Journal of Internet Services and Applications

RESEARCH Open Access

Clustering and reliability-driven mitigation of routing attacks in massive IoT systems

Aldri L. Santos[1*†], Christian A. V. Cervantes[1†], Michele Nogueira[1†] and Burak Kantarci[2†]

Abstract

As an integral component of the 5G communications, the massive Internet of Things (IoT) are vulnerable to various routing attacks due to their dynamic infrastructure, distinct computing resources, and heterogeneity of mobile objects. The sinkhole and selective forwarding attacks stand out among the most destructive ones for infrastructureless networks. Despite the countermeasures introduced by legacy intrusion detection systems (IDS), the massive IoT seeks novel solutions to address their unique requirements. This paper introduces *DeTection of SinkHole And SelecTive ForwArding for Supporting SeCure routing for Internet of THIngs* (THATACHI), a new IDS against sinkhole and selective forwarding attacks that target routing mechanism in massive and mobile IoT networks. To cope with the density and mobility challenges in the detection of attackers and ensuring reliability, THATACHI exploits watchdog, reputation and trust strategies. Our performance evaluation under an urban scenario shows that THATACHI can perform with a 99% detection rate, 6% of false negative and false positive rates. Moreover, when compared to its closest predecessor against sinkhole attacks for IoT, THATACHI runs with at least 50% less energy consumption.

Keywords: Routing attacks, IDS, Massive IoT

1 Introduction

The massive amount of objects equipped to heterogeneous communication and computing specifications is inter-connected through RFID, GPS and NFC that are the fundamental building blocks of the Internet of Things (IoT). Unique identity, communication and processing capability are the minimum requirements of the IoT devices (nodes) [1]. Key application areas of the IoT are industrial and home automation, public safety, home automation, environmental monitoring and so on [2]. A resilient IoT routing service is essential to data exchange and data dissemination between nodes. Among its threats, we highlight the sinkhole and selective forwarding attacks, seen as the most destructive attacks at the network layer [3]. A sinkhole attacker seeks to attract to it the largest amount of traffic in a given area to damage a sinking point of data sent by the devices [4], whereas selective forwarding attacker selectively drops some of the packets passing through it, and thus devices cannot forward data packet to the destination, damaging the network performance [5].

The current Intrusion Detection System (IDS) for IoT networks have included different methods, mechanisms and techniques for providing confidentiality, data authentication, access control, privacy and trust between users and things [6]. Agent-based IDS, statistical signal processing or machine learning are commonly applied in small, fixed networks but do not consider heterogeneous devices. Thus, they are not particularly tailored for massive and mobile IoT as they would lead to high resource consumption under massive IoT resulting in vulnerabilities to various types of attacks aimed at disrupting network communication. Thus, the state-of-the-art in IDS calls for novel solutions that are particularly tailored for massive IoT to cope with dynamic interconnection between heterogeneous devices, offer reliability, and isolate attackers in the routing service, protecting data dissemination.

*Correspondence: aldri@inf.ufpr.br
†All authors contributed equally to this work.
[1]Department of Informatics, Federal University of Paraná, Curitiba, Brazil
Full list of author information is available at the end of the article

This work presents a system for the mitigation of sinkhole and selective forwarding attacks that target routing services of massive mobile IoT networks. The proposed system, called deTection of sinkHole And SelecTive forwArding for supporting seCure routing for internet of tHIngs (THATACHI), aims at detecting and isolating IoT devices with sinkhole or selective forwarding behavior. THATACHI exploits routing based on hierarchical clustering to cope with the density and mobility of devices. It combines multi-level *watchdog*, reputation and confidence techniques for the detection of attackers so as to establish trust between devices. Our performance evaluation of THATACHI's under Cooja simulator confirms its effectiveness in mitigating both attacks with a detection rate of 99%, false negative of 6% and a low power consumption of $2e4\ mJ$.

The remaining of the article is organized as follows. Section 2 discusses the related works. Section 3 describes in details the proposed system, called THATACHI. Section 4 presents the performance evaluation and analyzes numerically its effectiveness. Finally, concluding remarks are presented in Section 5 alongside future directions.

2 Related work

Recent studies in the literature have highlighted the needs of sophisticated IDS to meet security requirements in IoT environments [3, 7–10]. In a nutshell, IDSs aim to identify attack behavior and locate an attacker within a system [3]. Particularly, IoT attackers targeting the routing service lead to the prominent sinkhole and selective forwarding attacks due to the common device heterogeneity.

Sinkhole and selective forwarding attacks are variations of the denial of service attack, whose major goal lies in denying service, in this case, routing service, to network nodes. Table 1 compares related works from the literature and classify them in four main classes: (1) works handling sinkhole attacks; (2) works handling only selective forwarding attacks, (3) works addressing both attacks; (4) works that intend to be generic to any type of denial of service attack.

In [11], the authors have proposed an IDS to detect sinkhole attacks on a network based on the Routing Protocol for Low-power and lossy networks (RPL) This protocol is designed for networks composed of energy constrained devices and low memory capacity. Hence, data transmission is unreliable, presenting low data rate and high loss rate. The IDS engages distributed supernodes to establish a finite state machine for the RPL protocol [16–18]. Those supernodes monitor destination nodes through requests derived from rules applied to check and monitor the nodes. However, this scheme results in high false positives compared to other IDS proposals for IoT existing in the literature, and thus the system loses its credibility.

Table 1 Comparing related works: IDS for attacks on IoT routing

Related works			
Targeted attack	Work	Approach	Issues
Sinkhole	Le et al. [11]	Use of supernodes, finite machine state, RPL	High positive rate compared to other IDSs for IoT in the literature.
	Cervantes et al. [7]	Use of reputation and trust model in INTI, the designed system	Neglect other types of attacks in IoT routing.
Selective forwarding	Mathur et al. [8]	Cryptography	High positive and negative rates, and high energy consumption.
Sinkhole and Selective forwarding	Sheikhan and Bostani [3]	MapReduce, supervised and unsupervised machine learning	Focuses only on static nodes and execution time of approximately 13 min.
	Khan and Herrmann [12]	Reputation, trust management	False positive and negative rates above 60%.
Other	Adeilson et al.-2018 [13]	IDS against DoS attacks	Not specific to a type of DoS generating high positive and negative rates
	Bhatti et al.-2018 [14]	IDS based on machine learning techniques for detecting network anomalies	Detection not effective for all targeted attacks, and high positive and negative rates.
	Sonar et al. [15]	Watchdog on hardware	Limited to a IoT with few nodes and the use of a limiar constraints its effectiveness.
	Yang et al. [10]	IDS based on watchdog for detecting false data injection on the network and uses a Bayesian spatio-temporal model	Model yields high energy consumption, and the probabilistic testing applied results in high error rates.

Furthermore, they assume static nodes and disregard node's energy consumption.

In [3], the authors have proposed a hybrid distributed IDS for real-time detection of sinkhole attacks and selective forwarding in 6LoWPAN. The model is based on the MapReduce approach that makes use of a new version of Optimum-Path Forest (OPF), the Modification of Supervised Optimum-Path Forest (MOPF), which is a supervised graph-based machine; and an unsupervised algorithm, Optimum Path Forest Clustering (OPFC), to classify nodes. However, again, that proposal addressed only static nodes and does not consider the impact of node mobility in attack detection.

The main drawbacks of the scheme presented in [3] consist in its high false positive and false negative rates compared to other IDS for IoT existing in the literature. The authors consider a static network, being its results limited, once taking into account mobility for IoT is a paramount. Finally, despite the authors' goal to propose an IDS for real-time detection, they present the execution time for MOPF of 837 s (approximately 13 min), what goes against real-time. They do not present the execution time for OPCF. In [8], the authors built a solution for selective forwarding attack detection in a medical IoT Wireless Sensor Network (WSN), where sensors send data sample to an Access Point (AP). Thus, the AP can address data encryption and forward over the Internet for storing and processing them on servers. Though, the solution generates high false positive and negative rates, and high energy consumption.

Watchdog is one of most traditional techniques applied to monitor networked system [19]. The authors in [10] proposed an IDS based on watchdog for the detection of false data injection attacks. It takes into account Bayesian Spatial-Temporal Hierarchical Model (HBT) to describe the features of the sensed data, and a sequential probability test is applied to identify an attacking device. However, the HBT model yields high energy consumption and the probabilistic test results large error rate. In [15], the authors built a hardware watchdog mechanism to monitor jailbrake of the device side channel and alert the user when a given threshold is reached, signalling thus a vulnerability. Though, the solution focuses on a small IoT network with few devices, and the use of threshold constraints the effectiveness of lateral channel attack detection.

In [12], an IoT IDS is proposed against selective forwarding, sinkhole, and modification of the identifier attacks. Those solution applies a trusted management mechanism to enable devices to manage neighbor reputation. However, it is not effective to the types of target attacks, and its false positive and negative rates are very high. In [7], an IDS, named Intrusion detection of siNkhole attacks on 6lowpan for interneT of thIngs (INTI), is built for detection and mitigation of sinkhole attacks, in which IoT devices tailor a hierarchical structure among themselves for supporting the routing and behavioral monitoring of devices. Thus, upon the detection of a sinkhole attack, the device alerts its neighbors to isolate the attacker. Though, this IDS is only tailored against sinkhole attacks, neglecting other attacks in routing.

