A Trust and Energy-based routing framework for the IoT network

Anahita Mahfoozi[1], Mohammad Yousef Darmani [2*]

Computer Engineering Dept., K. N. Toosi University of Technology, Tehran, Iran.

[1]Anahitamahfozi@email.kntu.ac.ir, [2]Darmani@kntu.ac.ir

[*]Corresponding author

Received 06/04/2023, Revised 08/07/2023, Accepted 24/12/2023.

Abstract
Today, when a connection is established between people's lives and the application space based on the Internet of Things, it is necessary to make the platform of this new technology more secure and reliable. The applications of IoT have an urgent need for security issues such as trust, and many attacks can easily target the sensor nodes. To achieve this goal, we chose the RPL protocol due to its wide application and weak security. We investigated it using an innovative method of penetration testing. In this research, an application-based selective forwarding attack has been implemented. In the literature, criteria such as the number of sent packets, the amount of remaining energy, and the packet discard rate were used to detect the attacker but in this solution, the signal strength indicator is used to detect the attacker, and the parameters of positive and negative behaviour are used to calculate the trust in the beta function. In this study, trust is calculated based on the application, and the attacking node reduces the signal strength instead of increasing it.The simulation results show that the proposed method has a 99% attack detection rate, less than 11% FNR while improving the packet delivery rate.

Keywords
Internet of Things (IoT), Routing Protocol for Low-Power and Lossy Networks (RPL), selective forwarding attack, energy, Received Signal Strength Indicator (RSSI).

1. Introduction

The IETF group developed a Routing Protocol for LLN (RPL) over Low-Power and Lossy Networks (ROLL) in 2012 that supports the behavior of IPv6 and the mechanism of 6LoWPAN standard [1]. RPL is a distance-vector routing protocol based on IP version 6, which forms its network route information using a set of directed acyclic graphs. When RPL starts, it builds an inverted tree-like topology called Destination Oriented Directed Acyclic Graphs (DODAG). A DODAG consists of sender sensor nodes and receiver sink node(s). The DODAG is uniquely assigned with DODAG IP and RPL instance ID [1]. The creation of the RPL network topology is maintained with five control messages, which are [1]:

1. DODAG Information Object (DIO) allows the construction of upward routing *in* which other nodes (non-root/sink nodes) can discover the root node (RPL instance) and join it as their parent node.
2. DODAG Information Solicitation (DIS) allows the construction of downward routing for soliciting DIO from RPL node and for neighbour node discovery.
3. Destination Advertisement Object (DAO) that allows broadcasting destination information up along DODAG and allows a node to join as a child to DODAG root or DAO parent.
4. DAO Acknowledgment (DAO-ACK) is a unicast acknowledgment packet message sent by DAO recipient as a response to the DAO message.
5. Consistency Check (CC) is used to check the count of secure messages and issue challenge-response messages for security.

Table 1 provides a summary of some RPL attacks on confidentiality, integrity, and availability along with countermeasures.

Table 1. Summary of RPL attacks and countermeasures.

Attack	*Classification of attacks*	*Impact on network performance*	*Attack remedial protocols*
Rank	*Confidentiality, Integrity attack*	*Low packet delivery rate and packet delay*	*Using solutions based on intrusion detection system [2]*
Selective Forwarding	*Confidentiality, Integrity attack*	*disruption of the routing path*	*heartbeat protocol [3]*
Sinkhole	*Confidentiality, Integrity attack*	*Capturing a lot of traffic passing through the attacker's node*	*Rank authentication method [4]*

Trust modelling is the practice of using trust in evaluating a system. A summary of some trust models is presented in

Table 2 based on their classification, characteristics, and weaknesses.

Table 2. Summary of trust models for secure routing in sensor networks.

Trust models	*Description*
Bayesian trust model	*This model uses Bayes theorem in reaching the truth of a value using probability distribution. This method states how the subjective degree of trust must realistically change in order to be considered as evidence [5].*
Entropy trust model	*This method considers communication data among nodes and is based on probability distribution. This method considers a set of all trust values of all nodes and calculates their values using probability distribution. From those values, it considers the value with the highest information entropy (trust) and this value is used as trust to decide the best path [6].*

2. Related Works

The truth is that the devices in the Internet of Things network are heterogeneous and many of them have limited resources and their global connection has made securing the Internet of Things challenging. There are many approaches have been proposed to protect IoT networks from selective forwarding attacks. For instance, Linus et al proposed the Heartbeat protocol, which detects selective forwarding attacks based on the reply received from the node, but the heartbeat protocol will work only when IPsec is used [8]. Several routing metrics have been introduced in the literature to represent unique node and link characteristics of wireless sensor networks, including the number of hops, expected transmission number, expected transmission time, link quality level, received signal strength indicator, and residual energy.

The attack that we are investigating in this research and we are looking for a solution to discover and fix it is a selective forwarding attack. To detect selective forwarding attacks, reduce and defend against this type of attack, strategies such as watchdog [9], trust mechanism, anomaly detection on sensors [10], neighbour-based monitoring [8], Acknowledgment monitoring [11], Packet drop reporting [8], failure detection framework [12], etc.

In [13], a hybrid intrusion detection system (IDS) called SVELTE is proposed to detect selective forwarding and sinkhole attacks. SVELTE works efficiently when 6mapper is prepared like when RPL network is configured, but in network with duty cycling (radio is mostly off), energy overhead increase with the number of nodes increases in the network [8]. In [14], a new IDS against sinkhole and selective forwarding attacks called Detection of Sinkhole And Selective Forwarding for Supporting Secure routing for Internet of Things (THATACHI) is proposed. THATACHI uses watchdog, reputation and trust strategies. In THATACHI approach, a low false-positive rate was noticed for low-power and low resource devices. The limitation of THATACHI IDS is that it performs well only in sinkhole and selective forwarding attacks. In [15], a trust-based and secure RPL routing protocol (SecTrust) against black hole and selective forwarding attacks is proposed. In this approach, a trust-based mechanism is included in RPL. The working method is such that a comparison is made between the nodes based on the expected transmission count and the rank of the nodes. The amount of trust for each node is calculated based on the number of sent and received packets. A threshold is considered for the number of expected transmission count and the rank, and any node that meets these two limits, its trust value is compared with the rest of the nodes, and the node with a higher trust value is selected as the parent. SecTrust is a composition of five systemic processes that operate in unison to provide secure route information among IoT nodes. V. Neerugatti et al in [16] proposed AI-based technique to detect selective forwarding attack in IoT. The detection technique based on artificial intelligence in this article is called artificial intelligence-based packet drop ratio (AIPDR). This technique works based on the packet drop rate (PDR) feature. The PDR will be calculated for every node in the destination-oriented directed acyclic graph (DODAG). PDR can be negative, zero or positive. If a node's PDR has a value other than zero, that node is considered an attacker. In [15] and [16], normal packet loss rates due to noise, congestion, and environmental obstacles are not taken into account so the false positive rate (FPR) increases and attack detection precision decreases. There are many works on node energy balance and effective improvement of the low-power network, which use energy or other parameters such as the number of hops, or expected number of transmissions, in combination with routing parameters. It can balance the energy and extend the network lifetime effectively [17]. The first technique to reduce the number of hops is to use multiple sinks. It was shown in [17] that when the number of sinks increases, the average number of hops that a data packet must travel decreases, therefore, the average energy cost also decreases. The effect of the number of sinks and mobile sink on the sensor network lifetime and energy consumption and delay is shown in [18]. In [19], a novel hierarchical trust-based mechanism “CTrust-RPL” is introduced which evaluates the trust of nodes based on their forwarding behaviors. This study ships complex trust-related computations to the higher layer, known as the controller, to save computational, storage, and energy resources at the node level.

