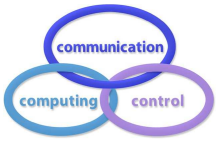
INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 15, Issue: 5, Month: October, Year: 2020
Article Number: 3825, https://doi.org/10.15837/ijccc.2020.5.3825



CCC Publications

Reduce Effect of Dependent Malicious Sensor Nodes in WSNs using Pairs Counting and Fake Packets

A. Ahmed, M. Hababeh, A. Abu-Hantash, Y. AbuHour, H. Musleh

Ashraf Ahmed*
Computer Graphics and Animation Department
Princess Sumaya University for Technology, Jordan
P.o.Box 1438 Al-Jubaiha, Amman 11941, Jordan
*Corresponding author: a.ahmad@psut.edu.jo

Mohammed Hababeh
Computer Engineering, German Jordanian University
Amman Madaba Street, P.O. Box 35247, Jordan
m.hababeh@gju.edu.jo

Alaa Abu-Hantash, Yousef AbuHour, Husam Musleh
University of Science and Technology, Irbid, Jordan
P.O.Box 3030, Irbid 22110, Jordan
amabuhantash12@cit.just.edu.jo
ysabuhour14@sci.just.edu.jo
hskhairimusleh13@cit.just.edu.jo

Abstract

In this paper, we propose a new technique for the enhancement of target detection in Wireless Sensor Networks (WSNs) in which sensor nodes are responsible for taking binary decisions about the presence or absence of a given target and reporting the output to the fusion center. We introduce the algorithm; Fail Silent Pair (FSP) to calculate global decision in the fusion center. The FSP algorithm randomly distributes all sensor nodes into pairs then considers pairs of the same local decision. Also, we present new detection and prevention methods to reduce the effect of dependent malicious sensor nodes. The detection method is based on the deception of suspicious sensor nodes with fake packets to detect a subset of the malicious sensor nodes, as these nodes eavesdrop on other sensor nodes packets to use their local decisions as a reference to build an intelligent decision. While the prevention method allows the fusion center to correct local decisions of some malicious sensor nodes with identified strategies, assisting in the increase of the accuracy of global decisions. We introduce a mathematical analysis to verify our methods, in addition to simulation experiments to validate our technique.

Keywords: Wireless Sensor Networks (WSN), dependent malicious sensor nodes, detection and prevention methods.

1 Introduction

Recently, autonomous sensors have been used in a wide range of various applications. An autonomous sensor consists of the sensor itself and a low-power computational unit. There are many types of autonomous sensors that can measure different physical phenomena such as temperature, pressure, sound, vibration, or proximity [19]. The sensor type is selected based on its use. For example, a temperature sensor can be used to measure the temperature of a car engine. Furthermore, a location sensor can be used on a specific area to detect the existence of an object. Since most implemented applications require many autonomous sensors, manufacturers tend to produce cost effective sensors using low-cost components [7].

A Sensor network is formed by many autonomous sensors. In a wired sensor network, usually, the star topology is used. A wired sensor network consists of one dominant central node and a group of sensor nodes physically connected to the central node. All sensor nodes send measurements directly to the central node to calculate the final decision.

Wired sensor networks were widely used for network reliability due to low interference, low latency, and privacy [10]. However, the main problem with wired sensor networks is the scalability, specifically when covering large areas. Adding a new sensor node to the network requires high-cost wire cabling. along with the complexity of network management. Therefore, Wireless Sensor Networks (WSNs) [15] were introduced to overcome these problems.

A WSN can be defined as a network of devices, denoted as sensor nodes, which sense the environment and send information gathered from the monitored field (e.g. an area or a volume) through wireless links [4] to a base station called the fusion center. The fusion center is responsible for calculations and decision making. WSNs are flexible, cover large areas, and easy to deploy [12]. WSNs work in various hostile environments difficult for humans to reach. Moreover, WSNs are inexpensive and consume low power. Due to these features, WSNs attracted the attention of researchers, and as a result, several applications were implemented using WSNs [16], [11], and [2].

WSNs are more vulnerable to communication failures due to its nature, being that WSNs have dynamic network topology which means the number of nodes can vary with time Zhang2016. These disadvantages make the communication in WSNs less reliable and increase interference. In addition, network sensor nodes have low computational power [6]. In brief, WSNs are more vulnerable to attack than wired sensor networks. Therefore, using WSNs for critical applications such as military applications is risky [17].

WSNs are exposed to different types of attacks in [20] and [3]. For instance, attacks aimed to affect the calculations of the fusion center, in which malicious sensor nodes send misleading packets to sabotage the WSN global decision. Network sensor nodes cannot be distinguished due to of the lack of unique identifiers for each node because of the large number of nodes. Additionally, network sensor nodes can easily connect and disconnect from the network to save power. Thus, malicious sensor nodes can easily attack WSNs.

Network sensor nodes in WSNs cannot run complex topologies, and unable to use malicious software detection algorithms. Due to these limitations, WSNs suffer from several problems such as lack of privacy [18] (which is the main security concern) because of the interference between WSNs. Thus, creating an algorithm which runs in the fusion center has been a research challenge. These algorithms must include intelligent methods to assist the fusion center with calculating correct decisions as well as dealing with malicious sensor nodes.

Various types of attacks have been proposed. In [1], attacks have been classified into dependent and independent attacks. In independent attacks, malicious sensor nodes behave statically during the entire period of the attack. For example, detecting the existence of an object requires binary location sensors. Binary location sensors respond either with 1 (present) or 0 (absent). Accordingly, the fusion center calculates the final result after collecting the measurements from all sensor nodes. Based on that, independent malicious sensor nodes send false measurements to the fusion center. An independent malicious sensor node follows one of the following patterns:

- Always-Zero; in which the malicious sensor node always sends zero regardless of the correct measured value.

- Always-One; in which the malicious sensor node always sends one regardless of the correct measured value.
- Always-False; in which the malicious sensor node always sends the opposite of the measured value.

Therefore, detection of independent attack is easy as it follows a simple pattern.

In dependent attacks, malicious sensor nodes have the ability to modify the values sent to the fusion center. These sensor nodes can make advanced and intelligent decisions based on WSN's behavior. If the malicious sensor nodes are unable to affect the fusion center decision, they will act normally in order to avoid exposure.

In this paper, we present a new technique for the management of the decision-making process in the fusion center. We have divided the technique into three phases; the Fail Silent Pair (FSP) algorithm, the detection phase, and the prevention phase. The fusion center uses the FSP algorithm to calculate the global decision. While the detection and prevention phases are used to detect a subset of the malicious sensor nodes and prevent them from influencing the global decision.

The rest of this paper is organized as follows. Section 2 gives some related works. Section 3 presents the system model. Section 4 details the proposed methods. Section 5 illustrates the simulation results. Finally, discussion and conclusions are drawn in Section 6 and Section 7, respectively.

