Hindawi Publishing Corporation
Mobile Information Systems
Volume 2015, Article ID 475030, 18 pages
http://dx.doi.org/10.1155/2015/475030

Research Article

Energy Efficient and Safe Weighted Clustering Algorithm for Mobile Wireless Sensor Networks

Amine Dahane,[1] Abdelhamid Loukil,[1] Bouabdellah Kechar,[2] and Nasr-Eddine Berrached[1]

[1] *Intelligent Systems Research Laboratory, University of Sciences and Technology of Oran, Algeria*
[2] *Laboratory of Industrial Computing and Networking, Ahmed Ben Bella Oran University, Algeria*

Correspondence should be addressed to Abdelhamid Loukil; abdelhamid.loukil@univ-usto.dz

Received 10 November 2014; Accepted 15 January 2015

Academic Editor: Zahoor Khan

The main concern of clustering approaches for mobile wireless sensor networks (WSNs) is to prolong the battery life of the individual sensors and the network lifetime. For a successful clustering approach the need of a powerful mechanism to safely elect a cluster head remains a challenging task in many research works that take into account the mobility of the network. The approach based on the computing of the weight of each node in the network is one of the proposed techniques to deal with this problem. In this paper, we propose an energy efficient and safe weighted clustering algorithm (ES-WCA) for mobile WSNs using a combination of five metrics. Among these metrics lies the behavioral level metric which promotes a safe choice of a cluster head in the sense where this last one will never be a malicious node. Moreover, the highlight of our work is summarized in a comprehensive strategy for monitoring the network, in order to detect and remove the malicious nodes. We use simulation study to demonstrate the performance of the proposed algorithm.

1. Introduction

After the success of theoretical research contributions in previous decade, wireless sensor networks (WSNs) have become now a reality [1–3]. Their deployment in many societal, environmental, and industrial applications makes them very useful in practice. These networks consisted of large number of small size nodes which sense ubiquitously some physical phenomenon (temperature, humidity, acceleration, noise, light intensity, wind speed, etc.) and report the collected data to the sink station by using multihop wireless communications. Although the nodes are able to self-organize and collaborate together in order to establish and maintain the network, they are battery powered, limited in terms of processing, storage, and communication capabilities [4]. WSNs are considered in many cases as stationary, but topology changes can happen due to a weak mobility (new nodes join the network and existing nodes experience hardware failure or exhaust their batteries) [5]. In other scenarios, the mobility can occur when nodes are carried by external forces such as wind, water, or air [6] so that the network topology can be affected accordingly and can be changed slowly. This second kind of mobility, known as one form of strong mobility in the literature in the sense where nodes are forced to move physically in the deployment area, has been considered in this paper. Clustering means grouping nodes which are closed to each other and it has been widely studied in ad hoc networks [3, 7–14]. More recently, it has been used in WSNs [14–21] where the purpose in general is to reduce useful energy consumption and routing overhead. Figure 1 illustrates how inside the cluster two kinds of nodes can be found: one node called cluster head (CH) or coordinator (in Figure 1: CH1, CH2, and CH3) which is responsible for coordinating the cluster activities and several ordinary nodes called cluster members (CMs) (in Figure 1: CM1 and CM2) that have direct access only to one CH. An ordinary node which is able to hear two or more CHs is called a gateway (G) (in Figure 1: the gateway G2 can hear CH1, CH2, and CH3, while the gateway G1 can hear CH1 and CH2) instead. So, each communication initiated by a cluster member to a destination inside the cluster must pass by CH. If the destination is outside the cluster, the communication must be forwarded by a gateway. Recent research studies recognize that organizing mobile WSNs, in the sense defined above, into clusters by using a clustering mechanism is a challenging task [9, 19]. This is due to the fact that CHs carry out extra

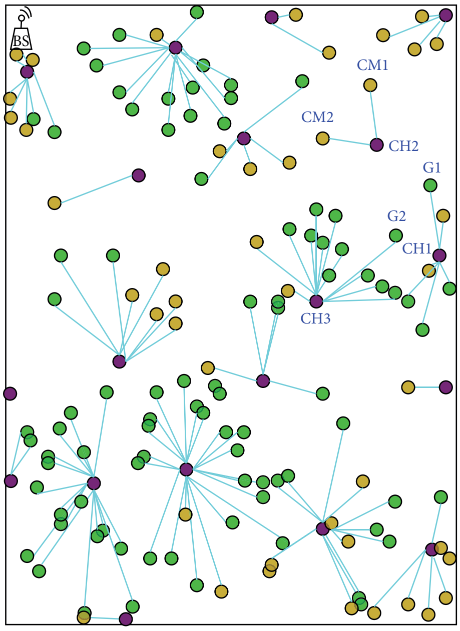


Figure 1: Clustering formation of WSNs composed of 150 sensor nodes deployed in a 570 m × 555 m space area with a radio range = 100 m.

work and consequently consume more energy compared to CMs during the network operations and this will lead to untimely death causing network partition and therefore failure in communication link. For this reason, one of the most frequently encountered problems in this mechanism is to search for the best way to elect CH for each cluster. Indeed, a CH can be selected by computing the quality of nodes. This may depend on several metrics: connectivity degree, mobility, residual energy, and the distance of a node from its neighbors. Significant improvement in performance of this quality can be achieved by combining these metrics [3, 9, 10, 12, 19, 21].

In this paper, we propose an energy efficient and safe weighted clustering algorithm for mobile WSNs using a combination of the above metrics to which we added a behavioral level metric. The latter metric is decisive and allows the proposed clustering algorithm to avoid any malicious node in the neighborhood to become a CH, even if the remaining metrics are in its favor. The election of CHs is carried out using weights of neighboring nodes which are computed based on selected metrics. So this strategy ensures the election of legitimate CHs with high weights. The preliminary results obtained through simulation study demonstrate the effectiveness of our algorithm in terms of the number of equilibrate clusters and the number of reaffiliations, compared to WCA (Weighted Clustering Algorithm) [3], DWCA (Distributed Weighted Clustering Algorithm) [9], and SDCA (Secure Distributed Clustering Algorithm) [21]. These results also reveal that our approach is suitable if we plan to use it in network layer reactive routing protocols instead of proactive ones once the clustering mechanism is launched.

We can enumerate the contributions of our paper as follows:

(i) maintaining stable clustering structure and offering a better performance in terms of the number of reaffiliations using the proposed algorithm ES-WCA (Energy Efficient and Safe Weighted Clustering Algorithm);

(ii) detecting common routing problems and attacks in clustered WSNs, based on behavior level;

(iii) showing clearly the interest of the routing protocols in energy saving and therefore maximizing the lifetime of the global network.

The remaining part of this paper is organized as follows. Section 2 briefly surveys the related works on clustering algorithms proposed for ad hoc networks and in particular those developed for WSNs. In Section 3, we emphasize on the security problems in WSNs. Section 4 introduces and explains the selected metrics for the proposed approach of clustering. More details on the proposed algorithm are given in Section 5. Section 6 presents the simulation tool developed for evaluation. Simulation results are provided to show the effectiveness of the proposed algorithm. Section 7 concludes the paper and outline directions of future works.

2. Related Works

In this section, we outline some approaches of clustering used in ad hoc networks and WSNs. Research studies on clustering in ad hoc networks involve surveyed works on clustering algorithms [11, 22] and cluster head election algorithms [10, 16]. Abbasi and Younis [17] presented taxonomy and classification of typical clustering schemes, then summarized different clustering algorithms for WSNs based on classification of variable convergence time protocols and constant convergence time algorithms, and highlighted their objectives, features, complexity, and so forth. A single metric based on clustering as in paper [23] shows that the node with the least stability value is elected as CH among its neighbors. However, the choice of CH which has a lower energy level could quickly become a bottleneck of its cluster. Er and Seah [8] designed and implemented a dynamic energy efficient clustering algorithm (DEECA) for mobile ad hoc networks (MANETs) that increases the network lifetime. The proposed model elects first the nodes that have a higher energy and less mobility as cluster heads, then periodically monitors the cluster head's energy, and locally alters the clusters to reduce the energy consumption of the suffering cluster heads. The algorithm defines a yellow threshold to achieve some sort of local load balancing and a red threshold to trigger local reclustering in the network. However, the cluster formation in this scheme is not based on connectivity so the formed clusters are not well connected; consequently, this increases the reaffiliation rate and maximizes reclustering situations. Jain and Reddy [24] have proposed a novel method of modeling wireless sensor network using fuzzy graph and energy efficient fuzzy based k-Hop clustering algorithm which takes into account the dynamic nature of network, volatile aspects of radio links, and physical layer uncertainty. They have defined a new centrality metric, namely, fuzzy k-hop centrality. The proposed centrality metric considers residual energy of individual nodes, link quality, hop distance between the prospective cluster head, and respective member nodes to ensure better cluster head selection and cluster quality, which results in better scalability, balancing of energy consumption of nodes, and longer network lifetime. Other proposals use a strategy based on computed weight in order to elect CHs [3, 9, 10, 12]. The main strategy of these algorithms is based mainly on adding more metrics such as the connectivity degree, mobility, residual energy, and the distance of a node from its neighbors, corresponding to some performance in the process of electing CHs. Although the algorithms which use this strategy allow us to ensure the election of better CHs based only on their high computed weight from the considered metrics, they unfortunately do not ensure that the elected CHs are legitimated nodes, that is, whether the election process of CHs is safe or not. Safa et al. [13] propose a novel cluster based trust-aware routing protocol (CBTRP) for MANETs to protect forwarded packets from intermediary malicious nodes. The proposed protocol ensures the passage of packets through trusted routes only by making nodes monitor the behavior of each other and update their trust tables accordingly. However, in CBTRP all nodes monitor the network which lead to rapid drainage of node energy and therefore minimize the lifetime of the network. In Section 3, we show that WSNs are vulnerable to various types of attacks [24, 25]. In the last decade, several studies proposed solutions to solve attacks in WSNs by using cryptography, such as SPINS [26]. However, cryptography alone is not enough to prevent node compromise attacks and novel misbehavior in WSNs [27]. Little effort has been made to include the security aspect in the clustering mechanism. Yu et al. [4, 28] try to secure the clustering mechanism against wormhole attack in ad hoc networks (communication between CHs). However, this is done after forming clusters, not during the election procedure of CHs. Liu [4, 29] surveyed the clustering algorithms available for WSNs but that was done from the perspective of data routing. Hai et al. [30] propose a lightweight intrusion detection framework integrated for clustered sensor networks by using an overhearing mechanism to reduce the sending alert packets. Elhdhili et al. [31] propose a reputation based clustering algorithm (RECA) that aims to elect trustworthy, stable, and high energy cluster heads but during the election procedure, not after forming clusters. Benahmed et al. [21] used clustering mechanism based on weighted computing as an efficient solution to detect misbehavior nodes during distributed monitoring process in WSNs. However, they focused only on the misbehavior of malicious nodes and not on the nature of attacks, the formed clusters are not homogeneous, the proposed algorithm SDCA is not coupled with a routing protocols, and it does not give much importance to energy consumption.