The proposed method does not use clustering of the nodes. Also, we do not assume the structural properties of the network.

3. BETA probability distribution function

In this section, a brief introduction to beta distribution is presented, which is the security foundation of our work. In statistics and probability theory, the beta distribution is a type of continuous probability distribution that is defined on the interval of $[0, 1]$ and has two parameters called a, and b. When interaction happens between two nodes, there are two states. For the trust of communication,

states refer to data transfer, which includes cooperative and non-cooperative states. For data trust, states refer to the collection and aggregation of data, including correct transmission and incorrect transmission. Therefore, the binomial distribution can be used to simulate the interaction between two nodes and hence can be used to simulate the trust distribution [20]. The two parameters a and b, which are expressed by the gamma function [20] are defined in (1):

$$P(x)=\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}x^{a-1}(1-x)^{b-1} \quad \forall\, 0\le x\le 1, a\ge 0, b\ge 0 \qquad (1)$$

Suppose there are $(a+b)$ times interactions between the nodes .In terms of communication, we assume that a is the number of positive behaviors observed from the neighbouring node and b is the number of its negative behaviors. In this case, the number of interactions between nodes is equal to the sum of positive and negative interactions between them. The trust value is obtained based on the beta probability distribution function [20] as presented in (2):

$$DT_{ij}=E\left(R_{ij}\right)=\frac{a+1}{a+b+2} \qquad (2)$$

DT_{ij} indicates Direct Trust evaluation. $E\left(R_{ij}\right)$ indicates statistical expectation of reputation function and R_{ij} indicates reputation of node i to node j. In terms of data trust, a indicates the number of normal data transmissions and b indicates the number of error data transmissions [20]. The amount of trust according to this formula is a number between 0 and 1, where 1 means the most trust and 0 means the least trust. Calculating trust with the beta probability distribution function has three advantages that make it a good choice for use on nodes with limited IoT resources:

1) High efficiency and simplicity and low calculation volume, which is very useful for devices with limited resources (especially energy).
2) The slow growth of trust due to positive behaviors and the rapid decline of trust due to negative behaviors is what is expected from a trust function.
3) Normalization of trust number between 0 and 1.

4. proposed method

The proposed method is a trust-based method that can detect selective forwarding attacks in a completely distributed manner. This method detects and isolates routing attacks. This method calculates and evaluates the trust behavior of a node.

4.1. Threat Method

In the conducted evaluations, each attacker discards all the packets related to one or more applications but passes the packets related to other applications without making any changes. As is common in the real world, attackers in the network first try to attract the maximum number of victims through the Rank attack so that they can attract the nodes around them as their children in the network, then they start the attack. According to Fig. 1, suppose node A is the attacker node which is the parent of the healthy node I. The parent of node A is node J. According to the figure, the distance between the attacker and its parent is greater than the distance between the attacker and its child. For each packet P that node I directs to its parent (node A), instead of normally forwarding the packet P to its parent node (node J), node A forwards packet P to node J with low signal strength so that Node I can listen to it but the packet does not reach the node J. After receiving each incoming packet, the attacker sends that packet to its parent with low signal strength, so that node I can listen to that packet but node J cannot receive the packet. According to Fig. 1, in step 1, node I sends the packet P related to application $a1$ to its parent, i.e. node A. In step 2, the attacker node A tries to reduce the signal strength so that node I listens to the packet, but node J does not receive the packet or receives it incompletely. In step 3, and after listening, node I thinks that its packet has been sent to node J, but in reality, its packet has not reached its destination or has arrived incompletely. According to the $RFC\ 6550$ standard, which is the RPL protocol standard, network nodes are fixed and are not moving. In every node, for each application that operates in the application layer, and for each packet sent from that application, a number agreed upon in the entire network (determined by the network manager) is placed in the auxiliary security header field of frames header as the identifier of that application. A number is assigned to each application and they are marked with the symbol $\{a1, a2\ and\ldots..\}$.

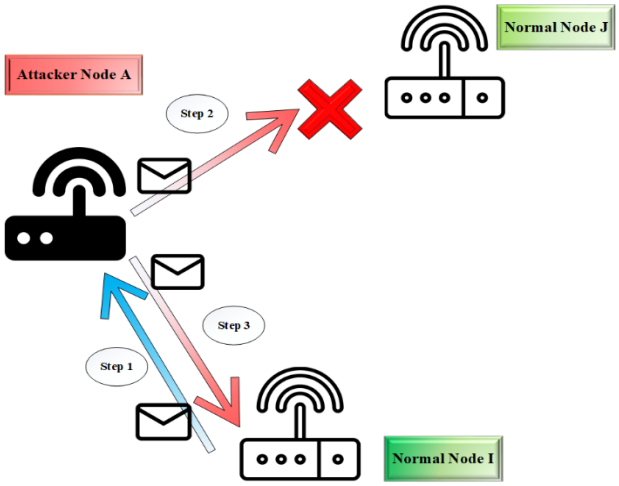

Fig. 1. Threat Method

4.2. Proposed method

The proposed detection method is a trust-based and distributed algorithm that can detect the occurrence of an attack as well as the attacker node without a central controller. In the proposed method, each child node is responsible for monitoring the behavior of its selected parent for directed packets, so that if an attack occurs, it blocks the parent node and neutralizes the attack. Each node in the network stores the average signal strength of incoming packets from its neighbours along with other information for each neighbour in one byte; these values are used in the algorithm to detect the attack. Here the proposed method is explained in detail. By directing every packet P related to the application $a1$ to its parent (node A), node I goes to the promiscuous mode of its network card to monitor the behavior of node A for the packet P.