2 Related work

In literature, many detection and prevention methods of malicious sensor nodes in WSNs have been proposed. Authors of [1] proposed an algorithm for detecting all types of malicious sensor nodes in WSNs, which has enhanced the reliability of network reports. Since the performance of an intelligent node heavily depends on its reporting order, the algorithm is based on changing the reporting order of all sensor nodes in the network periodically. This approach is effective and provides accurate results. However, it only works for the long-term and increases the overhead of the fusion center as well as its processing time, considering the results depend on many rounds to increase the detection accuracy.

Alternative methods proposed in [5], [14]. The Autoregressive (AR) model was used to detect and block the malicious sensor nodes. Each sensor node sends its output to the base station, and an autoregressive predictor computes an estimated value for each node. The base station detects a sensor node as a malicious node and blocks it if the difference between the sensor's output and the estimated value is greater than a chosen threshold. In addition to the autoregressive predictor, a neural predictor is used in [14] to calculate the estimated value of each node.

Another detection approach proposed in [8]. The authors introduced an abnormality-detection approach based on the abnormality detection in data mining. In the independent attacks, it is shown that the attacker can always be detected as the number of spectrum sensing rounds tends towards infinity. In contrast, a dependent attack is applicable, in which the attackers can avoid detection if they possess exact information on the miss-detection and false-alarm probabilities.

In [9], attackers have been divided into two types; a static attacker who sends false data to the sink, and a dynamic attacker who sends both correct and incorrect information to the sink. Malicious node detection has been studied under the Byzantine attack, in which the attacker has full control over a subset of verified nodes. The attacker may confuse the system by transmitting packets through non-optimal paths or dropping packets. This can be done using distributed discovery and probability state checking in the Byzantine environment.

Protocols for identifying unusual transmissions were presented in [13] to discover malicious sensor nodes in a WSN by detecting malicious message transmission in a network. A message transmission is supposedly suspicious if its signal strength is incompatible with its originator's geographical location.

3 System model

We have considered a WSN under a non-interference environment with a fixed number of sensor nodes arranged in a star topology. The sensor nodes send their measured values to the fusion center

in a collision-free Time Division Multiple Access (TDMA) protocol. Sensor nodes were unable to detect the Media Access Control address (MAC address) of other sensor nodes in the network. Each sensor node operated independently. All normal sensor nodes had the same behavior giving accurate measurement values of 80%.

The reporting order is defined as the order in which sensor nodes send their local decisions to the fusion center. A sensor node with reporting order t will send its local decision after the sensor node with reporting order $t-1$ sends its local decision.

In this paper, we have focused on a specific type of sensor nodes; binary sensor nodes. The binary sensor node responds by either Present (H_1) or Absent (H_0).

3.1 Mathematical technique analysis

Let the space of n sensor nodes be defined by $S = \{s_i : 1 \leq i \leq n\}$. Each sensor node gives a local binary decision, denoted by l_n , in which

$$l_n = \begin{cases} 1, & \text{for target-present,} \\ 0, & \text{for target-absent.} \end{cases}$$

The topological pairs space is defined as all possible pairs of n sensor nodes, and one fusion center, such that $\mathcal{T} = \left(\left({}_{2\}\}, \text{Fucion Center}} \right) \right)$, in which P_k is a pair of sensor nodes, and n is an even number of sensor nodes.

If n is an odd integer then we choose a random sensor node, and uses its decision twice (e.g. if sensor node s_i was chosen then the first local decision is denoted by ${}_1l_i$, and the second local decision is denoted by ${}_2l_i$).

For a chosen round ν, we define τ_ν as a collection of distinct random pairs (i.e $\tau_\nu = \{P_k = \{s_i,\, s_j\} : i \neq j \text{ and } \; i,\, j \in \{1, \ldots, n\}\}$ in which $\tau_\nu \subseteq S \times S$ and $\|\tau_\nu\| = n/2$, in which $\|.\|$ is the cardinality (number of elements)). Also, we define the local decision of the k^{th} pair P_k that consists of the i^{th} and j^{th} sensor nodes at a certain round ν as $l_\nu^k : \tau_\nu \longmapsto \{0,\, 1,\, \phi\}$, in which

$$l_\nu^k(P_k) = l_\nu^k(s_i,\, s_j) = \begin{cases} l_i, & \text{If } l_i \bigoplus l_j = 0 \\ \phi(\text{Ignored}), & \text{If } l_i \bigoplus l_j = 1 \end{cases}$$

in which $\bigoplus$ is the bitwise XOR operation, $P_k = \{s_i,\, s_j\}$, $i,\, j \in \{1, \ldots, n\}$, and $l_\nu^k(s_i,\, s_j) = l_\nu^k(s_j,\, s_i)$ since bitwise the XOR operation is commutative.

The set of local decisions made by τ_ν (all pairs of sensor nodes at a certain round ν) is denoted by $\mathcal{L}_\nu = \{l_\nu^k : l_\nu^k \neq \phi,\, k = 1, \ldots, n/2, \text{ and } i \neq j \in \{1, \ldots, n\}\}$ and $m = \|\mathcal{L}_\nu\| \leq n/2$. The ignored pairs are defined as silent pairs with cardinality $\|Silent\, Pairs\| = n/2 - m$.

The collection of all possible sets of sensor node pairs that are randomly chosen is denoted by $\mathbf{L} = \{\mathcal{L}_\nu, \nu = 1, \ldots, \Pi_{t=1}^{n/2}(2t-1)\}$, in which the cardinality of $\mathbf{L}$ represents the number of distinct random rounds the fusion center can runs. The cardinality of $\mathbf{L}$ can be calculated by multiplying the number of choices $(n-1)$ for the first pair by the number of choices $(n-3)$ for the second pair, and so on. More formally, $\|\mathbf{L}\|$ is determined by the product of all positive odd numbers less than n.

The local performance of each pair P_k is estimated by two factors, which are the local detection (P_{d_k}) probability and the false-alarm (P_{f_k}) probability. P_{d_k} is the probability that the target is correctly identified as present by a given pair k when it is actually present (i.e., $P_{d_k} = P_r\{l_\nu^k = 1 | H_1\}$), while P_{f_k} is the probability that the target is falsely identified as present by the pair when it is actually absent (i.e., $P_{f_k} = P_r\{l_\nu^k = 1 | H_0\}$).

