Hindawi
Security and Communication Networks
Volume 2021, Article ID 2658272, 20 pages
https://doi.org/10.1155/2021/2658272

Research Article

Selfish node Detection Based on Fuzzy Logic and Harris Hawks Optimization Algorithm in IoT Networks

Abbas Akhbari[1] **and Ali Ghaffari**[2]

[1]*Department of Information Technology Engineering, Payam Nour University, Jolfa, Iran*
[2]*Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran*

Correspondence should be addressed to Ali Ghaffari; a.ghaffari@iaut.ac.ir

Received 2 September 2021; Revised 19 October 2021; Accepted 15 November 2021; Published 30 November 2021

Academic Editor: Marimuthu Karuppiah

The Internet of things describes a network of physical things for example, "things" that are connected with the sensors, software, and other technologies to connect and exchange data with other devices and systems via the Internet. In this type of network, the nodes communicate with each other because of the low radio range by step by step with the help of each other until they reach their destination, but there are nodes in the network that do not cooperate with other nodes in the network, which are called "selfish nodes". In this paper, we try to detect selfish nodes based on a hybrid approach to increase the performance of our network. The proposed method consists of three stages: in the first stage, with the help of the Harris hawk operation, we try to set up the cluster and select head cluster; in the second stage, the sink investigates the existence or nonexistence of selfish nodes in the network by considering the general parameters of the network; and in the event of a selfish node in the network, it informs the head clusters to check the cluster members and recognize the selfish node. In the third stage, with the help of fuzzy logic, the amount of reputation of each of the nodes has been realized, and finally, with the help of fusion of head clusters and fuzzy logic, each node is decided to be cooperate or selfish nodes, and in case of head clusters and fuzzy logic in some cases, the opportunity node will be reestablished to participate in network activities otherwise the node will be isolated. The results show that the accuracy of selfish node detection has increased by an average of 12% and the false positive rate is 8% in comparison to existing methods.

1. Introduction

With the influence of new technologies, the coverage of the networks is increased, and the number of intelligent things connected to the network have also increased, and a new pattern has been created with the name of the Internet of things [1–3]. The Internet of things has evolved versions of wireless technologies, microelectromechanical systems (MEMS) that provide wireless connectivity of physical and virtual things. The Internet of things consists of connected smart mechanical devices to each other, computational devices, digital machines, and things that are connected to the cloud data center [4–6]. Internet of things devices are fixed and mobile things in providing simple and sophisticated services for cloud environments and macro data applications [7, 8].

The Internet of things is a dynamic global network structure that has its capabilities to configuration and is based on standard communication protocols and interoperability and uses smart interfaces that are integrated into the data network; the Internet of things has been made up of many technologies which can be referred to as range intelligence, Internet Protocol, Communication Technologies (WIFI, Bluetooth, and Zig bee), and RFID embedded devices and applications [9, 10]. With recent advancements in Internet of things technology, things have more influence on our daily lives and has begun offering new, interesting, and helpful services [11, 12]. The Internet of things has created hope for creating new opportunities in various industries such as health care, *e*-commerce, power plant (green energy), manufacturing industries, city/organization/smart

houses, as well as personal applications. In general, the main purpose of IoT is to increase the quality of human life [13].

In these networks, nodes use finite energy sources, so the management of energy consumption in these networks is one of the most important issues [14, 15]. Owing to the limited range of things in these networks, the data are carried out as step by step. Nodes having a noninvolvement in data transmission are a several steps to mitigate the performance of the network; these are called the selfish nodes [16]. Lack of a propellant depends on the interest of selfish nodes because they do not use energy to send data packets to other nodes. The selfish nodes are expecting cooperation from other nodes to send their data packets to the destination, but they are not willing to cooperate with other nodes [17].

Because of the lack of communication infrastructure, if a node in this network is willing to cooperate and transmit its neighbors' data packets to other neighbors to reach the final destination, the data packets will be lost and the network will face the problem [18]. These nodes do not perform any assistance to save energy consumption and communicate with other nodes in the routing and propellant operations, thus cutting off some nodes with the network [19]. The selfish nodes use the network facilities for their personal uses and do not contribute any assistance to reducing energy consumption, thus increasing network lifetime, as well as maintaining the relevance of other nodes [20].

The proposed method is used to reduce the effects of selfish nodes in a multistage method; in the proposed method, we will have three stages: the first stage includes the establishment and setting up the cluster and cluster head selection, which will be done with the help of the Harris Hawks optimization; in the second stage, head clusters communicate their opinions to members of their cluster. In the third stage, the fuzzy logic is used to achieve the reputation of each node and ultimately to help integrate the opinions of head clusters and fuzzy logic on the partner or being selfish of each node to make decision. If the opinion of head clusters and fuzzy logic is in conflict in some cases, the opportunity node will be reestablished to participate in network activities; otherwise, the node will be isolated.

Finally, the main task of this paper can be summarized as follows:

We propose a Harris Hawks optimization and fuzzy logic based on strategy to establish cooperation for network nodes of Internet of things. To encounter the challenges of noncooperative behaviors in the network of Internet of things, we introduce and use an evocative strategy. The main idea is that, with the help of fuzzy logic, we obtain the reputation of each of the nodes, as well as the rest of the nodes analyzed by the sync of behavior in the network and it requests the cluster head for recognition of selfish nodes in the face of selfish behavior in the network. Then, the ideas of the fuzzy system and clusters are integrated, which makes the proposed method strong and better in decisionmaking on each node.

Using this strategy, we demonstrate that nodes are encouraged to do cooperation because any deviation from the level of uncertainty of the data packet results in a low cooperation or even noncooperation step that reduces the reputation of nodes, which eventually quit the network activity and gives them a chance again for low cooperation.

We provide theoretical analysis of the mixed method, and we model the proposed method as a reputation based method and estimate the level of power of cooperation with a reputation by the fuzzy system. We also show that the energy consumption of this method is less than the other proposed methods.

It is true that the dynamics of nodes in networks such as VANETs can cause problems, but in the Internet of Things where nodes are intended for sensors, this is not the case, even in cases where things are sufficiently mobile by reforming clusters with the help of new nodes, our method will be able to detect selfish nodes with a slight delay and it does not decrease network performance by the dynamicity of the network.

In the first section, the networks of Internet of things and the selfish nodes are introduced. Next, in the second section, algorithms for discovering selfish nodes are presented and then compared with each other. In Section 3, the proposed method will be presented with the help of a hybrid algorithm to eliminate these nodes and will be simulated in Section 4 of the proposed method where their results are evaluated, and finally, the conclusion and future work are presented in Section 5.

2. Related Work

The detection mechanisms of selfish nodes are divided into five categories [21]: the reputation-based approach that in this approach is the behavior of nodes along with feedback and some validity that node has been credited with high reliability as a trusted node [22]. The credit-based approach to which the nodes that have data to send will pay for it and sell them to the business of the data packets and then sell it at a higher price [23]. The method based on the punishment and persuasion of each network node monitors directly or indirectly on how other nodes cooperate and thus use the observations obtained to determine the other nodes' reputation [24]. The authentication-based methods that ensure that sending a packet by a node will be authenticated using the message [25]. The method based on the game theory is to send or resend between nodes as a model game, which can be between two- or as one-to-several, that is, defined for each game of utility function so that each node tries to obtain the maximum value from the utility function [17, 26, 27]. The following algorithms are stated to deal with the selfish nodes recently.

In the Nobahari method, a multiphase mechanism based on the game theory and reputation for excitation of noncooperative nodes has been designed for collaboration on the Internet of things. In the first phase, nodes are grouped into clusters with head clusters and at least one base station for data collection. Then, in the second phase (the multigame phase and packet transmission phase), they perform when the data packet is advanced a multiplayer game by itself or others.

If the neighboring node run the data packets, the node optimizes the likelihood of the neighbor node selection for the next period. In the third phase (the discovery of the non-cooperative node and update), nodes can choose their strategy while driving data packing. Nodes classified other nodes by their function and updated the reputation table [28].

In the other method, Nobahari et al. has been proposed an algorithm based on credit detection of being selfish which is used for better control and monitoring of nodes from clustering. Clusters are discovered by controlling the general characteristics of the network such as delay, the number of sent packets, and the number of delivered packets, throughput, and traffic network in existence of the selfish node. The credits are kept in a table where nodes with low reputation are considered selfish. Three guard nodes are selected from within the suspicious node's neighbors to monitor the behavior of the suspicious node. The guard nodes will send their attention to the cluster head based on the selfishness or cooperation of the node. The cluster head decides the majority of the suspicious node by voting. In case of selfishness diagnosis, it informs the other nodes of the cluster to not send data to the node from the network nodes [29].

In the method of Babaei et al, removing the nodes from the network by seeing the first suspect behavior significantly reduces the number of active nodes and often reduces the number of regular network missions. While these disorders are likely to occur over time, solve and the node may be a partner node in the future. The three variables, the number of steps, the remaining energy, and the history of cooperation are considered as input to the fuzzy logic process, and the amount of cooperation of the nodes is considered as the output of the proposed system. The node is selected with a high trust factor to send data. The amount of cooperation of nodes is as binary digit, where the node state is determined between 0 and 1 [30].

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 [31]. 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.

Acknowledgment-based methods guarantee that sending a packet through a node will be performed using an authentication message. In these methods, a node sends a confirmation message to the source when it requires to forward the packet. If a source node does not receive a confirmation message, it is considered unreasonable behavior. In confirmation-based methods, they guarantee that sending a packet through a node will be based on a confirmation message. In these methods, a node sends a confirmation message to the source when it wants to forward the packet. If a source node does not receive a confirmation message, it is considered unreasonable behavior. Liu and et al have proposed the 2ACK method. Mahdi Bounouni et al. proposed an approach consists of four models [32]. The monitoring model is responsible for controlling the sending of routing packets and data packets using the acknowledgment packet. The reputation model evaluates each nodes' neighbors. The stimulator model manages and updates nodes' credit accounts, and malicious and selfish nodes are punished by the isolator model.

