SINKHOLE ATTACK DETECTION IN A WIRELESS SENSOR NETWORKS USING ENHANCED ANT COLONY OPTIMIZATION

BY

NWANKWO, Kennth Ejike
MTech/SICT/2017/7020

A THESIS SUBMITTED TO THE POSTGRADUATE SCHOOL FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA, NIGERIA IN PARTIAL FULFILLMENT OF THE DEGREE OF MASTER OF TECHNOLOGY IN CYBER SECURITY SCIENCE

June, 2021

ABSTRACT

Wireless sensor networks (WSN) comprise of tiny sensor nodes that are able to sense and process data. This environment has limitations of low energy, low computational power and simple routing protocols, making is susceptible to attacks such as sinkhole attack. Sinkhole attack occurs when an attacker node in a wireless sensor network disguises itself as the legitimate node closest to the base station, in order to have data sent by a source node to another destination node pass through it, hence having the opportunity to modify, drop or delay data from reaching to the base station as intended. In this thesis, the research developed a sinkhole detection scheme, enhancing ant colony optimization by including a hash table in the ant colony optimization technique to improve sinkhole attack detection and reduce false alarm rate in a wireless sensor network. An increase in detection rate of 96% was achieved and result out performed other related research works when compared and further research discussed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

WSN	Wireless Sensor Network
ACO	Ant Colony Optimization
RREP	Route Reply
RREQ	Route Request
VANET	Vehicular Ad Hoc Network
EACO	Enhanced Ant Colony Optimization
ADC	Analog to Digital Converter
GPS	Global Positioning System
OSI	Open System Interconnection
MAC	Medium Access Control
DDL	Data Link Layer
WBAN	Wireless Body Area Network
IMD	Implantable Medical Devices
MANET	Mobile Ad Hoc Network
OBU	On-Board Unit
EPSO	Enhanced Particle Swarm Optimization

EACS	Enhanced Ant Colony System
EEABR	Energy Efficient And-Based Routing
FBSD	Flow-Based Sinkhole Detection
KMT	Key Management and distribution Toolkit
ABC	Artificial Bee Colony
SMD	Sinkhole Modification Node
HWSN	Hierarchical Wireless Sensor Network
CH	Cluster Head
SDL	Sinkhole message Delay Node
SDP	Sinkhole message Dropping Node
PSO	Particle Swarm Optimization
NS3	Network Simulator 3

CHAPTER ONE

1.0 INTRODUCTION

1.1 Background to the Study

Wireless Sensor Network (WSN) is an interconnection of sensing nodes that can come in different sizes connecting to a base station that makes meaning out of the data received. Being one of the highly utilized network types and applied in many areas as in health care monitoring, area monitoring, earth and environment sensing including industrial monitoring. These sensors monitor the environment, collecting data and sending to the base station. WSN can be used in an environment that is physically without protection or attended to (Kumar and Poonam 2013). The peculiar nature of wireless sensor networks makes the vulnerable to security threats of different types and purposes. With the simplicity of their routing techniques, security is the greatest challenge hence making them more susceptible to many network attacks, some of which are Sinkhole attack, Selective-Forwarding attack, Wormhole, Hello Flood, sybil attacks, attack node replication, and Blackhole attack according to (Mohammadi and Jadidoleslamy 2011). Sinkhole attack formidable as it can lead to every one of the other attacks. Figure 1.1 depicts a typical WSN networks and all it comprises of.

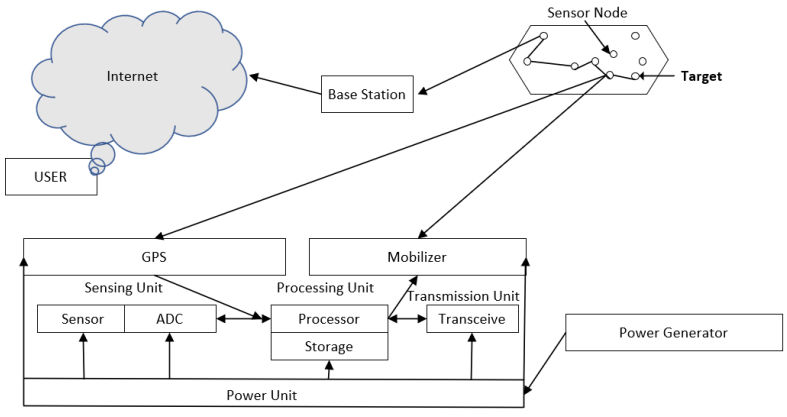

Figure 1.1 Sensor Node Architecture (Kumar and Poonam 2013).

Sensor networks are generally sent in an assortment of utilization going from military to natural and medicinal research. In numerous applications, for example, target following, war zone reconnaissance and gatecrasher recognition, WSNs regularly work in hostile and unattended situations. In this manner, there is a solid requirement for ensuring that information and readings are detected. In remote situations, an enemy not exclusively can listen in the radio traffic, yet additionally can catch or interrupt the traded messages. In this way, numerous conventions and calculations don't just work in unfriendly conditions without having sufficient safety efforts. Subsequently, security ends up one of the significant concerns while structuring security conventions in asset compelled WSNs. A portion of the uses of WSNs are in war zone surveillance, medicinal services applications, natural observing, shrewd home and vehicular specially appointed systems (VANETs) and some more.

In this research, ant colony optimization (ACO) is enhanced by inclusion of a hash table to the normal ACO that gave an increase in the detection rate and reduction in the false alarm rate for sinkhole attack detection is proposed.

1.2 Statement of Research Problem

In order to detect sinkhole attacks in WSNs, a number of researchers have previously proposed some optimization based solutions. Keerthana and Padmavathi (2016) achieved a detection rate of 87.062% with false alarm rate of 10.648% using ACO and Nadeem and Alghamdi, (2019) used a data aggregation algorithm and achieved detection rate of 90% and a false alarm rate of 9% in sinkhole detection. There is still the problem of detection and false alarm rate in detecting sinkholes in WSN, hence, the gap addressed in this

research, with the wealth of research on-going on the detection and prevention of sinkhole attack in WSN because of its malignant nature, there is room for improvement on solution of false alarm rate with this technique.

1.3 Aim and Objectives of Study

The aim of this research is to develop an enhanced ant colony optimization algorithm to detect sinkhole attack in WSN and the objectives are to:

i. design an enhanced ant colony optimization technique (EACO) for sinkhole attack detection.
ii. develop the EACO sinkhole detection algorithm in WSN.
iii. evaluate the EACO using detection rate and false alarm rate.

1.4 Significance of Study

This study provides a proper background of sinkhole detection techniques to new researchers and a clear view for experts seeking to improve already existing methods as it gives an objective view of the major methods already implemented, giving room for a lot of improvement.

1.5 Scope of Study

The scope of the research is limited to sinkhole attacks. It also focuses on WSNs with 250 or more nodes as this is necessary for impact of the algorithm to be seen. Known used techniques remains our core focus as to improve methodology.

CHAPTER TWO

2.0 LITERATURE REVIEW

2.1 Concept Review

2.1.1 Sensor node architecture

The technique sensing is used to collect the information of a physical entity as well as the occurrence of the events like modification in a state like fall in pressure and temperature. An object doing such a sensing job is known as a sensor. A sensor is a tool that converts parameters in the material world into signals that can be calculated and analyzed. Sensor node has onboard storage and embedded processing capabilities. The node contains so many sensors working in the acoustic; seismic; radio (radar); infrared; optical; magnetic; and chemical or natural fields. Every scattered sensor nodes have the skill Kazem *et al*. (2007) to assemble data; evaluate and direct them to a chosen sink position.

Sensors are classified into three parts: (i) Passive (ii) Omnidirectional sensors (iii) Narrow-beam sensors (iv) Active sensors. Passive sensors give the good judgment about the data without really influencing the surroundings via active probing. These are self-powered means that energy is only required to increase their analog signals. But the active sensors actively explore the atmosphere, for example; a sonar or radar sensor and wants nonstop energy from a power resource. The narrow-beam sensors include a definite concept of way of measurement, same as the concept of a camera. The Omni-directional sensors have not the concept of way of measurement and direction.

The main point of sensor networks is given as:

i. Sensor nodes are strongly deployed.

ii. Sensor nodes have chances of failure.

iii. The physical arrangement of a sensor network alters repeatedly.

iv. Sensor nodes have power; computational capacities; and memory that are very limited.

v. Sensor nodes have huge overhead and have many nodes so they may not have any global identification number.

In Figure 1.1, Architecture of sensor node contains six parts: Sensing unit; Processing unit; Transmission unit; Power unit; Global Positioning System and mobilizer.

I. **Sensing Unit:** Information gathering as input like pressure, temperature etc., from the environment is performed by the sensing unit Kumar and Poonam (2013) and generates the output in the form of electrical and optical signals. The analog signals are transformed into digital signal by ADC (Analog to Digital Converter).

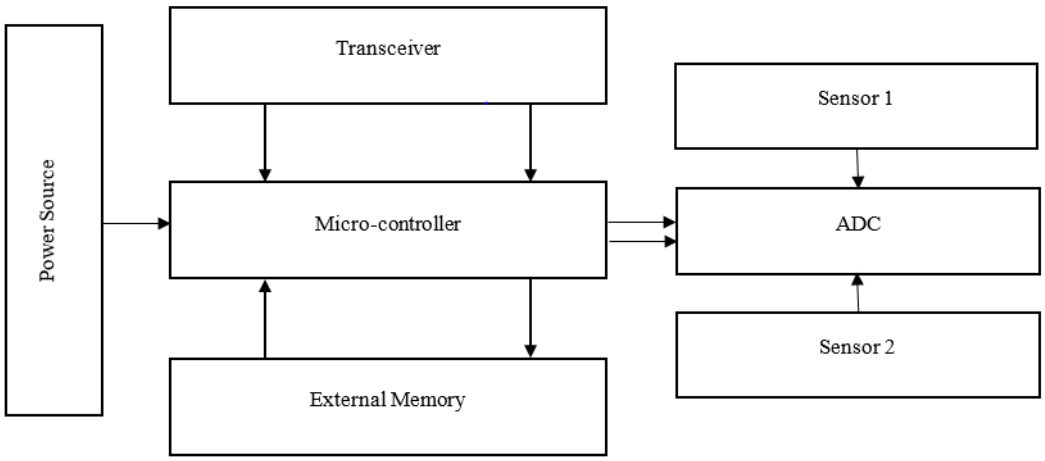

Figure 2.1 Sensor Node Structure (Kumar and Poonam 2013)

II. **Transceiver:** It contains the properties of both transmitter and receiver . The prepared states of transceiver devices are transmission, receive, unused, and sleep. The transceivers working in sate of idleness have same amount of power consumed in sending and receiving. So, it is good to stop the transceiver device when it is not used.

III. **Memory:** The requirement of memory is dependent on the person who is using it. There are two types of memory depend on the reason of storage. These are user memory used for saving the application-based data or personal data; and the program memory that is used in programming the device.

IV. **Power:** Power is provided to every sensor node through battery. As sensor nodes are put in place where it is difficult to reach so, it is difficult to provide main power supply and also difficult to change the battery regularly. Node required power for sensing, communicating and data processing. Power is stock up either in battery or in capacitors. The batteries can both rechargeable and non-rechargeable and are the key resource of power provider for sensor nodes. Now, solar sources are used by the sensors to renew their energy; temperature differences; or vibration.

V. **GPS**: The sensor node can also find out the location and position through the global positioning system (GPS).

VI. **The microcomputer/Controller**: The controller performs the task of handing out the data and has power to handle the other components' functionality in the sensor node.

2.1.2 Communication protocol in WSN

The Open Systems Interconnection (OSI) model is a conceptual model that demonstrates the communication functions of layers used in telecommunication with no consideration of the internal configuration and technology. The OSI models has aim is the interoperability of different communication systems with standard protocols. WSN based on OSI model have mainly five layers named application layer; transport layer; network layer; data link layer; and physical layer. Also, the three cross layers planes are added above those five layers of OSI model named as power management plane; connection management plane; and task management plane. The features of these layers are to handle the connection of network and permit the nodes to job together to boost the efficiency of the network as a whole.

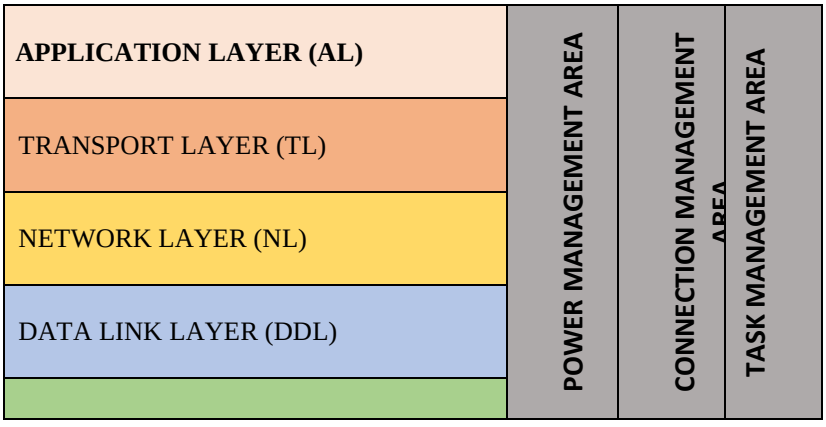

Figure 2.2 Layers of WSN (Wazid and Das 2016)

a) **Physical layer**: In this layer, bits are transmitted instead of complete packet over the transmitted medium. It interacts with medium access control (MAC) layer to help detect and correct error, performing transmission and modulation. Maximizing of network lifetime and minimizing of energy consumption is starts from the physical layer in WSN. Energy is consumed in service radio signals and in transmission of bit. Radio signal consumed the fixed energy but energy spent during bit transmission varies and depend on distance, channel loss and interference. The devices used at physical layer are repeater, hub, controller, network adapter. (Yick *et al., 2008).*

a) **Data link layer**: This is the layer that has the charge of transmit the data between the two nodes of similar link by breaking up the input data into data frames. It provides the services that comprise medium access control; detection of error; reliable delivery; and correction of error Jangra and Kait (2017). For data transfer in wireless, there is a requirement of medium access control and its management. DDL (Data Link Layer) has two sub layers:

i. LLC (Logical link Control) – present at the top of DDL provide flow control and provide flow control error notification, address and control of data link.

ii. MAC (Medium access control) - present at the bottom of DDL and provide user access, frame structure delivery and frame synchronization.

