# Countering Intelligent Dependent Malicious Nodes in Target Detection Wireless Sensor Networks

Saud Althunibat*, *Member, IEEE,* Angelos Antonopoulos†, *Senior Member, IEEE,* Elli Kartsakli‡, *Senior Member, IEEE,* Fabrizio Granelli§, *Senior Member, IEEE,* Christos Verikoukis†, *Senior Member, IEEE,*
*Department of Communications Engineering, Al-Hussein Bin Talal University (AHU), Ma'an, Jordan
†Telecommunications Technological Centre of Catalonia (CTTC), Barcelona, Spain
‡IQUADRAT Informatica S.L., Barcelona, Spain
§Department of Information Engineering and Computer Science, University of Trento, Trento, Italy
E–Mail: saud.althunibat@ahu.edu.jo, {aantonopoulos, cveri}@cttc.es, ellik@iquadrat.com, granelli@disi.unitn.it

***Abstract*—Target detection Wireless Sensor Networks (WSNs), where binary decisions are transmitted to declare the presence or absence of a given target, are expected to have a fundamental role in the Internet of Things (IoT) era. However, their simplicity makes these networks very susceptible to malicious attacks, while the problem is aggravated in the presence of intelligent malicious nodes that adapt their strategy depending on the behavior of other nodes in the network. In this paper, first, we analytically demonstrate that dependent and independent malicious nodes have the same impact on the overall performance of target detection WSNs in terms of detection and false alarm rates. Then, taking into account that dependent malicious users cannot be detected by conventional algorithms, we introduce an effective algorithm that detects malicious nodes in the network regardless of their type and number. Finally, theoretical and simulation results are provided to show the effects of dependent malicious nodes and analyze the performance of the proposed algorithm compared to existing state-of-the-art works.**

## I. Introduction

We are moving towards a new Internet of Things (IoT) era that is characterized by the interconnection among physical objects and enabled by sensors embedded in these objects. This paradigm shift has caused a significant increase in the size and number of wireless sensor networks (WSNs), creating an expectation of 50 billion connected devices by 2020 [1], [2]. However, the unparalleled proliferation of wireless networks implies a huge volume of data traffic and, therefore, stresses the need for data-light solutions. To that end, target detection WSNs are consistently gaining ground in an effort to disseminate the information without overloading the network.

Target detection WSNs are widely applied in several fields, including agriculture, health care and transportation, among others [3]–[5]. Unlike sensors in conventional WSNs that typically transmit detailed measurements (e.g., temperature, humidity, oxygen levels), the sensors in target detection WSNs are triggered by the existence of a specific event, i.e., when measurements exceed a given threshold. Hence, as their main role lies in the monitoring and identification of a specific target, the sensors may transmit information to the fusion center in a binary form (zero or one) regarding the target status (absent or present), reducing the amount of transmitted data and the power consumption at sensor nodes [6].

### A. Motivation

The simplicity of target detection WSNs makes them ideal candidates for a wide range of IoT applications. Nonetheless, due to their low complexity and the broadcast nature of the wireless medium, these networks are more prone to security attacks, since heavy security protocols that induce huge overhead and consume valuable power resources cannot be employed [7], [8]. Malicious nodes constitute a major threat for the WSN performance [9], as they are insider attackers that passed all conventional authentication protocols and became legitimate members [10]. More specifically, malicious nodes usually generate false alarm or misdetect correct alarms (either intentionally or unintentionally) and provide this information to the fusion center. Consequently, as the fusion center centrally processes this information in order to issue a global decision [11], fake reports by malicious nodes may significantly affect the reliability of this decision [12], [13].

In light of the above discussion, the design of novel solutions that protect the network against any malicious behavior has become of utmost importance, motivating the research activity towards this direction [14]- [24]. The majority of these works aim at identifying the abnormal activity in the network, considering that malicious nodes follow a specific strategy and act independently of the normal nodes in the network. However, as malicious nodes evolve, they are able to adopt more sophisticated and intelligent strategies, adapting their behavior in the network [25][1]. More specifically, intelligent malicious nodes have the ability to adjust their reports according to the reports overheard by other nodes in the network. Therefore, in case that their decisions are not going

[1]This work is funded by CellFive (TEC2014-60130-P), IoSense (692480) and AGAUR (2014-SGR-1551).

[1]Please note that although [25] has been introduced for Cooperative Spectrum Sensing in cognitive radio networks, it is completely applicable in target detection WSNs.

to affect the global decision in the fusion center, they may act as honest nodes, complicating further their detection.

### B. Related Work

Independent malicious nodes have been widely investigated in the literature. One effective approach that does not require their detection is the optimization of the most common fusion rule (i.e., K-out-of-N rule). This approach has been adopted in [14], where the decision threshold is optimized to minimize the false alarm rate, having as a constraint the misdetection probability. Despite its good performance, the specific approach requires prior knowledge about the percentage of malicious nodes in the network and the local performance of all nodes. Another common detection approach is the construction of reputation tables [15], [16]. These schemes assume the presence of some nodes with a high security level, which are considered as an evaluation base. These witness nodes are in charge of identifying misbehaving nodes by monitoring their local performance. Alternatively, maximum likelihood estimation is applied to detect malicious nodes in [17], which requires prior information and induces extra complexity in the process.

The spatial correlation among neighbors has been also investigated as a means of detecting malicious nodes in [18], where the network is divided into clusters and each node monitors the behavior of all the neighbors within its cluster. Nodes within a cluster exchange information about their performance and, accordingly, a node is identified as malicious if the majority decides that its behavior significantly deviates from the rest of the neighbors. Similar cluster-based approaches have been also proposed in [19]–[22]. However, composing clusters, monitoring neighbors and exchanging data consume valuable time-energy resources and increase complexity and overhead in WSNs. In the same context, the identification of the malicious users that deviate from the normal performance can be performed centrally by the fusion center by employing histogram information [23] or kernel functions [24] about the data distribution.

The aforementioned works, despite their satisfactory performance against independent malicious nodes, cannot deal effectively with dependent malicious nodes that adapt their behavior according to the reports received by other nodes. To the best of our knowledge, the case of dependent malicious nodes has only been considered in [25], where the authors propose a detection algorithm based on the monitoring of the history of local node decisions. The core idea of the algorithm lies in counting the number of mismatches in pairwise comparisons during a large time window, as normal nodes are expected to have similar number of mismatches, while the mismatches of a malicious node should be substantially different. Although the algorithm copes efficiently with both dependent and independent malicious nodes, there are several assumptions and conditions that should be satisfied: *i*) there should be a single malicious node within the network, *ii*) normal nodes should have almost identical local performance, *iii*) malicious identification can be performed after a large time window, and *iv*) the selected thresholds to detect the abnormalities should be carefully optimized. These conditions and assumptions limit the application of the algorithm in more complex networks and stress the need for novel security solutions. Thus, as a conclusion, SoA algorithms have assumptions and limitations that do not hold in practical applications. Therefore, we will propose an algorithm that is able to overcome the limitations of the SoA algorithms and is based on practical assumptions and scenarios.

### C. Contribution

In this paper, motivated by the importance of security in WSNs, we introduce a novel detection scheme for malicious nodes in target detection WSNs. The operation of the proposed scheme is based on altering the reporting order of all nodes in the network on a periodic basis. More specifically, as the nodes communicate their local decisions to the fusion center consecutively [26], the performance of an intelligent node heavily depends on its reporting turn (e.g., the later the node transmits the report, the more information has collected by other reports). As a result, periodic changes in the reporting order are expected to have an impact on the behavior of malicious nodes, while normal nodes will remain unaffected.

The proposed scheme consists in three phases: *i*) *detection phase*, where the nodes classified as normal, dependent malicious and independent malicious, *ii*) *identification phase*, where the particular strategy of each malicious user is identified, and *iii*) *potential countermeasures* to alleviate the impact of malicious activity in the network. It is worth noting that the proposed scheme is able to detect dependent malicious nodes regardless of their number and strategies and withdraws the assumption of identical local performance for normal nodes, without inducing extra overhead nor increasing the complexity of the network. The contributions of our work can be summarized as follows:

- We analyze the performance of the WSNs in the presence of independent and dependent malicious nodes, providing a mathematical formulation for the local and global network performance.
- We prove that, although the intelligent behavior of dependent malicious users complicates their detection, it does not increase the performance degradation of the WSN compared to other types of malicious activity.
- We analytically formulate the relationship between the local performance (i.e., detection and false-alarm probabilities) and the reporting turn.
- We introduce a novel scheme that effectively detects and identifies multiple malicious nodes regardless of their type.

The rest of the paper is organized as follows. Section II presents the system model of the considered WSN. The different malicious node types along with their mathematical models are discussed in Section III. In Section IV, we analyze the performance of the WSN in the presence of malicious nodes, while, in Section V, we introduce the novel malicious-detection algorithm. Simulation results are provided and discussed in Section VI, and conclusions are drawn in Section VII.

## II. System Model

We consider a WSN with star topology, consisting of $N$ sensor nodes, i.e., $\mathcal{S} = \{ s_n : 1 \leq n < N\}$ and a fusion center. We assume that all nodes report to the fusion center in a collision-free Time Division Multiple Access (TDMA) manner, which is compatible with the IEEE 802.15.4 standard [27]. The fusion center plays the role of the network coordinator and transmits beacons to mark the beginning of each frame. Then, the node reports take place within the Guaranteed Time Slots[2] of the Contention Free Period of the frame, as shown in Fig. 1.

