An efficient intrusion detection framework in cluster-based wireless sensor networks

Hichem Sedjelmaci, Sidi-Mohammed Senouci, Mohammed Feham

An efficient intrusion detection framework in cluster-based wireless sensor networks

Hichem Sedjelmaci[1]*, Sidi Mohammed Senouci[2] and Mohammed Feham[1]

[1] STIC Lab, University of Tlemcen, Tlemcen, Algeria
[2] DRIVE Lab, University of Bourgogne, 49 Rue Mademoiselle Bourgeois, 58000, Nevers, France

ABSTRACT

In the last few years, the technological evolution in the field of wireless sensor networks was impressive, which made them extremely useful in various applications (military, commercial, etc.). In such applications, it is essential to protect the network from malicious attacks. This presents a demand for providing security mechanisms in these vulnerable networks. In this paper, we design a new framework for intrusion detection in cluster-based wireless sensor networks. Our detection framework is composed of different protocols that run at different levels. The first protocol is a specification-based detection protocol that runs at intrusion detection system (IDS) agents (low level). The second one is a binary classification detection protocol that runs at cluster head (CH) node (medium level). In addition, a reputation protocol is used at each CH to evaluate the trustworthiness level of its IDSs agents. Each CH monitors its CH neighbors on the basis of a specification detection protocol with the help of a vote mechanism applied at the base station (high level). We evaluated the performances of our framework in the presence of four well-known attacks: hello flood, selective forwarding, black hole, and wormhole attacks. We evaluated specifically the detection rate, false positive rate, energy consumption, and efficiency. Simulation results show that our detection framework exhibits high detection rate (almost 100%), low number of false positives, less time to detect the attack, and less energy consumption. Our intrusion detection framework outperforms other schemes proposed in the literature in terms of detection, false positive rate, and energy consumption.

KEYWORDS

wireless sensor networks; clustering; intrusion detection system; detection rate; false positive; efficiency

***Correspondence**

Hichem Sedjelmaci, STIC Lab, University of Tlemcen, Algeria.
E-mail: hichem.sedjelmaci@mail.univ-tlemcen.dz

1. INTRODUCTION

Wireless sensor networks (WSNs) are being used in a variety of applications such as military monitoring, detection of forest fires, and human vital functions monitoring. These sensors are autonomous and very small in sizes, and they can be deployed in a random manner in a monitored field. Despite the services they provide and the advantages they bring, WSNs have several constraints related to energy consumption, computational capability, and memory storage. These specific characteristics must be taken into account when we deploy any of these applications into the corresponding devices.

Security is one of the most important issues in WSNs as sensors are often deployed in a hostile and insecure environment such as a battlefield. In addition, the nature of limited resources on sensor nodes restricts the use of conventional security techniques in sensor networks [1].

Cryptographic technique can protect WSNs against external attackers by ensuring data integrity of the ongoing communication and applying packets authentication from the source, which is classified into content-based and stream-based methods [2]. Key management is an important issue in all encryption-based security systems [3], where several researchers work on this field by providing a secure scheme that takes into account the energy consumption of the node [3,4].

However, cryptographic technique cannot detect an internal attacker that is aware of the cryptographic keys. In this context, intrusion detection system (IDS) allows a detection of a suspicious activity within the network by analyzing a target node and triggers an alarm when this node exhibits a malicious behavior. The IDS remains the best mechanism to identify and eject the intruder within the network itself.

In WSNs, IDS topology can be classified as follows [5]: (i) distributed approach and (ii) hierarchical approach. In the distributed approach, intrusion detection load is divided among the sensor nodes, which may collaborate with each other to form a global intrusion detection mechanism.

This architecture is more suitable for flat WSNs. In a flat architecture, the sender relies on multi-hop communication to reach the remote location (base station), leading to a high communication overhead. On the other hand, the hierarchical approach has been proposed for multilayered WSNs named cluster-based wireless sensor network (CWSN). In this approach, a network is divided into clusters where cluster heads (CHs) aggregate data collected from the member nodes. At the same time, all CHs can cooperate with the central base station to form a global IDS. An example of the clustering topology for WSN can be seen in Figure 1. This architecture exhibits a low communication overhead and prolongs the network lifetime. In this paper, we study the problem of intrusion detection on CWSN. We propose an efficient and lightweight intrusion detection framework that suits the application requirements in terms of fast detection time, detection rate, number of false positives, and energy consumption. Our intrusion detection framework uses a set of embedded protocols running at different levels (cluster members, CH, and base station) to detect with high accuracy four well-known attacks: hello flood, selective forwarding, black hole, and wormhole attacks. According to the simulation results, our detection framework outperforms other frameworks proposed in the literature in terms of detection, false positive rate, and energy consumption. The remainder of this paper is organized as follows: In Section 2, we give some background and related work. In Section 3, our intrusion detection framework is proposed. In Section 4, we provide simulation results and performance analysis of our scheme. Finally, we summarize the main results and give some perspectives that we envisage to carry out in Section 5.

2. BACKGROUND AND RELATED WORK

In this section, we highlight some related works and background necessary for understanding our propositions. We organize this section in four subsections. In the first one, we summarize some intrusion detection works that we found in the literature by describing their main shortcomings. In the second subsection, we describe some routing attacks and, especially, selective forwarding, black hole, hello flood and wormhole attacks. In the third subsection, we give some background information on the supervised learning algorithm support vector machines (SVMs). In particular, we describe how SVM can be used in either centralized or distributed fashion. We finally give, in the last subsection, some relevant information about clustering protocols in wireless networks. In particular, we describe the hybrid energy-efficient distributed clustering (HEED) algorithm [6], which was selected as a base of our clustering protocol used in our intrusion detection framework.

2.1. Intrusion detection in WSN

Currently, there are limited researches that use the IDS to identify the malicious behavior within the network. The authors in [7] are among the first who use the mechanism of intrusion detection in WSN. This research work is based on naturally occurring events and the analysis of fluctuations in sensor reading [8]. In [9], the authors propose a model that relies on the number of packets being dropped to detect black hole and selective forwarding attacks. The authors in [10] analyzed the packets by using both detection policies (i.e., anomaly detection based on SVM and a set of attacks signature) to detect the routing attacks with high accuracy. However, the major drawbacks of these schemes [9,10] are related to not taking into account that the IDS node can also be a malicious node and that the CH node is an attractive target of attackers because of their relevant data. In [11], the authors propose an intrusion prevention and detection framework in a one-hop clustering topology for WSNs. In the intrusion prevention phase, the authors propose a cryptographic technique to prevent the external threat to attack the networks. In the intrusion

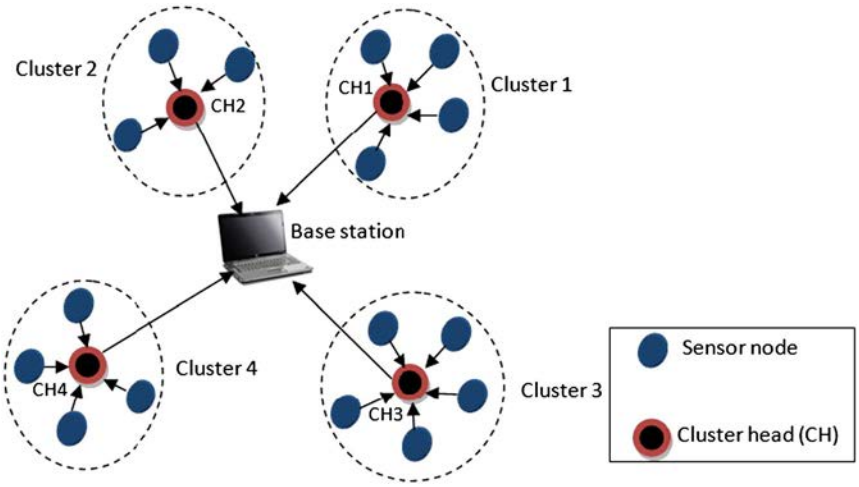

Figure 1. Clustered wireless sensor network topology.