The authors in [13] proposed a Distributed Denial of Service (DDoS) detection system and evaluated its performance on the Raspberry Pi platform. Their results show that the system detects DDoS attacks in IoT environments, but the scenario they use does not represent a real scenario. The authors in [14] presented an attack detection approach based on machine learning techniques for anomaly detection and on a decision module to identify relevant attacks in the IoT network. The approach is implemented on a single board computer and has been systematically evaluated using various protocol attacks and commercial IoT devices to verify their effectiveness and feasibility in a realistic scenario. Although the detection accuracy is good, it consumes a lot of energy.

As mentioned, Table 1 highlights the most relevant works from the literature concerned to the detection of attacks in IoT routing. These works are directly related to this one. From a carefully analysis of these works, we observe critical issues on them when applied to IoT networks, such as either they are so generic or so specific. For instance, the work [13] presents a solution to the big class of DDoS attacks, resulting in high false positive and false negative rates. Another example is the work [8], where a solution is presented specifically to a WSN medical IoT. In the majority of these works, we observe the lack of handling important requirements and features from the IoT networks, such as computational resource constraints, limited energy, specific routing protocol, node mobility and real-time operation for specific classes of applications. The work in [3] employed supervised and unsupervised machine learning, but the execution time for the solution achieves approximately 13 min, what is contradictory with the IoT requirements. Those works also present high false positive and negative rates, what in practice constraints their effectiveness. Hence, this work presents an intrusion detection system that can achieve a balance in the addressed scope, not being so generic and not so specific. It intends also to address node mobility and decrease false positive and negative rates.

3 The THATACHI IDS

This section details the proposed IDS, called THATACHI (*De**T**ection of Sink**H**ole **A**nd Selec**T**ive Forw**A**rding for Supporting Se**C**ure routing for Internet of **THI**ngs*). Figure 1 illustrates the components of the THATACHI architecture, which comprises of the **Clustering** and **Reliability** modules. The former sets up *clusters* and performs maintenance tasks; the latter performs tasks as monitoring, detection and isolation of attacking devices that target data routing functions.

The Clustering Module is responsible for configuring clusters. It acts on the classification of nodes as *members, associates and leaders, allowing paths through* leaders and associate nodes to offer scalability and extend the useful

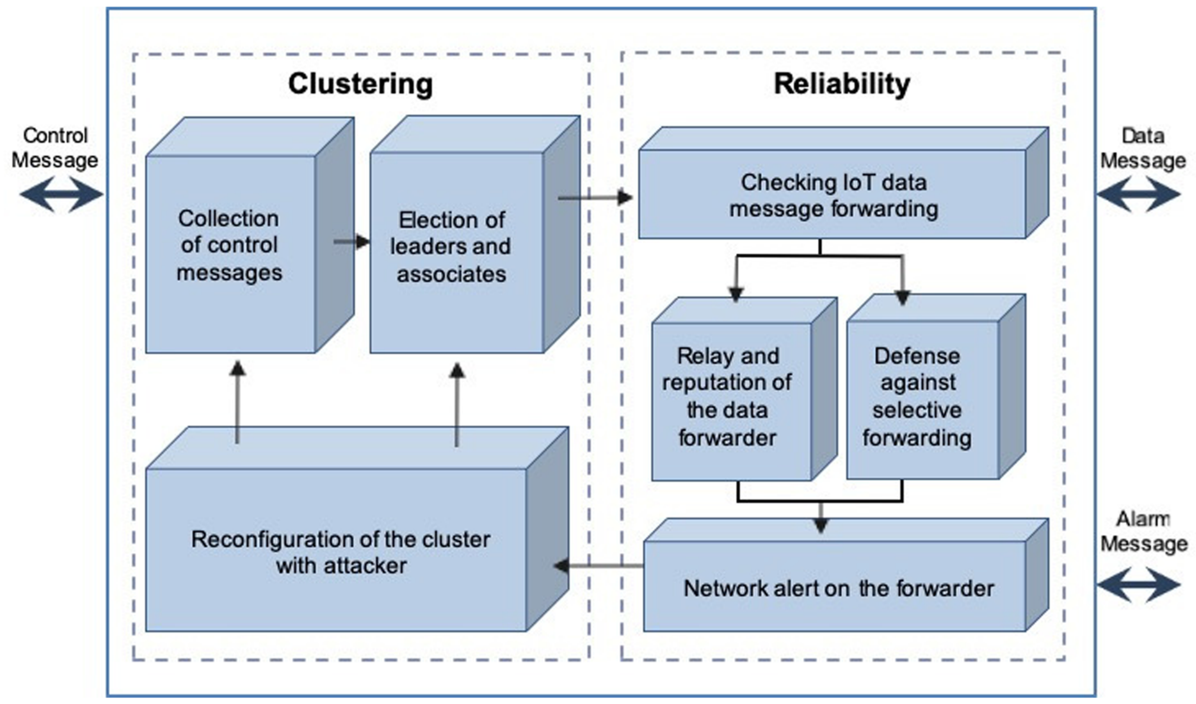
Clustering
Reliability
Control Message
Data Message
Alarm Message
Collection of control messages
Election of leaders and associates
Reconfiguration of the cluster with attacker
Checking IoT data message forwarding
Relay and reputation of the data forwarder
Defense against selective forwarding
Network alert on the forwarder

Fig. 1 The THATACHI Architecture

network lifetime, as shown in Fig. 2. The role assigned to each node is adaptable and may change across time with the network reconfiguration, due to node mobility or an attack event. Initially, nodes have no classification, being called as free nodes, collecting and transmitting broadcast control messages. These messages allow a node to estimate the number of neighbors for the leader election.

Free nodes are classified as leaders when they have a higher number of neighbor nodes than others in its communication range. After that, other free nodes are classified as member nodes and join the leader node forming

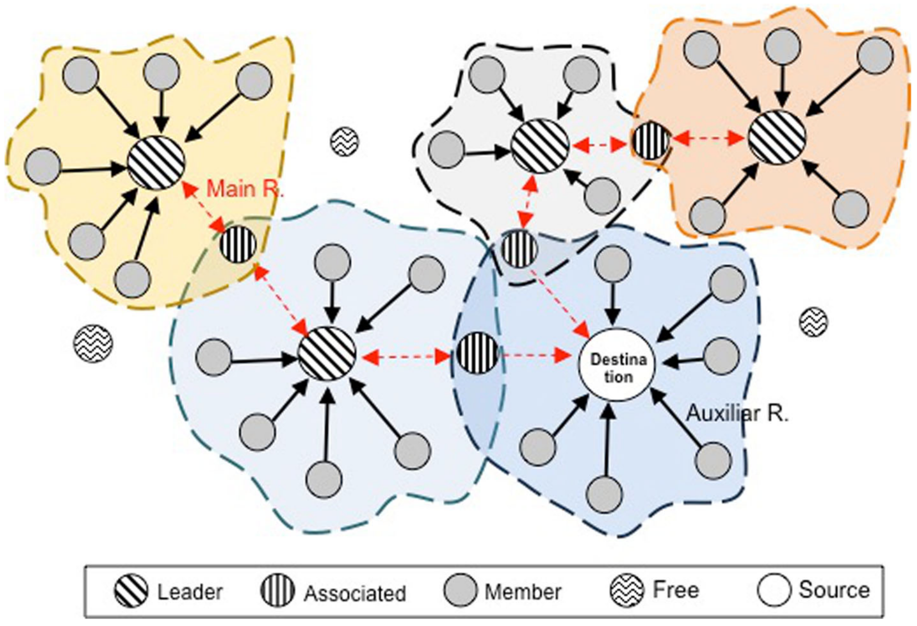

Fig. 2 Clusters configuration

clusters. Next, leaders check if there are member nodes receiving messages from other leaders, so that such nodes can play an associate role, since they are able to interconnect different clusters. If there are multiple nodes that can play an associate role at the same area, the node n_i with the highest amount of energy (QE_i) is chosen as associate. QE_i is determined by $QE_i = TEr_i - TEc_i$, so that TEr_i is the current amount of remaining energy in the node n_i, since the last time it was charged; and TEc_i is the total of energy spent by the node, in general, for performing tasks, such as packet transmission, packet processing and other. Hence, $TEr_i + TEc_i$ is equal to 100% of the energy capacity of a node n_i.

Cluster reorganization takes as basis the model presented in [7] and runs when one of the nodes (*i*) naturally fails, (*ii*) leaves the *cluster*; or (*iii*) an attack occurs. When a member node is affected by some of these issues, the leading node removes the identity (ID) of this node from its list, and the remaining member nodes can join other neighboring *clusters*. If a leader node fails or leaves the *cluster*, the member and associated nodes request a new leader election. In case of the leader node being an attacker, the member and associate nodes carry out a new cluster configuration. When an associate node fails or leaves the cluster, it is possible for the leader to choose another one, as long as it is within the common area shared by different *clusters*. Hence, the leader must remove the ID of this node from its list of neighbors. If an associate node is an attacker, the leading node propagates a rebuild message, then all affected nodes perform a new cluster configuration, and the top leader node adds the attacker node ID in a *blacklist*. Otherwise, when both leaders are within the same transmission range, their clusters are merged. Then, the *cluster* with highest number of members is absorbed by the other one. This method aims to reduce the number of routing leaders and to keep the network scalability.

