IST-Africa 2016 Conference Proceedings
Paul Cunningham and Miriam Cunningham (Eds)
IIMC International Information Management Corporation, 2016
ISBN: 978-1-905824-55-7

A Security Algorithm for Wireless Sensor Networks in the Internet of Things Paradigm

Ts'itso MAPHATS'OE[1], Muthoni MASINDE[2]
Central University of Technology, Free State,
Private Bag X20539, Bloemfontein, 9300, South Africa
[1]*Tel: +266 58420807, Fax: + 27 51 5073653, tsitsomaphatsoe@gmail.com:*
[2]*Tel: + 2751 5073091, Fax: +27 51 5073653, muthonimasinde@yahoo.com*

Abstract: In this paper we explore the possibilities of having an algorithm that can protect Zigbee wireless sensor networks from intrusion; this is done from the Internet of Things paradigm. This algorithm is then realised as part of an intrusion detection system for Zigbee sensors used in wireless networks. The paper describes the algorithm used, the programming process, and the architecture of the system developed as well as the results achieved.

Keywords: Intrusion Detection System (IDS), Wireless Sensor Networks (WSN), Over the Air Programming (OTA)

1. Introduction

Wireless sensor networks (WSNs) have made it easy to visualize and implement a variety of novel ideas; this is partly because of their flexibility nature and small portable size. The range of applications of WSNs are inclusive of simple every day uses such as automated doors, to life critical scenarios such as detection of natural disasters and possible overflow of a dam and even military applications, where they are often deployed to gather information in hostile environments [1]. They can also be used to combat drought [2] which could be an advantage for agriculture and farming [3].

The term Internet of Things (IoT), was first used in a presentation in 1999 by Proctor and Gamble employee, Kevin Ashton, when linking the Radio Frequency Identifier (RFID) to the Internet [4]. It has now evolved to describe a sector of information technology and research aimed at connecting all objects, electronic or living organisms, to the Internet [5], with the aim of efficient communication of all interconnected objects. WSNs are a crucial aspect in the realization of the IoT implementation because sensor devices are small, portable and can be implemented in a variety of applications.

It is imperative therefore to conclude that WSNs have to remain as secure as possible from any form of intrusion or security breaches [6] when being used in applications. This would bring about an enhanced sense of acceptance and security to end users entering the era of the Internet of Things (IoT). These users should be confident that the information they receive from WSN is reliable, accurate and has not tampered with.

For example, a farmer using a WSN to automate the irrigation system should have full confidence that the sensors have not been tempered with, such that they would let the soil dry beyond the desired point or let plants whither in heat before deploying the sprinkler system or alerting the farmer. Likewise, the military should be confident that information received from sensors has not been altered with, by enemies, to give wrong information, as well as dams relying on sensors to warn of a possible flooding before it happens [7].

The type of security threats usually directed at WSNs can be classified by schemes such as Outsider-Insider and passive-active [8]. An example of an Outsider attack could be one coming from the Internet which could be active in harming the nature of the network such as Denial of Service (DOS) attack [9] or passive such as the listening of an authorized user authentication's details. On the other hand, Insider attacks range from situations where someone physically captures a node and reprograms it such that it gives wrong information or for it to harm the integrity of the network from within [10], to an individual, authorized with encryptions keys, using software to inject malicious software into the network [8].

Intrusion Detection Systems (IDS) have long been researched on and employed in networked systems as secondary security measure to encryption mechanisms that usually focus on Outsider attacks [11]. They aim at detecting threats to the network and to report these threats, such that an action is performed to prevent threats before much damage is inflicted to the network [12]. They can also keep a log of all threats to a network, which could be vital for analysis purposes [13].

In this paper however, we explore the possibilities of employing a threat detection algorithm that is light enough to work in WSNs but efficient enough to cater for growth of the network. This algorithm is implemented as a desktop application such that there is no extra load on the actual sensors. The experimentation focuses exclusively on Zigbee wireless sensors, as they are a common choice when implementing a WSN.

Furthermore, fuzzy logic is used in the implementation of the algorithm to limit the margin of error that could be given by the proposed algorithm as well as to lower the number of false positives. A false positive occurs when a thread detection algorithm wrongful classifies a normal operation as a thread.