The definition of the global detection (P_D) probability and the false-alarm (P_F) probability for m non-silent pairs, is defined as:

$$P_D = \sum_{r=\kappa}^{m} \sum_{t=1}^{\binom{m}{r}} \prod_{k \in B_t^{(m,\,r)}} P_{d_k} \prod_{k \notin B_t^{(m,\,r)}} (1 - P_{d_k}) \tag{1}$$

and

$$P_F = \sum_{r=\kappa}^{m} \sum_{t=1}^{\binom{m}{r}} \prod_{k \in B_t^{(m,r)}} P_{f_k} \prod_{k \notin B_t^{(m,r)}} (1 - P_{f_k}) \tag{2}$$

in which $B_1^{(m,r)}, B_2^{(m,r)}, \ldots, B_{\binom{m}{r}}^{(m,r)}$represent all the possible $\binom{m}{r}$ combinations of r integers drawn from the interval [1, m]. In order to estimate the reliability of the network, both (P_D) and (P_F) probabilities are combined together to calculate the correct global decision (P_{CD}) probability, given as:

$$P_{CD} = P_{H_1} P_D + P_{H_0}(1 - P_F), \tag{3}$$

in which P_{H_1} is the probability that the target is present, and P_{H_0} is the probability that the target is absent.

3.2 Fusion center strategy

The fusion center considers a sensor node as a present sensor node if its local decisions is present, and as an absent sensor node if its local decision is absent.

The Fusion center uses the K-out-of-n rule. This rule determines that the global decision is present if the number of present sensor nodes is not less than a certain threshold, denoted by K. Based on the K-out-of-n rule, the fusion center determines that the target is present if $\sum_{k=1}^{n} l_k \geq K$, and the target is absent if $\sum_{k=1}^{n} l_k < K$.

3.3 Malicious sensor node behavior

In this paper, we have focused on the dependent malicious sensor nodes. Dependent malicious sensor nodes behave based on local decisions of other sensor nodes in the network. These sensor nodes can take advanced and intelligent decisions based on network behavior. If the malicious sensor nodes are unable to affect the fusion center decision, they will act normally in order to avoid exposure.

As assumed, the malicious sensor nodes recognize the value of K, but are unable to figure the reporting order of other sensor nodes in the network. Moreover, they are unable to distinguish between malicious and normal sensor nodes.

The behavior of a malicious sensor node depends on its reporting order. A malicious sensor node with reporting order t has the ability to attend to only local decisions of the previous $t-1$ sensor nodes and save these decisions. Suppose Γ_t is the sum of the first $t-1$ local decisions (i.e. $\Gamma_t = \Sigma_{i=1}^{t-1} l_i$). The local decision of a dependent malicious sensor node can be expressed as in Eq. (4), in which $\mathcal{D}_O$, $\mathcal{D}_Z$ and $\mathcal{D}_F$ refer to the subsets of dependent malicious sensor nodes with always-one, always-zero, and always-false strategies, respectively. Also, $\overline{H}$ means that the dependent malicious sensor node will choose to act maliciously and send the opposite of its real local decision.

$$l_k = \begin{cases} \begin{cases} 1, & \text{If } k \in \mathcal{D}_O, \\ 0, & \text{If } k \in \mathcal{D}_Z, \\ \overline{H}, & \text{If } k \in \mathcal{D}_F, \end{cases} & \text{if } K - (n - t + 1) \leq \Gamma_t < K \\ H_1, & \text{if } \Gamma_t \geq K \\ H_0, & \text{if } \Gamma_t < K - (n - t + 1) \end{cases} \tag{4}$$

The dependent malicious sensor node chooses to act normally and report a correct local decision as the following two cases:

- if $\Gamma_t \geq K$: In this case, the number of reported ones in the first $t-1$ local decisions is not less than K. So, regardless of the local decisions of the remaining sensor nodes the global decision will be present. Therefore, the malicious sensor node will send H_1 as a local decision.

- if $\Gamma_t < K-(n-t+1)$: In this case, the number of reported ones in the first $t-1$ local decisions is so low that, even if all the remaining sensor nodes report ones the threshold K is unreachable and the target will be detected as absent. Therefore, the malicious sensor node will send H_0 as a local decision.

Otherwise, the malicious sensor node will choose to act maliciously because its still has a chance to influence and amend the global decision (i.e. when $K-(n-t+1) \leq \Gamma_t < K$).

4 Proposed methods

In this section, we introduce the Fail Silent Pair (FSP) algorithm, detection and prevention phases. The fusion center uses FSP algorithm to calculate the global decision. While the detection and prevention phases are used to detect a subset of the malicious sensor nodes and prevent them from influencing the global decision. Both phases require only one iteration with $\mathcal{O}(n)$ execution time. The difference is that the fusion center runs the prevention phase at each round after collecting local decisions from sensor nodes, while the detection phase runs after every R rounds. Both phases require additional $\mathcal{O}(n)$ memory.

Initially, the fusion center distributes the reporting order of each sensor node in the setup phase. Upon each round, the fusion center collects local decisions from all sensor nodes in the network. Subsequently, the fusion center uses the prevention method to reverse the local decision of each sensor node identified as malicious with a known strategy if it acts maliciously during this round. Afterwards, the fusion center calculates the global decision using the Fail Silent Pair (FSP) algorithm. According to each R rounds, the fusion center runs the detection phase to detect a subset of malicious sensor nodes. The detection method depends on deceiving the malicious sensor nodes in an unrealistic environment using sequences of fake packets. Depending on the reporting order, some malicious sensor nodes respond with different local decisions for each sequence of fake packets, thus leading to detection.

4.1 Fail Silent Pair (FSP) algorithm

The fusion center strategy is based on the Fail Silent Pair (FSP) algorithm. The fail silence method is discussed in [22]. A component using the fail silence method either provides a correct result or none at all. The FSP algorithm uses the same approach to handle the local decisions of sensor nodes. All sensor nodes send their local decisions to the fusion center. When the fusion center collects all local decisions, it randomly distributes all sensor nodes into pairs. Then, initializes two counters; one for present pairs and another for absent pairs. Finally, the fusion center runs the FSP algorithm to calculate the global decision. This process is done in each round.

The FSP algorithm takes into consideration the pair P_k if both sensor nodes of the pair have the same local decision. In case the local decisions of both sensor nodes at P_k are present, the fusion center increases the number of present pairs. In case the local decisions of both sensor nodes at P_k are absent, the fusion center increases the number of absent pairs. Otherwise, the fusion center ignores the pair P_k.