In the NACK method, the destination node responds to a data packet sent by the origin, with a Nack packet. Each node has two lists: one for the packets sent and the other for the packets re-sent. As soon as the node is sending a data packet, it added the packet identifier to the list of sent data packets. As soon as the packet receives a Nack packet for the data packet, the associated data packet identifier is omitted from the submitted data packet list. The source node is compared with receiving the Nack message of the two paths in the Nack packet. In the absence of a change in the paths, the source node concludes that there is probably no malicious node in the path. In the presence of two different paths, the node in the source destination path is isolated as the path is different from the destination to the source path. The node is represented as the node of the bad behavior by the source node. The number of times a node is measured as a node is likely to be more than the threshold mark as the bad behavior node [33].

In this method (AISDTR), each node is analogous to the dendritic cell (DC) in the human immune system (HIS). As in the body, harmful cells are detected by pathogen-associated molecular pattern (PAMP) in this algorithm, nodes have a list of their well-behaved neighbors. The node will send its data source to its well-behaved neighbor node and then adds a confirmation message to the node's well-behaved frequency. With the receipt of the confirmation message, the source node perceives the bad behavior of the selfish node, and the data are transmitted from the other direction. Selfishness of a node is notified to other nodes so that the data will not be sent to the node [34–36].

The DSSAM method is proposed to identify the selfish and malicious nodes. This method uses digital signatures to establish node and packet authentication as well as a secure authentication message. A two-layered approach is used to secure the security of the authentication message. Additional bits and sequence numbers are used in the first layer for fixed transmission time and packets sequences. The next layer is the digital signature, where the seven bits are reserved, which are intended for maintaining sequence numbers. Data packets and confirmation packets are digitally signed by the source node and the destination node. RSA encryption is used in the proposed method [37–39].

3. Proposed Method

The selfish nodes are nodes that send their data but avoid sending other data nodes. The existence of such nodes paralyzes the network, hampering the natural flow of the

network and reduces network performance. To solve this problem, it tries to motivate the node to reduce the number of selfish nodes to work together. In the proposed method we will have three general phases. The phase of the setup and establishment phase (cluster formation and cluster head selection), which we will do with the help of the Harris Hawks optimization algorithm, is the second phase where the head clusters of each expressed their opinions about their cluster members, and the third phase is the use of fuzzy logic to achieve the reputation of each of the nodes, and at the end, the combination of the opinions decides of clusters and fuzzy logic on the being partner or selfishness of each node. In case of head clusters and fuzzy logic in some contrastive cases, the opportunity node will be reestablished to participate in network activities; otherwise, the node will be isolated.

3.1. Establishment and Clustering Phase. Each metaheuristic algorithm has its own advantages and disadvantages and is used in different usage; we can get the best results when they are used in their intended usage. The Harris Hawks optimization (HHO) algorithm is one of the best clustering algorithms that authors decided to use it for clustering and determining the cluster heads in each cluster in the proposed method. The method can select the best node as cluster head and cluster member' nodes.

The cluster head selection phase with the help of the HHO algorithm is that the HHO selects a group of sensors that are actually the network sensors. The location of each hawk with an identifier starts randomly from 1 to n (n is the total number of nodes in the network).

$H = (H_{i.1}(t), H_{i.2}(t), \ldots, H_{i.n}(t))$ in this equation i represent each of the hawks, and $H_{i.d}$ is the location of each hawk, which is $1 \le d \le n$, and m is the number of head clusters.

In the first phase, there are the n number of nodes given in the network. Each node has identified all of its neighboring nodes that have been able to send messages by sending the hello message in public and Provides information on its own status and its neighbors in a table consisting of four fields as shown in Figure 1 in its data base.

Further details of each of these fields will be investigated.

Node id field: this field is 8 bits and is used to store node identities.

Energy: One of the important parameters in the field of energy is because it affects network lifetime and network survival, so we need to store the remaining energy of each node. The node is chosen as the cluster head must have more energy since it is tasked with gathering data and making calculations.

$$E_{\mathrm{TX}}(l.d) = E_{\mathrm{TX-elec}}(l) + E_{\mathrm{TX-amp}}(l.d) = \begin{cases} lE_{\mathrm{elec}} + l_{\varepsilon \mathrm{fs}} d^2, & d < d_0, \\ lE_{\mathrm{elec}} + l_{\varepsilon \mathrm{amp}} d^4, & d \ge d_0, \end{cases} \tag{1}$$

$$d_0 = \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{\mathrm{amp}}}}, \tag{2}$$

$$E_{\mathrm{RX}}(l) = E_{\mathrm{RX-elec}}(l) = lE_{\mathrm{elec}}. \tag{3}$$

In this regard, E_{elec} is the necessary energy to activate the electronic circuits; ε_{fs} and $\varepsilon_{\mathrm{amp}}$, respectively, are the energies needed to amplify the signals sent to convey a bit in the open space model and multipath model. d is the distance between the sender and receiver node, and d_0 is also the threshold value of the distance obtained from equation (2), After obtaining the value of all these parameters, the value of $E_{\mathrm{TX}}(l.d)$ can be calculated using equation (1).

Equation (3) represents the amount of energy required to receive l bit of data, l is the number of data packets, and E_{elec} is the amount of energy required to amplify a bit of data.

Density: in a real network, nodes in different parts are scattered across different parts; a node chosen as a cluster head must have a lot of congestion and a large number of neighbors.

Centrality: sometimes, the density of a node is high, but the nodes around that node are only listed on one side of the node. When nodes are in the central part of the region, they play an important role in the network structure as the central nodes play an important role in the data in the next stage. Therefore, the cluster head is preferred to increase the load balance, the cluster head in the central neighborhood.

The average distance of the nodes: we first compute the two distances of each node from the other by the equation (4) and then compute the mean distance by equation (5).

$$\text{Distance of nodes} = \sqrt{(X - X_i)^2 + (Y - Y_i)^2}, \tag{4}$$

$$\text{Average distance} = \sum_{i=1}^{n} \frac{\text{Distance of nodes}}{n}. \tag{5}$$

Distance to sink: in most networks, the network lifetime is achieved with low energy consumption in the head clusters. It has a direct relationship with the distance between the cluster head and the sink. By observing the fact that the close proximity of the data stores more energy to transmit data from the cluster head to the sink. Owing to this

Node's ID	Node's energy	Density	Centrality	Average distance of nodes	Distance to sink

Figure 1: Information format of each node in the cluster header.

fact, close ties are more likely to be selected as head clusters, compared to other nodes. After calculating the values of each parameter, we normalize the values with equation (6); after generating unit values, we are placed in the fitness function equation.

$$X_n = \frac{X - X_{\min}}{X_{\max} - X_{\min}}. \tag{6}$$

After the identification phase of the neighboring network nodes, it is divided into head clusters. After saving each of the fields, we take action according to the values registered for each node to select head clusters. Each node has higher energy, higher density, better centrality, lower bound distance, distance to sink, considers with respect to the fitness function equation (7) as the cluster node.

$$\text{Fitness function} = (\alpha \times E_n) + (\beta \times D_n) + (\gamma \times C_n) + (\theta \times \text{Dis}_n) + (\mu \times \text{Diss}_n), \tag{7}$$

$$\alpha + \beta + \gamma + \theta + \mu = 1. \tag{8}$$

In equation (1), E_n is the energy of each node in joules, D_n is the Density of the node, which indicates the number of nodes near the target node, C_n is the centrality of the node, which considers the coordinates of the node, the node in the central neighborhood, Dis_n is the distance between nodes, and Diss_n is distance between the node and the sink.

In Equation (8), a coefficient is selected for each parameter, and the value of factors is selected based on the importance of each weighting factor parameter that the sum of the coefficients is equal to one.

3.2. Selfish Node Detection by the Cluster Head Phase. This phase is based on the results and analysis of the influence of selfish nodes on the general characteristics of the network. The sink in each cluster has the task of monitoring the cluster function to determine whether it occurs in the relevant cluster of selfish behavior. Note that this monitoring is carried out on the performance of network clusters within each network operation by the sink. In this study, only features of the network in the cluster will be examined, which will lead to the existence of a selfish node in the cluster. If the sink is in the distance all the way to the existence of the selfish node in the suspicious cluster, the procedure of detecting the selfish node will be called. The diagnosis procedure of the existence of selfish behavior in the cluster is costly because the sink in each cluster should monitor all its members through the parameters required to distinguish the selfish node and examine their behavior during the period required by the eavesdropping.

Selfish behaviors in parameters such as network throughput, network load, and the number of packets sent/received, packet delay, and other items have an impact. According to the different involved factors which have various units, we need to normalize these metrics. In this phase, the values of the general characteristics of the proposed technique are proposed for the discovery of selfish behavior which represents a combination of these properties for the calculation of the general parameter using equation (9).

$$\text{Examining selfishness} = \alpha^* \frac{1}{\left(D_{\text{present}}/D_{\text{normal}}\right)} + \beta^* \frac{1}{\left(L_{\text{present}}/L_{\text{normal}}\right)} + \gamma^* \frac{1}{\left(S_{\text{present}}/S_{\text{normal}}\right)} + \theta^* \frac{T_{\text{present}}}{T_{\text{normal}}} + \varphi^* \frac{R_{\text{present}}}{R_{\text{normal}}}, \tag{9}$$

$$\alpha + \beta + \gamma + \theta + \varphi = 1. \tag{10}$$

Equation (9) presents the subtitle of measurements and normal subtitles for network behavior in a nonselfish mode and normal mode. The delay parameter estimates mean packet delay with a microsecond unit, load network traffic with (byte/s) unit the total number of data packets received, and the total number of data packets sent, and throughput the average power rate of the network is calculated using the (byte/s) unit. The delay factor, network load, and the total number of data packets sent is as negative communication with the value of examining selfishness. The amount of weight factor must be selected based on the importance of each factor; the sum of these factors is equal to one according to the equation (10). In other words, the factor that differs from ordinary behavior must be given more to strengthen the examining selfishness composition and thus will be found to be selfish behavior. The sink is tasked with calculating examining selfishness, which indicates the cluster in the face of the selfish behavior and the existence of the selfish

node in the network (i.e., the value of examining selfishness is higher than the threshold θ value) to the cluster to run the detection procedure of the selfish node in the cluster and identify the selfish node.