2. Objectives

The research work reported in this paper aimed at creating an Intrusion Detection System for Zigbee Wireless Sensor Networks that can be run from a desktop environment.

This was achieved through the following sub-objectives;

1. Identification and enhancement (or development of new one if none exists) of an algorithm that can be used in intrusion detection for WSN.
2. Deployment of the algorithm as part of a desktop application.
3. Comparison of the proposed IDS with existing WSN IDS.
4. Testing the IDS in a controlled environment against Zigbee flood and
5. Application of the IDS as part of a real WSN application

3. Literature Review

3.1 The Internet of Things Paradigm

The internet of things (IoT) archetype refers to an evolving discipline of connecting addressable physical objects for communication, using traditional communications structures such as wireless technology and the internet [14].This interconnection also involves living organism such as human beings, animals and plant, where it is used to monitor and improve on the quality of life, of mentioned objects, while lowering the use of resources [15]. WSNs are the heart of the IoT paradigm. They are the senses embedded into this myriad of objects being connected, so as to gather the needed information [16].

3.2 Zigbee Wireless Sensors

Zigbee was developed by IEEE for low power long range devices. It offers very high security mechanisms to protect the integrity of the data through encryption. ZigBee is based on the IEEE 802.15.4, and when broken down, its specification can be identified as 802 which is in place for the networking standard, 14 for wireless network and 4 meaning low data transmission rate as well as low power consumption rate [17].

The table below shows the performance of the Zigbee communication protocol when compared to other communication protocols. This also serves as a justification of its selection in realising the Intrusion Detection System for this project.

Table 1: Comparison of wireless communication protocols

	ZigBee	**Bluetooth**	**Wi-Fi**	**Wireless USB**
Range	10-100 metres	10 metres	50-100 metres	10 metres
Topology	Peer-to-peer, mesh, ad-hoc or star	Ad hoc	Point-to-hub	Point-to-point
Power Consumption	Low	Medium	High	Low
Data Rate	240 Kbits/sec	1 Mbits/sec	54 Mbits/sec	63 Kbits/sec

Although it has been has been said of how secure Zigbee sensors are, it was found that they can be vulnerable to flood attacks in which a node continually associates itself with a network in order to cause a memory overload and eventual crash of the network [18].

3.3 Design and Analysis of Algorithm

Algorithm design can be described as a precise technique for creating a mathematical process for solving problems [19]. The success or the effectiveness of an algorithm is determined by its run time also commonly known as the BIG-OH [20]. Put simply, big-oh is used to check the efficiency of an algorithm in terms of processing time and memory requirements with varying input sizes [21].

Essentially, when dealing with algorithm analysis, the number of steps taken by an algorithm, to execute inputs of varying magnitudes, are counted with the best, worst and average case complexities being determined [22].

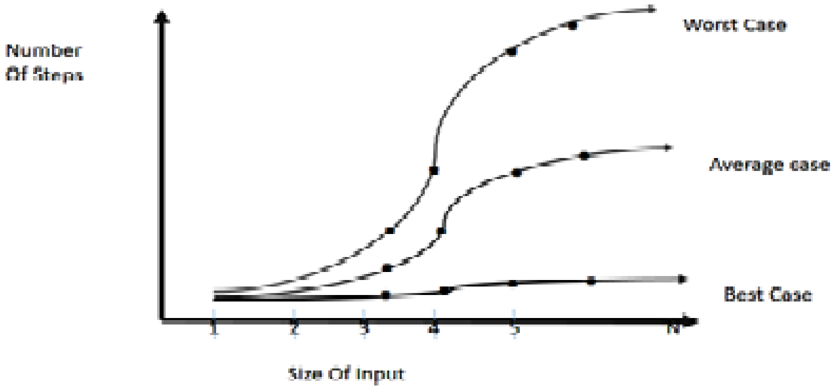


Figure 1: Worst, best and average cases of an algorithm

3.4 Intrusion Detection Systems

Intrusion detection systems (IDS) could be in the form of hardware or software, and were introduced as a secondary security for networks [23]. This is because Intrusion detection systems are more concerned about finding or sniffing out intrusions before or as they happen and alerting system administrators as opposed to just focusing on securing transmission of data, carried out using encryption schemes [24]. Hence these encryption schemes fall under the first level of security for networks. Intrusion detection systems have for a while been used for wired networks but with the introduction of wireless sensor networks, a need has risen to have intrusion detection systems in wireless networks as well [25].