In our technique we consider κ to be a dynamic threshold computed as $\kappa = \lceil \frac{K}{n} \times m \rceil$, in which $m = \|\mathcal{L}_\nu\|$. κ value is variable since the value of m can be variable at each round ν. The FSP algorithm uses $\kappa-$out$-$of$-n$ to calculate the global decision. This rule determines that the global detection is present if the number of present pairs is larger than κ. The mathematical formula of the $\kappa-$out$-$of$-n$ is represented by (5), in which $l_\nu^k \in \mathcal{L}_\nu$.

$$G = \begin{cases} 1 \equiv \text{target-present}, & \text{If } \sum_{k=1}^{m} l_\nu^k \geq \kappa, \\ 0 \equiv \text{target-absent}, & \text{If } \sum_{k=1}^{m} l_\nu^k < \kappa, \end{cases} \tag{5}$$

Comparing the local decisions of two sensor nodes within the same pair reduces the effect of incorrect local decisions. Incorrect local decisions may be reported either by a malicious sensor node or by a normal sensor node giving accurate measurement values of 80%. Since we cannot determine which sensor node has an incorrect local decision, we exclude the pair. This is the principle of the

Algorithm 1 Pseudocode of Fail Silent Pair Algorithm

```
function FSP (N, K, LocalDecision)
    RandomPer ← PERMUTATION(1, N)
    AbsentPairs ← 0
    PresentPairs ← 0
    for i ← 1 to N / 2 do
        P1 ← LocalDecision[RandomPer[i]]
        P2 ← LocalDecision[RandomPer[i + 1]]
        if P1 + P2 = 2 then
            PresentPairs ← PresentPairs + 1
        else if P1 + P2 = 0 then
            AbsentPairs ← AbsentPairs + 1
        end if
    end for
    κ = ⌈((K / N) × (PresentPairs + AbsentPairs))⌉
    if PresentPairs ≥ κ then
        GlobalDecision ← 1
    else
        GlobalDecision ← 0
    end if
    return GlobalDecision
end function
```

Fail Silent Pair algorithm; either to consider the local decision of a pair if both sensor nodes in the pair agree on the same local decision, or to ignore the pair if there is a difference between the local decisions of the sensor nodes in the pair.

4.1.1 FSP algorithm - Example

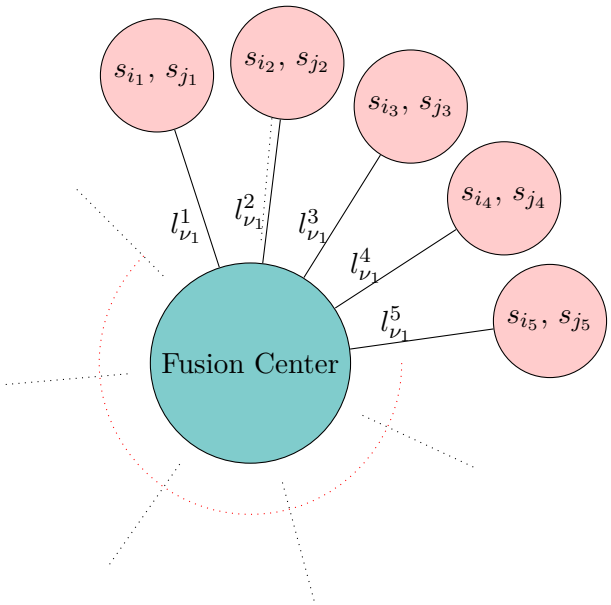


Figure 1: FSP algorithm

Table 1: Fusion center decision.

Sensor ID	i_1	j_1	i_2	j_2	i_3	j_3	i_4	j_4	i_5	j_5
Local Decision of the sensor node	1	0	0	0	1	1	0	0	1	0
Local Decision of the pair ($l^k_{\nu_1}$)	Drop		0		1		0		Drop	

To better understand the FSP algorithm, let us consider a fusion center with 8 sensor nodes as shown in Figure 1, in which the fusion center has the local decisions of each sensor node. Initially, the fusion center randomly distributes the sensor nodes into pairs then runs the FSP algorithm. The global decision of the fusion center is H_0 (absent), as the number of absent pairs is larger than the number of present pairs (in which $s_{i_1}, \ldots, s_{i_n}$ and $s_{j_1}, \ldots, s_{j_n}$ in Figure 1 are randomly chosen from the set of sensor nodes S. The pairwise counting technique is shown in table 1).

4.2 The detection phase

As mentioned in Section 3.3, a malicious sensor node with a reporting order t attends to the previous $t-1$ local decisions and saves them to determine if they act normally or maliciously. This means the reporting order plays an important role in the malicious sensor node decision. For that purpose, the fusion center uses the reporting order feature to detect the malicious sensor nodes.

As assumptions, the target will be under a certain situations (either present or absent) before starting the detection phase, and its situations will not change until the end of the phase.

The fusion center assumes that each sensor node in the network is a suspicious sensor node. A suspicious sensor node is a sensor node that may be a malicious sensor node, in which the fusion center must test its behavior to determine its type (either normal or malicious). In order to detect the malicious sensor nodes, the fusion center sends two sequences of fake packets to all suspicious sensor nodes and collects the local decisions of all sensor nodes after each sequence.

The detection technique is based on the dependent malicious sensor nodes strategy as described in 3.3. The aim of the algorithm is to trick a suspicious sensor node with two sequences of fake packets denoted by F_1 and F_2, respectively, and requests the local decision of the sensor node after each sequence. Building the sequences of fake packets depends on the reporting order of the suspicious sensor node. Assuming the existence of a suspicious sensor node with t as a reporting order, then F_1 will contains $t-1$ packets such that all of them are ones (i.e. $F_1 = (1, \cdots, 1)$), and F_2 will contains $t-1$ packets such that all of them are zeros (i.e. $F_2 = (0, \cdots, 0)$), in which in both sequences the i^{th} packets denotes the local decision of the i^{th} sensor node ($1 \leq i < t$).

Because of the nature of the sensor nodes (i.e. sensor nodes are low power consumption units), the malicious sensor nodes are unable to distinguish between normal and fake packets. So, a malicious sensor node will attend to fake packets and save them as normal packets sent from another sensor node in the network. This will create an unrealistic environment for the malicious sensor node, thus send its local decisions based on this environment. In contrast, a normal sensor node will not interact with fake packets, and send a correct local decision (of 80% accuracy).

If the suspicious sensor node is normal, its local decision will be identical for both sequences F_1 and F_2. On the other hand, if the suspicious sensor node is malicious, its strategy may force it to respond, in some cases, with two different local decisions, thus, be detected as a malicious sensor node.

Based on that, the fusion center can detect a subset of malicious sensor nodes, as follows:

- If $t > K$, the malicious sensor node will respond with H_1 for F_1 and with H_0 for F_2. Hence, be detected as malicious.
- If $t \leq K$ and $K - (n - t + 1) > 0$, then we have two cases, as follows:
  - If the strategy of the malicious sensor node is always-one, it will respond with H_1 for F_1 and with H_0 for F_2. Thus, be detected as malicious.
  - If the strategy of the malicious sensor node is always-false and the target is currently absent (i.e. $H = H_0$), it will respond with H_1 for F_1 and with H_0 for F_2. Thus, be detected as malicious.