The Reliability Module is responsible for maintaining the reliability property between the IoT network nodes. It consists of three components responsible for monitoring, detecting and isolating attacking nodes. The *Monitoring Component* checks the node's behavior in relation to forward the received data. For that, it makes use of *watchdog* strategies on two levels. This allows the monitor node to compute the number of transmissions forwarded by a "top" node relative to its own messages sent to it. Note that a node is said to be on top of another if it is in a lower rank. Thus, when the number of received transmissions by the top node is equal to the number of its output transmissions, that node is handled as normal. Otherwise, the monitor node assumes that some abnormal behavior is on going. Figure 3 shows a situation in which the node n_8 is an attacking node discarding packets. Since, a monitor node would need to be in the intersection of levels 1 and 2 in order to test n_8 sending packets to the destination through n_8 and observing if n_8 forwards the packets or not.

The *Detection Component* makes use of reputation and trust approaches for the detection of sinkhole or selective forwarding nodes. Reputation is the belief or perception that all nodes establish by iterations, actions, or information exchange between them. Thus, three metrics are valued: uncertainty (u), belief (b) and disbelief (d), using the Beta distribution, *Beta* (α, β), to represent the node's reputation, following Bayesian inference. Each node carries out the calculations, and the computation of these forecasts $(u,b,d) \in (0,1)^3 : u+b+d = 1$. Uncertainty ($u$) is the

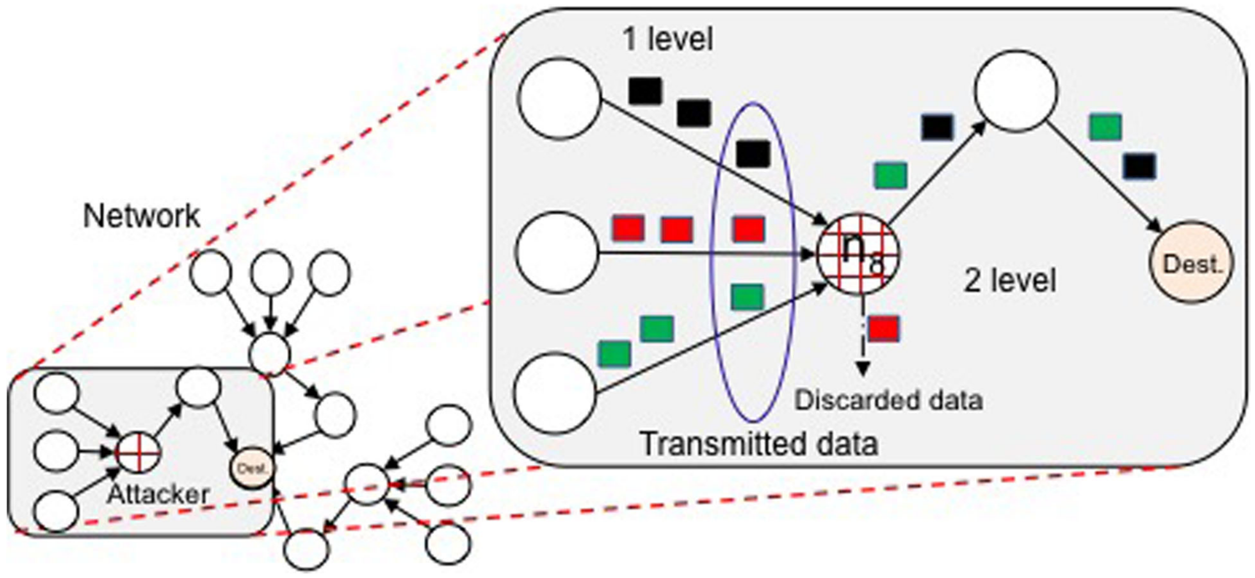

Fig. 3 The action of the watchdog component

normalized variance of the Beta distribution, established according to: $u = \frac{12*\alpha*\beta}{(\alpha+\beta)^2*(\alpha+\beta+1)}$. Certainty is formulated as $(1 - u)$, which can be divided into b and d according to its number of iterations. The two-node trust transmission is defined by $\frac{\alpha}{(\alpha+\beta)}$. The calculation of b is given by: $b = \frac{\alpha}{(\alpha+\beta)}(1-u)$. Finally, the calculation of unbelief (d) is achieved by: $d = (1-u) - b = \frac{\beta}{(\alpha+\beta)}(1-u)$.

After establishing the value of those metrics, the number of iterations between nodes (exchanged data messages) is computed using their status (St). Therefore, each node propagates its status about the routing service in all message transmissions to support the reputation calculation. These values are input data to the *Dempster-Shafer theory formula* [20] to raise the probability of detection and to reduce false alarms. As reputation (R) is a continuous value within the limits $P[0,1]$, when this value is greater than or equal to 0.5, the node is seen as "*good*", otherwise it is an "*invader*". R for each node is given by $m_1(H) \oplus m_2(H)$ varying a continuous value between $0 \leq m_2 \leq 1$ (Eq. 1). That result considers $m_2 < 0.5$ as bad reputation, and the opposite value as good reputation.

$$\begin{aligned}
m_1(H) \oplus m_2(H) &= \frac{1}{K}[m_1(H)m_2(H) + m_1(H)m_2(U) + m_1(U)m_2(H)] \\
m_1(\overline{H}) \oplus m_2(\overline{H}) &= \frac{1}{K}[m_1(\overline{H})m_2(\overline{H}) + m_1(\overline{H})m_2(U) + m_1(U)m_2(\overline{H})] \\
m_1(U) \oplus m_2(U) &= \frac{1}{K}[m_1(U)m_2(U)].
\end{aligned} \quad (1)$$

$$\begin{aligned}
Where: K =& m_1(H)m_2(H) + m_1(H)m_2(U) + m_1(U)m_2(H) + \\
& m_1(\overline{H})m_2(\overline{H}) + m_1(\overline{H})m_2(U) + m_1(U)m_2(\overline{H}) + \\
& m_1(U)m_2(U)
\end{aligned}$$

The next step is to establish the trust level (C) (Eq. 2), which means the relationship of honesty that one given node has with another. It obtains the confidence level of a node, which varies in the closed interval [0,1]. Thus, when the trust level is greater than 0.5, the node is seen as "*good*". Otherwise, it is seen as "*invader*". In Eq. 2, we also consider the level of uncertainty (u) and the values for γ and δ.γ and δ are values representing node interactions, being $\gamma, \delta > 0$ and given by $\gamma = u\gamma + R$ and $\delta = u\delta + (1-R)$.

$$C = \mathbf{E}(Beta(\gamma+1, \delta+1)) = \frac{\gamma+1}{\gamma+\delta+2} \quad (2)$$

The value threshold d_{thresh} guides the detection of a *SH* attack. It takes as basis the Packet court and Probability Transmission (CPPT), which consists of the expected number of routing packet transmissions necessary to successfully deliver one data message from the sender to the receiver, including retransmissions. The CPPT of a link is based on the direct and reverse link delivery relationships. The direct delivery relation (d_f) is probability of a data packet be successfully delivered at the receiver, and the reverse delivery relation (d_r) is the probability of the acknowledgment packet being successfully received by the sender. Thus, the CPPT of a link is calculated as: $CPPT = \frac{1}{(d_f \times d_r)}$. The inverse of CPPT corresponds to the index of the link delivery. The detection limit d_{thresh} of a route is calculated as the inverse of the sum of CPPT from all links i along the path p. $d_{thresh} = \frac{1}{\sum_{linki \in p} CPPT_i}$ and $AR = N \times d_{thresh}$, where AR is the Acceptance Rate and N is the number of packets transmitted by the source node.

Algorithm 1 details the operation of cluster configuration. All nodes periodically send control messages in *Broadcast* to inform their identifiers, category, energy index, type of operation and number of neighbors *(l.2)*. The periodic sending of such messages always inserts a random time around milliseconds to avoid simultaneous transmissions, minimizing the number of possible collisions. The time control variables consist of *Interval* and *Rand()*. The

Algorithm 1 Cluster configuration

```
1:  procedure SENDBEACON
2:      $Send(Beacon, Cat, IdMy, Rank, IdLeader, Root, IdLarger)$
3:      $Timer(SendBeacon, Interval + Rand)$
4:  end procedure
5:  procedure RECVBEACON
6:      $Recv(Beacon, Cat, IdMy, IdNeig, Rank, IdLeader, Root, IdLarger)$
7:      $AmountMyNeig \leftarrow AmountMyNeig + 1$
8:  end procedure
9:  procedure CHOOSELEADER
10:     if ($Cat \Leftrightarrow Free$) then
11:         if ($Rank > AmountMyNeig$) then
12:             $IdHighMembAmount \leftarrow IdNeig$
13:         else
14:             $IdHighMembAmount \leftarrow IdMy$
15:         end if
16:         if ($IdHighMembAmount \Leftrightarrow IdNeig \vee IdMy$) then
17:             $ChangeCat \Leftarrow Leader$
18:         end if
19:     end if
20:     if (($Cat \Leftrightarrow Associated \vee Member$)) then
21:         $ChangeCat \Leftarrow Leader$
22:     end if
23:     if ($Cat \Leftrightarrow Free \vee Associated \vee Member$) then
24:         $InsertListNeig \leftarrow \{IdNeig, Cat\}$
25:     else
26:         $DeletListNeig \leftarrow \{IdNeig, Cat\}$
27:     end if
28: end procedure
```

first one is a fix send period, and the second one adds a variation in the sending time of the messages *(l.3)*. The procedure *RecvBeacon* allows each node to receive the parameters sent by other nodes *(l.2)*. The *ChooseLeader (l.9)* makes the leader election, considering the number of neighbors of each cluster's node. The neighbors number is computed through the control messages received by the node. After knowing the number of neighbors, nodes broadcast such information to their neighboring nodes *(l.10-19)* to run the leader election. For that, the node with the largest number of neighbors is chosen as the leader node, and its category is changed to leader *(l.20-22)*. Each node keeps a list of neighbors (*ListNeig*) based on the received control messages *(l.23-27)*. If the transmitting node is categorized as *free*, the receiving node adds it in its list. However, a node must be removed of the list when it fails or leaves the cluster.

