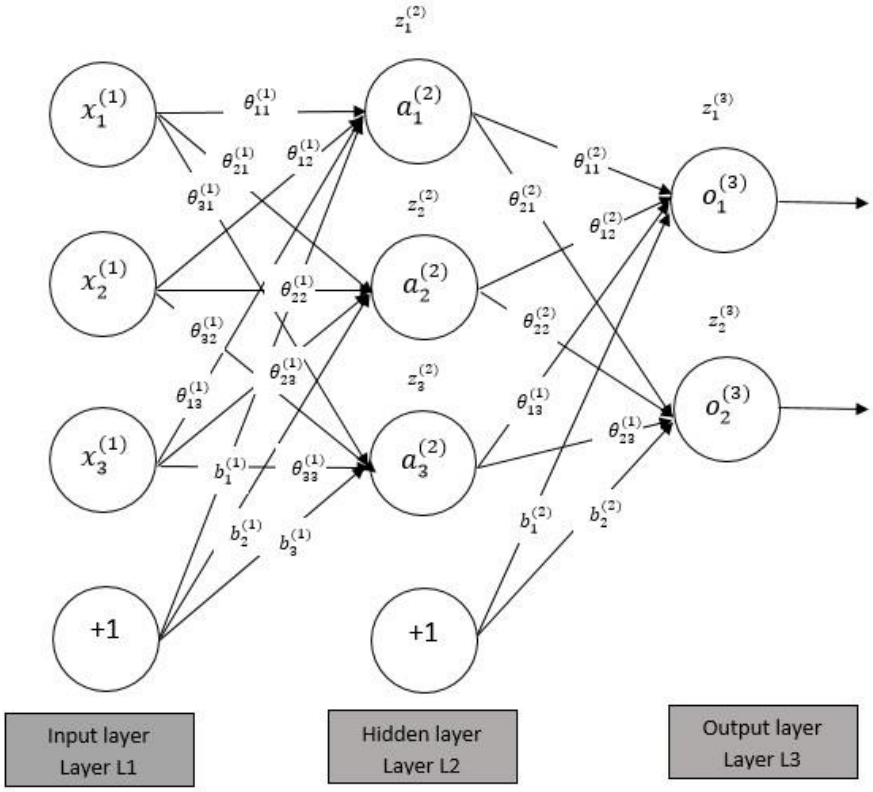

Figure 4.8 Backward propagation of the ANN.

Let's consider the above multilayer feed forward network. The impact of the change in the weight ($\theta_{11}^{(2)}$) on error occurred in the output layer is,

$$\frac{\partial E_1^{(3)}}{\partial \theta_{11}^{(2)}} = \frac{\partial E_1^{(3)}}{\partial o_1^{(3)}} \cdot \frac{\partial o_1^{(3)}}{\partial z_1^{(3)}} \cdot \frac{\partial z_1^{(3)}}{\partial \theta_{11}^{(2)}}$$

$$= \frac{\partial}{\partial o_1^{(3)}} \left[\frac{1}{2} \left(o_1^{(3)} - y^{(t)} \right)^2 \right] \frac{\partial}{\partial z_1^{(3)}} g\left(z_1^{(3)} \right) \frac{\partial}{\partial \theta_{11}^{(2)}} \left(\sum_{k=1}^{4} \theta_{1k}^{(2)} a_1^{(2)} + b_1^{(2)} \right)$$

$$= \left(o_1^{(3)} - y^{(t)} \right) \cdot o_1^{(3)} \left(1 - o_1^{(3)} \right) \cdot a_1^{(2)} \tag{8}$$

In general, the impact of the change in the weight ($\theta_{1j}^{(2)}$) on error $\left(E_1^{(3)}\right)$ in the output layer is,

$$\frac{\partial E_1^{(3)}}{\partial \theta_{1j}^{(2)}} = \left(o_1^{(3)} - y^{(t)} \right) \cdot o_1^{(3)} \left(1 - o_1^{(3)} \right) \cdot a_j^{(2)} \tag{9}$$

Similarly, the impact of the change in the weight ($\theta_{2j}^{(2)}$) on error ($E_2^{(3)}$) of the output layer is,

$$\frac{\partial E_2^{(3)}}{\partial \theta_{2j}^{(2)}} = \left(o_2^{(3)} - y^{(t)} \right) \cdot o_2^{(3)} \left(1 - o_2^{(3)} \right) \cdot a_j^{(2)} \tag{10}$$

Therefore, the impact of the change in the weight ($\theta_{kj}^{(2)}$) on error ($E_k^{(3)}$) of the output layer is,

$$\frac{\partial E_k^{(3)}}{\partial \theta_{kj}^{(2)}} = \left(o_k^{(3)} - y^{(t)} \right) \cdot o_k^{(3)} \left(1 - o_k^{(3)} \right) \cdot a_j^{(2)} \tag{11}$$

Let's compute the impact of the changes in weight ($b_i^{(2)}$) associated with bias value on error ($E_k^{(3)}$) occurred in output layer.

$$\frac{\partial E_k^{(3)}}{\partial b_i^{(2)}} = \left(o_k^{(3)} - y^{(t)} \right) \cdot o_k^{(3)} \left(1 - o_k^{(3)} \right) \tag{12}$$

According to the above equations (11) and (12), the update formulas for the weights associated with output layer are given below.

$$\theta_{kj}^{(2)} = \theta_{kj}^{(2)} - \alpha \left(o_k^{(3)} - y^{(t)} \right) \cdot o_k^{(3)} \left(1 - o_k^{(3)} \right) \cdot a_j^{(2)} \tag{13}$$

$$b_i^{(2)} = b_i^{(2)} - \alpha \left(o_k^{(3)} - y^{(t)} \right) \cdot o_k^{(3)} \left(1 - o_k^{(3)} \right) \tag{14}$$

In the above equations, α refers to the learning rate.

In the same way, we also calculate the impact of the changes in weights associated between input layer and hidden layer on the error ($E_1^{(3)}$). Let's start with $\theta_{11}^{(1)}$.

$$\frac{\partial E_1^{(3)}}{\partial \theta_{11}^{(1)}} = \frac{\partial E_1^{(3)}}{\partial o_1^{(3)}} \cdot \frac{\partial o_1^{(3)}}{\partial z_1^{(3)}} \cdot \frac{\partial z_1^{(3)}}{\partial a_1^{(2)}} \cdot \frac{\partial a_1^{(2)}}{\partial z_1^{(2)}} \cdot \frac{\partial z_1^{(2)}}{\partial \theta_{11}^{(1)}}$$

$$= \left(o_1^{(3)} - y^{(t)}\right) \cdot o_1^{(3)}\left(1 - o_1^{(3)}\right) \cdot \theta_{11}^{(2)} \cdot \quad a_1^{(2)}\left(1 - a_1^{(2)}\right) \cdot x_1^{(1)} \quad (15)$$

Consider that

$$\delta_1^{(3)} = \left(o_1^{(3)} - y^{(t)}\right) \cdot o_1^{(3)}\left(1 - o_1^{(3)}\right)$$

Then the equation (15) can be rewritten as

$$\frac{\partial E_1^{(3)}}{\partial \theta_{11}^{(1)}} = \delta_1^{(3)} \cdot a_1^{(2)}\left(1 - a_1^{(2)}\right) \cdot \theta_{11}^{(2)} \cdot x_1^{(1)} \quad (16)$$

Similarly, $\theta_{11}^{(1)}$ is required to be updated considering the error $E_2^{(3)}$ occurred in 2^{nd} unit of the output layer. Therefore, we can write

$$\frac{\partial E_2^{(3)}}{\partial \theta_{11}^{(1)}} = \delta_2^{(3)} \cdot a_1^{(2)}\left(1 - a_1^{(2)}\right) \cdot \theta_{21}^{(2)} \cdot x_1^{(1)} \quad (17)$$

In general, the impact of the changes in weights associated between input layer and hidden layer on the errors $(E_k^{(3)})$ occurred in the output layer can be written as follows,

$$\frac{\partial E_k^{(3)}}{\partial \theta_{ij}^{(1)}} = \delta_k^{(3)} \cdot a_i^{(2)} \cdot \left(1 - a_i^{(2)}\right) \cdot \theta_{ki}^{(2)} \cdot x_j^{(1)} \quad (18)$$

Similarly, the impact of the changes in weights ($b_i^{(1)}$) associated between bias value and hidden layer on error ($E_k^{(3)}$) occurred in output layer can be calculated.

$$\frac{\partial E_k^{(3)}}{\partial b_i^{(1)}} = \frac{\partial E_k^{(3)}}{\partial o_k^{(3)}} \cdot \frac{\partial o_k^{(3)}}{\partial z_k^{(3)}} \cdot \frac{\partial z_k^{(3)}}{\partial a_i^{(2)}} \cdot \frac{\partial a_i^{(2)}}{\partial z_i^{(2)}} \cdot \frac{\partial z_i^{(2)}}{\partial b_i^{(1)}}$$

$$\frac{\partial E_k^{(3)}}{\partial b_i^{(1)}} = \delta_k^{(3)} \cdot a_i^{(2)}\left(1 - a_i^{(2)}\right) \cdot \theta_{ki}^{(2)} \tag{19}$$

Therefore, the weights associated with input layer and the hidden layer can be updated using following equations.

$$\theta_{ij}^{(1)} = \theta_{ij}^{(1)} - \alpha\delta_k^{(3)} \cdot a_i^{(2)}\left(1 - a_i^{(2)}\right) \cdot \theta_{ki}^{(2)} \cdot x_j^{(1)} \tag{20}$$

$$b_i^{(1)} = b_i^{(1)} - \alpha\delta_k^{(3)} \cdot a_i^{(2)}\left(1 - a_i^{(2)}\right) \cdot \theta_{ki}^{(2)} \tag{21}$$

Table 4.1 Representation of the Symbols

Symbol	Description
l	No. of the layer of neural network
i, j	Index of the neurons input and hidden layer
k	Index of the neuron of the output layer
x_i	ith training example
y_i	Corresponding actual output of ith training example
θ_{ij}^{l}	The weight value going to the unit i in layer $(l+1)$ and coming from the unit j in layer l.
$Z_i^{(2)}$	Summed output of the i^{th} neuron in the layer 2.
$a_i^{(2)}$	The net output of the i^{th} neuron in the layer l.
$O_k^{(3)}$	Output of the k^{th} neuron in the output layer.
$E_k^{(3)}$	Error occurred at the k^{th} neuron of the output layer while estimating the actual output.
b_i^1	The bias value going to the unit i in layer 2 and coming from the layer 1.
α	Learning rate

4.2. Support Vector Machine (SVM)

The support vector machine (SVM) is one of the reliable and widely used supervised learning model that analyze the presented data samples to perform classification and regression analysis. The SVM learns the given data samples, each sample marked with a specific class, builds a model that can assign a class to a new data sample [54]. SVM learning model set a hyperplane in an optimum position in the data space so that Euclidian distance from the decision surface for all training data samples would be maximized. SVM has been emerged to provide the generalized performance to solve a wide range of classification and pattern recognition problems such as face detection, pedestrian detection, and text categorization.