Algorithm 2 Pseudocode of Detection Algorithm

```
function FakePackets (N)
    for i ← 1 to N do
        NodeState[i] ← Normal
    end for
    for i ← 1 to N do
        F1 ← GetOnes(i)
        Send F1
        Receive LocalDecision1
        F2 ← GetZeros(i)
        Send F2
        Receive LocalDecision2
        if LocalDecision1 ≠ LocalDecision2 then
            NodeState[i] ← Malicious_Unknown
        end if
    end for
    return NodeState
end function
```

When the fusion center detects a suspicious sensor node as malicious, it will not be able to identify its strategy. Therefore, the fusion center considers the strategy of new malicious sensor nodes as unknown at the end of the detection phase.

The fusion center does not run the detection phase at each round as there is a small percentage of change in the network topology between successive rounds. Instead, the fusion center runs the detection phase after every R rounds. The fusion center cannot guarantee the integrity of the network by running the detection phase once as some normal sensor nodes may be compromised with time. So, the detection phase must run periodically.

4.3 The prevention phase

As shown in Section 4.2, the fusion center can detect a subset of the malicious sensor nodes, assisting in the prevention of influencing the global decision. When a malicious sensor node has been detected, the fusion center cannot prevent it from sending its local decisions. As a substitute, the fusion center takes advantage of its knowledge of the behavior of malicious sensor nodes to correct their local decisions if possible and use the amended decisions alongside local decisions of normal sensor nodes to calculate the global decision using FSP algorithm.

The fusion center must identify the strategy of malicious sensor nodes to correct its local decision. The fusion center can only correct local decisions of malicious sensor nodes with always-false. Therefore, the fusion center ignores local decisions of other malicious sensor nodes with strategies but always-false.

The fusion center tracks the strategy of each malicious sensor node to handle its local decision (i.e. either to ignore or to correct it). Recall that when the fusion center detects a suspicious sensor node as a malicious, it will not be able to identify its strategy. Therefore, fusion center considers the strategy of new malicious sensor nodes as unknown at the end of the detection phase.

The fusion center runs the prevention phase at each round after collecting local decisions of all sensor nodes and before calculating the global decision. The prevention phase depends on the behavior of malicious sensor nodes, and the saved strategy and current local decision of each malicious sensor node. The prevention method is divided into 7 rules as shown in Table 2:

Table 2: Rules of prevention method.

Saved Strategy	Local Decision	New Strategy	New Local Decision
Unknown	Present	Always One	Present
Unknown	Absent	Always Zero	Absent
Always One	Present	Always One	Present
Always Zero	Absent	Always Zero	Absent
Always One	Absent	Always False	
Always Zero	Present	Always False	Reverse the local decision
Always False	Present or Absent	Always False	

Based on expression 4, the fusion center can determine if a malicious sensor node is acting normally or maliciously. So, The prevention method is only used with malicious sensor nodes whose has been identified as acting maliciously during this round.

When the prevention phase is over, the fusion center runs the FSP algorithm to calculate the global decision. The FSP algorithm will take into consideration normal sensor nodes, malicious sensor nodes acting normally, and malicious sensor nodes acting maliciously with always-false saved strategies during this round.

Algorithm 3 Pseudocode of Prevention Algorithm

```
function PreventionAlgorithm (N, K, LocalDecision, NodeState)
    NewLocalDecision ← LocalDecision
    for i ← 1 to N do
        Sum ← SumOfLocalDecision(LocalDecision, 1, i)
        if NodeState[i] ≠ Normal and Sum ≥ K − (N − i + 1) and Sum < K then
            if NodeState[i] = Malicious_Unknown then
                if LocalDecision[i] = 1 then
                    NodeState[i] ← Always_One
                else
                    NodeState[i] ← Always_Zero
                end if
            else if NodeState[i] = Always_One and LocalDecision[i] = 0 then
                NodeState[i] ← Always_False
            else if NodeState[i] = Always_Zero and LocalDecision[i] = 1 then
                NodeState[i] ← Always_False
            end if
            if NodeState[i] = Always_False then
                NewLocalDecision[i] ← 1 − LocalDecision[i]
            end if
        end if
    end for
    return NewLocalDecision
end function
```

5 Results

In this section, we present the simulation results to evaluate our methods. The simulation was done using Matlab R2017a platform and Ubuntu 16.04 LTS machine. We define the Success Rate as the ratio between the number of rounds in which the fusion center finds a correct global decision to the total number of rounds.

In this section, Percentage of H_1 is defined as the probability that the target is actually present in each round. This value is used to generate an assumed global decisions for each round before starting the simulation.

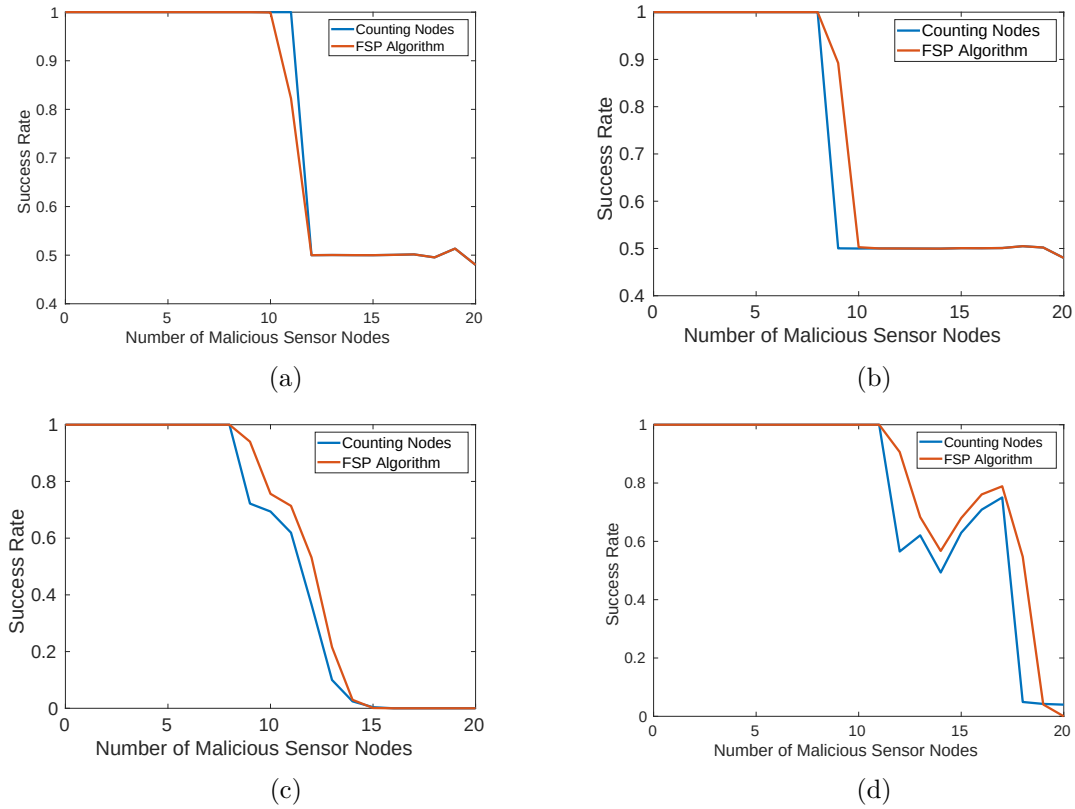


Figure 2: FSP algorithm versus counting sensor nodes technique.

