Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Selfish Node detection based on GA and Learning Automata in IoT

Solmaz Nobahary
science and research branch Islamic Azad University https://orcid.org/0000-0001-9233-7978
Hossein Gharaee Garakani (gharaee@itrc.ac.ir)
ITRC
Ahmad Khademzadeh
ITRC
Amir Masoud Rahmani
science and research branch Tehran AZAD university

Research

Keywords: Internet of Thing (IoT), selfish node, Genetic Algorithm (GA), Learning Automata (LA)

Posted Date: May 7th, 2020

DOI: https://doi.org/10.21203/rs.3.rs-22412/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Selfish Node detection based on GA and Learning Automata in IoT

Solmaz Nobahary
Computer engineering department, Science and Research Branch, Islamic Azad University name of organization
Tehran, Iran
solmaz.nobahary@srbiau.ac.ir

Hossein Gharaee*
Garakani
Iran Telecom Research Center ITRC
Tehran, Iran
gharaee@itrc.ac.ir
+989121329781

Ahmad Khademzadeh
Iran Telecom Research Center ITRC
Tehran, Iran
a.khademzadeh@itrc.ac.ir

Amir Masoud Rahmani
Department of Electrical and Computer Engineering Science and Research Branch, Islamic Azad University
Tehran, Iran
rahmani@srbiau.ac.ir

Abstract: It is critical to increasing the network throughput on the internet of things with short-range nodes. Nodes prevent to cooperate with other nodes in the network are known as selfish nodes. Previous studies have done on the selfish nodes detection that leads to increase throughput and reduce the end to end delay. The proposed method for discovering the selfish node is based on genetic algorithm and learning automata. It consists of three phases of setup and clustering, the best routing selection based on genetic algorithm, and finally, the learning and update phase. For appropriate network performance, the clustering algorithm implemented in the first phase. Nodes are working together to send the data packet to the destination in the second phase, and the neighbor node selected for forwarding the data packet in which that node has a high value of fitness function, among others. In the third phase, each node monitors the performance of its neighbor nodes in forwarding the data packet and uses the learning automata system to identify the selfish nodes. By preventing to cooperate selfish nodes and decreasing the probability selection of selfish nodes, it increases the throughput in the network. The results of the simulation show that the detection accuracy of selfish nodes in comparison with the existing methods average 12 %, and the false positive rate has decreased by 5 %.
Keywords: Internet of Thing (IoT), selfish node, Genetic Algorithm (GA), Learning Automata (LA).

1- Introduction

The modern world and communications technology have shown in the business world, people who have access to data and better information will control the future. The internet of Things (IoT) is indeed the product of coverage and evolution of three internet elements, wireless technology, and Micro-electronic and mechanical devices [1, 2]. Connectivity to the internet in the IoT has led to its application of all aspects of human life, including smart cities, smart water control environment, Security and emergencies, smart transportation, smart agriculture, industrial and health control, and so on. Wireless communication is a way of sending and receiving data in IoT. However, the low range of the wireless connections requires multi-step communication is based on the cooperation of all thing in IoT [3, 4].

One of the most critical challenges in the IoT presents is the lack of cooperation in some nodes to the data transmission in multi-step communications. It is called "selfish nodes" that have the ability to increase its lifetime and the rest of the nodes in cooperation with them but to achieve the maximum preferences and misuse of the other nodes. By increasing the number of selfish nodes, the average end-to-end delay is increased in the data packets and network traffic. It is generally impaired by network functionality and practicality [5-7].

To overcome the effects of selfish nodes, identifying and dealing with them is of particular importance. Different methods have been proposed for this purpose, which can be classified into several categories according to their functional nature. The reputation based methods provide feedback from certain nodes behavior that nodes cooperate or not. These methods have low throughput and detection accuracy and high energy consumption [8-11]. Credit based approaches trade data packets between nodes in the business network; the lack of encouragement and punishment of cooperation and selfish nodes is the disadvantage of these categories [13-15].

Acknowledgment messages are sent from destination nodes in acknowledgment based methods in the different approaches presented in this group due to the transmission of authentication packets, network traffic, and the average end to end delay to the result in increased data packets and network efficiency have decreased [16-18]. In the game

theory methods, each node plays a role as players in the game, which are interacting with each other and design the game and its profits to send the data packets. These methods have a higher false positive rate, and the average end to end delay is more than other methods [19-21].
Another group of selfish nodes detection and recognition methods in the network is hybrid and specified methods. These methods benefit from the advantages of different techniques. Because of this, they have a high accuracy of detection and low traffic, the average end to end delay, compared with other methods [22-26].
The proposed method is a multi-phase method based on GA and LA for the detection of the selfish nodes in IoT, and the scheme has been proposed in three phases, including setup and clustering, suitable path selection based on genetic algorithm, and the learning and update phase. In the first step, a set of things is clustered based on having communicated with the destination base station, and the cluster head chose the data packets to the base station. In the second phase, the chromosome selected by a genetic algorithm has shown the best route to send the data packets to the destination. The chromosome has a high value of fitness function, among others. In the third phase, the result of the sending operation evaluates by the source node is equipped LA. The source node updates the probability of LA about the neighbor nodes by receiving the acknowledgment. Each node whose probability value is less than the predetermined threshold in LA is detected as a selfish node. In the following, the main contributions of our paper are presented:

- Our mechanism is a hybrid method that takes advantage of using a genetic algorithm and learning automata to detect selfish nodes. All nodes are equipped by LA to learn about the status of the neighbor nodes. The nodes with low probability can't be active and send the data packet, so the node known as a selfish node. The nodes want to have high probability should cooperate with other nodes, and it means stimulation of the nodes.
- If the probability of the neighbor node of the spatial node is not less than the threshold, the mentioned node is provided the opportunity of cooperating with other nodes.
- We propose a multi-phase method to detect selfish nodes based on GA and LA in each round and prevent resending the data packets to selfish nodes. Then, it has low energy consumption. The metrics of throughput won't decrease, due to the source node does not need to resend the data packets that might be needed as a failure of the data packets to reach the destination. Network traffic will be decreased by not repeatedly sending the data packets, and it leads to a decrease in the average end-to-end delay of the data packets in the network. The proposed scheme has a low false positive/negative rate and high detection accuracy of selfish nodes.
- The different theoretical metrics are evaluated by executing several simulations. Results have shown significant improvement in distinct metrics.

The rest of the paper is as the following: The method is presented in section 2 and related works are presented in section 3. The selfish node detection mechanism phases is described in the next section. In section 5, the proposed method is simulated, and results are evaluated. And the conclusion is presented in section 6.

2- **Methods**

This paper presents the problem of detecting selfish nodes in the Internet of Things networks that the selfish nodes are using network facilities for their personal uses. These nodes do not provide any help in saving energy consumption and maintaining communication with other nodes as they do not participate in routing and forwarding operations of packets. As a result, network throughput is significantly reduced in the presence of selfish nodes.

Figure 1. Selfish node detection mechanism

In this paper, we design a detection and discovery mechanism based on genetic and learner automata in the IoT network, which represents three blocks of the diagram by the proposed protocol. We consider a multi-phase scenario, where there are base stations in each cluster have collected data packets from cluster member nodes (setup & clustering). In the second phase (The best routing selection phase based on a genetic algorithm), the nodes choose the best route for purpose while forwarding their data packets and cooperating to get the data to the base station. The best-selected route has fitness value higher than other neighbor nodes routs. In the third phase (learning and updating phase), to address and determine the selfish nodes who not forward the data packets at all. If the acknowledgment packets receive from the destination, the probability of a neighbor node in the selected chromosome will be increased in LA. But if not receive, it will be decreased. At the end of the round, the probability of the node is less than the predefined threshold; it will be known as a selfish node. We use the cooperation process analysis to identify selfish nodes and propose a genetic-based mechanism that utilizes the learner automata mechanism. While nodes are detected on abuse, we reduce the power transmission of data packing (or even cooperation with other nodes). Figure 1 shows the block diagram of the multi-phase mechanism.

We stimulate the nodes to cooperate with them in a selfish node. Finally, the main task of this paper can be summarized as follows:

- The multi-phase based on genetic and learning automata method has been proposed. According to the features such as node distance to cluster and the level of energy by a fitness function is calculated for the selected neighbor node.
- A strategy of detection and discovery based on genetic algorithm and learning automata is proposed to make cooperation the nodes in IoT. A learning strategy is introduced to address the challenge of selfish behavior. The main idea is that every node in a certain period reviews the responses of other things, and the nodes don't forward the data packets are applied as a selfish node in LA. By theoretical analysis, we demonstrate that by using this strategy, selfish nodes are significantly identified with high precision because any deviation from the level of uncertainty of the data packet tends to low cooperation or even non-cooperation, which leads to isolating of the selfish node.
- At certain times, the cluster head will ask members about the status of each neighbor who has stored in their neighborhood table. Nodes possessing the level of their cooperation in LA of each node are reported as selfish nodes, and the nodes have the largest report by cluster head and its neighbors known as a selfish node. That is reported to all nodes in the cluster with the message of all broadcasts to encourage them to cooperate with others. The results of the simulation show that the proposed strategy with high precision is to reduce network throughput due to the existence of the selfish nodes in the network.
- The detection method has been simulated in MATLAB to evaluate the network. The simulation results has shown the high detection accuracy, low false positive/negative rate. Energy consumption is reduced by using the detection method.