Node I waits for a maximum of t seconds in the promiscuous mode of its network card and checks all received packets:

1) If node A forwards the packet P to its parent node without changing it, node I compares the Received Signal Strength Indicator (RSSI) value of this packet with the average value of the Received Signal Strength Indicator stored for node A. It checks the difference between these two values was greater than the value of e (e is determined according to the standard and accuracy of the network hardware in calculating the Received Signal Strength Indicator), according to (3) and (4):

$$|AVGRSSI_A - RSSINEW_A| > e \quad (3)$$

And

$$RSSINEW_A < AVGRSSI_A \quad (4)$$

$AVGRSSI_A$ indicates average RSSI value of node A and $RSSINEW_A$ indicates new RSSI value of node A. According to (5) node I considers a negative behavior for node A for application $a1$:

$$N_A^{NEW}[a_1] = N_A^{OLD}[a_1] + 1 \quad (5)$$

where $N_A^{OLD}[a_1]$ indicates the number of negative behaviors of node A according to previous interactions. $N_A^{NEW}[a_1]$ is the number of negative behaviors of node A, which is updated according to the new interaction. Otherwise, if the difference between these two values was less than the value of e, according to (6), the amount of positive behavior of node A towards the packets of application $a1$ increases:

$$P_A^{NEW}[a_1] = P_A^{OLD}[a_1] + 1 \quad (6)$$

Where $P_A^{OLD}[a_1]$ indicates the number of positive behaviours of node A according to previous interactions. $P_A^{NEW}[a_1]$ is the number of positive behaviours of node A, which is updated according to the new interaction.

2) If node A forwards packet P with modifications to its parent node or does not forward packet P at all, the number of negative behaviors of node A for application $a1$ increases according to (7):

$$N_A^{NEW}[a_1] = N_A^{OLD}[a_1] + 1 \quad (7)$$

Now, the trust value of node A in application $a1$ is calculated based on the beta probability distribution function as presented in (8):

$$T_A[a_1] = \frac{P_A[a_1] + 1}{P_A[a_1] + N_A[a_1] + 2} \quad (8)$$

Where $T_A[a_1]$ indicates trust value of node A for application $a1$ and $N_A[a1]$ is the number of negative behaviours of node A towards the packets of application $a1$ and $P_A[a1]$ is the number of positive behaviours of node A towards the packets of application $a1$. Equations 5 to 7 are used to count the positive and negative behaviors of each node. The value of the parameters of equations 5 to 7 shows the history of positive and negative interactions of each node. Positive and negative interactions are measured with the signal strength indicator. If the signal strength level difference over time exceeds a certain limit, it is considered a negative behavior, otherwise it is considered a positive behavior. According to equation 8, a node that has more positive behaviors will have more trust, and a node that has more negative behaviors will have less trust. According to equations 5 to 8, a suitable trust value is obtained from the amount of interactions of each node, which causes a high detection rate. To detect the occurrence of an attack, according to (9), if the trust value of node A is lower than the attack detection threshold parameter (thr), then node A is considered an attacker. Therefore, node I blocks the IP address of node A and chooses a new parent for itself.

$$T_A[a_1] < thr \quad (9)$$

In cases where the distance between the attacker and its parent is not greater than the distance between the attacker and its child, If the attacker sends the packet with a weaker signal strength so that it does not reach its parent, this will cause the attacker's child node to not be able to listen to the sending of the packet and the confirmation of sending the packet will not reach the child node. As a result, not listening to the confirmation of sending the packet is considered a negative behavior and the counter of negative behaviors increases. But if the reduction of the signal strength does not occur and the packet is sent normally, the child node listens for the confirmation of sending the packet, and the counter of positive behaviors increases. Fig. 2 shows the diagram of the proposed detection method.

4.3. The value of the parameters

Ideally, when there is no attacker, all network packets reach their destination successfully but in the real world, the situation is not ideal. In the Internet of Things, due to the need of price reducing, the network hardware used in the devices is not 100% optimal. Therefore, the loss of some packets in the network lies in the essence of these networks. In the real world, the value of the Received Signal Strength Indicator is also not completely reliable, because environmental influences (such as moving an obstacle between the transmitter and the receiver node) can change the value of the Received Signal Strength Indicator of the packets. Apart from it, almost all network hardware also has an inherent error in determining the value of the received signal strength indicator. For example, the network hardware of $Tmote - Sky$ nodes, which is called $CC - 2420$, can have an error of ± 6 in the declared value of the received signal strength indicator [21].We include this error range of the received signal strength indicator in parameter e when comparing the average value of RSSI and new RSSI in the proposed method. Therefore, for the $Tmote - Sky$ hardware that we use in the simulations, the value of parameter e is considered equal to 6. The RSSI unit is $ecibel\ milliwatts$, which is represented by the symbol dBm. The range of signal strength is from $-100\ dBm$ to $0\ dBm$ [21], where the signal strength of $0\ dBm$ means the strongest signals.

As an example, the average RSSI of the attacker node is $-17\ dBm$. When the attacker node launches the attack, it reduces its signal strength and sends multi-application packets with a signal strength of $-25\ dBm$. According to equation (3), we have $|-17-(-25)| = 8 > 6$, As a result, the difference between the average RSSI and the new RSSI is higher than usual. So, the attacker node has launched an attack.

In real environments, a GPS receiver is located on each sensor node, thereby significantly increasing the overall deployment cost. In the simulation, sensor nodes can broadcast "hello" messages using their maximum power

level. Then, each sensor node estimates its physical distance from its neighboring nodes through the power and RSSI values of the received hello messages and sends this information to the sink.

4.4 Topology and deployment scenario for evaluation

Fig. 3 shows an example of the implemented scenario. There are three types of nodes in this figure. The green nodes represent the sink nodes, the yellow nodes represent the normal nodes and the purple nodes are the attackers. In this figure, the transmission range of the attacker node 31 and the nodes within this range are shown. The choice of this type of topology and the way the nodes are arranged are due to the structure and the proposed attack method, which is shown in Fig. 3 and Fig. 4. In Fig. 3, the position of the attacker node 31 is such that two nodes are located at a higher level. Considering that the distance between nodes in horizontal and vertical directions is 30 meters, according to the Pythagorean theorem, the distance between nodes is about 42 meters diagonally. According to Fig. 3, the distance between the attacker node 31 and node 24 (candidate parent) is greater than the distance between the attacker node 31 and node 17 (its child). In this case, the proposed method of the attacker occurs. In Fig. 4, the position of the attacker node 31 is such that three nodes are located at a higher level. In Fig. 4, the distance between attacker node 31 and nodes 23 and 5 (set of candidate parents) is greater than the distance between attacker node 31 and node 29 (its child). In these cases, the proposed attacker method occurs.