detection phases, each IDS monitors the nodes that are located within its radio range (one hop). The detection framework uses only a rule-based detection to identify the malicious node. In their experiments' results, the authors claim that using one-hop clustering for intrusion detection permits to all the IDS nodes within the same cluster to detect the malicious node when it occurs. In this scheme, the authors do not evaluate the performance of their framework in terms of energy consumption. In [12], the authors propose a mechanism of intrusion detection and isolation of malicious nodes. In their detection mechanism, the CH monitors its cluster member by using a set of rules related to a specific attack behavior. When the intruder occurs, the IDS isolates the attacker nodes and records malicious nodes in its isolation table. In their hierarchical topology, all CHs within the network are managed by a node called a primary CH (PCH). This later is an attractive target of the attackers. In order to avoid this issue, the CH and the cluster members monitor this node. According to the simulation results, their scheme permits to consume less energy compared with the schemes proposed by the authors in [13]. The major drawback of both detection frameworks proposed in [11,12] is the detection policies applied by IDS nodes, which are based only on a rule-based detection. Using only the rule-based approach for the detection process leads to low detection rate when several kinds of attacks occur. In [14], the authors propose a lightweight intrusion detection technique for clustered sensor nodes based on IDS framework developed by the authors in [15]. In this technique, the monitoring node has two detection engines identified as local agent and global agent. The former monitors only their own communication (e.g., sent and received messages and sensed data), and the latter observes the neighbors' communication. The global agent uses a rule-based approach with two-hop neighbor knowledge for the anomaly detection, and it sends alarms to the CH when the intrusion occurs. Both monitoring nodes use signature-based detection, which are computed and generated by the CH. The authors attempt to provide a cooperative mechanism between IDS agents that is based on trust priority in order to reduce the false alerts raised by the intruder. Nevertheless, the drawback of this scheme is the large increase of the size of the signature database, which in turn leads to an overload of the node.

As shown in Figure 2, detection policies for the intrusion in WSN can be classified into two main techniques:

(1) *Signature-based detection or misuse detection*: This approach is based on comparing the observed behavior to a set of attack signatures that are stored in the node's memory. If a match occurs, the analyzed node is defined as an attacker. This technique is quick and reliable to detect known attacks [16]. However, it cannot identify unknown attacks and hence requires constant signature updates to be reliable.

(2) *Anomaly detection*: This approach is based on first modeling the normal node behavior and then identifying anything that deviates from this model as anomalous. This technique is composed of two categories:
   - *Binary classification-based detection*: This category uses a supervised learning algorithm to model the normal behavior. The advantage of this technique is its ability to detect unknown attacks, but its high computational cost leads to a rapid decrease of the node's lifetime. As a consequence and in order to mitigate this cost, the technique must be embedded in a node that has considerable power resources. Among detection techniques proposed in the literature for WSNs, we find neural networks [17], SVM [10], and Markov chain approach [18].
   - *Specification-based detection*: This approach works by simply specifying a normal behavior using a set of rules. The advantage of this technique is the ability to detect unknown malicious behaviors with low computational cost. However, reliability of this detection approach relies on continuous updates of rules over time.

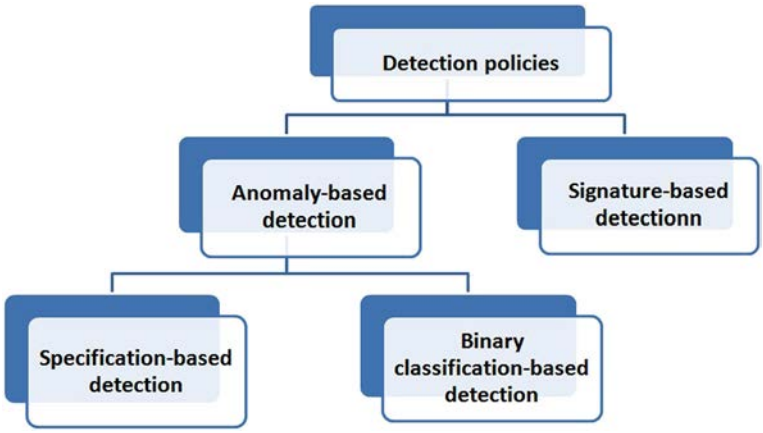

Figure 2. Detection techniques.

2.2. Routing attacks

The intruder could realize one of the four following attacks: selective forwarding, black hole, hello flood, and wormholes. We describe in the following these different attacks.

(1) *Selective forwarding*: In this case, the attacker stops forwarding certain packets and starts dropping them. This attack is therefore detected by calculating the packet-drop rate (PDR).
(2) *Black hole*: In this attack, the intruder pretends to be in the shortest path to the CH by using a high-power transmission. In this case, the intruder will be able to receive the messages and subsequently swallows the corresponding packets (drops all receiving packets). This attack can be detected by computing the PDR and the received signal strength intensity (RSSI).
(3) *Hello flood*: A malicious node broadcasts hello packets by generating a high signal strength compared with other sensor nodes. In this case, other legitimate nodes in the network will send their packets to the broadcasting node. As a result, the packets will then be dropped, spoofed, or altered. This attack can be detected by computing the RSSI.
(4) *Wormholes*: According to the work undertaken by the authors in [19], wormhole attacks are classified into passive or active attacks. In our research, we focus on active wormholes. In particular, the wormhole attacks tend to pretend to be one hop away from the CH by using high signal strength. As a consequence, the attacker forwards the messages received from a legitimate node to another attacker as illustrated in Figure 3. In this case, both malicious nodes take part in the network routing protocol. In Figure 3, note that M1 and M2 are the endpoints of wormhole tunnel and M1 generates a high signal strength in order to convince a node that is close to the CH (one hop away from CH). Node A wants to send its packets to the CH either by following the valid route (nodes B and C) or a malicious one (nodes M1, E, and M2). In both cases, node A chooses the lower-cost route via M1–M2 wormholes (shown in solid arrows) because M1 pretends to be close to the CH. Therefore, all packets received by M1 from A are forwarded directly to M2 and are not sent to E. In this case and in order to detect this attack, we simply monitor the signal strength. In addition, nodes located in the same neighborhood of this attack do not receive the packets from this malicious node; hence, the packet-dropping rate becomes high. As illustrated in Figure 3, the IDS1 agent hears the packets sent by M1 with a high RSSI. In addition, this monitoring node does not hear the message that must be forwarded by E to M2. With the RSSI and PDR, M1 will be detected as a malicious node that is carrying out a wormhole attack.