In the selfish cluster detection procedure in each cluster, the local characteristics are calculated by clusters for each cluster member of the selfishness value. Since a node acts as selfish, the number of packets sent by itself is greater compared to the number of packets sent by other nodes, the node prefers to send only its own packets and does not send and perform to other packets of nodes in a selfish manner. Either randomly sending some packets or sending down the packet by delay, so the head clusters of each cluster are discovered by monitoring the performance of cluster members and calculating the relationship of equation (11) in selfish nodes.

$$\text{Selfishness} = \delta^* \frac{S_{\text{total}_{Pi-i}}}{S_{\text{total}_{Pi}}} + \omega^* \frac{1}{\left(D_{\text{node}pi-i}/D_{\text{total}_{pi}}\right)}, \quad (11)$$

$$\delta + \omega = 1. \quad (12)$$

To calculate the existence of selfish behavior in the network, various parameters such as the average packet delay, network traffic, network throughput, and so on are considered, which include other parameters as well, to detect the selfish node in each cluster; the study of two parameters, average packet delay and the packet delivery rate, has provided satisfactory results in the detection of the selfish node. In addition, we can say that the average delay for all packet in all nodes in the network show hidden transmission propagation time and network delay as average delay.

In equation (11), the value of the coefficients, which are weighting factors, to combine the effect of the two packets delay and the number of packets sent by node i is the value of delay with the negative relation the need for the interaction of one of them with the other.in addition, to reflect the real effect of mutual interaction, the first two values are normalized. D is the mean delay packet and micro second unit and S is the total sent data packets. Subtitle ($S_{\text{total}pi}$) mean total data packets except the data packet of node i and subtitle ($S_{\text{total}pi-i}$) mean total data packets of node i except i. In other word, ($D_{\text{total}pi}$) is the total delay in sent data packets of node i with reversed affection and ($D_{\text{total}pi-i}$) is as the amount of sent data packets except the packet itself of node i. Although weight factors indicate the importance of each of these relationships, the number of packets sent from neighbors is more important; therefore, ($\omega < \delta$) is assumed, But the sum of these two weighting factors according to Equation (12) is equal to one. Figure 2 shows the Simi-code related to the diagnosis of selfish behavior in the cluster and the diagnosis of the selfish node in each cluster. In the following, see the Simi-code and flowchart of first phase and second phase of proposed algorithm.

In the proposed method of the Simi-code, HHO first selects a group of sensors which are based on the network sensors. The number of nodes n is the number of sensors starting to send and receive a Hello message to identify their neighbors. The specification of each neighbor is then recorded in the other nodes ' database, these specifications (according to line 5) include node identifier, energy, density, centrality, average distance of nodes, the distance to sink, after obtaining the values of each specification we normalize them (line 6), and then in the fitness function equation (line 7), and among them, we choose the highest values as head clusters. The sink begins to explore the existence of the selfish node in the network. In this way, values D, L, S, T, and R of examining selfishness are kept constantly checking if the values of threshold θ obtained are higher than those of the head clusters. To begin the discovery of selfish nodes in clusters, the clusters begin to compute values of delay, sent packets, and selfishness (line 11) to discover the selfish node.

We are going to have the flowchart of the first and second phases of the proposed model in Figure 3.

The top procedure initially forms the initial population of the HHO; then each of the nodes sends a hello message to each other to identify their neighbors In its database, the information of each of their neighbors contains the following: energy – density – centrality – the distance between the nodes – the distance to the sink. Then the obtained values are used to calculate the fitness function values of each node, and the best is chosen as the cluster head in each cluster, in the continuation of the workflow so that the sink begins to examine the network to detect the existence of a selfish node, hence calculating parameters of load, throughput, sent packets, received packets, delay, and obtained examining selfishness amount. If the obtained value is higher than the threshold Θ value, then it takes place in the self-serving behavior network. In this case, the sink will tell clusters to check their cluster head for the discovery of selfish nodes and the cluster value of delay which each node travels to packets of other nodes in time and also calculate the number of packets sent to other nodes and obtain the amount selfishness in packets.

3.3. The Third Phase (Determining the Reputation of Each Node with the Help of Fuzzy Logic). We need at least four key parameters to determine the reputation of nodes because determining the selfishness of nodes and isolating the nodes is not only by specifying one or two parameters in the proposed method; we use four parameters: number of omitted packets, mean delay, residual energy, and cooperation history as a fuzzy system input to consider the effect of more parameters and more efficient parameters in detecting selfish nodes. In Figure 4, the schematic diagram of the fuzzy system is plotted against four input parameters and ultimately determines the system output that identifies the reputation of each node.

In fuzzy logic, by determining the appropriate input parameters and in sufficient numbers, as well as the number of levels for membership functions, it can provide us with an accurate output. The cluster-head opinion is another factor influencing on correct judgment about the nodes statues; the cluster-head opinion will be combined by the fuzzy logic

Algorithm. Stage 1,2

1. Initialize population of the Harris hawks (i=1 to n)
2. For i=1 to n do
3. Send Hello message
4. Specify the neighbor nodes
5. Specify the properties of each node (Node's ID, Energy, *Density*-Centrality-Distance between nodes -Distance to the sink)
6. Normalize each parameter
7. Calculate Fitness function for each node
8. Choose the node that have best Fitness function as cluster heads
9. Calculating D, L, S, T, R, Examining selfishness by Sink
10. If Examining selfishness >Θ
11. Calculating D, S, selfishness by cluster heads

Figure 2: Simi-code of the first and second phases of the proposed method.

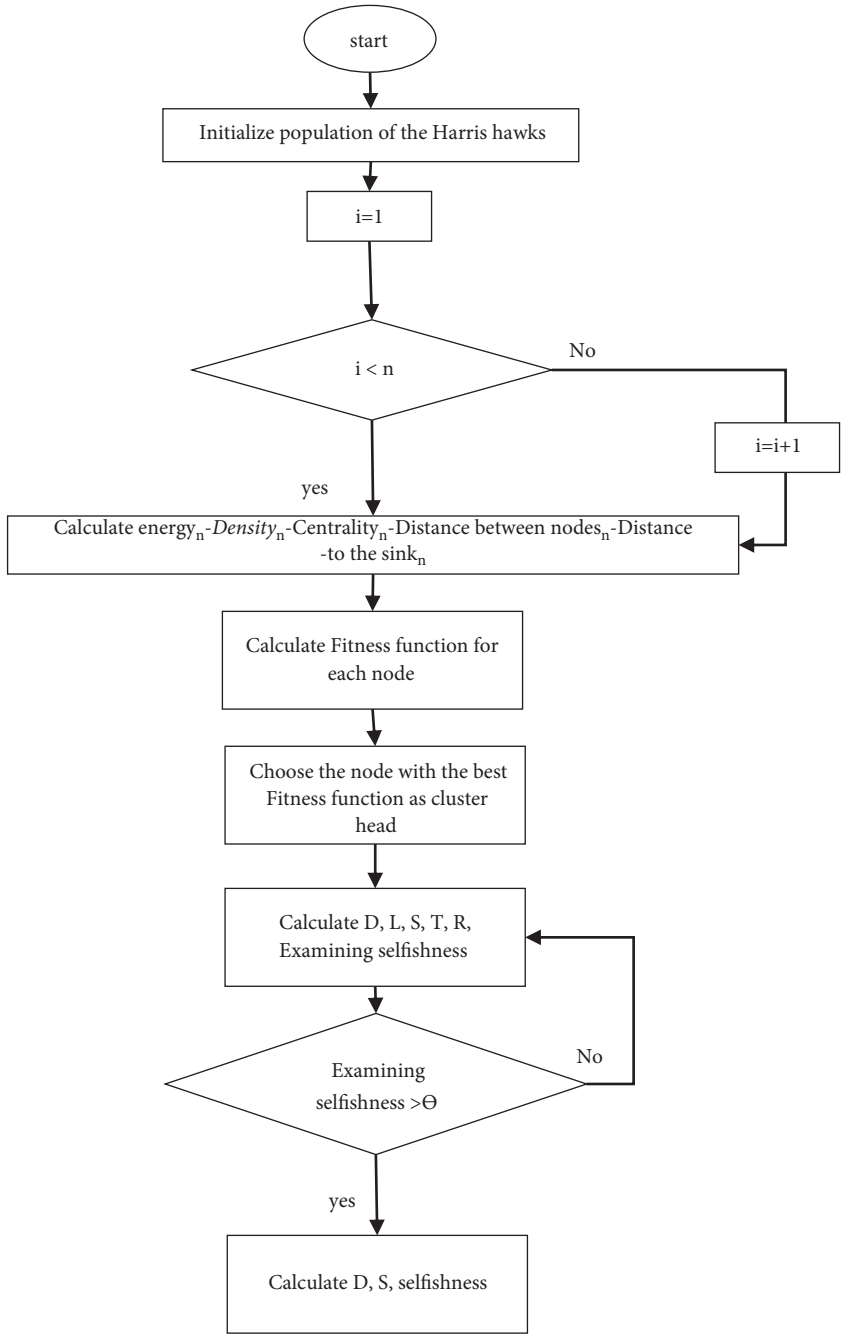

Figure 3: The first and second phases of the proposed method flowchart.

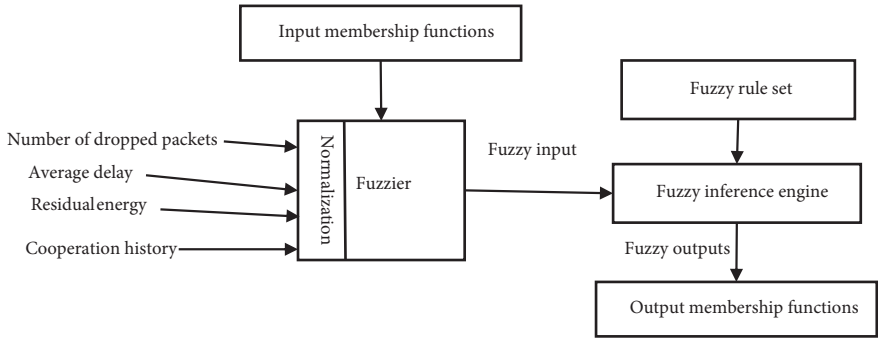

Figure 4: Fuzzy system diagram.

system opinion, and the simulation result has shown high quality in detection accuracy and other metrics.