A special case occurs when a free node is alone. It will become leader when receiving a message from a node categorized as associate or member. This strategy is used because if the received message is from a member node, when it becomes leader, the member node is classified as associate and the new leader can establish communication. If the message is from an associate, the new leader only establishes communication with it.

A SinkHole (SH) or Selective Forwarding (SF) node may act on the network as a leader, associate, or member. Algorithm 2 details the detection of attacking nodes in the face of a suspicion. It uses the uncertainty, belief and disbelief metrics explained before, using the Beta distribution to represent the node's reputation. Hence, *DetecRepT* receives the $< Id, St >$ values of the forwarder node detected as a suspect node to determine whether it is a good (normal) node or an attacking node. These values are based on their behavior in the transmission of messages. Thus, the node that detected the suspect node uses its own observations (values) in the reputation calculation *(l.2-4)*. Further, the proper observations of the node defined by *c* and the value of the qualification of the suspicious node (St) are used *(l.5)*. Next, the confidence of the suspicious node *(l.7)* is calculated. It will be taken the node as SH attacker when it has a reputation and trust below [0.5]; so it will not forward the information sent by other nodes *(l.10-16)*. Whereas the procedure for detecting SF attack initially calculates the expected CPPT *(l.18-20)*. Next, it determines the value of detection (threshold) and the acceptance rate *(l.21-25)*, which is conditioned for the detection of the attacking node.

The *Attacker Isolation Component* is responsible for isolating the attacking node and requesting the *clusters* rebuilding. Hence, when detecting a SH or SF attack, a node generates and propagates an alarm message to alert neighboring nodes. Then, it also promotes the attacker isolation by sending a reset message to its neighbors. The

Algorithm 2 Detection of attacking nodes

```
1:  procedure DETECSH(Id,St)
2:      $i \Leftarrow uncertainty \leftarrow \{af, bta\}$
3:      $c \Leftarrow belief \leftarrow \{af, bta\}$
4:      $d \Leftarrow disbelief \leftarrow \{af, bta\}$
5:      $DetecRep \Leftarrow m \leftarrow \{c, St\}$
6:      $u \Leftarrow 1 - (1/IteRations[Root]$
7:      $Gma \Leftarrow (u * Gma) + DetecRep$
8:      $Dlta \Leftarrow (u * Dlta) + (1 - DetecRep)$
9:      $Trust \Leftarrow (Gma + 1)/(Gma + Dlta + 2)$
10:     if $(DetecRep > 0.5) \wedge (Trust > 0.5)$ then
11:         $InKlin \Leftarrow$ "good"
12:     else
13:         $InKlin \Leftarrow$ "sinkhole"
14:     end if
15:     return $InKlin$
16: end procedure
17: procedure DETSELECFOR(df,dr)
18:     $CPPT \Leftarrow 1/(df * dr)$
19:     $D \Leftarrow 1/(CPPT_a + CPPT_b)$
20:     $AR \Leftarrow N * D$
21:     if $(AR > C_{pkt})$ then
22:         $InKlin1 \Leftarrow$ "selectiveforwarding"
23:     else
24:         $InKlin1 \Leftarrow$ "good"
25:     end if
26:     return $InKlin1$
27: end procedure
```

main data propagated in the restore message consists of the cluster classification, so that nodes in the same *cluster* start a re-clustering. Particularly, there are three ways to isolate sinkhole or selective forwarding attackers: (*i*) when the attacker is a member node, the leader itself isolates that node; (*ii*) when the attacking node acts as a leader, the member nodes isolate the attacking node, or if there is an associated node, it isolates the attack; (*iii*) when the attacking node acts as an associate node, it is isolated by the leader with the highest rating, closing the communication with the attacker. It is also important to check if there are other associate nodes with lowest number of neighbor within the *cluster*, so that they can route messages to the destination node. Otherwise, the leader propagates a restoration message to its members and they seek out neighboring *clusters*.

4 Analysis

This section presents the performance evaluation of the THATACHI system. We analyze its effectiveness for mitigating SH and SF attacks on a scenario that represents a dense IoT routing with mobile devices. Here, we show how THATACHI can detect SH and SF attacks in a street urban scenario [17, 21], where there are several types of devices. Furthermore, we compare THATACHI to the Intrusion detection of siNkhole attacks on 6lowpan for interneT of thIngs (INTI), once both systems has as common goal to detect sinkhole and also we can show improvement from THATACHI compared to this previous work. Both systems have been implemented in Cooja simulator considering the Contiki open code operating system. Table 2 shows our simulation settings.

The system has been evaluated in a scenario of 50 nodes following the Tmote Sky[1] setting emulated in Cooja, which features a 16-bit msp430 MCU, 10 kB RAM, 48 kB ROM, a cc2420 802.15.4 radio transceiver, an external Flash memory, and temperature, humidity and brightness sensor. Nodes are distributed randomly in an area of 200x200m. Mobile nodes follow speeds varying of 5, 15 and 25 km/h, randomly distributed, taking as basis the reference [17]. Nodes present different transmission ranges varying in 20m, 30m and 40m. The parameter values follow the work in [22]. The employed routing protocolo is a modification of RPL [17]. The mobile nodes represent devices from users with wireless devices, such as smartphones, PDAs, notebooks, smart watches. Each user moves inside an area following the Random Waypoint mobility model and they represent pedestrians, runners, cyclists and vehicles.

Edge nodes alternately generate one data packet at each interval of 10 s and they send it to the destination node. Nodes make use of the wireless channel, following the Unit Disk Graph Medium (UDGM) *Distance Loss propagation model*, enabling them to establish communication within the network. UDGM, as a class, models the transmission range as an ideal disk in which all the nodes outside that disk do not receive packets while the nodes within that disk receive all the packets [23]. The UDGM *Distance Loss* propagation model is an extension of the UDGM *Constant Loss*. Its main advantages, compared to the other propagation models in Cooja, consist in modelling interferences and the packets can be transmitted with a TX success ratio and received with RX success ratio, offering a higher sense of reality for the simulation. UDGM specifies the transmission success ratio in an asymmetric per-link base and can define propagation delays for the links. The total simulation time was 1500s, so that all nodes could exchange packets, in this case, an amount of 145 packets per node. Nodes run the User Datagramm Protocol (UDP), considering transmission ranges of 10m, 20m, 30m and 40m. We also consider attacking nodes percentages of 20% and 30% from the total number of nodes in the simulations. The attacking nodes act maliciously throughout the simulation period. Our scenario is meaningful to portray a dense IoT area, as

[1] http://www.contiki-os.org/start.html

Table 2 Simulation settings

Parameters	Values
Number of nodes	50 Tmote Sky motes
Simulation time	1500s
Speeds	5 km/h, 15 km/h e 25 km/h
Node pause time	60s, 90s e 120s
Area	200x200 meters
Package type used	UDP
Time to generate data packet	10s
Standard	RPL
Wireless channel	Unit disk graph Medium
Transmission radius	20m, 30m e 40m
Percentage of attacking nodes	20% e 30%
Data rate	10^2 kbps

the devices establish interconnected clusters where data is routed to an access point of a structured network.

Each plotted point in the figures means an average of 35 executions with 95% confidence level. In the figures, we compare results for THATACHI (THA) and INTI (INT), under 20% and 30% of attacking nodes from the total amount of nodes in the network. The metrics applied to measure the effectiveness of both systems are detection rate, false positive rate, false negative rate, packet delivery rate, energy consumption, accuracy, overload, throughput.

Detection rate (T_{det}) accounts the attacks correctly identified. This metric is achieved by Eq. 3, where X means the total number of attacker identified by the system, given in the form of $X=(d, c)$, where d is the value of the detection performed, and c is the current behavior of the node $n_i \in P$.

$$T_{det} = \frac{\sum D_i}{|X|} \forall_i \in X, \text{ where } \quad D_i = \begin{cases} 1, \text{ if } & d_i = c_i, \\ 0, \text{ if } & d_i \neq c_i. \end{cases} \quad (3)$$

False negative rate (Tx_{fn}) indicates how many times that attackers were considered by the system as trusted. This metric is calculated by Eq. 4, where X counts the total number of iterations performed by the system, and T_{det} is the detection rate achieved by Eq. 3.

$$Tx_{Fn} = |X| - T_{det} \quad (4)$$

False positive rate (Tx_{fp}) determines the amount of times that the system has detected attack as negative. Tx_{Fp} is calculated by Eq. 5, and Z is the set of iterations of normal nodes, so that $Z = (d, c)$, where d means the value of the detection performed by system and c is the real status of the node $n_i \in P$, where $c= 1$ means an attacker.

$$Tx_{Fp} = \frac{\sum Dp_i}{|Z|} \forall_i \in Z, \text{ where } \quad Dp_i = \begin{cases} 1, \text{ if } & d_i = c_i, \\ 0, \text{ if } & d_i \neq c_i. \end{cases} \quad (5)$$

Packet delivery rate ($Tx_{Delivery}$) accounts the number of data packets successfully received. It consists of the number of received packets divided by the number of packets originated by the source.