An IDS is basically a system which monitors data traffic coming in and out of a network to determine if there has been an intrusion or attack [26]. In the event of an apparent attack, some IDS have been designed to alert system administrators who will in turn take the appropriate response to protect data, while some IDS have been programmed to act accordingly on their own in case of an apparent attack [27].

With regard to type of reaction to intrusion, the IDS, can take one of two forms; it can have a passive reaction to an intrusion or it can be reactive in nature.

Passive reactive IDS respond to an intrusion detection by keeping a record of the intrusion and then alerting the system administrator for a preventive measure to be taken. Reactive reaction IDS, on the other hand, usually respond by taking off the infected node from the network or re-setting the connection from the infected node [28].

Furthermore, under detection techniques, there are also two types. There is the Anomaly Detection Technique which basically checks the data packets, in a network or from a node, for deviation from a set baseline or threshold value, which is deemed to be normal [29]. If the traffic is not considered normal then an alarm is raised.

There is also the Signature Based technique where the IDS basically checks the data traffic in a network or from a node, for known sets of attack patterns, stored in its database. If a match is found then an alarm is raised [30].

3.5 Algorithms used in Intrusion Detection

There are various algorithms that can be used for the detection of algorithms in intrusion detection systems. Most of these algorithms are borrowed from data mining algorithms as they very much deal with similar issues, of collecting data for analysis or statistical analysis algorithms.

For example, some popular algorithms from data mining like fuzzy logic and genetic algorithms, and have been used for anomaly detection [31].

Fuzzy logic algorithms use rule based logic to find information about the state of a system even when incomplete data is provided because they try to incorporate the dynamics of human reason [32]. This characteristic makes them to not give answers as strict yes or not but also possibilities of the in between.

Genetic algorithms on the other hand are inspired by the natural evolution of living organisms that include natural selection, inheritance and mutation [33]. As such, they have been used to find solutions in search problems in computer sciences. With selection, a solution that is considered best has a higher chance of being selected, in inheritance, old solutions are combined to form new ones and in mutation there is possibility of random changes in the solutions [34].

There are also statistical algorithms which work by finding a deviation from a certain traffic pattern, which will be considered an anomaly [35]. An example of a statistical algorithm that can be used in intrusion detection system is the Cusum [36].

Cusum literally means cumulative sum and it can be defined as follows;

Let n be a sample size greater or equal to 1 (n>=1)

$$S_i = \sum_{j=1}^{i} (\bar{X} - \mu_0)$$

Where $\bar{X}$ is the average of the Jth sample and μ_0 is the in control system mean. If the Cusum value revolves around 0, the system is in control but if mean shifts value. For example if $\mu_1 > \mu_0$ or $\mu_1 < \mu_0$, it means the system is in abnormal state.

4. Methodology

This project took the route of an algorithm development to be followed by an experimental design to benchmark the algorithm against existing algorithms of a similar nature.

The algorithm encompasses a learning stage that is always studying the structure of the network as well as a detection stage. In this stage all nodes in the network are recorded as well as their initial mac address as well as their Id addresses, after they have been granted access to joining the network. A node can join the network at a later stage, after initialisation of network, but the learning phase of the algorithm makes a recording of such an association. This was done such that the algorithm was always aware of all the sensors in the network as well as their expected ID and Mac addresses while they were associated to the network.

For the detection stage, the algorithm checks if all the nodes are still sending data under their initially recorded Mac and ID address, as recorded in the learning phase. In the case where some nodes have ID addresses that are frequently changing in a bit to associate themselves to the network as many times as possible, which is the basis for a flood attack, the detection engine records the frequency of such changes from every node. The recorded frequencies are sent to the fuzzy engine, which decides if the frequency is high enough to raise an alarm. The fuzzy engine also decides if the node should be reset or isolated.