In this paper, the proposed approach focuses around strategy of distributed resolution which enables us to generate a reduced number of balanced and homogeneous clusters in order to minimize the energy consumption of the entire network and prolong sensors lifetime. The introduction of a new metric (the behavioral level metric) promotes a safe choice of a cluster head in the sense where this last one will never be a malicious node. Thus, the highlight of our work is summarized in a comprehensive strategy for monitoring the network, in order to detect and remove the malicious nodes.

The fact that WSNs include limited energy resources (batteries) due mainly to their small size, our algorithm shows clearly the interest of the routing protocols in energy saving which therefore maximize the lifetime of the network by coupling it with AODV and then DSDV protocols [5, 32, 33].

3. Security in WSNs

The typical attacks in WSNs include Sinkhole attack, Black Hole attack, Hello Flood attack, and Node Outage which are the most common network layer attacks on WSNs [30, 34–38]. These selected attacks have been summarized in the following sections.

3.1. Sinkhole. Sinkhole attack is one of the most devastating ones: it is very hard to protect against [36, 39]. In a Sinkhole attack, the adversary's goal is to redirect nearly all the traffic from a particular area through a compromised node, creating a metaphorical sinkhole with the adversary at the center so that all traffic in the surrounding will be absorbed by the malicious node. Because nodes, on or near the path followed by transmitted packets, have many opportunities to tamper with application data. Sinkhole attacks can enable many other attacks such as selective forwarding, for example [40].

3.2. Black Hole. In this attack, malicious nodes advertise very short paths (sometimes zero-cost paths) to every other node, forming routing black holes within the network [41]. As their advertisement propagates, the network routes more traffic in their direction. In addition to disrupting traffic delivery, this causes intense resource contention around the malicious node as neighbors compete for limited bandwidth. These neighbors may themselves be exhausted prematurely, causing a hole or partition in the network.

3.3. Hello Flood Attack. Many routing protocols use "Hello" broadcast messages to announce themselves to their neighbor nodes. The nodes that receive this message assume that source nodes are within range and add source nodes to their neighbor list. The Hello Flood attacks can be caused by a node which broadcasts a Hello packet with very high power, so that a large number of nodes even far away in the network choose it as the parent node [14]. These nodes are then convinced that the attacker node is their neighbor, so that all the nodes will respond to the Hello message and waste their energy.

3.4. Node Outage. If a node acts as an intermediary, an aggregation point, or a cluster head, what happens if the node stops working? Protocols used by the WSNs must be robust enough to mitigate the effects of failures by providing alternate routes [34].

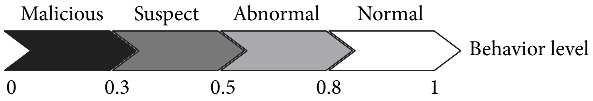


Figure 2: Behavior level $BL_i \in [0, 1]$.

4. Metrics for CHs Election

This section introduces the different metrics used for cluster head election by focusing on behavior level metric.

4.1. The Behavior Level of Node n_i (BL_i). The behavioral level of a node n_i is a key metric in our contribution. Initially, each node is assigned an equal static behavior level "$BL_i = 1$." However, this level can be decreased by the anomaly detection algorithm if a node misbehaves. For computing the behavior level of each node, nodes with a behavior level less than threshold behavior will not be accepted as CH candidates even if they have the other interesting characteristics such as high energy, high degree of connectivity, or low mobility. Nevertheless, abnormal nodes and suspect nodes may belong to a cluster as CM but never as CH. So, we define the behavior level of each sensor node n_i, noted BL_i, in any neighborhood of the network as illustrated in Figure 2.

BL_i is classified by the following mapping function:

$$\mathrm{Mp}(BL_i) = \left\{ \begin{array}{ll} \text{Normal node:} & 0.8 \le BL_i \le 1 \\ \text{Abnormal node:} & 0.5 \le BL_i < 0.8 \\ \text{Suspect node:} & 0.3 \le BL_i < 0.5 \\ \text{Malicious node:} & 0 \le BL_i < 0.3 \end{array} \right\}. \quad (1)$$

The values in formula (1) are chosen on the basis of several reputed models of WSNs adopted by numerous researchers like Shaikh et al. [42] and Lehsaini et al. [43]. The monitor node watches its neighbors to know what each one of them does with the messages it receives from another neighbor. If the neighbor of the monitor changes, delays, replicates, or simply keeps a message that should be retransmitted, the monitor counts a failure. Number of failures have influence on the behavior of neighbors; for instance, if the monitor counts one failure from a neighbor, its behavior will decrease by 0.1 units. This allows the monitor (cluster head) to differentiate malicious nodes (that make much failure) of a legitimate node (that make fewer failure) in case there are collisions.

4.2. The Mobility of Node n_i (M_i). Our objective is to have stable clusters. So, we have to elect nodes with low relative mobility as CHs. To characterize the instantaneous nodal mobility, we use a simple heuristic mechanism as presented in the formula below (2) [4, 44]:

$$M_i = \frac{1}{T}\sum_{t=1}^{T}\sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2}, \quad (2)$$

where (x_t, y_t) and (x_{t-1}, y_{t-1}) are the coordinates of node n_i at time t and $t-1$, respectively. T is the period for which this parameter is estimated.

In our previous paper [4], the considered mobility has a particular sense by the fact that a mobile node does not move from one location to another in the space area of its own will, but in our case, it moves through the forces acting from the outside. These external forces can act from time to time sporadically. In contrary, the malicious node can use its own ability to move freely in the space area. The behavior of the malicious node by moving frequently inside the same cluster (case illustrated by Figure 3) or from a cluster to another is a normal behavior to not attract attention of the neighborhood and therefore be detected. The idea of our algorithm to ensure the choice of a legitimate CH is to never elect a node that moves frequently and even it has the best performance metrics, but this malicious node does nothing just mobility, so in this paper our algorithm (ES-WCA) detects the internal misbehavior of nodes during distributed monitoring process in WSNs by the follow-up of the messages exchanged between the nodes. ES-WCA is based on the ideas proposed by da Silva et al. [45] used in his efficient and accurate IDS in detecting different kinds of simulated attacks.

4.3. The Distance between Node n_i and Its Neighbors (D_i). This is likely to reduce node detachments and enhance cluster stability. For each node i, we compute the sum of the distance D_i with all its neighbors j. This distance is given, as in [3, 4, 9], by

$$D_i = \sum_{j \in N(i)} \{\mathrm{dist}(i, j)\}. \quad (3)$$

4.4. The Residual Energy of Node n_i (Er_i). The residual energy of a node n_i, after transmitting a message of k bits at distance d from the receiver, is calculated according to [4, 16]

$$Er_i = E - (E_{Tx}(k, d) + E_{Rx\,\mathrm{elec}}(k)), \quad (4)$$

where

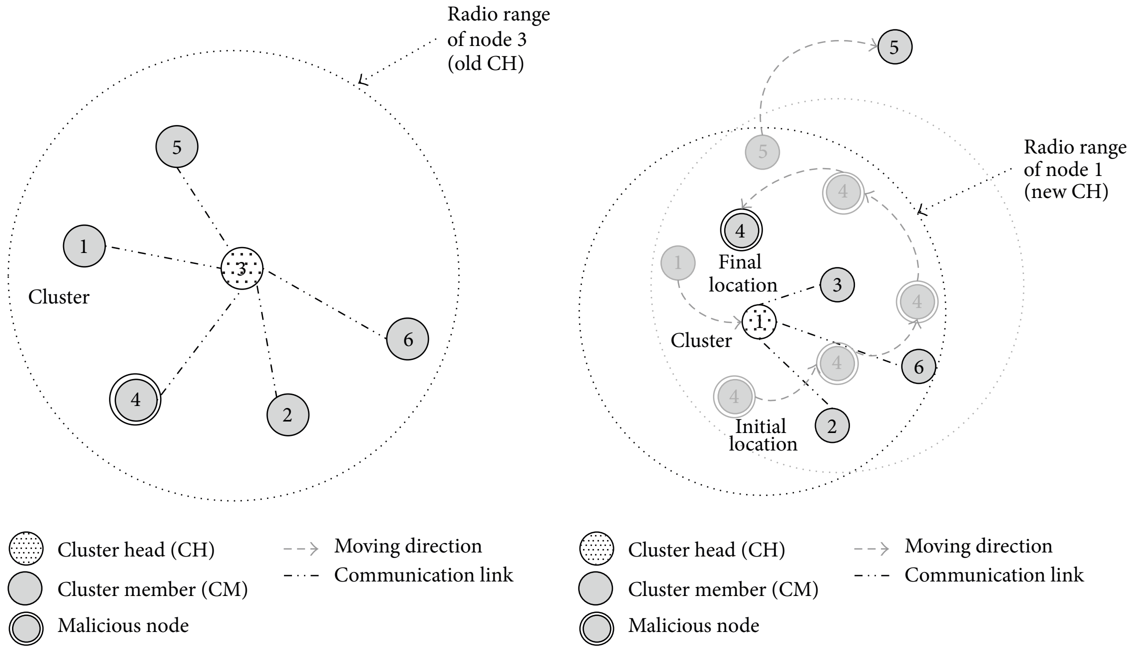
(i) E: the node's current energy;

(ii) $E_{Tx}(k, d) = k \cdot E_{\mathrm{elec}} + k \cdot E_{\mathrm{amp}} \cdot d^2$: it refers to the required energy to send a message, where E_{amp} is the required amplifier energy;

(iii) $E_{Rx\,\mathrm{elec}}(k) = kE_{\mathrm{elec}}$: it refers to the energy consumed while receiving a message.

4.5. The Degree of Connectivity of Node n_i at Time t (C_i). It represents the number of n_i's neighbors given by (5) according to [4]

$$C_i = |N(i)|, \quad (5)$$



(a)

(b)

Figure 3: (a) Clustering mechanism in mobile WSNs before moving nodes and (b) after moving nodes 1, 5, and 4.

where

(i) $N(i) = \{n_i/\text{dist}(i, j) < tx_{\text{range}} \text{ with } i \neq j\}$,

(ii) $\text{dist}(i, j)$: outdistance separating two nodes n_i and n_j,

(iii) tx_{range}: the transmission radius.

For each node, we must calculate its weight P_i, according to the equation:

$$P_i = w_1 * \text{BL}_i + w_2 * \text{Er}_i + w_3 * M_i + w_4 * C_i + w_5 * D_i, \tag{6}$$

where w_1, w_2, w_3, w_4, and w_5 are the coefficients corresponding to the system criteria, so that

$$w_1 + w_2 + w_3 + w_4 + w_5 = 1. \tag{7}$$

We propose to generate homogeneous clusters whose size lies between two thresholds: $Thresh_{Upper}$ and $Thresh_{Lower}$.