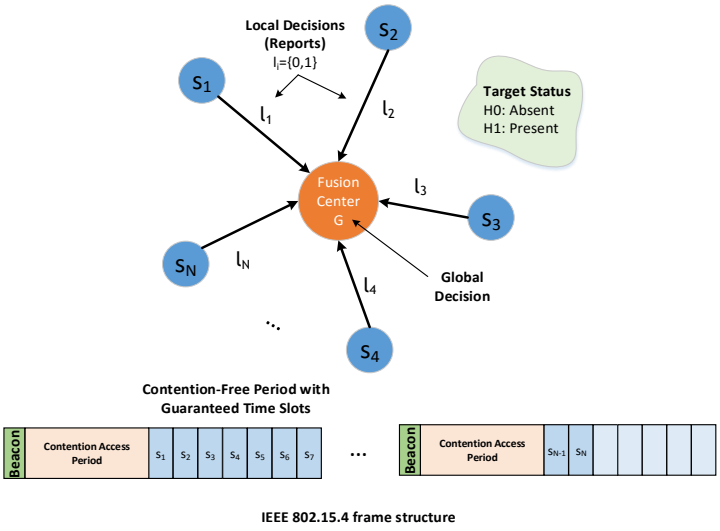


Fig. 1. An example of a typical target detection WSN

We focus on a target detection WSN where all nodes measure the binary status of a specific target, which can be either *Present* ($H_1$) or *Absent* ($H_0$). Each node processes its individual measurements in order to make a local binary decision, denoted by $l_n$, where

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

The set of local binary decisions made by all nodes is denoted by $\mathcal{L} = \{ l_n : 1 \leq n < N\}$.

The local performance of each node $s_n$ is assessed by the local detection ($P_{d_n}$) and false-alarm ($P_{f_n}$) probabilities. The former defines the probability that the target is correctly identified as present by a given node $n$ (i.e., $P_{d_n} = Pr\{l_n = 1|H_1\}$), while the latter is the probability that the target is falsely identified as present by the node, when it is actually absent (i.e., $P_{f_n} = Pr\{l_n = 1|H_0\}$).

Local decisions are reported to the fusion center, which employs a fusion rule in order to make a global decision $G$ regarding the target status. In general, target WSNs usually employ the *K-out-of-N* fusion rule, according to which the target is identified as present when the number of positive reports (indicating the presence of the target) exceeds a specific predefined threshold, denoted by $K$. The *K-out-of-N* fusion rule can be formulated mathematically as

$$G = \begin{cases} 1 \equiv \text{target-present}, & \text{if } \sum_{n=1}^{N} l_n \geq K \\ 0 \equiv \text{target-absent}, & \text{if } \sum_{n=1}^{N} l_n < K \end{cases} \tag{2}$$

where $G$ is the global decision.

Similar to the local decisions, we define the global detection ($P_D$) and false-alarm ($P_F$) probabilities, expressed, respectively, as

$$P_D = \sum_{i=K}^{N} \sum_{j=1}^{\binom{N}{i}} \prod_{n \in A_j^{(N,i)}} P_{d_n} \prod_{n \notin A_j^{(N,i)}} \left(1 - P_{d_n}\right) \tag{3}$$

and

$$P_F = \sum_{i=K}^{N} \sum_{j=1}^{\binom{N}{i}} \prod_{n \in A_j^{(N,i)}} P_{f_n} \prod_{n \notin A_j^{(N,i)}} \left(1 - P_{f_n}\right), \tag{4}$$

where $A_1^{(N,i)}, A_2^{(N,i)}, ..., A_{\binom{N}{i}}^{(N,i)}$ represent all the possible $\binom{N}{i}$ combinations of $i$ integers drawn from the interval $[1, N]$.

These two probabilities are combined to estimate the probability of correct global decision ($P_{CD}$), which is usually introduced as a comprehensive metric to describe the reliability of the network, given as

$$P_{CD} = P_{H_1} P_D + P_{H_0}(1 - P_F), \tag{5}$$

where $P_{H_1}$ is the probability that the target is actually present, and $P_{H_0}$ represents its complementary probability.

## III. Malicious Nodes Strategies

The reliability of the global decision depends on the accuracy of the local reports and can be significantly degraded by the presence of malicious nodes. Nodes are characterized as malicious if they systematically report an incorrect local decision to the fusion center. Even though in some cases malicious behavior may be unintentional (e.g., due to a hardware malfunction), malicious nodes usually provide false information intentionally, in order to manipulate the global decision, based on specific strategies. A malicious node may either adopt an independent strategy, unaffected by the local decision of other nodes, or may follow a more intelligent adaptive approach. The key strategies for independent and dependent malicious nodes will be described in Sections III-A and III-B, while a specific example of malicious behavior will be given in Section III-C.

### A. Independent Malicious Strategies

Independent malicious nodes determine their local decision without taking into consideration the behavior of the other nodes

[2]Note that the IEEE 802.15.4 standard specifies only 7 collision-free slots per frame. Hence, without loss of generality we consider that if $N > 7$, the measurement reports may be completed over successive frames.

in the network. There are three popular types of independent malicious strategies: *i*) *always-one*, where the malicious node always reports that the target is present [28], *ii*) *always-zero*, where the malicious node always reports that the target is absent [28], and *iii*) *always-false*, where the malicious node always report a false status (i.e., the opposite status than the one detected) [14], [29].

For the above strategies, the local decision $l_n$ can be expressed as

$$l_n = \begin{cases} 1\,, & n \in \mathcal{I}_{\mathcal{O}} \\ 0\,, & n \in \mathcal{I}_{\mathcal{Z}} \\ \overline{H}, & n \in \mathcal{I}_{\mathcal{F}} \end{cases} \tag{6}$$

where $\mathcal{I}_{\mathcal{O}}$, $\mathcal{I}_{\mathcal{Z}}$ and $\mathcal{I}_{\mathcal{F}}$ refer to the subsets of independent malicious nodes with strategy *always-one*, *always-zero* and *always-false*, respectively. Furthermore, $\overline{H}$ represents the false target-status.

### B. Dependent Malicious Strategies

The same strategies apply with some modifications in the case of dependent malicious nodes. Dependent nodes adapt their behavior based on the local decisions of the other nodes of the network, in order to reduce the likelihood of being detected. Hence, they act maliciously only when they can affect the global decision; otherwise they act honestly and report the actual target status. When the nodes report to the fusion center using TDMA (as in our scenario), dependent malicious nodes cannot listen to the local decisions of all others in order to determine their course of action. In particular, a dependent malicious node scheduled to report on the $t^{th}$ time slot must decide how to behave (i.e, honestly or maliciously) based on the previous $t-1$ local decisions. Thus, the behavior of the malicious node is significantly affected by its reporting turn.

Let us define $\Gamma_t$ as the sum of the first $t-1$ reported local decisions, i.e., $\Gamma_t = \sum_{n=1}^{t-1} l_n$. Accordingly, the local decision issued by a dependent malicious node that has the $t^{th}$ reporting turn can be expressed as

$$l_n = \begin{cases} \begin{cases} 1\,, & n \in \mathcal{D}_{\mathcal{O}}\,, \\ 0\,, & n \in \mathcal{D}_{\mathcal{Z}}\,, \\ \overline{H}\,, & n \in \mathcal{D}_{\mathcal{F}}\,, \end{cases} & \text{if } K-(N-t+1) \le \Gamma_t < K \\ \\ H\,, & \text{otherwise} \end{cases} \tag{7}$$

where $\mathcal{D}_{\mathcal{O}}$, $\mathcal{D}_{\mathcal{Z}}$ and $\mathcal{D}_{\mathcal{F}}$ refer to the subsets of dependent malicious nodes with *always-one*, *always-zero* and *always-false* strategies, respectively. As shown in (7), the behavior of a dependent malicious node, regardless of its strategy, depends on the value of $\Gamma_t$. Given the *K-out-of-N* fusion rule, the dependent malicious node chooses to act honestly and report a correct local decision ($H$) in the following two cases:

*i*) $\Gamma_t \ge K$: In this case, the number of reported '1's in the first $t-1$ slots exceeds (or is equal to) the detection threshold $K$. Hence, the target will be detected as present by the fusion center, regardless of the malicious node's local decision.

*ii*) $\Gamma_t < K-(N-t+1)$: In this case, the number of reported '1's in the first $t-1$ slots is so low that, even if all the remaining nodes (including the malicious one) report '1', the threshold cannot be reached and the target will be detected as absent.

On the other hand, if $K-(N-t+1) \le \Gamma_t < K$, the dependent malicious node will act maliciously since there is still a chance to affect the global decision.

### C. Malicious Behavior Example

For a better understanding of the dependent malicious nodes, let us consider a WSN consisting of $N=4$ nodes $\{s_1, s_2, s_3, s_4\}$ that report to the fusion center in that order. In this example, we assume that $s_3$ is a dependent *always-one* malicious node (i.e., $s_3 \in \mathcal{D}_{\mathcal{O}}$) and that the target is absent ($H_0$). The malicious node is scheduled at the third reporting turn ($t=3$), thus being able to hear from nodes $s_1$ and $s_2$ before making its own local decision. Table II lists all possible local decisions, as well as the corresponding global decisions, for two values of the detection threshold ($K=2$ and $K=3$).