The algorithm works as follows;

```
1.  Loop
    1.1.   Receive_frame ()
        1.1.1. If message_complete ()  then
        1.1.2. Evaluate_address ()
        1.1.3. If adresses_unique () then
        1.1.4. Update_adress_tables ()
        1.1.5. End if
        1.1.6. End if
       Else
        1.1.7. If  Node_Calculate_frequency ()> 1 then
        1.1.8. Fuzzy_reference ()
            1.1.8.1.  If attack mild () {issue warning}
            1.1.8.2.  Else if attack_medium {rese node: drop communication}
            1.1.8.3.  Else if attack_severe {isolate node: drop communication}
            1.1.8.4.  Update_environment ()
    1.2.   else
        1.2.1. Update databases ()
        1.2.2. Update_environment ()
2.  End if
3.  End if
4.  End loop
```

As soon as sensor node has send a full frame of a message (1.1), the address is evaluated to determine if this is a new node or a node that is already associated to the network. Both the network address as well as the unique mac address of the node are evaluated in this stage (1.1.2), and this constitutes as a perpetual learning segment each time a new frame is received. If the address being evaluated is found to not be unique but already associated with the network, the algorithm checks if the address set up of the sending node have remained constant, from a previous transmission or if it has changed in its addressing to a frequency or more than once (1.1.7). If the frequency evaluates to more than one for the number of times the address has changed, the fuzzy logic module is invoked to determine if there is a possible attack based on the set thresholds of mild, medium and severe attacks. Based on the fuzzy evaluation an appropriate response is taken.

The database is updated with regard to any changes that may have occurred. This happens for every time a new message frame is received from a sensor.

This algorithm was then realised in a programming language as part of desktop IDS application.

During initial testing the algorithm has been checked for effectiveness and correctness based on the following presumptions;

- For every instance of the inputs (that is, the specified properties of the inputs are satisfied)
- The algorithm halts, and the outputs produced satisfy the specified input/output relation

The network was put under test using all healthy sensors and during certain time frames, one of sensors was reprogrammed in the middle of execution to misbehave, and every time the application would capture the anomalous sensor.

5. Technology Description

In this section a brief description of the Technology used in the realisation of the project is given.

Three Zigbee sensors and one gateway were used in order to develop and carry out preliminary tests. These sensor are called Waspmode sensors and are manufactured by Libelium [37]. For this to function, the gateway sensor is supposed to be connected to the computer at all times. It also acts as the main coordinator node of the network. All sensors connect to the gateway and after successfully associating with the gateway, with regard to channel and frequency selection, they continuously keep on sending all the sensed data variables to the computer via the gateway.

As the network grows and more sensors are added, and in cases where sensors are too far from the gateway, some sensors that are nearer to the gateway become cluster heads which are responsible for coordinating a certain cluster of nodes and sending their data to the gateway. Conversely, the nodes are also able to work in a system where they relay the data in a series of hops all the way to the gateway.

The coding of the IDS, was done entirely in Java. Java also offers a database that can be integrated as part of an application and this was helpful as it made the application portable and transportable to other computer systems with relative ease, as opposed to using an external database [38]. This database was used to collect all data coming in from sensors and for recording characteristics of the sensor that send the data. The java language also provided an easy way to integrate a fuzzy logic engine for the application through a library called fuzzy-lite [39].

6. Developments

The software application has been fully programmed and it houses an internal database to keep track of the sensors in deployment at any given time. The database works to also keep track of how many times any single node has been associated to the network.

The algorithm together with the fuzzy logic engine were successfully implemented as part of this desktop application. The system is currently being tested to determine how well it interfaces with other applications which need to access data coming in from the sensors.

The reason for this is quite simple; there is one port which receives data coming in from sensor nodes and all this data is first routed to the internal database of the IDS and from there, after presence or lack, of threads has been verified, other external applications can also connect to the database to make use of the data. An architectural framework of the system has also been created and it shows how the different part of the system integrate and communicate with each other.