Let's consider a training data set, $D = \{(x^m, y^m)\}$ contains m training data samples where $x^m \in R^n$ and $y^m \in \{-1, +1\}$, represents the label of the m^{th} data sample. The hyperplane in form of decision surface can be defined as

$$\sum_{i=1}^{m} \boldsymbol{w}^T x^{(i)} + b = 0 \tag{22}$$

Where $\boldsymbol{w}$ and b represent a weight vector and a bias term that can be determined through the training process. Through these parameters (w, b), the decision surface places itself in the optimum position in the data space. As we know, the SVM places the decision surfaces in the data space in such way that it maximizes the geometric margin of all training data samples. In this circumstance, the optimization problem is

$$\min_{w,b} \frac{1}{2} \| W \|^2 \tag{23}$$

Subject to

$$\sum_{i=1}^{m} y^{(i)}(W^T x^{(i)} + b) - 1 \geq 0$$

The concept of the Lagrange multiplier is implemented to solve this optimization problem with constraint boundary stated in the equation (23). Therefore, the Lagrange function is

$$L(w, b, \alpha) = \frac{1}{2} \| W \|^2 - \sum_{i=1}^{m} \alpha(y^{(i)}(W^T x^{(i)} + b) - 1) \quad (24)$$

Where α is the multiplication factor and $\alpha \geq 0$. If we differentiate the Lagrange function with respect to w, b and, α; then the optimization problem mention in the equation (20) can be formulated as

$$\max_{\alpha} L(\alpha) = \max_{\alpha} (\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y^{(i)} y^{(j)} x^{(i)} x^{(j)}) \quad (25)$$

Subject to

$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0$$

$$\alpha_i \geq 0, i = 1,2, \ldots \ldots, m$$

The solution of the equation (25) drives to get optimum decision surface that is able to separate the positive and negative training data samples. If the presented data samples are not linearly separable, then the non-linear kernel trick can be implemented to deal with this problem. The Gaussian kernel is widely used as the kernel function. Because this function can increase the dimension of the data samples infinitely [54]. The expression of the Gaussian kernel is given as follows.

$$K\left(x^{(i)}, x^{(j)}\right) = \exp\left(\frac{-\|x^{(i)} - x^{(j)}\|^2}{2\sigma^2}\right) \quad (26)$$

4.3. Non-linear Logistic regression (LR)

Logistic regression is a statistical model in which the category, from a predefined list of categories, of a new observation or data sample, is predicted, by estimating the probability of the class using a logistic function (e.g. email sorting) [55]. It is evolved from linear regression and designed to solve classification problems. That's why, in some literature, it is referred as logistic classification.

Logistic classification can be shown as a special case of linear regression, but we can draw a distinct line between these two statistical models. In the linear regression, the predictor predicts the continuous value by a fitting curve to the given training input data samples [56]. In contrary, in logistic classification, a predictor learns how to classify the data through a training phase and predicts a discrete value for the corresponding new data sample. Like linear regression, the same decision boundary equation is used for logistic classification.

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 {x_2}^2 + \cdots + \theta_n {x_n}^n = \theta^T \mathbf{x} \quad (27)$$

Here, $\boldsymbol{x}$ represents a given input vector that contains n input features. The θ refers to the model parameter that is required to be optimized by using the training input data samples. This decision boundary can be either linear or nonlinear that depends on the application of the logistic classification.

$$Z = \theta^T \boldsymbol{x} \quad (28)$$

$$g(z) = \frac{1}{1+e^{-z}} \quad (29)$$

In logistic regression, a sigmoid function is used to measure the certainty level of the class that new observation or data sample belongs to. The main advantage of using sigmoid function is that it is continuous, differentiable at any point and monotonically increasing. The probabilistic interpretation can be done in a simple way; as the value of the $g(z)$ in greater than 0.5, the predictor would be more certain about the class of the given data sample [57]. If the same cost function of linear regression is used, $J(\theta)$ yields non-convex cost function. Consequently, special kind of the cost function is used in logistic classification to learn the model from the training data which is convex.

$$J(\theta) = -\left[\frac{1}{m}\sum_{i=1}^{m} y^i \log(\, h_\theta(x^i)) + (1 - y^i)\log(1 - h_\theta(x^i))\right] + \frac{\lambda}{m}\sum_{j=1}^{n} \theta_j^2 \quad (30)$$

$J(\theta)$ denotes the cost function of the logistic classification with regularization term, where m refers to the total training examples of a given training set, λ is the regularization parameter, y^i represents the target output for i^{th} training example, and x^i represents the i^{th} training example. There are two learning methods used in the training procedures: Gradient decent and Newton's method. In this research, the Newton's method is applied as the learning model, because it converges to the optimal solution within few iterations.

Chapter 5

5. Proposed algorithm

The proposed algorithm is a network based approach, in which neighborhood counts are used as the detection feature to detect a wormhole attack.

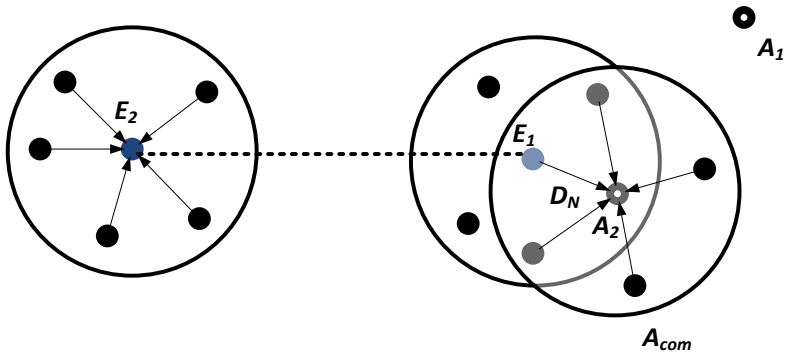

Figure 5.1 Impact of wormhole attack on neighborhood count.

A mobile sensor node, known as detector node (D_N), is deployed in an area where sensor nodes could be uniformly or non-uniformly distributed. The detector node D_N moves around this sensor field and collects a neighborhood count and the coordinate at each site it visited. When it reaches to the wormhole infected zone, the neighborhood population would increase abnormally sharply or abnormally lightly, depending on the sensor distribution and the position of the wormhole nodes. For instance, Figure 5.1 shows the impact of wormhole nodes on the neighborhood counts. Let's say, the detector node D_N is moving spontaneously around the area where the sensor nodes are deployed. At the time t_1, D_N moves from a location A_1 to another location A_2. Since the detector node collects the neighborhood count at each site, it transmits neighbor discovery message (NDM) to the adjacent sensor nodes within its communication range. According to the Figure 5.1, the wormhole node (E_1) also hears the broadcast as it is located in the

transmission range of D_N. Since wormhole node is not able to read the content of the packet for a while, it encapsulates and forwards the packets along the virtual tunnel to another malicious node (E_2). Furthermore, E_2 retransmits the received packets towards its neighbors. The neighbors of the distant malicious nodes (E_2) respond back with valid neighbor ID (NID) through the wormhole link. After that, E_1 unicasts the received responses to the originator, D_N. E_1 compels D_N to think that the responses have come from the sensors located within its radio ranges. Though some of those responses have traveled long distant within the network. That's how the neighborhood counts from the infected zone increases sharply due to the wormhole node. This may be true for uniform sensor distribution. As we know, the sensor density in the non-uniform sensor distribution is inconsistent over the area. If the wormhole nodes are placed in the sparsely populated area, then the number of neighbors would not increase abruptly as expected. In this circumstances, it would be difficult for ANN based detection scheme to detect wormhole attack based on the collected neighborhood counts (also true for the sparse network). Sometimes the neighborhood counts taken from the infected zone are the same as, or far smaller than the counts taken from the non-infected zone. In this case, the ANN based detector may suffer to distinguish between positive and negative data samples. Therefore, another detection attributes along with neighborhood count is required in order to detect wormhole attack from the infected network (either uniform or non-uniform) more precisely and confidently.

In this research work, we introduce a new detection feature called 'average *residual energy pattern of the neighbors (AREPN)'*. As we know, wormhole nodes circulate false route information into the network that the base station is multiple hops away from the wormhole nodes. Therefore adjacent sensor nodes of the wormhole get influenced and transmit their data packets to the base station through wormhole nodes. However, the wormhole nodes force them to hand over the

collected data packets to one of the wormhole nodes. That means the wormhole nodes receive more data packets after the base station and its neighbors. Now, the question is how the wormhole nodes get those data packets coming from the adjacent nodes. Apparently, the data packets are coming through its neighbors.

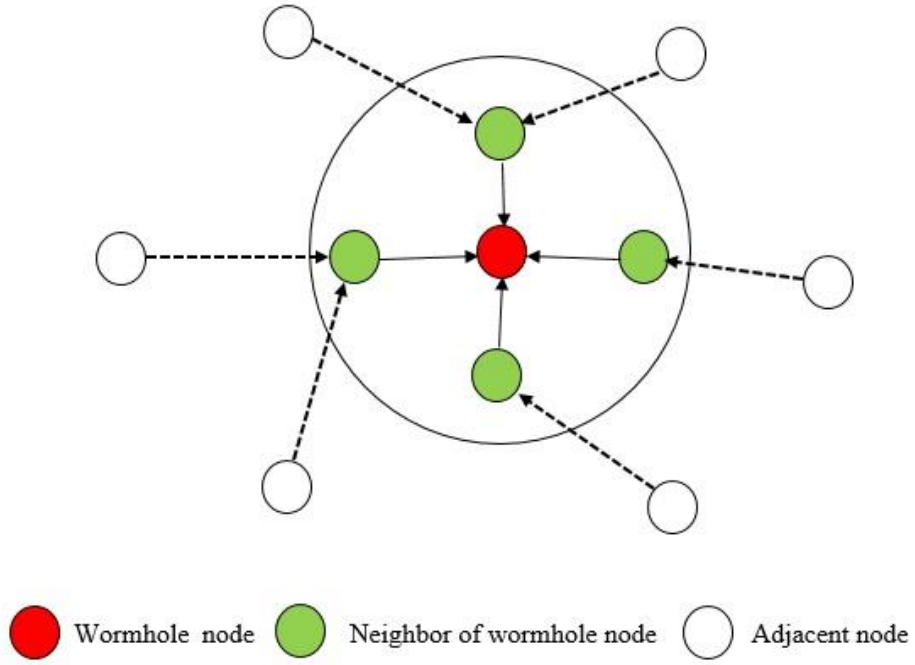

Figure 5.2 Impact of wormhole nodes on adjacent nodes.

More packets arrive at the wormhole node means, more packets have been received and transferred by its neighbors. As we know, a node dissipates a significant amount of energy due to the transmission and reception of the data packets. Since the wormhole nodes are getting the numerous data packets from the adjacent nodes, so the neighbors of the wormhole nodes are also losing energy more quickly by retransmitting packets intended for the wormhole node. Now it can be said that the sensors located in the infected zone lose more energy than other nodes located in the non-infected zone.

In our proposed scheme, the detector node broadcasts NDM to the adjacent sensor nodes at each site visited. The adjacent nodes reply back by transmitting a data packet incorporating valid NID and information of residual energy. In this way, D_N is able to calculate the number of neighbors and average residual energy of the neighbors (AREPN). D_N captures this evidences as a two-featured sample and stores it. It is expected that AREPN taken from the infected zone is much smaller than the AREPN taken from the non-infected zone. As we know, the adjacent nodes of the base station dissipate energy quicker than the normal node. That's why, D_N doesn't cover a certain geographic area based on the location of the base station. Once the mobile node reaches close to the base station, it transfers all the collected samples for further analysis.

The efficacy of an artificial neural network highly depends on the method of training and the dataset containing potential attributes. The base station gathers (with the help of the detector node) a data set (D_{set}) that consists of two featured N data samples. In this first half, D_N gathers two featured K data samples ($K \in N$) from the non-infected zone which are called negative data samples. Similarly, the same amount of data samples is collected from the wormhole infected zone, known as positive data samples. After that, those two types of data samples are mixed up randomly so that training can be performed without any bias. Then, M data samples ($M \in N$) are drawn from the D_{set} and stores in a training dataset D_{train} ($D_{train} \subset D_{set}$). At the same time, rest of the P data samples ($P \in N$) are stored in D_{test} ($D_{test} \subset D_{set}$) to evaluate the learning performance of the trained neural network.

Proposed Algorithm

1. Collect two featured K negative data samples from the non-infected zone
2. Collect two featured K positive data samples from the wormhole infected zone
3. Store the both type of data samples in D_{set} which consists of two featured N data samples
4. Select two featured M data samples from D_{set} and store in D_{train} where $(D_{train} \subset D_{set})$
5. Rest of the P data samples are stored in D_{test} where $(D_{test} \subset D_{set})$
6. Train the neural network with the data set D_{train} and appropriate network parameters up to T epochs
7. Test the neural network by using the samples of D_{test}
8. If *output* for a specific sample ≥ 0.5, then this sample represents wormhole attack
9. If *output* for a specific sample < 0.5, then this sample doesn't represent wormhole attack and probable location of the malicious node are identified.
10. Update D_{train} by D_{test} for further training
11. Reset D_{test} and update with new data samples gathered by D_N

Furthermore, data samples of D_{train} are fed into the input layer of the ANN. The training procedure is performed repeatedly until it reaches a predefined maximum number of training cycles i.e. T epochs. The testing procedure involves checking the learning progress of the ANN-based detector. In the testing part, each data sample passes through the trained neural network. If the output for a specific data sample is greater than 0.5, then this sample represents wormhole attack. Since, D_N stores the coordinate of the locations along with the two featured data samples,

so the probable position of the wormhole node can also be identified by the detection scheme. After that, D_{train} is updated by the D_{test} for further training. This would minimize the error level that has achieved in the training phase. The D_{test} entries are cleared up and updated with new data samples collected by the D_N in real time.