2.1.2.1 Network layer: It handles the routing of data throughout the network from source to the target and successfully routing packets along the path. The requirement of

different routing protocol varies and depends on the communication path set up. Some routing protocol favor the path that assist WSN to convey the QoS and other the best lifetime and so on (Kaur *et al*., 2014).Routing protocols in WSNs are at the variance from conventional routing protocols in numerous ways like IP addresses are not contains in sensor nodes that is why routing protocols that are based on IP are not used in a WSN.

2.1.2.2 Transport layer: This layer ensures the consistency and superiority of data at the source and destination. It provides host to host and end to end communication services like flow control, multiplexing and reliability. It uses TCP (Transmission Control Protocol) connection oriented and UDP (User Datagram Protocol) connectionless protocol.

2.1.2.3 Application layer: The AL layer resides close to the user of the system (Yang, 2014). Various applications are implemented here including Telnet, Hyper Text Transfer protocol (HTTP), File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP) etc. In case of WSN, the application layer programming mainly deals with processing, encryption, formatting and storage of sensed information. It also examines the essential layers to sense if satisfactory network assets and services are existing to meet the user's necessity.

2.1.2.4 Cross layer area: To increase the network efficiency, management of network and node coordination, the cross-plane layers are used.

i. **Power management area**: It manages the power that a node used for sensing, processing and communication. For example, when the node has low power

intensity, it informs the cross plane and node does not participate in any activities like sensing.

ii. **Connection management area:** Management of network configuration is done by this layer for better connectivity of nodes.

iii. **Task management area:** Distribution of task among the nodes for better utilization of resources, so the network lifetime is better.

2.1.3 Applications of WSN

Sensor networks are widely installed in a variety of applications ranging from medical to environmental and military research (Wazid and Kumar Das 2016). In many applications, such as battlefield surveillance, target tracking and intruder detection, WSNs mostly operate in hostile and unattended environments. Therefore, there is a strong need for protecting the sensing data and sensing readings. In wireless environments, an enemy not only can eavesdrop the radio traffic, but also has the ability to capture or interrupt the transmitted messages. Thus, many protocols and algorithms do not simply work in hostile environments without having adequate security measures. Hence, security becomes one of the major concerns while designing security protocols in resource-constrained WSNs. Some applications of WSNs are discussed in section 2.1.3.

2.1.3.1 Battlefield surveillance

Consider the scenario of battle field surveillance, which is one of the major military applications. A large number of sensor nodes can be rapidly installed in a battlefield by low flying airplanes or trucks (Wazid and Kumar Das 2016). Each and every individual sensor node sense activities and conditions from its surrounding area after placement in the battle

field, and then sends these sensing observations to the *BS* through wireless communications through the neighboring sensor nodes. The *BS* then can conduct a more accurate detection on the activities (for example, possible attacks) of the opposing force after collecting a large number of sensing observations from the sensor nodes. Thus, the appropriate decisions as well as responses can be made quickly in the battle field.

2.1.3.2 Healthcare applications

Consider an application of WSNs in the area of healthcare medical research. In a wireless body area network (WBAN), the low-power tiny sensor nodes (also called as implantable medical devices (IMDs)) are placed around (or implanted in) the body of a patient for monitoring the patient's body functions and neighboring environment of a patient (Ghasemzadeh and Jafari 2011). An example of a WBAN is shown in Figure 1.4. With the help of WBAN, the patient's health related information, such as respiration, temperature, pulse oximeter, heart rate, blood sugar, blood pressure, pH, etc. can be monitored remotely. These health-related information

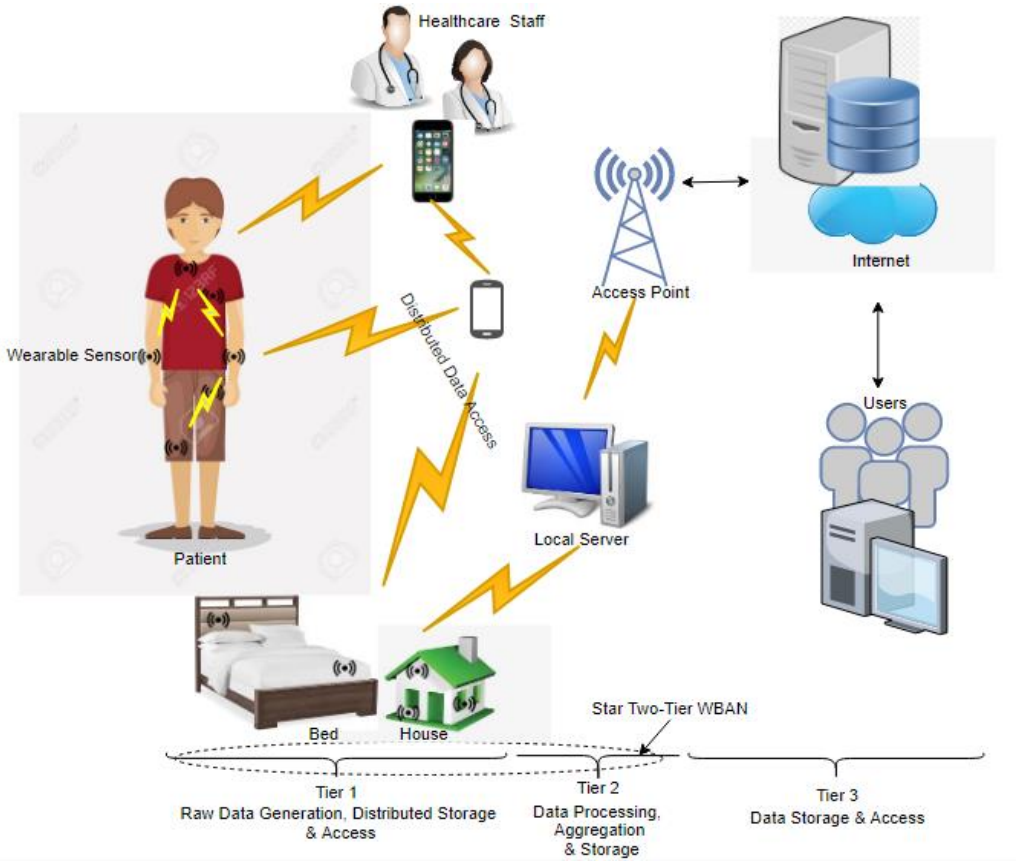

Figure 2.3 Architecture of WBAN (Jiang *et al*,.2016)

must be uninterruptedly processed in real time. The medical information needs to be shared and accessed by various levels of users, such as healthcare staff, researchers, government agencies, and insurance companies for taking the important decisions such as clinical diagnosis and emergency medical responses about the patients. The bio-sensors are placed in a patient's body in order to convey sensing data through a secure channel to a smaller body area network gateway. The gateway processes the data locally and resends through a secure channel to the router of the external network to the medical server in the hospital. The results then are observed and then analyzed by medical staffs/doctors to monitor patients. In this scenario, a patient can wear various bio-sensors. A centralized control device is used for data transmission from in and out of the network. This control device

also can be used as the gateway between internal network and the *BS*. The *BS* is connected to the external network.

2.1.3.3 Environmental monitoring

Some of the environmental applications of WSNs include the following (Akyildiz *et al*., 2002);

i. Tracking of movements of birds, small animals, and insects.
ii. Monitoring of the environmental conditions that affects crops and livestock.
iii. Macro devices for large-scale planetary exploration and earth monitoring.
iv. Chemical/biological detection.
v. Monitoring of underwater life.
vi. Monitoring of atmospheric contexts, radioactivity and studies of types of pollutions.
vii. Detection of forest fires and floods.

2.1.3.4 Smart home

These days sensors are also used for controlling electrical devices (for example, lighting system, geyser, air conditioner, etc.). So, smart homes can be built by deploying sensor nodes as they are capable to provide better lighting, heating and cooling systems (Sachan *et al*,.2013). A typical scenario of a smart home is given in Figure 2.4, in which a smart home user can access and control various smart devices such as light, temperature, humidity, etc., using his/her smartphone remotely via the Internet.

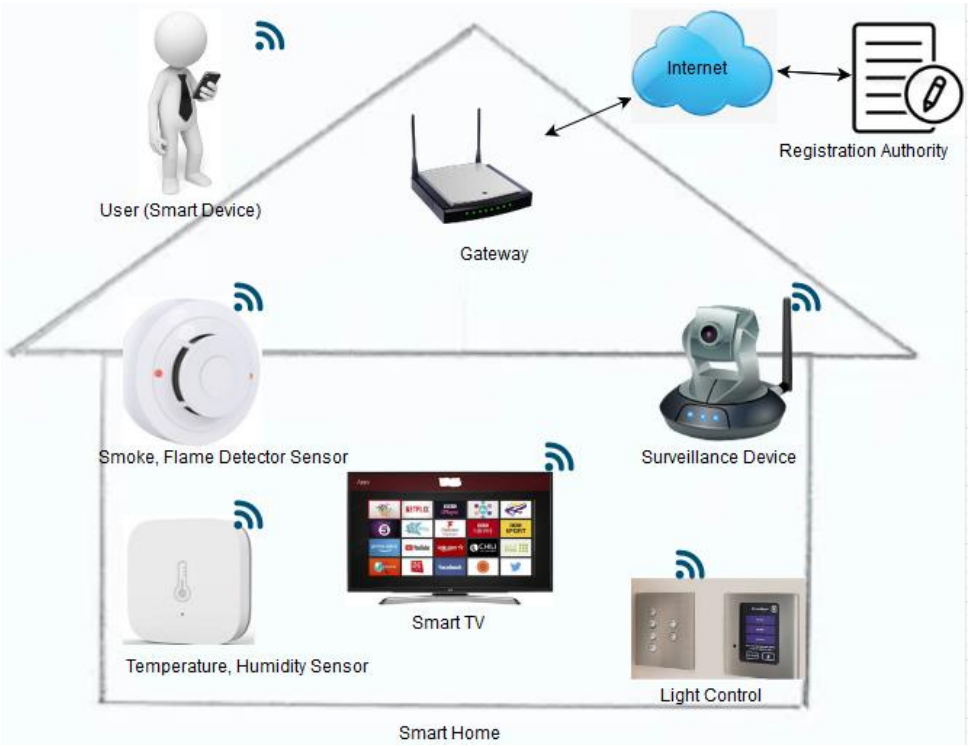

Figure 2.4: A smart home environment (Kumar *et al*., 2016)

2.1.3.5 Vehicular ad hoc networks (VANETs)

Vehicular Ad hoc Network (VANET) is considered as a special type of Mobile Ad hoc Network (MANET), which allows the vehicles on roads to form a self-organized network. In VANET, each vehicle is equipped with on-board unit (*OBU*) which collects and processes the data from all sensors fitted inside the vehicle which helps to analyze the condition of the vehicle and surroundings (Jiang *et al*,. 2016).