$$Tx_{Delivery} = \frac{NreceivedPackets}{NsentPackets} X100 \tag{6}$$

Energy consumption (E_{gc}) indicates the total power consumption of the network nodes during the simulation. This calculation is represented by Eq. 7, so that $\sum_{z=1}^{i} TE_i$ means the total initial power sum of all nodes in the network and $\sum_{z=1}^{i} TE_r$ is the total sum of the remaining energy of the nodes. At where $\sum_{z=1}^{i} n_z = 1$ and $\forall\ R$ thus obtaining the total energy consumed when the system runs. The energy consumption in each simulated node follows the Energest module [24], implemented in the simulator Cooja.

$$E_{gc} = \sum_{z=1}^{i} (TE_i - TE_r) \tag{7}$$

Accuracy (A_{cc}) how much degree of accuracy a result has in comparison to the true value. The metric is calculated by Eq. 8, so that $\sum_i vp$ means the sum of the number of true positives, $\sum_i vn$ is total sum of true negatives, $\sum_i fn$ is the sum of the number of false positives and $\sum_i fp$ is the sum of false positives, where i means the number of legitimate nodes in the network. This metric is established between the values of $0 \leq Acc \leq 100$, being the value closer to 100 denotes a greater accuracy of the system.

$$A_{cc} = \left(\frac{\sum_i vp + \sum_i vn}{\sum_i vp + \sum_i fn + \sum_i fp + \sum_i vn} \right) * 100 \tag{8}$$

Overload (O_{load}) accounts the average overload of the network when the packets are transmitted. This metric is obtained by Eq. 9, in which $Tx_{Delivery}$ means the delivery rate of packets over the total time of the simulation ($T_{Tsimulation}$).

$$O_{load} = \frac{Tx_{Delivery}}{T_{Tsimulation}} \tag{9}$$

Throughput (T_{gpt}) refers to the performance of the IDS over a specific period, as shown in Eq. 10, where T_{dt} is the detection rate and $T_{Tsimulation}$ is the amount of time spent.

$$T_{gpt} = \frac{T_{dt}}{T_{Tsimulation}} \tag{10}$$

4.1 Results

The attack detection rates of THATACHI and INTI are shown in Figs. 4 and 5. THATACHI has gotten between 96 to 100% over attacks. Such performance is due to the joint reliability strategies of watchdog, reputation and trust between devices. INTI has gotten 95% face SH attacks, but less than 35% in face of SF attacks.

The False Negative (FN) and False Positive (FP) rates of both systems are shown in Figs. 6 and 7. THATACHI has obtained a FN rate lower than 5% for both attacks. An issue on detecting an attacker can occur when there is an error in the calculation made by the lower node about the amount of packets transmitted by the upper node. Thus, some nodes take a long time to identify attackers. INTI got higher FN rate, about 8% for sinkhole and up to 20% for selective forwarding. The reason of those values are because INTI cannot detect when an attacker discards all

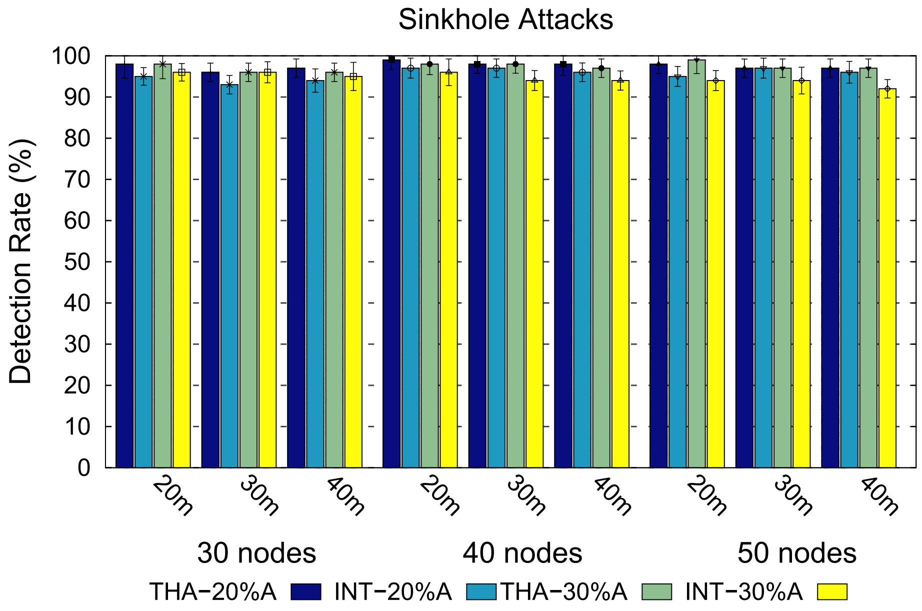

Fig. 4 SH Detection rate

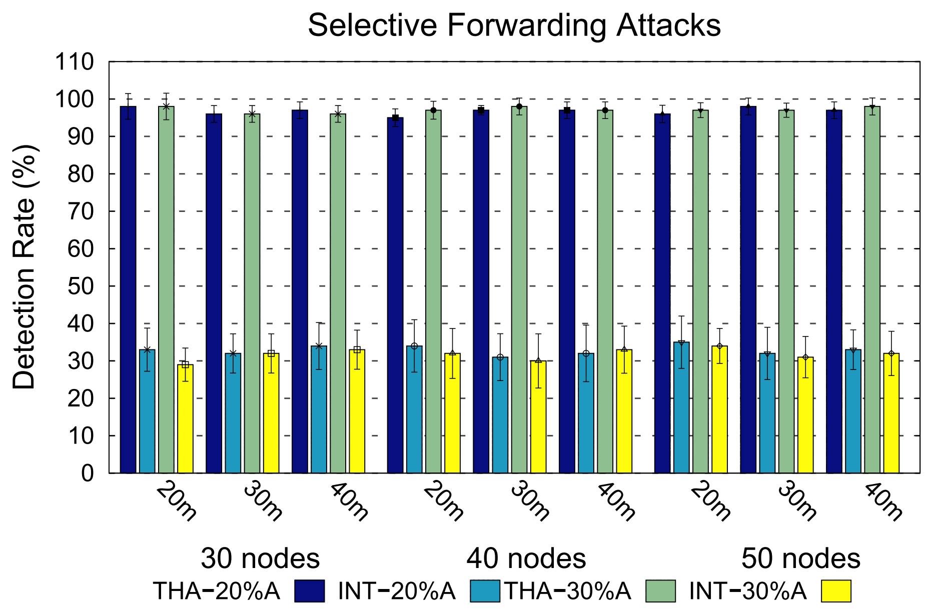

Fig. 5 SF Detection rate

packets. THATACHI has achieved rates lower than 3% for both attacks, and INTI got rates ranging from 5% to 12%. THATACHI false detections occur when forwarding nodes delay to forward packets. Thus, temporarily those are seen as attackers, but as they move on and the packet is forwarded, they are identified as good ones. Hence, THATACHI has shown for both attacks lower detection rates than solutions presented in works as [25, 26].

The delivery packet rate of THATACHI is close to 99%, in face of both attacks, whereas INTI got 95% and 54% under SH and SF attacks, respectively, as shown in Figs. 8 and 9. The THATACHI improvement is due to all techniques applied to rise the reliability and lifetime of the devices. As shown in Figs. 10 and 11, the energy consumption of INTI is higher under both attacks, achieving value higher than 29000 (mJ) with SH attacks, and higher than 33000 (mJ) with SF attacks. While the energy consumption of THATACHI was lower than 20500 (mJ) for both attacks. Those improvements presented by THATACHI are resulted from the application of the CPPT technique,