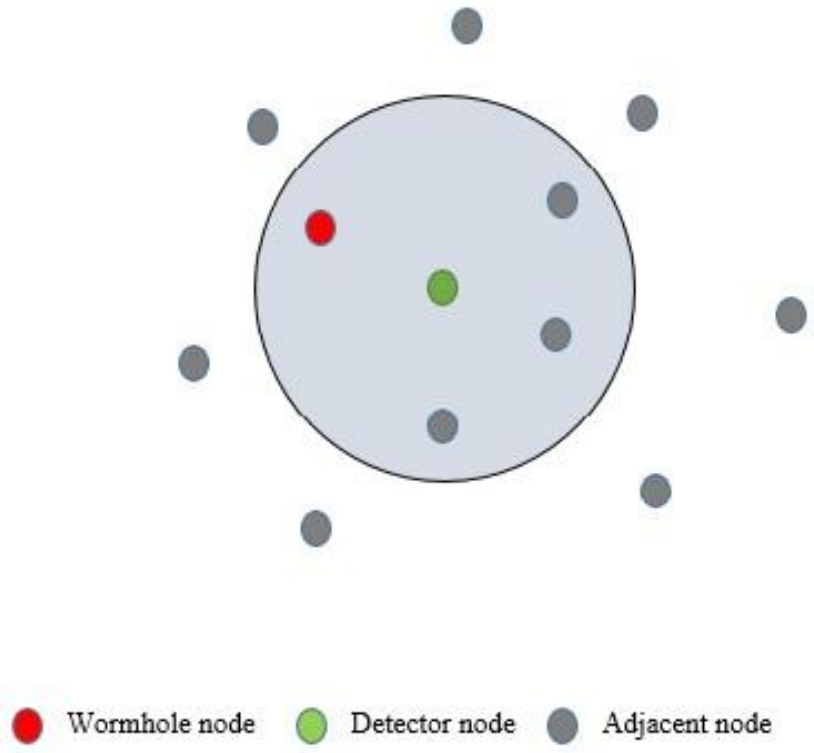

Figure 5.3 Probable location of a wormhole node.

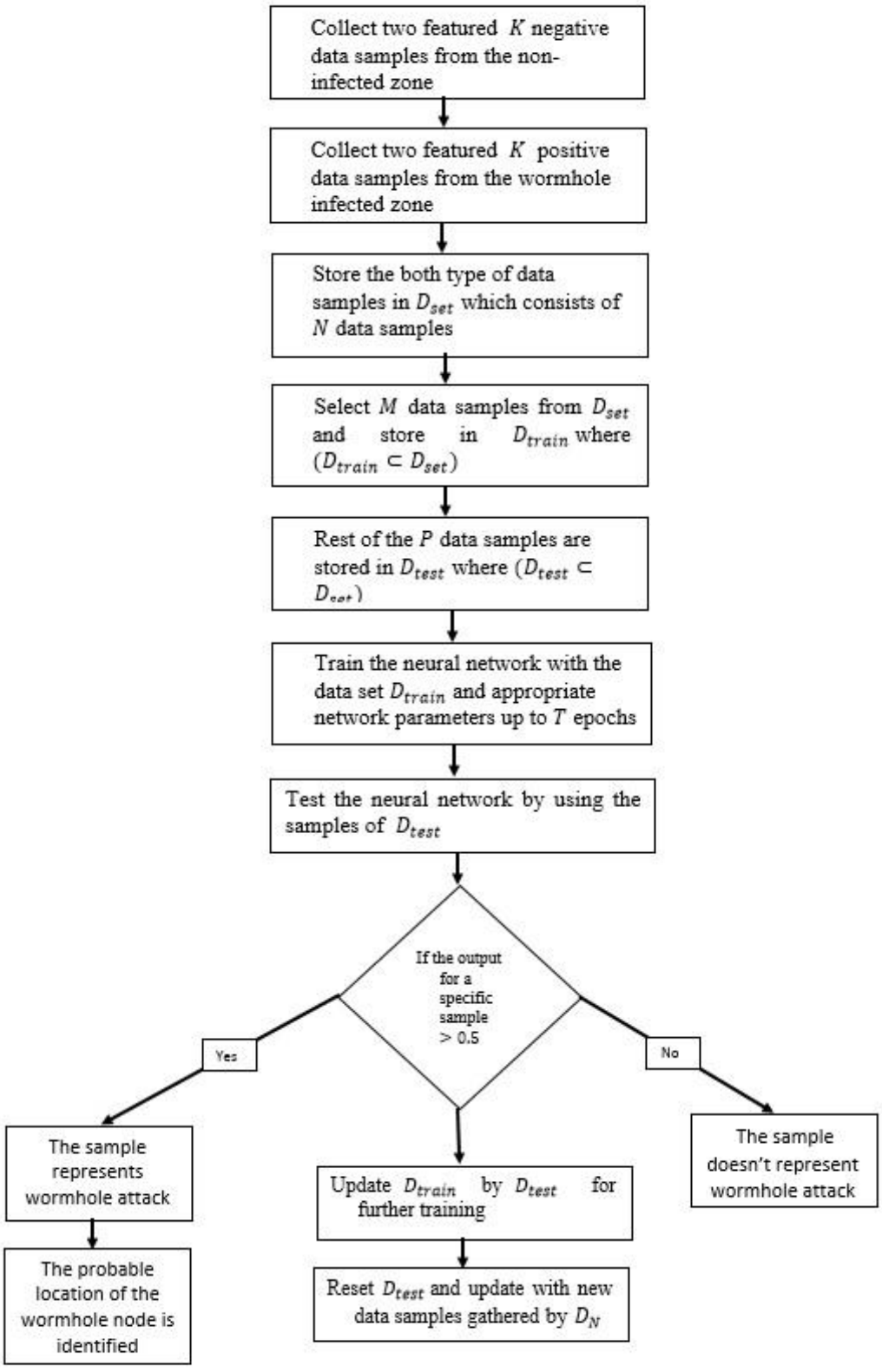

Figure 5.4 Flowchart of the ANN-Based proposed algorithm.

Chapter 6

6. Simulation and results

In this section, extensive research experiments are conducted under various network scenarios in order to assess the effectiveness of the proposed algorithm in detecting wormhole attack from the affected sensor network. The first phase of the experiments is conducted to see if the proposed scheme is able to classify the malicious data samples that represent wormhole attack. Furthermore, we evaluate the performance of the detection scheme in terms of detection accuracy, false positive rates, and false negative rates. In the second phase, the efficacy of the proposed algorithm is explored considering single featured data samples and two featured (i.e. number of neighbors and AREPN) data samples. In the literature, it is mentioned that wormhole attack can invade pro-active (DSR, DSDV) and reactive (on-demand base routing protocol) routing protocols of WSN. What's impact of wormhole attack on cluster based routing protocol like LEACH- It is not explored in the previous researches. AODV and LEACH are the most widely used routing protocols for WSN and the energy dissipation of the sensor node depends on the routing protocol; hence, the proposed algorithm is also tested considering 'On demand' based routing protocol (AODV) and cluster based routing protocol (LEACH). Afterward, we record and analyze the efficacy of the proposed algorithm in detecting wormhole attack from different sensor distributions. Finally, the performance of the proposed algorithm is compared with the performance of other machine learning technique based detection schemes like support vector machine (SVM) and regularized non-linear logistic regression (LR). In addition, all experiments have been performed in MATLAB.

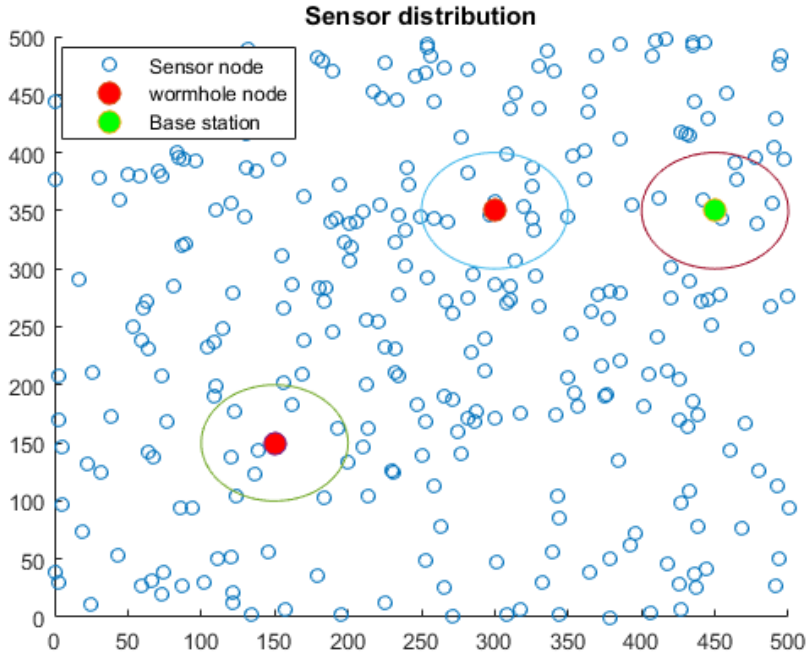

Figure 6.1 The depiction of the simulation set up (uniform sensor distribution).

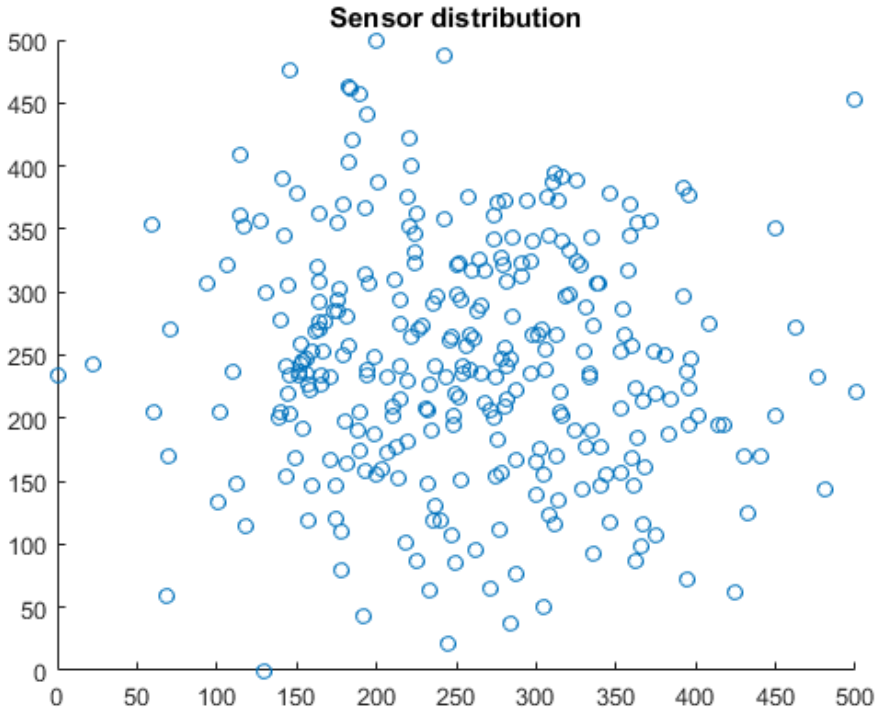

Figure 6.2 Non- uniform sensor distribution (Gaussian).