These thresholds are arbitrarily selected or they depend on the topology of the network. Thus, if their values depend on the topology of the network, they are calculated as follows according to [43]:

(i) u: the node that has the maximum number of neighbors with one jump:

$$\delta_{12}(u) = \min(\delta_{12}(u_i) : u_i \in U), \tag{8}$$

(ii) v: the node that has the minimal number of neighbors with one jump:

$$\delta_{12}(v) = \min(\delta_{12}(v_i) : v_i \in U). \tag{9}$$

We denote AVG by the average cardinal of the groups with one jump of all the nodes of the network:

$$\text{AVG} = \frac{\sum_{i=1}^{n} \delta_{12}(u_i)}{N}, \tag{10}$$

where N represents the number of nodes in the network. Thus, the two thresholds are calculated as follows:

$$\begin{aligned} Thresh_{Upper} &= \frac{1}{2}(\delta_{12}(u) + \text{AVG}), \\ Thresh_{Lower} &= \frac{1}{2}(\delta_{12}(v) + \text{AVG}). \end{aligned} \tag{11}$$

The calculated weight for each sensor is based on the above parameters ($\text{BL}_i, M_i, D_i, \text{Er}_i$, and C_i). The values of coefficients w_i should be chosen depending on the basis of the importance of each metric in considered WSNs applications. For instance, it is possible to assign a greater value to the metric BL_i compared to other metrics if we promote the safety aspect in the clustering mechanism. It is also possible to assign the same value for each coefficient w_i in the case where all metrics are considered as having the same importance. An approach based on these weight types will enable us to build a self-organizing algorithm which forms a small number of homogenous clusters in size and radius by geographically grouping close nodes. The resulting weighted clustering algorithm reduces energy consumption and guaranties the choice of legitimate CHs.

5. Weighted Clustering Algorithm (ES-WCA)

In this section, we first present some assumptions of the proposed algorithm: Energy Efficient and Safe Weighted

Clustering algorithm (ES-WCA). Then we present in detail an extended version of ES-WCA [4] followed by an illustrative example.

5.1. Assumptions. This paper is based on the following assumptions.

(i) The network formed by the nodes and the links can be represented by an undirected graph $G = (U, E)$, where U represents the set of nodes ni and E represents the set of links ei [3, 4].

(ii) All sensor nodes are deployed randomly in a 2-dimension (2D) plane.

(iii) A node interacts with its one-hop neighbors directly and with other nodes via intermediate nodes using multihop packet forwarding based on a routing protocol such as ad hoc on demand distance vector [5, 32] or DSDV [33].

(iv) The radio coverage of sensor nodes is a circular region centered on this node with radius R.

(v) Two sensor nodes cannot be deployed in exactly the same position x, y in a 2D space.

(vi) All sensor nodes are identical or homogeneous. For example, they have the same radio coverage radius R.

(vii) Each node can determine its position at any moment in a 2D space.

(viii) Each cluster is monitored by only one CH.

(ix) Each CM communicates directly with its CH for the transmission of security metrics.

(x) A CH communicates directly with the base station for the transmission of security information and possible alerts.

5.2. Proposed Algorithm. The ES-WCA algorithm that we present below is based on the ideas proposed by Chatterjee et al. [3], Lehsaini et al. [43], and Zabian et al. [10], with modifications made for our application. This algorithm runs in three phases: the setup phase, the reaffiliation phase, and the monitoring phase. ES-WCA combines each of the above system parameters with certain weighting factors chosen according to the system needs.

5.2.1. The Setup Phase. ES-WCA uses three types of messages in the setup phase (Algorithm 1). The message CHmsg is sent in the network by the sensor node which has the greatest weigh. The second one is the JOINmsg message which is sent by the neighbor of CH if it wants to join this cluster. Finally, a CH must send a response ACCEPTmsg message as shown in Figure 4.

The node which has the greatest weight begins the procedure by broadcasting CH message to their 1-hop neighbors to confirm its role as a leader of the cluster. The neighbors confirm their role as being member nodes by broadcasting a JOINmsg message. In the case when nodes have the same maximum weight, the CH is chosen by using the best parameters ordered by their importance. If all parameters of nodes are equal, the choice is random.

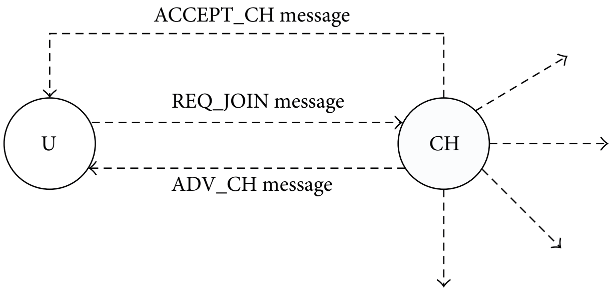


Figure 4: Procedure of affiliation of node “U” to a cluster.

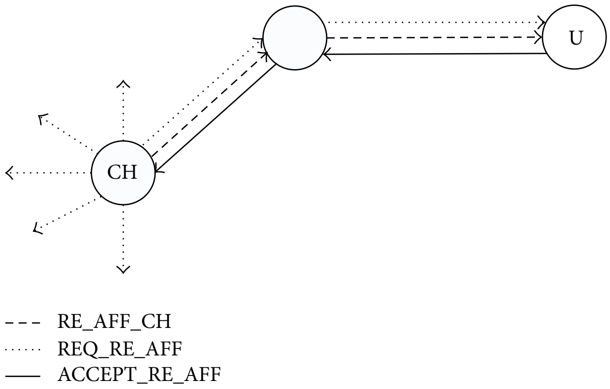


Figure 5: Procedure of reaffiliation of node “U” to a cluster.

Table 1: Values of the various criteria of normal nodes.

Ids	BL_i	Er_i	C_i	D_i	M_i	P_i
1	0.86	3842.12	3	1.15	1.20	769.632
4	0.81	4832.54	5	2.30	0.30	968.133
5	0.88	4053.25	3	1.30	0.55	811.829
6	0.85	4620.43	0	0.00	0.20	924.361
8	0.81	4816.80	4	1.05	1.40	964.753
10	0.95	3650.25	2	0.55	0.10	730.805
11	0.91	4819.60	1	0.70	2.20	964.753

5.2.2. The Reaffiliation Phase. ES-WCA uses four types of messages in the reaffiliation phase (Algorithm 2). The message RE_AFF_CH is sent in the network by the CH whose cluster size is less than $Thresh_{Upper}$. The second one is the REQ_RE_AFF message which is sent by the neighbors of CH if it wants to join this cluster. Finally a CH must send a response ACCEPT_RE_AFF message or DROP_AFF message as illustrated by Figure 5. Accordingly, in this phase we propose to reaffiliate the sensor nodes belonging to clusters that have not attained the cluster size $Thresh_{Lower}$ to those that did not achieve $Thresh_{Upper}$ in order to reduce the number of clusters formed and organize them so as to obtain homogeneous and balanced clusters.

With the help of 3 figures (Figures 6, 7, and 8), our algorithm setup phase is demonstrated. Table 1 shows the quantitative results of the different criteria applied on the normal nodes ($BL_i \geq 0.8$). Table 2 shows the weights P_i of neighbors for each node which has behavior BL_i higher

```
Begin
(1)  Assign values to the coefficients $w_1, w_2, w_3, w_4, w_5$;
(2)  **For** any node $n_i \in G$ **make**:
(3)       $n_i$ forms a list of its neighbors $N(i)$ through the Message who_are_neighbors;
(4)       $N(i) = \emptyset$;
(5)       Calculate its weight $P_i$:
(6)           $P_i = w_1 * \mathrm{BL}_i + w_2 * \mathrm{Er}_i + w_3 * M_i + w_4 * C_i + w_5 * D_i$;
(7)       Initialize Time Cluster and the state vector of all
          nodes $n_i \in G$ Vector_State (Id, CH, Weight, List_Neighbors, Size, Nature)
(8)       CH = 0, Size = 0;
(9)       Nature = "None";
(10)      Repeat
(11)          Any node $n_i \in G$ Broadcasts a message "Hello";
(12)          If $N(i)$ <> $\emptyset$ Then
(13)              Choose $v \in N(i)$;
(14)              $Weight(v) = \max\{weight(w)/\ w \in N(i)\}$;
(15) the node that have the same maximum weight, the CH is
the node that has the best criteria ordered by their
importance ($\mathrm{BL}_i$, $\mathrm{Er}_i$, $C_i$, $D_i$ and $M_i$) if all criteria of
nodes are equal, the choice is random.
(15)          Else $n_i$ is a CH of itself.
              EndIf
(16)          Update the state vector of the elected CH;
(17)          CH = ID;
(18)          Size = 1;
(19)          Nature = CH;
(20)          Send the message "CHmsg" by CH to its neighbors $N$(CH);
(21)          $J$ = Count ($N$(CH));
(22)          For $I = 1$ to $J$ Do
(23)              If ($n_i \in N$(CH) receives the message &&$n_i$ → CH = 0)
(24)                Then $n_i$ sends a message "JOINmsg" to CH
(25)              If (CH → Size < $Thresh_{Upper}$)
(26)                Then CH sends a message "ACCEPTmsg" to Node $n_i$;
(27)                      CH executes the accession process;
(28)                      CH → Size = CH → Size + 1;
(29)                      $n_i$ executes the accession process;
(30)                      $n_i$ → CH = CH → Id;
(31)                Else go to (10);
                  EndIf
              EndIf
          End For
(32) Until expired (TimeCluster);
End.
```

Algorithm 1: Algorithm setup phase.

Table 2: Weights of neighbors.

Ids	1	4	5	6	8	10	11
1	769.632				**964.753**		**964.753**
4		**968.133**	811.829		964.753		
5		**968.133**	811.829			730.805	
6				**924.361**			
8	769.632				**964.753**		
10		**968.133**	811.829			730.805	
11	769.632						**964.753**

```
Inputs: $Thresh_{Upper}$, $Thresh_{Lower}$;
Outputs: set of clusters
Begin
(1)  For num_cl = 1 to Count (Cluster) Do
(2)      If (Size (Cluster [num_cl]) < $Thresh_{Upper}$)
           Then
(3)        CH sends a message "RE_AFF_CH" to its neighbors
           ($N$(CH));
(4)        $J$ = Count ($N$(CH));
         EndIf
(5)  For $I$ = 1 to $J$ Do
(6)      If ($n_i \in N$(CH) receives the message)
           && ($n_i$ ∈ (Size (Cluster [num_cl]) < $Thresh_{Lower}$)
           Then
(7)        $n_i$ sends a Select message "REQ_RE_AFF" to the CH;
(8)      If (Size (Cluster [num_cl]) < $Thresh_{Upper}$)
           Then
(9)        CH sends a message "ACCEPT_RE_AFF" to $n_i$;
(10)       CH updates its state vector;
(11)       CH → CH → Size = Size + 1;
(12)       $n_i$ updates its state vector;
(13)       $n_i$ → CH → ID = ID;
(14)       Else CH sends a "FIN_AFF" message to $n_i$;
(15)       Go to (2);
         EndIF
(16)   Else $n_i$ sends a "DROP_AFF" message to CH;
       EndIf
     End For
   End For
End.
```

Algorithm 2: Algorithm reaffiliation phase.