For $K=2$, in Case 1, nodes $s_1$ and $s_2$ have reported the target absent ('0'). The malicious node selects to report '1', since in that case, it has the potential (depending on the report of $s_4$) to affect the outcome of the global decision, given that $K=2$ (i.e., the fusion center must receive at least two '1's to declare the target present). In Cases 2 and 3, the malicious node also reports '1', knowing with certainty that its contribution will affect the final decision, regardless of the action of $s_4$. Finally, in Case 4, the first two nodes have reported '1', meaning that the global decision will be '1', regardless of the reports of nodes $s_3$ and $s_4$. In that case, the malicious node chooses to act honestly and report the actual target status (i.e., $H_0$).

The same concepts apply for $K=3$, however, in that case, the fusion center must receive at least three '1's to consider the target present. Hence, the malicious node will act honestly only in Case 1, when there is no possibility of affecting the global decision. This happens because the first two nodes have reported the target absent and, even if both nodes, $s_3$ and $s_4$, report '1's, the number of 1's will never equal to the threshold $K=3$. In all the other cases, the malicious node will always report '1' as there is always a probability to obtain a global decision of '1'.

## IV. Performance Analysis in Presence of Malicious Nodes

In this section, we examine the importance of combating against the malicious nodes in WSNs, by exploring their impact the performance in terms of detection and false alarm probabilities. Specifically, in Section IV-A, we study the impact of dependent behavior on the local performance of malicious nodes, whereas in Section IV-B we show that dependency does not affect the overall performance of the network. Finally, in Section IV-C, we optimize the selection of the detection threshold $K$, in order

TABLE I
AN EXAMPLE FOR A WSN OF 3 NORMAL NODES AND A SINGLE DEPENDENT ALWAYS-ONE MALICIOUS NODE.

$H_0$, $K = 2$

| Node ID and Type | $s_1$ Normal | $s_2$ Normal | $s_3$ Malicious $\in \mathcal{D}_\mathcal{O}$ | $s_4$ Normal | FC Global Decision |
|---|---|---|---|---|---|
| Case 1 | 0 | 0 | 1 | 1/0 | 1/0 |
| Case 2 | 0 | 1 | 1 | 1/0 | 1/1 |
| Case 3 | 1 | 0 | 1 | 1/0 | 1/1 |
| Case 4 | 1 | 1 | 0 | 1/0 | 1/1 |

$H_0$, $K = 3$

| Node ID and Type | $s_1$ Normal | $s_2$ Normal | $s_3$ Malicious $\in \mathcal{D}_\mathcal{O}$ | $s_4$ Normal | FC Global Decision |
|---|---|---|---|---|---|
| Case 1 | 0 | 0 | 0 | 1/0 | 1/0 |
| Case 2 | 0 | 1 | 1 | 1/0 | 1/0 |
| Case 3 | 1 | 0 | 1 | 1/0 | 1/0 |
| Case 4 | 1 | 1 | 1 | 1/0 | 1/1 |

to minimize the impact of malicious behavior on the system's performance.

### A. Impact of Malicious Behavior on Local Performance

For the different types of malicious strategies described in the previous section, the local detection probability $P_{d_n}^{(t)}$ of a malicious node that has the $t^{th}$ reporting turn is given by

$$P_{d_n}^{(t)} = \begin{cases} 1\,, & n \in \mathcal{I}_\mathcal{O} \text{ or } \mathcal{D}_\mathcal{O} \\ 0\,, & n \in \mathcal{I}_\mathcal{Z} \text{ or } \mathcal{I}_\mathcal{F} \\ 1 - \sum_{i=\alpha}^{N'} \sum_{j=1}^{\binom{N'}{i}} \prod_{n \in B_j^{(N',i)}} P_{d_n} \prod_{n \notin B_j^{(N',i)}} \left(1 - P_{d_n}\right), & n \in \mathcal{D}_\mathcal{Z} \text{ or } \mathcal{D}_\mathcal{F} \end{cases}, \quad (8)$$

whereas the local false alarm probability $P_{f_n}^{(t)}$ can be expressed as

$$P_{f_n}^{(t)} = \begin{cases} 1\,, & n \in \mathcal{I}_\mathcal{O} \text{ or } \mathcal{I}_\mathcal{F} \\ 0\,, & n \in \mathcal{I}_\mathcal{Z} \text{ or } \mathcal{D}_\mathcal{Z} \\ \sum_{i=\alpha}^{N'} \sum_{j=1}^{\binom{N'}{i}} \prod_{n \in B_j^{(N',i)}} P_{f_n} \prod_{n \notin B_j^{(N',i)}} \left(1 - P_{f_n}\right), & n \in \mathcal{D}_\mathcal{O} \text{ or } \mathcal{D}_\mathcal{F} \end{cases}, \quad (9)$$

where $\alpha = \min\{K - (N - t + 1), 0\}$, $N' = \min\{K - 1, t - 1\}$ and $B_1^{(N',i)}, B_2^{(N',i)}, ..., B_{\binom{N'}{i}}^{(N',i)}$ represent all the possible $\binom{N'}{i}$ combinations of $i$ integers drawn from the interval $[1, N']$.

An initial observation that can be confirmed is that, for independent types ($\mathcal{I}_\mathcal{O}$, $\mathcal{I}_\mathcal{Z}$ and $\mathcal{I}_\mathcal{F}$), the detection and false alarm probabilities do not depend on $t$ (the reporting turn), since they do not listen to other nodes in the network. On the other hand, the reporting turn $t$ plays a significant role in the local performance of the dependent malicious nodes, i.e, $\mathcal{D}_\mathcal{O}$, $\mathcal{D}_\mathcal{Z}$ and $\mathcal{D}_\mathcal{F}$.

The above equations clearly describe the effect of the dependency of malicious nodes on their local performance. Conventional malicious detection algorithms can easily identify malicious nodes if they have very poor local performance compared to the rest of the nodes. The poor local performance can be represented by a high local false alarm probability (as in $\mathcal{I}_\mathcal{O}$), a low local detection probability (as in $\mathcal{I}_\mathcal{Z}$), or both of them (as in $\mathcal{I}_\mathcal{F}$). Thus, dependent malicious nodes aim to improve their local performance as much as possible, without losing the intention to mislead the global decision. From equations (8) and (9), we can clearly observe that dependent malicious nodes exhibit a better local performance with respect to independent nodes, as shown below.

*i*) For the *always-one* strategy, the false alarm probability is higher for independent malicious nodes ($\mathcal{I}_\mathcal{O} : P_{f_n}^{(t)} = 1$) with respect to dependent malicious nodes ($\mathcal{D}_\mathcal{O} : P_{f_n}^{(t)} < 1$).

*ii*) For the *always-zero* strategy, independent malicious nodes have a lower detection probability ($\mathcal{I}_\mathcal{Z} : P_{d_n}^{(t)} = 0$) with respect to dependent malicious nodes ($\mathcal{D}_\mathcal{Z} : P_{d_n}^{(t)} > 0$).

*iii*) For the *always-false* strategy, independent malicious nodes have a worse local performance in terms of both probabilities ($\mathcal{I}_\mathcal{F} : P_{d_n}^{(t)} = 0, P_{f_n}^{(t)} = 1$) with respect to dependent malicious nodes ($\mathcal{D}_\mathcal{F} : P_{d_n}^{(t)} > 0, P_{f_n}^{(t)} < 1$).

Focusing on the dependent malicious nodes, the reporting turn has an essential role in determining their local performance. Intuitively, as the reporting turn of a malicious node is delayed, the malicious node will have the opportunity to hear from more nodes, thus being able to make a better decision. The worst case for a dependent malicious node is to have the first reporting turn. In this case, it will not listen other reports and, consequently, it will act as an independent malicious node. On the other hand, having the last reporting turn constitutes the best case for a dependent malicious node, since it will be able to listen to all other nodes before sending its report.

### B. Impact of Malicious Behavior on Overall Performance

In the previous section, we examined the impact of malicious behavior on the local performance, showing that dependent strategies generally improve the performance of malicious nodes. Now, the question is: *does dependency of a malicious node on other nodes increase its influence on the overall performance of the WSN?* The direct answer is no, and this can be interpreted as follows.

According to (7), a dependent malicious node will act as a normal (honest) node in two cases, depending on the counter $\Gamma_t$: $i$) if $\Gamma_t \geq K$ or $ii$) if $\Gamma_t < K - (N - t + 1)$. Hence, the probability that a dependent malicious node behaves normally, denoted by $P_{DN}$, can be expressed as

$$P_{DN} = \Pr\{\Gamma_t \geq K\} + \Pr\{\Gamma_t + N - t + 1 < K\}. \quad (10)$$

Notice that, in the first case, the number of reported '1's already exceeds the threshold $K$ and, hence, the global decision is '1', regardless of the decisions made by the malicious node and the other nodes occupying the time slots after $t$. Similarly, in the second case, even if we assume that all the nodes occupying the time slots after $t-1$ report the target as present, the reported '1's will never exceed the threshold $K$. As a result, in both cases, the decision made by a dependent malicious node when behaving normally will never affect the reliability of the global decision.

On the other hand, the probability that a dependent malicious node behaves as an independent malicious node ($P_{DI}$) can be expressed as

$$P_{DI} = 1 - P_{DN} = \Pr\{K - (N - t + 1) \leq \Gamma_t < K\}. \quad (11)$$

The range of $K - (N - t + 1) \leq \Gamma_t < K$ implies that the number of received '1's at the fusion center by the time slot $t$ is not enough yet to make the global decision. Therefore, the local decision made by the malicious node will be effective. Thus, we can conclude that the influence of the malicious node will be the same whether it is dependent or independent as long as it follows the same strategy (i.e, *always-one*, *always-zero* or *always-false*).