VANETs become the emerging technology as they provide multiple benefits, such as the in-built warning system, which warns the driver about the accidents and occurred collisions so a quick decision on the basis of the information provided would be taken.

information. The velocity information exchanged between vehicles prior to collision is helpful for the law-enforcement agency to reconstruct the accident scenario and also to provide digital evidences to them for the court room procedures. It also provides information about traffic congestion at the different roads so that driver can take decision on the basis of this and choose alternative roads. In summary, VANET helps to improve the road infotainment, environment, dissemination, and traffic safety for passengers as well as drivers. A typical VANET is shown in Figure 2.5

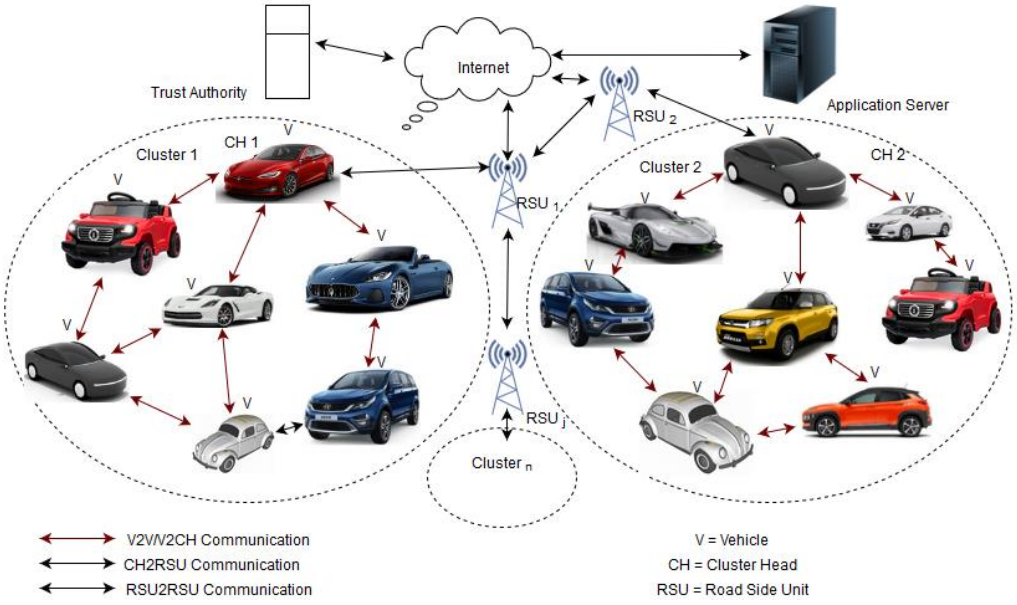

Figure 2.5: Vehicular Ad hoc Networks (VANETs) (Zhang *et al*., 2008)

2.1.3.6 Other applications

Some of other commercial applications of WSNs include the following (Akyildiz *et al.*, 2002);

i. In Monitoring of material fatigue.

ii. In Monitoring of product quality.

iii. For Construction of smart office spaces.

iv. For Robot control and guidance in automatic manufacturing environments.

v. Factory automation and process control.

vi. For Monitoring disaster area.

vii. In Building of smart structures with sensor nodes embedded inside.

viii. In Building of smart transportation system.

ix. In Factory instrumentation.

x. For Monitoring of wind tunnels.

2.1.4 Security requirements in WSN

Just as recommended, WSNs have also the following general security requirements (Jangra and Ph, 2017):

i. *Authentication:* It includes authenticating other sensor nodes, base stations, and cluster heads before granting a limited resource, or revealing information

ii. *Integrity:* It ensures that the message or the entity under consideration must not be altered.

iii. *Confidentiality:* It provides privacy of the wireless communication channels in order to prevent false reports injection.

iv. *Availability:* It aims to ensure the desired network services are live or available even in the presence of denial-of-service attacks.

v. *Non-repudiation:* It prevents malicious nodes from hiding their activities.

vi. *Authorization:* It ensures that only the sensor nodes, those who are authorized, is involved in providing information to network services.

vii. *Freshness:* It ensures that the data is recent, and no enemy can replay old messages.

Apart from these security requirements, the forward and backward secrecy are also important and these need to be observed as new sensors can be installed in the network and old sensors may fail due to energy problems.

i. *Forward secrecy:* When an existing sensor node leaves the WSN, it must not read future messages after its departure.

ii. *Backward secrecy:* When a new deployed node joins in the WSN, it must not read any previously transmitted message.

2.1.5 Limitations of sensor networks

The following limitations of sensor nodes of the WSN make it particularly challenging task to provide the security requirements as follows;

i. *Limited resources of sensor nodes:* Each node has a primitive processor feature with very low computing power and small amount of programmable memory.

ii. *Life-time limitation of sensor nodes:* Each node is battery-powered. So, after months or several weeks of operation, some nodes in the network tend to exhaust their power and as a result, the security protocols must be energy efficient.

iii. *Limited communication abilities of sensor nodes:* Sensor nodes have the ability to communicate with each other and the *BS*s using short range wireless radio transmission at low bandwidth.

iv. *Lack of knowledge about deployment configuration:* In most applications, the post-deployment network configuration is not possible to decide a prior. As a result, it may not be always possible to use security algorithms that have strong dependence on locations of sensor nodes in a sensor network.

v. *Issue of node capture:* Wireless sensor networks often operate in an unattended environment (Messerges *et al*., 2002). An enemy may physically capture some sensors to steal their stored sensitive secret data and codes from their memory as they are not generally equipped with tamper-resistant hardware using the power analysis attacks (Kocher *et al*., 1999),

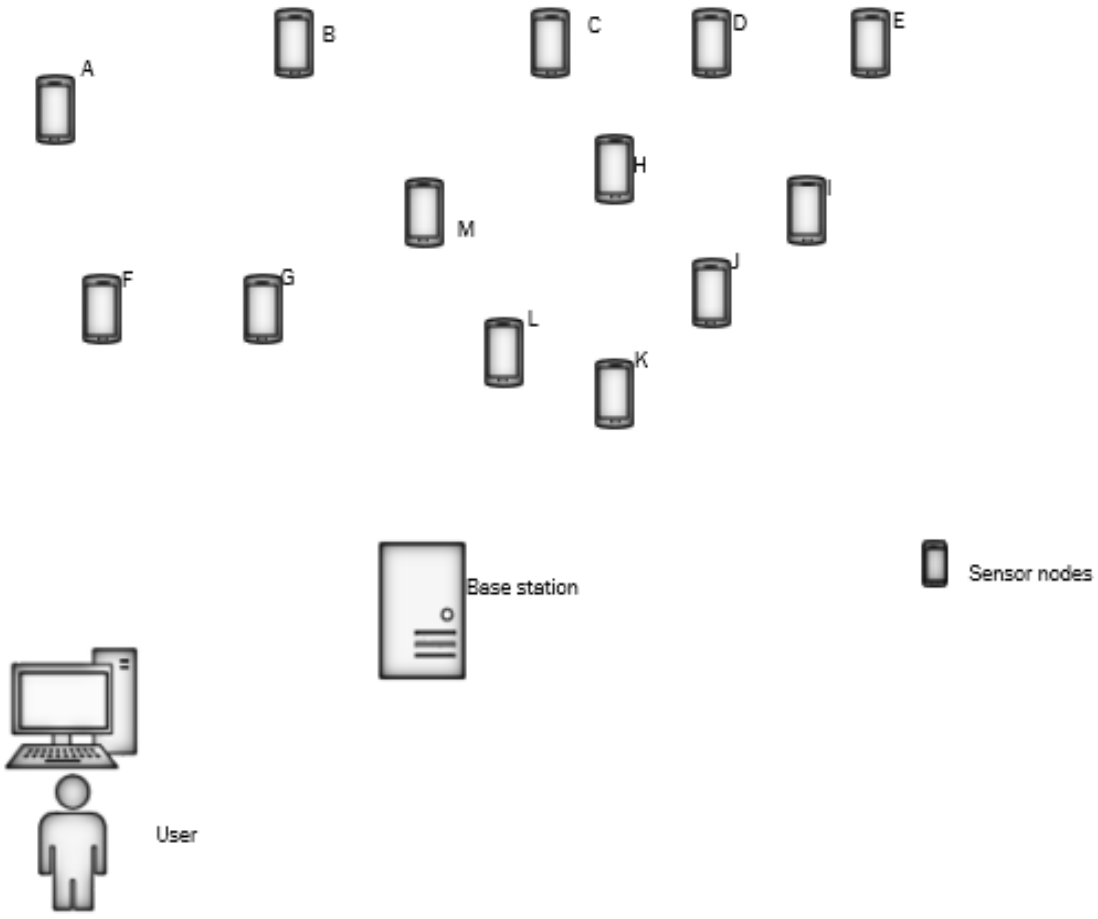

Figure 2.6 Wireless Sensor Network

2.1.6 Attacks in WSN

WSN attacks can be explained from different points of view (Zhou *et al*,. 2008)

I. Attack techniques

Various types of attacks are possible in a WSN. The functioning of WSN can be disrupted by using different attack techniques. Most of the communication protocols used in WSN are known publicly. So, the attackers can eavesdrop the packets and perform cryptanalysis or traffic analysis. The eavesdropped packets can be used in replay at a later time or at some other location to cause inconsistency. An attacker can inject false packets into the network to confuse the nodes. The deployed malicious sensor nodes can also modify received packets before forwarding them towards to the destination.

Physical node capturing is one of the most harmful attacks in a WSN. Usually, sensor nodes are installed in hostile environment without any supervision and monitoring. Therefore, an enemy can easily capture a node and extract all useful information (for example, identity, secret key, etc.) from compromised sensor node.

Sometimes the attackers are not interested in the information leakage or modification and they just want to disrupt the functionality of the network. They may introduce strong radio signal into the network to cause radio jamming that further disrupt the communication between the nodes, and cause denial of service (DoS) attack.

a) Passive and active attacks

Attacks can be categorize on the basis of operation mode, such as passive attack and active attack (Wazid and Kumar Das 2016). Sometimes an attacker only wants to know what information is exchanging among the nodes and does not want to disrupt the network. These types of attacks are passive in nature. The attacker collects a large volume of data by participating passively in the network. Then, by performing analysis on the generated data, the attacker may extract some secret information. The detection of passive attacks can be very difficult as they leave very less evidences.

In active attacks, an enemy exploits the vulnerabilities in the implemented security protocols and launch various types of attacks, such as packet modification, injection and replaying. The impact of active attacks is high on the network as they affect most of the network performance parameters. Examples of some active attacks include blackhole attack, wormhole attack and sinkhole attack.

b) External and internal attacks

In the normal flow of the network, the nodes are honest and cooperative entities, whereas the attacker nodes are precluded from the network and have no access to the network. The external attacks can be launched only from the outside of the scope of the network. So, the effect of these attacks is limited especially in case of WSN.

The attacker can physically seize a sensor node and extract useful information such as its identity and secret key. from that node, and can also deploy some fake sensor nodes by using the extracted information (Wazid and Das 2016). In this way, an internal attack can be performed. Internal attacks, such as blackhole, misdirection, wormhole and sinkhole, are very harmful in nature as they cause severe damage to the performance of the network.

II. Attacks on different layers of WSN stack

The layered architecture of WSN is divided into five layers, which are physical layer, data link layer, network layer, transport layer and application layer. The attacks corresponding to various layers of WSN stack are provided in Table 2.1 (Wazid and Das 2016)

Table 2.1: Layer wise attack on WSN stack (Sachan *et al.*, 2013)

Layer	Attacks
Physical	Tampering, sybil attack, jamming, interception
Data link	Sybil attack, collision, exhaustion, unfairness, replay attack, spoofing and altering routing attack, traffic analysis and monitoring
Network	Selective forwarding attack, black hole attack, sybil attack, neglect and greed, hello flood attack, spoofing attack, internet smurf attack, wormhole attack, grey-hole attack, misdirection attack, homing
Transport	Desynchronization, flooding attack
Application	False data injection, spoofing and altering routing

attack

Among all these attacks listed in Table 2.1, the network layer attacks are malicious as they disrupt the entire functionality of the network, particularly routing procedure that further causes denial of service (DoS) attack in the network. In Sinkhole attack, a compromised node attempts to get information to it from neighboring node. Thus, the sensor node picks up information, and knows what data is being communicated between neighboring nodes. This attack occurs when the compromised node disguises itself to be the most attractive to its neighboring node with respect to the algorithm by false advertising of itself as the node closest to the base station (Karlof *et al*., 2004). The figure 2.7 clearly illustrates a sinkhole attack in a wireless sensor network. As shown in the figure 2.7, nodes with data seeking to send to the base station, first sends a route request (RREQ) to all nearby nodes. Normally the node with the quickest path to the base station will be known from the route reply (RREP) respond coming from all the nearby nodes, this is where the sink node (SN) sends (RREP) reply too nodes no matter how far it is but pretends to have the quickest route to base station as shown in figure 2.7. Hence it attracts all data to itself and carries out any number malicious purposes like altering the information, dropping the information or dallying the information from getting to the base station, giving birth to other attacks like blackhole, wormhole and grey hole in the wireless sensor network Shafiei *et al*. (2014).

An attacker can compromise the integrity, confidentiality and authenticity of the wireless sensor network through a compromised node that becomes the sink node, hence, the wealth of research on ways to detect and mitigate such an attack on a wireless sensor network.