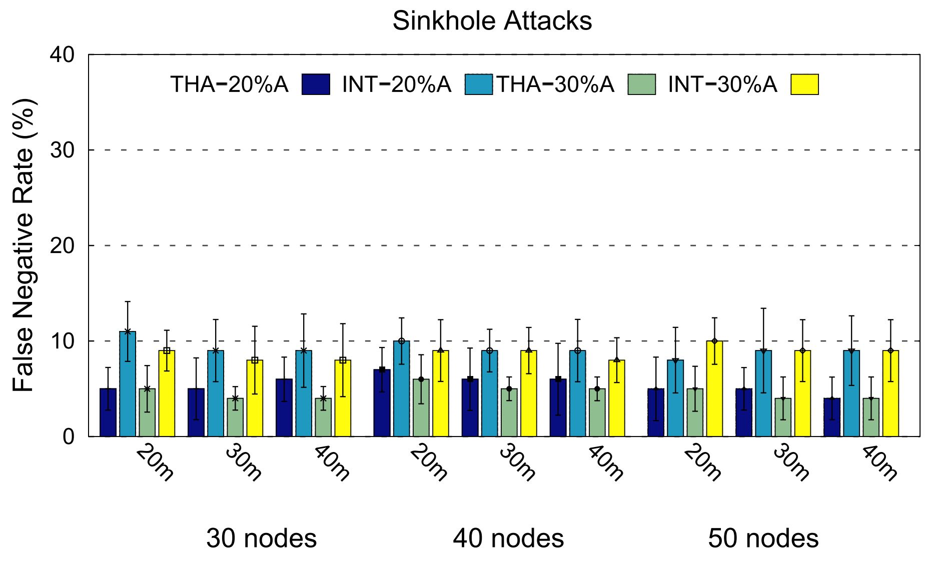

Fig. 6 FN rates for SH attacks

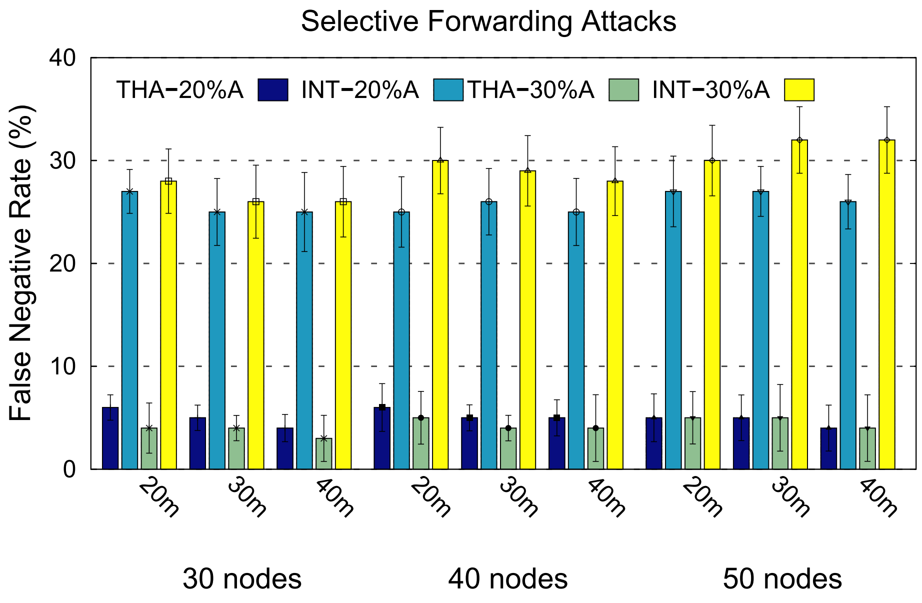

Fig. 7 FN rates for SF attacks

which enables the verification of packet forwarding by the receiver of its messages, thus avoiding that the honest node stays connected for a long time transmitting packets to a potential attacking node. In addition, THATACHI allows the detector node to change its state from "on" to "rest", since it does not matter if it stays in that state. Hence, the node saves energy. The energy consumption is calculated by using the power trace module, which is a program that can be added as a plug in to simulators like Cooja. It enables to estimate the energy consumption generated by running the simulated application [27].

Figures 12 and 13 exhibit the accuracy rate of both systems taking into account transmission ranges of 30m and 40m, and different amount of nodes. In both scenarios, THATACHI exhibits a continuous increase, then the simulation time goes by reaching a rate of almost 99% for both scenarios. While INTI only reaches 70% at the scenario with transmission range of 30m, and it shows a lower performance at the scenario with range of 40m.

Figure 14 shows the distribution of the amount of leaders, members, associate and free nodes allocated at the task of data routing, considering the variety of nodes and

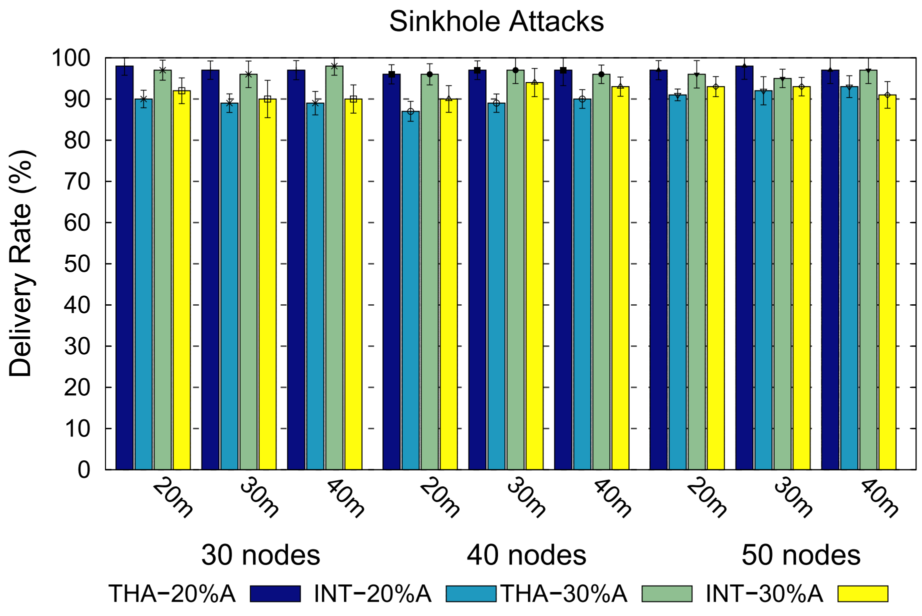

Fig. 8 PDR under SH Attacks

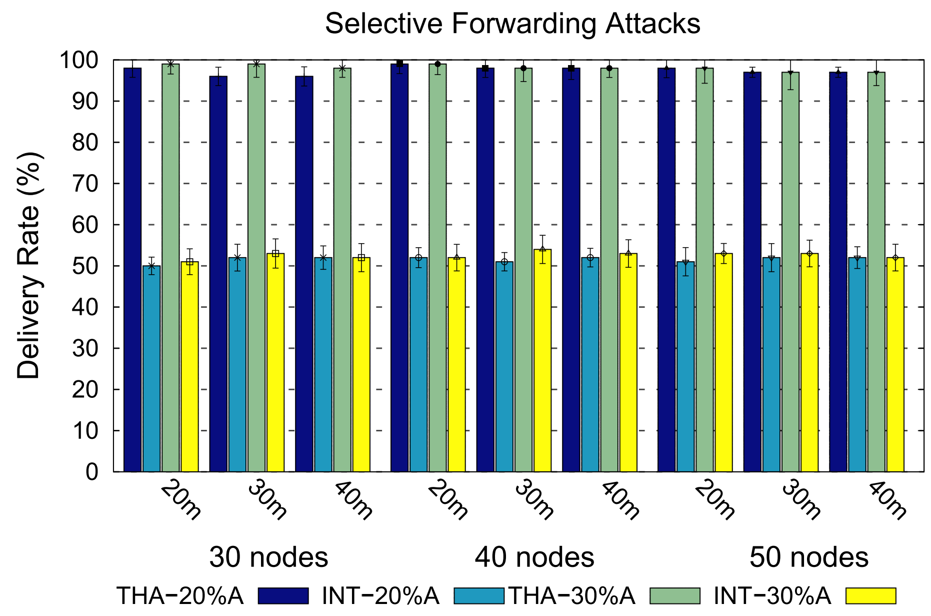

Fig. 9 PDR under SF attacks

their transmission ranges. These values show the roles played by nodes in THATACHI for the simulation. Due to clustering, THATACHI has exhibited a proportional distribution of roles for all investigated node amount.

The rates of throughput and overload are shown in Figs. 15 and 16, respectively. THATACHI's throughput is higher than INTI, reaching almost 99%, whereas INTI achieved a maximum throughput of 48%. Those results are because THATACHI ensures a secure communication in presence of both attacks. Hence, the overload by running THATACHI is greater than INTI.

5 Conclusion

This article presented an IDS, named THATACHI, for detecting sinkhole and selective forwarding attacks on the routing service in massive and mobile IoT. THATACHI deals with the density and mobility of the network by clustering the task of data forwarding. It monitors