It is worth mentioning that $P_D$ and $P_F$ in the presence of malicious nodes can be still computed by (3) and (4), respectively, regardless of the total number of malicious node or their specific strategies. However, the local performance of each node, i.e., $P_{d_n}$ and $P_{f_n}$ should be carefully substituted based on its type, as described in (8) and (9).

### C. Optimizing the detection threshold $K$

The detection threshold $K$ has a strong impact on the reliability of the global decision, but its optimal selection is not straightforward. For instance, high values of $K$ lead to a reduced overall false alarm probability $P_F$, which may improve the correct decision by the fusion center (see (5)). On the other hand, high $K$ values also cause a decrease of the overall detection probability $P_D$, which has a negative impact of performance. Similarly, low values of $K$ will again have a contrasting effect on the probability of a correct decision. Thus, the most appropriate selection of $K$ should be obtained through optimization in order to maximize the reliability of the global decision.

The optimization problem can be formulated as a maximization problem as

$$\max_K P_{CD}, \quad (12)$$

which can be rewritten by substituting the value of $P_{CD}$ from (5)

$$\max_K \Big( P_{H_1} P_D + P_{H_0}(1 - P_F) \Big), \quad (13)$$

which can be solved by setting the first derivative equal to 0, i.e., $\frac{\partial P_{CD}}{\partial K} = 0$, as

$$\frac{\partial P_{CD}}{\partial K} = P_{H_1} \frac{\partial P_D}{\partial K} - P_{H_0} \frac{\partial P_F}{\partial K} = 0. \quad (14)$$

As $K$ is an integer, $\frac{\partial P_D}{\partial K}$ can be expressed as

$$\frac{\partial P_D}{\partial K} = P_D(K + 1) - P_D(K), \quad (15)$$

which can be seamlessly obtained using (3) as

$$\frac{\partial P_D}{\partial K} = -\sum_{j=1}^{\binom{N}{K}} \prod_{n \in A_j^{(N,K)}} P_{d_n} \prod_{n \notin A_j^{(N,K)}} \big(1 - P_{d_n}\big). \quad (16)$$

Similarly, $\frac{\partial P_F}{\partial K}$ can be computed using (4) as

$$\frac{\partial P_F}{\partial K} = -\sum_{j=1}^{\binom{N}{K}} \prod_{n \in A_j^{(N,K)}} P_{f_n} \prod_{n \notin A_j^{(N,K)}} \big(1 - P_{f_n}\big). \quad (17)$$

Substituting (16) and (17) in (14), the optimal value of the detection threshold ($K^*$) that maximizes the probability of a correct global decision should satisfy the following equation

$$\sum_{j=1}^{\binom{N}{K}} \Bigg( - P_{H_1} \prod_{n \in A_j^{(N,K)}} P_{d_n} \prod_{n \notin A_j^{(N,K)}} \big(1 - P_{d_n}\big) + P_{H_0} \prod_{n \in A_j^{(N,K)}} P_{f_n} \prod_{n \notin A_j^{(N,K)}} \big(1 - P_{f_n}\big) \Bigg) = 0. \quad (18)$$

The optimal value of $K$ can be computed through an exhaustive search algorithm, as it cannot be formulated in a closed form expression, mainly due to the assumption of non-identical local performance among nodes.

## V. Proposed Malicious-Detection Scheme

As the local performance of intelligent dependent malicious nodes varies, their behavior cannot be combated by conventional schemes. In this section, we introduce an effective algorithm that is able to identify malicious nodes regardless of their type, without inducing any extra overhead, energy consumption or complexity. The proposed algorithm is performed in three consecutive phases: *i*) malicious node *detection*, *ii*) malicious strategy *identification*, and *iii*) *countermeasures*. The three phases of the proposed algorithm are explained in detail in the remaining part of this section.

### A. The Detection Phase

The main idea of the proposed scheme lies in the fact that the local performance of dependent malicious nodes depends on their reporting turn, unlike normal and independent malicious nodes that remain unaffected. Hence, allowing the fusion center to change the reporting turn of the nodes facilitates the detection of intelligent nodes with adaptive behavior, while independent malicious nodes can be also detected due to their poor performance. To that end, in our scheme, the reporting order in the network changes every $T$ rounds.

As the fusion center is not aware of the actual target status, we introduce the indicator $\sigma_n$ that counts the number of '1's received from each node $n$ to evaluate its local performance (i.e. detection and false alarm probabilities). This indicator is zero-initialized and updated at each reporting round $i$ as

$$\sigma_{n,i} = \begin{cases} \sigma_{n,i-1} + 1 & \text{, if } l_n = 1 \\ \sigma_{n,i-1} & \text{, otherwise} \end{cases}. \quad (19)$$

Since the reporting order changes every $T$ rounds, the probability $\hat{\rho}_n$ that node $n$ reports 1 can be estimated as

$$\hat{\rho}_n = \frac{\sigma_{n,T}}{T}, \tag{20}$$

and can be computed theoretically for a normal node as

$$\rho_n = P_{H_1} P_{d_n} + P_{H_0} P_{f_n}. \tag{21}$$

Notice that $\rho_n$ is the theoretical value, while $\hat{\rho}_n$ represents the estimated value.

By substituting in (21) the corresponding values of $P_{d_n}$ and $P_{f_n}$ ( (8) and (9), respectively), we may obtain the $\rho_n$ for the different strategies of independent and dependent malicious nodes as

$$\rho_n = \begin{cases} 1\,, & n \in \mathcal{I}_{\mathcal{O}} \\ 0\,, & n \in \mathcal{I}_{\mathcal{Z}} \\ P_{H_0}\,, & n \in \mathcal{I}_{\mathcal{F}} \\ P_{H_1} + P_{H_0} P_{f_n}^{(t)}\,, & n \in \mathcal{D}_{\mathcal{O}} \\ P_{H_1} P_{d_n}^{(t)}\,, & n \in \mathcal{D}_{\mathcal{Z}} \\ P_{H_1} P_{d_n}^{(t)} + P_{H_0} P_{f_n}^{(t)}\,, & n \in \mathcal{D}_{\mathcal{F}} \end{cases} \tag{22}$$

Apparently, only dependent malicious nodes have values of $\rho_n$ that are affected by the reporting turn. Such an initial observation leads to a seamless detection of dependent malicious nodes. Then, independent malicious nodes can be detected due to their very bad performance compared to other normal nodes, based on (22).

**Algorithm 1:** Pseudocode of DMND algorithm

```
Initialization: Define T, Δ, i = 1, I = {}, I_O = {},
I_Z = {}, I_F = {}, D_O = {}, D_Z = {}, D_F = {} ;
Start reporting ;
if i is a multiple of T then
    for n = 1 to N do
        estimate ρ̂_{n,i}
        if ρ̂_{n,i} > ρ̂_{n,i−T} + Δ or ρ̂_{n,i} < ρ̂_{n,i−T} − Δ then
            D = D + s_n
            run DMNI algorithm
        else
            run IMNI algorithm
        end
    end
    randomly change the reporting order;
end
i = i + 1;
Resume reporting;
```

The detection phase can be more concisely described through the pseudocode in Algorithm 1. The proposed algorithm, called Dependent Malicious Nodes Detection (DMND), separates the nodes in two subsets: *i*) dependent malicious nodes, and *ii*) normal nodes and independent malicious nodes. After each set of $T$ reporting rounds, the estimated $\hat{\rho}_n$ is compared to the corresponding previous value for each node. If the difference between them is larger than $\Delta$, the node is identified as dependent malicious node. The parameter $\Delta$ is added as an error margin in order to avoid a false identification of dependent malicious nodes, which may occur due to estimation errors. However, the value of $\Delta$ should not be large, in order to ensure the detection of dependent malicious nodes. The selection of $\Delta$ will be discussed in detail in Section V-D.

### *B. The Identification Phase*

After the completion of the detection phase, the dependent malicious nodes have been separated from the rest of the nodes (i.e., independent malicious or normal nodes). The second phase of the proposed algorithm includes the identification of the specific strategy of each malicious node. This is of paramount importance for the fusion center, enabling it to take the appropriate countermeasures to improve the global reliability (explained in Section V-C).

At this point, two identification algorithms are defined, for the classification of the independent and dependent malicious nodes, respectively.

*1) Independent Malicious Nodes Identification (IMNI) Algorithm:* Algorithm 2 contains the pseudocode of the IMNI algorithm, that aims to identify the specific strategy of independent malicious nodes. The key to determining the malicious strategy of each node lies in the value of $\rho_n$. As indicated in (22), an independent malicious node that belongs to $\mathcal{I}_{\mathcal{O}}$ should have $\rho_n = 1$, while an independent malicious node that belongs to $\mathcal{I}_{\mathcal{Z}}$ should have $\hat{\rho}_n = 0$. If $\hat{\rho}_n$ is equal to $P_{H_0}$ with an error margin $\Delta$, the corresponding node is added to $\mathcal{I}_{\mathcal{F}}$. Otherwise, the corresponding node is considered a normal node.