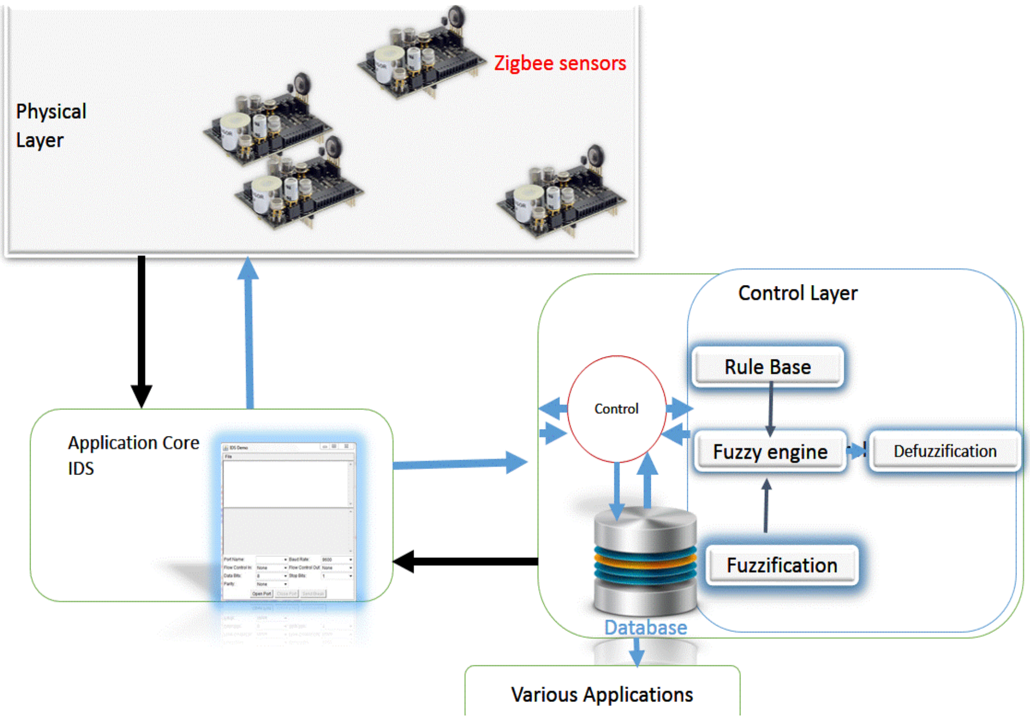
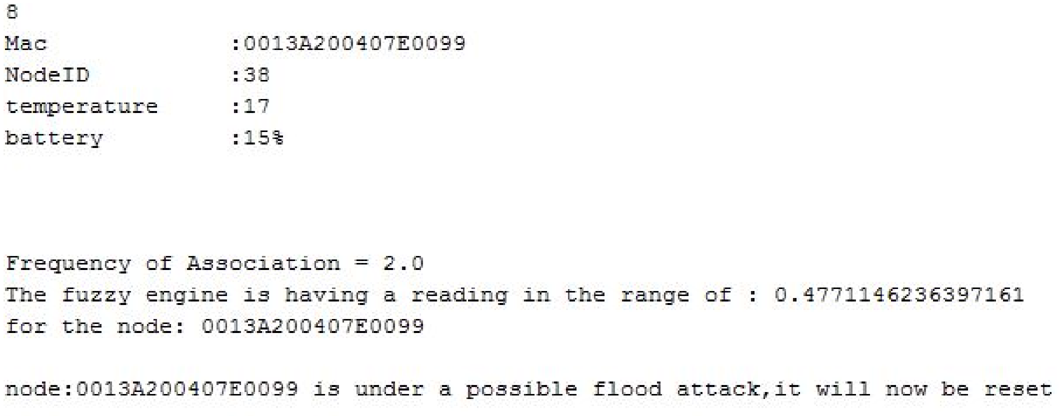


Figure 2: Architectural Framework

7. Results

Preliminary testing shows that the system is able to detect rogue sensor nodes that are trying to cause a possible flood attack on the coordinator node. The figure below shows an actual screen shot of the system as it detects a rogue node in the network. The fuzzy logic engine gives a reading of the intensity of the attack and also issues a plan of action accordingly.



```
8
Mac             :0013A200407E0099
NodeID          :38
temperature     :17
battery         :15%



Frequency of Association = 2.0
The fuzzy engine is having a reading in the range of : 0.4771146236397161
for the node: 0013A200407E0099

node:0013A200407E0099 is under a possible flood attack,it will now be reset
```

Figure 3: Screen shot of the IDS in action

8. Business Benefits

Individuals who wish to be sure that the data received from sensors is accurate and that their network will not lose integrity in the event where a hacker has created rogue nodes as part of their network could use this system.

Ideally, the purpose of a WSN network should not be an issue, as the IDS will work primarily with studying the integrity of the data received before it is presented in a user application.

The tricky part, for end users, lies in making sure the end user's system interfaces with the IDS as this requires some programming background to connect said application with the internal database of the IDS.