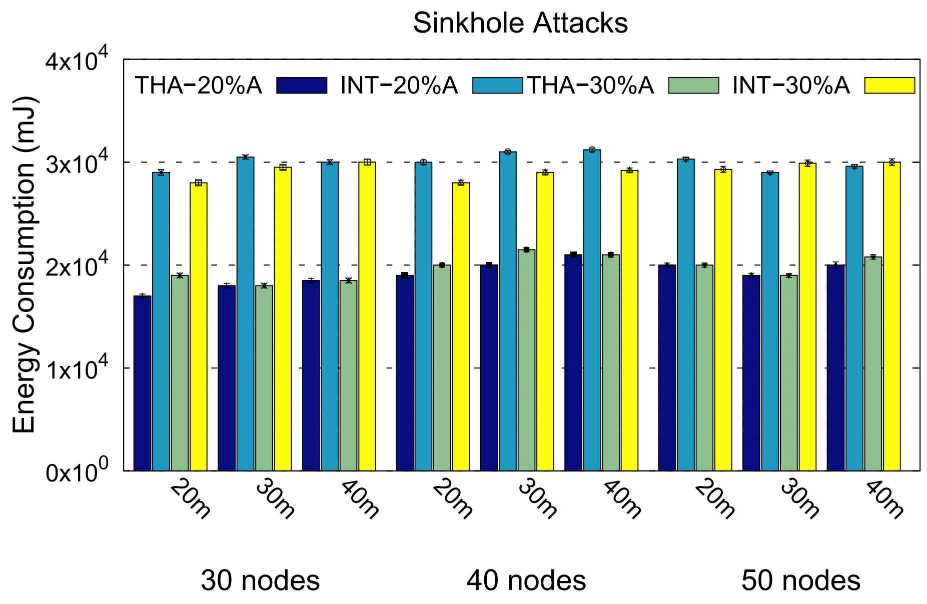

Fig. 10 Energy consumption under SH attacks

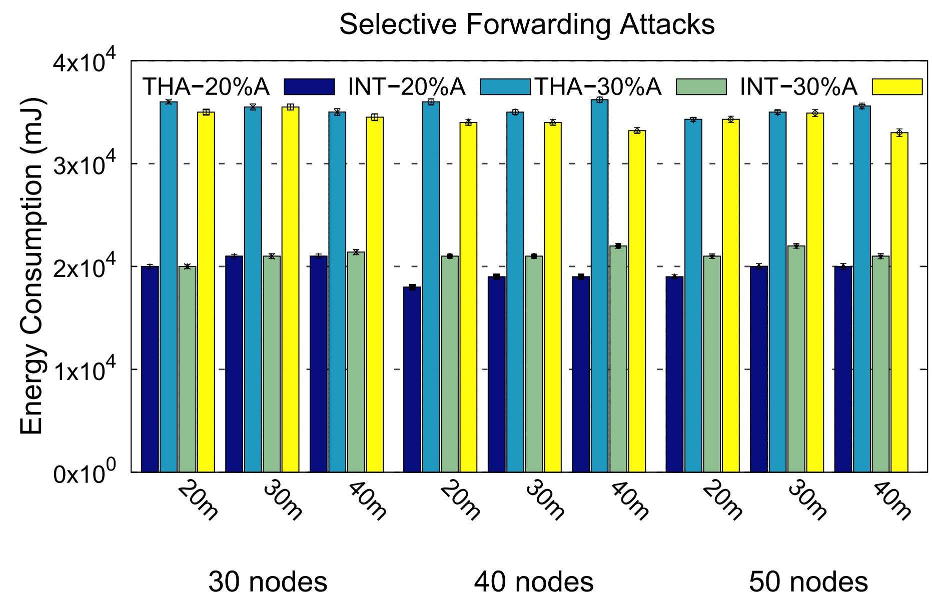

Fig. 11 Energy consumption under SF attacks

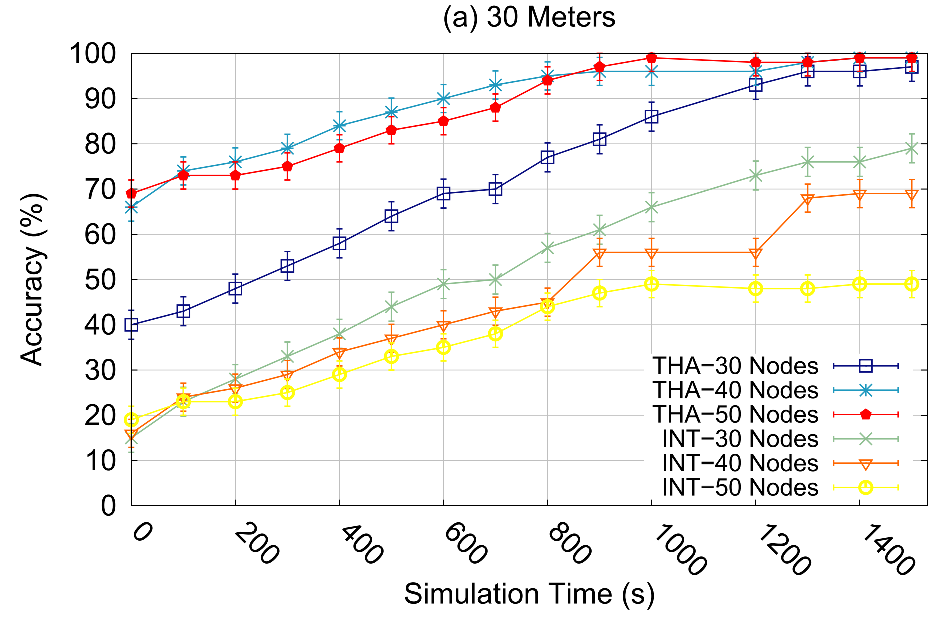

Fig. 12 Accuracy rate for range of 30 meters

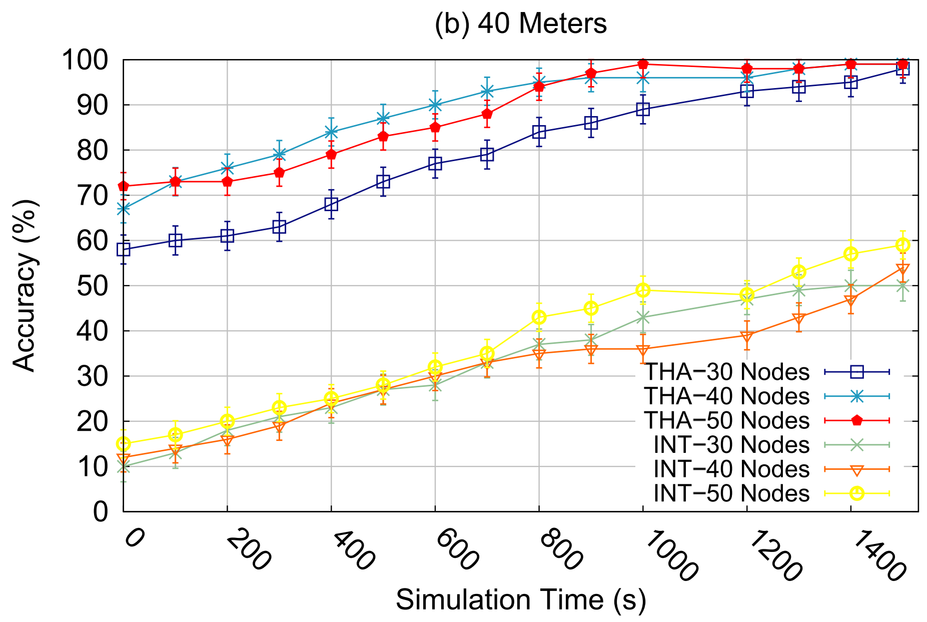

Fig. 13 Accuracy rate for range of 40 meters

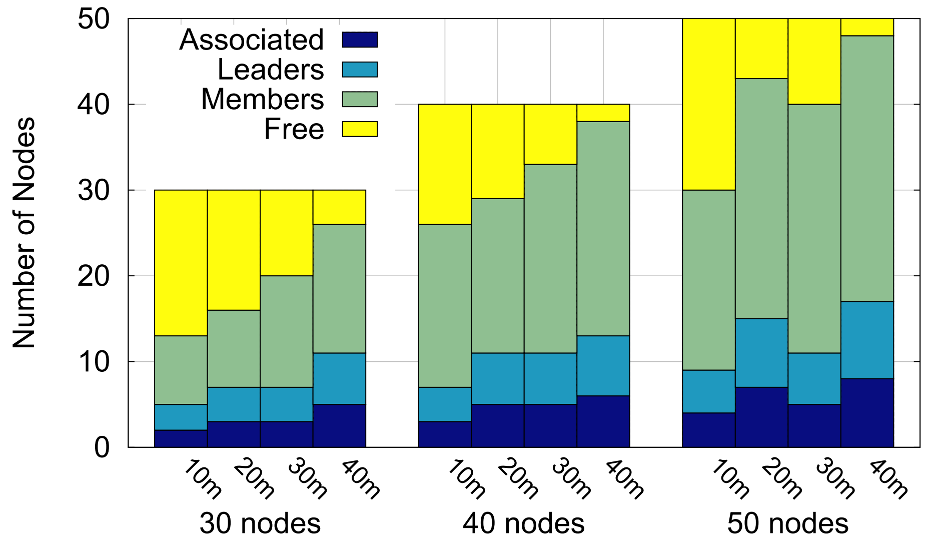

Fig. 14 Distribution of nodes role at THATACHI

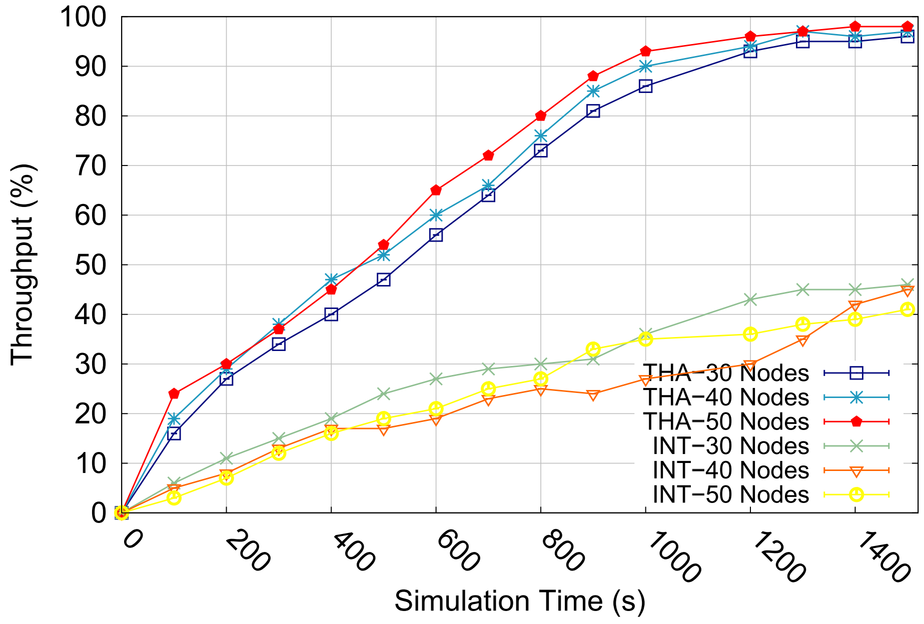

Fig. 15 Throughput rate

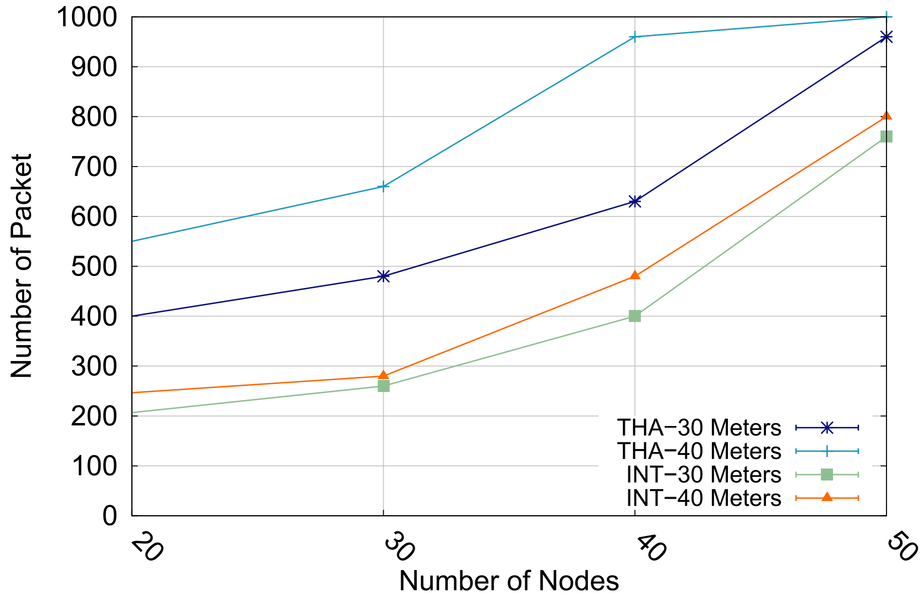

Fig. 16 Overload rate

node's behavior in data routing, then misbehaving nodes can quickly be identified and isolated of the network. Simulation results have shown the THATACHI effectiveness in protecting the IoT network against sinkhole and selective forwarding attacks, as well as its low false positive and negative rates, and low energy consumption. Future work will investigate the THATACHI under other types of attacks, such as personification attacks in the IoT routing service.

Abbreviations

AP: Access point; CPPT: Packet court and probability transmission; DDoS: Distributed denial of service; FN: False negative; FP: False positive; HBT: Bayesian spatial-temporal hierarchical model; ID: Identity; IDS: Intrusion detection system; INTI: Intrusion detection of siNkhole attacks on 6lowpan for interneT of thIngs; IoT: Internet of Things; MOPF: Modification of supervised optimum-path forest; OPF: Optimum-path forest; OPFC: Optimum path forest clustering; PDA: Personal device assistant; RPL: Routing protocol for low-power and lossy networks; SF: Selective forwarding; SH: SinkHole; THATACHI: DeTection of sinkHole And SelecTive forwArding for supporting seCure routing for internet of tHIngs; UDGM: Unit Disk graph medium; UDP: User datagramm protocol; WSN: Wireless sensor network

Acknowledgements

Not applicable.

Authors' contributions

AS, CC, MN and BK made a substantial contribution to this manuscript. All authors read and approved the final manuscript.

Funding

This work has been supported by CNPq and CAPES in Brazil, and the Visiting Researcher Program (VRP) – Office of Vice President-Research (OVPR) at University of Ottawa, Canada. CNPq assisted this work by the Prof. Santos and Prof. Nogueira's fellowships that allowed them to supervise, review and write the manuscript together with the student. CAPES funded this research by means of the student scholarship that assisted the student in the related work study, implementation, and the analysis of the results. The funding from the VRP, University of Ottawa, allowed the collaboration between the research group in Brazil and Prof. Kantarci's team at University of Ottawa. The VRP assistance allowed fruitful discussion among the researchers that enriched the work.

Availability of data and materials

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Competing interests

The authors declare that no competing interests exist.

Author details

[1]Department of Informatics, Federal University of Paraná, Curitiba, Brazil. [2]School of Electrical Eng. and Computer Science, University of Ottawa, Ottawa, Canada.

Received: 10 December 2018 Accepted: 31 July 2019
Published online: 06 September 2019

References

1. Bari N, Mani G, Berkovich S. Internet of things as a methodological concept. In: Fourth IEEE International Conference on Computing for Geospatial Research and Application (COMGeo). San Jose: IEEE; 2013. p. 48–55.
2. Zarpelão BB, Miani RS, Kawakani CT, de Alvarenga SC. A survey of intrusion detection in internet of things. J Netw Comput Appl. 2017;84: 25–37.
3. Sheikhan M, Bostani H. A security mechanism for detecting intrusions in internet of things using selected features based on mi-bgsa. Int J Inf Commun Technol Res. 2017;9(2):53–62.
4. Rassam MA, Zainal A, Maarof MA, Al-Shaboti M. A sinkhole attack detection scheme in mintroute wireless sensor network. In: IEEE International Symposium on Telecommunication Technologies (ISTT). Kuala Lumpur: IEEE; 2012. p. 71–75.
5. Airehrour D, Gutierrez J, Ray SK, et al. A trust-aware rpl routing protocol to detect blackhole and selective forwarding attacks. Aust J Telecommun Digit Econ. 2017;5(1):50.