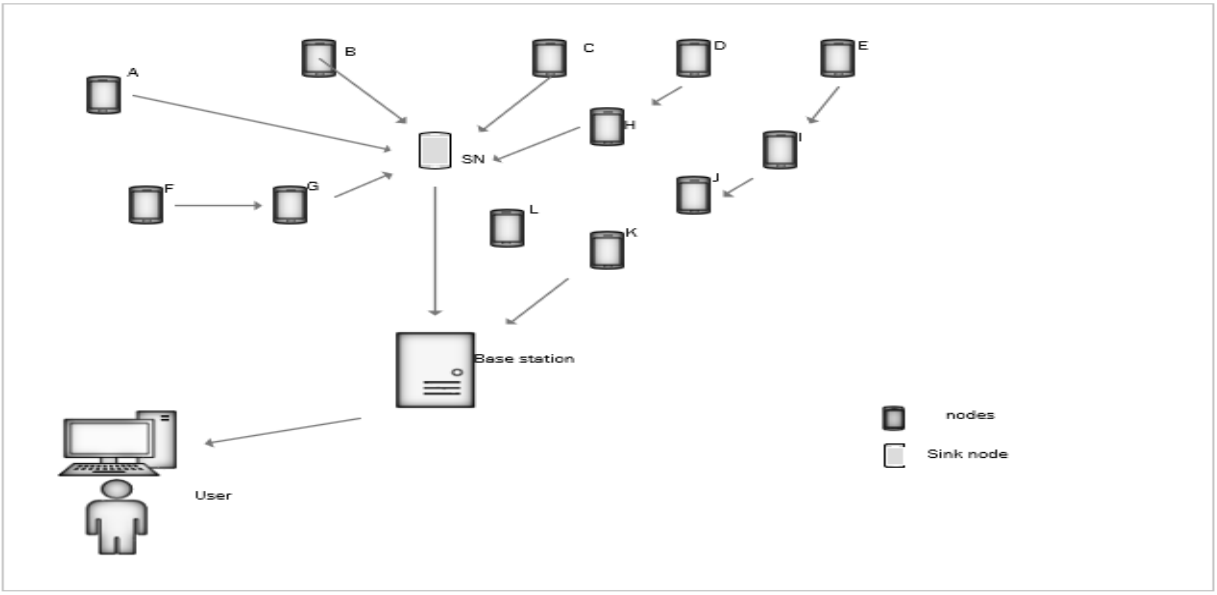

Figure 2.7 Sinkhole Attack Illustration (Shafiei *et al* 2014)

2.1.9 Ant colony optimization

ACO algorithm is derived from the behavior of ants when they search for food (Dorigo and Gianni, 1992). ACO is an intelligent algorithm, a type of swarm intelligence technique used to simulate food search process by ants. Ants secrete pheromones as they search for food to help bring them back to their initial location and to help discover the likely fastest distance that will bring success. Since ants have the ability to feel the amount of pheromone, as one ant finds food, others will surely follow the path which it paves to find the same food through the secreted pheromone. Ants always follow paths with high pheromone concentration that hence increases the pheromone secreted on the path which signal success. In theory, the more the amount of pheromone there are on the path, the

more other ants connect and follow. The path having the highest amount of pheromone soon becomes the optimal path for the ant colony to the food location.

Formulated model of ACO according to Zhao *et al*., (2016) are given in equation (2.1) to (2.6).

$$P_{ij}^{k} = \frac{[\tau_{ij}(t)]^{a}.[\delta_{ij}]^{\beta}}{\sum_{k\in\{N-tabuu_{k}\}}[\tau_{ij}(t)]^{a}.[\delta_{ij}]^{\beta}} \quad (2.1)$$

$$\tau_{ij}(t+1) = P * \tau_{ij}(t) + \sum_{k=1}^{N} \Delta\tau_{ij}(t)^{k} \quad (2.2)$$

$$\Delta\tau_{ij} = \frac{Q}{L_{k}(t)} \quad (2.3)$$

$$\delta_{ij} = \frac{1}{d_{ij}} \quad (2.4)$$

$$(X_{lt} - X_{li+1})^{2} + (Y_{lt} - Y_{lt+1})^{2} = d_{li+1,lt^{2}} \quad (2.5)$$

Q being constant, P_{ij}^{k} is probability of following node $d_{l_{i+1}}$ is distance between node l_{i+1} while the end node l_t , the $\tau_{ij}(t)$ is amount of pheromone in the curve (i,j). (x_{l_t}, y_{l_t}) and $(x_{l_{t+1}}, y_{l_{t+1}})$ are respectively coordinates l_t and l_{i+1} . δ_{ij} is the heuristic data in curve. α and β are weight factors of $\tau_{ij}(t)$ and δ_{ij}. where N is number of ants, $L_k(t)$ shows the object function.

Hence, the model updating and coverage of pheromone.

Once an ant selects the next node location, the roulette wheel technique is used, which continues till the next node destination is obtained by the ants. At once, pheromones of all the nodes are restructured equation according to (Zhao *et al*., 2016).

$$\tau_{li,li+1}(k, \aleph + 1) = \rho\tau_{li,li+1}(k, \aleph) + \Delta\tau_{li,li+1}(k, \aleph) \quad (2.6)$$

where ρ represents the rate evaporation of pheromones

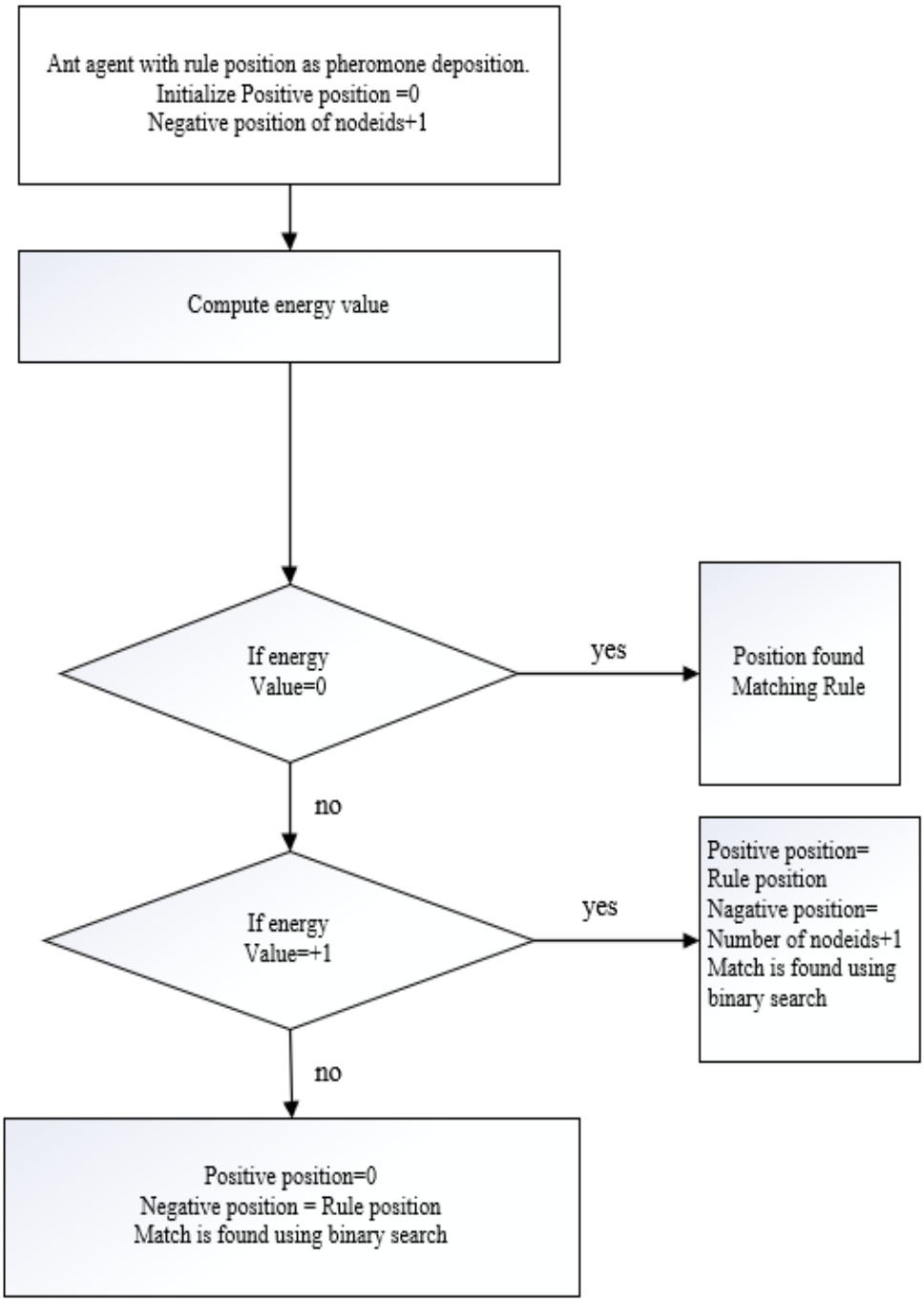

Figure 2.8 Ant System Model(Dorigo and Gianni, 1992)

Table 2.2 **Pseudocode for ACO** (Dorigo and Gianni, 1992)

Procedure ACO ( )

```
{
Input n, α , β , ρ
set  the ant colony configuration
set the initial pheromone and heuristic value
get ant colony optimization system based on the calculated cost matrix
i = 1
while (I <= n)
{                r = 1
While ( r <= i)
{
Reset the ants
Build ant s' solution
initiate local search
Update path best for i
Update pheromones
r = r + 1
}
Choose  path best for i
i = i + 1
}
}
```

2.2 Review of Related Works

Sinkhole attack detection and prevention in wireless sensor networks have been on for over a decade, and the summary of some of the recent researches include Jamal *et al*. (2019) who presented parameter evaluation for ant colony system to obtain the best values for

throughput, energy consumption and latency guiding research results to achieve optimal performance for packet routing process as summarized in Appendix A.

Nadeem and Alghamdi (2019) came up with a detection algorithm using information gotten from data aggregation algorithm to detect a sinkhole attacker in body area network, using omnet++ for simulation achieved a good performance in detection.

Nithiyanandam, Parthiban, Rajalingam and Scholar (2018) employed an enhanced particle swarm optimization (EPSO) modifying flocking that is based clustering algorithm that involved separation, alignment and cohesion of clustered nodes in the WSN employed to mitigate and detect sinkhole attack in a larger instance.

Nasir, Ku-Mahamud and Kamioka (2018) proposed an algorithm enhancing ant colony system. Inspired from a variant of ant colony optimization with the global and local pheromone updates to improve path exploitation and exploration with the effect of reduction in packet loss and increase in energy efficiency of sensor nodes.

Iwendi and Du (2018) improved the (KMT) technique using ant colony optimization for path panning to deceive intruders, improve and safe guard data collection from node to base station and vice versa. Using COOJA simulator to simulate, taking into consideration ACO key protection allocation and ACO-pheromone vanishing mechanism effectively balancing the blending speed and communication of the nodes.

Nasir *et al*. (2018) proposed an improved ant colony optimization algorithm to solve packet loss problems for nodes conveying more data packets than its capacity. Experiments

were conducted to compare the performance of proposed Enhanced Ant Colony System (EACS) using Energy Efficient Ant-Based Routing (EEABR) algorithm and Cost-aware Ant-Routing (SC) algorithm and the proposed algorithm is promising for implementation in static WSN systems.

Devibala *et al.* (2017) proposed a flow-based mitigation model with time variant snapshot (FBSD) for sinkhole detection and mitigation. Using the physical and geographical features of the nodes, the base station maintains the location details of the nodes that enables it to detect the presence of sink node in the network. NS-2 simulator was employed for implementation, the proposed method highly reduces over-head generated by flood of control messages.

Raghav *et al*. (2017) proposed a swarm intelligence algorithm, an Enriched Artificial Bee Colony Optimization (EABC) to observe and detect sink nodes in WSN, monitoring the position of estimated malicious nodes continuously. Implementing this algorithm as an evolutionary algorithm Using MATLAB for performance evaluation shows that with the intimation of the base station, the risk factor of the attacker node it known and hence the sinkhole discovered and its purpose.

Dharshini and Chinnaswamy (2017) proposed Artificial Bee Colony (ABC) technique for sink hole detection and compared it with an existing Enhanced Particle Swarm Optimization (EPSO) technique. They got better detection rate, false alarm rate, packet delivery ratio and average delay using NS-2 as simulator.

Wazid and Das (2016) designed a mechanism for sinkhole detection by first considering three types of sinkhole malicious node in WSN, the sinkhole modification node (SMD), sinkhole message dropping node (SDP), and sinkhole message delay node (SDL), then providing a detection scheme able to detect the different types of sinkhole node in a hierarchical wireless sensor network (HWSN). In this approach, the network was divided into clusters with high sensing nodes used as the cluster head (CH) responsible for detecting sinkhole node in the cluster. Using NS-2 for simulation gaining better performance in terms of detection rate.

Keerthana and Padmavathi (2016); these authors proposed Enhanced Particle Swarm Optimization and the technique tested in a simulated environment. This technique proved to have a better performance than the initial techniques Particle Swarm Optimization (PSO) and Ant Colony Optimization in areas of packet delivery ratio, detection rate and average delay.

Some of the gaps discovered, are false alarm rate problems that gave way to through put reduction, increase in end-to-end delay thereby impacting on the effectiveness in the detection of sinkhole attacks.

2.3 Findings from Literature

Researchers have employed different techniques ranging from swarm intelligence, to cryptographic to machine learning methods for sinkhole attack detection. In all of these existing techniques, swarm intelligence still remains the technique with the minimal trade-offs as WSN proves to be a fragile environment with need for a lot of monitoring, less computational overhead and high energy management.

ACO, a swarm intelligence technique is one of the most successful methods as it has been used to solve problems in different environments such as the travel salesman problem (TSP).

i. From the literature findings there still lies a need for new techniques or enhancement of already exiting methodology to improve attack detection and reduce the trade-offs on the WSN.
ii. Very importantly there is little emphasis on reduction of false alarm rate which highly sabotages the work done for attack detection.

CHAPTER THREE

3.0 METHODOLOGY

3.1 Research Process Framework

This proposed research comprises of first and second stages. The first stage involves problem formulation and planning, dataset description with design and stage two involves implementation which include Enhanced Ant Colony Optimization (EACO) detection on NS-3.29 simulator, flowchart and pseudocode.