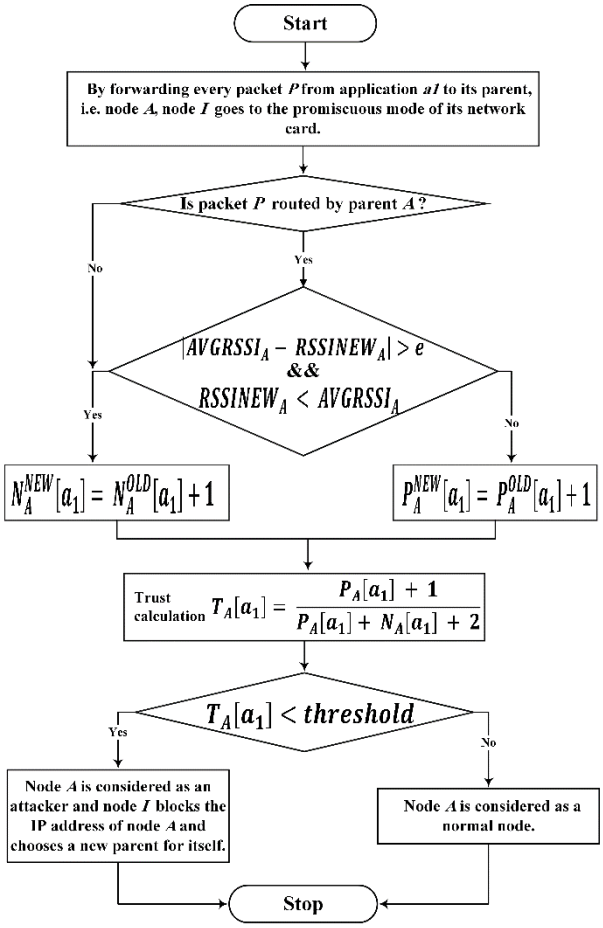

Fig. 2. The proposed method

5. Analysis and evaluation of the proposed method

The dependent parameters (evaluation criteria) considered in this research are: 1) True Positive Rate (TPR), 2) False Positive Rate (FPR), 3) Packet Delivery Rate (PDR), 4) End to End delay (EED), 5) Energy Consumption. The independent parameters considered to evaluate the proposed method are: 1) Detection threshold parameter (thr), 2) Error probability parameter when listening to packets, 3) Network size, 4) The ratio of the number of attackers in the network, 5) Network Density, 6) Number of applications, 7) The Received Signal Strength Indicator value.

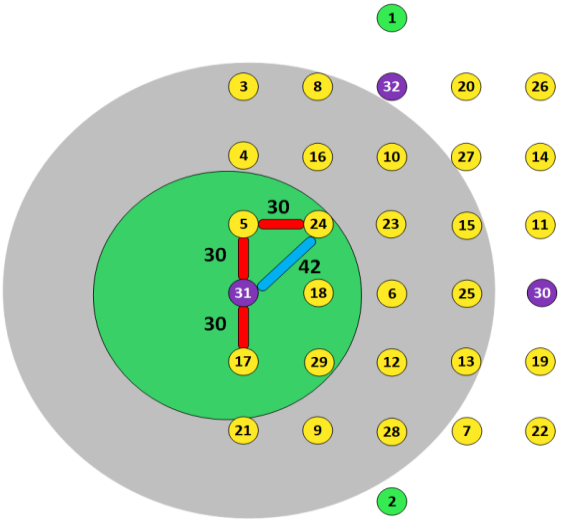

Fig. 3. An example of the network topology and showing the range of the attacker node 31

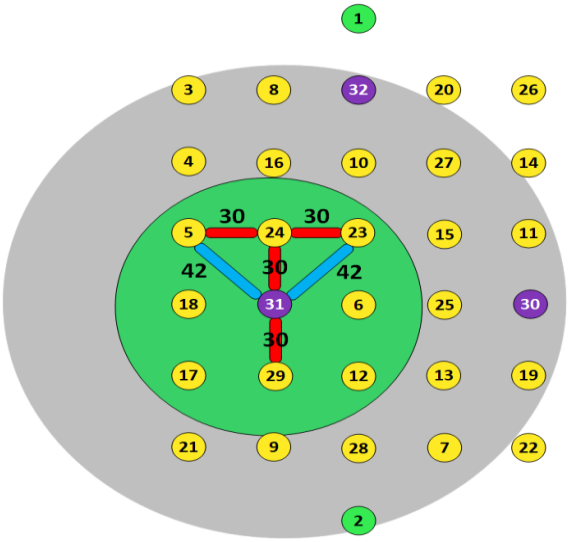

Fig. 4. An example of network topology shows the implementation of the proposed attacker method with three candidate parents

5.1. Simulator and simulation parameters

The simulations in this study were performed using the $Cooja$ emulator. It simulates the nodes with Contiki operating system and is considered the most widely used simulator in the field of IoT. The Contiki operating system is designed to work on resource-constrained sensor nodes that operate on batteries, and Contiki requires at least 10 KB of RAM and 30 KB of ROM to run. In this operating system, both IP version 4, and 6 protocol stacks along

with the RPL and 6LoWPAN protocols are fully implemented. Table 3 shows the simulation parameters. To perform simulations in the *Cooja* simulator, *Tmote Sky* hardware is used, which uses MSP430 microcontrollers and has 10 KB of RAM and 48 KB of FLASH memory. In the simulations, the UDGM model has been used as a radio communication simulation model, which considers a circular area around the nodes as their coverage area, the radius of the coverage area of each node is assumed to be 50 meters. Nodes send messages to the root alternately, and the frequency of sending packets is 60 seconds with a little time randomization to avoid collisions. The packet injection rate is considered 1packet/min. The packets sent are of UDP type and their size is 40 bytes. The detection thresholds are chosen from 0.3 to 0.6. The power trace is the function to calculate the power usage in the network simulation. Powertrace perform the process of calculate the system power consumption based on the power state tracking and also the energy capsules structure used to set the attribute energy consumption to activities such as packet transmissions and receptions. Powertrace tracks the duration of activities of a node being in each power state. There are 6 defined power states: CPU, LPM (Low Power Mode energy consumption), TRANSMIT, LISTEN, IDLE_TRANSMIT, IDLE_LISTEN.