However, since concepts of IoT and WSN are only coming to age with the prospect of concepts such as smart cities, smart health care an array of innovative usages to come [author], this system could prove to be timeless or at least help in setting the pace in security development for WSN usage.

9. Conclusions, Discussions and Futherwork

In this paper an algorithm, which can be used in detection of Zigbee flood attacks, has been presented as well as its implementation as part of a desktop application such that the load on actual sensors is lowered. This was done because sensor nodes have very limited memory and processing abilities and need to retain their battery power for prolonged periods of time.

The paper also presented the results thus far of the project. The next step of the project is the benchmark of the developed IDS against other existing IDS when dealing with varying intensities of Zigbee flood attacks as well as to establish the run time efficiency of the algorithm. The benchmark will be bound by the following criteria;

- Relevance of threat
- Adequacy of information about threat provided to administrator
- The frequency of False and positive alarms
- Efficiency of the algorithm when size of input increases

Simultaneous to the above tests, the following tests for efficiency will also be made:

- Use of Node resources.
- Number of steps used by the IDS to identify an intrusion.

Success of the IDS is measured from an overall score in all tested categories and it should at least be in the top three brackets to qualify as competitive IDS that can be used in applications. The value of scores are measured from set criteria, being one (1) for poor performance to three (3) for excellent performance in a category

For future work, the research will look to apply the algorithm technique to a broader spectrum of communication protocols, so as to cater for more IoT WSN sensors, as opposed to just Zigbee specific sensors, in isolation. The study will also look to carry out more tests with an emphasis on enhancement of interoperability with existing applications of WSNs.

References

[1] J. Stankovic and A. Wood, "Realistic application for wireless sensor networks," 2011.

[2] D. Satish, M. Vaidehi and N. Nithya, "Severity Prediction of Drought in a Large Geogrpahical Area Using Distributed Wireless Sensor Networks," Patiala, 2009.

[3] M. Muthoni, "An Innovative Drought Early Warning System for Sub Saharan Africa:Intergrating Modern and Indigenous Approaches," *African Journal of Science, Technology,Innoation and Development,* vol. 7, no. 1, pp. 8-25, 2015.

[4] Dassault Systems, "The Internet of Things:The past,The present,The Future," 2013.

[5] Lopez Research, "Building Smarter Manufacturing With The Internet of Things," cisco, San Fransisco,CA, 2014.

[6] I. Butun and S. Morgera, "A Survey of Intrusion Detection Systems in Wireless Sensor Networks," IEEE, 2014.

[7] V. Seal, A. Raha and S. Maity, "A SIMPLE FLOOD FORECASTING SCHEME USING WIRELESS

SENSOR NETWORKS," *International Journal of Ad hoc, Sensor & Ubiquitous Computing,* vol. 3, no. 1, pp. 1-16, 2012.

[8] A. Lee, J. Loo and A. Lasebae, "6LoWPAN: a study on QoS security threats and countermeasures using intrusion detection system approach," *INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS,* pp. 1-25, 2012.

[9] M. Krishnan, "Intrusion Detection in Wireless Sensor Networks," 2014.

[10] L. Honus, "Design, implementation and simulation of intrusion detection system for wireless sensor networks," 2009.

[11] H. Soliman, "A comparative performance evaluation of intrusion detection techniques for hierarchical wireless sensor networks," *Egyptians Informatics Journal,* vol. 13, pp. 225-237, 2012.

[12] S. Priyanka, "Incremental Intrusion Detection System for Wireless Sensor Networks," *International Journal of Emerging Trends & Technology in Computer Science,* vol. 2, no. 6, pp. 322-326, 2013.

[13] A. Strikos, "A full approach for Intrusion Detection in Wireless Sensor Networks," Stocklhom,Sweden, 2007.

[14] A. M. Mhlaba, "An Integrated Internet of Things Based System for Tracking and Monitoring Assets – the case of the Central University of Technology," in *IST-Africa 2015 Conference Proceedings*, Lilongwe, 2015.

[15] N. David, *How Internet Of Things is Revolutionising Health care,* freescale.com/healthcare, 2013.

[16] M. Muthoni, "IoT Applications that work for the African Continent: Innovation or Adoption?," Bloemfontein.