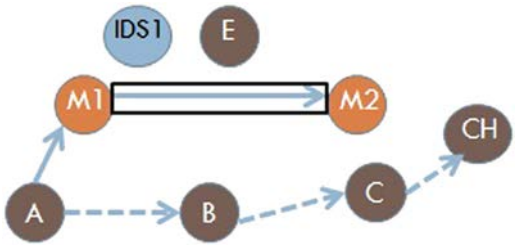

Figure 3. Active wormhole attack.

2.3. Support vector machines

Support vector machine is a supervised learning method developed by Vapnik in 1995, and it is used for classification and regression analysis. The aim of the SVM classifier is to construct a hyperplane that separates data into two classes defined by the number of support vectors. These vectors define the boundary of each class. In situations where SVM cannot separate data into two classes (nonlinear separation), it solves this problem by mapping input data into high-dimensional attributes spaces using a kernel function [20]. As a result, it allows a linear separation. We note that, in our research, we focus only on two-class problems. The binary SVM classification provides a decision function [21]:

$$f(a,x,b) = \operatorname{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i k(x_i,x) + b\right) = \pm\{1\} \quad (1)$$

Here, $k(x_i,x)$ is the kernel function, and α are the Lagrange multipliers, which can be found by solving the following nonlinear optimization equations:

$$\begin{cases} \max \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j k(x_i,,x_j) \\ \text{subject to} \sum_{i=1}^{m} y_i \alpha_i = 0, 0 \le \alpha_i \le C \end{cases} \quad (2)$$

SVM can be used either in a centralized or distributed fashion.

In the first case, the SVM, which is embedded at the base station, collects the packets from all nodes and then trains an SVM classifier. This approach forces a node to send a considerable amount of data to a remote location, which leads to a high communication overhead and subsequently decreases the lifetime of the sensor nodes.

In the distributed approach, the cost of energy is reduced. The support vectors, which are much less than the input data, are computed at each node. These key vectors are then exchanged between nodes with the exception of the centralized approach where packets are sent to remote

nodes. As a consequence, the network lifetime is increased as this approach meets the energy consumption constraint. In our anomaly detection model, a distributed SVM learning between CHs is applied to detect the anomaly behavior.

2.4. Clustering

A hierarchical topology divides the sensor network into clusters, each one having a CH. The objective of this architecture is to help the deployment of protocols (especially routing) and save the energy that enhances the survivability of the network. This is achieved by designating a CH node the responsibility of forwarding a packet (which contains the aggregated data received from cluster members) to the base station rather than having all nodes send their sensed data to a remote location (base station).

Among the large number of cluster-based routing protocols proposed in the literature, we cite the following: LEACH [22], PEGASIS [23], and HEED [6]. The aim of HEED protocol is to use a combination of the residual energy and an intra-cluster communication cost to elect the CH. In our study, a modified version of this routing protocol is selected (by using only the residual energy) to embed our intrusion detection framework.

In HEED, the authors defined two kinds of nodes: "uncovered" and "covered." In this case, the first node announces itself to become a CH by broadcasting an announcement message to other nodes. This process occurred when the execution algorithm is completed without electing a CH. The "covered" node is a cluster member who selects a lower-cost CH according to the overheard message sent by the CH. To this end, a node can be elected to become a CH by using the following probability formula:

$$CH_{\text{prob}} = \frac{E_{\text{residual}}}{E_{\max}} \tag{3}$$

where E_{residual} and $E_{\max}$ are the residual and maximum energy respectively in the node.

3. PROPOSED FRAMEWORK

In our framework, the intrusion detection process is carried out at three levels as detailed in the following. In the low level, a set of nodes called IDS agents monitor the communication of their neighbors and report their feedbacks to their CH for further detections. To identify any suspected behavior, these agents use the specification-based detection technique. This technique relies on a set of rules do detect and prevent the malicious behavior (more details in (2) of Section 3.1). Because of energy constraints and because one bit transmitted in WSNs consumes about as much power as executing 800 to 1000 instructions [24], the agent node has to limit the amount of information that is exchanged between him or her and the CH.

In the medium level, a powerful CH uses SVM training technique to detect any anomaly. This approach allows separating data into two classes (normal and anomalous). It is called a binary classification. Given that no node is assumed to be trustworthy, a reputation mechanism is applied at the CH in order to evaluate the trustworthiness of their IDSs membership. Detection process occurring between level one (IDS agents) and level two (CH) is illustrated in Figure 4.

In the high level, each CH monitors its CH neighbors on the basis of a specification detection technique and sends a ballot form to the base station containing the suspected CH. The base station is used as the counter to collect the

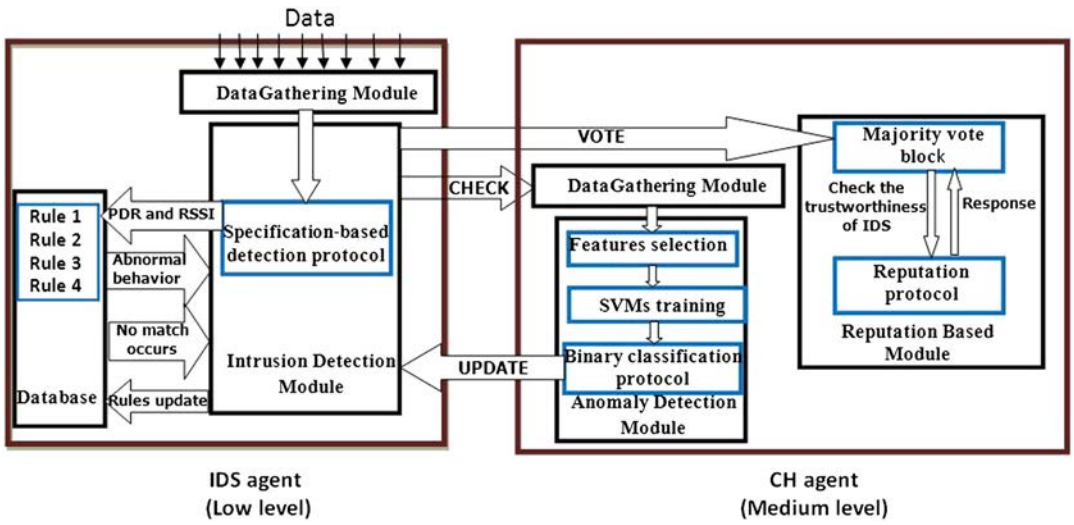

Figure 4. Detection process occurring between intrusion detection system (IDS) and cluster head (CH) agents. SVM, support vector machine.

votes that are generated by CHs in order to take a final decision on any suspected node that may be found. Detection process occurring between levels two (CH) and three (base station) is illustrated in Figure 5.

In the following, we give more details about the different protocols of our framework.

3.1. Sensor (low) level intrusion detection

In each cluster and for each communication link, there must be at least one IDS agent for collecting and analyzing the packets according to the set of rules within the radio range. As shown in Figure 4 (IDS agent), data gathering and intrusion detection modules are the most important components of this agent. These modules are detailed in the following:

(1) *Data gathering module*. Because of the broadcast nature of wireless networks, IDS nodes gather the packets within their radio range [14] and pass it to the intrusion detection module for process analysis as shown in Figure 4.