6. Sicari S, Rizzardi A, Grieco LA, Coen-Porisini A. Security, privacy and trust in internet of things: The road ahead. Comput Netw. 2015;76:146–64.
7. Cervantes C, Poplade D, Nogueira M, Santos A. Detection of sinkhole attacks for supporting secure routing on 6lowpan for internet of things. In: IFIP/IEEE International Symposium on Integrated Network Management (IM). Ottawa: IEEE; 2015. p. 606–11.
8. Mathur A, Newe T, Rao M. Defence against black hole and selective forwarding attacks for medical wsns in the iot. Sensors. 2016;16(1):118.
9. Thanigaivelan NK, Nigussie E, Kanth RK, Virtanen S, Isoaho J. Distributed internal anomaly detection system for internet-of-things. In: IEEE Annual Consumer Communications & Networking Conference (CCNC). Las Vegas: IEEE; 2016. p. 319–20.
10. Yang L, Ding C, Wu M, Wang K. Robust detection of false data injection attacks for the data aggregation in internet of things based environmental surveillance. Comp Netw. 2017;129(part 2):410–28.
11. Le A, Loo J, Chai KK, Aiash M. A specification-based ids for detecting attacks on rpl-based network topology. Information. 2016;7(2):25.
12. Khan ZA, Herrmann P. A trust based distributed intrusion detection mechanism for internet of things. In: IEEE International Conference on Advanced Information Networking and Applications (AINA). Taipei: IEEE; 2017. p. 1169–76.
13. da Silva Cardoso AM, Lopes RF, Teles AS, Magalhães FBV. Real-time ddos detection based on complex event processing for iot. In: 2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI). Orlando: IEEE; 2018. p. 273–4.
14. Bhatt P, Morais A. HADS: hybrid anomaly detection system for iot environments. In: International Conference on Internet of Things, Embedded Systems and Communications (IINTEC). Hammamet: IEEE; 2018. p. 191–6.
15. Sonar S, Roy DB, Chakraborty RS, Mukhopadhyay D. Side-channel watchdog: Run-time evaluation of side-channel vulnerability in fpga-based crypto-systems. IACR Cryptol EPrint Archive. 2016;2016:182.
16. Winter T, Thubert P, Brandt A, Hui J, Kelsey R, Levis P, Pister K, Struik R, Vasseur J, Alexander R. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks (RFC 6550). Internet Eng Task Force (IETF). 2012;1:59–66.
17. Dohler M, Watteyne T, Winter T, Barthel D. Routing Requirements for Urban Low-Power and Lossy Networks. Internet Eng Task Force (IETF). 2009;1:1–20.
18. Accettura N, Grieco LA, Boggia G, Camarda P. Performance analysis of the rpl routing protocol. In: IEEE International Conference on Mechatronics (ICM); 2011. p. 767–72.
19. Hasan MM, Mouftah HT. Optimization of watchdog selection in wireless sensor networks. IEEE Wirel Commun Lett. 2017;6(1):94–97.
20. Tang H. A novel fuzzy soft set approach in decision making based on grey relational analysis and dempster–shafer theory of evidence. Appl Soft Comput. 2015;31:317–25.
21. Bellavista P, Cardone G, Corradi A, Foschini L. Convergence of manet and wsn in iot urban scenarios. IEEE Sensors J. 2013;13(10):3558–67.
22. Sanshi S, Jaidhar C. Assessment of objective functions under mobility in rpl. In: Recent Findings in Intelligent Computing Techniques. Gateway East: Springer; 2018. p. 565–76.
23. Bitencourt HV, da Cunha AB, da Silva DC. Simulation domains for networked embedded systems. In: SBMicro – Microelectronics Students Forum. Brasilia: Brazilian Microelectronics Society (SBMicro) and the Brazilian Computer Society (SBC); 2012. p. 1–4.
24. Casado L, Tsigas P. Contikisec: A secure network layer for wireless sensor networks under the contiki operating system. In: Nordic Conference on Secure IT Systems. Berlin Heidelberg: Springer; 2009. p. 133–47.
25. Hosseinpour F, Amoli PV, Farahnakian F, Plosila J, Hämäläinen T. Artificial immune system based intrusion detection: innate immunity using an unsupervised learning approach. Int J Digit Content Technol Appl. 2014;8(5):1.

26. Hosseinpour F, Vahdani Amoli P, Plosila J, Hämäläinen T, Tenhunen H. An intrusion detection system for fog computing and iot based logistic systems using a smart data approach. Int J Digit Content Technol Appl. 2016;10(5):34–46.
27. Dunkels A, Eriksson J, Finne N, Tsiftes N. Powertrace: Network-level power profiling for low-power wireless networks. Swed Inst Comput Sci. 2011;1:1–14.

Publisher's Note