3.4. The Normalization Process of Numbers. In the decision-making process to facilitate the selection of an appropriate option among existing solutions, the real numbers are first converted into fuzzy terms. In this case, the user will have a clearer understanding of the level of an attribute relative to the extent of its domain. For example, if the number of missed packets can be expressed as a real number, the user cannot judge either too much or lack of it, unless the attribute domain can observe the number of missing packets and then comment on it. However, the expression of the attributes in the form of fuzzy terms helps the user approximate the extent of its value regardless of the values of the variable in domain. The simplest method for converting real numbers into fuzzy terms is to use expert opinions, but it is not always possible because the expert is not always available. Another method of using membership functions is functions that are trapezoidal and triangular.

The amplitude of the input variable is divided into triangular intervals, and each interval represents an expression quantity; the true value of the input variable is converted into a linguistic term, which has the closest distance with the corresponding range. In some of the past work, different fuzzy parts have been used for different features; different features may differ from different types (continuous, ordinal, and relative). The four input parameters include the number of missing packets, the mean of delay, remained energy, cooperation history. We use the Equation (6) to normalize the numbers, after normalization of all numbers in the range [0–1].

We apply the membership functions of four input parameters, considering three levels for each one. For the number of omitted packets, three levels (high, medium, low) of delay average parameters and the third parameter, where the remaining energy of the node has three levels (high, medium, low) ,and the last parameter which defines cooperation history has three levels (strong, medium, weak), whose diagrams are as follows: in the form of membership functions, if we are to compute the distance between the membership functions centers, we use equation (13) and n represents the number of membership functions.

$$\text{Interval} = \frac{\text{Max} - \text{Min}}{n - 1}. \tag{13}$$

The number of membership functions per one of the fuzzy system inputs is 3, so the distance between the membership functions centers is calculated as

$$\text{Interval} = \frac{\text{Max} - \text{Min}}{n - 1} = \frac{1 - 0}{3 - 1} = \frac{1}{2} = 0.5. \tag{14}$$

In Figures 5 and 6, we present three membership functions related to the four fuzzy input parameters (average latency, the number of discarded packets, residual energy, cooperation history) and also the distance between the membership functions which is calculated with the help of equation (14) is equal to 0.5.

3.5. Draw the Fuzzy Rules Base. We multiply the number of membership functions of all inputs, that is, the number of membership functions the numbers of removed packets $*$ the number of membership functions in average of delays $*$ the number of remaining energy membership functions $*$ the number of functions of the cooperation history $3 * 3 * 3 * 3 = 81$.

To map the fuzzy rules base $3 * 3 * 3 * 3 * 3$, we write different scenarios of each input parameters. This table of rules indicates how much the reputation of each node acts as a self-serving which must be isolated or cooperating.

The degree of selfishness of each node is determined by its behavior, the amount of each of the variables, like delay in sending packets, the number of dropped packets, the residual energy, and the history of cooperation determine the degree of cooperation or non-cooperation of the nodes.

Table 1, which is the fuzzy rules table, shows that 81 different modes of nodes parameters conditions indicate that VVVH is the best node (associated node) in the network. Thus, in the last row, the VVVL node is selfish node in the network, which the cluster head node will decide. The purpose of this system is to identify the reputation of each node. R output variable defines the degree of selfishness of each node and allocates different values that represent different modes of reputation of each node according to Table 2.

After gaining the reputation of each node by fuzzy system (Phase III) and gathering comments from the second

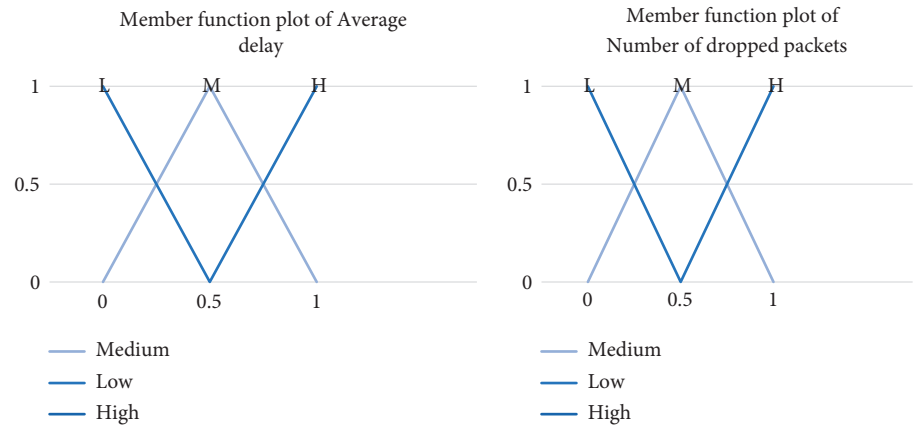

Figure 5: Average delay and Number of dropped packets member function chart.

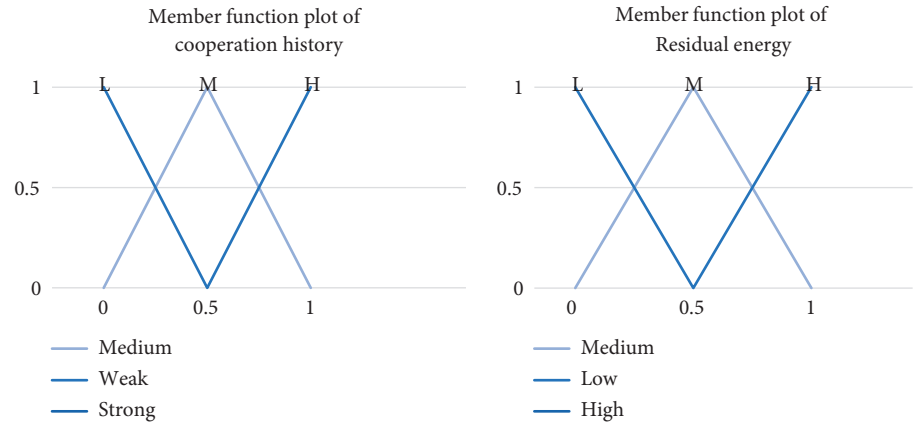

Figure 6: Cooperation history and residual energy member function chart.

Table 1: Fuzzy role database.

Average delay Number of dropped packets	Residual energy cooperation history	Strong	Medium	Weak
Low Low	High	VVVH	VVH	VH
	Medium	VVH	VH	H
	Low	VH	H	VVVG
Low Medium	High	H	VVVG	VVG
	Medium	VVVG	VVG	VG
	Low	VVG	VG	G
High	High	VG	G	M
	Medium	G	M	L
	Low	M	L	VL
Low Medium	High	VVH	VH	H
	Medium Low	VH	H	VVVG
	Low	H	VVVG	VVG
Medium Medium	High	VVVG	VVG	VG
	Medium	VVG	VG	G
	Low	VG	G	M
High Medium	High	G	M	L
	Medium	M	L	VL
	Low	L	VL	VVL

Table 1: Continued.

Average delay Number of dropped packets	Residual energy cooperation history	Strong	Medium	Weak
Low Low	High	VH	H	VVVG
	Medium	H	VVVG	VVG
	Low	VVVG	VVG	VG
Medium Low	High	VVG	VG	G
	Medium	VG	G	M
	Low	G	M	L
High Low	High	M	L	VL
	Medium	L	VL	VVL
	Low	VL	VVL	VVVL

Table 2: Output fuzzy sets rang-reputation rate of each node.

Symbol	Description
Very very high (VVH)	Having maximum reputation (100% cooperative node)
Very high (VH)	Having above reputation (100% cooperative node)
High (H)	Having high reputation (100% cooperative node)
Medium (M)	Having average reputation (maybe cooperative node, need to be checked)
Low (L)	Having low reputation (having selfish behavior)
Very low (VL)	Having very low reputation (having selfish behavior)
Very very low (VVL)	Having little reputation (having selfish behavior)

phase (nodes that call the cluster head as selfish) to decide to end each node the results of the fuzzy system and the ideas of cluster heads together as shown in Table 3.

In the following, we observed the Simi code and flowchart of the proposed method, which are presented in Figures 7 and 8, respectively.

We observed the Simi code and flowchart of the proposed method, which are presented in Figures 7 and 8, respectively. The third phase of the proposed method first (line 1) specifies the input parameters and the values from these parameters are determined for each node, and then these values are normalized in [1, 0]. In the following (line 7) for each input parameters, the number of levels (number of membership functions) and distance between the levels are calculated and then the membership functions are plotted for each input parameter (line 10). We calculate the number of membership functions and show in the fuzzy rules base table (line 13). The cluster head theory and fuzzy system are then combined into each other and isolate in the face of the full selfishness of the node (line 16), and in the face of low selfishness, the opportunity node will be reestablished.

The procedure in Figure 7 at the beginning of the round identifies four input parameters and then the values of each parameter are normalized in [1, 0]. In the next step, the membership functions for each input parameter are determined and then the membership functions for each inputs are plotted with respect to the number of membership functions; then the number of rules specified and each of the rules is represented in the fuzzy rules base and the degree of reputation of each state becomes known. The cluster is then combined with each node and each state of the fuzzy system in the case of nodes together, and in the case of full selfishness detection, the node will be isolated and will continue to operate in the form of less selfishness, it will give a new chance and if it works, the node continues to operate.

4. Evaluations and Simulation Results

A network of Internet of things is distributed in an environment of 1000 ∗ 1000 square meters that are uniform and random, and nodes for 4 types of networks and Internet nodes are different things with different numbers and parameters in the environment. In the simulation performed at the center of all four types of network types, a data collection station is located and the network of Internet of things is considered to include nonmobility things and with a limited energy source similar to that of wireless sensor networks with four different types of nodes. All of these networks have wireless communications. Note that each of these different networks has a different simulation model. Therefore, in the first place, we will have a clustering by specifying head clusters in accordance with the HHO clustering algorithm [22] as shown in Figure 9.