(2) *Intrusion detection module*. This module follows a specification-based detection protocol to detect and prevent the malicious nodes. The purpose of this protocol is to categorize the target behavior as normal or abnormal according to a set of rules. In our case, there are four rules related to each attack. The rule for detecting "selective forwarding" attack can be defined as the PDR, which is greater than a certain threshold (δsf). The rule for detecting "hello flood" is the value of RSSI that exceeds a certain predefined threshold ($\delta rssi_{\text{h}}$). The rule for detecting a "black hole" is defined as the number of PDR (which is greater than δ_{bh} threshold) and the excess in signal strength (higher than $\delta rssi_{\text{bh}}$ threshold). Finally, the rule for detecting "wormhole" attack is the excess in signal strength (higher than $\delta rssi_{\text{wo}}$ threshold), and none of the nodes, which are located in the same neighborhood of this malicious node, forwards a received packet sent by this adversary (by computing the PDR which excess δwo threshold). We note that, in an area where the links are very unstable or collisions occur, all the nodes located in this area will have an important packet loss. In our detection approach, when the nodes in this area are an attacker, its packets dropping rate will be higher compared with its neighbors (i.e., node located within its radio range).

All these rules used for attacks detection are illustrated in Figure 6.

As illustrated in Figure 4, when abnormal behavior is detected according to the selected rule, a *VOTE* message is submitted to the majority vote block (located at a CH) to make a vote process. This message includes the suspected node and the attack type. When a vote exceeds a certain threshold, the CH will not assign any time slot to this malicious node and will be removed from the cluster. However, when the detection evidence is not very conclusive (no match occurs), a *CHECK* message is forwarded by the IDS agent to the anomaly detection module (located at the CH also) for further detections. This message includes the analyzed node with the PDR and RSSI.

3.2. Cluster (medium) level intrusion detection

Inspired by the work of authors in [6], our clustering algorithm, which was implemented under TOSSIM Simulator [25], elects at each cluster a CH that has more power resources to manage and aggregate data received from the cluster members. As illustrated in Figure 4 (CH agent), this powerful node comprises three modules: data gathering, anomaly detection, and reputation modules. They are detailed in the following:

(1) *Data gathering module*. This module is responsible to collect the *CHECK* messages sent by the IDS agent. This message includes the address of the

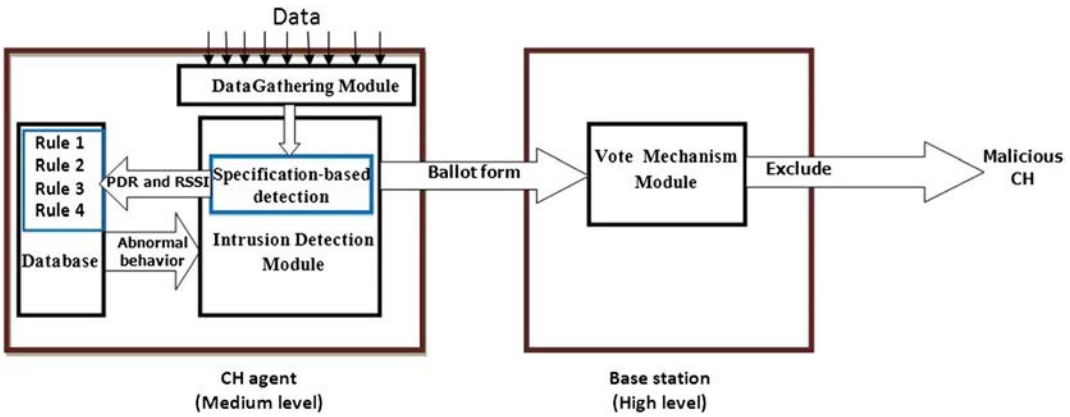

Figure 5. Detection process occurring between cluster head (CH) and base station. PDR, packet-drop rate; RSSI, received signal strength intensity.

```
// Rule for selective forwarding attack
if (PDR>δsf)
 // node_ID is performing a selective forwarding attack
    send_VOTE_message_CH (node_ID);
else
    send_CHECK_message_CH (node_ID, PDR);

// Rule for hello flood attack
if (RSSI>δrssi_h )
 //node_ID is performing a hello flood attack
    send_VOTE_message_CH (node_ID);
else
    send_CHECK_message-CH (node_ID,RSSI);

// Rule for Black Hole attack
if (PDR>δbh  && RSSI>δrssi_bh)
 //node_ID is performing a black Hole attack
    send_VOTE_message_CH(node_ID);
else
    send_CHECK_message_CH (node_ID,PDR,RSSI);

// Rule for Wormholes attack
if (RSSI>δrssi_wo) {
    Monitor(neighbors(node_ID));
    if (PDR>δwo)
     // node_ID is performing a Wormholes attack
        send_VOTE_message_CH (node_ID);
    else
        send_CHECK_message_CH (node_ID, RSSI, PDR);
}
```

Figure 6. The detection rules of the four attacks.

node analyzed by IDS agent and the following features: PDR and RSSI. These features are then forwarded to the anomaly detection module for the training and classification process.

(2) *Anomaly detection module*. Anomaly detection procedure is divided into three steps:

- *Step1: Features selection*. This is an important factor that increases the classification accuracy, reduces the false positive, and speeds up the training time. In this research, the PDR and RSSI are used as input data for the training process.
- *Step2: SVMs training process*. The anomaly detection uses a distributed learning algorithm for the SVM training to classify data as normal or anomalous (a binary classification problem). Each CH trains the SVM locally, then computes a set of data vectors called support vectors that are generally less in number than the input data used during the learning process. These vectors will be sent to an adjacent CH that is located in the same radio range. Each CH that receives the support vectors from their CH neighbors updates its corresponding information by unifying the received data and its own support vectors. They will then retransmit the resulted set of support vectors to the nearby CHs.
- *Step3: Binary classification protocol*. When the training process is completed, each CH classifies new incoming data according to the attacks and the normal pattern. Any deviation from the normal behavior is considered as anomalous. In this case, an *UPDATE* message (including a new sign of attack) is sent back to their IDS members to compute the new rule of this attack as illustrated in Figure 4.

The packet frame of all exchanged messages (*CHECK*, *VOTE*, and *UPDATE*) is illustrated in Figure 7.

(3) *Reputation-based module*. When an IDS agent detects an attack, it sends a *VOTE* message to its CH as illustrated in Figure 4. This message includes the suspected node and the type of the attack. The CH node uses a majority vote block to determine if the suspected node is an intruder or not, while a beta reputation protocol has to evaluate the confidence level of IDS agents [26]. If a vote exceeds the predefined threshold, the suspected node is ejected from the network and the reputation of the IDS nodes that detect the attack will be increased. Otherwise, the reputation of IDSs will be decreased. We note that for each cluster, the threshold is $n/2$,