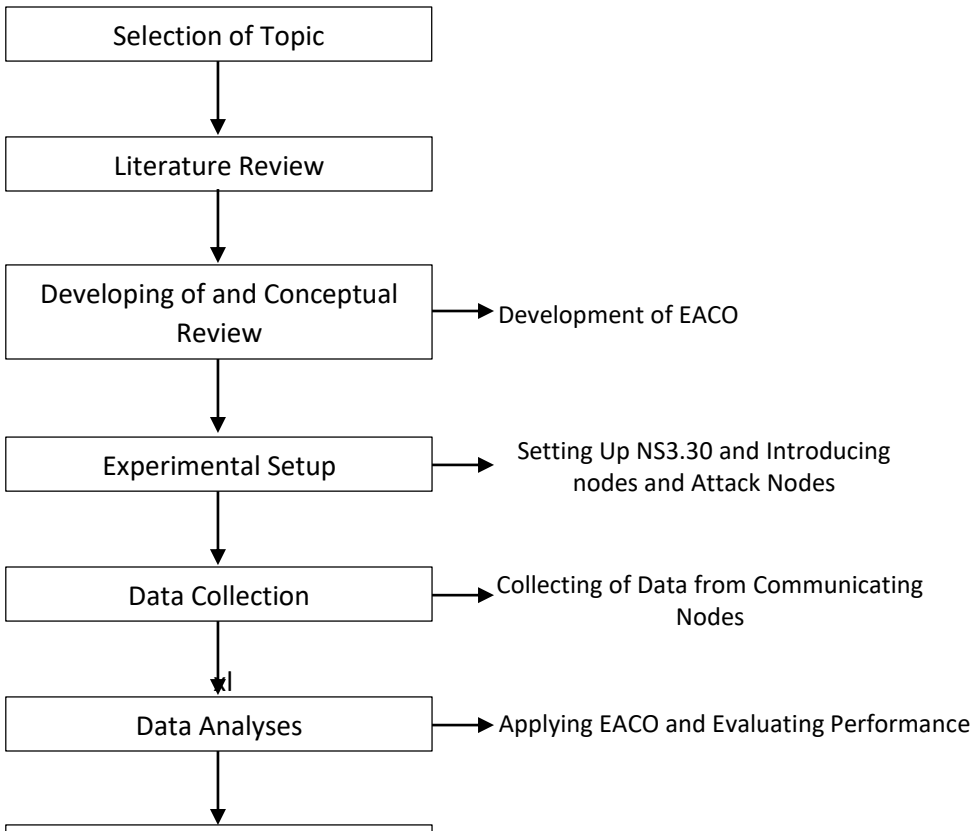

Figure 3.1 Research Process Framework

3.2 Research Methodology Flowchart

The methodology overview in figure 3.2 shows the major steps to be taken for the experimentation and implementation of the proposed enhancement.

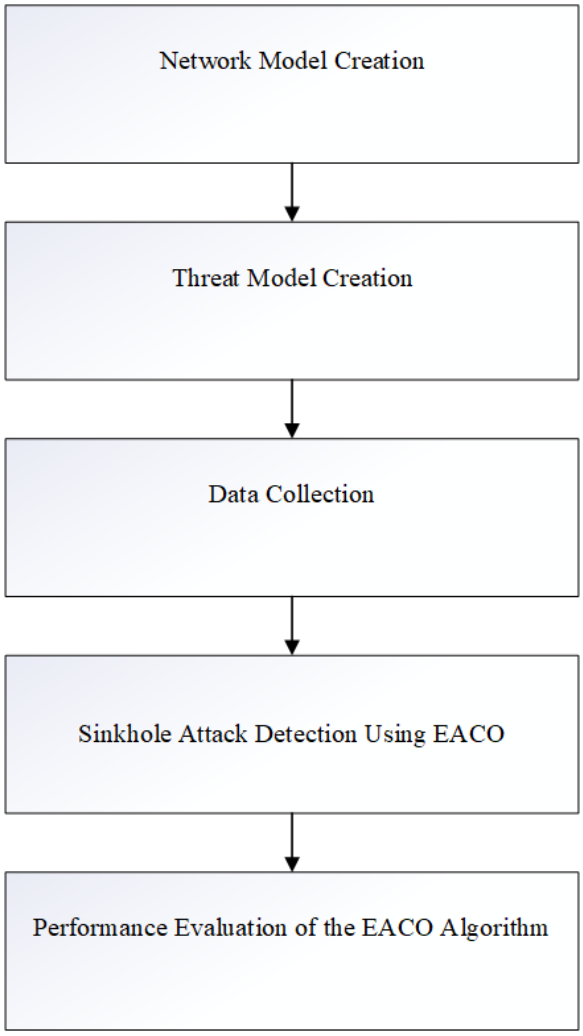

Figure 3.2 Proposed Methodology Overview

3.2.1 Network model creation:

This involves the simulation set up to the point of having an effective wireless sensor network running. In this experiment, Mac OS 10.15 Catalina is used, cloning NS3.30.1 from, the latest developer release version.

3.2.2 Threat model creation:

This stage involves the introduction of the attacker nodes in the network. This is done by calling the class written for attacker nodes in C++ in the simulator and the accessing of the damage to be caused by the attacker nodes

3.2.3 Sinkhole Attack detection using EACO:

At this stage, the algorithm is implemented. The EACO class is called in the C++ script hence, the attack detection begins. Data is then collated by the simulator and logged

3.2.4 Performance evaluation of the algorithm:

Here the result will be evaluated with the performance metrics.

Figure 3.3 clearly illustrates the proposed methodology process flow of the EACO. As the nodes communicates in the network, a list is generated known as the solution list from the communications between nodes in the wireless sensor network, a hash table is used as a pointer in the list and nodes with hash collision and saved in a collision list that makes up the suspect list that is passed from node to node for signing and voting upon which the attacker is discovered.

Unlike the regular ACO that passes the solution list to the suspect list. Figure 3.3 and 3.4 shows the difference in the process flow.