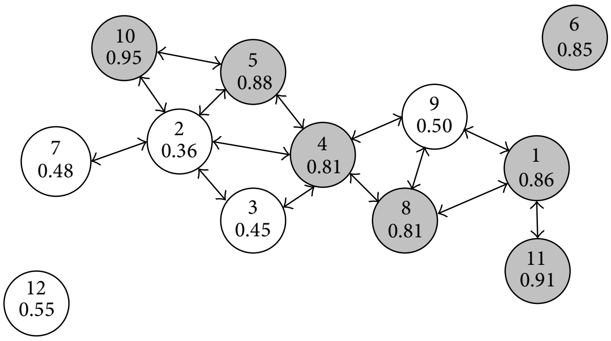


Figure 6: Topology of the network.

than 0.8. The circles in Figure 6 represent the nodes, their identity Ids are at the top, and their levels of behavior are at the bottom. According to Table 2, node 1 could be attached to either CH11 or CH8 (since they have the same weight). However, the behavior level of node 11 is greater than that of node 8 ($BL_{11} > BL_8$). So, node 1 will be attached to CH11. For the other nodes, we have various conditions. Node 4 declares itself as a CH. Node 5 will be attached to CH4. Node 6 declares itself as a CH, because it is an isolated node. Node 8 will be attached to CH4. Node 10 is connected to CH5, but node 5 is attached to CH4. Thus, node 10 declares itself as a CH. Node 11 declares itself as a CH. These results give us the representation shown in Figure 7. Node 2 is connected to CH4 and CH10. Node 2 will be attached to CH4, because CH4 has the maximum weight (968.133). Node 3 is connected to CH4, which implies that node 3 will be attached to CH4. Node 7 is not connected to any CH, so node 7 declares itself as CH. Node 9 is connected to CH4, and then node 9 will be attached to CH4. Node 12 is not connected to any CH, which implies that node 12 declares itself as a CH. These results give us the representation shown in Figure 8. We propose to generate homogeneous clusters whose size lies between two thresholds: $Thresh_{Upper} = 9$ and $Thresh_{Lower} = 6$. For that, we suggest to reaffiliate the sensor nodes belonging to the clusters that have not attained the cluster size $Thresh_{Lower}$ to those that did not reach $Thresh_{Upper}$. Node 4 has the highest weight and his size is less than $Thresh_{Upper}$. Nodes 1, 7, and 10 are neighbors of node 4 with 2 hops and belong to the clusters that have not attained the cluster size $Thresh_{Lower}$, so these nodes get merged to cluster 2. Clusters 1, 3, and 4 will be homogeneous with cluster 1 when the network becomes densely.

At the end of this example, we obtain a network of four clusters (as shown in Figure 9).

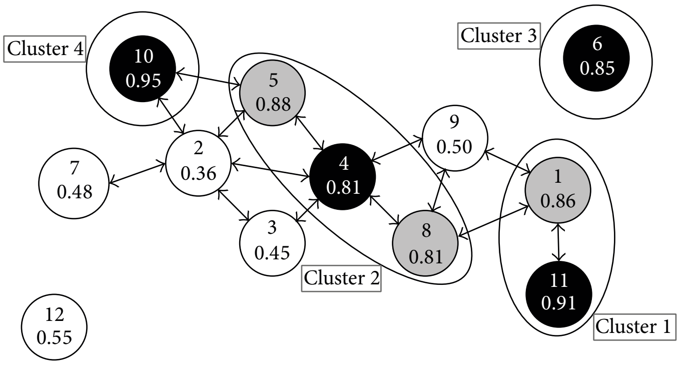


Figure 7: Identification of clusters node.

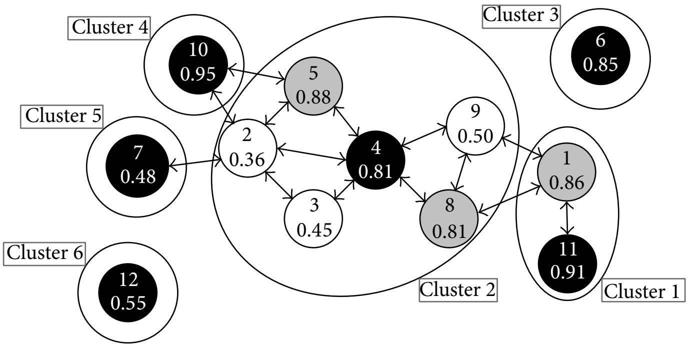


Figure 8: The final identification of clusters.

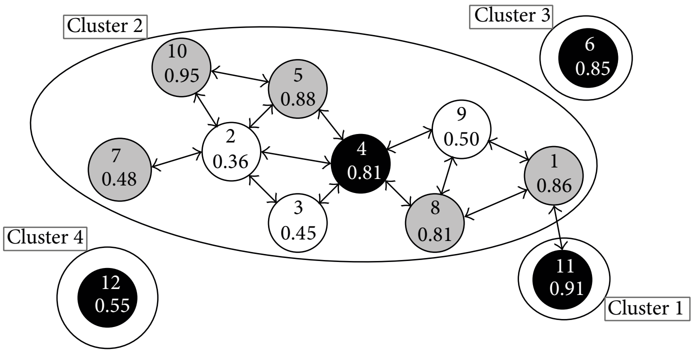


Figure 9: Final cluster structure (reaffiliation phase).

There are five situations that require the maintenance of clusters:

(i) battery depletion of a node,

(ii) behavior level of a node less than or equal 0.3,

(iii) adding, moving, or deleting a node.

In all of these cases, if a node n_i is CH then the setup phase will be repeated.

5.2.3. The Monitoring Phase. Monitoring in WSNs can be both local and global. The local monitoring can be with respect to a node and the global monitoring can be with respect to the network, but in sensor networks, for detecting some types of errors and security anomalies, the local monitoring would be insufficient [46]. For this reason, we adopt in this paper a hybrid approach that is global monitoring based on distributed local monitoring. The general architecture of our approach is illustrated in Figure 10. Our simulator, baptized "Mercury," detects the internal misbehavior nodes during distributed monitoring process in WSNs by the follow-up of the messages exchanged between the nodes. We assume that the network has already a mechanism of prevention to avoid the external attacks. By using a set of rules, all the received messages are analyzed. A similar approach is used by da Silva et al. [45] and Benahmed et al. [21].

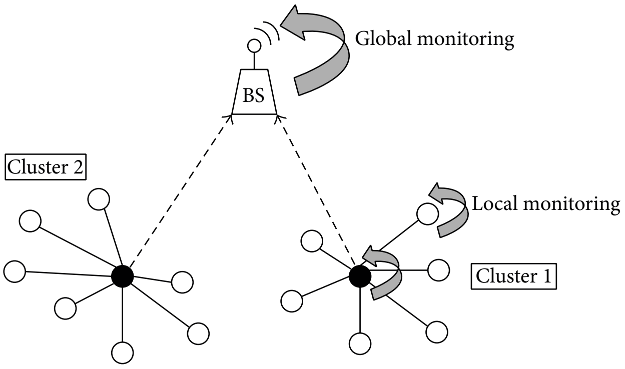


Figure 10: Monitoring phase architecture.

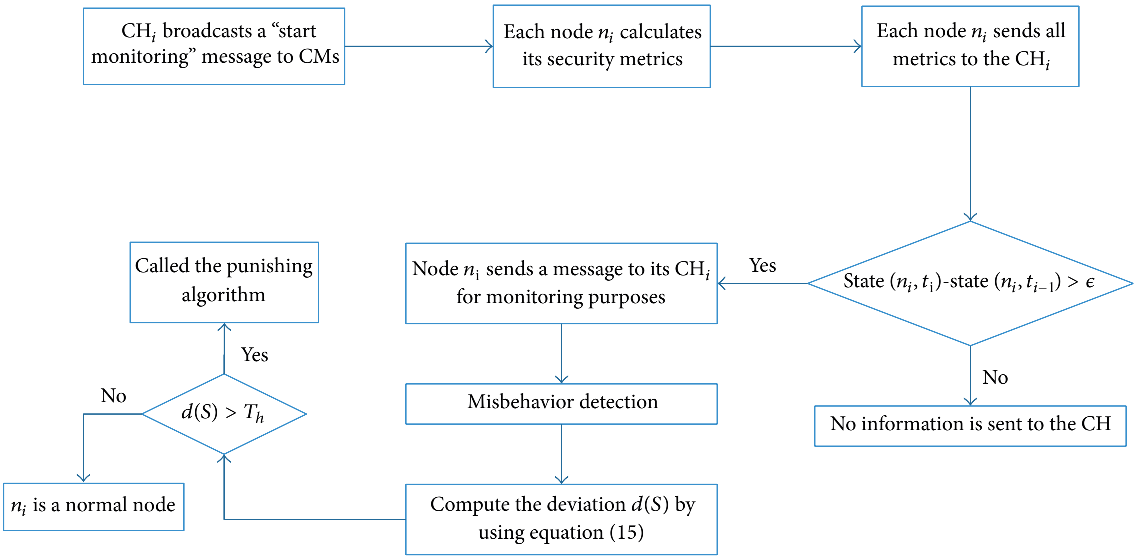

CH$_i$ broadcasts a "start monitoring" message to CMs
Each node n_i calculates its security metrics
Each node n_i sends all metrics to the CH$_i$
State (n_i, t_i)-state $(n_i, t_{i-1}) > \epsilon$
Yes
No
No information is sent to the CH
Node n_i sends a message to its CH$_i$ for monitoring purposes
Misbehavior detection
Compute the deviation $d(S)$ by using equation (15)
$d(S) > T_h$
Yes
No
Called the punishing algorithm
n_i is a normal node

Figure 11: Monitoring phase.

Algorithm 4 (monitoring phase algorithm). The monitoring process involves a series of steps as illustrated by the flowchart in (Figure 11).

Step 1 (this step runs in each CH$_i$). Each CH$_i$ becomes the monitor node of its cluster members and broadcasts a "*Start Monitoring*" message with its Id$_i$ to its entire cluster CMs.

Step 2 (calculation of security metrics performed by each member n_i of the cluster i). Each node n_i $(i <> j)$ receives the message "*Start Monitoring*" and calculates its security metrics as follows.

(i) Number of packets sent by n_i at time interval is $\Delta t = [t_0, t] : \mathit{Nbp\_Send}(ni, \Delta t)$.

(ii) Number of packets received by node n_i at time interval is $\Delta t = [t_0, t_0] : \mathit{Nbp\_Received}(n_i, \Delta t)$.