**Algorithm 2:** Pseudocode for IMNI algorithm

```
foreach s_n ∉ D do
    switch ρ̂_{n,T} do
        case ρ̂_{n,T} = 1
            I_O = I_O + s_n;
        case ρ̂_{n,T} = 0
            I_Z = I_Z + s_n;
        case P_{H_0} − Δ ≤ ρ̂_{n,T} ≤ P_{H_0} + Δ
            I_F = I_F + s_n;
        endsw
        otherwise
            s_n is normal;
        endsw
    endsw
end
```

*2) Dependent Malicious Nodes Identification (DMNI) Algorithm:* Algorithm 3 provides the pseudocode for the DMNI algorithm, aiming to specify the strategy of dependent malicious nodes. Recall that a dependent malicious node will act exactly as an independent malicious node if its reporting turn is the first. Thus, DMNI algorithm schedules each dependent malicious node

to the first reporting turn and checks the estimated $\hat{\rho}_n$. If it is equal to 1, the corresponding node is added to $\mathcal{D}_{\mathcal{O}}$. If it is equal to 0, the corresponding node is added to $\mathcal{D}_{\mathcal{Z}}$. Otherwise, for $0 < \hat{\rho}_n < 1$, the corresponding node is added to $\mathcal{D}_{\mathcal{F}}$.

**Algorithm 3:** Pseudo code of DMNI algorithm

```
foreach s_n ∈ D do
    set t = 1;
    if i is a multiple of T then
        switch ρ̂_{n,T} do
            case ρ̂_{n,T} = 1
                D_O = D_O + s_n;
            case ρ̂_{n,T} = 0
                D_Z = D_Z + s_n;
            endsw
            otherwise
                D_F = D_F + s_n;
            endsw
        endsw
    end
end
```

### C. The Countermeasures Phase

The intuitive action to be taken by the fusion center after identifying malicious nodes is to discard their reported decisions. However, by knowing the exact strategy of each node, it is possible to extract useful information on the status of the target, thus improving the reliability of the global decision.

In the case of independent nodes, reports from *always-one* and *always-zero* malicious nodes should be always ignored, since they are made regardless of the actual target status. However, reports from *always-false* malicious nodes are always opposite to the actual target status and can, thus, provide truthful information once inverted.

In the case of dependent nodes, it is important to detect when the nodes act honestly or maliciously. Clearly, when the malicious node acts honestly, the reported information is accurate and can be taken into account by the fusion center. On the other hand, when the dependent nodes act maliciously, the same principles described for independent nodes apply. In other words, reports from *always-false* nodes should be inverted, whereas reports from the other two strategies should be ignored.

As a further action, the fusion center can reorder the reporting turns of dependent malicious nodes in such a way so as to maximize the probability of being honest. Referring to (7) and (10), the probability that a dependent malicious node reports a honest decision increases for higher values of $t$. Thus, setting $t$ to the maximum value, i.e, $t = N$ , the probability that a dependent malicious node acts honestly will be maximized. In other words, a dependent malicious node scheduled to report on the last time slot will have a higher chance of behaving honestly, since its decision will take into account the reports of all the previous nodes.

Substituting $t = N$ in (11), the probability that a dependent malicious node acts maliciously decreases to the probability that $\Gamma_N = K - 1$. On the other hand, the probability that a dependent malicious node will be honest will be increased to be equal to the probability that $\Gamma_N \geq K$ or $\Gamma_N < K - 1$.

### D. Determining the parameters $T$ and $\Delta$

The two important parameters, $T$ and $\Delta$, have a significant influence on the performance of the proposed algorithms. In fact, $T$ should be large enough in order to better evaluate the performance of each node. However, very high values of $T$ may delay the detection and identification of the malicious nodes, which, in turn, prolongs their negative effects on the overall performance of the WSN. Similarly, $\Delta$ should be carefully selected, as low or large values of $\Delta$ may lead to misdetecting some malicious nodes or identifying normal nodes as malicious. It is worth highlighting here that both parameters are related to each other, since increasing the value of $T$ should decrease the sufficient value of $\Delta$ that maximizes the performance of the proposed scheme.

Notice that the estimation of $\rho_n$ can be represented as a mean estimation problem, where $T$ is the sample size and $\Delta$ is the margin error. Therefore, the relation between $\Delta$ and $T$ can be expressed as [30]

$$\Delta = t_{\zeta/2} \frac{\gamma}{\sqrt{T}}, \tag{23}$$

where $\gamma$ represents the standard deviation of the sample, and $\zeta = 1 - \frac{\text{CL}}{100}$ where CL is the confidence level of the estimation process. The value of $t_{\zeta/2}$ is obtained from Student's t tables [30]. Typical values of the confidence level are 0.90, 0.95 and 0.99.

Based on (23), for a given value of $T$, the maximum allowed error margin $\Delta$ is computed as follows: *i*) choose a confidence level, *ii*) compute $\zeta$, *iii*) obtain $t_{\zeta/2}$ from Student's t tables (use degrees of freedom of $T - 1$), *iv*) compute the standard deviation of the samples $\gamma$, and *v*) substitute in (23) and compute $\Delta$.

## VI. Simulation and analytical Results

The performance of the proposed algorithm is evaluated in this section through analytical results and Monte Carlo simulations. We start by showing the impact of different malicious strategies on the overall performance when no defense policy is applied. After that, we focus on dependent malicious node, showing their local performance and describing how they elude the fusion center by acting as normal nodes in some cases. The performance of the proposed algorithm is then discussed and compared to other well-known algorithms in the literature.

The considered WSN consists of $N = 10$ nodes, with $M < N$ nodes being malicious. Initially, normal nodes will be assumed to have identical local performance, whereas non-identical normal nodes will be considered later-on. Considering the detection and false alarm probabilities as indicators of the local performance, a node is considered normal if and only if $P_{d_n} > 0.5$ and $P_{f_n} <$

0.5. Otherwise (i.e. if $P_{d_n} < 0.5$ or $P_{f_n} > 0.5$), the corresponding node is considered a malicious node [14], [31]. With that in mind, in our scenario, the identical local performance for normal nodes is represented by $P_{d_n} = 0.8$ and $P_{f_n} = 0.15$. The probability that the monitored target is present is assumed to be 0.3.

### A. Global performance under the presence of malicious nodes

Fig. 2 plots the probability of a correct global decision versus the detection threshold ($K$) for different numbers of *always-one* malicious nodes. A general observation that can be taken is that if $M \geq K$ (the number of *always-one* malicious nodes is greater or equal to the detection threshold), the probability of a correct decision is equal to $P_{H_1} = 0.3$. This is due to the fact that if $M \geq K$, there are always at least $K$ '1's at the fusion center at each round, which makes the global decision always '1,' and consequently, $P_D = 1$ and $P_F = 1$. By substituting these values in (5), the correct global decision probability should equal to $P_{H_1} = 0.3$, as confirmed in Fig. 2. However, if $K$ is tuned to be larger than $M$, the overall performance is improved. Specifically, when $K > M$, both $P_D$ and $P_F$ start decreasing with a higher influence on $P_F$, which improves $P_{CD}$ as in Fig. 2. However, after the optimal value of $K$, the performance is degraded, since higher falues of $K$ reduce $P_D$.

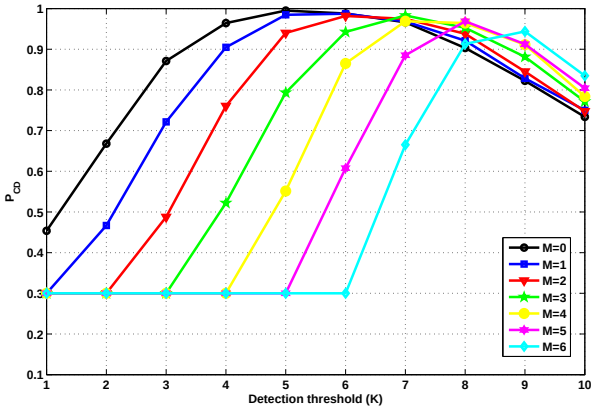


Fig. 2. The correct global decision probability ($P_{CD}$) of a WSN of 10 nodes ($M$ of them are *always-one* malicious) versus the detection threshold ($K$).

Similar to Fig. 2, Figs. 3 and 4 show the effect of *always-zero* and *always-false* malicious nodes, respectively, on the probability of a correct global decision. In Fig. 3, for $M > N - K$, the number of reported '1's is at most $K - 1$, making the global decision always '0'. Thus, both $P_D$ and $P_F$ are equal to '0' in that case. Substituting these values in (5) results in a probability of a correct global decision that is equal to $P_{H_0} = 0.7$, as shown in Fig. 3. On the other hand, if $M \leq N - K$, the number of reported '1's may be equal to (or exceed) $K$, which increases both $P_D$ and $P_F$, with the impact on $P_D$ being stronger. Below the optimal value of $K$, the increase in $P_F$ will cause the reduction of $P_{CD}$.

As mentioned earlier, *always-false* malicious nodes cause the worst performance compared to the other two strategies. This is

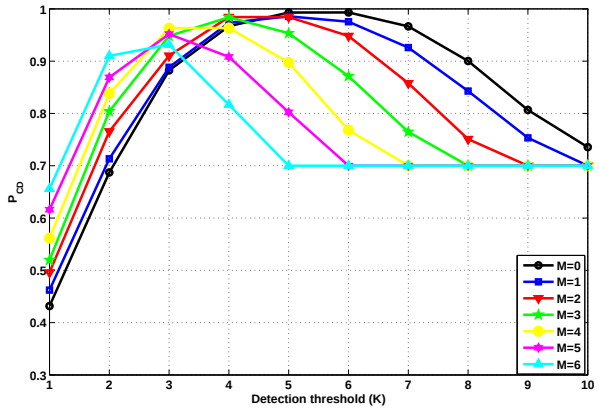


Fig. 3. The correct global decision probability ($P_{CD}$) of a WSN of 10 nodes ($M$ of them are *always-zero* malicious) versus the detection threshold ($K$).