Each of the proposed mechanism phases is presented in detail by following.

3- Related work

Several approaches have been developed to discover and deal with selfish nodes, stimulated them to cooperate with other nodes in the network. These approaches, according to their nature, are divided into six groups known as reputation-based approaches, credit-based approaches, punishment-based approaches, acknowledgment-based approaches, game theory approaches, and hybrid and specified approaches. It categorized the methods to incentive protocols and identifying, isolating selfish nodes protocols; then, it expressed the weakness and strength of techniques in each group [8-9].

In reputation-based methods, network nodes cooperate with each other to provide feedback for a set of particular nodes. Each node is assigned a reputation value with respect to its feedback [10]. An intelligent reputation-based approach called the Separation of Detection Authority (SDA) is designed to detect selfish nodes in the network. Unlike previous approaches in this approach, the reliability of the network is also considered. This approach is based on a central organization to recognize the credit of the nodes, which consists of three sections of reporters, agents, and a central authority. In this approach, when a node observes suspicious behavior from its neighboring node(s), it introduces itself as a reporter to the central authority. Then the central authority assigns nodes to the neighboring suspect nodes as agents to determine the behavior of the suspect nodes and determine whether the node is suspicious forwards the data packets or not. After observing a period, each node sends the results of its observations to the central authority [11].

An approach is proposed to detect selfish nodes and stimulate them to cooperate with the network [12]. The proposed method uses a control data packet to detect the selfish node. So that when the data packet is sent from the source node to the destination node when the data packet reaches the intermediate node as a selfish node, then the data packet will not be sent by this node, and due to not receiving the control packet from the destination node, the source node will retransmit the packet data, and the number of retransmissions will increase. If the number of retransmission packets is more than the predetermined threshold, then the network will have a selfish node. The self-node is detected by listening to the channel of other nodes.

In credit-based or virtual-based methods, the nodes that have a data packet to send are paying for it, or the nodes trade their data packets between themselves and sell it at a higher price after buying a packet [13, 14]. A credit-based method is proposed to detect selfish nodes in a MANET [15]. The algorithm is clustered the network nodes and selected the cluster head and watchdog nodes. The cluster head nodes control the network feature of cluster member nodes such as traffic, delay, throughput, etc. But the watchdog nodes monitor the nodes in the clusters and report the selfish nodes which aren't forward the packets to the cluster head. When the cluster head finds abnormally behavior in the member nodes, it will call the watchdogs to monitor the nodes. The disadvantages of the method are high latency and communication overhead.

In acknowledgment-based methods, it ensures that sending a packet to a node using an acknowledgment message. In these methods, a node sends an acknowledgment message to the source node when it wants to forward the packet. If a source node does not receive an acknowledgment message, it is taken as a misbehavior node [16, 17]. In 2018, Mahdi Bounouni et al. proposed an acknowledgment-based method to discover malicious and selfish nodes [18]. The proposed approach consists of four models for punishing malicious nodes and stimulating selfish nodes to cooperate with other nodes. The monitoring model is responsible for controlling the sending of routing packets and data packets by using the acknowledgment packet in the network. The reputation model, which evaluates each nodes' neighbors, by sharing the nodes' reputation between each other and according to the rules of trust, for this purpose, three types of direct, indirect, and general reputation are defined and fulfilled. Stimulator model manages and updates nodes' credit accounts that this module is intended to stimulate nodes by cooperating to send routing and data packets, they can increase their credit account balance and improve their reputation among neighboring nodes, and finally, malicious and selfish nodes are punished by isolator model whose reputation is lower than the threshold. The proposed method has high overhead and unable to detect collision attacks and selective forwarding misbehavior.

Game theory is an applied mathematical theory, it models and analyzes systems in which each person tries to find the best strategy that has been chosen by others to find success [19].

C. Vijayakumaran et al. proposed a novel detection of the selfish node, which consists of two phases: 'Generation Phase' and 'Verification Phase.' The generation phase also includes routing task confirmation step and the routing-report generation step, and coordination-confirmation report generation step. The routing task confirmation step is done when the source node is routed to the destination node by using the DSR method. A new routing task is assigned by the middle relay node to the new node, and this assignment confirmation should be created for it, which is assumed by the hash function as a signature function by the supervisor in the verification phase [20].

The proposed mechanism is a multi-phase method based on reputation and game theory for stimulation of cooperation between selfish nodes in the internet of things, and this mechanism has been designed in three phases including setup and clustering, sending data and playing a multi-person game, and update and detecting selfish and malicious nodes.[21]

In hybrid methods, the methods use credit-based or reputation-based or other groups of methods to provide the benefit of the hybrid methods [22]. TEEM is a trust-based approach to detect malicious and selfish nodes in mobile ad hoc networks and wireless sensor networks, which is usually dependent on the watchdog approach, although such monitoring devices have more energy consumption. This method is based on the time division of the monitoring strategy to achieve high-security levels. This method includes both the trust and the link duration between the true cooperation pairs relative to the diving period of the monitoring, which is completely distributed by switching Hello messages between the nodes. In TEEM, network nodes are commonly monitored from the beginning. After that, the task of network monitoring will be distributed among the trusting pairs. Hence, they can store their energy power over other nodes [23]. This paper proposes to detect selfish nodes in IoT (DISOT) in three phases: Setup and Clustering phase, which identifies and then clusters all the nodes in the network. The global phase, which indicates whether a selfish node(s) exists in the clusters or not using the main cluster head and the cluster heads in each cluster, must identify the selfish node(s) within the local phase [24].

The main responsibility of the payment punishment scheme (PPS) involves three steps sending the data packets, monitoring other nodes, and reporting them. The encouragements and punishments considered in this approach for nodes make them cooperate with each other. The method has clustered the nodes and used three watchdogs to monitor the nodes. The cluster head applies the modified Extended Dempster-Shafer model by using watchdogs to detect the selfish node. The advantages of this approach are increasing cooperation between nodes, reducing the percentage of the false alarm rate. The disadvantages are reduced performance by increasing bandwidth and high power consumption [25].

The trust management scheme has consisted of the detection and prevention steps. To detect the nodes, an algorithm that has been used is called an adaptive threshold algorithm. A repeated game is avoided the selfish node behavior. The nodes' behavior is compared in normal state and the current one. The packet forward ratio (PFR) is calculated in the current state, and it is compared by the pre- threshold. If the PFR is lower than the threshold value, the node is selfish. Otherwise, the threshold value will be set with the current PFR. In the prevention phase, the game is designed nodes to gain fewer payoffs if the nodes choose the selfish strategy; hence they are unwilling to choose this strategy [26].

All aforementioned schemes and algorithms are important and cannot be ignored; however, each of them has weaknesses in some circumstances that must be improved. Provide an efficient algorithm that detects selfish nodes in IoT; their strong points can be beneficial.

Table (1): advantages and disadvantages in different categories to detect selfish node

4. Proposed method phases

4-1 Setup and clustering phase

In the first phase, several things are randomly distributed in the desired environment of different applications. After nodes distributed in the desired area, each node has identified all its neighbors by broadcasting the hello message and saves the data packets about its status and neighbors in a table consisting of four fields, as in Figure 2 in the database.

Figure2: the data packet content in nodes' Table

In the following, more details are discussed about each field, as shown in Figure 2.

- Node_id: It has 16 bits to save the node's Identification.
- Number_neighbor_node: It has 8 bits to save the number of neighbor nodes.
- Distance_nearest_Base station: It has 16 bits to save node's distance between the nearest base station and node.
- Node_coordinate: It is an array of two elements with 16 bits in each of them to save the (x, y) coordinate of the node. The distance between nodes can calculate by the coordinate of nodes.
- Residual _energy: It has 16 bits to remain energy of the node and updated by nodes in a time period.

After identifying the neighbor nodes using the clustering algorithm introduced in 2016 for IoT, J Sathish Kumar, Mukesh A Zaveri, was proposed about it. The function of this method is that all things with each attribute are assumed to be a node, and with the nodes that the nodes share the relationships, it reduces the overhead of communication. Nodes, naturally, are heterogeneous in IoT and are connected from different networks, which also assumed the nodes heterogeneous. Clusters are varied at regular intervals and are dynamic because of the dynamic nature of the Internet of things topology. This method promises energy savings by selecting different nodes as the cluster head.

4-2 The best routing selection phase based on genetic algorithm

Genetic algorithms are an adaptive innovation search algorithm that is one of the types of developmental algorithms that have been inspired by biologists, such as mutation, selection, and crossover [27]. The high interest of these algorithms is that the final results are more significant. The genetic algorithm encoding the issue as a set of strings (chromosomes) containing tiny particles (genes), each chromosome in the genetic algorithm represents a point in the search space and a possible solution for the desired issue. During the study, the genetic algorithm selects the appropriate and valuable strands for it, and It removes a cluster of strands that are more fitness with the population (the number of chromosomes), constantly correcting a community of individual answers. At each stage, the genetic algorithm randomly selects people from the current generation as parents and uses them to develop children who are members of the next generation. During successive generations, the population of the answers will reach an optimal solution of "evolution." At each stage, to create the next generation of the popular community, the current community uses three basic types of legislation: Selection rules select the specific answers to which parents are being said. Crossover laws combine the traits of parents to form their child, who will be a member of the next generation. Mutation rules are randomly applied to one parent (or both of them) to form the children of the next generation.