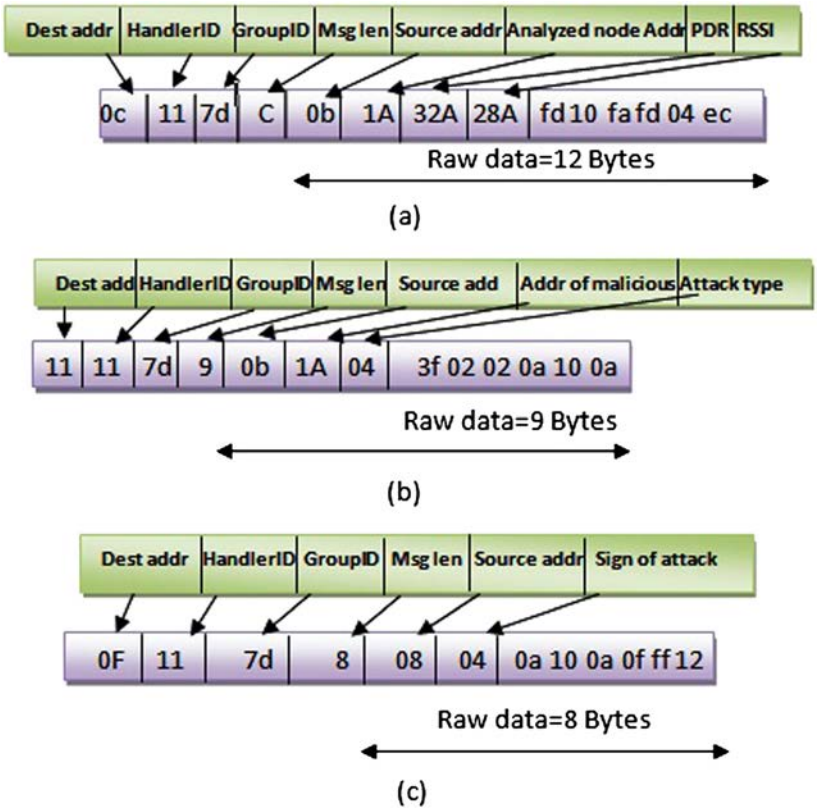

Figure 7. Packet format of (a) CHECK, (b) VOTE, and (c) UPDATE messages.

where n is the number of IDS agents per each cluster. Reputation-based protocol takes a step further in helping to identify compromised nodes as early as possible [27].

The reputation of the IDS_i maintained at its corresponding CH is defined as follows [26]:

$$R_i = \beta(\alpha_i + 1, \beta_i + 1) \tag{4}$$

Here, α_i and β_i represent the normal and suspected behaviors, respectively, of IDS_i claimed by CH. The updating of this two parameters (i.e., α_i and β_i) can be found in [26].

The trust metric is defined as the level of trustworthiness of an IDS node, which can be computed as follows:

$$T_i = E[R_i] \tag{5}$$

where $E[R]$ is the statistical expectation of the reputation function. The Trust value is classified by the following mapping function:

$$M(T_i) = \begin{cases} high & T_i \geq TH \\ low & T_i < TH \end{cases} \tag{6}$$

After computing the trust value, each CH sets this value according to the aforementioned mapping function to indicate the trust level requirement. Only IDSs having a high trust value can trigger the detection process. Otherwise, they will be defined as normal node and not able to play the IDS role. As a result, a community of trustworthy IDS nodes will be generated.

3.3. Intra-cluster-heads (high) level monitoring

The CH is an attractive target of an attacker because it contains relevant data. As a consequence, the intruder uses all its capacity to launch an attack against this hot point. In order to avoid this issue, each CH monitors its CH neighbors. The CH is equipped with data gathering module as in the cluster level intrusion detection and another module, which is intrusion detection. The base station is equipped with a vote mechanism module. These modules are illustrated in Figure 5 and described as follows:

(1) *Data gathering module*. Each CH captures the packets from other CHs that are situated in the same radio range then computes the RSSI and PDR. Subsequently, this information will be forwarded to the intrusion detection module for monitoring purpose as shown in Figure 5.
(2) *Intrusion detection module*. Each CH monitors its nearby CHs by adopting a specification-based detection protocol as used before by IDS agents.

According to the rules related to each attack (refer to (2) of Section 3.1 about these rules), if an abnormal behavior occurs, the monitoring CH sends a ballot form that includes the suspected CH and the attack type to the base station as shown in Figure 5. The base station performs a voting mechanism in order to identify suspect nodes. In particular, if more than half of votes are in favor of attack, the CH will be excluded from the network and a new CH will be elected.

4. PERFORMANCE EVALUATION

In our experiment, we used TOSSIM simulator [25], a simulator for TINYOS application. The main advantage of this simulator, compared with other tools such as NS2 [28], is the fact that we can easily embed the source code written in NESC on real sensor nodes (with TINYOS operating system). However, the TOSSIM simulator does not have the ability to model the energy dissipated during the execution of the application. To this end, an improved version of the tool was proposed by Harvard University called POWERTOSSIM [29] allowing the simulation of nodes' energy consumption and hence the determination of the network's lifetime.

4.1. Simulation assumptions

We used in our simulations 168 nodes deployed randomly in a square area of $88 * 88\,m^2$. We notice that the network was composed of eight clusters with one CH in each. All sensors are static. In order to avoid collisions, a time division multiple access (TDMA) protocol is used. We use the chipcon CC1000 [30] as a transceiver, and each node transmits its packets at a frequency between 433 and 868 MHz. All the key parameters of the simulation are summarized in Table I.

Table I. Simulation parameters.

Simulation time	875 s
Simulation area	$88 * 88\,m^2$
Number of nodes	168
Radio model	Lossy radio model
Number clusters	8
Number of IDSs (i.e., number of IDS agents per cluster)	1–10
Routing	Modified hybrid energy-efficient distributed clustering algorithm
MAC	TDMA
Radio range	15 m
Sensor initial energy	5 J
δsf	64%
$\delta rssi_h$	−41 dBm
δ_{bh}, $\delta rssi_{bh}$	94%, −47 dBm
$\delta rssi_{wo}$, δwo	−44 dBm, 99%

The optimal thresholds for each detection attack were decided by computing the detection and false positives rates, and then, a trade-off between these two metrics that meet our requirements (i.e., high detection and low false positives rates) is determined. We note that these thresholds were determined by carrying out several simulations. The summary of these thresholds are illustrated in Table I.

The purpose of our simulations is to investigate the effect of each attack in the network in isolation and then all together. In addition, we assume that there are no attacks at the beginning of simulation. We have varied the number of IDS nodes per cluster from 1 to 10 in order to assess the performances of our detection framework for different configurations. The binary classification detection protocol used in our simulation is a simple version of SVM learning algorithm that is able to classify only the four routing attacks cited before.

In order to evaluate our framework, we used different metrics:

- *Detection rate (DR)*: defined as the ratio between the number of correct detected intrusions and the total number of intrusions.
- *False positives rate (FPR) or false alarms*: defined as the ratio between the number of normal connections incorrectly classified as intrusions and the total number of normal connections [20].
- *Energy consumption (EC)*: defined as the energy consumed by all sensor nodes and computed as follows:

$$E_t = \frac{\sum_{i=1}^{N} E_{node_i}}{N} \tag{7}$$

where E_t is the energy total of the network and N the number of nodes.

- *Efficiency (E)*: This metric determines the required time for our IDS agents to detect the occurrence of the first adversary node. It is computed as follows:

$$E = \frac{ED - ET}{Sampling\ frequency} \tag{8}$$

where ET is the time of a first malicious behavior starts and ED is the detection time of the first malicious node, respectively.