5.1 Compare FSP algorithm with counting sensor nodes technique

We compare the performance of FSP algorithm against a previous technique such as the counting sensor nodes technique over a different number of malicious sensor nodes. The FSP algorithm was introduced in Section 4.1. While counting sensor nodes was explained in Section 3.2. Figure 2 shows the results of the comparison in different parameters. Table 3 shows the used parameters for each experiment, and the Improvement Rate. The Improvement Rate is defined as the ratio between the number of cases in which FSP algorithm is better than the counting sensor node technique to the total number of cases.

Table 3: Used parameters for each experiment in Figure 2

Figure	Rounds	n	K	Malicious Sensor Nodes			Percentage of H_1	Improvement Rate
				$\mathcal{D}_O$	$\mathcal{D}_Z$	$\mathcal{D}_F$		
2a	100	20	$0.6 \times n$	100%	0	0	50%	90.48%
2b	100	20	$0.6 \times n$	0	100%	0	50%	100%
2c	100	20	$0.6 \times n$	0	0	100%	95%	95.24%
2d	100	20	$0.6 \times n$	30%	30%	40%	95%	90.48%

The results show that the FSP algorithm performs better than the counting sensor nodes technique in different scenarios and with at least 90.00% improvement rate.

5.2 Detection and prevention

We compare the FSP algorithm with detection and prevention methods to the FSP algorithm without detection and prevention against the counting sensor node technique. Figure 3 results of the comparison in different parameters. Table 4 shows the used parameters for each experiment, and the Improvement Rate. The Improvement Rate is defined as the ratio between the number of cases in

which the FSP algorithm with detection and prevention is better than the FSP algorithm without detection and prevention to the total number of cases.

Table 4: Used parameters for each experiment in Figure 3

Figure	Rounds	n	K	Malicious Sensor Nodes			Percentage of H_1	Improvement Rate
				$\mathcal{D}_O$	$\mathcal{D}_Z$	$\mathcal{D}_F$		
3a	100	20	$0.6 \times n$	100%	0	0	50%	100%
3b	100	20	$0.6 \times n$	0	100%	0	50%	100%
3c	100	20	$0.6 \times n$	0	0	100%	95.00%	100%
3d	100	20	$0.6 \times n$	30.00%	30.00%	40.00%	95.00%	90.48%

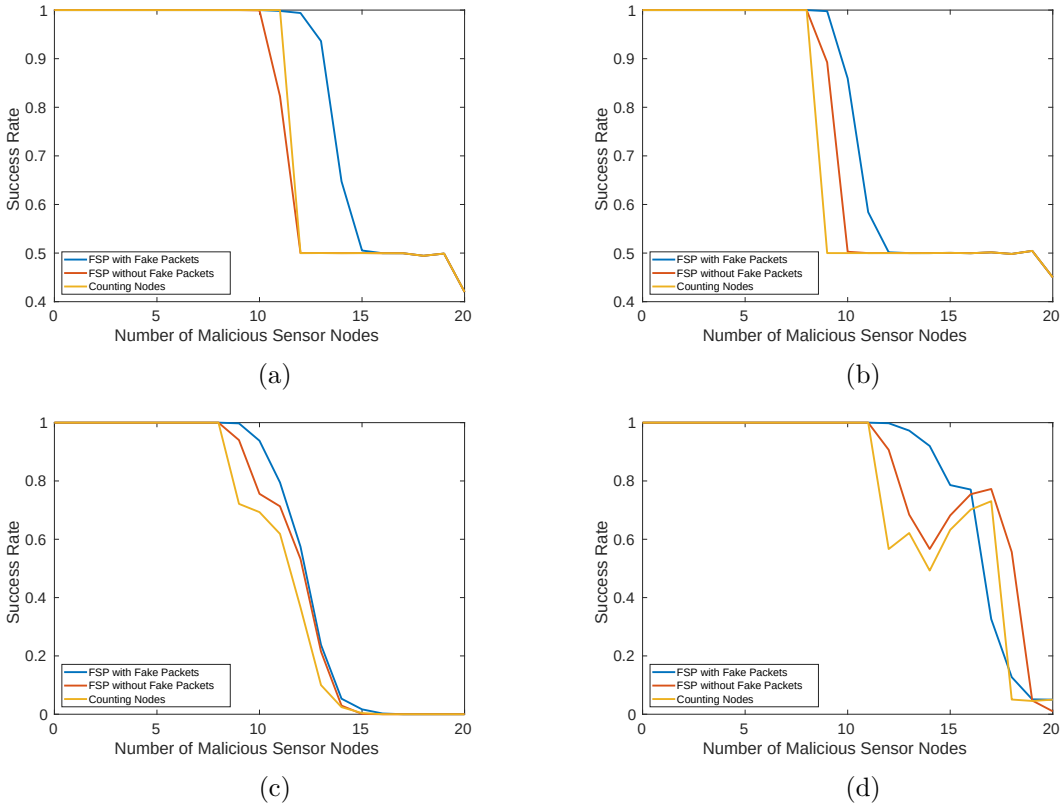


Figure 3: Compare the performance of FSP algorithm with and without detection and prevention.

The results show that using the detection and prevention methods increase the performance of the FSP algorithm with at least 90% improvement rate.

5.3 $K-$ thresholds values

The fusion center uses K value to calculate κ that is used to determine if the target is either present or absent. Choosing K value will significantly affect the performance of the fusion center. Figure 4 shows the performance of the fusion center under different values of K and different numbers of malicious sensor nodes. Table 5 shows the used parameters for experiments in Figure 4.

Table 5: Used parameters for experiments in Figure 4.