Fitness of heredity

The fitness function is designed to solve any problem using a genetic algorithm. This function turns a non - negative numeric function for each chromosome, which represents the competence or the individual's ability of the chromosome.

Parameters related to the quality and ability of the chromosomes are expressed as follows:

Numbers: The number of neighbor's node is the number of nodes that node i can send or receive the data packets according to Eq. (1). The more the number of neighbors in a node, the more likely it will be sent the data packet from one neighbor node, hence it is one of the critical metrics in the fitness function.

$$|n| = \left\{ |D_{ij}| < board_j, |D_{ij}| < board_{i_{i,j \in N}} \right\} = \quad (1)$$

Distance from the selected neighbor to the nearest base station: The distance to the nearest base station, which is the ultimate destination of all network nodes, leads to higher energy consumption; therefore, the choice of neighbor nodes has the minimum distance to the base station of the function parameters. Regarding the coordinates of nodes and base stations, it has been Eq. (2) to calculate the distance:

$$D_s = Min\left(\sqrt{(x_i - x_s)^2 + (y_i - y_s)^2}\right), s \in Sink \quad (2)$$

Distance from the selected neighbor to the neighbor node: The range of node i from the neighbor node chosen, which is one of the metrics in the fitness function. The less energy required to send the data packet through the nodes in the range of node i. regarding the coordinates of the nodes, it has been Eq. (3) to calculate the distance:

$$D_n = \sqrt{(x_i - x_n)^2 + (y_i - y_n)^2}, n \in neighbors \quad (3)$$

Residual energy field: The remaining energy of the selected neighbor node as a parameter increased the probability of packet transmission to the destination if it is increased. If this parameter decreases, the probability of node inclination as the selfish node will be increased.

Fitness function: Fitness function F is defined by all the introductory fitness metrics below as Eq. (4):

$$F = \omega_1 * \frac{|n|}{|n_{max}|} + \omega_2 * \frac{E_n}{E_{max}} + \omega_3 * \frac{D_n}{D_{max}} + \omega_4 * \frac{D_s}{D_{max}}$$

$$, \quad \omega_1 + \omega_2 + \omega_3 + \omega_4 = 1 \qquad (4)$$

4-3 learning and updating phase

For each nodeI_k, the proposed algorithm appointed a learning automata. Each one of $LA_k^1 \ldots LA_k^{NR}$learning automata LA_k^m in the round R^m to assist nodeI_k, it activates to select the best neighbor node in the round R^m to forward and send the packets to the destination. Each learning automata have three operations expansion, contraction and without change, which is calleda_1 ,a_2 and a_3 for each operation, and the probability of choice isp_1,p_2 $p_3$respectively. Node I_k increases (or decreases) the probability of selecting a neighbor node in the first round for forwarding the packet. If the expansion operation (or contraction) chooses, and if it wants a without change state, it remains unchanged.

At the beginning of the algorithm, all neighbors of node I_k have the same probability of choice. In each R^m, for each neighbor is equal with P_1=0.5,…, P_{N-1}=P_N=0.5, which is given to all neighbor nodes according to its fitness function at the beginning of the algorithm, and the increase or decrease of each of this probability is the choice of independent from another neighbor node. On the other hand, the increase in the probability of the one neighbor node selection will not lead to any further reduction. If the sum of the neighbor nodes probabilities is assumed equal to 1; it leads to the probabilities of the dependence selecting neighboring nodes by increasing the probability of choosing a neighbor node to maintain a total of 1 whole probability, it must reduce the possibility of selecting any other neighbor nodes that are not logical. Other neighbor nodes have no practical application to reduce their choice probability. Hence, the option of selecting the neighboring nodes are assumed to be independent of each other and is proportional to the learning function in $I_k$node, the probability that each neighbor node will be able to select its appropriate value.

Each round simultaneously begins at each I_K node by activatingLA_k^m. At the beginning of the round, the I_K node selects one of the neighboring nodes that have the most probability to cooperate and the highest amount of fitness function to forward the data packet. If the node receives a message from the cluster head as the neighboring node is the selfish node, it reduces the probability of the selecting node avoiding to forward the data packets in the next rounds. Also, if the neighbor node does not forward the I_K node's data packet, the node is reported to the cluster head. The purpose of the paper is to select the best neighbor node to forward the data packet, avoid reducing network throughput and performance by not sending the data packet to the selfish nodes.

Selection of the best neighbor node for forwarding packets: As mentioned earlier, at the beginning of R^m each I_K node uses theLA_k^m learning automata and fitness function to select the best node for forwarding the data packet. At first, LA_k^m randomly selects one of its operations based on the probability vector and is represented by α. If α is the expansion action, the probability of selecting the desired neighbor node is increased by a predetermined constant, and if the state is unchanged, its value remains constant, but the probability values must be in the range [min, max]. (Minimum node selection probability (min = 0) and maximum node selection probability (max = 1))

Figure 3. Simi-code of learning automata

The network performance in each cluster collected for the R^m period and the performance of neighboring nodes in the cluster evaluated then; the nodes do the learning correctly. For this purpose, in line 6, each cluster member node generates a random value between 0 and 1 to send its data packet through its neighbors to correct the probability of neighbor node selection for action α1, α2, α3 which are expansion, contraction or unchanged with probabilities p1, p2 and p3, respectively, which are equal to 1. If the value of a probability decreases, it will increase to the other until the selected actions are correctly chosen, as shown in line 28. These values will be encouraged and punished at the end of the round by evaluating the situation in order to select the actions accurately; If the random number is more significant than 0.33, the expansion operation will increase the probability of choosing the neighbor node in the next round. If this increase reaches the maximum probability value of the neighbor node selection, the status of the neighbor node will change to the cooperation node. It will not change unless a message sent from the cluster head in line 32. If the random number is between 0.33 and 0.66, the probability value remains unchanged. If the random number is more significant than 0.66, the contraction action reduced, and the probability of the neighboring selection node will reduce in the next round. If this decrease reaches the minimum value of neighbor node selection, the status of the neighbor node is likely changed to be a selfish node, and the node suspecting is reported to the cluster head, which done in line 41. At the end of the round, the destination received an acknowledgment message from the destination, and the

probability decisions will be determined to be correct or incorrect are shown in Figure 2. The flowchart relates to the learning phase in Figure 4. It performed in parallel to all nodes in the clusters.

Figure 4.Flow chart of learning phase

At the beginning of round R, the above procedure uses the best neighbor node with the most fitness function for forwarding the data packet node K. The probability will change according to neighbor nodes behaviors in clusters at the end of the round R.

4-3-1 reinforcement signal

Based on the assumptions, the network throughput will increase by selecting the best neighbor node with the highest fitness of IK node for forwarding the data packet. And performance indicates that the value of fitness function and reinforcement signal in IK node learning automaton is useful in selecting the best neighbor node for transmitting the data packet. Still, careful selection of the network could not predict in advance. Therefore, its variety calculated by the time the node's data collection has done in the node's work environment. At the end of round R, the neighbor node selected for forwarding, and this message will provide the reinforcement signal for the learning automaton in the node IK as follows:

If the Ik node selected action has selected a specific neighbor N node to forward the packet:

- If the desired data packet reaches the destination and correctly sent by neighbor node and the selective action of learning automata is to expand the probability of the neighbor node selection, the reinforcement signal rewarded for the selected action.
- If the desired data packet doesn't reach to the destination and not forwarded by neighbor node and the selective action of learning automata is to expand the probability of the neighbor node selection, the reinforcement signal punished for the selected action.

If the Ik node selected action has not chosen a specific neighbor N node to forward the packet:

· If the desired data packet reaches the destination and correctly transmitted by neighbor node and the selective action of learning automata is to expand the probability of the neighbor node selection, the reinforcement signal punished for the selected action.

- If the desired data packet doesn’t reach to the destination and not forwarded by neighbor node and the selective action of learning automata is to expand the probability of the neighbor node selection, the reinforcement signal is rewarded for the selected action.

Each I_k node will stop learning individually if one of the following conditions occurs:

- The probability of an operation in LA has reached a certain threshold.
  - The probability of actions selected by LA is higher than maximum value or lower than minimum one.

In order to determine the probability values for the selective actions to apply if any of the states for punishment or rewarded to do, if action α_i is selected in step n and this action receives a favorable response from the environment, it is rewarded and the probability of $p_i(n)$ increases and other probabilities decrease. For the unfavorable response and punishment state, the choice of action α_i decreases the probability of $p_i(n)$ and the other probabilities increase. However, changes are made so that the sum of $p_i(n), i = 1,2,3$ is always constant and equal to one. The increase or decrease the probability of different conditions in LA with fixed structure as equation (5).

Favorable response and rewarded the action α_i

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)]$$
$$\forall j\ j \neq i \qquad p_i(n+1) = (1-a)p_i(n) \quad (5)$$

Unfavorable response and punishment the action α_i

$$p_i(n+1) = \frac{b}{r-1} + (1-b)p_i(n)$$
$$(6)\forall j\ j \neq i \qquad p_i(n+1) = p_i(n) - (1-b)p_i(n)$$

The semi-code of the reinforcement signal phase is shown in Figure 4, which is performed after receiving the acknowledgment message at the end of each round.