4.2. Results analysis

4.2.1. Hello flood attack scenario

This attack was implemented as a node that has a high signal strength compared with the other nodes. As shown in Figure 8(a), when the number of IDSs (i.e., the average number of IDS agents per each cluster) increases, the detection rate increases together with the number of false positives. When the average number of IDSs in each cluster is four, the detection rate and false positive rate are close to 98% and 2%, respectively. In addition, as shown in Figure 8(b), our detection framework requires

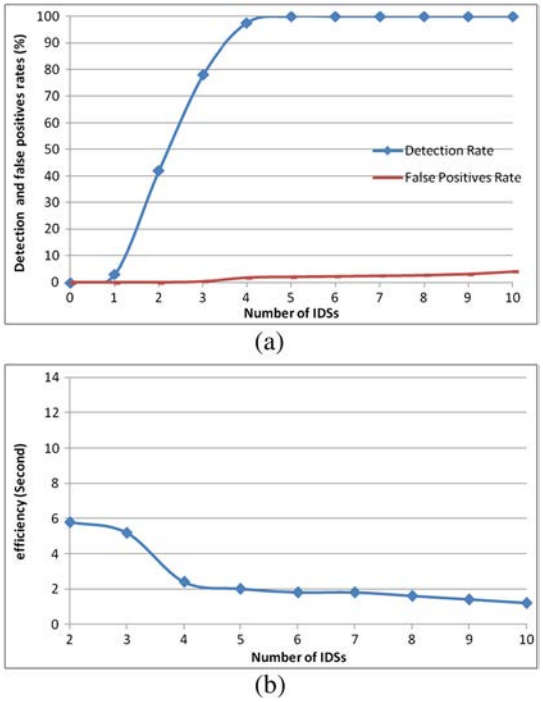

Figure 8. Hello flood attack scenario: (a) detection and false positives rates and (b) efficiency. IDS, intrusion detection system.

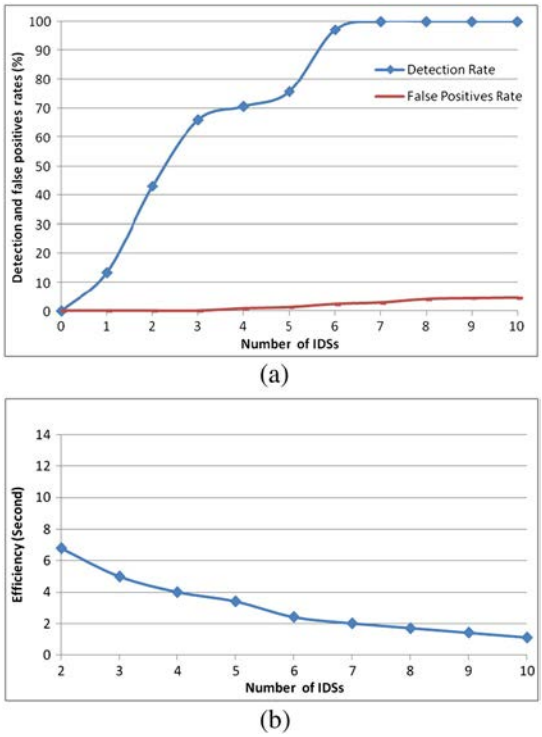

Figure 9. Selective forwarding attack scenario: (a) detection and false positives rates and (b) efficiency. IDS, intrusion detection system.

less time to detect the hello flood attack when the average number of IDSs in each cluster is four (the efficiency is close to 2 s). As a consequence, an optimal number of IDSs is a crucial characteristic that makes our scheme effective. Finally, we conclude that when an optimal number of IDS agents per each cluster is determined (four agents per cluster), our framework exhibits a high detection rate and low number of false alarms, and requires less time to detect this attack.

4.2.2. Selective forwarding attack scenario

The selective forwarding attack is recognized when a node drops a considerable number of packets compared to legitimate node. The detection rate and the number of false alarms are related to the number of IDS agents per each cluster. As shown in Figure 9(a), both metrics increase when the number of agents increases. Therefore, the optimal number of IDSs per each cluster for detection of selective forwarding with less occurrence of false positive is equal to six. In addition, according to this optimal number of agents, our detection framework requires 2 s to detect the selective forwarding attack as shown in Figure 9 (b). Therefore, a trade-off between the number of IDS nodes per each cluster and false positives must be considered in order to suit our application requirements.

4.2.3. Black hole attack scenario

This attack was implemented as a node that has high signal strength and drops all receiving packets. The detection performance of our scheme under black hole attacks is illustrated in Figure 10(a). Our detection framework yields a good detection of black hole attacks, exceeding 96% when the average number of IDSs per each cluster is equal to five. This later is an optimal number of agents, under black hole attacks, that meet our application requirements in terms of detection rate and low number of false positives. The required time of an IDS agent to detect this adversary reaches almost 1.5 s when the number of IDS agents per each cluster is equal to 10, as illustrated in Figure 10(b). However, a high number of false alarms occurred when we select 10 agents per cluster. As a result, the optimal number of IDS nodes per each cluster that meets our application requirements in terms of fast detection time, detection rate, and the number of false alarms is equal to five.

4.2.4. Wormholes attack scenario

This attack was implemented as both the node that generates a high signal strength as well as the nodes located in the same neighborhood of the attack that do not receive the message from this adversary. The detection rate reaches almost 100% when the number of agents increases as shown in Figure 11(a). In this case, the optimal number of IDS agents per cluster that provides a trade-off between the detection rate and the number of false alarms under wormhole attacks is equal to five. The detection of wormhole attack requires a considerable

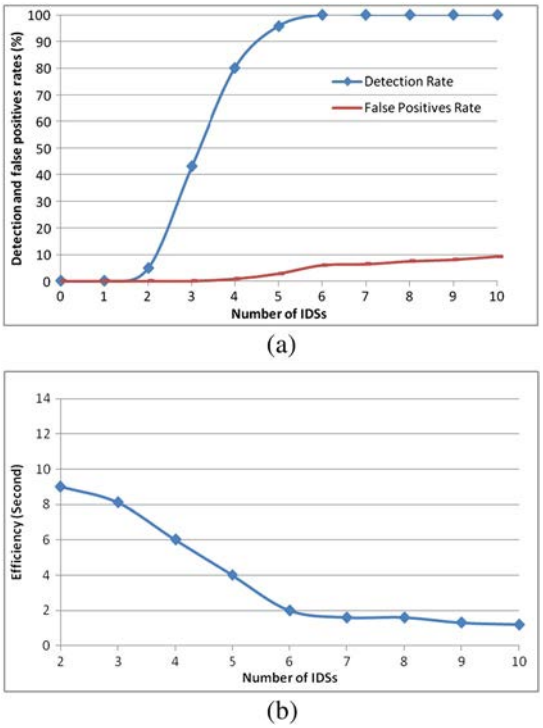

Figure 10. Black hole attack scenario: (a) detection and false positives rates and (b) efficiency. IDS, intrusion detection system.

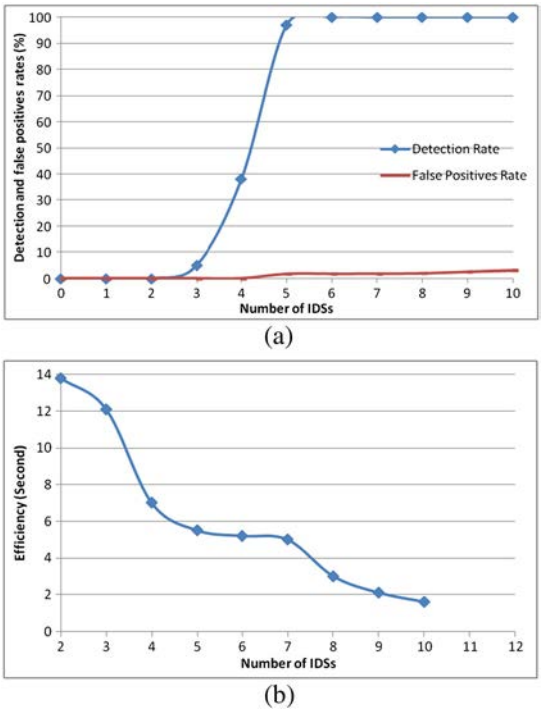

Figure 11. Wormhole attack scenario: (a) detection and false positives rates and (b) efficiency. IDS, intrusion detection system.

amount of time compared with other detection attacks, as illustrated in Figure 11(b). Using six agents per cluster yields to a detection time reaching 4.5 s. As a conclusion, the optimal number of IDS agents per cluster under wormhole attacks for low number of false positives, a high detection rate, and fast detection time is equal to six.