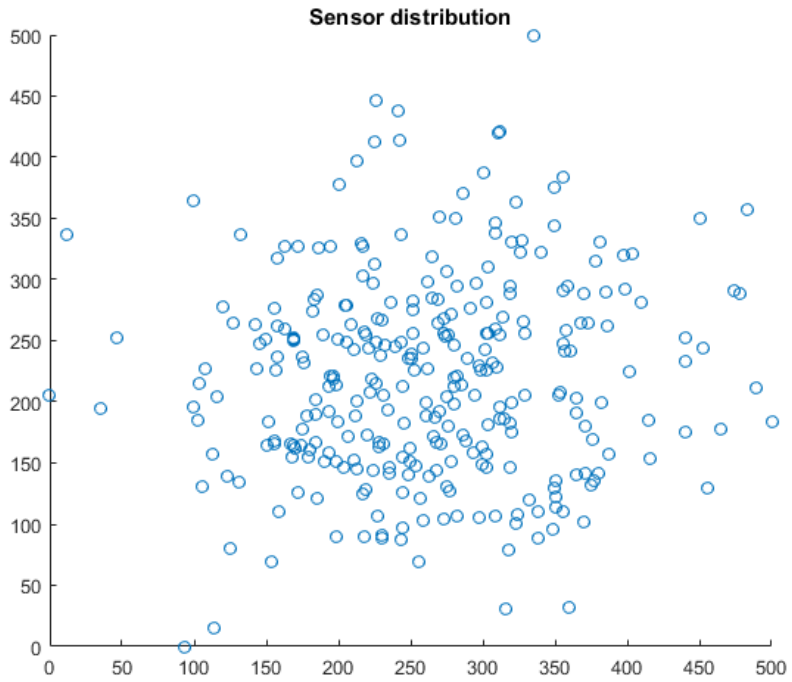

Figure 6.3 Non- uniform sensor distribution (poisson).

In the simulation, 300 sensor nodes are distributed (uniform or non-uniform) within the square field of $500m \times 500m$. The deployed sensor nodes and the base station are static in nature. Radio range of each sensor node equals to $50\ m$. Initially, each sensor node has five (05) units of energy that would be used in sensing and transferring packets. A detector mobile sensor node, D_N is deployed as a mobile observation point of the network. The basic task of the D_N is to collect neighborhood counts and AREPN of neighboring population at each site it visited within this deployed area. The radio range of the detector node is same as deployed sensor nodes. We assume that the detector node is fully aware of its position, the boundary of the targeted area, and any obstacles in the area that may restrain its movement. A pair of wormhole nodes is placed at the locations $150m \times 150m$, and $300m \times 350m$. Random waypoint model is implemented in the simulation as the mobility model for the D_N. Similarly, the base station is positioned at a location

of $350m \times 450m$ .In these experiments, we only consider the data transmission between a node and the base station. Moreover, the experiments have been conducted on thirty (30) different instances for each sensor distribution (e.g. uniform sensor distribution, Gaussian sensor distribution, Poisson sensor distribution, Gamma and Beta Sensor distribution) in order to get valid (average) performance measures of the proposed detection scheme.

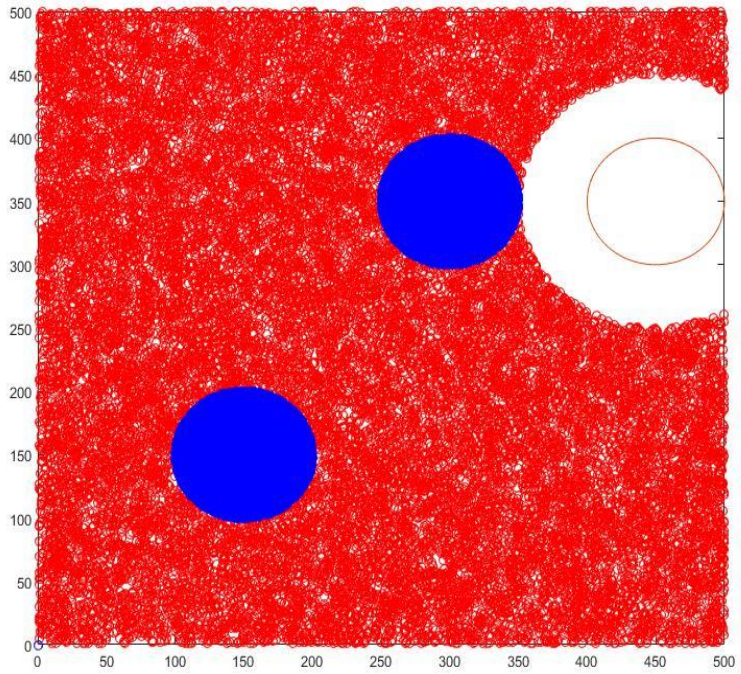

Figure 6.4 The location visited by the D_N (red ,blue and white indicate non-infected zone, infected zone and the area not covered by D_N).

In this simulation, the detector node collects two featured data sample (i.e. number of neighbors and AREPN) at each site it visited. For each instance of each sensor distribution (uniform or non-uniform sensor distribution), the detector node, D_N collects 50000 data samples from deployed area in which 25000 samples are negative data samples, and 25000 samples are positive data samples. The collected data samples are stored at the D_{set} in the base station. After that, 49000

randomly selected samples are stored in the D_{train} for the training purpose. Rest of the data samples is used to test the learning performance of the proposed detection scheme.

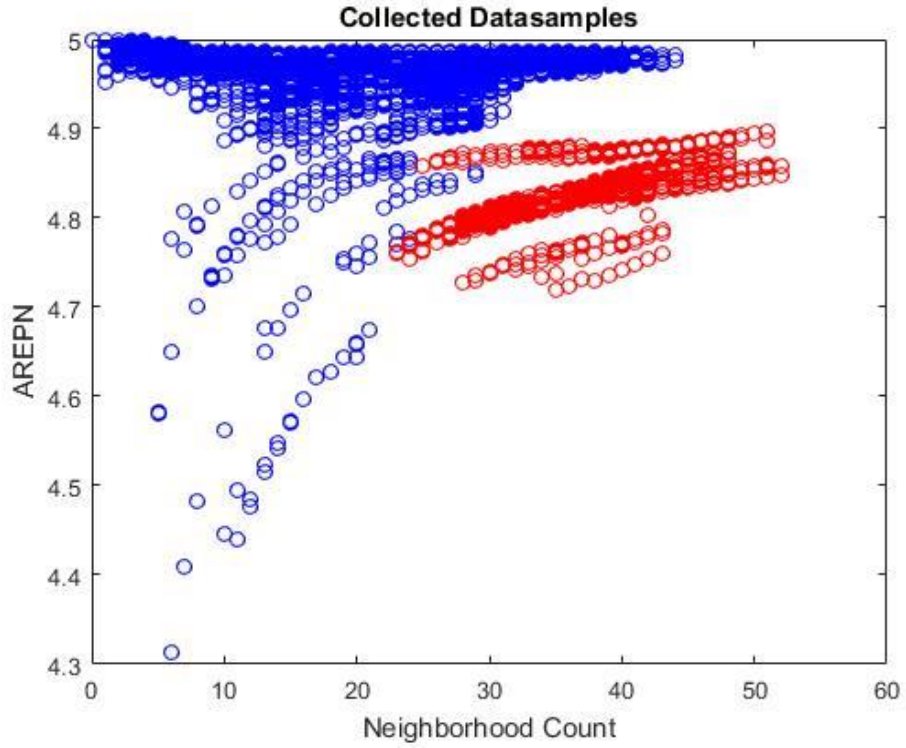

Figure 6.5 Collected two featured data samples (poisson sensor distribution).

Table 6.1 The network parameters of the Simulation setup

Network Parameters	**Value**
Network Area	$500m \times 500m$
No. of the sensor nodes	300
Radio range	$50m$
Nature of the sensor nodes	Static
Position of the base station	$350m \times 450m$
Position of the first wormhole node	$150m \times 150m$
Position of the second wormhole node	$300m \times 350m$
Sensor distribution	Uniform or Non-uniform

6.1. The performance of the proposed ANN-based detection scheme

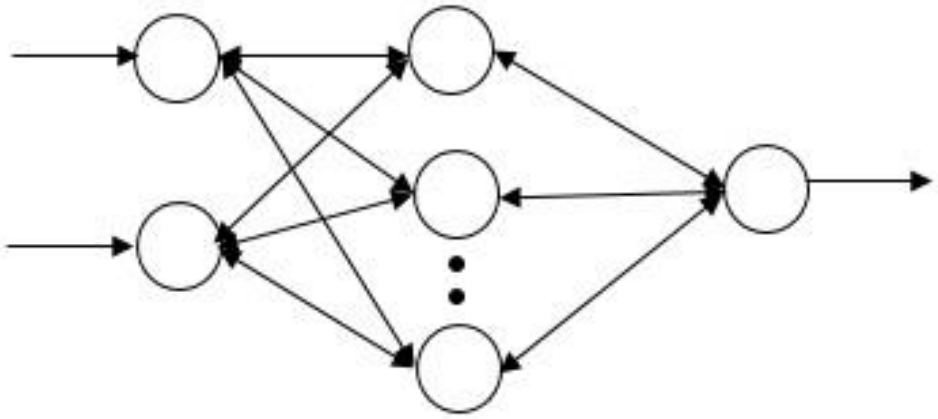

Figure 6.6 The structure of the ANN for two featured data samples.

A multi-layer perceptron (with back propagation algorithm) is implemented for the experiments. The input layer contains one or two neurons considering the type of data samples (one featured or two featured) used to train the network. Moreover, the hidden layer consists of 100 neurons and the output layer has only one (01) neuron. We used a sub data set, D_{train} comprised of 49000 randomly selected data samples from D_{set} for training collected by D_N. At first, the neural network was trained by the single featured training examples from D_{train} and evaluated learning performance of ANN by the rest of the data samples stored in the D_{test} . After that, two featured training samples are fed into the neural network and acquired results are compared with the results obtained by using single feature data samples. During the training period, the minimum error tolerance level is set to 10^{-5}. The Table 6.2 represents the parameters which are used during the training phase.

Table 6.2 Parameters used for ANN.

Parameter	Value
No of attributes	2
No of Data points (training)	49000
No of Data points (testing)	1000
Architecture (for one feature)	[1,100,1]
Architecture (for two features)	[2,100,1]
Minimum gradient	0.00001
Learning rate, α	0.001
Epoch	500
CPU time (for one instance)	2.43 mins

In the testing phase, the test data set, D_{test} is fed into the input layer. Then the output of the ANN based model is observed if it could identify the existence of malicious node in the given network. Figure 6.7 shows that, the ANN based wormhole attack detection scheme can classify the samples that represent wormhole attack from the both single featured and two featured data samples successfully.

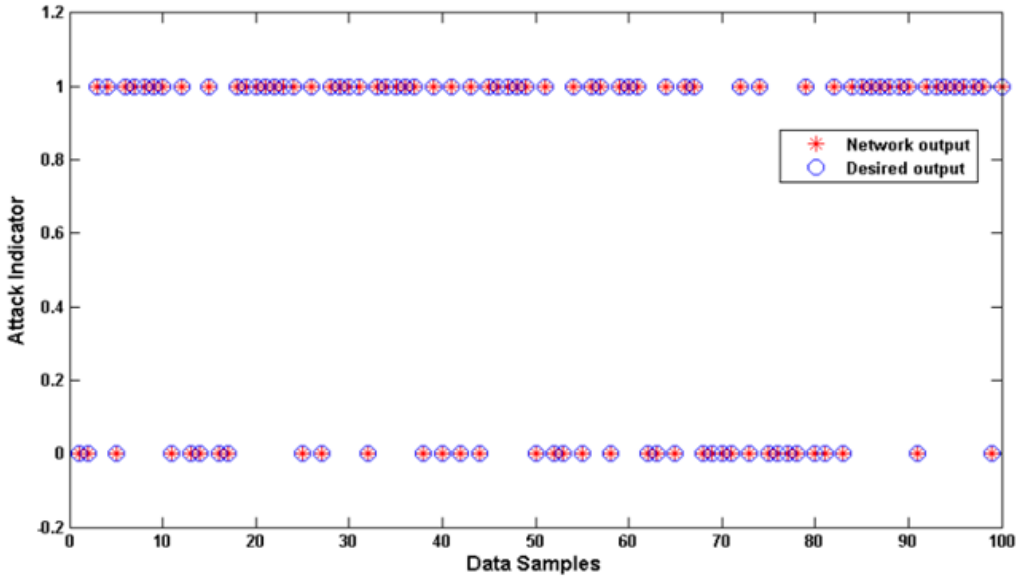

Figure 6.7 Classification of wormhole attack (for single feature).