Figure	Rounds	n	Malicious Sensor Nodes			Percentage of H_1
			$\mathcal{D}_O$	$\mathcal{D}_Z$	$\mathcal{D}_F$	
4	100	16	0	0	100%	50%

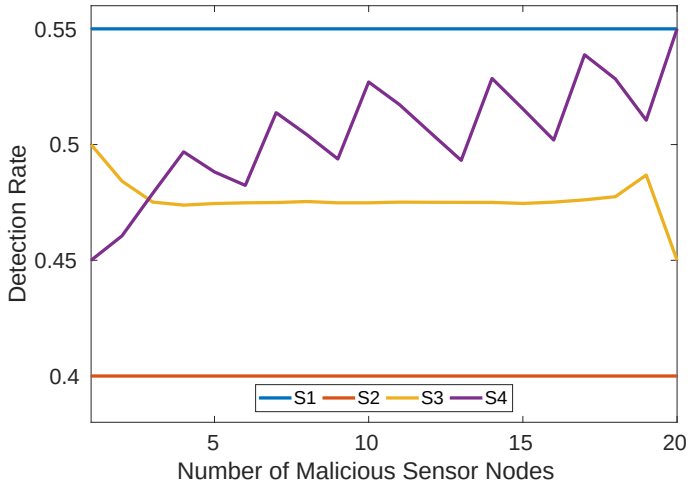


Figure 5: Detection rate over different number of malicious sensor nodes.

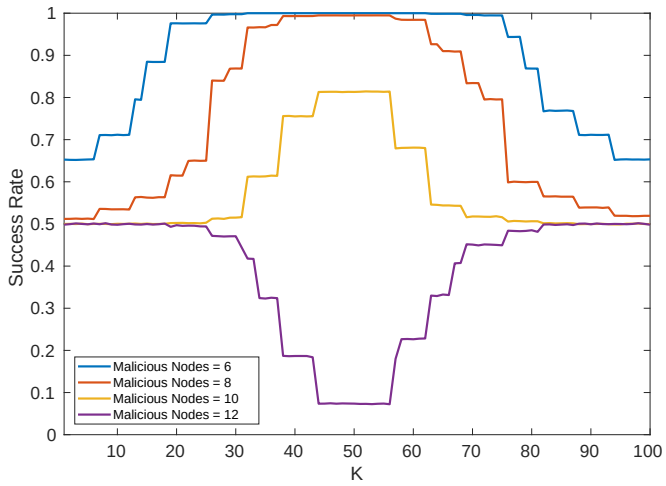


Figure 4: Performance of the fusion center with different values of K.

5.4 Fake packets detection technique

The detection technique using fake packets was introduced in Section 4.2. We have evaluated the performance of this technique in Figure 5. Table 6 shows the used parameters for each experiment in Figure 5. In Figure 5, the Detection Rate is defined as the ratio between the number of malicious sensor nodes identified as malicious to the total number of malicious sensor nodes.

Table 6: Used parameters for experiments in Figure 5.

Scenario	Rounds	n	K	Malicious Sensor Nodes			Percentage of H_1
				$\mathcal{D}_O$	$\mathcal{D}_Z$	$\mathcal{D}_F$	
S1	100	20	$0.6 \times n$	100%	0	0	50%
S2	100	20	$0.6 \times n$	0	100%	0	50%
S3	100	20	$0.6 \times n$	0	0	100%	50%
S4	100	20	$0.6 \times n$	30%	30%	40%	50%

The results show that using the fake packets technique assists the fusion center in reducing the effect of malicious sensor nodes through its ability to detect some.

After detecting a sensor node as malicious, the fusion center must identify its strategy in order to correct its local decision. Figure 6 studies the fusion center's ability identify strategies of malicious

sensor nodes. Table 7 shows the used parameters for experiment in Figure 6. In Figure 6, the Detection Rate is defined as the ratio between the number of malicious sensor nodes with correct identified strategies to the total number of malicious sensor nodes of that strategy.

Table 7: Used parameters for experiment in Figure 6.

Rounds	n	K	Malicious Sensor Nodes			Percentage of H_1
			$\mathcal{D}_O$	$\mathcal{D}_Z$	$\mathcal{D}_F$	
100	20	$0.60 \times n$	30.00%	30.00%	40.00%	50.00%

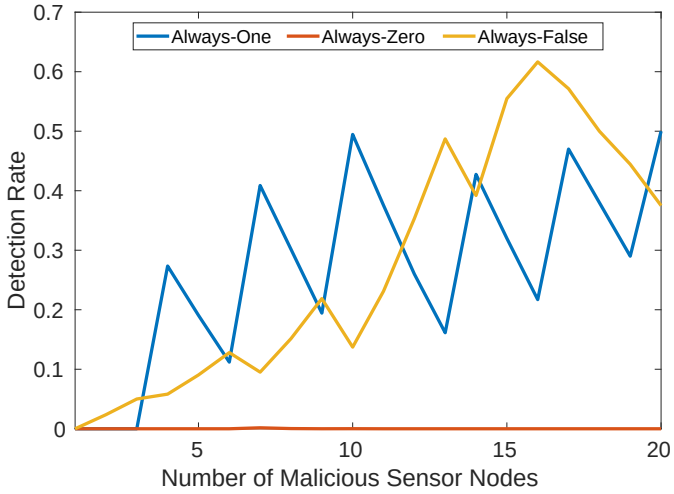


Figure 6: Identifying strategies of malicious sensor nodes.

The results show that the fusion center is able to identify strategies of always-one and always-false malicious sensor nodes with various detection rates. Moreover, the results show that malicious sensor nodes with always-zero strategy are able to withstand the fake packets technique.

6 Discussion

In this paper, we have introduced a new technique to improve the performance of target detection in Wireless Sensor Networks (WSNs). Our technique is divided into three phases: the Fail Silent Pair (FSP) algorithm, the detection phase based on the fake packets technique, and the prevention phase.

The FSP algorithm was introduced in Section 4.1. The FSP algorithm randomly distributes the sensor nodes into pairs. Then it takes into consideration the pair P_k if both sensor nodes in the pair have the same local decision. Otherwise, the FSP algorithm ignores the pair P_k. Section 5.1 shows that the FSP improves the accuracy of the global decision. However, we think the FSP algorithm will have a larger effect when utilized with other types of sensors such as sensors with the ability to measure distance or determine location.

The detection technique was introduced in Section 4.2. The detection technique is based on misleading malicious sensor nodes with an unrealistic environment using sequences of fake packets. Section 5.4 showed that this technique assists the fusion center to reduce the effect of malicious sensor nodes through its ability to detect some.

The prevention technique was introduced in Section 4.3. The fusion center uses the prevention technique to correct the local decisions of malicious sensor nodes of always-false strategy. Section 5.4 fusion center's ability to identify the strategies of malicious sensor nodes. The fusion center is able to identify strategies of always-one and always-false malicious sensor nodes with various detection rates. In contrast, malicious sensor nodes with always-zero strategies can be detected as malicious as shown in Figure 5, but the fusion center is incapable to identify their strategy as shown in Figure 6.

The introduced methods can enhance the fusion center's performance and the accuracy of the global decision. However, more improvements can be made. Future work will focus on using other types of sensor nodes, enhancing the detection and prevention methods to increase the detection rate, optimizing the fake packets technique to reduce the overhead of the fusion center, and using artificial intelligence to optimize the process of choosing the system's parameters (e.g. K, R, etc.).