(iii) Delay between the arrivals of two consecutive packets is

$$\mathit{Delay\_BP}\left(n_i, t\right) = \mathit{Arrival\_PT}_i - \mathit{Arrival\_PT}_{i-1}. \quad (12)$$

(iv) Energy consumption: the energy consumed by the node j in receiving and sending packets is measured using the following equation:

$$Ec\left(n_i, \Delta t\right) = \mathrm{Er}\left(n_i, t_0\right) - \mathrm{Er}\left(n_i, t_1\right), \quad (13)$$

where Δt is the time interval $[t_0, t_1]$; $\mathrm{Er}(n_i, t_0)$ is the residual energy of node n_i at time t_0; $\mathrm{Er}(n_i, t_1)$ is the residual energy of node n_i at time t_1 and $Ec(n_i, \Delta t)$ is the energy consumption of node n_i at time interval Δt.

Step 3 (sending all metrics to the CH). After each consumption of the security metrics, the state of a node n_i at time t is denoted by state (n_i, t_i). For storage volume economy, each node keeps only the latest calculation state.

(i) In the initial deployment, each CM in cluster "i" sends some states $(\text{state}(n_i, t_i))$ to the CH$_i$ for making a normal behavior model of node n_i by using a learning mechanism.

(ii) Each state contains the following information:

$$\left(Id, Nbp_{Send(ni,\Delta t)}, Nbp_{Received(n_i,\Delta t)}, Delay_{BP(n_i,t)}, Ec\left(n_i, \Delta t\right)\right). \tag{14}$$

(iii) If (state (n_i, t_i) − state $(n_{i}, t_{i-1}) > \epsilon$)

then node n_i sends a message (ϵ a given threshold):

> $Msg = (Id, Nbp_{Send(ni,\Delta t)}, Nbp_{Received(n_i,\Delta t)}, Delay_{BP(n_i,t)}, Ec(n_i, \Delta t))$ to its CH$_i$ for monitoring purposes.

Otherwise, no information is sent to the CH.

(iv) The message received by CH$_i$ will be stored in a table Tmet for future analysis.

(v) If a sensor node n_i does not respond during this monitoring period, it will be considered as misbehaving.

(vi) The behavior level of sensor node n_i is computed using the following equation:

$$\mathrm{BL}_i = \mathrm{BL}_i - \text{rate}. \tag{15}$$

The "rate" is fixed on the basis of the nature of the application. For example, if it is fault tolerant or not. In our case, we took rate = 0.1.

Step 4 (misbehavior detection, which is performed by CH$_i$).

(i) For each node n_i in the cluster "i," the state in time slot "t" is expressed by the three-dimensional vector:

$$S = \left(S_{t1}, S_{t2}, S_{t3}\right), \tag{16}$$

where

(a) S_{t1} is the number of packets dropped by n_i, defined as follows:

$$S_{t1} = \sum Ps_{Received\ by\ n_i} - \sum Ps_{Sent\ by\ n_i} - \sum Ps_{destined\ by\ n_i}, \tag{17}$$

with

$$\sum Ps_{Received\ by\ n_i} = \sum Ps_{Sent by\ n_i} + \sum Ps_{destined by\ n_i} + \sum Ps_{lost\ by\ n_i}. \tag{18}$$

For a normal node, $S_{t1} \approx 0$.

(b) S_{t2} is the delay between the arrival of two consecutive packets:

$$S_{t2} = Delay\_BP\left(n_i, t\right). \tag{19}$$

(c) S_{t3} is the energy consumption:

$$S_{t3} = Ec\left(n_i, \Delta t\right). \tag{20}$$

Here, $t \in [t_0, \mathrm{t}] = \Delta t$.

(ii) In our case, the first interval is used for the training data set of n time slots. We calculate the mean vector $\overline{S}$ of S by using

$$\overline{S} = \frac{\sum_{t=t_0}^{t_{n-1}} S_t}{n}. \tag{21}$$

(iii) After modeling a normal behavior model for each sensor node, the behaviors of all nodes are sent to the base station for further analysis. We then compute the deviation $d(S)$ by using

$$d\left(S\right) = \left|S - \overline{S}\right|. \tag{22}$$

(iv) When the deviation $d(S)$ is larger than threshold T_h (which means that it is out of the range of normal behavior), it will be judged as a misbehaving node. In this case, the level of behavior is BL$_i \approx 0$. This is called the punishing algorithm:

$$\begin{aligned} &d\left(S\right) > T_h\text{: } n_i \text{ is an abnormal node} \\ &d\left(S\right) \leq T_h\text{: } n_i \text{ Is a normal node.} \end{aligned} \tag{23}$$

The punishing algorithm is presented in Algorithm 3.

6. Simulation Results

This section presents the implementation of the proposed approach using the Borland C++ language and the analysis of the obtained results.

6.1. The Simulator "Mercury". We try to complete the theoretical study by implementing our own wireless sensor network simulator "Mercury." On the other hand, a bit of simulators for WSNs such as TOSSIM [47] and Power-TOSSIM [48] are irrelevant with our goal and purpose and in order to avoid many complications we established our own mercury simulator. It is established on an object-oriented design and a distributed approach such as self-organization mechanism which is distributed at the level of each sensor; it provides a set of interfaces for configuring a simulation and for choosing the type of event scheduler used to drive the simulation. A simulation script generally begins by creating an instance of this class and calling various methods to create nodes and topologies and configure other aspects of the simulation. Mercury uses two routing protocols for delivering data from sensor nodes to the Sink station: a reactive protocol AODV (ad hoc on demand distance vector) [5] and a proactive protocol DSDV (destination sequenced distance vector) [6]. To determine and evaluate the results of the execution of algorithms that are introduced previously; the number of sensors to deploy must be inferior or equal to 1000. There are two types of sensor nodes deployment on the sensor field: random and manual. Mercury offers users the ability to select a sensor type from 5 types of existing sensor, each of them has its proper characteristics (radius, energy, etc.).

```
Begin
  $I := 0$;
  $I := I + 1$;
  If (($I$ = Rate) && ($BL_i <= 0.1$))
      // Rate: parameter of maximum number of faults
         defined by the administrator
      $BL_i = BL_i - \text{Rate}$;
      // Classification of the node according to its $BL_i$
      $\text{Mp}(BL_i) = \begin{cases} \text{Normal node: } 0.8 \le BL_i \le 1 \\ \text{Abnormal node: } 0.5 \le BL_i < 0.8 \\ \text{Suspect node: } 0.3 \le BL_i < 0.5 \\ \text{Malicious node: } 0 \le BL_i < 0.3 \end{cases}$
      If ($BL_i \le 0.3$) Then
        If ($n_i$ is CM) Then
            Suppression of the node of the list of the members;
          Addition of the node to the blacklist;
        EndIf
        If ($n_i$ is CH) Then              // CH: Cluster Head
            Addition of the node to the blacklist;
            Set up Phase;
        EndIf
      EndIf
  EndIf
End.
```

Algorithm 3: Punishing algorithm.

Unity of the energy used is as Nanojoules: (1 Joule = 10^9 NJ). Mobility has influence on energy and the behavior of sensors; for instance, if the sensor moves one meter away from its original location, its energy will diminish by 100,000 NJ and its behavior will also decrease by 0.001 units. This allows users to differentiate a malicious node (that moves frequently) of a legitimate node (that can changes position with reasonable distances). Since sensors nodes move due to the forces acting from the outside, no power consumption for mobility must be taken into consideration in all simulations that we have carried for evaluation [4].

6.2. Discussion and Results. To evaluate our ES-WCA algorithm, we have performed extensive simulation experiments. This section provides our experimental results and discussions. In all the experiments, N varies between 10 and 1000 sensor nodes. The transmission range (R) varies between 10 and 175 meters (m) and the used energy (E) is equal to 50000 NJ. The sensor nodes are randomly distributed in a "570 m × 555 m" space area by the following function:

```
for (int n = 0; n < node_tobe_deployed; n + +).

    {
    X_ = rand()  %  image_Field_Of_Collecting
    → width;
    Y_ = rand()  %  image_Field_Of_Collecting
    → Height;
    }
```

The performance of the proposed ES-WCA algorithm is measured by calculating (i) the number of clusters, (ii) number of reaffiliations, (iii) choice of ES-WCA with AODV or DSDV, and (iiii) detection of misbehavior nodes and the nature of attacks during the distributed monitoring process.

In our experiments, the values of weighting factors used in the weight calculation are as follows: $w_1 = 0.3$, $w_2 = 0.2$, $w_3 = 0.2$, $w_4 = 0.2$, and $w_5 = 0.1$. It is noted that these values are arbitrary at this time and for this reason they should be adjusted according to the system requirements. To evaluate the performance of the proposed ES-WCA algorithm by comparing it with alternative solutions, we studied the effect of the density of the networks (number of sensor nodes in a given area) and the transmission range on the average number of formed clusters. Then we compared it with a WCA proposed in [3], DWCA proposed in [9], and SDCA proposed in [21].

Figure 12 illustrates the variation of the average number of clusters with respect to the transmission range. The results are shown for N which varies between 200 and 1000. We found that there is opposite relationship between clusters and transmission range. This is on the grounds that a cluster head with a considerable transmission range will cover a large area.

Figure 13 depicts the average number of clusters that are formed with respect to the total number of nodes in the network. The communication range used in this experiment is 200 m. From Figure 13, it is seen that ES-WCA consistently provides about 61.91% less clusters than DWCA and about 38.46% than SDCA, when there were 100 nodes in the network. When the node number is equal to 20 nodes,

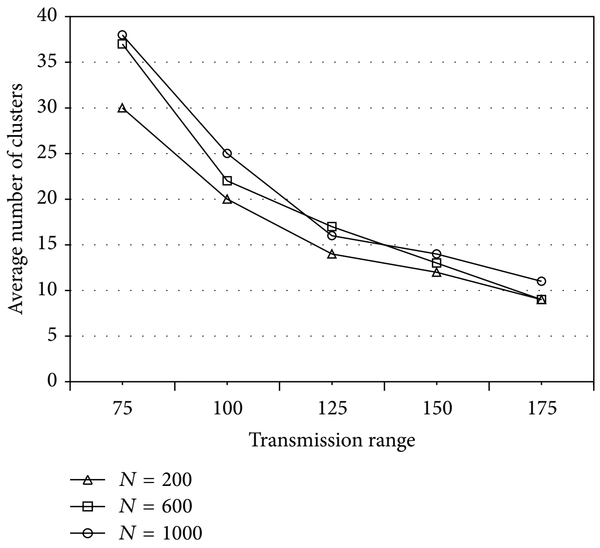


Figure 12: Average number of clusters versus transmission range (R).