To perform simulation at the center of each type of network, four data collection stations are considered. In different regions, the base station is fixed and the positions are considered as (250–250), (750–750), (250–750), and (750–250). Simulation has been repeated over 100 iterations, and the initial energy for network nodes is equal to 0.5, 10, 1, 150 J, and 200, 100, 150, 20 and with radio range of 80, 70, 90, 70 m, respectively. The energy consumption model and the type of nodes are similar. Table 4 shows the parameters used in this simulation.

4.1. Evaluating the Performance of the Proposed Algorithm. To investigate the effectiveness of the proposed algorithm, the results obtained from the simulation, which are implemented in an operating system of 8.1 with Intel(R) core (TM) i7 with speeds of 4.2 GHz and internal memory 16 GB, were implemented in MATLAB 2018 software background.

Table 3: Fuzzy output and cluster head output situation.

Cluster head decision fuzzy logic output	Cooperative	Non-cooperative
VVH	Cooperative	Given second chance
VH	Cooperative	Given second chance
H	Cooperative	Given second chance
M	Given second chance	Given second chance
L	Selfish	Selfish
VL	Selfish	Selfish
VVL	Selfish	Selfish

Algorithm. Stage 3

1. STEP 0: Specify the number of inputs and the amount of each input for each node (Number of dropped packets, Average delay, Residual energy, Cooperation history)
2. For i=1 to n do
3. Specify the values of each parameter for nodes
4. Normalize the values of each parameter for nodes
5. End
6. for i=1 to Number of inputs do
7. Specify the levels of each input
8. Specify the interval of each level
9. Calculate the interval between the centers of membership functions
10. Draw a membership function diagram for each input
11. Calculate the number of rules
12. End
13. Create a fuzzy rule database table
14. Combining cluster head and fuzzy logic decisions
15. Specify cooperative and selfish nodes
16. Isolate or give second chance to the selfish nodes

Figure 7: The Simi-code of proposed methods.

The effectiveness of the proposed strategy is evaluated and its efficiency performance is evaluated. For this purpose, accuracy parameters of nonassociated nodes detection, the positive warning rate, successful delivery rate, and mean of end-to-end delay algorithms as acknowledgment-based [32], TEEM [31], and game theory based [28] are compared with the proposed method. The Fuzzy system is designed, and each metric is described in detail with the corresponding diagram.

4.2. Assessment of the Designed Fuzzy System. The fuzzy system is designed with four inputs and an output. The number of missing data packets, the average packet delay, and the remaining energy of the nodes and the association history of the nodes are in forwarding packets. The output of the system shows the degree and level of interoperability within the network, which are stored in cluster head nodes. The proposed protocol can determine the highest level of cooperation of nodes and their lowest cooperation level. The input and output of the fuzzy system and its maximum and minimum are shown in Figure 10 and Table 5, respectively.

Doing Fuzzy on inputs and outputs: the triangular membership functions are used to make fuzzy the inputs. For different inputs, the fuzzy variable and its input range are shown in Table 5. The four input parameters include the number of missing packets, the mean of delay, remained energy, and cooperation history. In other words, the amount of linguistic variables can range from that domain. The number of discarded packets by each node as a linguistic variable with the range of *?* (Number of dropped packets) T, U [800, 1] domain is 1 to 800 packets. The range is divided into three levels. Each variable in the number of dropped (T) packet is described by a fuzzy set, such as $U = [800, 1]$. Therefore, each level has a fine function that represents the degree of dependency of each value to this level.

The optimization of these algorithms is often done through test and error method to achieve the desired performance of the designed fuzzy system, but there is only one output, the level of collaboration of the node, and, like the inputs, the membership function is assigned as a triangular function. In Figure 11 membership functions of the inputs of the fuzzy system are represented.

The fuzzy inference method in MATLAB is used to determine the level of cooperation of network nodes and so-called MAMDANI (max-min) method is applied. The fuzzy rules are represented in the optimal performance of the system proportional to the inputs and output of the system in Figure 12. The output, which is node collaboration level for different inputs, for example, remained energy level, removed packets, and cooperation node history (Figure 13).

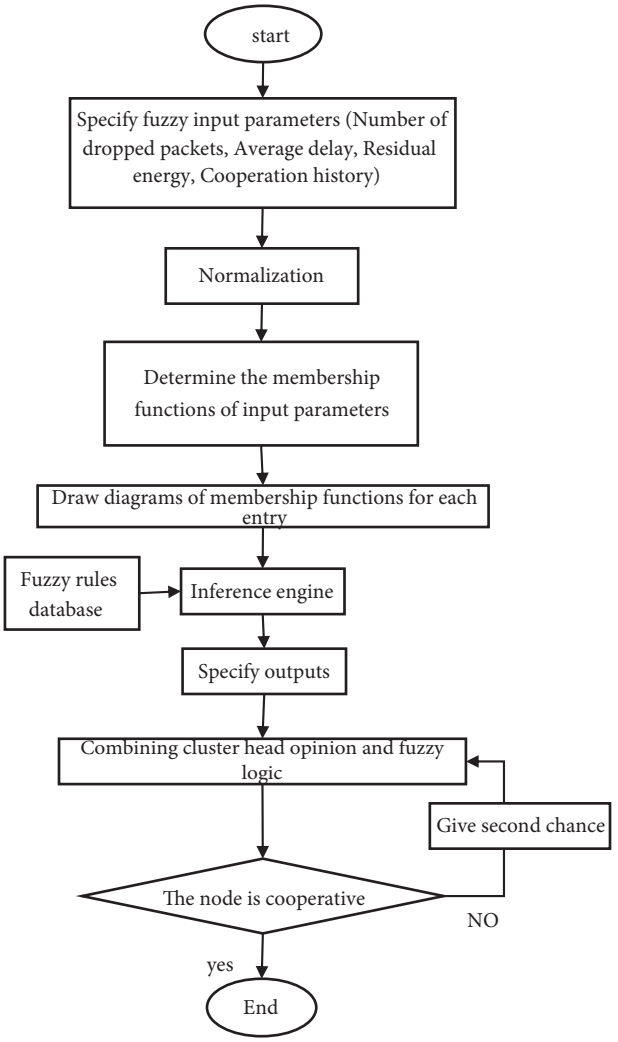

Figure 8: The flowchart of proposed methods.

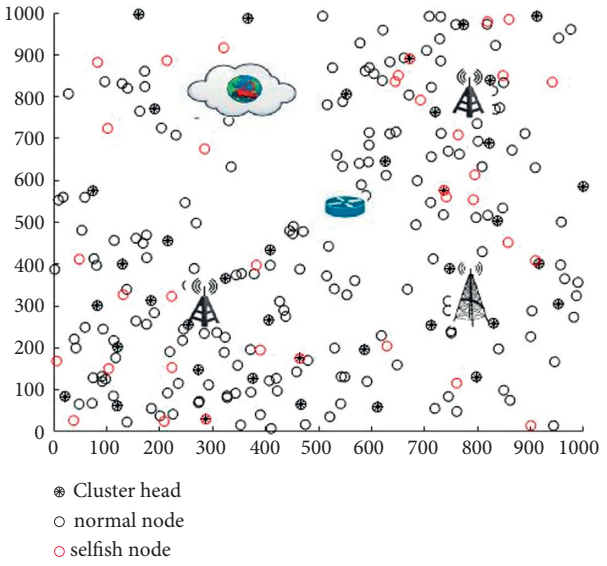

Figure 9: The simulation network area.

Table 4: Simulation parameters.

Parameters	Values
Simulation time	100 rounds
Number of nodes	470
Number of selfish nodes	10%–40%
Network size	1000 ∗ 1000
Transmission range	70–90 m
Traffic type	Constant bit rate (CBR)
Initial energy of nodes	0.5 joule 10 joule 1 joule 15 joule

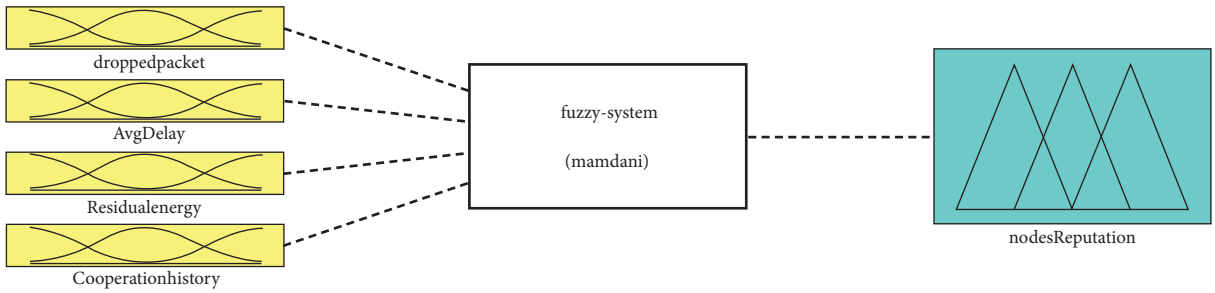

Figure 10: Fuzzy system model with inputs and outputs.

Table 5: Input parameters and their interval in the proposed fuzzy system.

Level	Interval of each level
Number of dropped packets	
L1 (low)	1–200
L2 (medium)	200–700
L3 (high)	700–800
Average delay	
L1 (low)	1–20
L2 (medium)	20–40
L3 (high)	40–60
Residual energy of nodes	
L1 (low)	$0.1j$–$0.5j$
L2 (medium)	$0.5j$–$0.8j$
L3 (high)	$0.8j$–$1j$
History of nodes 'cooperation	
L1 (weak)	1–40
L2 (medium)	40–80
L3 (strong)	80–100

4.3. Detection Accurate (DA) of the Selfish Node. Detection Accurate (DA) of the selfish node represents the number of normal nodes detected to the total selfish and associate nodes in the network. Thus, if TP represents the number of discovered normal nodes, FN indicates the number of selfish nodes found incorrectly as associated. Equation (15) shows the accuracy of diagnosing the selfish node.

$$DR = \frac{T_p}{\left(T_p + F_N\right)}. \tag{15}$$

To investigate the important parameter of the discovery of the selfish node in the Internet network of things, it is assumed at the beginning of the simulation of 10% of the total selfish network nodes, while the rate of selfish nodes gradually increased by 15%, 20%, . . . to 40%. However, it is noted that in the real world, the nodes tend to maintain energy and avoid sending packets to other nodes on the initial energy level, and they will be as selfish node.