4.2.5. Multiple attacks scenario

In this section, we evaluate the performances of our framework when various kinds of attackers appear within the WSN. First, we evaluate our IDS framework under black hole and selective forwarding attacks with one proposed by the authors in [9] in terms of detection rate. Second, we compare our detection framework when all the attacks cited earlier appear (i.e., hello flood, selective forwarding, black hole, and wormhole attacks). Here, we compare its performances against another scheme proposed in the reference [14] in terms of detection rate, false positives rate, and efficiency. In addition, in order to determine the energy efficiency of our model, we compare the results with the ones obtained in the scheme [31]. As shown in Figure 12, our detection framework performs a better detection against black hole and selective forwarding attacks than the scheme proposed in [9], specifically when the number of IDSs (i.e., the average number of IDS agents per each cluster) is important. In this case, the number of false alarms is related to the number of IDS nodes per each cluster. As a result, increasing the number of IDS agents results in an increase in the rate of false positives. We must

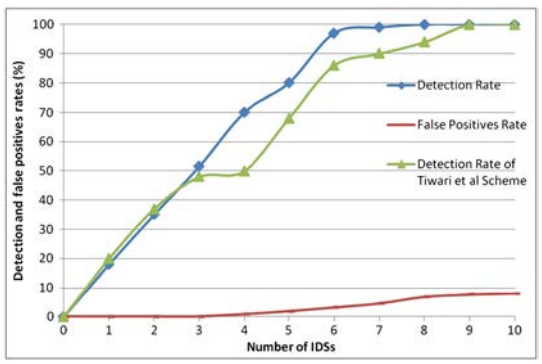

Figure 12. Comparison of our framework under selective forwarding and black hole Attacks. IDS, intrusion detection system.

therefore consider a balance between the low false positives rate and high detection rate metrics. As result, the optimal number of IDSs per cluster meeting our application requirements is equal to six. As shown in Figure 13 (a), our intrusion detection framework is effective against all attacks (cited earlier) when the number of IDS agents per cluster increases. However, the number of false positives will affect the performance of our framework when the number of IDSs per cluster is important (exceed six agents). Therefore, we must consider a balance between the number of IDS agents and the false positive rate. As a result, the optimal number of IDS agents per each cluster that meets our application requirements is

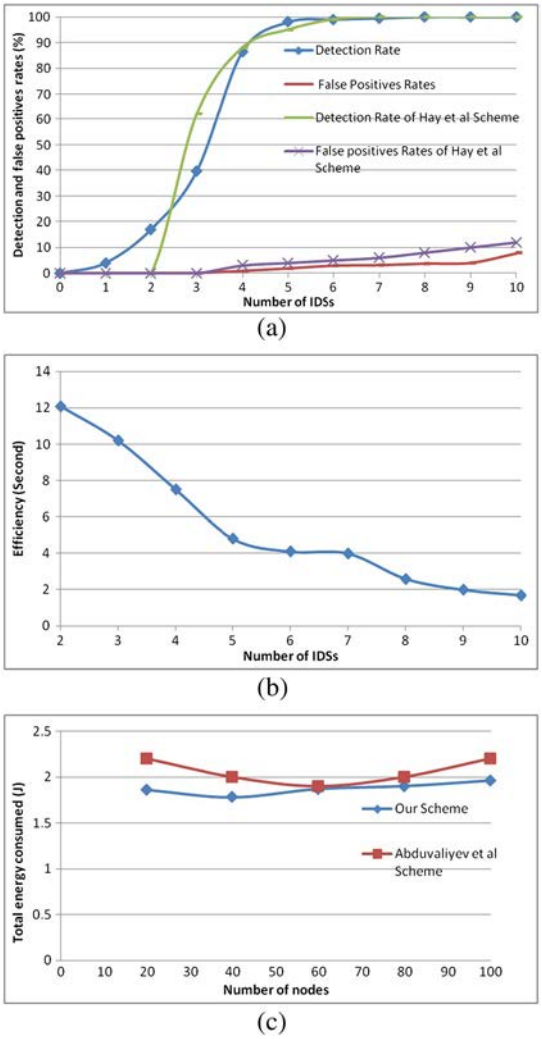

Figure 13. Multiple attacks scenario: (a) detection and false positives rates, (b) efficiency, and (c) energy consumption. IDS, intrusion detection system.

equal to five. The detection and false positives rates are close to 98% and 2%, respectively. As illustrated in Figure 13(a), both schemes exhibit a high detection and low false alarms rates. Otherwise, our scheme performs a better detection and a low number of false alarms compared with the scheme in [14] when an optimal number of IDS agents per each cluster is selected (five agents per each cluster). In other side, according to this optimal number of agents, the required time of IDS node to detect the first malicious node in the network is close to 4 s as illustrated in Figure 13(b), which is suitable for our application requirements. Finally, we conclude that using an optimal number of IDS agents at each cluster, our intrusion detection framework exhibits a low number of false positives, a high detection rate, and fast detection time.

We can observe in Figure 13(c) that our proposed detection framework requires less energy to detect all the attacks that are given earlier in comparison with the approach used by the authors in [31]. This improvement has been achieved because of two main reasons: the first is that we use a clustering topology that aims to select only one node per cluster (CH) that forwards the aggregated data to the base station rather than all nodes sending their sensed data to a remote location (base station). The second reason is the fact that each IDS agent relies on a policy that minimizes the packets transmission that in turn will save energy consumption. As a conclusion, we can state that our scheme improves the network lifetime.

5. CONCLUSION AND FUTURE WORKS

In this paper, we propose an efficient and lightweight intrusion detection framework against common routing attacks that have high severity damage in WSNs. The aim of our framework was to apply a set of intrusion detection protocols on cluster-based WSNs that run at different levels (i.e., at the sensor node level, CH, and base station levels) in order to identify and prevent any adversary node disturbing the network. In particular, at a sensor node level, rule-based detections are implemented at the IDS agents to identify any incoming attack. At the same time, at a CH level, the binary classification detection embedded at each CH aims to update the rules of the IDS agents. In addition, a reputation protocol is used at each CH to evaluate the trustworthiness level of its IDSs member. At a high level, the CH agent sends an intrusion report on the suspected CH to the base station that in turn will perform a voting mechanism about the suspected node. Simulation results show that our scheme presents superior performances for detecting attacks (such as hello flood, selective forwarding, black hole, and wormhole attacks) compared with other schemes. This is mainly specific for networks with an optimal number of IDS agents per cluster. In this case, the IDS agent will generate fast detection time with low number of false alarms. Simulation results confirmed the lightweight of our detection framework in terms of energy used and show that our scheme uses less energy than other model proposed in current literature.