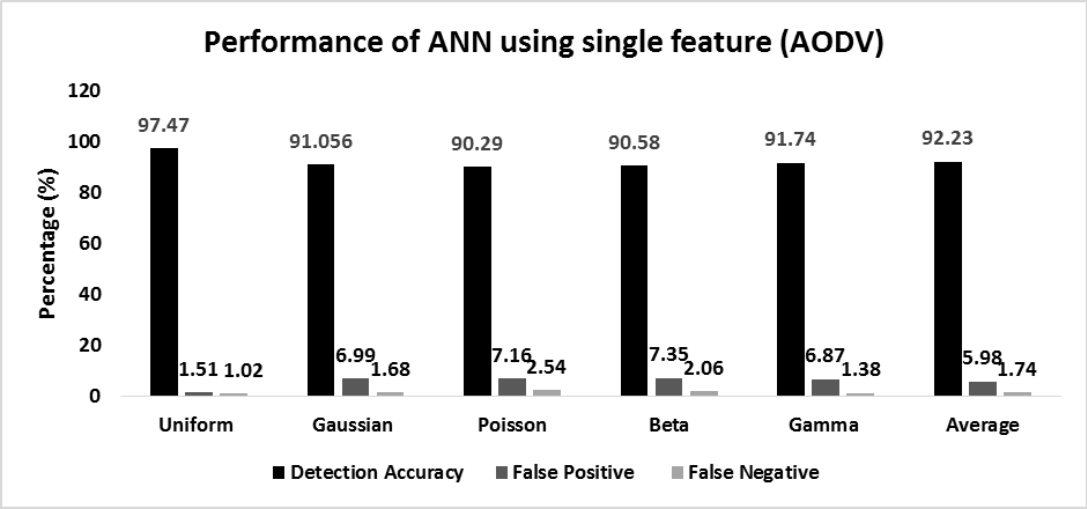

Figure 6.8 Performance of the ANN-based detection scheme using single featured data samples (AODV).

Figure 6.8 shows the performance of the ANN-based detection model in detecting wormhole attack using single featured data samples (i.e. neighborhood count) when the network uses AODV routing protocol. In this graph, the highest detection accuracy is recorded as 97.47% when sensors are distributed uniformly in the square field, whereas the lowest detection accuracy is measured 90.29% for poisson sensor distribution. Furthermore, detection accuracy for Gaussian distribution is almost same as the gamma distribution. Accordingly, 91.056%, 91.74%, and 97.873% detection rates are measured for Gaussian, Gamma, and Beta sensor distribution. However, the average detection accuracy is calculated as 92.23%. The false positive rate and false negative rate entirely follow the same trend of detection accuracy. The lowest false positive rates and false negative rates are measured for uniform sensor distribution. The average false positive rates and false negative rate are accordingly 5.98% and 1.74%.

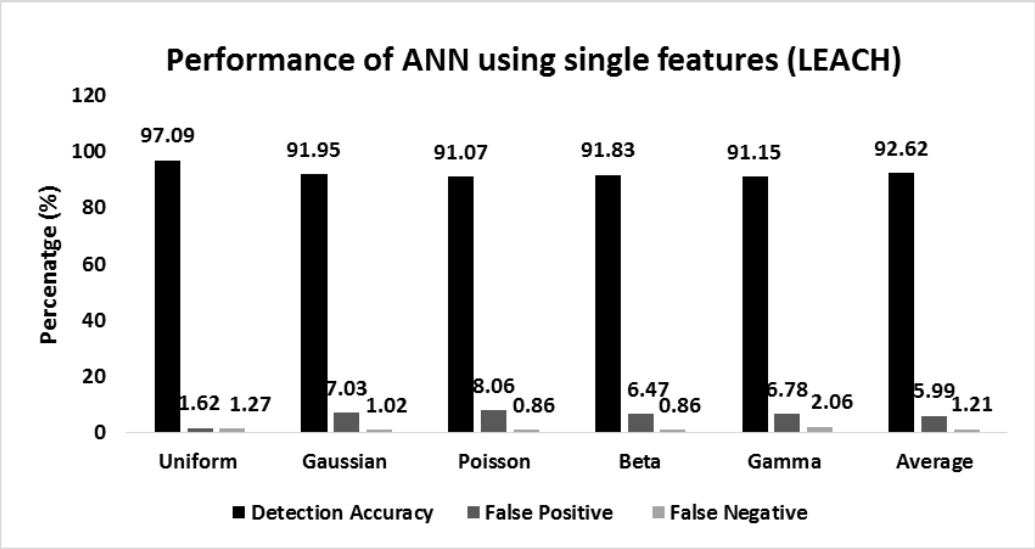

Figure 6.9 Performance of the ANN-based detection scheme using single featured data samples (LEACH).

Figure 6.9 shows the results obtained by applying single featured data samples on ANN based detection model considering LEACH routing protocol. It is observed that this algorithm performs better for uniform sensor distribution as compared to non-uniform sensor distribution. 97.63% detection accuracy, 1.62% false positive rate, and 1.23% false negative rate are achieved for the uniform sensor distribution. After that, it has performed better for Gaussian sensor distribution among all non-uniform sensor distributions. The detection accuracy, false positive rate and false negative rate for Gaussian sensor distribution are 91.95%, 7.03% and 1.02% accordingly. The similar trend in detection accuracy is observed for Poisson, beta, and gamma sensor distribution. However, considering all the sensor distributions, the false negative rates are lower than the false positive rates. The average detection accuracy and false positive rate for all sensor distributions are accordingly 92.62% and 5.99%.

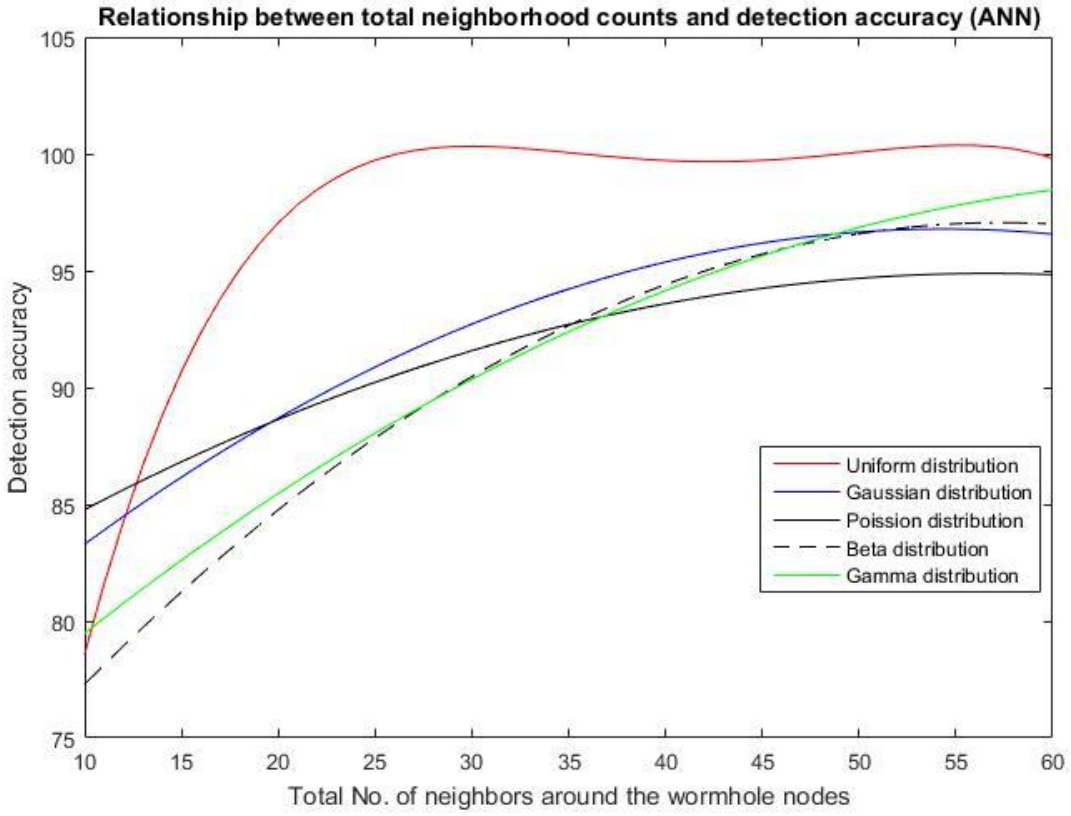

Figure 6.10 The relationship between neighborhood count and detection accuracy (ANN).

Figure 6.10 represents the relationship between the neighborhood counts and the detection accuracy. Undeniably, the detection accuracy of the proposed algorithm (considering single featured data samples) increases as the total number of neighbors around wormhole nodes increase for all sensor distributions. The detection accuracy of the system has a non-linear relationship with the total number of the neighbors around the wormhole nodes. For the uniform sensor distribution, we can model this relationship as 4^{th} degree polynomial. For the other non-uniform sensor distribution, the relationship can be modelled as 2^{nd} degree polynomial. It means that the proposed scheme may not perform well considering single featured data samples if the wormhole nodes are located in the sparsely populated area in the network.

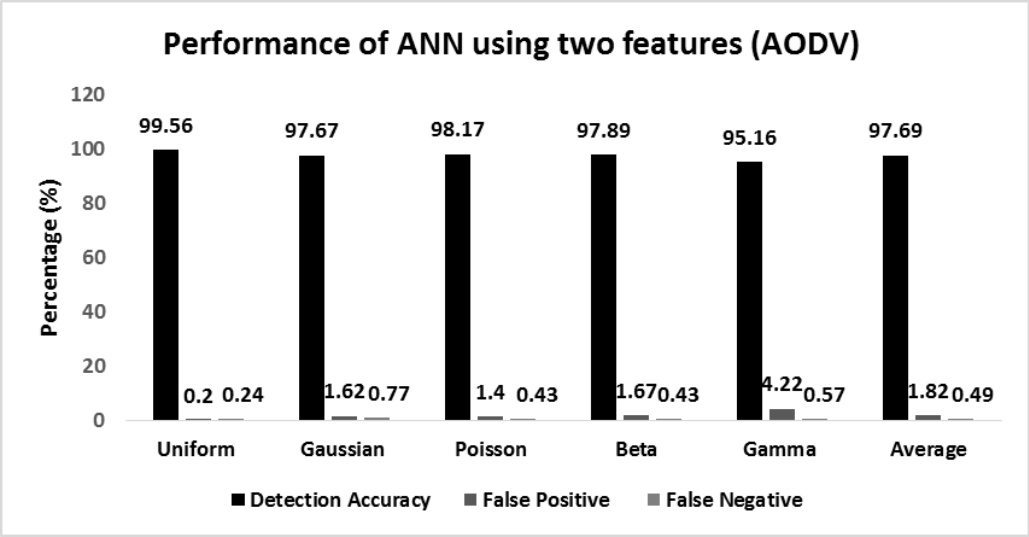

Figure 6.11 Performance of the ANN-based detection scheme using two featured data samples (AODV).

The two featured data samples are applied to the ANN-based detection scheme. Like prior, the performance of the ANN is evaluated considering both AODV and LEACH routing protocol. Figure 6.11 presents the performance of the ANN-based model considering two featured samples and AODV routing protocol. The proposed algorithm gives a better performance to detect the wormhole attack for the uniform sensor distribution. 99.56% detection accuracy, 0.2% false positive rates, and 0.24 % false negative rate are achieved for the uniform sensor distribution. For all non-uniform sensor distribution, the detection accuracy varies in between 95% to 97%. In this circumstance, the average detection accuracy and false positive rate for all the sensor distributions are 97.69% and 1.82%. Most importantly, the detection accuracy, considering all sensor distribution, is significantly increased for the application of the two featured data samples.