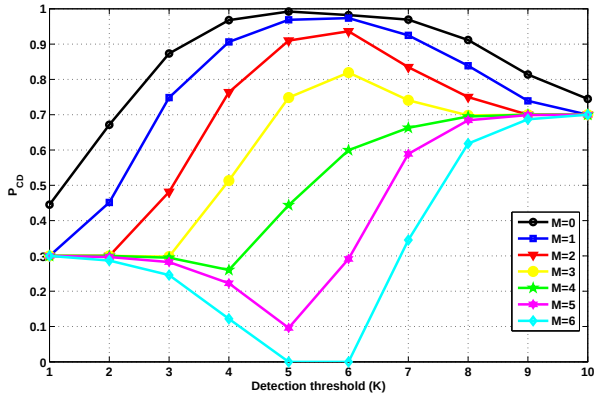


Fig. 4. The correct global decision probability ($P_{CD}$) of a WSN of 10 nodes ($M$ of them are *always-false* malicious) versus the detection threshold ($K$).

because (as the name implies) they always provide incorrect local decisions to the fusion center. The curves shown in Fig. 4 can be grouped in two different groups: *i*) the cases where the malicious nodes represent the minority ($M < \frac{N}{2}$), and *ii*) the cases where the malicious nodes represent the majority ($M \geq \frac{N}{2}$) . In the first group, the sever effect on the global decision occurs if $K \leq M$ or $K > N - M$. In the first case (i.e. $K \leq M$), malicious nodes can turn $P_F$ to 1, making $P_{CD}$ equal to $P_{H_0} = 0.3$. In the other case (i.e., $K > N - M$), $P_D$ will be equal to 0 due to the effect of the malicious nodes, which makes $P_{CD}$ equal to $P_{H_1} = 0.7$. However, for moderate values of $K$, the degradation on the reliability of the global decision can be alleviated since the global decision follows the normal nodes (that represent the majority in this group). Nevertheless, for the second group of curves (i.e., when $M \geq \frac{N}{2}$), moderate values of $K$ lead to a worse performance compared to more extreme values of $K$. This is due to the fact that moderate values of $K$ imply that the fusion center will obey the majority of the nodes, which are malicious in this case.

Another important observation on the results shown in Figs. 2-4 is that optimizing the detection threshold $K$ can improve the performance of the WSN. Specifically, optimizing $K$ in the presence of *always-one* or *always-zero* malicious nodes (regardless

of their number) yields in a $P_{CD}$ that is even better than what is achieved when no malicious nodes are present in the WSN ($M = 0$). Such an observation can be interpreted by referring to (8)-(9). Clearly, an *always-one* malicious node provides a local detection probability of 1, and an *always-zero* malicious node provides a local false alarm probability of 0. Thus, despite of their bad local performance in one aspect, they may provide a good performance in the other local performance aspect. On the other hand, in the presence of *always-false* malicious nodes, optimizing the detection threshold cannot achieve a global performance better than what is achieved when no malicious nodes are present.

It is worth emphasizing that results in Fig. 2, Fig. 3 and Fig. 4 do not depend on whether the malicious nodes are dependent or independent, which confirms that the dependency on other nodes does not increase the influence on the global decision reliability. This is because dependency on other nodes aims to minimize the probability of detection by the fusion center, without degrading the overall performance.

### B. Local performance of malicious nodes

Although independent and dependent malicious nodes have the same effect on the reliability of the global decision, they show different local performance. Referring to (8)-(9), the independent malicious nodes have constant values of $P_d$ and $P_f$ since they follow fixed malicious strategies. On the other hand, the local performance of dependent malicious nodes depend mainly on two factors, namely, the reporting turn $t$ and the detection threshold $K$. Therefore, our attention is focused on the local performance of dependent malicious nodes.

Fig. 5 shows a 3D graph of the local false alarm probability of an *always-one* dependent malicious node versus the reporting turn and the detection threshold. Notice that, at any value of $K$, the local false alarm probability of a dependent malicious node is a decreasing function of $t$. In other words, as $t$ increases (its reporting turn is delayed), the dependent malicious node will be able to hear from more other nodes ($t-1$ nodes), in order to decise whether to act maliciously or honestly. Consequently, decreasing the number of times in which the malicious node reports a false decision will definitely decrease its local false alarm probability. On the other hand, by fixing $t$, the local false alarm probability becomes a concave function of $K$. Generally, at low or high values of $K$, the malicious node can avoid sending incorrect false alarms, since the probability that $\Gamma_t$ exceeds $K$ becomes higher. Thus, according to (7), the probability of an honest behavior of the dependent malicious node increases, and hence, its local false alarm probability decreases.

Regarding *always-zero* malicious nodes, the local detection probability is the only performance metric that is dependent on the detection threshold and the reporting turn. In Fig. 6, the local detection probability of an *always-one* malicious node is plotted versus the detection threshold and the reporting turn. Clearly, the local detection probability will be improved as $t$ increases for all values of $K$.

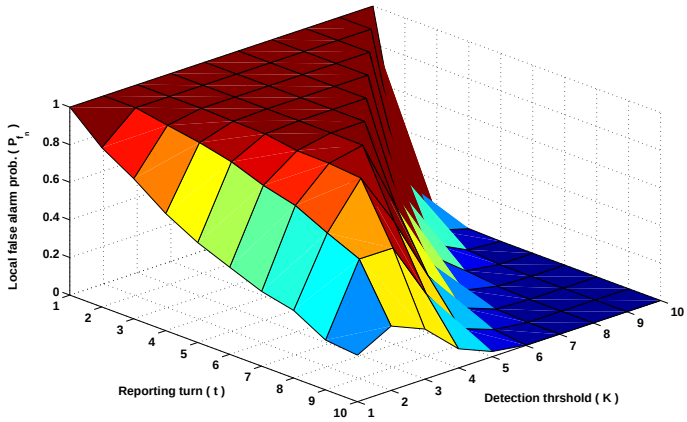


Fig. 5. The local false alarm probability ($P_{f_n}$) of an *always-one* dependent malicious node versus the detection threshold ($K$) and the reporting turn ($t$). ($N = 10$)

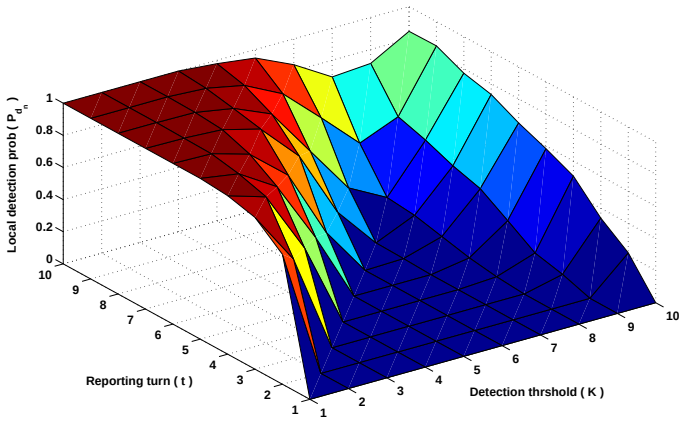


Fig. 6. The local detection probability ($P_{d_n}$) of an *always-zero* dependent malicious node versus the detection threshold ($K$) and the reporting turn ($t$). ($N = 10$)

Notice that the local detection probability for *always-one* dependent malicious nodes and the local false alarm probability for *always-zero* dependent malicious nodes have not been shown in the results since they are constant values, 1 and 0, respectively, as indicated in (8) and (9). However, for an *always-false* dependent malicious node, both probabilities are dependent on the reporting turn and the detection threshold. The local detection probability of an *always-false* dependent malicious node is exactly the same as the local detection probability of an *always-zero* dependent malicious node, as indicated in (8). Also, the local false alarm probability of an *always-false* dependent malicious node is exactly the same as the local false alarm probability of an *always-one* dependent malicious node, as indicated in (9).

### C. Performance of the proposed algorithm

Results shown in Fig. 5 and Fig. 6 prove the intelligence of the dependent malicious nodes, leading to the need for a novel

detection algorithm. In the following, we show the ability of the proposed algorithm to easily identify those malicious nodes.

As the proposed algorithm is based on observing the variation in the local performance of each node while changing the reporting turns, a WSN of $N = 10$ nodes is considered. The number of normal nodes is 4 with identical performance ($P_{d_n} = 0.8$ and $P_{f_n} = 0.15$), while the rest of nodes ($M = 6$) are assumed malicious, a single node from each type. The parameters $T$ and $\Delta$ should be carefully selected in order to maximize the performance. Thus, initially, we pick one of the normal nodes randomly, and we show the effect of $T$ and $\Delta$ parameters in its estimated $\hat{\rho}_n$. Notice that, based on (22) and the assumed values of $P_{H_0}$, $P_{d_n}$ and $P_{f_n}$, the long term value of $\rho_n$ for a normal node is 0.345. Fig. 7 shows the relation between $\Delta$ and $T$ for a different confidence levels. The maximum error margin for different values of the confidence level is obtained using (23). Notice that as the confidence level decreases, the probability that the actual estimation error exceeds the maximum error margin increases. Thus, we choose to set the confidence level to the maximum value, i.e., 0.99. In order to select an appropriate value of $T$, the corresponding value of $\Delta$ (the maximum expected error margin) should be as small as possible. From Fig. 7, for $T \geq 150$, the maximum error is $\leq 0.1$. For our simulation setup, we choose the $T = 10^3$, yielding an error margin $\Delta < 0.04$.