Accuracy is shown in Figure 14 clearly by increasing the accuracy of the discovery of the selfish node. The proposed method is performed by increasing the percentage of selfish nodes in the network number of more games in order to update the reputation of each node during the games and crossing the different nodes. For this reason, when only 10% of the network nodes are selfish nodes, there is less game to prevent greater energy loss, and the popularity associated with the network nodes is not sufficient and only

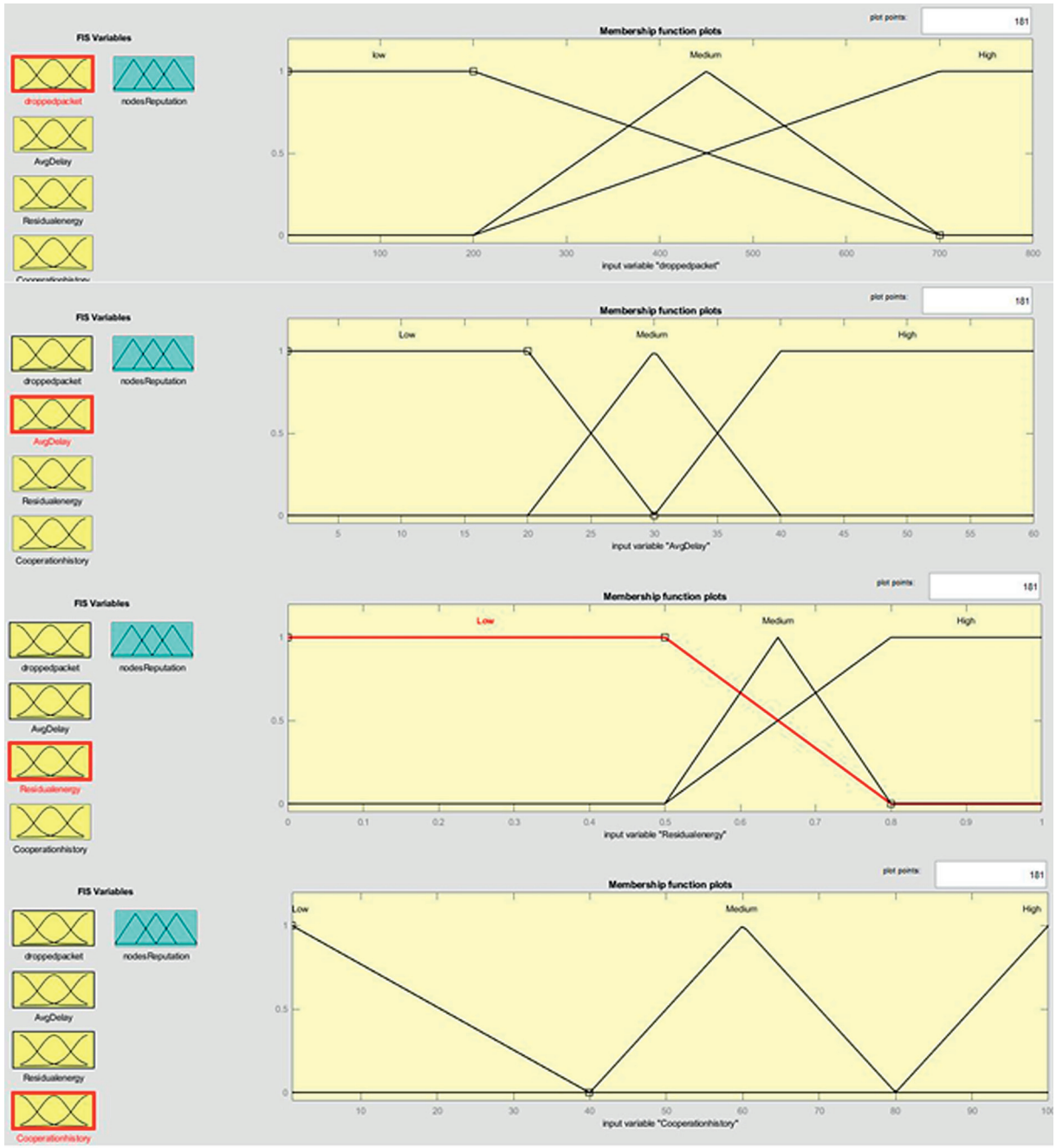

Figure 11: Fuzzy system of the input membership function.

92% of selfish nodes have been identified, thus increasing the percentage of selfish nodes in the network and increasing the number of games, the reputation of nodes updates and recognition is performed correctly, and even to 100% selfish nodes will be identifiable; But the increase in the number of selfish nodes will be subject to many games that will not be reasonable with respect to energy consumption, and the number of games can also be a reasonable percentage of the diagnosis of selfish nodes even with a percentage of the presence of selfish nodes. The numerical values of Game theory-based [28], acknowledgment - based [32], TEEM [31] algorithms show that by increasing the percentage of selfish nodes in the network, the accuracy of the algorithm is higher than the other algorithms and at the same time, it has a lenient slope compared to other similar methods for the discovery of the Selfish node while other methods have a steeper slope with increasing the existence of selfish nodes. The fact that the proposed method has a gentler slope compared to other existing methods is clearly visible because the method uses the general characteristics of the network in each cluster to uncover the selfish nodes and check the recognition by using fuzzy rules to check the accuracy of detecting selfish nodes. Whereas other methods in higher percentages of the number of selfish nodes are usually incapable of detecting them in a high percentage.

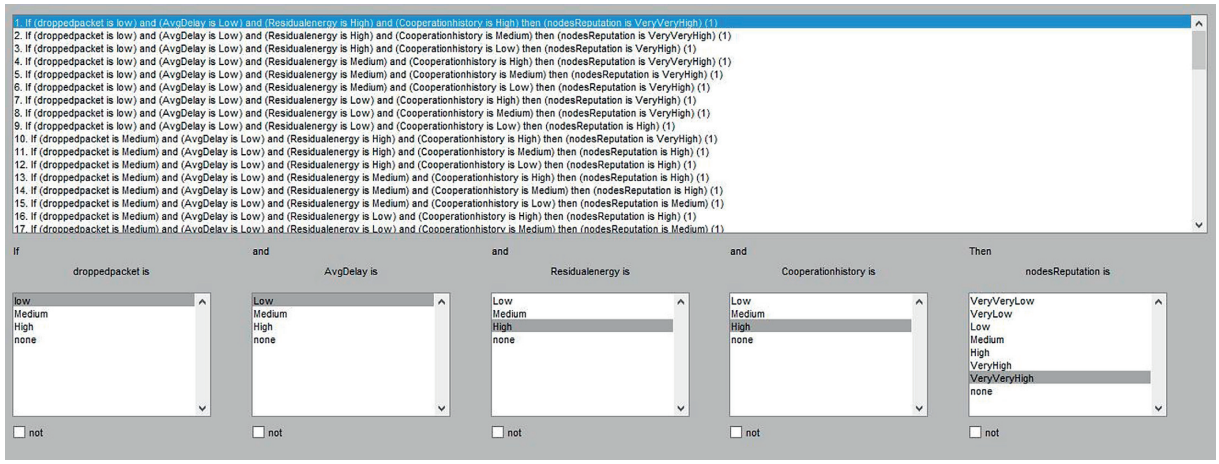

Figure 12: Fuzzy system rules.

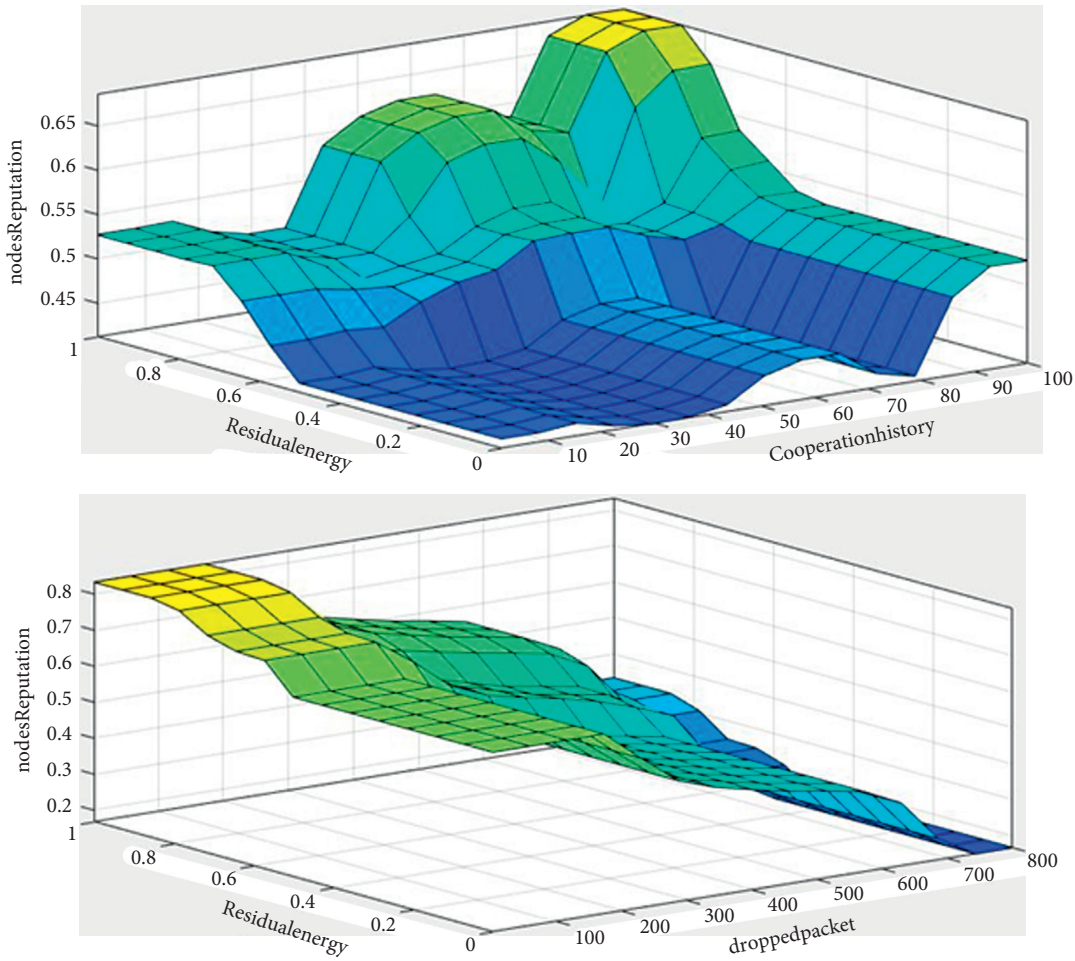

Figure 13: Fuzzy system of different inputs relation.