In other words, it is possible to evaluate neighboring nodes for forwarding packets by receiving an acknowledgment message from the destination and the neighboring nodes should be rewarded or punished is determined in third phase.

Figure 5. Reinforcement signal in learning phase

Figure 5 shows a flowchart of the proposed method; it is clear that, at the beginning of the first round, the cluster heads monitor the operation of their clusters and member nodes. It can also learn to detect the behavior of other nodes in the learning phase. Notify each other if any nodes or cluster heads prove to be suspicious or to be selfish.

5-Simulation and evaluation

This paper encountered the problem of the selfish nodes in IoT. The nodes don't cooperate with other nodes to forward the data packets, and waste nodes' energy by dropping the packets are called selfish nodes. So, the network throughput and end-to-end delay are active by the presence of selfish nodes. The different criteria introduced to stimulate the mentioned problem in the next section, and the proposed method are compared with other similar methods and evaluated the simulation results.

5-1 Evaluation Criteria

Different criteria are reviewed for the proposed scheme using GA and LA for detecting selfish nodes in IoT. The evaluation metrics defined in the following:

Detection accuracy:

The selfish node detection accuracy indicated the ratio number of identified the selfish nodes to all the selfish nodes in IoT is denoted DA and TP as the number of detected cooperation nodes, and FN indicated the number of selfish nodes, but as mistake recognized as cooperation nodes, the detection accuracy of the selfish node is according to equation (7) in Table (2).

False positive rate (FPR): The false positive rate is another metric to evaluate selfish nodes detection proposed method in IoT. The false positive rate indicated the ratio of the cooperation nodes number detected as a selfish node by error to the total number of cooperation nodes identified by mistake and the number of detected selfish nodes in IoT. FP denotes the cooperation nodes number recognized as a selfish node by error, and TN indicated the number of identified selfish nodes. Therefore, the false positive rate (FPR) is defined in equation (8) in Table (2).

False negative rate (FNR): the metric is related to the accuracy to evaluate the efficiency of selfish node detection methods. The false negative rate defined in equation (9) in Table (2), which is the ratio of the number of the selfish nodes detected as cooperation by mistake to the total number of detected selfish nodes as cooperation and the number of identified cooperation nodes in IoT.

Throughput: Throughput is one of the evaluation metrics in bits per second in most fields of IoT. The average rate of successful packets delivered to the destination to the number of all packets produced in the network. Throughput is according to equation (10) in Table (2), which is PD indicates the number of successful packets delivered to the destination, and PP indicates the number of all packets produced in IoT.

End-to-End delay: The average end-to-end delay is the arrival time of a packet from the source node to the destination.

Energy consumption: IoT system nodes assume sensor nodes in this article. So, each node uses the energy model as equation (11) in Table (2) l denotes the number of packets in bits and Eelec indicates the consume energy to activate the circuits. Eamp and Efs mean energy required to amplify the signals to transmit a bit in open space and multipath, respectively. d0 denotes the threshold destination, and d is the destination between source and destination nodes.

Table (2): equation of the evaluation metrics

5-2 Simulation result

The proposed approach has made decisions about both the cooperation and selfish nodes by using GA and LA. It simulated in core i7 processors, 370 M processors, 2.40 GHz of speed with a memory of 8 GB, Window 8.1 basic (64-bit), and MATLAB 2018 software. The simulation results of the proposed method compared with the Game theory-based [21], PPS [25], and Trust management [26] protocols in evaluation metrics like throughput, average end-to-end delay, detection accuracy, false positive/negative rate, and energy consumption. The simulation performed 100 runs, and the simulation results have shown and indicated in different charts.

A network performed in an intelligent agriculture application environment with an area of 1000*1000 m^2, and some base stations placed to collect data—the nodes randomly distributed in IoT for four different types of sensor nodes. The nodes have different numbers and parameters with four different types of nodes which can use in agricultural fields as controlling water, controlling soil, controlling the weather, and controlling temperature. The considered internet network includes fixed things with limited energy source similar to wireless sensor networks. All of the nodes have wireless communications. The proposed mechanism clustered the nodes, and the cluster heads have contact with cluster members in clusters and try to transfer the data packets to the base stations are closer to the cluster heads, as mentioned in section 3.1.

However, the initial energy of the nodes in the clusters are 0.5, 1.5, 1, and 1.1 Jules, and 200, 100, 200 and 200 number of nodes in clusters with a radio range of 80, 70, 75, 70 m respectively. But the energy model and the type of nodes are the same and following equation (11) in Table (2).

The detection accuracy (DA) is one of the critical metrics to detect the selfish node in IoT. 10% of the total nodes are assumed the selfish nodes in the simulation environment; further, the rate of selfish nodes gradually increased by 15%,

20% to 40%. In real situations, whenever, the nodes' energy level is decreased than the initial level, the nodes want to be work as a selfish node. They don't want to forward the other nodes' data packets to save their energy resources. According to Fig. 6, detection accuracy of the selfish node has shown increasing in comparison with other methods. When the number of selfish nodes increases in the network, it doesn't lead to a more significant changing in diagram slope of the proposed method. The probability of each node selection will be updated during the third phase and while forwarding the data packets through the neighbor nodes. Therefore, when 10% of the nodes in the network are selfish nodes, the proposed method detection rate is higher than other methods, and the probability of the neighbor nodes are well known, and 94% of the selfish nodes have detected. Changing in diagram slope is invisible even with the increased percentage of the selfish nodes in the network, the probability of the neighbor nodes will be updated in the third phase by using LA, and detection has done accurately. Up to 98% of the selfish nodes will be detected. However, an increased number of selfish nodes in the network needs to select more routes to the destination, but due to the more energy consumption, it is not a rational way. GA decides the best routs to forward the data packets, an acceptable percentage of selfish node detection will be achieved even by a high rate of the selfish nodes. Comparing the proposed approach has shown even in the numerical values, the algorithm detection will be more accurate than other algorithms even by the high percentage of the selfish nodes and has a slighter slope as a comparison other similar mechanisms.

Fig. 6. Comparison of detection accuracy (DA) in IoT

The fact that the proposed scheme has a slighter slope compared to the methods Game theory-based [21], PPS [25], and Trust management [26] protocols have shown in Table 3. The proposed method uses GA, and LA processes in each cluster to detect the selfish nodes. In contrast, other processes in higher percentages of the selfish nodes are usually unable to identify them in high detection accuracy.

The other metrics to evaluate the proposed scheme is FPR, which has inverse relation means that how it is low, the accuracy is high. If the number of normal nodes has detected, the network throughput will be high. The reason for that is the nodes in the network aren't cooperate in forwarding the data packets with the nodes detected as selfish nodes. The throughput will be low if the cooperation nodes are identified as selfish nodes by error. As mentioned before, using more routes and repeated to forward the data packets in the network can help the nodes LA to learn better and refuse to have an error in detecting the selfish neighbor nodes. The different situation is implemented and simulated to evaluate the network throughput. The numerical comparison has shown that the false positive rate of the proposed scheme is lower than the other algorithms spatially in the high percentage of the selfish node in the network. As shown in Fig. 7, the FPR has less numerical value than different algorithms when more than 25% of the network nodes are selfish nodes and fewer mistakes detected than others. It has a slighter slope as a comparison of other similar mechanisms Game theory-based [21], PPS [25], and Trust management [26] protocols. The fact that the proposed method has a lower false positive rate compared to other methods have shown in Table 3.

Fig. 7. Comparison of the different algorithm in false positive rate (FPR) metrics

Fig. 8 has shown three metrics to evaluate the proposed scheme for the detection accuracy (DA), the false positive rate (FPR), and the false negative rate (FNR) in the percentage of selfish nodes from 10% to 40%. FNR metrics have a slighter slope in the proposed method chart, which increases with the increase in the number of selfish nodes in the network. But it has a disproportionate effect on network performance and, considering the diagrams in Fig. 11, this weak point of the proposed approach was negligible, and further work on this issue will examine further.

Fig. 8. DA, FPR,FNR metrics in the proposed method

Throughput is one of the critical metrics to evaluate the performance of the network. The high rate of the selfish node leads to decrease throughput. The selfish nodes by refusing to forward the data packets make to resend them and increase the traffic in the network. Resending the packets leads to decreasing the throughput and is a weak point in the system. The proposed mechanism can detect selfish node so, it led to high throughput and proper usage of resources, including bandwidth or limited energy batteries in the nodes. The throughput chart observed the proposed method has high numerical value by early and accurate selfish node detection in figure 9. Not only has the scheme had high throughput but also low traffic bandwidth and average end-to-end delay by preventing the repeated data packets to the same destination. Table 3 shows the network throughput in the proposed method and similar algorithms PPS [14], Game theory-based [29], and Trust management [32] protocols.

Another point is that throughput has a direct relationship to the detection accuracy. If the accurate of the scheme is high and the selfish node detected correctly, the successful data packets will deliver a high rate, and throughput of the algorithm will be in high standard.