[17] D. Vishwakarma, "IEEE 802.15.4 and ZigBee: A Conceptual Study," *International Journal of Advanced Research in Computer and Communication Engineering,* vol. 1, no. 7, 2012.

[18] J. Wright, "KillerBee:practical Zigbee Exploitation Framework".

[19] A. Levitin, Introduction to the Design and Analysis of algorithms, 1 ed., Philadelphia: Addison-Wessly, 2003.

[20] D. Mount, "Design and Analysis of Computer Algorithms," University Of Maryland, College Park, 2003.

[21] R. Kalle, G. Gómez-Herrero, K. Egiazarian and S.-L. Eriksson, "A general definition of the big-oh notation for algorithm analysis," Tampere University of Technology, Tampere, 2014.

[22] G. Tovey, "Tutorial On computational Complexity," *Interfaces Informs,* vol. 32, no. 3, pp. 30-61, 2002.

[23] P. Ning and S. Jajodia, "Intrusion Detection Techniques".

[24] SANS Institude, "Understanding Intrusion Detection Systems," SANS Institute InfoSec Reading Room, 2001.

[25] S. Khan, . J. Lloret and . J. Loo, "Intrusion Detection and Security Mechanisms for Wireless Sensor networks," *International Journal of Distributed Sensor Networks,* pp. 1-4, 2014.

[26] H.-J. Liao, C.-H. Lin and Y.-C. Lin, "Intrusion Detection System:A comprehensive view," *Journal of computer and Netwrok applications,* vol. 36, no. 1, pp. 16-24, 2013.

[27] A. Cort, "Algorithm based Approaches to Intrusion Detection and Response," SANS Institude, 2004.

[28] A. Kumar and C. Chhabra, "Intrusion detection system using Expert system (AI) and Pattern recognition (MFCC and improved VQA)," *International Journal of Advance Research in Computer Science and Management Studies,* vol. 2, no. 5, pp. 145-151, 2014.

[29] P. Jyothsna and V. V. Rama, "A Review of Anomaly based IntrusionDetection Systems," *International Journal Of Computing Applications,* vol. 28, no. 7, pp. 26-34, 2011.

[30] D. Gaikwad, P. Pabshettiwar, P. Musale, P. Paranjape and A. S. Pawar, "A Proposal for Implementation of Signature Based Intrusion Detection System Using Multithreading Technique," *International Journal Of Computational Engineering Research ,* vol. 2, no. 7, p. 59, 2012.

[31] M. M. M. Hassan, "Current Studies On Intrusion Detection System,Genetic Algorithm and Fuzzy Logic," *International Journal of Distributed and Parallel Systems (IJDPS),* vol. 4, no. 2, pp. 35-47, 2013.

[32] C. Taylor and D. Meldrum, "ALGORITHM DESIGN, USER INTERFACE, AND OPTIMIZATION PROCEDURE FOR A FUZZY LOGIC RAMP METERING ALGORITHM:A TRAINING MANUAL FOR FREEWAY OPERATIONS ENGINEERS," Washington, 2000.

[33] L. Scrucca, "GA: A Package for Genetic Algorithms in R," *Journal of Statistical Software,* vol. 53, no. 4, pp. 1-34, 2013.

[34] P. Ranjan Srivastava and T.-h. Kim, "Application of Genetic Algorithm in Software Testing," *International Journal of Software Engineering and Its Applications* , vol. 3, no. 4, pp. 87-94, 2009.

[35] S. Obaid Amin, M. Shoaib Siddiqui and C. Seon Hong, "RIDES: Robust Intrusion Detection System for IP-Based Ubiquitous Sensor Networks," *Open Access Sensors Journal,* vol. 9, no. 1, pp. 3447-3468, 2009.

[36] P. Ravi, "An analysis of a widely used version of the CUSUM tracking signal," *JOURNAL OF THE OPERATIONAL RESEARCH SOCIETY,* vol. 65, p. 1189–1192, 2014.

[37] "Libelium," 2015. [Online]. Available: http://www.libelium.com/products/waspmote/. [Accessed 26 December 29015].

[38] D. Debrunner, "Introducing Apache Derby," 2004.

[39] "Fuzzy Lite," 2015. [Online]. Available: http://www.fuzzylite.com/downloads/. [Accessed 26 December 2015].