5.2. *Examining the effect of attack detection threshold parameter* (thr)

The purpose of this section is to study the effect of the attack detection threshold parameter on the evaluation criteria. For the evaluations, networks with 30 transmitter nodes (along with two server or root nodes) are assumed, and all nodes are placed at a distance of 30 meters from each other. Ten percent (3 nodes) of the sender nodes are attackers and the probability of error during listening is 0.1, also four networks are considered with different detection thresholds. Three applications are considered and the attacker discards the packets of two applications and directs the other ones intact.

Table 3. Simulation parameters.

parameter	*value*
simulator	*Contiki Cooja*
Node type	*Sky mote*
Simulation time	*60 minutes*
Radio interface and interference model	*UDGM*
The range covered by each node	*50 m*
packet size	*40 bytes*
Frequency of sending data packets	*60 seconds*
Arrangement of nodes Layer 1 and 2	*Linear with two sinks Based on the IEEE 802.15.4 standard*

As can be seen in Fig. 5, the TPR value for the proposed method is almost equal to 1 for $thr > 0.3$ and decreases for $thr <= 0.3$, The reason for this is that when the value of the detection threshold parameter is less than 0.5, several false negative alarms (FN) have occurred in the network (many attackers have been detected later or not detected). Also, in all cases, the FPR value was almost zero. According to this figure, with the increase of thr, the value of TPR has increased. TPR (True Positive Rate) is the rate of the number of nodes that have detected the attack. FNR (False Negative Rate) is the rate of the number of nodes that have not detected an attack. Fig. 6 shows the changes of TPR and FNR according to the changes of the threshold value.

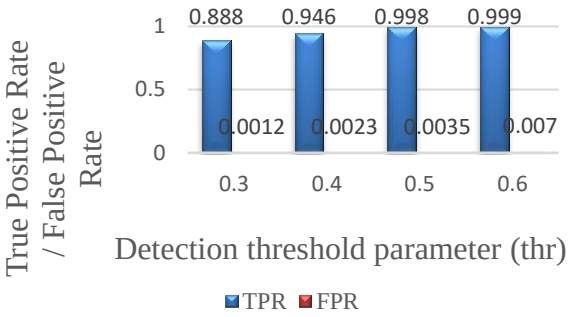

Fig. 5. Effect of attack detection threshold parameter on TPR and FPR in the proposed method

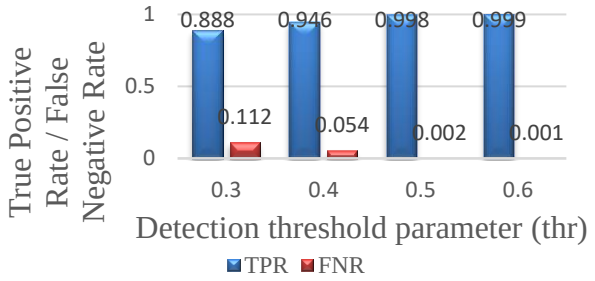

Fig. 6. Effect of attack detection threshold parameter on TPR and FNR in the proposed method

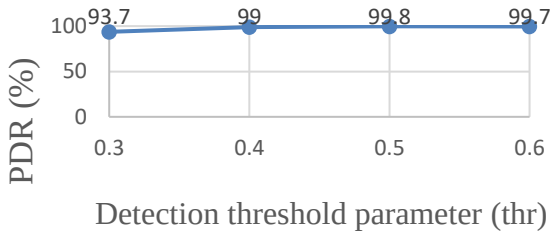

Fig. 7. Effect of attack detection threshold parameter on PDR

According to Fig. 7, with the increase of thr value, the packet delivery rate (PDR) has been slightly upward, and for thr values greater than 0.4, the PDR value is almost equal to one. According to Fig. 8, with the increase of thr, the end-to-end delay has been almost constant. Also, according to Fig. 9, with the increase of thr, the power consumption has slightly increased. According to Fig. 5, when the detection threshold parameter increases, the value of TPR and FPR also increase, as a result, when FPR increases, the number of healthy nodes that are mistakenly identified as attackers increases, so the child nodes start the process of replacing the parent, and this increases the calculations and operations of sending and receiving packets, and finally, it increases the power consumption.

5.3. *Examining the effect of the error probability parameter when listening to packets*

The settings of this section are the same as the previous section, with the difference that the thr value is set to 0.5, and networks with different values of error probabilities such as $0.1, 0.2, 0.3$ are simulated. Fig. 10 shows that even when the error probability is 30% when listening to the packets, the FPR value was less than 1%, and also the value of TPR has been almost equal to 1, at the same time, the value of FPR has ascent and the value of TPR has descent. In Fig. 10, the values of the proposed method, in addition to the basic RPL protocol, are also compared with the SVELTE approach, which is an intrusion detection system for the IoT network.

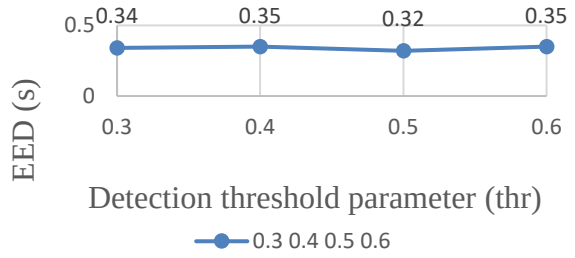

Fig. 8. Effect of attack detection threshold parameter on the end-to-end delay in the proposed method

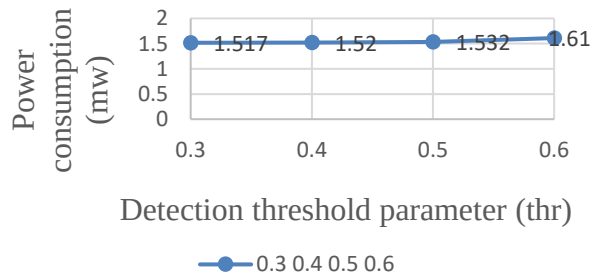

Fig. 9. The effect of attack detection threshold parameter on power consumption in the proposed method

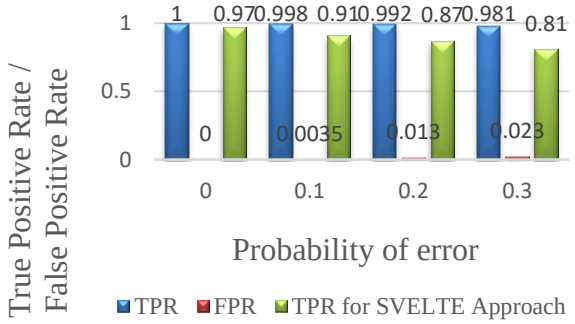

Fig. 10. Effect of error probability parameter on TPR and FPR for the proposed method

Fig. 11 shows that the PDR value for the proposed method has decreased slightly with the increase in the error probability. However, even when the error probability was 30%, the PDR value was equal to 0.973. Moreover, it can be seen that the PDR value for the RPL protocol is much lower than the proposed method, which indicates the significant effect of the attack on the RPL protocol.