7 Conclusions

In this paper, we have presented a new and simple technique to improve the performance of target detection in Wireless Sensor Networks (WSNs). We introduced the Fail Silent Pair (FSP) algorithm to calculate the global decision in the fusion center. The FSP algorithm randomly distributes all sensor nodes into pairs then only consider pairs in which both sensor nodes have the same local decision. This technique can reduce the effect of incorrect local decisions. Also, we introduced a new technique to improve the detection of dependent malicious sensor nodes in the network leading to the improvement of the accuracy of the global decision. The detection technique depends on misleading the malicious sensor nodes with an unrealistic environment using sequences of fake packets. A malicious sensor node will respond with different local decisions for each sequence of fake packets, leading to its detection. Finally, we introduced a new prevention technique to handle the local decisions of the malicious sensor nodes. This technique is based on prior knowledge of the behavior of malicious sensor nodes. The fusion center runs the prevention technique before calculating the global decision to reverse local decisions of malicious sensor nodes of identified strategies. This assists in improving the accuracy of the global decision.

Funding

This work is supported by King Abdullah II Design and Development Bureau (KADDB).

Author contributions

The authors contributed equally to this work.

Conflict of interest

The authors declare no conflict of interest.

References

[1] Althunibat, S.; Antonopoulos, A.; Kartsakli, E.; Granelli, F.; Verikoukis, C. (2016). Countering intelligent-dependent malicious nodes in target detection wireless sensor networks. *IEEE Sensors Journal*, 16(23), 8627–8639, 2016.

[2] Antonopoulos, A.; Verikoukis, C. (2016). Misbehavior detection in the internet of things: A network-coding-aware statistical approach. In *Industrial Informatics (INDIN), 2016 IEEE 14th International Conference on*, 1024–1027. IEEE, 2016.

[3] Anwar, R. W.; Bakhtiari, M.; Zainal, A.; Abdullah, A. H.; Qureshi, K. N.; Computing, F.; Bahru, J. (2014). Security issues and attacks in wireless sensor network. *World Applied Sciences Journal*, 30(10):1224–1227, 2014.

[4] Buratti, C.; Conti, A.; Dardari, D.; Verdone, R. (2009). An overview on wireless sensor networks technology and evolution. *Sensors*, 9(9), 6869–6896, 2009.

[5] Curiac, D.-I.; Banias, O.; Dragan, F.; Volosencu, C.; Dranga, O. (2007). Malicious node detection in wireless sensor networks using an autoregression technique. In *Networking and Services, 2007. ICNS. Third International Conference on*, IEEE, 83–83, 2007.

[6] Dâmaso, A.; Freitas, D.; Rosa, N.; Silva, B.; Maciel, P. (2013). Evaluating the power consumption of wireless sensor network applications using models. *Sensors*, 13(3), 3473–3500, 2013.

[7] Demirbas, M. (2005). *Wireless sensor networks for monitoring of large public buildings*, 2005.

[8] Di Pietro, R.; Mancini, L. V.; Soriente, C.; Spognardi, A.; Tsudik, G. (2008). Catch me (if you can): Data survival in unattended sensor networks. In *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on*, IEEE, 185–194, 2008.

[9] Hiregoudar, S.; Manjunath, K. (2017). Effective malicious node detection and data fusion under byzantine attacks, 2017.

[10] Kim, T.; Kim, I. H.; Sun, Y.; Jin, Z. (2015). Physical layer and medium access control design in energy efficient sensor networks: An overview. *IEEE Transactions on Industrial Informatics*, 11(1), 2–15, 2015.

[11] Lara, R.; Benítez, D.; Caamaño, A.; Zennaro, M.; Rojo-Álvarez, J. L. (2015). On real-time performance evaluation of volcano-monitoring systems with wireless sensor networks. *IEEE Sensors Journal*, 15(6), 3514–3523, 2015.

[12] Li, J.; Andrew, L. L.; Foh, C. H.; Zukerman, M.; Chen, H.-H. (2009). Connectivity, coverage and placement in wireless sensor networks. *Sensors*, 9(10), 7664–7693, 2009.

[13] Pires, W.; de Paula Figueiredo, T. H.; Wong, H. C.; Loureiro, A. A. F. (2004). Malicious node detection in wireless sensor networks. In *Parallel and distributed processing symposium, 2004. Proceedings. 18th international*, page 24. IEEE, 2004.

[14] Plastoi, M.; Volosencu, C.; Banias, O.; Tudoroiu, R.; Curiac, D.-I.; Doboli, A. (2009). Integrated system for malicious node discovery and self-destruction in wireless sensor networks. *International Journal on Advances in Networks and Services Volume 2, Numbers 2&3, 2009.*

[15] Salahuddin, M. A. et al. (2015). Introduction to wireless sensor networks. In *Wireless sensor and mobile ad-hoc networks*, 3–32. Springer, 2015.

[16] Spachos, P.; Hatzinakos, D. (2016). Real-time indoor carbon dioxide monitoring through cognitive wireless sensor networks. *IEEE sensors journal*, 16(2), 506–514, 2016.

[17] Đurišić, M. P.; Tafa, Z.; Dimić, G.; Milutinović, V. (2012). A survey of military applications of wireless sensor networks. In *Embedded Computing (MECO), 2012 Mediterranean Conference on*, pages 196–199. IEEE, 2012.

[18] Wang, Y.; Attebury, G.; Ramamurthy, B. (2006). A survey of security issues in wireless sensor networks, 2006.

[19] Webster, J. G.; Eren, H. (2017). *Measurement, instrumentation, and sensors handbook: spatial, mechanical, thermal, and radiation measurement*. CRC press, 2017.

[20] Yu, Y.; Li, K.; Zhou, W.; Li, P. (2012). Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures. *Journal of Network and computer Applications*, 35(3), 867–880, 2012.

[21] Zhang, Y.; He, S.; Chen, J. (2016). Data gathering optimization by dynamic sensing and routing in rechargeable sensor networks. *IEEE/ACM Transactions on Networking*, 24(3), 1632–1646, 2016.

[22] Zurawski, R. (2005). *Embedded systems handbook*. CRC press, 2005.

Journal's webpage: http://univagora.ro/jour/index.php/ijccc/

This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).
https://publicationethics.org/members/international-journal-computers-communications-and-control

Cite this paper as:
Ahmed, A.; Hababeh, M.; Abu-Hantash, A.; AbuHour, Y.; Musleh, H. (2020). Reduce Effect of Dependent Malicious Sensor Nodes in WSNs using Pairs Counting and Fake Packets, *International Journal of Computers Communications & Control*, 15(6), 3825, 2020. https://doi.org/10.15837/ijccc.2020.5.3825