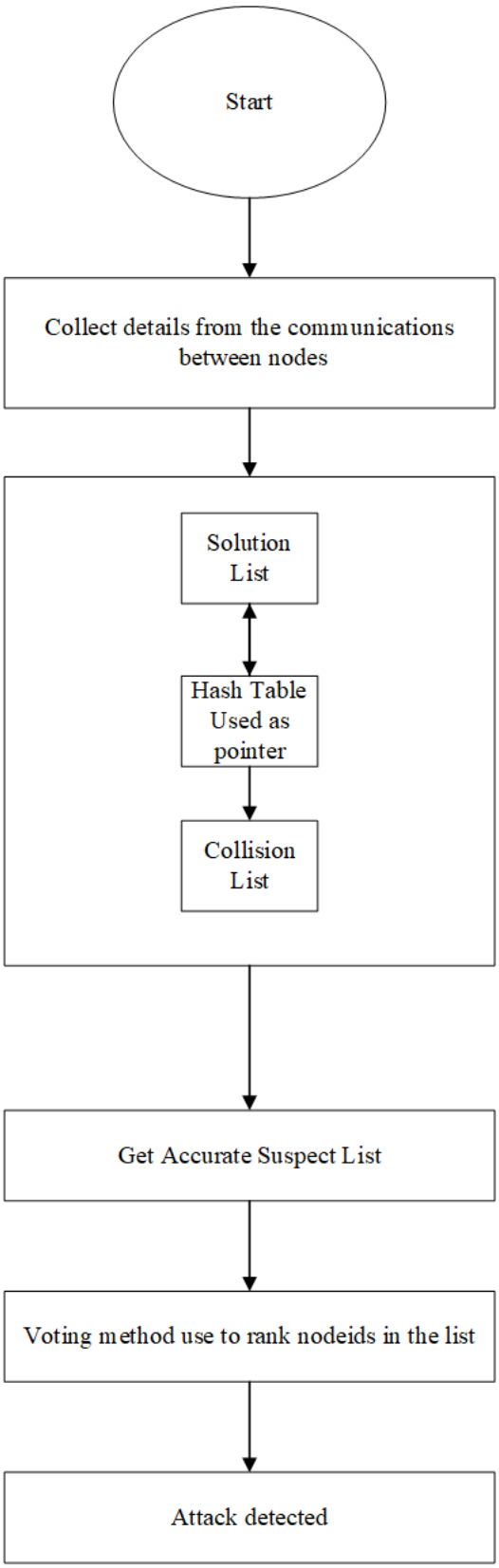

Figure 3.3 Process flowchart of the proposed EACO

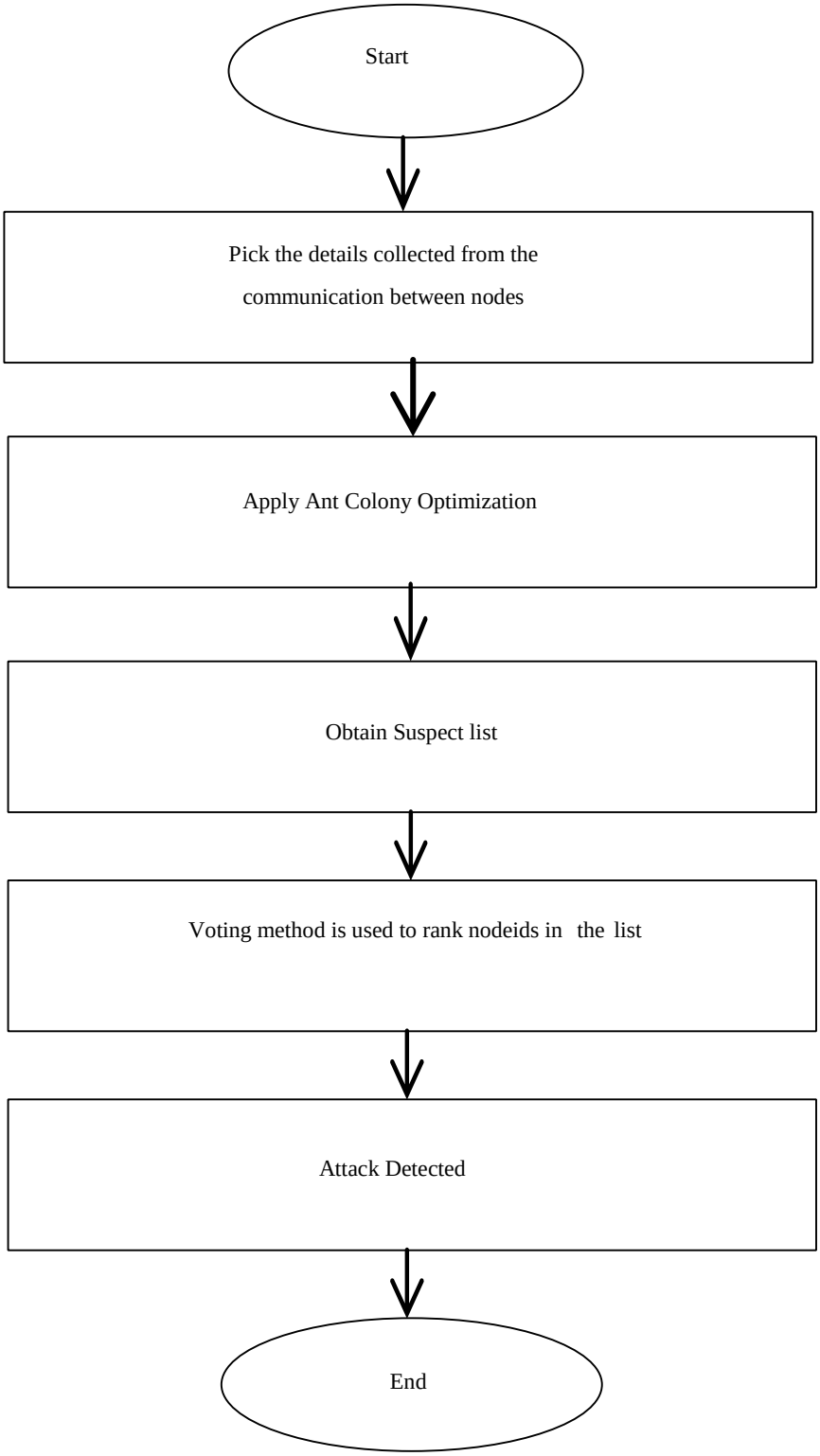

Figure 3.4 Flowchart For ACO

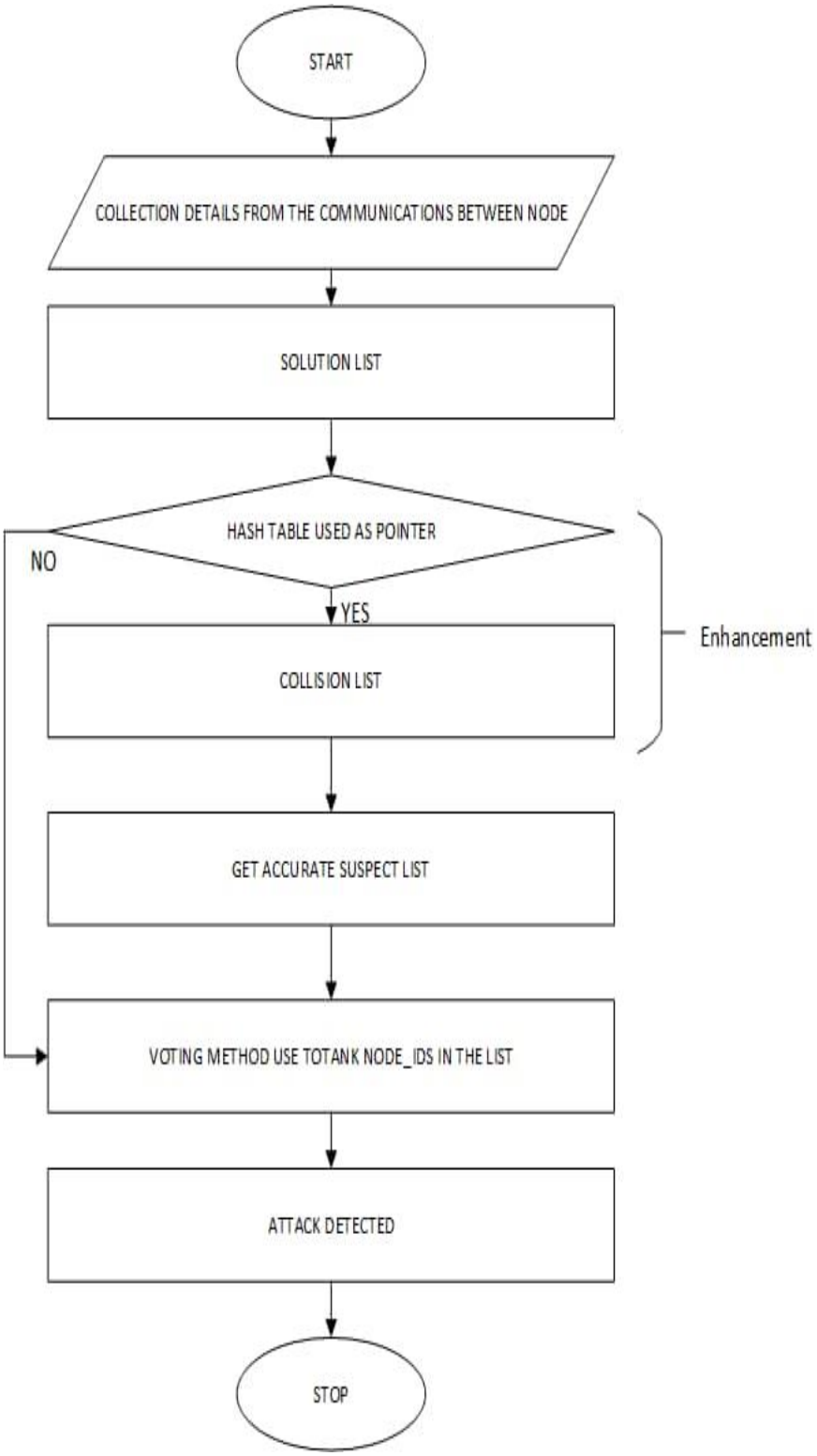

Figure 3.5 Flowchart For EACO

Table 3.1 **Pseudocode for EACO**

Procedure ACO ( )

```
{
Input n, α , β , ρ
set  the ant colony configuration
set the initial pheromone and heuristic value
get ant colony optimization system based on the calculated cost matrix
i = 1
while (I <= n)
{                r = 1
While ( r <= i)
{
Reset the ants
Build ant s' solution
Assume d(N)
// hash table implementation
if (h[d(N)]=0) { t=t+1 h[d(N)]=t and SL[t]= N //SL-Solution List
 return TRUE}
if (h[d(N)] ? 0 and SL[h[d(N)] ] = N) then return FALSE if h[d(N)] ? 0 and SL[h[d(N)] ] ? N { if N
? CL {cl = cl+1
//CL-Collision List
CL[cl] = N
return TRUE
}else return FALSE
initiate local search
Update path best for i
Update pheromones
r = r + 1
}
Choose  path best for i
i = i + 1
}
}
```

A lot of research has been done on sinkhole attacks and several techniques have been proposed for detecting sinkhole attack in WSN. Wireless sensor network has various optimization problems in design, computational complexity, deployment and security management. To handle these problems, heuristic methods are employed. One of these heuristic methods is Swarm Intelligence like the EACO, which has proved effective in threat detection in wireless sensor network.

Swarm Intelligence (SI) as one of the effective methods against sinkhole attacks uses collective behavior of decentralized, self-organized systems, artificial and natural. It has many advantages that include Robustness, Flexibility, Scalability and its self-organized and decentralized feature. The two popular Swarm Intelligence inspired methods are Ant Colony Optimization (ACO) and Particle Swarm Optimization.

3.3 Performance Evaluation

The parameters used to evaluate the performance of sinkhole attack detection techniques are as follows; Detection Rate (DR), False Alarm Rate (FAR), Packet Delivery Ratio (PDR), Message Drop and Average Delay. These parameters clearly show the behavior of the WSN network under different situations.

(1) Detection Rate: Detection rate is defined as the percentage of correct attacks detected by the total number of attacks present in the network. The formula to estimate the detection rate is,

$$\text{Detection Rate} = \frac{\text{Number of attacks detected}}{\text{Total number of attacks present}} \text{ x } 100 \qquad (3.1)$$

(2) False Alarm Rate: False alarm rate is the ratio between numbers of attacks not detected to the total number of attacks in the network.

$$\text{False Alarm Rate} = \frac{\text{Number of attacks} - \text{Number of attacks correctly found Total number of attacks}}{\text{Total number of attacks}} \text{ x } 100 \qquad (3.2)$$

(3) Packet Delivery Ratio: Packet Delivery Ratio is defined as the percentage of number of received packets and the total number of sent packets.

$$\text{Packet Delivery Ratio} = \frac{\text{Number of Packets received}}{\text{Number of packets sent}} \text{ x } 100 \qquad (3.3)$$

(4) Message Drop: Message drop is defined as the ratio between number of messages not received to the total number of messages.

$$\text{Message Drop} = \frac{\text{Total No.of Messages} - \text{No. of Messages recieved}}{Total\ number\ of\ messages} \text{ x } 100 \qquad (3.4)$$

(5) Average Delay: Average delay is defined as the ratio between sum of all packets delayed to the total number of packets received.

$$\text{Average Delay} = \frac{\text{Sum of all Packets delay}}{Total\ number\ of\ recieved\ packets} \text{ x } 100 \qquad (3.5)$$

3.4 Simulation Environment

The EACO was simulated on a Mac OS Catalina platform using NS3.30.1 simulator. Deployment area of 700 x 300 m^2 was used. Within this area, 300 nodes was deployed. Table 3.7 shows the simulation parameters used in the experiment. The c++ code for the enhancement was written as a class and called when the program initiates.

Table 3.7 **Simulation parameters**

Parameter	Description
Platform	Mac OS Catalina
Deployment area	700 x 300 m^2
Network topology	tree
Network size	250 nodes
Number of attacker nodes	50
Simulation time	1900 seconds
Traffic type	CBR/UDP
Packet size	512 bytes
Packet transmission rate	25 Kbps
Routing protocol	AODV
Medium access control type	IEEE 802.11
Communication range of sensor node	25 m
Communication range of cluster head	50 m

3.5 Simulation Scenarios

Simulation under normal WSN flow was carried out first, under sinkhole attack and then include the EACO technique.

i. **WSN scenario in normal flow:** This scenario consists of 200 nodes in normal flow sending and receiving and transmitting to the base station with negligible end-to-end delay and packet drop ration.

ii. **WSN scenario under sinkhole attack:** this scenario again will consist of 200 nodes with 50 additional attacker nodes and observing the parameters change

iii. **WSN scenario under attack and EACO implemented:** here the new attack detection is implemented and parameters observed.

CHAPTER FOUR

4.0 RESULTS AND DISCUSSION

4.1 Results

From the simulations, the following statistics of the network were computed: end-to-end delay (in ms), detection rate (DR), packet delivery ratio (PDR), throughput (in kps), and false alarm/false positive (FPR).

i. **Effect on end-to-end delay:** End-to-end delay (EED) is given as the time it takes for the data packets to arrive at the BS. Figure 4.1 shows the EED (in ms) for normal flow, when attacked by a sinkhole and when our EACO is implemented. The value for the EED is 70.06ms while under sinkhole attacks it is 736.66 as the effect of traffic diversion from the original destination and 153.46ms when EACO is implemented.

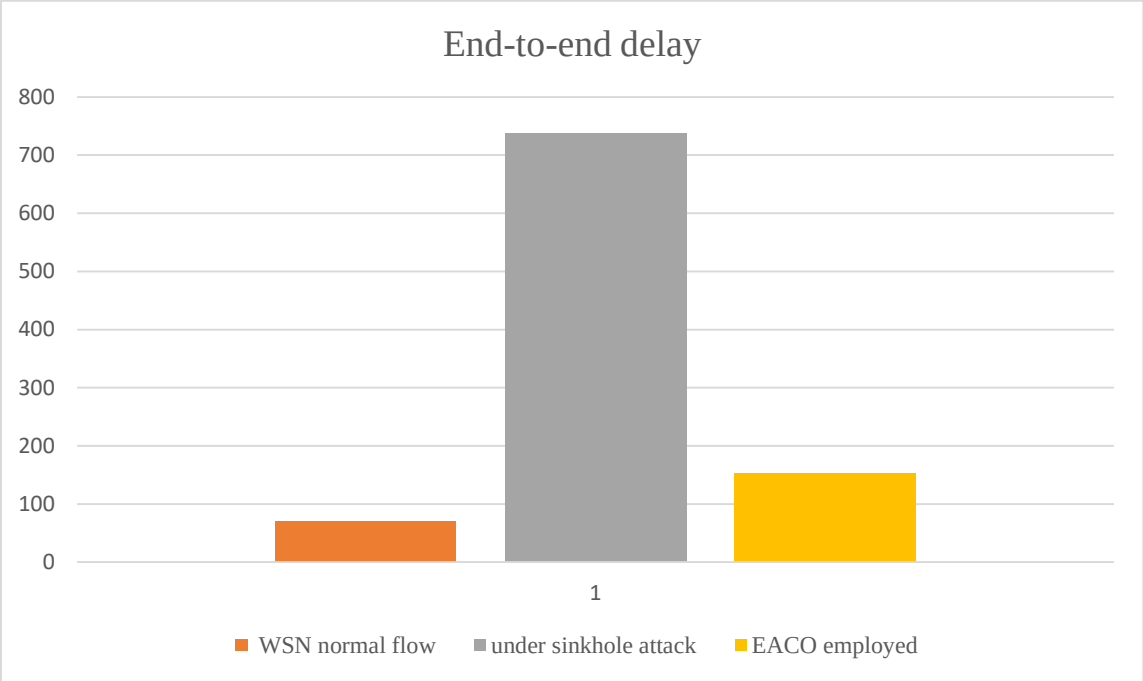

Figure 4.1 Simulation result s of end-to-end delay

ii. **Effect on Packet delivery ratio**: this is the ratio of the number of packets received at the BS to number of packets sent by the source nodes. Figure 4.2 shows PDR ration for normal WSN flow, while under sinkhole attack and under our EACO technique are 0.93,0.46 and 0.9 respectively. Apparently, there is a significant improvement on the PDR.

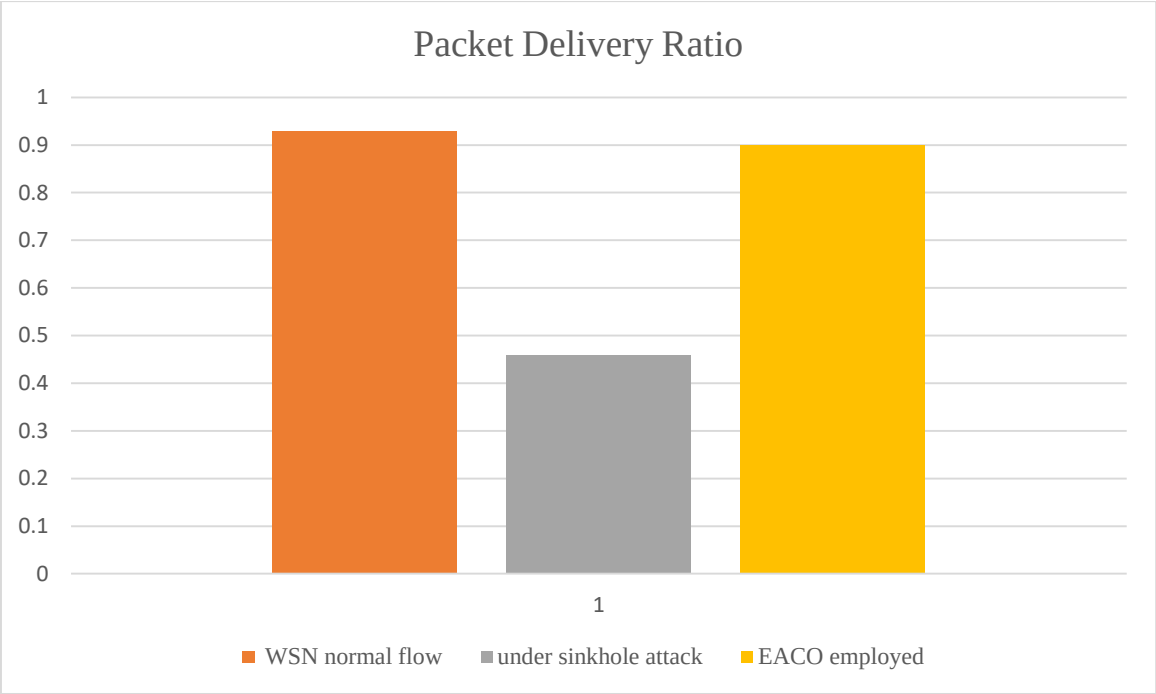

Figure 4.2 Simulation results of packet delivery

iii. **Effect on throughput:** this is the number of transmitted bits per unit time in the network. As shown in figure 4.3, the network throughput (in kps) in normal WSN flow, while under sinkhole attack and when EACO is implemented. Getting values of throughput for normal WSN flow is 9.2 kps whereas under attack it is 4.25 and under the EACO it is 8.72. Hence an increase of throughput by 94.32% under our technique.A summary of the results is shown in table 5.1.