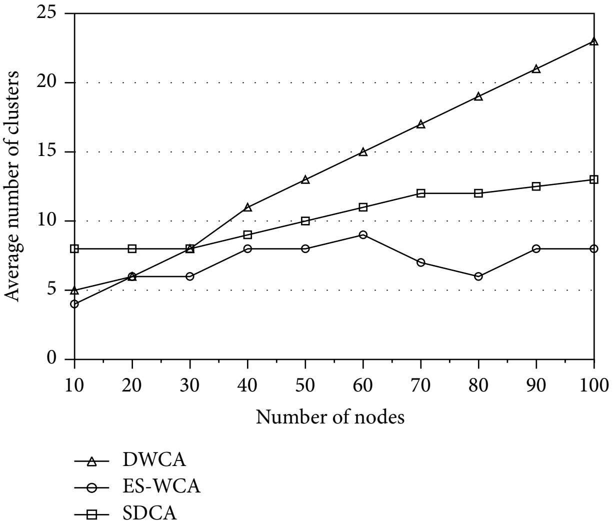


Figure 13: Average number of clusters versus number nodes (N) for ES-WCA, DWCA, and SDCA.

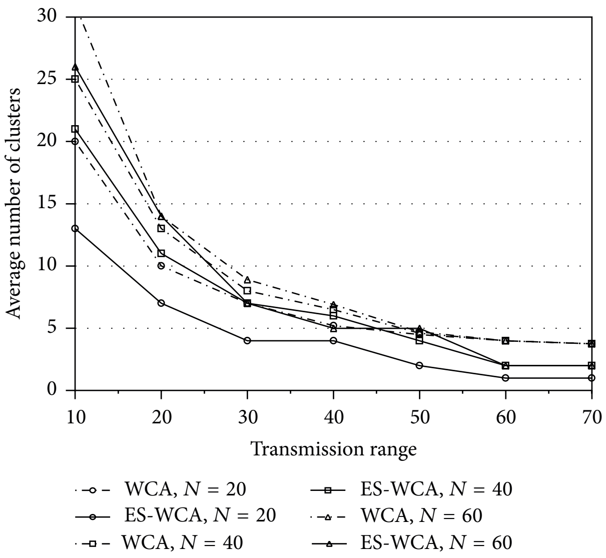


Figure 14: Average number of clusters versus transmission range ES-WCA and WCA.

the performance of ES-WCA is similar to DWCA in terms of number of clusters; however, if the node density had increased, ES-WCA would have produced constantly less clusters than SDCA and DWCA, respectively, regardless of the node number. Because of the use of a random deployment, the result of ES-WCA is unstable between 60 and 90. So, the increase in the number of clusters depends on the increase of the distance between the nodes. As a result, our algorithm gave better performance in terms of the number of clusters when the node density in the network is high, and this is due to the fact that ES-WCA generates a reduced number of balanced and homogeneous clusters, whose size lies between two thresholds: $Thresh_{Upper}$ and $Thresh_{Lower}$ (reaffiliation phase) in order to minimize the energy consumption of the entire network and prolong sensors lifetime.

Figure 14 shows the variation of the average number of clusters with respect to the transmission range. The results are shown for varying N. We notice an inverse relationship, and the average number of clusters decreases with the increase in the transmission range. As shown in Figure 14, the proposed algorithm produced 16% to 35% fewer clusters than WCA [3] when the transmission range of nodes was 10 m. When the node density increased, ES-WCA constantly produced less clusters than WCA regardless of the node number. With 70 nodes in the network, the proposed algorithm produced about 47% to 73% less clusters than WCA. The results show that our algorithm gave a better performance in terms of the number of clusters when the node density and transmission range in the network are high.

Figure 15 interprets the average number of reaffiliations that are established with esteem to the total number of nodes in the network. The number of reaffiliations incremented linearly when there were 30 or more nodes in the network for both WCA and DWCA. But for our algorithm, the number of reaffiliations increased starting from 50 nodes. We submit to engender homogeneous clusters whose size is between two thresholds: $Thresh_{Upper} = 18$ and $Thresh_{Lower} = 9$. According to the results, our algorithm presented a better performance in terms of the number of reaffiliations. The benefit of decreasing the number of reaffiliations mainly comes from the localized reaffiliation phase in our algorithm. The result of the remaining amount of energy per node for each protocol AODV and DSDV is presented in Figure 16 such as R equal to 35 m. As shown in the above-mentioned figure, the remaining energy for each node in AODV protocol is greater than that in DSDV protocol such as AODV which consumes 22, 74% less than DSDV. According to the results, the network consumes 19, 23% of the total energy when we use an AODV protocol (192322.091 NJ). However, it consumes

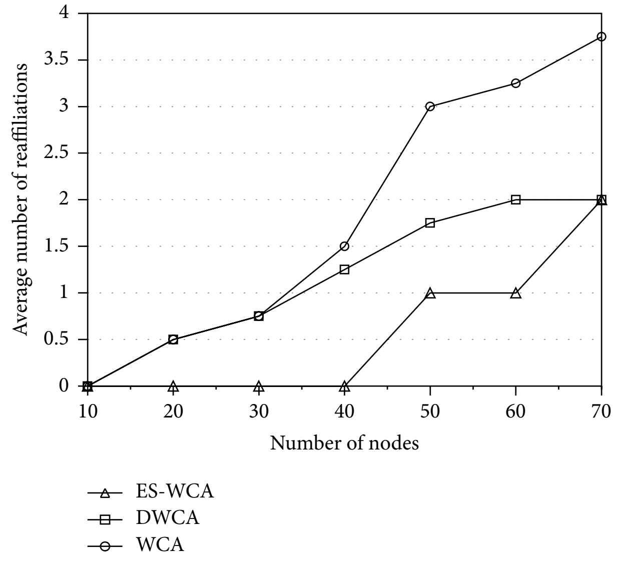


Figure 15: Average number of reaffiliations.

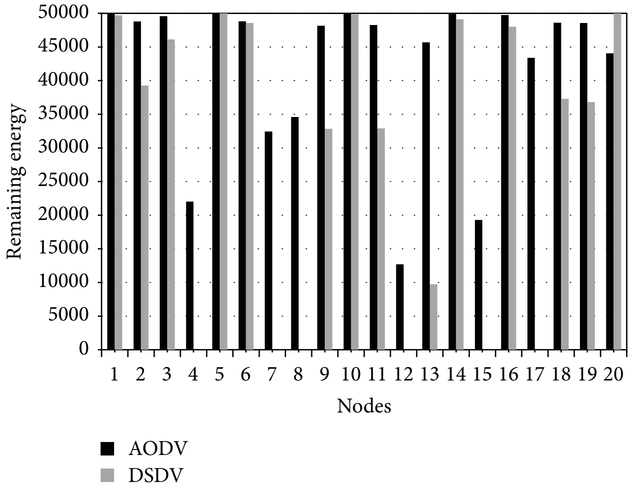


Figure 16: Remaining energy per node using ES-WCA.

41, 97% with a DSDV protocol (419740.129 NJ). We also observe that the network lost 6 nodes with DSDV but only one node with AODV because of the depletion of its battery. This result clearly shows that AODV outperforms DSDV. This is due to the tremendous overhead incurred by DSDV when exchanging routing tables and the periodic exchange of the routing control packets. So, our algorithm gave a better performance in terms of saving energy when it is coupled with AODV.

We consider that the network will be inoperative when the nodes of the neighborhood of the sink exhaust their energy as exemplified. In Figure 17, we appraise the network lifetime by changing the number of nodes such as R equal to 70 m. When there were 20 nodes in the network, AODV increases the network period about 88, 47% compared to DSDV and about 57,9% for $N = 100$. Also, this is for the reason that in a DSDV protocol each node must have a global view of the network. This in turn raises the number of the exchanged control packets (overhead) in the full network and it decreases the residual energy of each node which has a direct effect on the network lifetime. There are 9 nodes in an active state but the network is inoperative. We discover that the increase in the total of nodes does not have a powerful factor on the network lifetime except between $N = 60$ and $N = 80$.

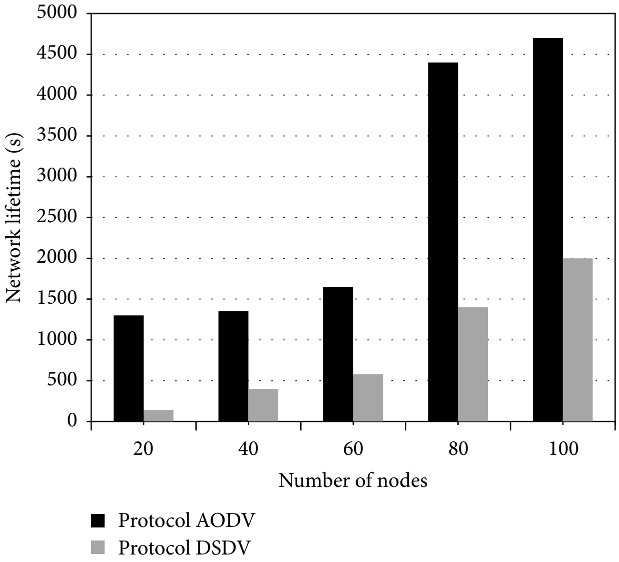


Figure 17: Network lifetime depending on number of nodes using ES-WCA.

Table 3: Detection of the nature of attacks.

IDs	Packets_Sent	Packets_Received	Attack
41	(19, 13)	(16, 14)	Node Outage
71	(24, 152)	(20, 34)	Hello Flood
162	(15, 8)	(22, 112)	Sinkhole
181	(16, 179)	(26, 42)	Hello Flood
190	(58, 32)	(50, 51)	Black Hole

To illustrate the effect of abnormal behavior in the network, in our experiments we propagated 200 nodes with 5 malicious nodes. The cases of the malicious nodes will pass from a normal node with a yellow color to an abnormal node with a blue color, to a suspicious node with a grey color, and lastly, to a malicious node with a black color. All the cases of the CMs are discovered by their CH. Malicious CHs are disclosed by the base station.

Figure 18(b) displays the total of clusters established according to the transmission range. Figures 19(a), 19(b), and 19(c) display the measure results for a scenario with malicious nodes which are achieved by the generator of bad behavior. The generated attacks are explained in Section 3. We can identify that these nodes migrate from a normal case to an abnormal or suspicious state and finally to a malicious state as expected. Table 3 presents the Ids of malicious nodes and

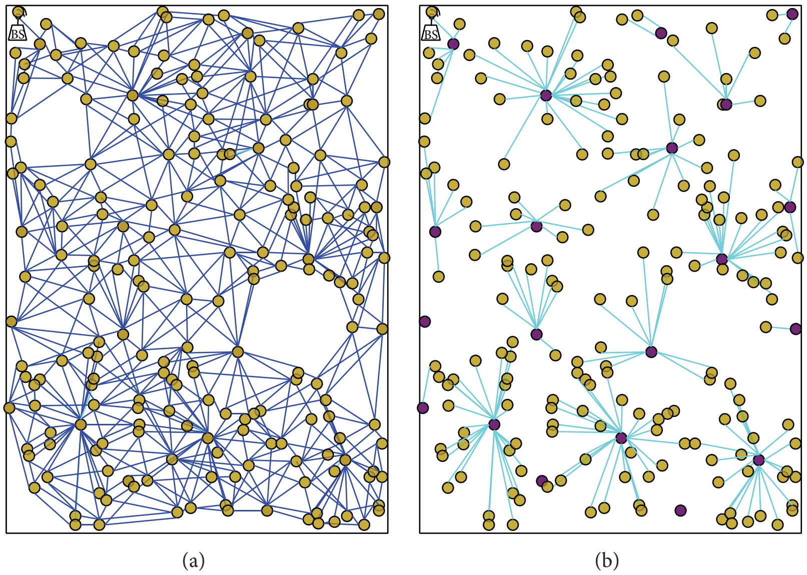


Figure 18: (a) Graph connectivity of 200 nodes. (b) Network after clustering formation.