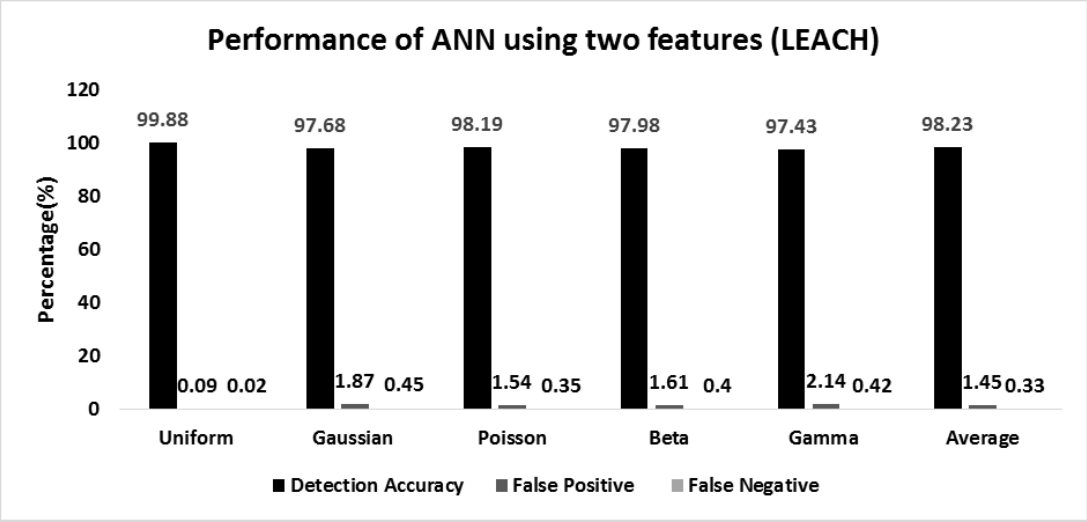

Figure 6.12 Performance of the ANN-based detection scheme using two featured data samples (LEACH).

As shown in the Figure 6.12, the performance of the ANN-based detection scheme improves significantly considering LEACH routing protocol and the application of two featured data samples on the ANN. Evidently, the ANN-based model performs better for the uniform sensor distribution as compared to all non-uniform sensor distribution. The highest achieved detection accuracy for the uniform sensor distribution is 99.88%. The detection accuracy for the non-uniform sensor distributions fluctuates from 97.43% to 97.98%. Furthermore, the lowest false positive rates and false negative rates are achieved for the poisson sensor distribution which is 1.54% and 0.35% respectively. In this circumstances, the average detection accuracy, false positive rate, and the false negative rate for all the sensor distribution are accordingly 98.23%, 0.45%, and 0.33%. In summary, the application of the two featured data samples on the ANN enhances the performance of the proposed scheme for both AODV and LEACH protocol.

6.2. The performance of the proposed SVM based detection scheme

For the different sensor distributions, one and two featured training examples are applied to the SVM accordingly. The average result of detection accuracy, false positive rates, and false

negative rates are calculated to measure the efficacy of the SVM-based algorithm for different network scenarios. Table 6.3 represents the parameters which are used during the training phase.

Table 6.3 Parameters used for SVM

Parameter	Value
No of attributes	2
No of Data samples (training)	49000
No of Data samples (testing)	1000
Kernel	Gaussian
Sigma	default
Tool	MATLAB
CPU time (for one instance)	2.39 mins

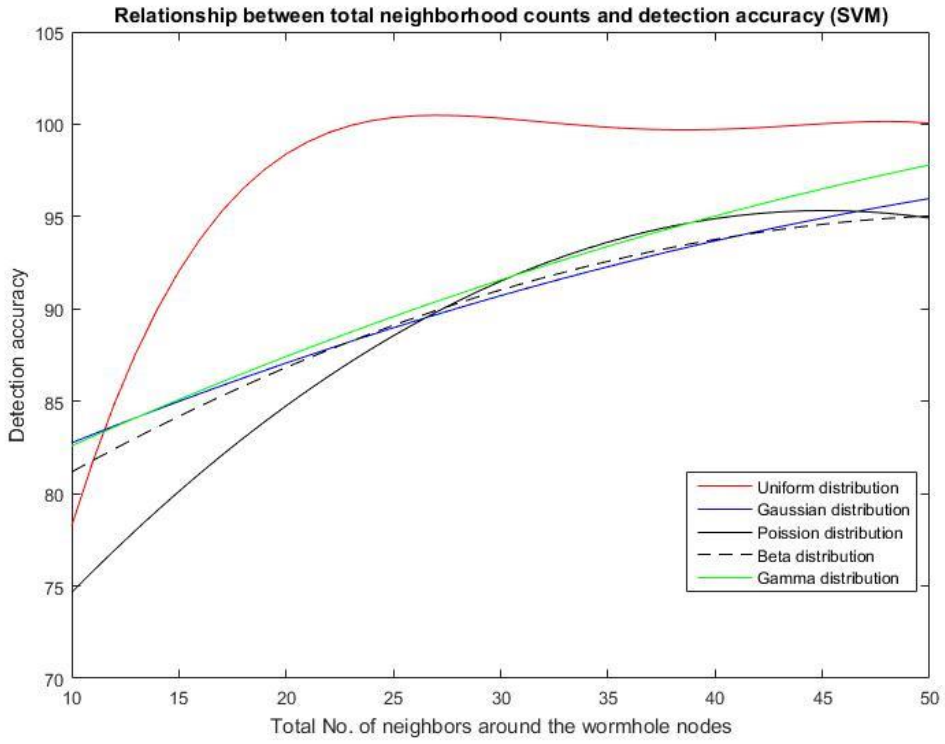

Figure 6.13 The relationship between the neighborhood count and the detection accuracy (SVM).

Figure 6.13 represents the relationship between the counts of neighbors around wormhole nodes and the detection accuracy considering one featured data samples applied to the SVM-based detection scheme. Like the previous case, the detection accuracy follows the non-linear relationship with the total number of neighbors for all sensor distribution. As the neighborhood count increases, the detection accuracy of the SVM-based algorithm also increases. Similar to the ANN-based detection scheme, it may suffer to detect wormhole attack if the wormhole nodes are placed in the sparsely populated area.

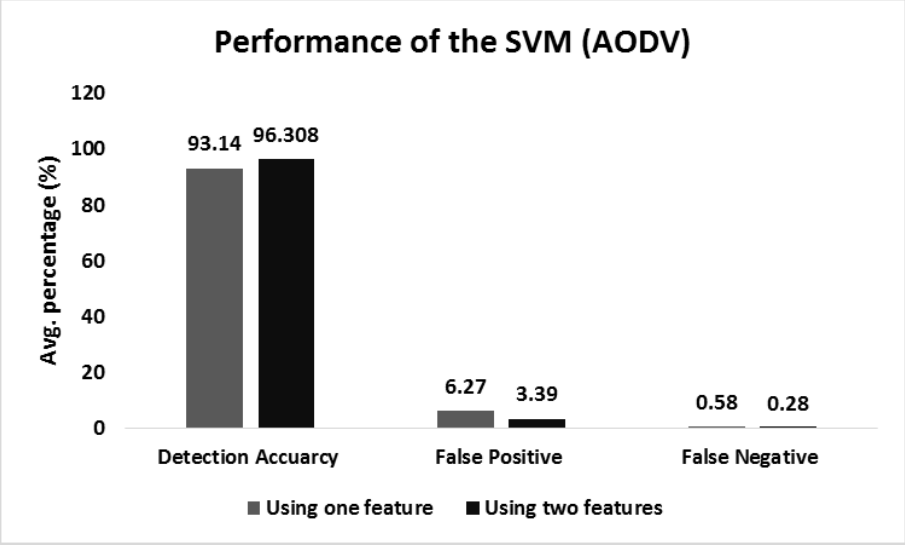

Figure 6.14 Performance of the SVM-based detection scheme (AODV)

Figure 6.14 shows that the performance of the SVM-based algorithm when the network uses AODV as the routing protocol. According to the Figure 6.14, SVM-based detection scheme gives better the performance considering two featured data samples like the ANN-based model. SVM-based algorithm reaches approximately 96.30% on an average of all sensor distribution. Similarly, it achieves lowest false

positive and false negative rates for two featured data samples which are 6.27% and 0.58% approximately. Using two featured data samples, the performance of the SVM-based algorithm enhances around 3.39%.

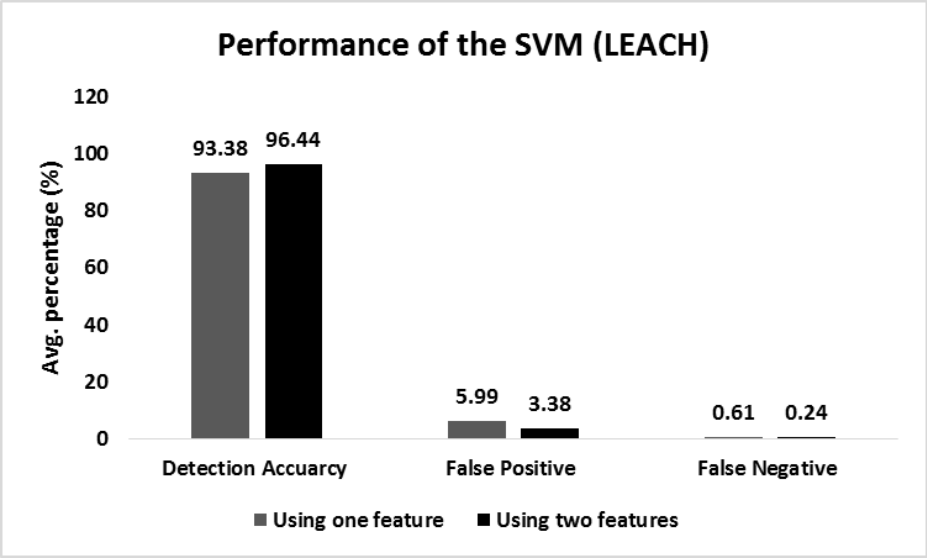

Figure 6.15 Performance of the SVM-based detection scheme (LEACH).

Figure 6.15 shows the performance of the SVM-based detection scheme considering LEACH protocol. Similar kind of trend in detection accuracy, shown in Figure 6.12, is observed for the LEACH protocol. In Figure 6.15, SVM-based detection scheme performs better on the two featured data samples. It achieves averagely 96.44% detection accuracy for all sensor distribution, whereas, 93.38% detection accuracy is measured for the one featured data samples. In this case, the detection accuracy increases approximately 3.27% for the two featured data samples. Inversely, the false positive rates and false negative rates declined significantly for the two featured data samples.

6.3. The performance of the proposed LR based detection scheme

Non-linear logistic regression (LR) algorithm is used to measure the performance on this classification problem and compare its outcomes with the proposed algorithm. Table 6.4 shows the parameters which are used during the training phase of the non-linear logistic classification algorithm.

Table 6.4 Parameters used for logistic linear classification

Parameter	**Value**
No of attributes	2
No of Data samples (training)	49000
No of Data samples (testing)	1000
Iteration	07
Regularized parameter (λ)	default
Learning method	Newton's methods
CPU time (for one instance)	3.17 mins

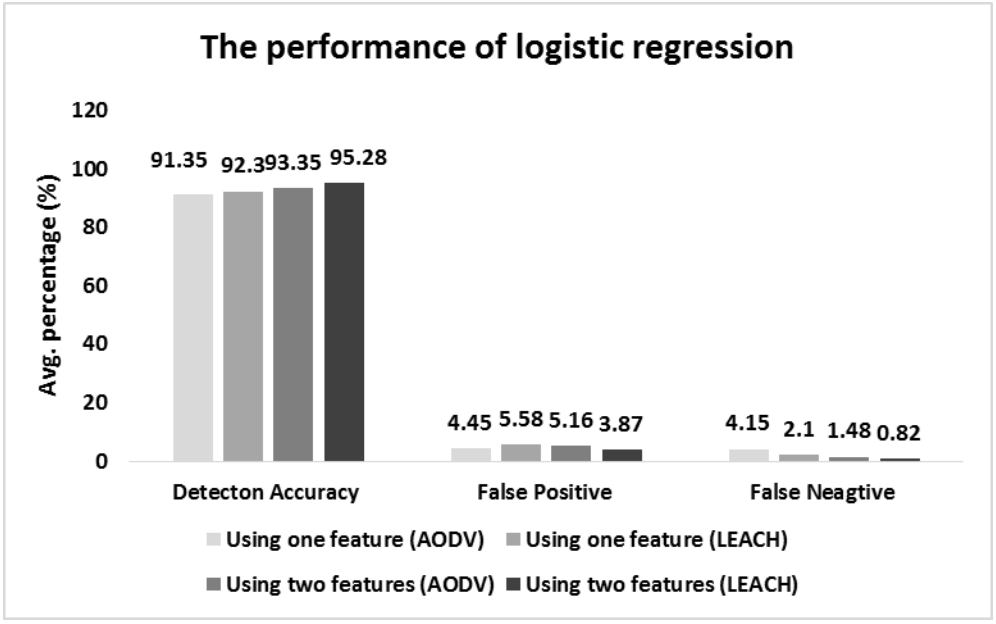

Figure 6.16 Performance of the LR.