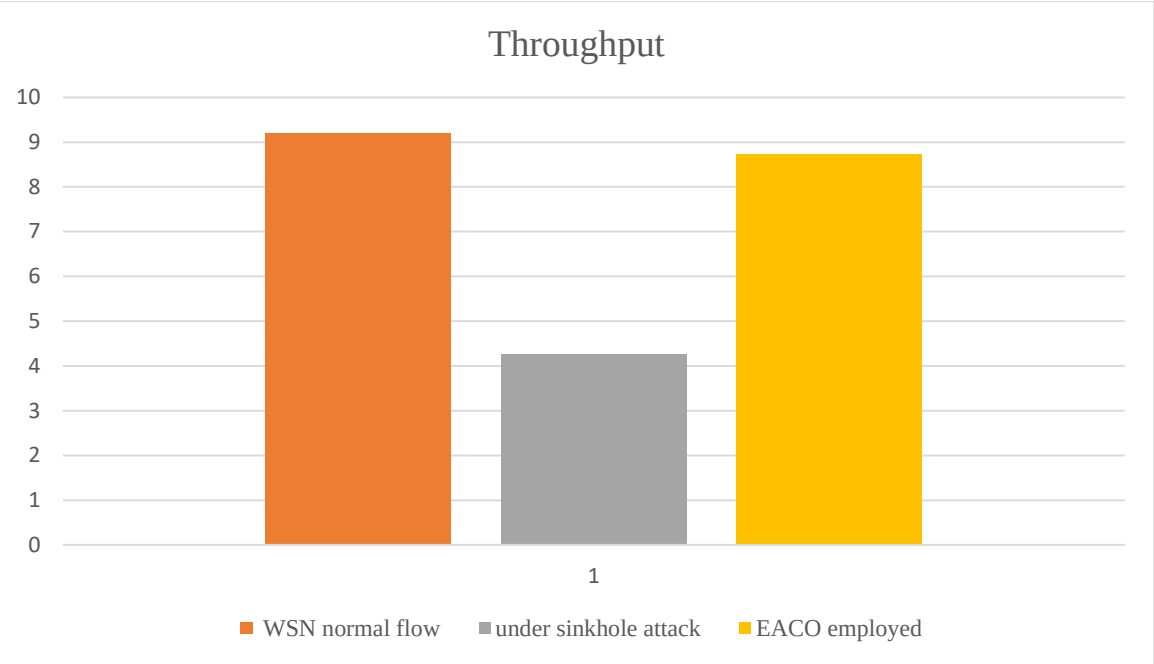

Figure 4.3 Network throughput result

Table 4.1 **Statistics summary**

Parameter	WSN normal flow	under sinkhole attack	EACO employed
Throughput (kbps)	9.2	4.25	8.72
End-to-end delay (ms)	70.06	736.66	153.46
Packet delivery ratio	0.93	0.46	0.9

iv. **Effect of detection rate and false alarm/false positive rate:** these are the parameters of focus in our technique. Detection rate (DR) which also is the true positive rate (TPR) and the false alarm rate which is also known as the false positive rate (FPR) were measured in the simulation. These parameters show the effectiveness of sinkhole detection. Where TP represents the number of true positives, whereas, TN the number of true negatives, FN is number of false

negatives, and FP the number of false positives. There for Detection Rate (DR) is gotten as the number of sinkhole attackers detected by the technique divided by the total number of sinkhole attackers present in the simulation Kasliwal *et al*. (2014); Sun *et al.* (2015) and given below as

$$DR = \frac{TP}{TP+FN}$$

(4.1)

While FPR on the other hand is defined by the number of falsely detected nodes presumed to be attacker nodes (Kasliwal *et al*., 2014)

$$FPR = \frac{FP}{TN+FP}$$

(4.2)

The following where observed during the simulation:

I. With the confusion matrix in Table 4.2, the EACO technique detected a total of 48 true positive (TP) attacker nodes (real attackers), and 2 false positive (FP) nodes (legitimate nodes), with 198 true negatives (TN) nodes, (legitimate nodes) and 2 false negative (FN) (an attacker mistaken for a legitimate node).

II. Hence, in this research a total of 50 sinkhole attacker nodes and 200 legitimate nodes where used in the simulation and a detection rate (DR) of 96% was achieved with a false positive rate (FPR) 1.0%.

Table 4.2 **Confusion Matrix**

		Actual value	
		No. of positives	No. of negatives
Predicted value	No. of positives	TP:48	FP:2

	No. of negatives	FN:2	TN:198

4.2 Performance Evaluation

Here, performance comparison of our technique with other related works such as Dharshini and Chinnaswamy (2017), Wazid and Dass (2016), Keerthana and Padmavathi (2016) was carried out. From the comparison shown in Table 4.3 and figure 4.4, apparently the EACO technique performs better.

Table 4.3 Accuracy Comparison

Authors	Detection (%)	False alarm (%)
Wazid and Das (2016)	95	1.25
(Keerthana and Padmavathi, 2016)	87.062	10.648
Dharshini and Chinnaswamy (2017)	53	18
(Nadeem and Alghamdi, 2019)	90	10
Our Technique	96	1.0

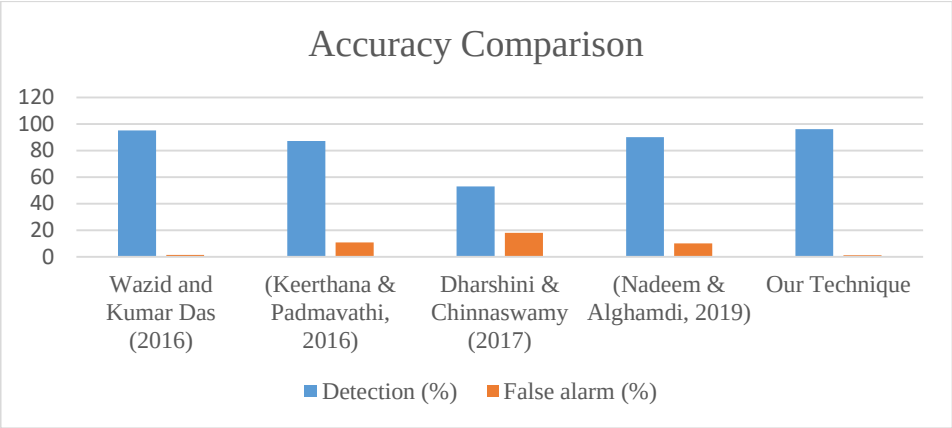

Figure 4.4 Accuracy Comparison

CHAPTER FIVE

5.0 CONCLUSION AND RECOMMENDATIONS

The summary of the experiment, the major contributions and a highlight of further research directions to help secure wireless sensor network from every level of attack. Time taken to design, code and implement gave way to very important findings.

5.1 Conclusion

In this research, ACO was enhanced for better sinkhole attack detection with the following contributions; The design of an enhanced ant colony optimization algorithm for sinkhole attack detection in WSN. The implementation of the enhanced ant colony optimization technique EACO in a simulation and an increased sinkhole attack detection rate of 96% and minimizes false alarm rate to 1% was gotten. A performance evaluation of the enhanced ant colony optimization technique by comparing the detection rate and false alarm rate with related works such as (Nadeem and Alghamdi, 2019), Keerthana and Padmavathi (2016).

In conclusion, the enhanced ant colony optimization detection technique performed better in sinkhole detection in a wireless sensor network with a detection rate of 96%. Further research in wireless sensor network security protocols is highly necessary to enable an even better detection while saving energy and time of operation.

5.2 Recommendations

In future research, attention on how to improve security on sensors encryption would be a very great area as sensors are everywhere in everything. With the problem of low compute power of these tiny sensors, lightweight algorithms can be gotten from hybridizing the known available. Thus, enhancing them for the peculiarity of the sensor environment and their low power requirement.

5.3 Contribution to knowledge

In the first contribution an algorithm was designed enhancing the known ant colony optimization algorithm by including a hash table; which is the is the indexing od known nodes to avoid collision that leads to false alarm. This reduces time taken in identifying the attacker with use of indexing that creates a more accurate suspect list hence reducing the false alarm rate in the attacker detection. In the second contribution the algorithm was developed using C++ and simulated a scenario using NS3.30.1 on a mac OS Catalina platform. Introducing legitimate nodes and attacker nodes enable traffic to be exchanged allowing us to collect data.

5.4 Published articles

Nwankwo, K. E. (2019, October). Sinkhole Attack Detection in A Wireless Sensor Networks using Enhanced Ant Colony Optimization to Improve Detection Rate. In *2019*

2nd International Conference of the IEEE Nigeria Computer Chapter (NigeriaComputConf) (pp. 1-6). IEEE.

Nwankwo *et al.,* (2021, May). A Panacea to Soft Computing Approach for Sinkhole Attack Classification in a Wireless Sensor Networks Environment. Futuristic Trends in Network and Communication Technologies (pp.78-87)

REFERENCE

C. Zhang, R. Lu, X. Lin, P.-H. Ho, & X. S. Shen. (2008) An Efficient Identity-Based Batch Verification Scheme for Vehicular Sensor Networks. *In 27th IEEE International Conference on Computer Communications (INFOCOM),* (pp. 816-824). Phoenix, AZ, USA: IEEE.

Devibala, K., Balamurali, S., Ayyasamy, A., & Archana, M. (2017). Flow based mitigation model for sinkhole attack in wireless sensor networks using time-variant snapshot. *International Journal of Advances in Computer and Electronics Engineering*, *2*(5), 14–21. Retrieved from http://www.ijaceeonline.com/VOL2ISS5/

Dharshini, Y. N., & Chinnaswamy, C N. (2017). Swarm Intelligence Technique For Sinkhole Attack Detection In Wireless Sensor Network Performance Comparison of the Algorithms. *International Journal of Advance Research and Innovative Ideas in Education*, 3(4), 647–656. Retrieved from http://ijariie.com/AdminUploadPdf/

Dorigo, M., & Gianni, D. C. (1992). Ant Colony Optimization: A New Meta-Heuristic. *In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406),* (pp1470–1477). Washington, DC, USA: IEEE.

H. Ghasemzadeh & R. Jafari. (2011). Physical movement monitoring using body sensor networks: A phonological approach to construct spatial decision trees. *IEEE Transactions on Industrial Informatics*, 7(1), 66–77.

I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, & E. Cayirci. (2002). Wireless sensor networks: A survey. *Computer Networks*, 38(4), 393–422.

Iwendi, C., Zhang, Z., & Du, X. (2018). ACO based key management routing mechanism for WSN security and data collection. *Proceedings of the IEEE International Conference on Industrial Technology*,(pp1935–1939). Lyon, France: IEEE

Jamal, H., Nasir, A., & Ku-mahamud, K. R. (2019). Parameter adaptation for ant colony system in wireless sensor network. *Journal of Information and Communication Technology, 2*(2), 167–182. https://doi.org/10.32890/jict2019.18.2.8286

Jangra, R., & Kait, R. (2017). Principles and concepts of wireless sensor network and ant colony optimization: A Review. *International Journal of Advanced Research in Computer Science, 8*(5), 1180-1191. Retrieved from https://www.researchgate.net/profile/Utsav-Kakkad-2/post/

Karlof, C., Sastry, N., & Wagner, D. (2004). TinySec. *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems,* (pp.162–175). New York, NY, USA: ACM

Kasliwal, B., Bhatia, S., Saini, S., Thaseen, I. S., and Kumar, C. A. (2014). A hybrid anomaly detection model using G-LDA. *Souvenir of the 2014 IEEE International Advance Computing Conference,* (pp288–293). Gurgaon, India: IEEE

Kaur, K., Kaur, P., & Singh, E. S. (2014). Wireless sensor network: Architecture, design issues and applications. *International Journal of Scientific Engineering and Research*, *2*(11), 6–10. Retrieved from https://www.ijser.in/archives/v2i11/

Kazem, S., Daniel, M., & Taieb, Z. (2007). *Wireless Sensor Networks Technology, Protocols, and Applications*. New Jersey USA: John Wiley & Sons, INC.