The intrusion detection is the best solution to detect and prevent any malicious node that aims to disturb the network. The proposed detection solution can be very useful in a military application for the protection of the relevant data collected by the sensor against intruder that attempt to alter a data. In addition, the extended version of our intrusion detection approach can be used to monitor human intrusion in a battlefield area and track the positions of moving target. We can also imagine further practical applications, such as applying our detection framework in a forensic environment. In this case, the IDS collects the forensic evidence and monitor any anomaly occurrence or abuse attempt.

In the near future, we will expand the detection range of our framework by adding a sophisticated distributed SVM training model that has the capability to detect any

attack. Few detection approaches are implemented on a large scale in computer networks because they are impractical and would increase the delay within the networks. In addition, in this study, we do not take into account the context of mobile WSNs. Therefore, in our future works, these two limitations will be handled by carry out new simulations.

REFERENCES

1. Chen CY, Chao HC. A survey of key distribution in wireless sensor networks. *Security and Communication Networks* 2011. doi:10.1002/sec.354
2. Zhou L, Chao HC, Vasilakos AV. Joint forensics-scheduling strategy for delay-sensitive multimedia applications over heterogeneous networks. *IEEE Journal on Selected Areas in Communications* 2011; **29**(7):1358–1367.
3. Zhou L, Chao HC. Multimedia traffic security architecture for the Internet of things. *IEEE Network* 2011; **25**(3):29–34.
4. Kim HY, Lee C. A key management scheme for security and energy efficiency in sensor networks. *Journal of Internet Technology* 2012; **13**(2):223–232.
5. Bhattasali T, Chak R. Lightweight hierarchical model for HWSNET. *International Journal of Advanced Smart Sensor Network Systems (IJASSN)* 2011; **1**(2):17–32. doi:10.5121/ijassn.2011.1202
6. Younis O, Fahmy S. HEED: a hybrid energy efficient distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing* 2004; **3**(4):366–379.
7. Doumit SS, Agrawal DP. Self-organized criticality and stochastic learning based intrusion detection system for wireless sensor networks. *Proceedings of the IEEE Military Communications Conference (MILCOM)*, 2003; 609–614.
8. Mitrokotsa A, Karygiannis A. Intrusion detection techniques in sensor networks. In *Book: Wireless Sensor Network Security, Cryptology and Information Security Series*. IOS Press: Amsterdam, Netherlands, 2008; 251–272.
9. Tiwari M, Arya KV, Choudhari R, Choudhary KS. Designing intrusion detection to detect black hole and selective forwarding attack in WSN based on local information. *IEEE Fourth International Conference on Computer Sciences and Convergence Information Technology*, Seoul, Korea, 2009; 824–828.
10. Sedjelmaci H, Feham M. Novel hybrid intrusion detection system for clustered wireless sensor network. *International Journal of Network Security & Its Applications (IJNSA)* 2011; **3**(4):1–14. doi:10.5121/ijnsa.2011.3401
11. Shin S, Kwon T, Jo GY, Park Y, Rhy H. An experimental study of hierarchical intrusion detection for wireless industrial sensor networks. *IEEE Transactions on Industrial Informatics* 2010; **6**(4):744–757.
12. Chen RC, Hsieh CF, Huang YF. A new method for intrusion detection on hierarchical wireless sensor networks. In *Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication*. ACM, SKKU: Suwon, Korea, 2009; 238–245. doi:10.1145/1516241.1516282
13. Su WT, Chang KM, Kuo YH. eHIP: an energy-efficient hybrid intrusion prohibition system for cluster-based wireless sensor networks. *Computer Networks* 2007; **51**(4):1151–1168.
14. Hai TH, Huh EN, Jo M. A lightweight intrusion detection framework for wireless sensor networks. *Wireless Communications and Mobile Computing* 2010; **10**(4):559–572. doi:10.1002/wcm.785
15. Roman R, Zhou J, Lopez J. Applying intrusion detection systems to wireless sensor networks. *The 3rd IEEE Consumer Communications and Networking Conference*, Las Vegas, USA, 2006; 640–644.
16. Jalali H, Baraani A. Process aware host-based intrusion detection model. *International Journal of Communication Networks and Information Security (IJCNIS)* 2012; **4**(2):117–124.
17. Yan KQ, Wang SC, Wang SS, Liu CW. Hybrid intrusion detection system for enhancing the security of a cluster-based wireless sensor network. *The 3rd IEEE International Conference on Computer Science and Information Technology*, Chengdu, China, 2010; 114–118.
18. Uemura T, Dohi T, Kaio N. Dependability analysis of a scalable intrusion tolerant architecture with two detection modes. *Journal of Internet Technology* 2010; **11**(2):289–298.
19. DeGraaf R, Hegazy I, Horton J, Safavi-Naini R. Distributed detection of wormhole attacks in wireless sensor networks. In *Proceedings of 1st International Conference on Ad Hoc Networks*. Springer: Niagara Falls, Canada, 2009; 208–223.
20. Haijun X, Fang P, Ling W, Hongwei L. Ad hoc-based feature selection and support vector machine classifier for intrusion detection. *Proceedings of IEEE International Conference on Grey Systems and Intelligent Services*, Nanjing, China, 2007; 1117–1121.
21. Gama J, Pedersen R. Predictive learning in sensor networks. In *Learning from Data Streams*, Gama J, Gaber M (eds). Springer: Heidelberg, Germany, 2007; 143–164.
22. Heinzelman WR, Chandrakasan A, Balakrishnan H. Energy efficient communication protocol for wireless microsensor networks. *The 33rd IEEE International*

Conference on System Sciences, Vol. **2**, Hawaii, USA, 2000; 1–10.
23. Lindsey S, Raghavendra C. PEGASIS: power efficient gathering in sensor information system. *Proceedings of the IEEE International Conference on Aerospace* 2002; **3**:1125–1130.
24. Stetsko A, Folkman L, Matay V. Neighbor-based intrusion detection for wireless sensor network. *IEEE 6th International Conference on Wireless and Mobile Communications*, Valencia, Spain, 2010; 420–425.
25. Simulating TinyOS networks, 2003. Available at: http://www.cs.berkeley.edu/pal/research/tossim.html
26. Ganeriwal S, Srivastava MB. Reputation based framework for high integrity sensor networks. *Proceeding of the Second ACM Workshop on Security of* Ad Hoc *and Sensor Networks*, New York, USA, 2004; 66–77. DOI: 10.1145/1029102.1029115
27. Alzaid H, Foo E, Nieto JG, Ahmed E. Mitigating on–off attacks in reputation-based secure data aggregation for wireless sensor networks. *Security and Communication Networks* 2012; **5**(2):125–144. doi:10.1002/sec.286
28. The network simulator ns-2, 2000. Available at: http://www.isi.edu/nsnam/ns/
29. Efficient power simulation for TinyOS applications, 2004. Available at: http://www.eecs.harvard.edu/shnayder/ptossim/
30. CC1000 chip, very low power RF transceiver, 2007. Available at http://www.ti.com/lit/ds/symlink/cc1000.pdf.
31. Abduvaliyev A, Lee S, Lee YK. Energy efficient hybrid intrusion detection system for wireless sensor networks. *IEEE International Conference on Electronics and Information Engineering*, Kyoto, Japan, 2010; 25–29.