Figure 6.16 represents the overall performance of the LR to detect the wormhole attack considering one featured and two featured data samples. According to the Figure 6.16, LR based detection scheme performs better with two featured data samples when the network uses LEACH as the routing protocol. The detection accuracy reaches to 95.28 % considering two featured data samples. In this case, lowest false positive rate and false negative rate are achieved for the LEACH routing protocol, which is 3.87% and 0.82% respectively. Most importantly, this LR based detection scheme gives better performance with two featured data samples like ANN or SVM based detection scheme.

6.4. Performance comparison

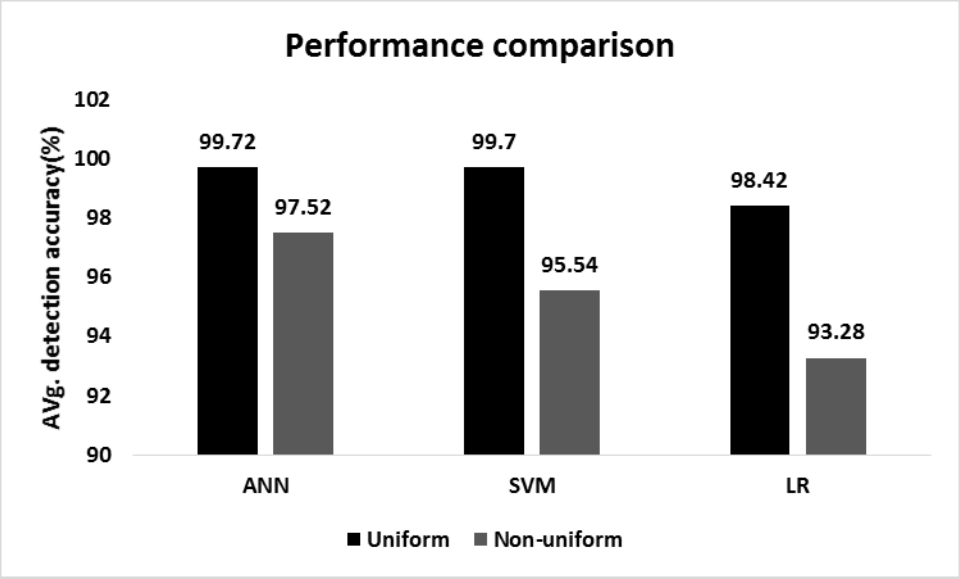

Figure 6.17 Performace analysis on the uniform and the non-uniform sensor distribution.

Figure 6.17 represents the performance of the detection schemes on the uniform and the non-uniform sensor distributions considering two featured data samples. Clearly, for the uniform sensor

distribution, three (03) detection schemes achieve higher detection accuracy in contrast to the non-uniform sensor distributions. As the sensor density of the area is uniform; the neighborhood counts increase significantly with the presence of the wormhole nodes in the network. In contrary to the non-uniform sensor distribution, the neighborhood counts increase slightly or significantly depends on the position of the wormhole nodes in the network. Sometimes the neighborhood counts are smaller than the count taken from the wormhole non-infected zone (wormhole nodes can be placed in the sparsely populated area in the network). Since the uniform sensor distribution has a consistency in the node density, the total neighbor population of the wormhole nodes is much smaller than the total neighborhood counts in non-uniform sensor distribution. Since the neighbors of the wormhole nodes are the only way to reach the wormhole nodes; therefore, the neighbors of wormhole nodes in uniform distribution dissipate more energy than the neighbors of the malicious node in non-uniform sensor distribution. Hence, positive data samples are easily distinguished from the presented data set. That's why the detection accuracy of these schemes is much better for uniform sensor distribution.

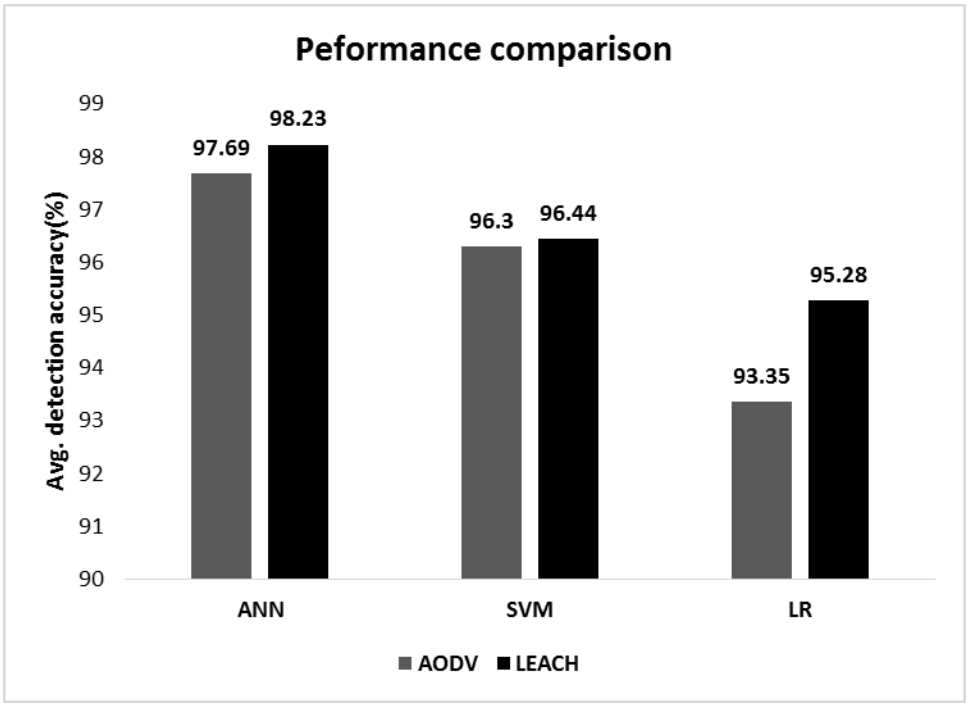

Figure 6.18 Performance comparison among ANN,SVM, and LR

If we analyze the performance of the three detection schemes, ANN outperforms SVM and LR in detecting the malicious samples that represent wormhole attack considering both AODV and LEACH routing protocol. According to the Figure 6.18, ANN achieves 97.69% and 98.23% detection accuracy respectively for AODV and LEACH routing protocol. SVM based detection scheme performs better than LR based detection scheme (achieves approximately 96% detection accuracy) in view of AODV and LEACH. Considering LEACH routing protocol, the LR based detection scheme attains 95.28% detection accuracy which is the lowest among all the detection schemes. As we know, the performance of the machine learning techniques highly depends on the dataset. In this case, ANN distorts the two featured data samples in the higher dimension better than other two machine learning techniques; and efficiently put dynamic decision surface in between positive and negative data samples. That would help the ANN-based detection scheme to gain highest detection accuracy.

Another important observation of this research is that the three detection schemes give better performance in classifying positives data samples for LEACH protocol. In the cluster based routing protocol like LEACH, a node is nominated as a cluster head among the members of a cluster for a specific round of data transmission to the base station. Once the cluster head is selected, other members send their data packets to the cluster head. Afterward, selected cluster head transfers the data packets to the base station. For the next round, a new member is randomly chosen as cluster head from the other members who is not selected as cluster head before for any round. However, wormhole node manipulates the cluster head selection process in a particular cluster of nodes. Furthermore, in the active mode of attack, wormhole node can be selected as a cluster head. If it happens, the wormhole node gets a bunch of data packets easily from the other members through its neighbors. As we know, wormhole nodes create shortcut path to the base station in the network. Therefore, the cluster head is deceived throughout the data transmission process to the base station. If wormhole nodes are placed in two different clusters, the members of the cluster may also be deceived while they are selecting closest cluster head to transmit their data packets. Hence, the neighbors around wormhole node dissipate more energy in LEACH compared to AODV. The extra dissipation of the energy reflects in the feature called AREPN collected by detector node. That would help in separating the positive data samples and negative data samples when two featured data samples are presented to the detection schemes.

Chapter 7

7. Conclusion and Future works

Wormhole attack is one of the detrimental network layer attacks for wireless sensor networks. This thesis presents a novel detection model based on neighborhood count and AREPN using an ANN for wireless sensor networks. The goal of this proposed detection scheme is to detect wormhole attacks (In-band, out of band, hidden or active mode of wormhole attack) with higher precision and accuracy, especially in a non-uniform network environment. The experimental results confirm that the proposed detection scheme is able to identify the existence of the wormhole node without requiring any special hardware both in uniform and non-uniform sensor distribution. Another important aspect of this detection model is it doesn't increase the significant amount of network overhead flow throughout the network. The simulation results also validate that our ANN-based approach performed better than SVM or LR based detection scheme. This ANN based detection scheme achieves around 99.72% (average) and 97.52% (average) accordingly for uniform and non-uniform sensor distribution. Most significantly, the probable location of the wormhole node can be identified by this scheme. Future works are required to enhance the performance of the detection scheme and to investigate in different directions so that we can evaluate the efficacy of the proposed detection scheme perfectly. Proposed future works are noted as follows.

a) Nowadays, deep neural networks and convolutional neural networks are used to enhance the performance internet threat detection schemes. We want to apply this advanced machine learning algorithms on the acquired data set to evaluate their performance. After that, we will compare the results with the Proposed ANN based scheme.

b) In this research, the relationship between detection accuracy and the total count of neighbors around wormhole nodes is investigated. In the future work, we want to investigate the impact of changes in radio range of sensor nodes ($60m$,$70m, and$ $80m$ etc) and number of the sensor nodes (deployed) and position of the wormhole nodes on the detection accuracy of the proposed model.

c) If we use more than one mobile node, the two featured data samples will be gathered more quickly rather than using single detector node. Multiple detector nodes will help to locate the positions of the wormhole nodes by triangulation algorithm. Therefore, in the future work, we want to investigate the performance of the proposed detection scheme by deploying multiple detector nodes in the different sensor network environments.

d) As we know, the performance of the ANN depends on its own architecture. In the future work, we want to apply the acquired data set on different network architecture (Varying number of neurons in a layer, varying number of hidden layer) to optimize the performance of the proposed ANN-based detection scheme.

e) In this detection scheme, probable location of the wormhole node is identified through D_N. We want to work further in this direction to find the exact location of the malicious nodes in the infected sensor network.

f) The energy dissipation of a sensor node depends on the routing protocol that being used in the network. In this research, the performance of the proposed detection model is verified considering on demand based and cluster based routing protocol. In the future,

we want to the test the proposed algorithm on the hybrid routing protocol such as zoned based routing protocol (ZRP) and wireless ad-hoc routing protocol (WARP).

References

[1] M. E.-S. Marianne Azer, Sherif El-Kassas, “A Full Image of the Wormhole Attacks Towards Introducing Complex Wormhole Attacks in wireless Ad Hoc Networks,” *Int. J. Comput. Sci. Inf. Secur.*, vol. 1, no. 1, pp. 41–52, 2009.

[2] Y. Xu, G. Chen, J. Ford, and F. Makedon, “Detecting wormhole attacks in wireless sensor networks using connectivity information,” *Crit. Infrastruct. Prot.*, vol. 2006, 2007.

[3] L. Hu and D. Evans, “Using Directional Antennas to Prevent Wormhole Attacks,” *Netw. Distrib. Syst. Symp. NDSS*, no. February, pp. 1–11, 2004.

[4] Y.-C. Hu, a. Perrig, and D. B. Johnson, “Packet leashes: a defense against wormhole attacks in wireless networks,” *IEEE INFOCOM 2003. Twenty-second Annu. Jt. Conf. IEEE Comput. Commun. Soc. (IEEE Cat. No.03CH37428)*, vol. 3, no. C, pp. 1976–1986, 2003.