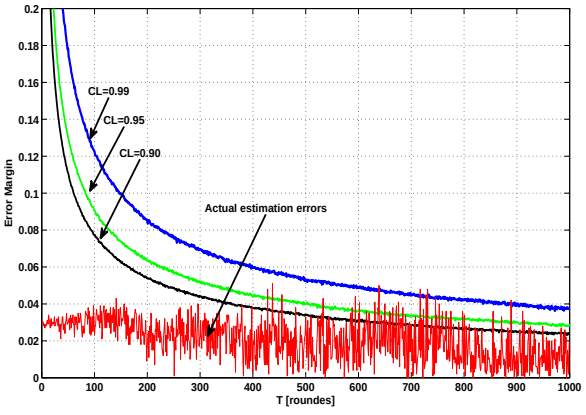


Fig. 7. The error margin versus $T$ for different values of confidence levels.

Following the proposed algorithm, the reporting order is randomly changed every $T$ sensing rounds. Every $T$ rounds, the probability of reporting '1' to the fusion center by each node ($\hat{\rho}_n$) is computed as depicted in (19)-(20). Fig. 8 plots the corresponding $\hat{\rho}_n$ for each node over $10T$ rounds. Notice that only dependent malicious nodes show a variable $\hat{\rho}_n$ as the the reporting order varies each $T$ rounds. As $\rho_n$ implicitly refers to the local performance, nodes with variable $\hat{\rho}$'s should be identified as dependent malicious nodes. On the other hand, independent malicious nodes and normal nodes show almost fixed local performance (constant value of $\hat{\rho}_n$) regardless of the reporting turn. Furthermore, independent malicious nodes can be easily detected as they show very bad performance ($\rho_n = 1, 0$ and $P_{H_0}$ for $\mathcal{I}_{\mathcal{O}}$, $\mathcal{I}_{\mathcal{Z}}$ and $\mathcal{I}_{\mathcal{F}}$, respectively).

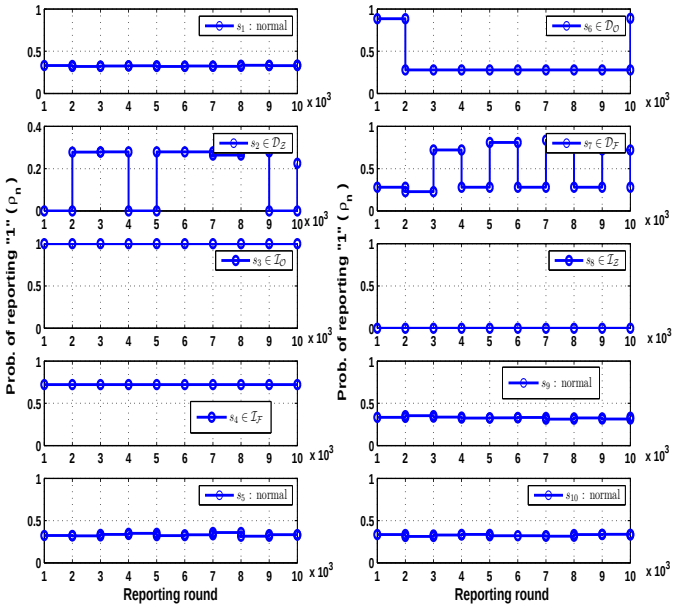


Fig. 8. The probability of reporting '1' ($\hat{\rho}_n$) versus different reporting orders for all nodes.

The global performance of the proposed algorithm is shown in Fig. 9, by plotting the global detection probability versus the global false alarm probability. Three different scenarios are considered for the purpose of comparison. The first scenario refers to the case when no malicious detection algorithm is applied, while the proposed algorithm is applied in the second and third scenarios. Specifically, in the second scenario, once a malicious node is detected, its local decision will always be ignored by the fusion center. On the other hand, in the third scenario, aiming at improving the global performance, the fusion center will consider local decisions reported by dependent malicious nodes when they act as normal nodes. Results shown in Fig. 9 clearly demonstrate the performance improvements achieved by the proposed algorithm, which obtains high detection probability at a low false alarm probability by considering some honest local decisions reported by dependent malicious nodes. Notice that, in the third scenario, the achievable global performance approaches the ideal performance (i.e, $P_D = 1$ and $P_F = 0$).

### D. Comparison with others algorithms

Finally, we compare the performance of our algorithm with the following two well-known state-of-the art schemes for the detection of malicious nodes, detailed below:

*i*) the algorithm proposed in [14], based on the number of the mismatches between each node and the global decision. This scheme estimates the local misdetection and false alarm probabilities of each node. If any of the estimated probabilities exceeds a predefined threshold, the corresponding node is identified as malicious.

*ii*) the algorithm proposed in [25], which is the only work that considers dependent malicious nodes. It counts the number of the mismatches between the local decisions of each two

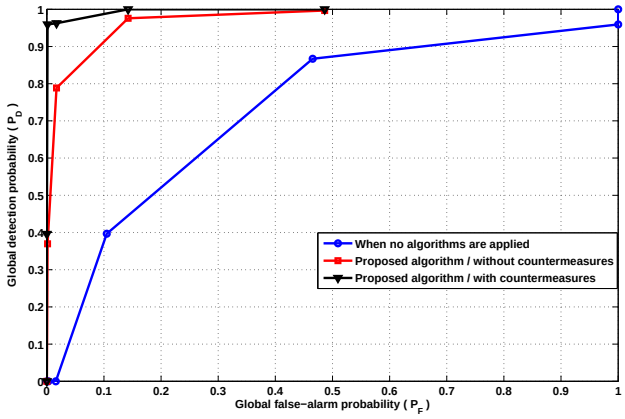


Fig. 9. The global detection probability versus the global false alarm probability for three different scenarios. ($N = 10$)

nodes. Normal nodes will have almost identical number of mismatches, while a malicious node should have abnormal number of mismatches with respect to normal node.

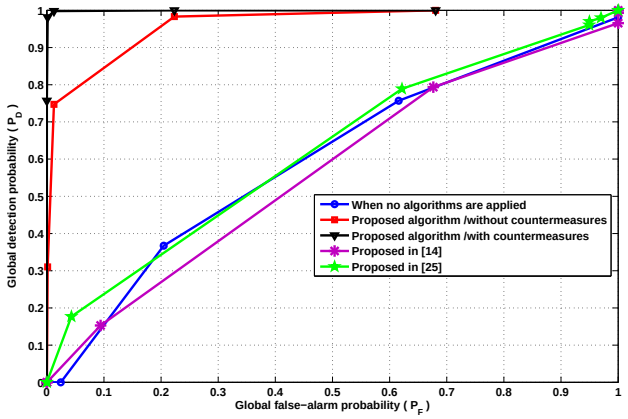


Fig. 10. The global detection probability versus the global false alarm probability for different algorithms.

A WSN of 10 nodes is considered. The number of malicious nodes is 6, a single node from each type. The normal nodes (4 nodes) are assumed to have different local performance. Particularly, the local detection probabilities of the considered normal nodes are 0.8, 0.6, 0.8 and 0.9, while the local false alarm probabilities are 0.15, 0.45, 0.25 and 0.05, respectively. Fig. 10 shows the global detection probability versus the global false alarm probability for five different scenarios. The first three scenarios are identical to those assumed in Fig. 8, while the others refer to the algorithms proposed in [25] and [14]. Since the algorithm proposed in [14] is mainly proposed for independent malicious nodes, it shows marginal improvement in the global performance, compared to the scenario when no algorithms are applied. Another reason for such performance is that the algorithm in [14] results in a high performance only if the malicious nodes represent the minority, which is not the case in the considered setup. Similarly, the improvement in the global performance obtained by applying the algorithm proposed in [25] is also marginal although the algorithm considers the presence of dependent malicious nodes. This is due the fact that the algorithm in [25] assumes only a single malicious node and many normal nodes with identical local performance. On the other hand, our proposed algorithm achieves promising results, as it is able to provide high detection probability at low false alarm probability. The performance algorithm is further enhanced if the fusion center exploits the intelligent behavior of some malicious nodes and considers their honest reports.