5.4. *Examining the effect of network size*

In this section, the effect of the network size on evaluation criteria is analyzed. The settings of this section are the same as the previous one. In all these networks, the value of thr is 0.5. The numbers of nodes in the networks are 20, 30, 40, and 50 nodes respectively. Fig. 12 shows that with the increase in the network size, the TPR value remained equal to one, and the FPR value was slightly raised, however, the FPR value, even in a network with 50 nodes, remained less than 1%. The value of TPR in approaches such as SVELTE and THATACHI, which use the intrusion detection system and the trust solution, respectively, is close to 1, which means that these methods also detect the attacker and the attack well.

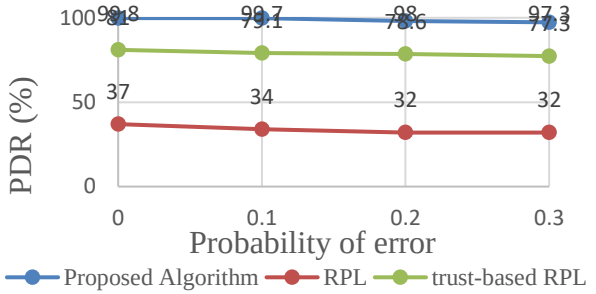

Fig. 11. Effect of error probability parameter on PDR

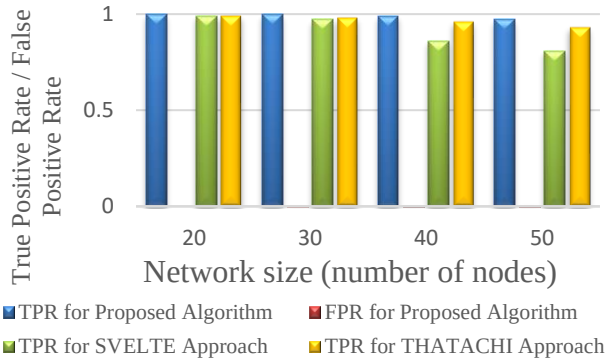

Fig. 12. Effect of network size on TPR and FPR

Fig. 13 shows that the PDR value for the proposed method remains close to 1 with the increase in the number of network nodes, also there is a significant difference between the PDR related to the RPL protocol and the PDR resulting from the proposed method. Fig. 14 shows that as the network size increases, the end-to-end delay value for the proposed method and RPL has increased because increasing the number of nodes increases the average number of steps toward the root and increases the delay in receiving packets. According to Fig. 15, the amount of power consumption for the proposed method and the RPL protocol has increased with the increase in the number of nodes in the network, and also the proposed method has a little energy overhead compared to the RPL. The proposed method consumes more power than the CTrust-RPL method, because in the proposed method, the trust mechanism is implemented at the node level, and a resource-constrained node itself does all computations, but in the CTrust-RPL method, the trust mechanism is implemented in the controller layer.

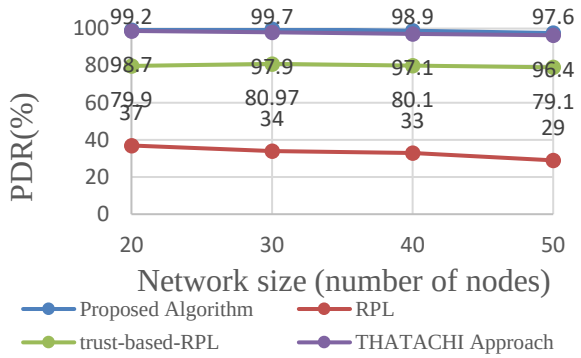

Fig. 13. Effect of network size on PDR

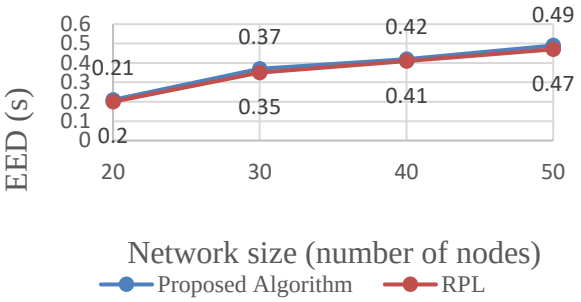

Fig. 14. Effect of network size on end-to-end delay

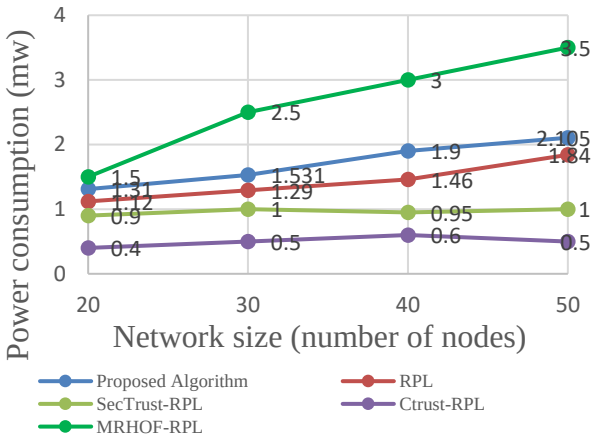

Fig. 15. Effect of network size on power consumption

5.5. Examining the effect of the ratio of the number of attackers in the network

In this section, the effect of attackers on the total node number ratio is discussed. The settings of this section are the same as in the previous section. The ratio of the number of attackers in different networks is set differently: 10% of all nodes sending information (3 nodes), 20% (6 nodes), and 30% (10 nodes). Fig. 16 shows that even with the increase in the number of attackers, the value of TPR is almost 1 and the value of FPR is almost 0. According to Fig. 17, even when 30% of the network nodes are attackers, the PDR value remains almost one. Fig. 17 shows that increasing the number of attackers reduces the PDR. This allows the nodes to change preferred parents of them more frequently which makes the network topology to be unstable and unable to make optimize routes. In Fig. 18, it can be seen that with the increase in the number of attackers in the network, the amount of the end-to-end delay for the proposed method remains almost constant. As can be seen in Fig. 19, the increase in the number of attackers in the network has not had much effect on the power consumption of the network, which indicates that the proposed method can detect and deal with a large number of attackers without reducing efficiency.