[5] N. Sastry, U. Shankar, and D. Wagner, “Secure verification of location claims,” *Proc. 2003 ACM Work. Wirel. Secur. WiSe 03*, vol. 0, no. Section 2, pp. 1–10, 2003.

[6] N. Song, L. Qian, S. Ning, Q. Lijun, and L. Xiangfang, “Wormhole attacks detection in wireless ad hoc networks: a statistical analysis approach,” *Parallel Distrib. Process. Symp. 2005. Proceedings. 19th IEEE Int.*, p. 8 pp., 2005.

[7] L. Buttyán, L. Dóra, and I. Vajda, “Statistical Wormhole Detection in Sensor Networks,” *Secur. Priv. Adhoc Sens. Networks*, pp. 128–141, 2005.

[8] S. Song and H. Wu, “Statistical Wormhole Detection for Mobile Sensor Networks,” pp. 322–327, 2012.

[9] Mohammad Nurul Afsar Shaon and Ken Ferens, “Wireless Sensor Network Wormhole Detection using an Artificial Neural Network,” *ICWN*, pp. 115–120, 2015.

[10] M. a Azer, S. Member, S. M. El-kassas, M. S. El-soudani, and S. Member, “An Innovative Approach for the Wormhole Attack Detection and Prevention In Wireless Ad Hoc Networks,” pp. 366–371, 2010.

[11] Y. Zhou, L. Lamont, and L. Li, “Wormhole attack detection based on distance verification and the use of hypothesis testing for wireless ad hoc networks,” *2009 IEEE Mil. Commun. Conf. MILCOM 2009*, 2009.

[12] J. Tian, M. Gao, and F. Zhang, “Network Intrusion Detection Method Based on Radial Basic Function Neural Network,” *2009 Int. Conf. E-bus. Inf. Syst. Secur.*, pp. 1–4, 2009.

[13] S. Chattopadhyay and G. Bandyopadhyay, “Artificial neural network with backpropagation learning to predict mean monthly total ozone in Arosa, Switzerland,” *Int. J. Remote Sens.*, vol. 28, no. 20, pp. 4471–4482, 2007.

[14] C. Hon Sun and L. King-Shan, “DelPHI: wormhole detection mechanism for ad hoc wireless networks,” *1st Int. Symp. Wirel. Pervasive Comput.* , no. 852, p. 6 pp., 2006.

[15] P. Van Tran, L. X. Hung, Y. K. Lee, S. Lee, and H. Lee, “TTM: An efficient mechanism to detect wormhole attacks in wireless ad-hoc networks,” *2007 4th Annu. IEEE Consum. Commun. Netw. Conf. CCNC 2007*, pp. 593–598, 2007.

[16] T. Korkmaz, “Verifying physical presence of neighbors against replay-based attacks in wireless ad hoc networks,” *Int. Conf. Inf. Technol. Coding Comput. - Vol. II*, p. 704–709 Vol. 2, 2005.

[17] S. Capkun, L. Buttyan, and J.-P. Hubaux, “SECTOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks,” *Proc. 1st ACM Work. Secur. ad hoc Sens. networks*, vol. 67322, no. 5005, pp. 21--32, 2003.

[18] F. Hong, L. Hong, and C. Fu, “Secure OLSR,” *Proc. - Int. Conf. Adv. Inf. Netw. Appl. AINA*, vol. 1, pp. 713–718, 2005.

[19] Z. T. and A. H. Maw., “Wormhole attack detection in wireless sensor networks,” in *Proceedings of World Academy of Science Engineering and TechnologyEngineering and Technology*, 2008, vol. 46, no. 3, pp. 545–50.

[20] D. Johnson and D. Maltz, “The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4,” pp. 1–107, 2007.

[21] K. U. R. Khan, R. U. Zaman, a. V. Reddy, K. A. Reddy, and T. S. Harsha, “An Efficient DSDV

Routing Protocol for Wireless Mobile Ad Hoc Networks and its Performance Comparison," *2008 Second UKSIM Eur. Symp. Comput. Model. Simul.*, pp. 506–511, 2008.

[22] T. Van Phuong, N. T. Canh, Y. K. Lee, S. Lee, and H. Lee, "Transmission time-based mechanism to detect wormhole attacks," *Proc. 2nd IEEE Asia-Pacific Serv. Comput. Conf. APSCC 2007*, pp. 172–178, 2007.

[23] S. Yi, Y. Pei, and S. Kalyanaraman, "On the capacity improvement of ad hoc wireless networks using directional antennas," *Proc. 4th ACM Int. Symp. Mob. ad hoc Netw. Comput. - MobiHoc '03*, pp. 108–116, 2003.

[24] M. Takai, J. Martin, R. Bagrodia, and A. Ren, "Directional virtual carrier sensing for directional antennas in mobile ad hoc networks," *Proc. 3rd ACM Int. Symp. Mob. ad hoc Netw. Comput. - MobiHoc '02*, pp. 183–193, 2002.

[25] R. Ramanathan, "On the performance of ad hoc networks using beamforming antennas," *Proc. ACM MOBIHOC, Oct.*, pp. 1453–1458, 2001.

[26] S. Brands and D. Chaum, "Distance bounding protocols.pdf," in *In Theory and Application of Cryptographic TechniquesTheory and Application of Cryptographic Techniques*, 1993, pp. 344–359.

[27] Y. Hu, A. Perrig, and D. B. Johnson, "Wormhole Detection in Wireless Ad Hoc Networks," pp. 1–15, 2001.

[28] and Ý. G. Özdemir, S., M. Meghdadi, "A time and trust based wormhole detection algorithm for wireless sensor networks," *3rd Inf. Secur. Cryptol. Conf.*, vol. 94, no. 20, pp. 1–5, 2008.

[29] A. A. Pirzada and C. McDonald, "Circumventing sinkholes and wormholes in wireless sensor networks," *Conf. Wirel. Ad Hoc Networks*, 2005.

[30] W. Wang and B. Bhargava, "Visualization of wormholes in sensor networks," pp. 51–60, 2004.

[31] K. B. Rasmussen and S. Capkun, "Implications of radio fingerprinting on the security of sensor networks," *Proc. 3rd Int. Conf. Secur. Priv. Commun. Networks, Secur.*, pp. 331–340, 2007.

[32] T. Okamoto, "Introduction to wormholes," pp. 1–14, 1988.

[33] M. Meghdadi, S. Ozdemir, and I. Gueler, “A Survey of Wormhole-based Attacks and their Countermeasures in Wireless Sensor Networks,” *Iete Tech. Rev.*, vol. 28, no. 2, pp. 89–102, 2011.

[34] M. Khabbazian, H. Mercier, and V. K. Bhargava, “NIS02-1: Wormhole Attack in Wireless Ad Hoc Networks: Analysis and Countermeasure,” *IEEE Globecom 2006*, vol. 8, no. 2, pp. 736–745, 2006.

[35] P. Lee and S. Member, “A Passivity Framework for Modeling and Mitigating Wormhole Attacks on Networked Control Systems,” *Tac*, vol. 59, no. 12, pp. 3224–3237, 2014.

[36] A. Rasheed and R. Mahapatra, “Mobile sink using multiple channels to defend against wormhole attacks in wireless sensor networks,” *2009 IEEE 28th Int. Perform. Comput. Commun. Conf.*, pp. 216–222, 2009.

[37] T. Giannetsos, T. Dimitriou, and N. R. Prasad, “State of the art on defenses against wormhole attacks in wireless sensor networks,” *Wirel. Commun. Veh. Technol. Inf. Theory Aerosp. Electron. Syst. Technol. 2009. Wirel. VITAE 2009. 1st Int. Conf.*, pp. 313–318, 2009.

[38] A. D. Wood and J. A. Stankovic, “A Taxonomy for Denial-of-Service Attacks in Wireless Sensor Networks,” *Handb. Sens. Networks Compact Wirel. Wired Sens. Syst.*, pp. 739–763, 2004.

[39] S. Bhanu, “Data Security in Wireless Sensor Network,” vol. 2, no. 2, pp. 3196–3204, 2014.

[40] I. Khalil, S. Bagchi, and N. B. Shroff, “LITEW ORP : Detection and isolation of the wormhole attack in static multihop wireless networks,” vol. 51, pp. 3750–3772, 2007.

[41] S. Pal, A. Mukhopadhyay, and P. Bhattacharya, “Defending Mechanisms Against Sybil Attack in Next Generation Mobile Ad Hoc Networks,” *IETE Tech. Rev.*, vol. 25, no. 4, p. 209, 2008.

[42] T. R. Andel and A. Yasinsac, “The invisible node attack revisited,” *Conf. Proc. - IEEE SOUTHEASTCON*, pp. 686–691, 2007.

[43] D. Glynos, P. Kotzanikolaou, and C. Douligeris, “Preventing impersonation attacks in MANET with multi-factor authentication,” *Proc. - WiOpt 2005 Third Int. Symp. Model. Optim. Mobile, Ad Hoc, Wirel. Networks*, vol. 2005, pp. 59–64, 2005.

[44] J. Douceur, “The sybil attack,” *Peer-to-peer Syst.*, pp. 251–260, 2002.

[45] S. Ughade, R. K. Kapoor, and A. Pandey, “An Overview on Wormhole Attack in Wireless Sensor Network : Challenges , Impacts , and Detection Approach,” vol. 2, no. 4, pp. 105–110, 2014.

[46] H. Yih-Chun and A. Perrig, “A survey of secure wireless ad hoc routing,” *IEEE Secur. Priv. Mag.*, vol. 2, no. 3, pp. 28–39, 2004.

[47] and J. S. atel, Hinal, “Wormhole Attack Detection in Wireless Sensor Networks: A Survey.,” *Int. J. Innov. Res. Dev.*, pp. 545–550, 2014.

[48] K. Akkaya and M. Younis, “A survey on routing protocols for wireless sensor networks,” *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[49] M. London and M. Häusser, “Dendritic Computation,” *Annu. Rev. Neurosci.*, vol. 28, no. 1, pp. 503–532, 2005.

[50] Marvin Minsky and Seymour Papert, “Review of Perceptrons, An Introduction to Computational Geometry,” *Bull. Am. Math. Soc.*, vol. 78, no. January, pp. 12–15, 1972.

[51] W. Huang and L. Ju, “A novel intrusion detection method based on conjugate gradient neural network,” in *2010 International Conference On Computer Design and Applications*, 2010, vol. 2, no. Iccda, pp. V2-470-V2-472.

[52] D. C. Park, M. a. El-Sharkawi, R. J. Marks, L. E. Atlas, and M. J. Damborg, “Electric load forecasting using an artificial neural network,” *IEEE Trans. Power Syst.*, vol. 6, no. 2, pp. 442–449, 1991.

[53] P. G. Kumar and D. Devaraj, “Network Intrusion Detection using Hybrid Neural Networks,” *2007 Int. Conf. Signal Process. Commun. Netw.*, pp. 563–569, 2007.

[54] J. Ren, “ANN vs. SVM: Which one performs better in classification of MCCs in mammogram imaging,” *Knowledge-Based Syst.*, vol. 26, pp. 144–153, 2012.

[55] R. S. Sutton, P. J. Werbos, N. K. Gupta, E. Rosenfeld, J. Mccool, and L. Jolla, “Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences .,” 1988.

[56] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities.,” *Proc. Natl. Acad. Sci.*, vol. 79, no. 8, pp. 2554–2558, 1982.

[57] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *Bull. Math. Biophys.*, vol. 5, pp. 115–133, 2008.

[58] N. Brunel, V. Hakim, and M. J. E. Richardson, “Single neuron dynamics and computation,” *Curr. Opin. Neurobiol.*, vol. 25, pp. 149–155, 2014.

[59] Nola Taylor Redd, "what is wormhole?" 13 April 2015. [Online]. Available: http://www.space.com/20881-wormholes.html [Accessed 05 August 2016].

[60] C.shanika, "Wireless sensor network" 12 June 2015. [Online]. Available: http://www.winstudent.com/wireless-sensor-networks/ [Accessed 05 August 2016].