In the game theory-based methods, their computational overhead and the energy consumption are high. This method also requires playing several games in order to be able to detect the selfish node in the network, which is a time overhead. But the proposed method has no computational and time overhead.

4.4. False-Positive Rate. This metric shows the ratio of the number of normal nodes found to be selfish incorrectly on the sum of the number of normal nodes found incorrectly (FP)and the number of normal nodes found correctly (TP) in the network. Equation (16) shows a positive warning rate. This parameter actually has a photo relationship so that its

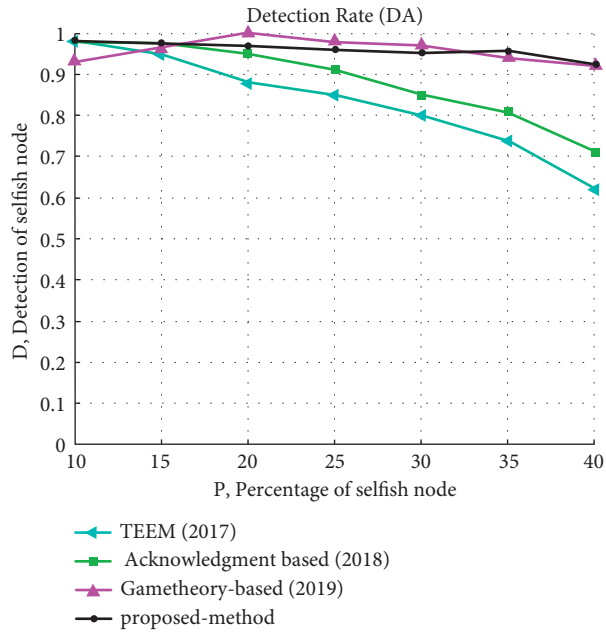

Figure 14: Detection Accuracy of proposed method.

level of its level accuracy indicates the accuracy of the presented approach. That is, the higher true detection of the number of associates nodes, the higher the performance of the network, since the time network detect a node as a selfish node, avoids cooperating with that node, so if this diagnosis is a mistake, it will reduce the efficiency and delivery of less data packets to the destination.

$$\text{FPR} = \frac{FP}{TP + FP}. \tag{16}$$

The implementation of the algorithm is repeated in such a cycle so that the performance of the algorithm is evaluated in different states. The numerical values of the comparisons show that by increasing the existence of selfish nodes in the network, the positive warning rate of the proposed algorithm is lower than other algorithms. As indicated in the diagram, the positive warning rate exceeds 25% higher than the other algorithms and has fewer errors than the other algorithms, but it operates by 25% on the level of the proposed methods of acknowledgment-based [32] TEEM [31] and Game theory–based [28] algorithms, and its positive warning rate is lower than the other two algorithms. Other methods make more errors by increasing the percentage of selfish nodes. This reduces the throughput and efficiency of the network. The results have shown in Figure 15.

4.5. Data Packet Delivery Rate. The delivery rate of successful data packets is one of the most important parameters of network evaluation in most working areas in the context of the networks of Internet of things. This parameter is basically the number of packets delivered successfully during the routing and transmission process from source to destination, so the packet delivery rate is the average number of packets delivered to the destination from all network nodes to the total number of packets generated in the network.

$$\text{PDR} = \frac{R_i}{S_i}. \tag{17}$$

Equation (17) shows that R_i is the total number of packets received in the destination nodes and S_i is the total number of packets transmitted at the origin nodes. This metric has a significant impact on network performance. The higher the number of packets received at the destination, the lower delay on the number of packets arriving at the destination, the network delivery rate increases. For when the selfish nodes do not send the data packets in the network and the confirmation message is not received at the source, the node is again sent back in the network, increasing the traffic and the total number of packets produced and sent in the network, which has been ineffective. Therefore, the higher the packet delivery rate in the network, the more efficient use of the network resources include the bandwidth or limited energy resources of the nodes. In Figure 16, the high performance of the proposed approach is found to be due to early detection of selfish nodes in the network, which avoid repetitive data packet production and increase network traffic and delay in nodes.

4.6. The Average of End-to-End Delay. The average time period, which takes the length of the data packet from the source to the destination, is denoted by the time unit of the second. The mean delay is as time for storing data packets from the sender to the receiver as the scale of the different networks. This time is actually the total amount of time spent by the sender with different steps from relay nodes to reach the receiver and it has calculated by

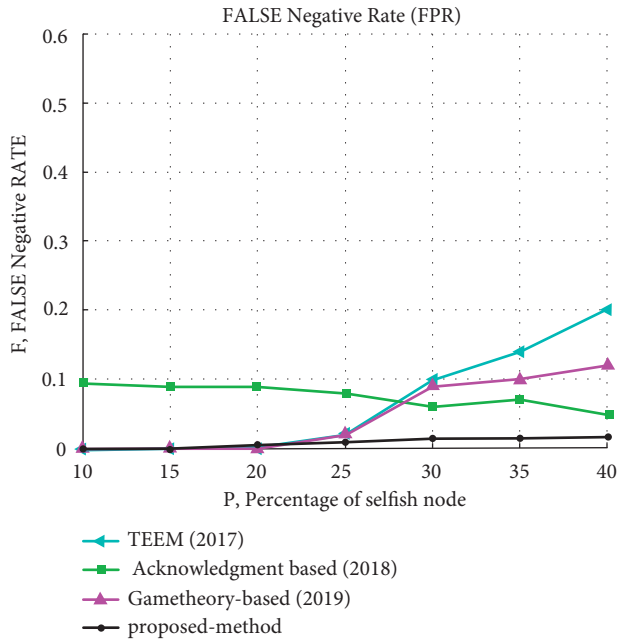

Figure 15: False Negative Rate (FNR) of proposed method.

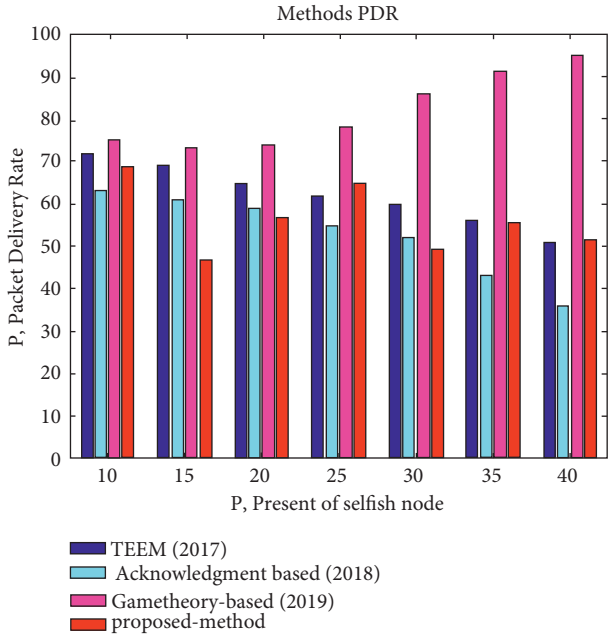

Figure 16: Packet Delivery Rate (PDR) of proposed method in comparison with other methods.

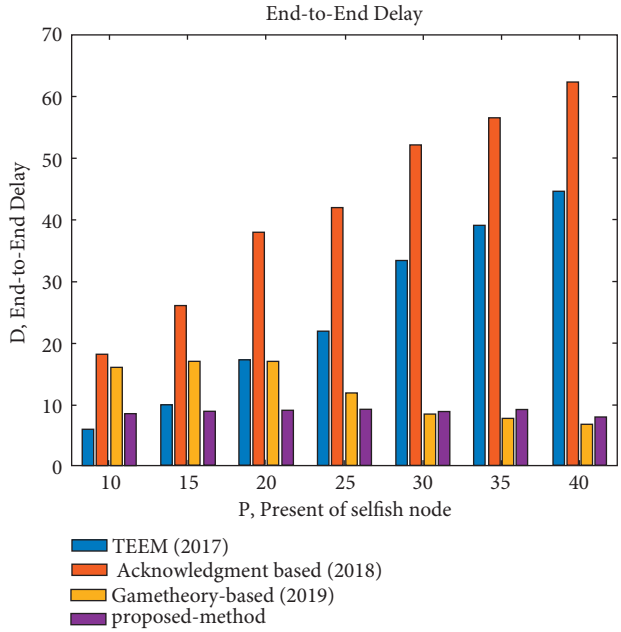

Figure 17: Average end-to-end Delay of proposed method.

$$\text{end\_to\_end delay} = \frac{\text{sum of time taken packet to recieve destination}}{\text{number of recieved packet}}. \tag{18}$$

As Figure 17 shows the mean delay in the proposed method, especially in the high percentages of the selfish node in the network, is less than the other methods. The reason for this is that performance and transmission scale in the proposed method are higher than other methods and because there is less traffic on the network, the average packet delivery time on the network and the average waiting time is reduced to network nodes.

Some selfish nodes in the network are associated with the nature of malice in the network which will increase the delay of end to the end at network nodes, As the packet lasts until it is alive, the packet remains in the buffer middle node, and then it is sent away, which causes the closure of the packet to be discarded at the other node. Since the time of survival of packet is over, the same issue increases the mean delay end to end at the network nodes. Obviously, the less time the packet delivery time is delivered to the network, and it is actually the desired goal of the network.