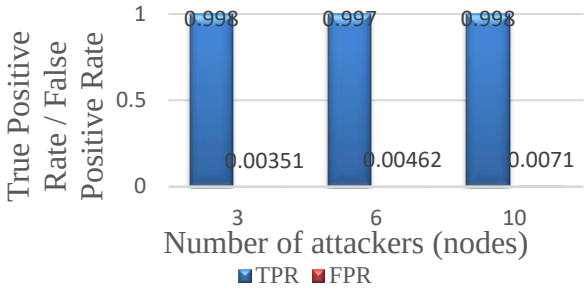

Fig. 16. Effect of number of attackers on TPR and FPR

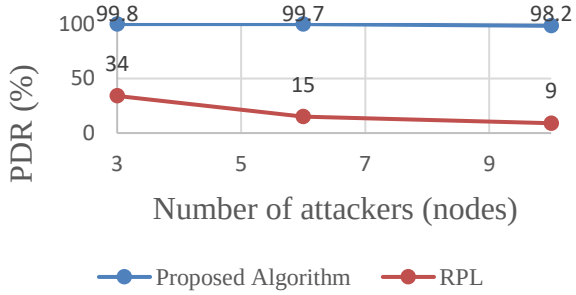

Fig. 17. Effect of number of attackers on PDR

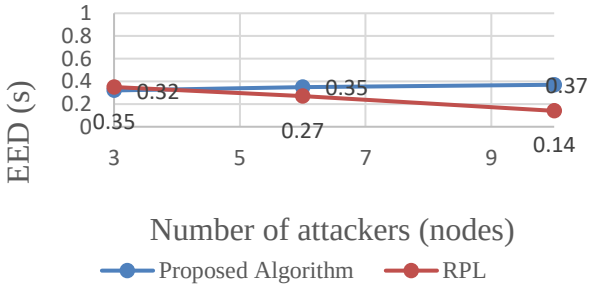

Fig. 18. Effect of number of attackers on end-to-end delay

5.6. Examining the effect of network density

According to the obtained results, by increasing the distance between the nodes, the TPR value remains close to 1 and the FPR value remains close to zero, Also, the PDR value for the proposed method remains close to one with the increase of the distance between the nodes. By increasing the distance between the nodes, it can be seen that the end-to-end delay for the proposed method and the RPL protocol is almost the same, however, the delay for the proposed method is slightly higher than the RPL protocol. By increasing the distance between nodes, the energy consumption has increased for both the proposed method and the RPL protocol. The reason for this is that the average number of steps to reach the root increases as the distance between the nodes increases.

5.7. Examining the effect of increasing the number of applications

According to the results obtained by increasing the number of applications, the TPR value remains close to 1

and the FPR value remains close to 0, Also, the PDR value for the proposed method has remained close to one with the increase of the application numbers. The amount of end-to-end delay for the proposed method has not changed much even with the increase in the number of applications. As the number of applications increases, energy consumption has a constant rate for the proposed method, which indicates that the proposed method can detect attacks with high efficiency even when the number of applications is high.

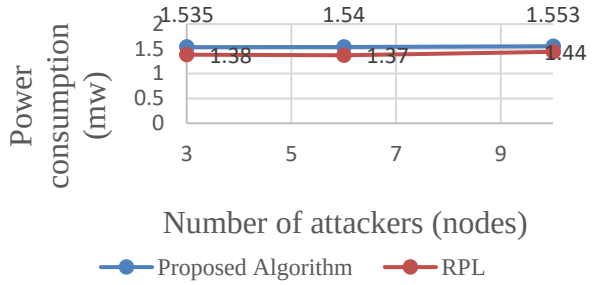

Fig. 19. Effect of number of attackers on the power consumption

5.8. Examining the effect of increasing the packet transmission rate

RPL exhibits increasing packet loss ratios (PLR) and decrease packet delivery ratio with the increase of the packet transmission rate. The simulation results show that when transmission rates become large the PDR decreases slightly. This result is mainly due to network congestion and packet collision. The proposed method consumes more energy with the increase of the packet transmission rate. This increase is due to the additional amount of data packets transmitted. Heavy data traffic causes more collisions and packets must be buffered, thus increasing the overall delay of the path to the sink. By increasing the packet transmission rate, the proposed method can correctly detect the attacker according to the calculations and workflow, and the TPR remains close to 1, but the increase in trust calculations increases the power consumption.

6. Conclusion and future work

In this research, the security vulnerabilities in the routing layer of the IoT have been investigated, and a trust based method has been proposed. A new method of forwarding attacks and attackers is implemented and investigated. In this research, a solution has been proposed to detect this new attack. In this solution, the concept of received signal strength is used to detect positive and negative behaviours. The parameters of the positive and the negative behaviours obtained in the beta function are used to obtain the trust value, and a threshold is considered for the trust value. If the trust value of a node is greater than the threshold, it acts as a normal node in the network, and if the trust value of a node is less than the threshold, it is considered an attacker. The child node decides whether a negative behaviour or a positive behaviour has occurred by using the average received signal strength and the value of the new received signal strength. If the difference between the average received signal strength and the new received signal strength exceeds a certain limit, it means that the parent (attacker) has greatly reduced its signal strength and this is a suspicious behaviour, therefore, it is considered a negative behaviour and the counter of negative behaviours increases. On the contrary, if the difference does not exceed a certain limit, it means that the parent has not changed significantly in its signal strength, therefore, this behaviour is considered a positive behaviour and the counter of positive behaviours increases. Afterward, according to the parameters of positive and negative behaviours, and using the beta function, a trust value for the node is obtained and this trust value is compared with the trust threshold value. If the trust of a node is lower than the value of the attack detection threshold parameter, that node is considered an attacker. If the node trust is higher than the value of the attack detection threshold parameter, that node is considered a normal node.