Fig. 9. Comparison of the throughput metrics in a different algorithm

The average end-to-end delay decreases for the packets in the system by detecting the selfish nodes. If the selfish node rate is increasing in the network, the average end-to-end delay will increase, and it will take a lot of time to deliver the packets to the destination. As mentioned before, the selfish nodes dropped the packets and the source node resend it and the process will increase energy consumption and the average end-to-end delay. The proposed mechanism detects the selfish node, and it causes to reduce the side effect of the selfish node like increasing the average end-to-end delay in the system. Some of the selfish nodes maintain the packets in their buffer and send it with delay. It will increase the average end-to-end delay or even drop the packet by expiring the lifetime by the intermediate nodes. Fig.10 has shown an average end-to-end delay in the proposed method and the numerical value determined in Table (3) by different comparison methods. The proposed scheme has high accuracy in detecting the selfish node so, it will prevent to resend of the data packets, and it will reduce the average end-to-end delay. The emergency or real-time applications need a low end-to-end delay, and the scheme is suitable for them. The delay metric has inverse relation in the network, and the more accurate in the proposed method can be one of the essential advantages and reduce delays in the system.

Fig. 10. Comparison of the average end-to-end delay by a different algorithm

Energy consumption is an essential metrics effected on network efficiency. IoT nodes (sensor) have battery resources, then they have limited energy power, and lower energy consumption led to more lifetime in nodes. The simulation results indicate energy consumption varies 3.1409 ~ 3.1915 in micro-Joule. Resending the data packets increase the system traffic and energy consumption to forward the repeated packets are not useful. Selfish node detection can prevent to improve energy consumption. Figure 14 depicts average energy consumption in the simulation area by applying different packet traffic in 2, 4, 6, 8, 10 CBR during 100 rounds. During the field is collected the packets and proposed scheme tries to detect the selfish nodes. The energy charts illustrated less energy consumption due to the proposed method of detecting the selfish nodes and reduces energy consumption.

Fig. 10. Comparison of energy consumption in different traffic (2, 4, 6, 8, 10 CBR)

Table (3): different metrics of proposed methods in compare with other methods

With the most real-time application and other smart applications in IoT, the dataset hasn't recognized the standard deviation. If the distribution assumes the mean of the samples as $\bar{x}$ and the standard deviation will be as $\frac{s}{\sqrt{n}}$. But if the t distribution with mean μ and size of the sample is n, it will define the freedom degrees as n-1. The standard error is estimated by the exact value of the standard deviation as σ. If the sample dataset isn't known as standard distributed, the mean of the samples assume $\bar{x}$ and interval to the sample as a random sample is $\bar{x} \mp t^* * \frac{s}{\sqrt{n}}$ where $t^*$is the value of the upper bound in the critical situation. For example, in agriculture application with controlling weather by the sample mean 28.5 degrees of centigrade the sample estimated mean is $\frac{0.73}{\sqrt{700}} = 0.082$ with 90% random interval is approximated $28.5 \mp 1.64 * 0.082 = (28.5 \mp 0.13) = (28.37,28.63)$ or the other sample assessed mean is $\frac{0.73}{\sqrt{700}} = 0.082$ with 90% random interval is approximated $31.6 \mp 1.64 * 0.082 = (31.6 \mp 0.13) = (31.47,31.77)$

Table (4): descriptive statistic

6- Conclusion

The paper presented a new multi-phase scheme based on Genetic Algorithm (GA) and Learning Automata (LA) to detect the selfish node in IoT. The proposed mechanism is a multi-step method that is performed nodes gene in a clustered to send data to source and this gene is evaluate by fitness function if it has the highest value the gene is selected as rout to forward the data. The acknowledgment packet from destination learn the LA about the nodes status are cooperate or selfish. The performance of the method has been tested on the network and compared with Game theory-based [21], PPS [25], Trust management [26]. The results have shown that the proposed method can detect

nodes in high accuracy and decreasing end-to-end delay and consumption of node resources (energy, battery, memory, etc.). The average throughput is as an important criteria to evaluate successful data packets are delivered to the destination up to 15% and the average end-to-end delay is reduced by 12%. Also, the percentage of selfish nodes detection accuracy increased by 10% compared to other methods, and the false positive rate and false negative rate is decreased by 8%. Finally, the proposed mechanism gives the second opportunity to the selfish nodes cooperating with other nodes. All nodes are equipped by LA and can give second chance to the selfish nodes to prevent the crash of network.

Abbreviations
DA: Detection Accuracy; FPR: False Positive Rate; FNR: False Negative Rate; TP: the number of selfish nodes detected; FN: the number of nodes which are selfish nodes but detected as normal nodes; FP: the normal selfish node detected as normal node; TN: the total number of normal nodes detected by mistake; FN: the number of the selfish nodes detected the normal node by error; TP: the total number of selfish nodes detected by normal node; and also the Table 2 has shown more abbreviations and notation are used in this manuscript.

Competing Interest
The authors declare that they have no competing interests.

Funding
The research was not funded

Availability of data and materials
We declare that the MATLAB code used for the simulation will not be shared, and we assure that we will send it on demand.

Authors Contribution
SN contributed to the main idea and drafted the manuscript, algorithm design, and performance analysis and HGG and AK has performed the statistical and performance analysis. AMR conceived of the study, and participated in its design and coordination and helped to draft the manuscript. All authors read and approved the final manuscript.

Acknowledgements
Not applicable

References

[1] Balaji, S., Karan Nathani, and R. Santhakumar. "IoT technology, applications and challenges: a contemporary survey." Wireless personal communications 108.1 (2019): 363-388.
[2] Kumar, Sachin, Prayag Tiwari, and Mikhail Zymbler. "Internet of Things is a revolutionary approach for future technology enhancement: a review." Journal of Big Data 6.1 (2019): 111.
[3] Serpanos, Dimitrios, and Marilyn Wolf. Internet-of-things (IoT) systems: architectures, algorithms, methodologies. Springer, 2017.
[4] Aleksandrovičs, Vladislavs, Eduards Filičevs, and Jānis Kampars. "Internet of things: Structure, features and management." Information Technology and Management Science 19.1 (2016): 78-84.
[5] Hassija, Vikas, et al. "A survey on IoT security: application areas, security threats, and solution architectures." IEEE Access 7 (2019): 82721-82743.
[6] Khattak, Hasan Ali, et al. "Perception layer security in Internet of Things." Future Generation Computer Systems 100 (2019): 144-164.
[7] Khattak, Hasan Ali, et al. "Perception layer security in Internet of Things." Future Generation Computer Systems 100 (2019): 144-164.
[8] Vijithanand, J., and K. Sreerama Murthy. "A Survey on Finding Selfish Nodes in Mobile Ad Hoc Networks." International Journal of Computer Science and Information Technologies3.6: 5454-5461, 2012.
[9] Padiya, S. A. G. A. R., Rakesh Pandit, and Sachin Patel. "Survey of innovated techniques to detect selfish nodes in MANET." International Journal of Computer Networking, Wireless and Mobile Communications (IJCNWMC), ISSN (2013): 2250-1568.

[10] Samian, Normalia, et al. Cooperation stimulation mechanisms for wireless multihop networks: A survey. Journal of Network and Computer Applications 54 (2015): 88-106.
[11] O. León, J. Hernández-Serrano, and M. Soriano, “Outwitting smart selfish nodes in wireless mesh networks,” Int. J. Commun. Syst., vol. 23, no. 5, pp. 633–652, 2010.
[12] S. K. Das, P. S. Chatterjee, and M. Roy, “Detecting and Punishing the Selfish Node and Its Behavior in WSN,” vol. 4, no. 2, pp. 11–15, 2014.
[13] G. Rizwana and G. Wasim, “Enhanced Intrusion Detection & Prevention Mechanism for Selfishness in MANET,” Int. J. Innov. Res. Comput. Commun. Eng., vol. 3, no. 10, pp. 10131–10138, 2015.
[14] C. Science and S. Engineering, “A Comparative Study of Selfish Node Detection Methods in Manet,” Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 5, no. 8, pp. 306–310, 2015.
[15] Nobahary, S., & Babaie, S. (2018). A Credit-based Method to Selfish Node Detection in Mobile Ad-hoc Network. Applied Computer Systems, 23(2), 118-127.
[16] T. S. A. Al-roubaiey, E. Shakshuki, and A. Mahmoud, “Video transmission enhancement in presence of misbehaving nodes in MANETs,” pp. 273–282, 2009.
[17] E. M. Shakshuki, S. Member, N. Kang, and T. R. Sheltami, “EAACK — A Secure Intrusion-Detection System for MANETs,” vol. 60, no. 3, pp. 1089–1098, 2013.
[18] M. Bounouni, “Acknowledgment-based punishment and stimulation scheme for mobile ad hoc network,” J. Supercomput., 2018.
[19] Basar, Tamer, and Geert Jan Olsder. Dynamic noncooperative game theory. Vol. 23. Siam, 1999.
[20] C. Vijayakumaran and T. A. Macriga, “An integrated game theoretical approach to detect misbehaving nodes in MANETs,” Proc. 2017 2nd Int. Conf. Comput. Commun. Technol. ICCCT 2017, pp. 173–180, 2017.
[21] Nobahary, Solmaz, et al. "Selfish node detection based on hierarchical game theory in IoT." EURASIP Journal on Wireless Communications and Networking 2019.1 (2019): 255.
[22] I. Introduction, “A REVIEW ON NODE ACTIVITY DETECTION , SELFISH & MALICIOUS BEHAVIORAL PATTERNS USING WATCHDOG,” pp. 1–5, 2017.
[23] A. Lupia, C. A. Kerrache, and F. De Rango, “TEEM : Trust-based Energy-Efficient Distributed Monitoring for Mobile Ad-hoc Networks,” no. 1, pp. 133–135, 2017.
[24] Nobahary, Solmaz, et al. "DISOT: Distributed Selfish Node Detection in Internet of Things." International Journal of Information and Communication Technology Research 10.3 (2018): 19-30.
[25] A. Jesudoss, S. V. Kasmir Raja, and A. Sulaiman, Stimulating truth-telling and cooperation among nodes in VANETs through payment and punishment scheme, Ad Hoc Networks, vol. 24, no. PA, pp. 250–253, 2015.
[26] Zhang, Wei, and et al. A novel trust management scheme based on Dempster–Shafer evidence theory for malicious nodes detection in wireless sensor networks. The Journal of Supercomputing 74.4 (2018): 1779-1801.
[27] Rani, Shalli, Syed Hassan Ahmed, and Ravi Rastogi. "Dynamic clustering approach based on wireless sensor networks genetic algorithm for IoT applications." Wireless Networks (2019): 1-10.