TABLE II
SUMMARY OF THE COMPARISON BETWEEN SOA ALGORITHMS AND THE PROPOSED ALGORITHM

| | Proposed in [14] | Proposed in [25] | Our proposal |
|---|---|---|---|
| Number of malicious nodes assumed | A small number ( should be the minority) | Only a single malicious node | Any number of malicious nodes |
| Types of malicious nodes assumed | Independent | Independent and dependent | Independent and dependent |
| Local performance of normal nodes | Should be identical | Should be identical | Can be non-identical |
| Global performance | Poor | Poor | High |

## VII. CONCLUSIONS

In this paper, the performance of WSNs in the presence of dependent and independent malicious nodes has been thoroughly investigated. Our analysis has proven that dependency on other nodes does not increase the effect on the overall performance. Instead, it only helps dependent malicious nodes to occasionally act as normal nodes, hindering their detection by conventional detection algorithms. To that end, we introduced a novel algorithm to effectively detect all types of malicious nodes in the network. Moreover, the overall performance has been further enhanced by exploiting some honest reports from dependent malicious nodes. The results have shown that the proposed algorithm outperforms existing state-of-the-art algorithms. For example, the proposed algorithm is able to achieve a detection probability of 0.85-0.99 at a false alarm probability of 0.1 for a WSN that includes 6 different malicious nodes out of 10 nodes, while the achievable detection probability by other algorithms does not exceed 0.25 at the same false alarm probability.

## REFERENCES

[1] J. Macaulay et al., "Internet of Things in Logistics," *DHL Customer Solutions & Innovation,* 2015.
[2] F. Li and P. Xiong, "Practical Secure Communication for Integrating Wireless Sensor Networks Into the Internet of Things," *IEEE Sensors J.,* vol. 13, no. 10, pp. 3677-3684, Oct. 2013.
[3] P. Spachos and D. Hatzinakos, "Real-Time Indoor Carbon Dioxide Monitoring Through Cognitive Wireless Sensor Networks," *IEEE Sensors J.,* vol. 16, no. 2, pp. 506-514, Jan.15, 2016.
[4] Y. Yu et al., "A Study on Data Loss Compensation of WiFi-Based Wireless Sensor Networks for Structural Health Monitoring," *IEEE Sensors J.,* vol. 16, no. 10, pp. 3811-3818, May15, 2016.
[5] R. Lara et al., "On Real-Time Performance Evaluation of Volcano-Monitoring Systems With Wireless Sensor Networks," *IEEE Sensors J.,* vol. 15, no. 6, pp. 3514-3523, June 2015.
[6] R. Viswanathan and V. Aalo, "On counting rules in distributed detection", *IEEE Trans. Acoust., Speech, Signal Process.*, 37.5 (1989): 772-775.
[7] X. Chen et al., "Sensor network security: a survey", *IEEE Communications Surveys & Tutorials,* vol.11, no.2, pp.52-73, Second Quarter 2009.

[8] A. Antonopoulos and C. Verikoukis, "Misbehavior Detection in the Internet of Things: A Network-Coding-aware Statistical Approach", *IEEE INDIN* 2016, Poitiers, France, July 2016.
[9] Y. Zhang et al., "Outlier detection techniques for wireless sensor networks: A survey", *IEEE Commun. Surveys Tuts.,* vol. 12, no. 2 (2010): 159-170.
[10] Y. Wang et al., "A survey of security issues in wireless sensor networks," *IEEE Commun. Surveys Tuts.,* vol.8, no.2, pp.2-23, Second Quarter 2006.
[11] A. Nordio et al., "Sensor Selection and Precoding Strategies for Wireless Sensor Networks," *IEEE Trans. Signal Process.,* vol.63, no.16, pp.4411-4421, Aug.15, 2015.
[12] F. Ye et al., "Statistical en-route filtering of injected false data in sensor networks", *IEEE J. Sel. Areas Commun.,* vol. 23, no. 4, pp. 839-850, 2005.
[13] V.P. Illiano, and E.C. Lupu, "Detecting Malicious Data Injections in Event Detection Wireless Sensor Networks", *IEEE Trans. Netw. Service Manag.,* vol. 12.3 (2015): 496-510.
[14] M. Abdelhakim et al., "Distributed detection in mobile access wireless sensor networks under byzantine attacks," *IEEE Trans. Parallel Distrib. Syst.,* vol.25, no.4, 2014, pp. 950-959.
[15] A. Vempaty et al., "Localization in wireless sensor networks: Byzantines and mitigation techniques", *IEEE Trans. Signal Process.,* 61.6 (2013): 1495-1508.
[16] J. Koh et al., "Mitigating byzantine attacks in data fusion process for wireless sensor networks using witnesses", *IEEE ICCS,* 2014.
[17] E. Soltanmohammadi et al., "Decentralized hypothesis testing in wireless sensor networks in the presence of misbehaving nodes", *IEEE Trans. Inf. Forens. Security,* 8.1 (2013): 205-215.
[18] F. Liu et al., "Insider attacker detection in wireless sensor networks," *IEEE INFOCOM,* 2007.
[19] W. Wu et al., "Localized Outlying and Boundary Data Detection in Sensor Networks", *IEEE Trans. Knowl. Data Eng.*, Vol. 19, No. 8, pp. 1145-1157, 2007.
[20] L.A. Bettencourt et al.,"Separating the Wheat from the Chaff: Practical Anomaly Detection Schemes in Ecological Applications of Distributed Sensor Networks", *IEEE Int. Conf. on Distributed Computing in Sensor Syst.,* 2007.
[21] Y. Hida et al., "Aggregation Query under Uncertainty in Sensor Networks", 2003.
[22] M.C. Jun et al., "Distributed Spatio-Temporal Outlier Detection in Sensor Networks",*SPIE,* 2006.
[23] B. Sheng et al., "Outlier Detection in Sensor Networks", *MobiHoc*, 2007.
[24] T. Palpanas et al.,"Distributed Deviation Detection in Sensor Networks", *ACM Special Interest Group on Management of Data,* pp. 77-82, 2003.
[25] H. Li and Z. Han, "Catch Me if You Can: An Abnormality Detection Approach for Collaborative Spectrum Sensing in Cognitive Radio Networks," *IEEE Trans. Wireless Commun.,* vol. 9, no. 11, pp. 3554-3565, Nov. 2010.
[26] H. Wang et al., "Network lifetime maximization with cross-layer design in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 10, 2008, 3759-3768.
[27] IEEE Standard 802.15.4, Part 15.4: *Wireless Medium Access Control and Physical Layer Specification for Low Rate Wireless Personal Area Networks*. IEEE Std. 802.15.4, December 2003.
[28] P. Kaligineedi et al., "Secure Cooperative Sensing Techniques for Cognitive Radio Systems, *IEEE ICC,* 2008, pp. 3406-3410.
[29] S. Marano et al., "Distributed detection in the presence of Byzantine attacks", *IEEE Trans. Signal Process.,* vol. 57, no. 1, pp. 16 29, jan. 2009.
[30] T.T. Soong, "Fundamentals of probability and statistics for engineers", *John Wiley & Sons,* 2004.
[31] Y.L. Sun et al., "Information Theoretic Framework of Trust Modeling and Evaluation for Ad Hoc Networks, *IEEE J. Sel. Areas Commun.,* vol. 24, no. 2, pp. 305- 317, Feb. 2006.

**Saud Althunibat** received the Ph.D. degree in Telecommunications from the University of Trento (Italy) in 2014. Currently, he is a faculty member at Al-Hussein Bin Talal University (Jordan). He has authored more than 30 scientific papers. He has received the best paper award in IEEE CAMAD 2012, and was selected as Exemplary Reviewer for IEEE Communications Letters in 2013. His research interests include cognitive radio networks, wireless sensors networks, physical-layer security and resource allocation.

**Angelos Antonopoulos** received the Ph.D. degree from the Signal Theory and Communications (TSC) Department of the Technical University of Catalonia (UPC) in December 2012. He has authored over 50 research papers on various topics, including 5G mobile networks, cooperative communications, radio resource management, network sharing, and energy efficient network planning. In January 2015, he was nominated as Exemplary Reviewer for the IEEE Communications Letters, while he has received the best paper award in IEEE GLOBECOM 2014, the best demo award in IEEE CAMAD 2014, the 1st prize in the IEEE ComSoc Student Competition 2015 (as a Mentor) and the EURACON best student paper award in EuCNC 2016.

**Elli Kartsakli** got her Ph.D. in Wireless Telecom. from UPC in 2012. She holds a degree in Electrical and Computer Engineering from the National Technical Univ. of Athens, Greece (2003) and an M.Sc. in Mobile and Satellite Com. from the Univ. of Surrey, UK (2004). She has participated in several national and European projects and is currently a senior researcher at IQUADRAT. Her primary research interests include 5G networks and architectures, channel access protocols and energy efficient schemes.

**Fabrizio Granelli** is Associate Professor and Delegate for Education at Department of Information Engineering and Computer Science of the University of Trento (Italy). He was IEEE ComSoc Distingushed Lecturer in the period 2012-2015 and he is currently ComSoc Director for Online Content. He is author of more than 170 papers published in international journals, books and conferences in the field of networking, with specific emphasis on wireless networks, cross-layer optimization, cognitive radios and networks, software defined networking.

**Christos Verikoukis** received the Ph.D. degree from UPC in 2000. He is currently a Head of the SMARTECH Department at CTTC and an Adjunct Professor at the University of Barcelona. He has published 98 journal papers and over 160 conference papers. He is also a co-author of three books, 14 chapters in other books, and two patents. He has participated in more than 30 competitive projects, and has served as the principal investigator of national projects in Greece and Spain. He was General Chair of the IEEE CAMAD12, CAMAD13 & CAMAD14, and the TPC Co-Chair of the IEEE Healthcom13 and the LATINCOM 2014. He has also served as the co-chair of the CQRM symposium in ICC 2015 and ICC 2016 and the chair of the eHealth symposium in Globecom 2015. He is currently Chair of the IEEE ComSoc Technical Committee on Communication Systems Integration and Modeling (CSIM).