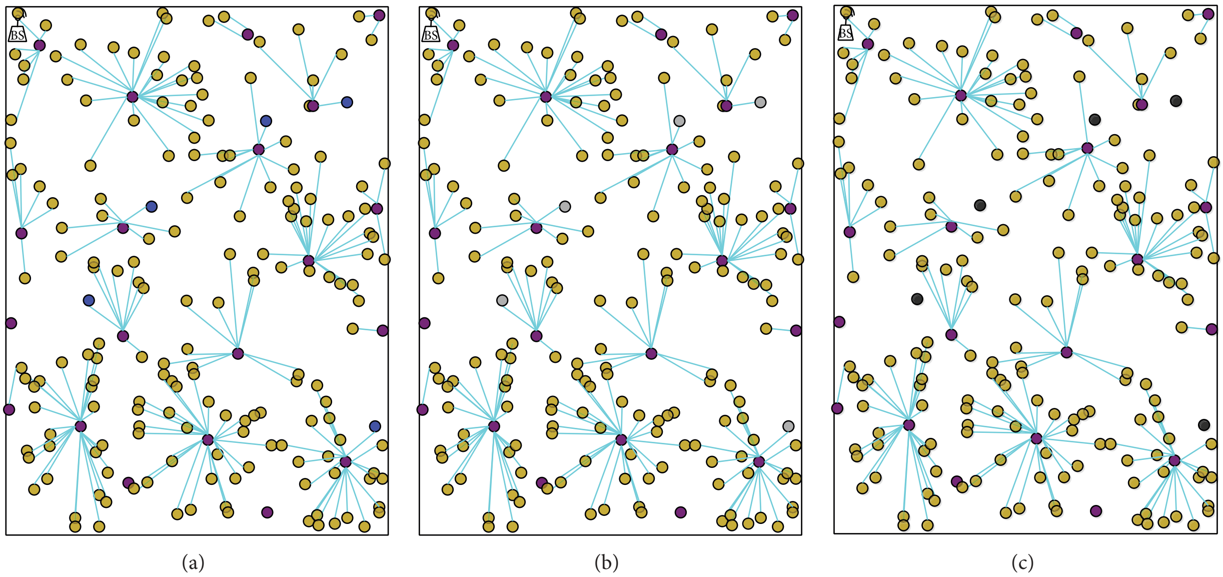


Figure 19: (a) Sensors with a blue color are abnormal but not malicious. (b) The grey sensors have a suspect behavior. (c) The sensors with a black color are compromised and are exhibiting malicious behavior.

their categories of attacks in the course of the dissemination of a monitoring mechanism in the network by the follow-up of the messages exchanged between the nodes. When Packets_sent [$N1, N2$], Packets_received [$N3, N4$]. Thus, $N1$ is the total of packets sent before attacks, and $N2$ is the total of packets sent after attacks, while $N3$ is the total of packets received before attacks and $N4$ is the total of packets received after attacks. We regard that these malicious nodes increment $N1$, as the sensors (71, 181), reduce $N1$, like the sensor (190), increment $N3$, as the sensor (162), and lastly break sending data like node (41). From Figure 20 it is observed that the sensor nodes (3, 17) are malicious and have a behavior level less than 0.3, its behavior decreased by 0.1 units, and when the monitor (CH) counts one failure an alarm is raised. However, packets from malicious nodes are not processed and no packet will be forwarded to them. The sensor node (11) has the behavior level less then threshold behavior so it will not be accepted as a CH candidate even if it has the other interesting characteristics (Er_i, C_i, D_i, and M_i). On the other side the behavior level in Figure 21 decreased by 0.001 units in our first work [4] when the malicious node moves frequently. We note that sensor (6) is suspicious so if it continues to move frequently its behavior will gradually be decreased until it reaches the malicious state; in this case this node will be

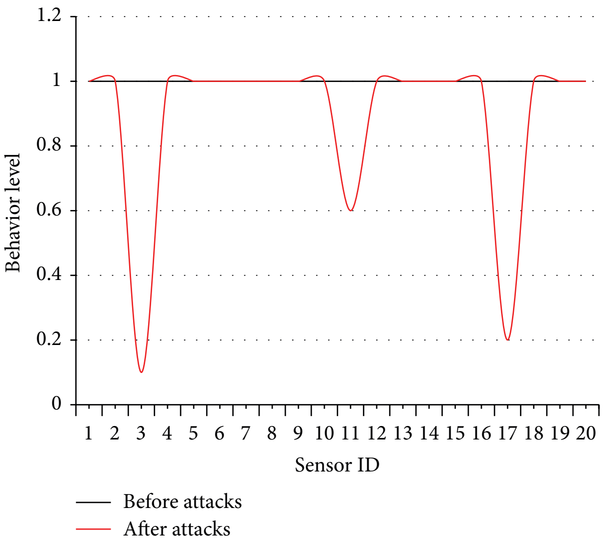


Figure 20: Behavior level of some sensors (moves frequently).

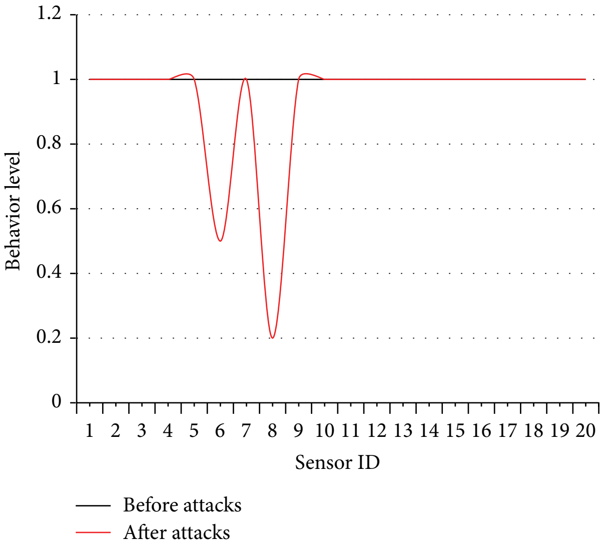


Figure 21: Behavior level of some sensors before and after attacks.

deleted from the neighborhood and finally it will be added to the black list.

7. Conclusion and Future Works

In this paper, we have presented a new algorithm called "ES-WCA" for promoting the self-organization of mobile sensor networks. This algorithm is fully decentralized and aims at creating a virtual topology with the purpose to minimize frequent reelection of the cluster head (CH) and avoid overall restructuring of the entire network. Simulations result attest of the outperformance of our algorithm compared to WCA and DWCA in every sense. It yields a low number of clusters and it preserves the network structure better than WCA and DWCA by reducing the number of reaffiliations. The proposed algorithm selects the most robust and safe CHs with the responsibility of monitoring the nodes in their clusters and maintaining clusters locally. Our third algorithm analyses and detects specific misbehavior in the WSNs. The results show that in scenarios in which mobile WSNs are with a low density or with a small size, the choice of ES-WCA with AODV is comparable to ES-WCA with DSDV to show clearly the interest of the routing protocols in energy saving. However, the difference in favor between ES-WCA and AODV becomes very important in case of a high node density. This is due to the tremendous overheads incurred by ES-WCA with DSDV when exchanging routing tables and exchanging routing control packets. Future work includes considering further the concept of redundancy by using the "sleep" and "wakeup" mechanism in case of node failure, providing in-network processing by aggregating correlated data in order to reduce both the energy consumption and the congestion issue.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors are grateful to the anonymous referees for their insightful comments and valuable suggestions, which greatly improved the quality of the paper.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[3] M. Chatterjee, S. Das, and D. Turgut, "WCA: a weighted clustering algorithm for mobile ad hoc networks," *Journal of Cluster Computing*, vol. 5, no. 2, pp. 193–204, 2002.

[4] A. Dahane, N. E. Berrached, and B. Kechar, "Energy efficient and safe weighted clustering algorithm for mobile wireless sensor networks," in *Proceedings of the 9th International Conference on Future Networks and Communications (FNC '14)*, vol. 34, pp. 63–70, Procedia Computer Science (Elsevier), Niagara Falls, Canada, August 2014.

[5] Q. Dong and W. Dargie, "A survey on mobility and mobility-aware MAC protocols in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 88–100, 2011.

[6] M. Ali, T. Suleman, and Z. A. Uzmi, "MMAC: a mobility-adaptive, collision-free MAC protocol for wireless sensor networks," in *Proceedings of the 24th IEEE International Performance, Computing, and Communications Conference (IPCCC '05)*, pp. 401–407, IEEE, April 2005.

[7] Y. Yu and L. Zhang, "A secure clustering algorithm in mobile ad-hoc networks," in *Proceedings of the IACSIT Hong Kong Conferences*, vol. 29, pp. 73–77, 2012.

[8] I. I. Er and W. K. G. Seah, "Mobility-based d-hop clustering algorithm for mobile ad hoc networks," in *Proceedings of the*

IEEE Wireless Communications and Networking Conference (WCNC '04), pp. 2359–2364, March 2004.

[9] W. Choi and M. Woo, "A distributed weighted clustering algorithm for mobile ad hoc networks," in *Proceedings of the IEEE Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW '06)*, p. 73, February 2006.

[10] A. Zabian, A. Ibrahim, and F. Al-Kalani, "Dynamic head cluster election algorithm for clustered Ad-Hoc networks," *Journal of Computer Science*, vol. 4, no. 1, pp. 42–50, 2008.

[11] M. Chawla, J. Singhai, and J. L. Rana, "Clustering in mobile ad- hoc networks: a review," *International Journal of Computer Science and Information Security*, vol. 8, no. 2, pp. 293–301, 2010.

[12] R. Agarwal, R. Gupta, and M. Motwani, "Review of weighted clustering algorithms for mobile ad-hoc networks," *Computer Science and Telecommunications*, vol. 33, no. 1, pp. 71–78, 2012.

[13] H. Safa, H. Artail, and D. Tabet, "A cluster-based trust-aware routing protocol for mobile ad hoc networks," *Wireless Networks*, vol. 16, no. 4, pp. 969–984, 2010.

[14] Sikander, M. Zafar, A. Raza, M. Babar, S. Mahmud, and G. Khan, "A survey of cluster-based routing schemes for wireless sensor networks," *Smart Computing Review Networks*, vol. 3, no. 4, pp. 261–275, 2013.

[15] X. Liu, "A survey on clustering routing protocols in wireless sensor networks," *Sensors*, vol. 12, no. 8, pp. 11113–11153, 2012.