Table (1): advantages and disadvantages in different categories to detect selfish node

Systems	Advantages	Disadvantages
Reputation-based	High throughput Less end-to-end delay High detection rate Less channel traffic	High energy Consumption High overhead and complex systems No Robustness Against Collusions High false positive rate
Credit-based	less channel traffic High throughput Less end-to-end delay	No Robustness Against Collusions No second chance Less detection rate Less energy Consumption
Acknowledgment-based	Less false positive rate High detection rate	High channel traffic Less throughput High end-to-end delay
Game theory-based	Less end-to-end delay High detection rate Less channel traffic	No second chance High energy Consumption High false positive/negative rate Less throughput

Table (2): equation of the evaluation metrics

Evaluation metrics	**Equation**
Detection Accuracy	$\mathrm{DA} = \frac{\mathrm{TP}}{TP+FN}$ (7)
False Positive Rate	$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP+TN}}$ (8)
False Negative Rate	$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{FN+TP}}$ (9)
Throughput	$\mathrm{Th} = \frac{\mathrm{PD}}{\mathrm{PP}}$ (10)
Energy	$E_{TX}(l,d) = E_{TX-elec}(l) + E_{TX-amp}(l,d) = \begin{cases} lE_{elec} + l_{\varepsilon fs}d^2 & d < d_0 \\ lE_{elec} + l_{\varepsilon amp}d^4 & d \geq d_0 \end{cases}$ (11)

Table (3): different metrics of proposed methods in compare with other methods

Present of selfish node / Algorithms	Metrics	10	15	20	25	30	35	40
Proposed-method	**Detection Accurate (DA)**	0.95	0.96	0.97	0.98	0.95	0.96	0.96
Game theory-based [21]		0.93	0.99	1	0.99	0.98	0.96	0.95
PPS [25]		0.75	0.72	0.78	0.79	0.77	0.76	0.74
Trust management [26]		1	0.95	0.88	0.85	0.8	0.74	0.62
Proposed-method	**False Positive Rate (FPR)**	0.001	0.002	0.003	0.003	0.009	0.031	0.039
Game theory-based [21]		0	0	0	0.002	0.0127	0.0347	0.0549
PPS [25]		0.25	0.22	0.28	0.29	0.27	0.27	0.31
Trust management [26]		0	0	0	0.02	0.1	0.14	0.2
Proposed-method	**Throughput**	85	81	84	82	85	88	91
Game theory-based [21]		75.85	73.05	74.14	78.92	86.71	91.07	95
PPS [25]		48	41	43	38	35	32	19
Trust management [26]		85	78	81	84	82	71	73
Proposed-method	**End-to-End Delay** (ms)	17	15.02	11.08	8.8	6.2	3.01	0.9
Game theory-based [21]		16.35	17.01	17	12	7.93	4.1	1.47
PPS [25]		48	41	43	38	35	32	19
Trust management [26]		85	78	81	84	82	71	73

Table (4): descriptive statistic

N	Mean($\bar{x}$)	t^*	standard deviation	SE mean
700	28.5	1.64	0.73	0.082
700	31.6	1.64	0.73	0.082

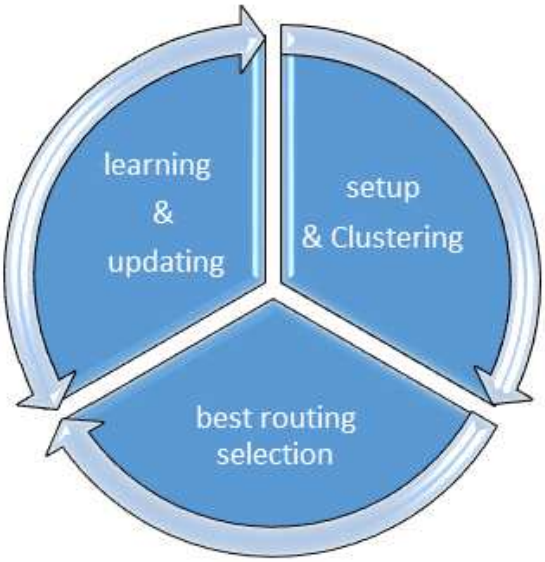

Figure 1. Selfish node detection mechanism

Residual _energy	Node_coordinate	Distance_nearest_Base station	Number_neighbor_node	Node_id

Figure2: the data packet content in nodes' Table

Algorithm. Learning Step

1: **For** r=1 **to** R^m
2: Each N_i choose the neighbors with max Fitness to send packet $N_j \in$ Neighbor N_i
3: **For** N_j=1 to n **do**
4: **If** $P_j <> P_{max}$
5: **or** $P_j <> P_{min}$
6: Compute Rand (0,1)
7: **If** rand< 0.33
8: Expansion ($P_j \in$ Neighbor N_i)
9: **If** $P_j=P_{max}$
10: $State_j$=C
11: **endif**
12: **endif**
13: **If** 0.33<rand< 0.66
14: No Change ($P_j \in$ Neighbor N_i)
15: **endif**
16: **If** rand> 0.66
17: Constraction ($P_j \in$ Neighbor N_i)
18: **If** $P_j=P_{min}$
19: $State_j$=LS and report N_{CHi}
20: **endif**
21: **endif**
22: **Endif**
23: **Endfor**
24: **…..**