Keerthana, G., & Padmavathi, G. (2016). Detecting sinkhole attack in wireless sensor network using enhanced particle swarm optimization technique. *International Journal of Security and Its Applications*, *10*(3), 41–54. https://doi.org/10.14257/ijsia.2016.10.3.05

Kocher, P., Jun, B., & Jaf, J. (1999). Di erential power analysis. *Analysis*, *1666*, 387–397. https://doi.org/10.1007/3-540-48405-1_25

Messerges, T. S., Dabbish, E. A., & Sloan, R. H. (2002). Examining smart-card security under the threat of power analysis attacks. *IEEE Transactions on Computers, 51*(5), 541–552. https://doi.org/10.1109/TC.2002.1004593

Mohammad Wazid, Ashok Kumar Das, S. K. & M. K. K. (2016). Design of sinkhole node detection mechanism for hierarchical wireless sensor networks. *Wiley Online Library*, *9*(19), 4596–4614. https://doi.org/10.1002/sec

Mohammadi, S., & Jadidoleslamy, H. (2011). A comparison of link layer attacks on wireless sensor networks. *International Journal on Applications of Graph Theory In Wireless Ad Hoc Networks And Sensor Networks*, *3*(1), 35–56. https://doi.org/10.5121/jgraphhoc.2011.3103

Nadeem, A., & Alghamdi, T. (2019). Detection algorithm for sinkhole attack in body area sensor networks using local information. *International Journal of Network Security 21*(4), 670–679. https://doi.org/10.6633/IJNS.201907 21(4).16

Nasir, H. J. A., Ku-Mahamud, K. R., & Kamioka, E. (2018). Enhanced ant colony system for reducing packet loss in wireless sensor network. *International Journal of Grid and Distributed Computing*, *11*(1), 81–88. https://doi.org/10.14257/ijgdc.2018.11.1.08

Ngai, E. C. H., Liu, J., & Lyu, M. R. (2006). On the Intruder Detection for Sinkhole Attack in Wireless Sensor Networks. *IEEE* (pp. 3383–3389. Istanbul, Turkey: IEEE

Nithiyanandam, N., Parthiban, P. L., Rajalingam, B., & Scholar, R. (2018). Effectively suppress the attack of sinkhole in wireless sensor network using enhanced particle swarm optimization Technique. *International Journal of Pure and Applied Mathematics* 118(9), 313–329. Retrieved from http://www.ijpam.eu

P. Kumar, A. Gurtov, J. Iinatti, M. Ylianttila, & M. Sain. (2016) Lightweight and secure session-key establishment scheme in smart home environments. *IEEE Sensors Journal*, 16(1):254–264.

Raghav, R. S., Pothula, S., & Ponnurangam, D. (2017). An enriched artificial bee colony (EABC) algorithm for detection of sinkhole attacks in Wireless Sensor Network. *International Journal of Mechanical Engineering and Technology*, *8*(8), 193–202. Retrieved from https://iaeme.com/MasterAdmin/Journal_uploads/IJMET/

R. S. Sachan, M. Wazid, D. P. Singh, & R. H. Goudar. (2013) Different Attacks happen at Various Layers of WSN Stack with a Case Study of Misdirection Attack. *In International Conference on Human Computer Interactions.* (pp. 1–6).

S. Jiang, X. Zhu, & L. Wang (2016). An Efficient Anonymous Batch Authentication Scheme Based on HMAC for VANETs. *IEEE Transactions on Intelligent Transportation Systems,* 17(8),2193–2204

Sanjeev Kumar, G., & Poonam, S. (2013). Overview of survey on security of wireless sensor network. *International Journal of Advanced Research in Computer and Communication Engineering* . *3*(1), 5201–5207. Retrieved from https://ijarcce.com/wp-content/uploads/2012/03

Sun, X., Yan, B., Zhang, X., & Rong, C. (2015). An integrated intrusion detection model of cluster-based wireless sensor network. *PLoS One*, *10*(10), 1–16. https://doi.org/10.1371/journal.pone.0139513

Yang, S. (2014). *Principle of Wireless Sensor Networks*. UK, London: Springer . https://doi.org/10.1007/978-1-4471-5505-8

Yick, J., Mukherjee, B., & Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks*, *52*(12), 2292–2330. https://doi.org/10.1016/j.comnet.2008.04.002

Zhao, J., Cheng, D., & Hao, C. (2016). An improved ant colony algorithm for solving the path planning problem of the omnidirectional mobile vehicle. *Mathematical Problems in Engineering*. (pp. 1–10). https://doi.org/10.1155/2016/7672839

Zhou, Yun & Fang, Yuguang & Zhang, Y. (2008). Securing wireless sensor networks: a survey. *IEEE Communications Surveys &. Tutorials. 10*(3), 6–29.

APPENDIX A

Table 2.1 **Summary of Reviewed Literature**

Reference	Method	Metric/Parameter	Result	Comparison	Problem Solved	Limitations/Gaps
(Nadeem and Alghamdi, 2019)	A detection algorithm using data aggregation algorithm	Throughput, latency, packet breakdown	Good performance in terms of high success (85% on average) in sinkhole detection		Effective detection of sinkhole attack in BAN	Limited to body area network scenarios. Cannot be applied in multi BAN scenario and in terms of privacy and security incases of wearable shimmer sensors
(Nithiyanandam *et al*., 2018)	Enhanced Particle Swarm Optimization Technique	Detection Rate (DR), False Alarm Rate (FAR), Packet Delivery Ratio (PDR), Message Drop and Average Delay	Efficient detection of the Sinkhole attacks	ACO and PCO	Improved sinkhole attack detection rate over PSO	Detection is limited to velocity and position of node

(Devibala *et al*., 2017)	Flow Based Mitigation Model using Time-Variant Snapshot	Overhead, Throughnb put performance, Packet delivery fraction, Average end-to-end delay	High reduction in overhead in the network and increases the performance	Leader based and G-Hazard	Sinkhole detection with reduced control overhead	Sinkhole detection is limited to only traffic pattern
(Raghav *et al*., 2017)	Enriched Artificial Bee Colony (EABC) Algorithm	Average Energy Consumed for NL, Energy consumed by SH and PDR for NL	Better detection	ACO and PSO	Reduced time for sinkhole detection	High cost computational
(Dharshini and Chinnaswamy, 2017)	Swarm Intelligence Technique using ABC Algorithm	Detection rate, False Alarm rate, Packet delivery ratio, Message drop and Average delay	Significant increase in detection rate	EPSO and PSO	Slight improvement in packet delivery ratio and decrease in false alarm	High cost computational
Rashmi *et al*. [2016]	Secured and Intelligent Multipath Routing using AOMDV Algorithm	delivery ratio, Average delay		Compared the results of RSA and Diffie Hellman Algorithms in term of time	Better routing	Energy loss

(Keerthana and Padmavathi, 2016)	Enhanced Particle Swarm Optimization (EPSO) technique for intrusion detection	Detection rate, False Alarm rate, Packet Delivery ratio, Message Drop, Average delay	Enhanced Particle Swarm Optimization (EPSO) technique is more effective than particle swarm optimization (PSO) due to the use of hash table.	ACO and PSO	Improved detection	
Zhang *et al.* (2014)	Redundancy mechanism Algorithm	Detection Rate	Better detection rate	classical detection algorithm- B	Detection of sinkhole attack	High cost computational overhead for onlu one parameter improvement
Shafiei *et al.* (2014)	Energy holes estimation using geostatistical hazard model	Number of hops, threshold and number of monitors	decrease in false positive detection	Varying monitor nodes	Detection of sinkhole attack	High Energy use ,low detection rate and false positive rate in congested areas
Nandana *et al* (2014)	Enhancement AODV protocol algorithm			AODV algorithm	Improved detection rate archived	Computational overhead

(Ngai, Liu, and Lyu, 2006)	Proposed Novel Algorithm for Intruder detection	FN,FP, Intruder Detection rate	More effective and accurate in intruder detection, plus enhancements to deal with cooperative malicious nodes that attempt to hide the intruder.	Numerical Analysis and Simulations	communication and computation overheads are reasonably low for wireless sensor networks.	Identifying data inconsistency, and thus correctly locating suspected nodes in sinkhole attack

APPENDIX B

C++ codes for EACO

```
#include "ns3/aodv-module.h"
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/mobility-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/wifi-module.h"
#include "ns3/v4ping-helper.h"
#include "ns3/netanim-module.h"
#include "scratch/routing-algos/ken-e-ant-colony/ant-colony-helper.h"
#include <iostream>
#include <cmath>

using namespace ns3;
```

```
/**
 * \brief Stationary 4x4 Matrix using AODV script.
 *
 * This script creates 2-dimensional grid topology and then ping last node from the
first one:
 *
 * [10.0.0.1] <-- step --> [10.0.0.2] <-- step --> [10.0.0.3] <-- step -->
[10.0.0.4]
 *
 * ping 10.0.0.4
 */
class StationaryMatrix
{
  public:
    StationaryMatrix ();
    /// Configure script parameters, \return true on successful configuration
    bool Configure (int argc, char **argv);
    /// Run simulation
    void Run ();
    /// Report results
    void Report (std::ostream and os);

  private:
    ///\name parameters
    //\{
    /// Number of nodes
    uint32_t size;
    /// Distance between nodes, meters
    double step;
    /// Simulation time, seconds
    double totalTime;
    /// Write per-device PCAP traces if true
    bool pcap;
    /// Print routes if true
    bool printRoutes;
    //\}

    ///\name network
    //\{
    NodeContainer nodes;
    NetDeviceContainer devices;
    Ipv4InterfaceContainer interfaces;
    //\}

  private:
    void CreateNodes ();
    void CreateDevices ();
    void InstallInternetStack ();
    void InstallApplications ();
};

int main (int argc, char **argv)
{
  StationaryMatrix sim;
  if (!sim.Configure (argc, argv))
    NS_FATAL_ERROR ("Configuration failed. Aborted.");
```

```
  sim.Run ();
  sim.Report (std::cout);
  return 0;
}

//-----------------------------------------------------------------------------
StationaryMatrix::StationaryMatrix () :
  size (16),
  step (100),
  totalTime (10),
  pcap (true),
  printRoutes (true)
{
}

bool StationaryMatrix::Configure (int argc, char **argv)
{

  SeedManager::SetSeed (12345);
  CommandLine cmd;

  cmd.AddValue ("pcap", "Write PCAP traces.", pcap);
  cmd.AddValue ("printRoutes", "Print routing table dumps.", printRoutes);
  cmd.AddValue ("size", "Number of nodes.", size);
  cmd.AddValue ("time", "Simulation time, s.", totalTime);
  cmd.AddValue ("step", "Grid step, m", step);

  cmd.Parse (argc, argv);
  return true;
}

void StationaryMatrix::Run ()
{
  CreateNodes ();
  CreateDevices ();
  InstallInternetStack ();
  InstallApplications ();

  std::cout << "Starting simulation for " << totalTime << " s ...\n";

  /* NetAnim XML output
  AnimationInterface anim ("animation.xml");
  anim.SetMobilityPollInterval(Seconds(0.25));
  anim.EnablePacketMetadata(true);
  */

  Simulator::Stop (Seconds (totalTime));
  Simulator::Run ();
  Simulator::Destroy ();
}

void StationaryMatrix::Report (std::ostream and)
{
}

void StationaryMatrix::CreateNodes ()
{
  std::cout << "Creating " << (unsigned)size << " nodes " << step << " m apart.\n";
```

```
  nodes.Create (size);
  // Name nodes
  for (uint32_t i = 0; i < size; ++i)
    {
      std::ostringstream os;
      os << "node-" << i;
      Names::Add (os.str (), nodes.Get (i));
    }
  // Create static grid
  MobilityHelper mobility;
  mobility.SetPositionAllocator ("ns3::GridPositionAllocator",
    "MinX", DoubleValue (0.0),
    "MinY", DoubleValue (0.0),
    "DeltaX", DoubleValue (step),
    "DeltaY", DoubleValue (step),
    "GridWidth", UintegerValue (4),
    "LayoutType", StringValue ("RowFirst"));
  mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
  mobility.Install (nodes);
}

void StationaryMatrix::CreateDevices ()
{
  NqosWifiMacHelper wifiMac = NqosWifiMacHelper::Default ();
  wifiMac.SetType ("ns3::AdhocWifiMac");
  YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default ();
  YansWifiChannelHelper wifiChannel = YansWifiChannelHelper::Default ();
  wifiPhy.SetChannel (wifiChannel.Create ());
  WifiHelper wifi = WifiHelper::Default ();
  wifi.SetRemoteStationManager ("ns3::ConstantRateWifiManager", "DataMode",
StringValue ("OfdmRate6Mbps"), "RtsCtsThreshold", UintegerValue (0));
  devices = wifi.Install (wifiPhy, wifiMac, nodes);

  if (pcap)
    {
      wifiPhy.EnablePcapAll (std::string ("aodv"));
    }
}

void StationaryMatrix::InstallInternetStack ()
{
  AntColonyHelper aodv;
  InternetStackHelper stack;
  stack.SetRoutingHelper (aodv);
  stack.Install (nodes);
  Ipv4AddressHelper address;
  address.SetBase ("10.0.0.0", "255.0.0.0");
  interfaces = address.Assign (devices);

  if (printRoutes)
    {
      Ptr<OutputStreamWrapper> routingStream = Create<OutputStreamWrapper>
("aodv.routes", std::ios::out);
      aodv.PrintRoutingTableAllAt (Seconds (8), routingStream);
    }
}

void
```

```
StationaryMatrix::InstallApplications ()
{
  V4PingHelper ping (interfaces.GetAddress (size - 1));
  ping.SetAttribute ("Verbose", BooleanValue (true));

  ApplicationContainer p = ping.Install (nodes.Get (0));
  p.Start (Seconds (0));
  p.Stop (Seconds (totalTime) - Seconds (0.001));
}
```