[16] S. Soro and W. B. Heinzelman, "Cluster head election techniques for coverage preservation in wireless sensor networks," *Ad Hoc Networks*, vol. 7, no. 5, pp. 955–972, 2009.

[17] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications Journal*, vol. 30, no. 14-15, pp. 2826–2841, 2007.

[18] K. A. Darabkh, S. S. Ismail, M. Al-Shurman, I. F. Jafar, E. Alkhader, and M. F. Al-Mistarihi, "Performance evaluation of selective and adaptive heads clustering algorithms over wireless sensor networks," *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 2068–2080, 2012.

[19] V. Geetha, P. Kallapur, and S. Tellajeera, "Clustering in wireless sensor networks: performance comparison of LEACH & LEACH-C protocols using NS2," *Procedia Technology*, vol. 4, pp. 163–170, 2012.

[20] Y. Wang, X. Wu, J. Wang, W. Liu, and W. Zheng, "An OVSF code based routing protocol for clustered wireless sensor networks," *International Journal of Future Generation Communication and Networking*, vol. 5, no. 3, pp. 117–128, 2012.

[21] K. Benahmed, M. Merabti, and H. Haffaf, "Distributed monitoring for misbehaviour detection in wireless sensor networks," *Security and Communication Networks*, vol. 6, no. 4, pp. 388–400, 2013.

[22] J. Y. Yu and P. H. J. Chong, "A survey of clustering schemes for mobile ad hoc networks," *IEEE Communications Surveys and Tutorials*, vol. 7, no. 1, pp. 32–47, 2005.

[23] J. Y. Yu and P. H. J. Chong, "A survey of clustering schemes for mobile ad hoc networks," *IEEE Communications Surveys and Tutorials*, vol. 7, no. 1–4, pp. 32–47, 2005.

[24] A. Jain and B. V. R. Reddy, "A novel method of modeling wireless sensor network using fuzzy graph and energy efficient fuzzy based k-hop clustering algorithm," *Wireless Personal Communications*, vol. 82, no. 1, pp. 157–181, 2015.

[25] T. Kavita and D. Sridharan, "Security vulnerabilities in wireless sensor networks: a survey," *Journal of Information Assurance and Security*, vol. 5, pp. 31–44, 2010.

[26] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002.

[27] S. Ganeriwal and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," in *Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04)*, pp. 66–77, October 2004.

[28] Y. Yu and L. Zhang, "A secure clustering algorithm in mobile ad-hoc networks," in *Proceedings of the 2012 IACSIT Hong Kong Conferences*, vol. 29, pp. 73–77, Hong Kong, 2012.

[29] X. Liu, "A survey on clustering routing protocols in wireless sensor networks," *Sensors*, vol. 12, no. 8, pp. 11113–11153, 2012.

[30] T. H. Hai, E.-N. Huh, and M. Jo, "A lightweight intrusion detection framework for wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 10, no. 4, pp. 559–572, 2010.

[31] M. E. Elhdhili, L. B. Azzouz, and F. Kamoun, "Reputation based clustering algorithm for security management in ad hoc networks with liars," *International Journal of Information and Computer Security*, vol. 3, no. 3-4, pp. 228–244, 2009.

[32] S. Taneja and A. Kush, "A survey of routing protocols in mobile ad-hoc networks," *International Journal of Innovation, Management and Technology*, vol. 1, no. 3, pp. 279–285, 2010.

[33] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proceedings of the Conference on Communications Architectures, Protocols and Applications (SIGCOMM '94)*, pp. 234–244, ACM, London, UK, September 1994.

[34] D. G. Padmavathi and D. Shanmugapriya, "A survey of attacks, security mechanisms and challenges in wireless sensor networks," *International Journal of Computer Science and Information Security*, vol. 4, no. 1-2, pp. 1–9, 2009.

[35] P. Li, L. Sun, X. Fu, and L. Ning, "Security in wireless sensor networks," in *Wireless Network Security*, pp. 179–227, Higher Education Press, Springer, Berlin, Germany, 2013.

[36] W. Stallings, *Cryptography and Network Security, Principles and Practice*, Prentice Hall, 5th edition, 2010.

[37] M. Safiqul-Islam and S. Ashiqur-Rahman, "Anomaly intrusion detection system in wireless sensor networks: security threats and existing approaches," *International Journal of Advanced Science and Technology*, vol. 36, pp. 1–8, 2011.

[38] P. Berwal, "Security in wireless sensor networks: issues and challenges," *International Journal of Engineering and Innovative Technology*, vol. 3, no. 5, pp. 192–198, 2013.

[39] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Ad-Hoc Networks Journal*, vol. 1, no. 2-3, pp. 293–315, 2003.

[40] S. Dai, X. Jing, and L. Li, "Research and analysis on routing protocols for wireless sensor networks," in *Proceedings of the International Conference on Communications, Circuits and Systems*, vol. 1, pp. 407–411, May 2005.

[41] M. Tripathi, M. S. Gaur, and V. Laxmi, "Comparing the impact of black hole and gray hole attack on LEACH in WSN," in *Proceedings of the 4th International Conference on Ambient Systems, Networks and Technologies (ANT '13) and the 3rd International Conference on Sustainable Energy Information Technology (SEIT '13)*, vol. 19, pp. 1101–1107, June 2013.

[42] R. A. Shaikh, H. Jameel, S. Lee, S. Rajput, and Y. J. Song, "Trust management problem in distributed wireless sensor networks," in *Proceedings of the 12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA '06)*, pp. 411–414, August 2006.

[43] M. Lehsaini, H. Guyennet, and M. Feham, "An efficient cluster-based self-organisation algorithm for wireless sensor networks," *International Journal of Sensor Networks*, vol. 7, no. 1-2, pp. 85–94, 2010.

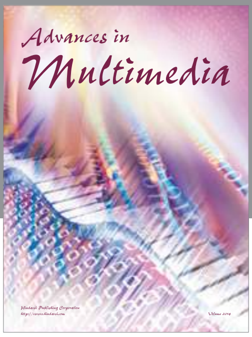
[44] Y. Li, F. Wang, F. Huang, and D. Yang, "A novel enhanced weighted clustering algorithm for mobile networks," in *Proceedings of the IEEE 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '09)*, pp. 2801–2804, IEEE, Beijing, China, September 2009.

[45] A. da Silva, M. Martins, B. Rocha, A. Loureiro, L. Ruiz, and H. Wong, "Decentralized intrusion detection in wireless sensor networks," in *Proceedings of the 1st ACM International Workshop on Quality of Service & Security in Wireless and Mobile Networks*, pp. 16–23, 2005.

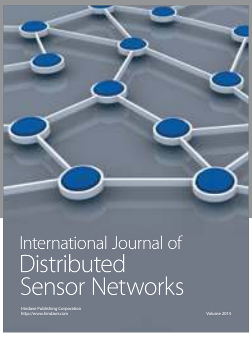
[46] K. Benahmed, H. Haffaf, and M. Merabti, "Monitoring of wireless sensor networks," in *Sustainable Wireless Sensor Networks*, Y. K. Tan, Ed., chapter 3, InTech, 2010.

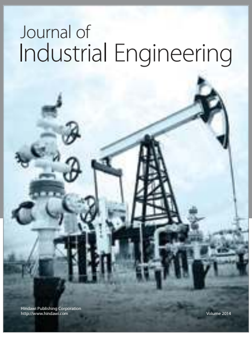
[47] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: accurate and scalable simulation of entire TinyOS applications," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 126–137, November 2003.

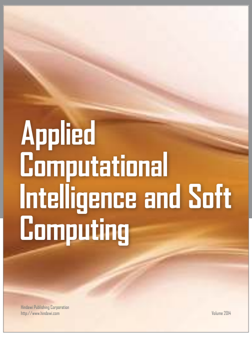
[48] V. Shnayder, M. Hempstead, B.-R. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 188–200, November 2004.

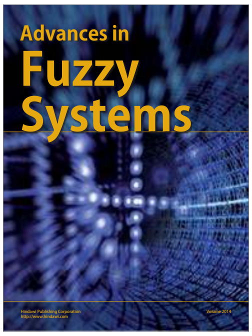

Advances in
Multimedia

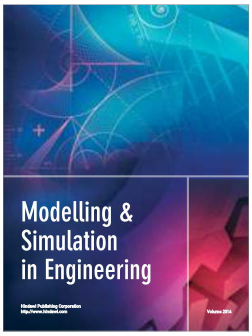
The Scientific
World Journal

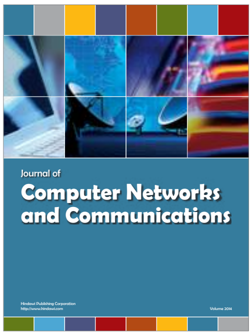

International Journal of
Distributed
Sensor Networks

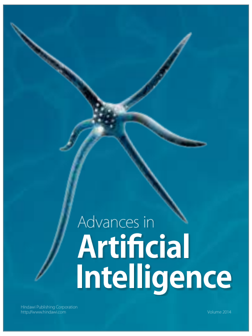

Journal of
Industrial Engineering

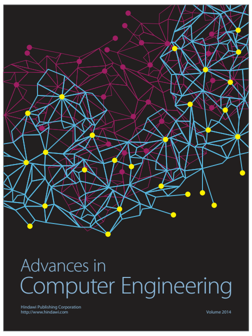

Applied
Computational
Intelligence and Soft
Computing


Advances in
Fuzzy
Systems

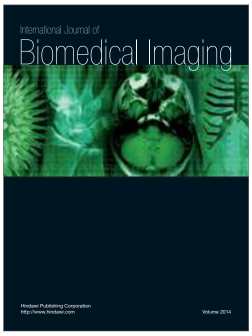

Modelling &
Simulation
in Engineering

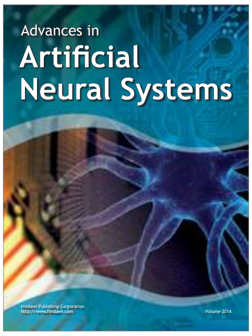

Journal of
Computer Networks
and Communications

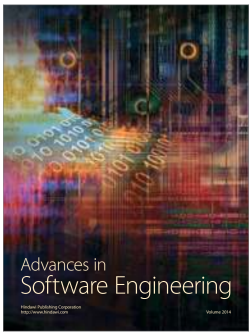

Advances in
Artificial
Intelligence


Advances in
Computer Engineering

International Journal of
Computer Games
Technology

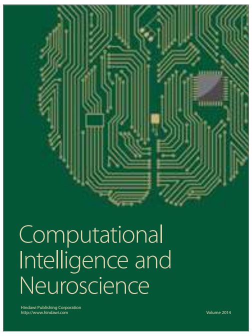

International Journal of
Biomedical Imaging


Advances in
Artificial
Neural Systems


Advances in
Software Engineering

Journal of
Robotics

Advances in
Human-Computer
Interaction


Computational
Intelligence and
Neuroscience

International Journal of
Reconfigurable
Computing

Journal of
Electrical and Computer
Engineering