Figure 3. Simi-code of learning automata

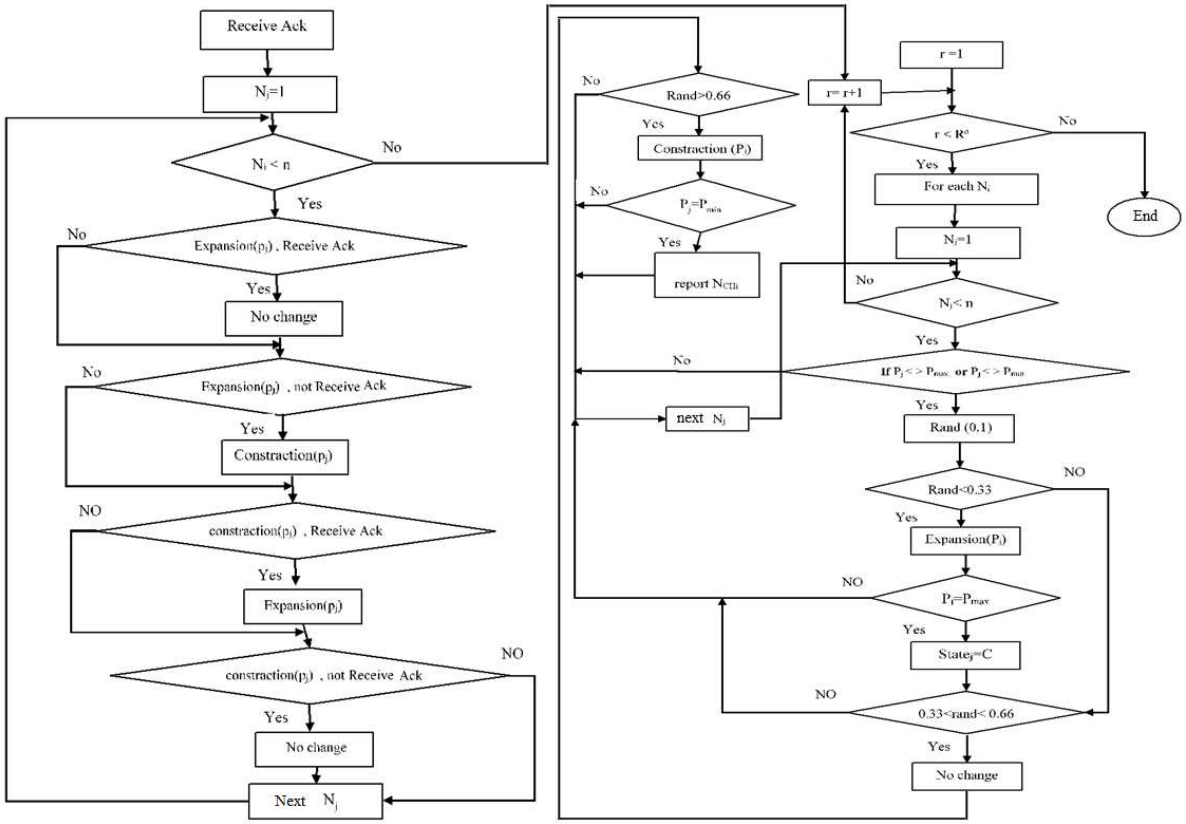

Figure 4.Flow chart of learning phase

Algorithm. reinforcement signal
25: After ack of data packet recieved from destination each N_i check status
26: **For** N_j=1 to n **do**
27: **If** Expansion (P_j∈ Neighbor N_i) **and** (ack of packet received)
28: No Change (P_j∈ Neighbor N_i) means reward
29: **Endif**
30: **If** Expansion (P_j∈ Neighbor N_i) **and** (ack of packet Not received)
31: Constraction (P_j∈ Neighbor N_i) means punishment
32: **Endif**
33: **If** Constraction (P_j∈ Neighbor N_i) **and** (ack of packet received)
34: Expansion (P_j∈ Neighbor N_i) means reward
35: **Endif**
36: **If** Constraction (P_j∈ Neighbor N_i) **and** (ack of packet Not received)
37: No Change (P_j∈ Neighbor N_i) means punishment
38: **Endif**
39: **Endfor**
40: **Endfor**

Figure 5. Reinforcement signal in learning phase

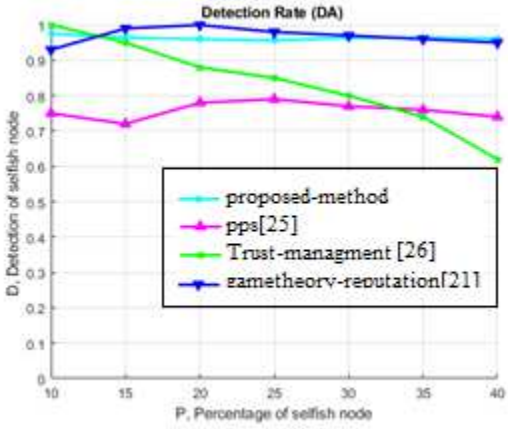

Fig. 6. Comparison of detection accuracy (DA) in IoT

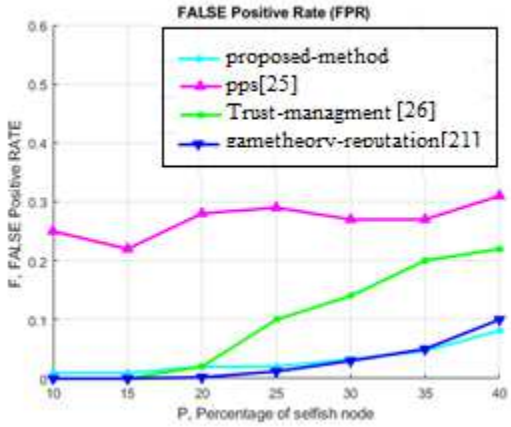

Fig. 7. Comparison of the different algorithm in false positive rate (FPR) metrics

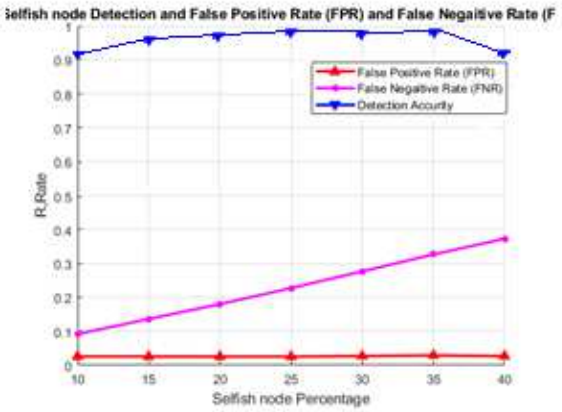

Fig. 8. DA, FPR,FNR metrics in the proposed method

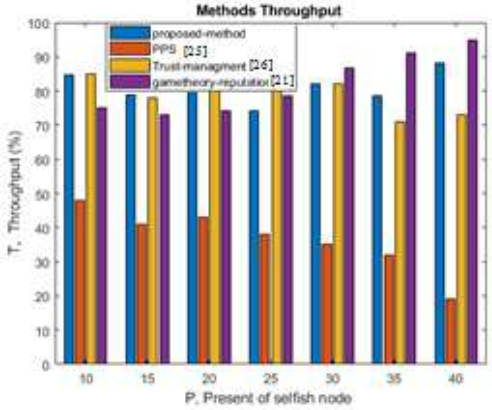

Fig. 9. Comparison of the throughput metrics in a different algorithm

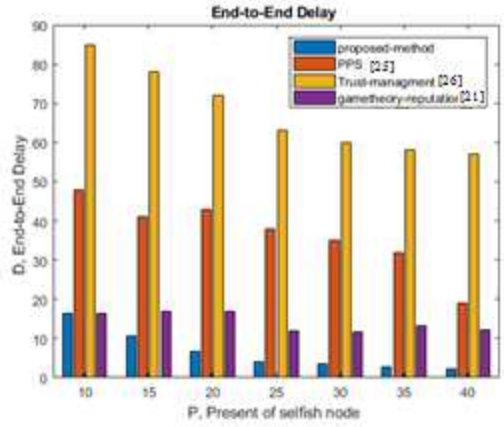

Fig. 10. Comparison of the average end-to-end delay by a different algorithm

.

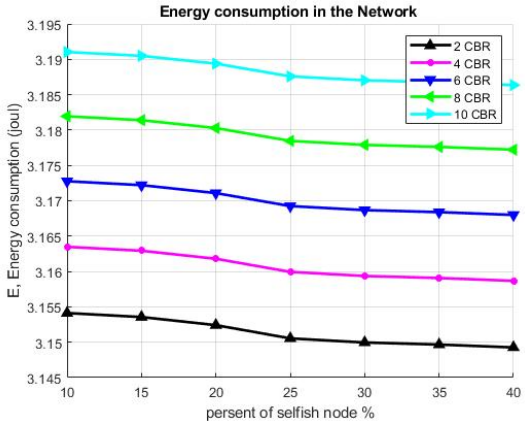

Fig. 11. Comparison of energy consumption in different traffic (2, 4, 6, 8, 10 CBR)

Figures

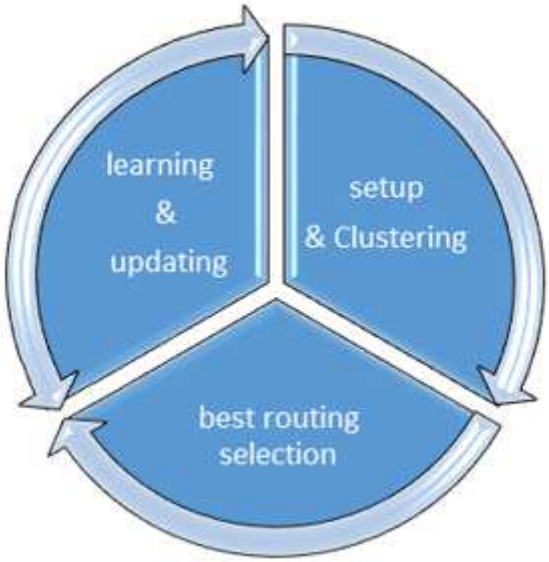

Figure 1

Selfish node detection mechanism

Residual _energy	Node_coordinate	Distance_nearest_Base station	Number_neighbor_node	Node_id

Figure 2

the data packet content in nodes' Table

Algorithm. Learning Step

1: **For** r=1 **to** R^m
2: Each N_i choose the neighbors with max Fitness to send packet $N_j \in$ Neighbor N_i
3: **For** N_j=1 to n **do**
4: **If** $P_j <> P_{max}$
5: **or** $P_j <> P_{min}$
6: Compute Rand (0,1)
7: **If** rand< 0.33
8: Expansion ($P_j \in$ Neighbor N_i)
9: **If** $P_j=P_{max}$
10: $State_j$=C
11: **endif**
12: **endif**
13: **If** 0.33<rand< 0.66
14: No Change ($P_j \in$ Neighbor N_i)
15: **endif**
16: **If** rand> 0.66
17: Constraction ($P_j \in$ Neighbor N_i)
18: **If** $P_j=P_{min}$
19: $State_j$=LS and report N_{CHi}
20: **endif**
21: **endif**
22: **Endif**
23: **Endfor**
24: **…..**