5. Conclusion and Future Works

This paper presents a new hybrid method that is used to identify the selfish node on the Internet of things. The proposed method will benefit from the advantages of the Harris Hawks Optimization (HHO) and fuzzy logic, it operates in three stages of trying to discover selfish nodes and decide how to deal with those nodes in a way that the node is isolated or given the opportunity again. The performance of this method is tested in the network and compared with acknowledgment -based [32], Game theory based [28], TEEM [31] algorithms. The results showed that the proposed method significantly identifies the selfish nodes and prevents the consumption of node resources (energy, battery, memory) and ending the delay end to end of data packets to reach the destination. The average throughput which is the percentage of successful delivery of data packets to the destination is 20% and is reduced by 12% at the same time the end - to - end delay at the end of the data packets. The percentage of selfish nodes in comparison with the same methods has increased by 10% and the false positive and false negative rate decreased 8% that indicates the accuracy of the selfish nodes detection accuracy. In future work, we will try to add more parameters that are effective in node reputation as input parameters to our fuzzy system as well as other Metaheuristic algorithms with higher performance to choose cluster nodes.

Data Availability

This manuscript has no data availability statement.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Z. K. Zhang, M. C. Y. Cho, C. W. Wang, C. W. Hsu, C. K. Chen, and S. Shieh, "IoT security: ongoing challenges and research opportunities," in *Proceedings of the 2014 IEEE 7th international conference on service-oriented computing and applications*, pp. 230–234, IEEE, Matsue, Japan, 2014 November.

[2] A. Seyfollahi and A. Ghaffari, "Reliable data dissemination for the Internet of Things using Harris hawks optimization," *Peer-to-Peer Networking and Applications*, vol. 13, no. 6, pp. 1886–1902, 2020.

[3] S. K. Mousavi, A. Ghaffari, S. Besharat, and H. Afshari, "Improving the security of internet of things using cryptographic algorithms: a case of smart irrigation systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 2033–2051, 2021.

[4] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A survey on the internet of things (IoT) forensics: challenges, approaches, and open issues," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1191–1221, 2020.

[5] A. Selvaraj, R. Patan, A. H. Gandomi, G. G. Deverajan, and M. Pushparaj, "Optimal virtual machine selection for anomaly detection using a swarm intelligence approach," *Applied Soft Computing*, vol. 84, Article ID 105686, 2019.

[6] A. J. Manuel, G. G. Deverajan, R. Patan, and A. H. Gandomi, "Optimization of routing-based clustering approaches in wireless sensor network: review and open research issues," *Electronics*, vol. 9, no. 10, Article ID 1630, 2020.

[7] M. Mohammed Sadeeq, N. M. Abdulkareem, S. R. M. Zeebaree, D. Mikaeel Ahmed, A. Saifullah Sami, and R. R. Zebari, "IoT and Cloud computing issues, challenges and opportunities: a review," *Qubahan Academic Journal*, vol. 1, no. 2, pp. 1–7, 2021.

[8] L. Krishnasamy, R. Dhanaraj, D. Ganesh Gopal, T. Reddy Gadekallu, M. Aboudaif, and E. Abouel Nasr, "A heuristic angular clustering framework for secured statistical data aggregation in sensor networks," *Sensors*, vol. 20, no. 17, Article ID 4937, 2020.

[9] S. Nižetić, P. Šolić, D. López-de-Ipiña González-de-Artaza, and L. Patrono, "Internet of Things (IoT): opportunities, issues and challenges towards a smart and sustainable future," *Journal of Cleaner Production*, vol. 274, Article ID 122877, 2020.

[10] D. G. Gopal and R. Saravanan, "Selfish node detection based on evidence by trust authority and selfish replica allocation in DANET," *International Journal of Information and Communication Technology*, vol. 9, no. 4, pp. 473–491, 2016.

[11] D. Glaroudis, A. Iossifides, and P. Chatzimisios, "Survey, comparison and research challenges of IoT application protocols for smart farming," *Computer Networks*, vol. 168, Article ID 107037, 2020.

[12] D. G. Gopal and R. Saravanan, "Fuzzy based energy aware routing protocol with trustworthiness for MANET," *International Journal of Electronics and Information Engineering*, vol. 3, no. 2, pp. 67–80, 2015.

[13] T. Bartlett, *Privacy and Security Management Practices of Emerging Technologies: Internet of Things (Doctoral Dissertation*, Robert Morris University, Coraopolis, PA, USA, 2020.

[14] S. Nobahary, H. Gharaee Garakani, A. Khademzadeh, and A. M. Rahmani, "ISOT: distributed selfish node detection in internet of things," *International Journal of Information and Communication Technology Research*, vol. 10, no. 3, pp. 19–30, 2018.

[15] D. G. Gopal and R. Saravanan, "A novel trust based system to detect the intrusive behavior in MANET," *International Journal of Electronics and Information Engineering*, vol. 3, no. 1, pp. 31–43, 2015.

[16] Y. Lv, "Security issues in multi-hop device-to-device communication networks - secure routing protocols solution," in *Journal of Physics: Conference Series,* vol. 1828, no. 1, Article ID 012117, IOP Publishing, 2021.

[17] A. Shan, X. Fan, C. Wu, X. Zhang, and S. Fan, "Quantitative study on the impact of energy consumption based dynamic selfishness in MANETs," *Sensors*, vol. 21, no. 3, p. 716, 2021.

[18] A. Seyfollahi and A. Ghaffari, "A lightweight load balancing and route minimizing solution for routing protocol for low-power and lossy networks," *Computer Networks*, vol. 179, Article ID 107368, 2020.

[19] A. Vij, V. Sharma, and P. Nand, "Selfish node detection using game theory in MANET," in *Proceedings of the 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pp. 104–109, IEEE, Greater Noida, India, 2018, October.

[20] F. Afghah, A. Shamsoshoara, L. Njilla, and C. Kamhoua, "A reputation-based stackelberg game model to enhance secrecy rate in spectrum leasing to selfish iot devices," in *Proceedings of the IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 312–317, IEEE, Honolulu, HI, USA, 2018, April.

[21] R.-I. Ciobanu, C. Dobre, M. Dascălu, Ş. Trăuşan-Matu, and V. Cristea, "Sense: a collaborative selfish node detection and incentive mechanism for opportunistic networks," *Journal of Network and Computer Applications*, vol. 41, pp. 240–249, 2014.

[22] Y. Mao, C. Zhou, J. Qi, and X. Zhu, "A fair credit-based incentive mechanism for routing in DTN-based sensor network with nodes' selfishness," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, pp. 1–18, 2020.

[23] S. Subramaniyan, W. Johnson, and K. Subramaniyan, "A distributed framework for detecting selfish nodes in MANET using Record-and Trust-Based Detection (RTBD) technique," *EURASIP Journal on Wireless Communications and Networking*, vol. 205, no. 1, pp. 1–10, 2014.

[24] R. I. Ciobanu, C. Dobre, M. Dascalu, S. Trausan-Matu, and V. Cristea, "Collaborative selfish node detection with an incentive mechanism for opportunistic networks," in *Proceedings of the 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, pp. 1161–1166, IEEE, Ghent, Belgium, 2013, May.

[25] S. Kumar, "Detecting and avoiding selfish nodes in delay tolerant networks (DTNs)," *International Journal of Recent Research Aspects*, vol. 5, pp. 325–329, 2018.

[26] B. U. I. Khan, F. Anwar, R. F. Olanrewaju, B. R. Pampori, and R. N. Mir, "A game theory-based strategic approach to ensure reliable data transmission with optimized network operations in futuristic mobile adhoc networks," *IEEE Access*, vol. 8, pp. 124097–124109, 2020.

[27] M. Ponnusamy, "Detection of selfish nodes through reputation model in mobile adhoc network-MANET," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 9, pp. 2404–2410, 2021.

[28] S. Nobahary, H. G. Garakani, A. Khademzadeh, and A. M. Rahmani, "Selfish node detection based on hierarchical game theory in IoT," *EURASIP Journal on Wireless Communications and Networking*, vol. 255, no. 1, pp. 1–19, 2019.

[29] S. Nobahary and S. Babaie, "A credit-based method to selfish node detection in mobile ad-hoc network," *Applied Computer Systems*, vol. 23, no. 2, pp. 118–127, 2018.

[30] H. Hasani and S. Babaie, "Selfish node detection in ad hoc networks based on fuzzy logic," *Neural Computing & Applications*, vol. 31, no. 10, pp. 6079–6090, 2019.

[31] A. Lupia, C. A. Kerrache, F. De Rango, C. T. Calafate, J. C. Cano, and P. Manzoni, "TEEM: trust-based energy-efficient distributed monitoring for mobile Ad-Hoc networks," in *Proceedings of the 2017 Wireless Days*, pp. 133–135, IEEE, Porto, Portugal, 2017 Mar.

[32] M. Bounouni and L. Bouallouche-Medjkoune, "Acknowledgment-based punishment and stimulation scheme for mobile ad hoc network," *The Journal of Supercomputing*, vol. 74, no. 10, pp. 5373–5398, 2018 Oct.

[33] S. Usha and S. Radha, "Co-operative approach to detect misbehaving nodes in MANET using multi-hop acknowledgement scheme," in *Proceedings of the 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies*, pp. 576–578, IEEE, Bangalore, India, 2009, December.

[34] L. E. Jim and M. A. Gregory, "Improvised MANET selfish node detection using artificial Immune system based decision tree," in *Proceedings of the 2019 29th International Telecommunication Networks and Applications Conference (ITNAC)*, pp. 1–6, IEEE, Auckland, New Zealand, 2019, November.

[35] O. Uviase and G. Kotonya, "IoT architectural framework: connection and integration framework for iot systems," 2018, https://arxiv.org/abs/1803.04780.

[36] K. K. Patel and S. M. Patel, "Internet of things-IOT: definition, characteristics, architecture, enabling technologies, application & future challenges," *International journal of engineering science and computing*, vol. 6, no. 5, 2016.

[37] A. Srivastava, S. K. Gupta, M. Najim, N. Sahu, G. Aggarwal, and B. D. Mazumdar, "DSSAM: digitally signed secure acknowledgement method for mobile ad hoc network," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, pp. 1–29, 2021.

[38] F. Olivier, G. Carlos, and N. Florent, "New security architecture for IoT network," *Procedia Computer Science*, vol. 52, pp. 1028–1033, 2015.

[39] A. Khanna and S. Kaur, "Internet of things (IoT), applications and challenges: a comprehensive review," *Wireless Personal Communications*, vol. 114, no. 2, pp. 1687–1762, 2020.