The IoT network includes elements such as sensors, cameras, GPS locators. The IoT network has many applications on a large scale, such as creating better enterprise solutions, smart homes, innovating agriculture, building smarter cities, etc. An IoT-backed security solution uses real-time data to provide mitigation tactics and prevent cybersecurity attacks. A smart home uses sensors to control and maintain lighting, resource management, and security systems. Agriculture, as an industry, could massively benefit from the Internet of Things. Sensors are used to provide details of soil chemistry and fertilizer profiles. Livestock tracking involves the use of RFID chips to keep track of an animal's vitals, vaccination details, and location. A smart city is an urban city that uses sensors and cellular or wireless technology placed in ubiquitous places such as lamp posts and antennae. Aspects of the IoT in the performance of a city: Traffic management, Pollution monitoring, Resource management, Parking solutions, etc. In the examples mentioned above, the security of the environment and nodes is very important. In the mentioned applications, attackers with a strong RSSI value may try to infiltrate the network and be selected as the parent of other nodes and disrupt the network performance. Therefore, we need a method that recognizes the abnormal difference in the RSSI value of nodes as a suspicious behavior and thus identifies the malicious node and blocks it. Although this paper has solved some problems, there are some limitations. In future work, it is possible to add a simple encryption system to this trust method. Symmetric and asymmetric encryption systems can also be investigated by considering the characteristics of the IoT environment. This encryption system can encrypt the data inside the package of different applications so that more difficult attackers can identify which packages belong to which application. The movement of nodes increases resource consumption and energy consumption because the amount of data exchanged increases. Therefore, by considering the mobility in the proposed method to compensate the energy reduction, we can increase the number of sink nodes. The mobility of the nodes causes

changes in the conditions of the distance between the nodes in the proposed method. For example, the destination node may move and the distance between the destination node and the attacker becomes equal to the distance between the source node and the attacker. In case the attacker reduces the RSSI value, the message may be delivered to the destination. As a result, in order to launch an attack, the attacker must reduce the signal value more than before so that the message does not reach the destination. In this case, the source node can detect the occurrence of suspicious behavior and attack by checking the changes of the new RSSI value and the average value of the RSSI. Operational and more detailed investigation of mobility in the proposed method can be considered as future research.

In RPL, topology change operation, parent selection and DODAG change are performed periodically. To reduce energy consumption, we can define a variable or threshold for the mentioned operation, for example $thr - o$. We can also define a threshold to determine the "fully trusted node", for example $thr - f$. After the above-mentioned operations have been performed up to the desired threshold ($thr - o$), we introduce the nodes whose trust value is higher than a threshold ($thr - f$) in all these periods as "fully trusted nodes" in the network. In the future periods of the mentioned operations, to reduce energy consumption, the trust calculation will not be performed for fully trusted nodes and their previous trust value will be considered. This idea can reduce the operations and calculations of the nodes, and as a result, the energy consumption of the nodes is reduced, while it does not have a negative effect on the value of TP and FN. The effectiveness of this idea can be investigated as future research.

7. References

[1] Almusaylim, Z. A., Alhumam, A., & Jhanjhi, N. Z. (2020). Proposing a secure RPL based internet of things routing protocol: a review. Ad Hoc Networks, 101, 102096.

[2] Ioulianou, P. P., Vassilakis, V. G., & Shahandashti, S. F. (2022). A trust-based intrusion detection system for RPL networks: Detecting a combination of rank and blackhole attacks. *Journal of Cybersecurity and Privacy*, *2*(1), 124-153.

[3] Ribera, E. G., Alvarez, B. M., Samuel, C., Ioulianou, P. P., & Vassilakis, V. G. (2020, July). Heartbeat-based detection of blackhole and greyhole attacks in RPL networks. In 2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP) (pp. 1-6). IEEE.

[4] Jain, A., & Jain, S. (2019). A survey on miscellaneous attacks and countermeasures for RPL routing protocol in IoT. In Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2018, Volume 3 (pp. 611-620). Springer Singapore.

[5] Wang, R., Zhang, Z., Zhang, Z., & Jia, Z. (2018). ETMRM: An energy-efficient trust management and routing mechanism for SDWSNs. Computer Networks, 139, 119-135.

[6] Nie, S. (2019). A novel trust model of dynamic optimization based on entropy method in wireless sensor networks. Cluster Computing, 22(Suppl 5), 11153-11162.

[8] Patel, A., & Jinwala, D. (2022). A reputation-based RPL protocol to detect selective forwarding attack in Internet of Things. International Journal of Communication Systems, 35(1), e5007.

[9] Zhang, Q., & Zhang, W. (2019). Accurate detection of selective forwarding attack in wireless sensor networks. International Journal of Distributed Sensor Networks, 15(1), 1550147718824008.

[10] Ioannou, C., & Vassiliou, V. (2019, May). Classifying security attacks in IoT networks using supervised learning. In 2019 15th International conference on distributed computing in sensor systems (DCOSS) (pp. 652-658). IEEE.

[11] Yu, B., & Xiao, B. (2006, April). Detecting selective forwarding attacks in wireless sensor networks. In Proceedings 20th IEEE international parallel & distributed processing symposium (pp. 8-pp). IEEE.

[12] Udhayavani, M., & Chandrasekaran, M. (2019). Design of TAREEN (trust aware routing with energy efficient network) and enactment of TARF: A trust-aware routing framework for wireless sensor networks. *Cluster Computing*, *22*(Suppl 5), 11919-11927.

[13] Raza, S., Wallgren, L., & Voigt, T. (2013). SVELTE: Real-time intrusion detection in the Internet of Things. Ad hoc networks, 11(8), 2661-2674.

[14] Santos, A. L., Cervantes, C. A., Nogueira, M., & Kantarci, B. (2019). Clustering and reliability-driven mitigation of routing attacks in massive IoT systems. Journal of Internet Services and Applications, 10, 1-17.

[15] Airehrour, D., Gutierrez, J. A., & Ray, S. K. (2019). SecTrust-RPL: A secure trust-aware RPL routing protocol for Internet of Things. Future Generation Computer Systems, 93, 860-876.

[16] Neerugatti, V., & Rama Mohan Reddy, A. (2020). Artificial intelligence-based technique for detection of selective forwarding attack in rpl-based internet of things networks. In Emerging Research in Data Engineering Systems and Computer Communications: Proceedings of CCODE 2019 (pp. 67-77). Springer Singapore.

[17] Yarinezhad, R., & Sarabi, A. (2018). Reducing delay and energy consumption in wireless sensor networks by making virtual grid infrastructure and using mobile sink. AEU-International Journal of Electronics and Communications, 84, 144-152.

[18] Yarinezhad, R. (2019). Reducing delay and prolonging the lifetime of wireless sensor network using efficient routing protocol based on mobile sink and virtual infrastructure. Ad Hoc Networks, 84, 42-55.

[19] ul Hassan, T., Asim, M., Baker, T., Hassan, J., & Tariq, N. (2021). CTrust-RPL: A control layer-based trust mechanism for supporting secure routing in routing protocol for low power and lossy networks-based Internet of Things applications. Transactions on Emerging Telecommunications Technologies, 32(3), e4224.

[20] Fang, W., Zhang, C., Shi, Z., Zhao, Q., & Shan, L. (2016). BTRES: Beta-based trust and reputation evaluation system for wireless sensor networks. Journal of Network and Computer Applications, 59, 88-94.

[21] Bauer, J., & Aschenbruck, N. (2020). Towards a low-cost rssi-based crop monitoring. ACM Transactions on Internet of Things, 1(4), 1-26.