Figure 3

Simi-code of learning automata

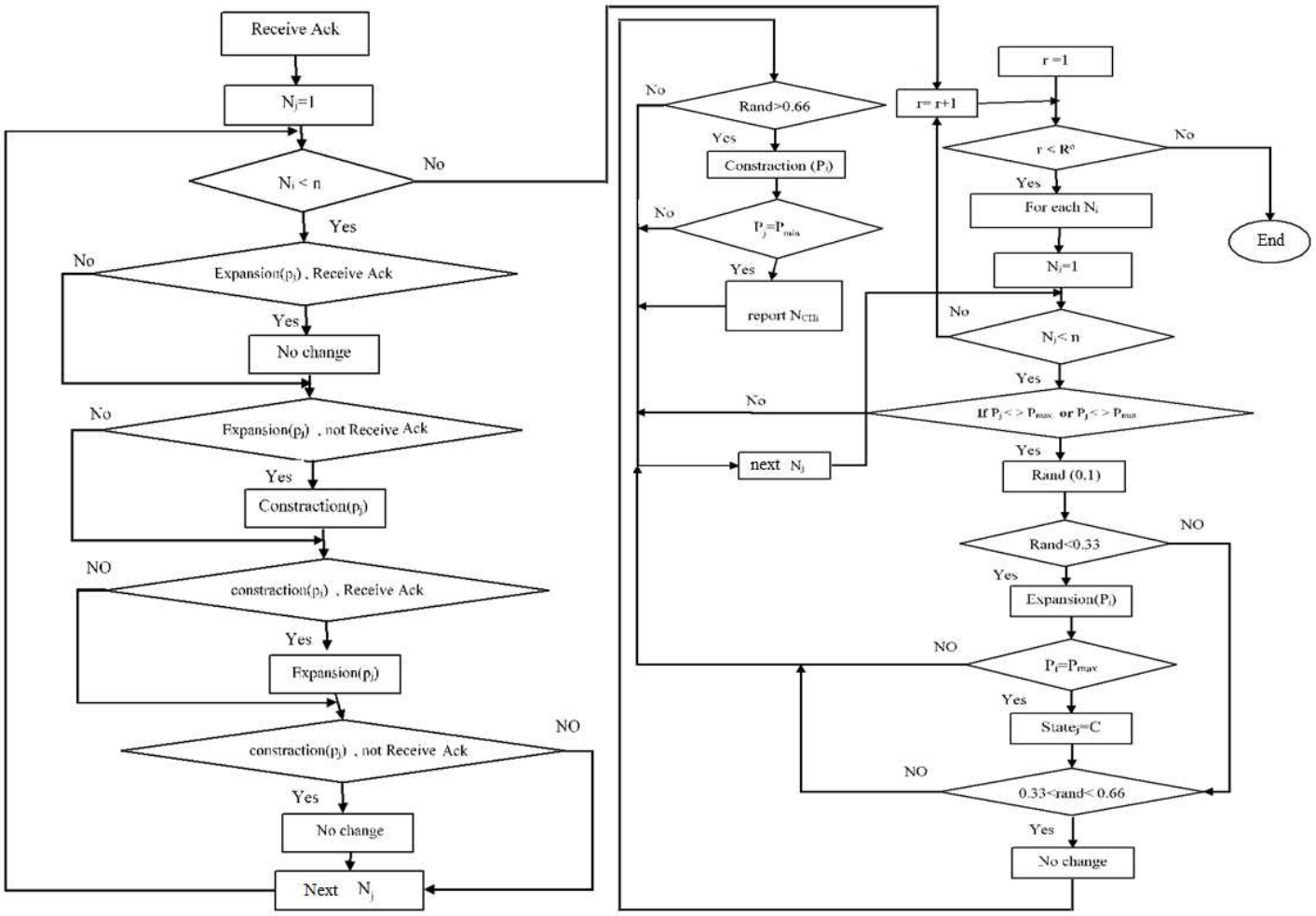

Figure 4

Flow chart of learning phase

Algorithm. reinforcement signal
25: After ack of data packet recieved from destination each Ni check status
26: **For** Nj=1 to n **do**
27: **If** Expansion (Pj∈ Neighbor Ni) **and** (ack of packet received)
28: No Change (Pj∈ Neighbor Ni) means reward
29: **Endif**
30: **If** Expansion (Pj∈ Neighbor Ni) **and** (ack of packet Not received)
31: Constraction (Pj∈ Neighbor Ni) means punishment
32: **Endif**
33: **If** Constraction (Pj∈ Neighbor Ni) **and** (ack of packet received)
34: Expansion (Pj∈ Neighbor Ni) means reward
35: **Endif**
36: **If** Constraction (Pj∈ Neighbor Ni) **and** (ack of packet Not received)
37: No Change (Pj∈ Neighbor Ni) means punishment
38: **Endif**
39: **Endfor**
40: **Endfor**

Figure 5

Reinforcement signal in learning phase

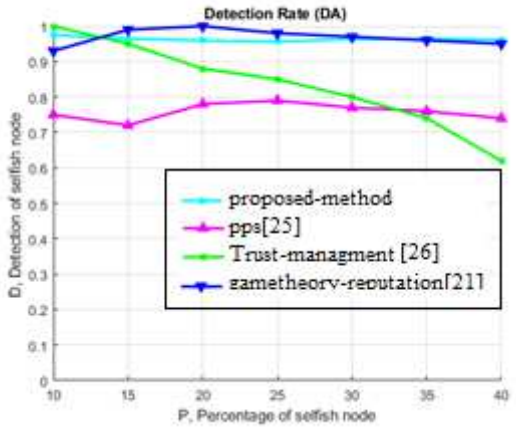

Figure 6

Comparison of detection accuracy (DA) in IoT

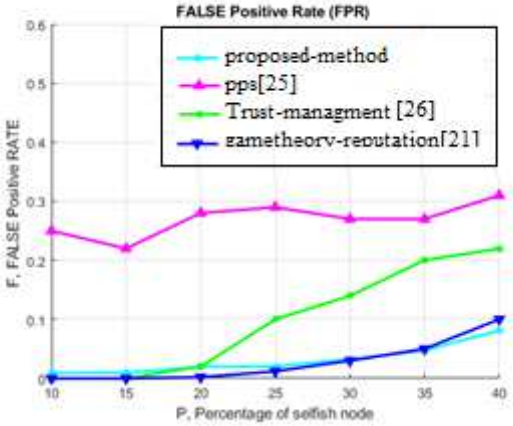

Figure 7

Comparison of the different algorithm in false positive rate (FPR) metrics

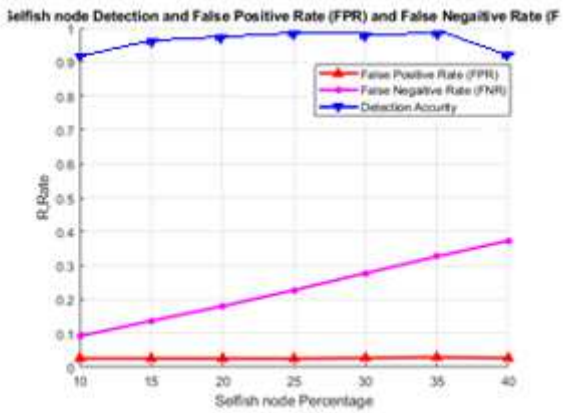

Figure 8

DA, FPR,FNR metrics in the proposed method

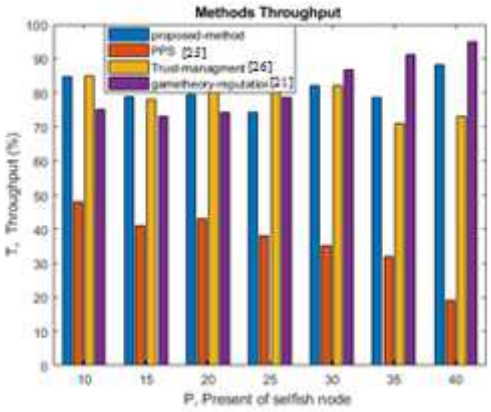

Figure 9

Comparison of the throughput metrics in a different algorithm

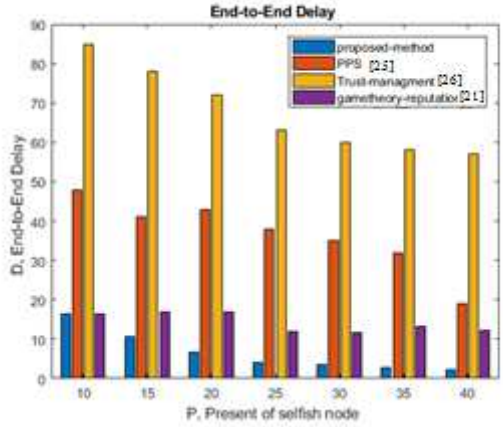

Figure 10

Comparison of the average end-to-end delay by a different algorithm

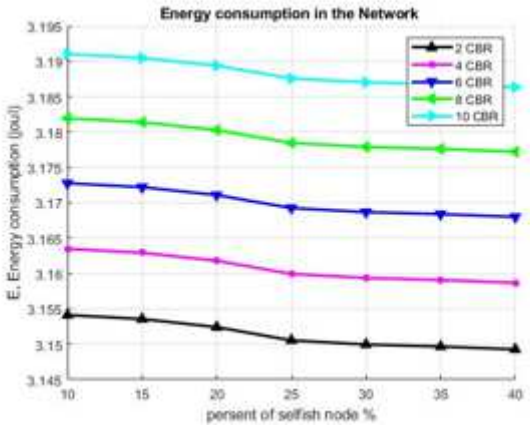

Figure 11

Comparison of energy consumption in different traffic (2, 4, 6, 8, 10 CBR)

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Tablesandfigure.docx