PLOS ONE

RESEARCH ARTICLE

A cluster-tree-based trusted routing algorithm using Grasshopper Optimization Algorithm (GOA) in Wireless Sensor Networks (WSNs)

Mehdi Hosseinzadeh[1,2,3], Omed Hassan Ahmed[4], Jan Lansky[5], Stanislava Mildeova[5], Mohammad Sadegh Yousefpoor[6], Efat Yousefpoor[6], Joon Yoo[7], Lilia Tightiz[7]*, Amir Masoud Rahmani[8]*

1 Institute of Research and Development, Duy Tan University, Da Nang, Vietnam, **2** School of Medicine and Pharmacy, Duy Tan University, Da Nang, Vietnam, **3** Department of Computer Science, University of Human Development, Sulaymaniyah, Iraq, **4** Department of Information Technology, University of Human Development, Sulaymaniyah, Iraq, **5** Department of Computer Science and Mathematics, Faculty of Economic Studies, University of Finance and Administration, Prague, Czech Republic, **6** Department of Computer Engineering, Dezful Branch, Islamic Azad University, Dezful, Iran, **7** School of Computing, Gachon University, Seongnam, Korea, **8** Future Technology Research Center, National Yunlin University of Science and Technology, Yunlin, Taiwan

* rahmania@yuntech.edu.tw (AMR); liliatightiz@gachon.ac.kr (LT)

OPEN ACCESS

Citation: Hosseinzadeh M, Ahmed OH, Lansky J, Mildeova S, Yousefpoor MS, Yousefpoor E, et al. (2023) A cluster-tree-based trusted routing algorithm using Grasshopper Optimization Algorithm (GOA) in Wireless Sensor Networks (WSNs). PLoS ONE 18(9): e0289173. https://doi.org/10.1371/journal.pone.0289173

Editor: Praveen Kumar Donta, TU Wien: Technische Universitat Wien, AUSTRIA

Received: April 18, 2023

Accepted: July 12, 2023

Published: September 8, 2023

Data Availability Statement: All relevant data are within the paper.

Funding: The author(s) received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

Abstract

In wireless sensor networks (WSNs), existing routing protocols mainly consider energy efficiency or security separately. However, these protocols must be more comprehensive because many applications should guarantee security and energy efficiency, simultaneously. Due to the limited energy of sensor nodes, these protocols should make a trade-off between network lifetime and security. This paper proposes a cluster-tree-based trusted routing method using the grasshopper optimization algorithm (GOA) called CTTRG in WSNs. This routing scheme includes a distributed time-variant trust (TVT) model to analyze the behavior of sensor nodes according to three trust criteria, including the black hole, sink hole, and gray hole probability, the wormhole probability, and the flooding probability. Furthermore, CTTRG suggests a GOA-based trusted routing tree (GTRT) to construct secure and stable communication paths between sensor nodes and base station. To evaluate each GTRT, a multi-objective fitness function is designed based on three parameters, namely the distance between cluster heads and their parent node, the trust level, and the energy of cluster heads. The evaluation results prove that CTTRG has a suitable and successful performance in terms of the detection speed of malicious nodes, packet loss rate, and end-to-end delay.

1 Introduction

Wireless sensor network (WSN) is an important element for designing Internet of Things (IoT). It includes sensor nodes, which monitor the environment to gather data and send it to the base station [1, 2]. In a WSN-based IoT network, intelligent routing is a main and necessary phenomenon for improving the quality of service (QoS) [3, 4]. Furthermore, providing the energy needed for communications is a major challenge to decrease packet loss. In the routing process, it is necessary to prevent the rapid discharge of sensor nodes and unbalanced energy distribution in the network [5, 6]. Hence, it is essential to manage the energy consumed by nodes using intelligent machine learning techniques, metaheuristic algorithms, or other optimization strategies to make effective routing decisions and improve network performance [7, 8]. Many energy-efficient routing approaches are currently available in the literature for WSNs. However, they need to be enhanced for a WSN-based IoT environment [9, 10].

Today, IoT has improved universal access for deploying intelligent networks. A network edge provides intelligent services and computing for IoT devices [11, 12]. Additionally, this deployment improves user's experience and presents efficient and flexible services when any unpleasant event occurs. Edge computing provides fast response and high-quality services because it utilizes an architecture close to end users [13, 14]. However, there are some security concerns, and security protocols must protect the network from the vulnerability of attacks (VoA). Existing techniques are mainly designed to infer intrusions in the network [15, 16]. However, determining secure and valid sensor nodes is a challenging issue in recent research. Moreover, attackers continuously change their locations to do their hostile activities around the network [17, 18]. Trust is an important component of cybersecurity. This component is responsible for determining the security level of sensor nodes during their interaction with each other. It actively identifies trusted nodes and prevents security risks caused by privacy violations, data manipulation or deletion, and other cybersecurity attacks [19, 20]. This illustrates the importance and necessity of trusted routing protocols.

Because of the special characteristics of sensor nodes, like small size, limited memory capacity, constrained energy source, and low computing power, energy consumption management is essential when designing a trust mechanism. Security and energy efficiency are two very important concepts in WSN-based IoT networks. However, they contradict each other. The deployment of these networks in vulnerable and unfriendly environments has led to their vulnerability to various attacks. Hence, existing routing protocols in WSNs need to implement powerful security mechanisms to secure the data transmission process. Although, the design of these mechanisms is associated with complex calculations that lead to high memory and energy consumption. Solving the challenges mentioned above is our main motivation for designing a trust-aware, energy-efficient, and lightweight routing algorithm. In this paper, a cluster-tree-based trusted routing approach using the grasshopper optimization algorithm (GOA) called CTTRG for WSNs is introduced. In addition to focusing on energy efficiency, the proposed scheme attempts to neutralize several routing attacks, especially black hole attack (BH), sinkhole attack (SH), wormhole attack (WH), gray hole attack (GH), and flooding attack (FA). To ensure security, CTTRG proposes a distributed time-variant trust (TVT) model to evaluate the trust of sensor nodes in the network. Also, to ensure energy efficiency, CTTRG uses a tree-cluster hierarchical topology to determine data transmission paths to the base station. In CTTRG, a technique to construct a GOA-based trusted inter-cluster routing tree (GTRT) is presented. BS is responsible for building this routing tree. In summary, the most important contributions of CTTRG are as follows:

- Designing a time-variant trust model based on three trust criteria, namely BH, SH, and GH probability, WH probability, and FA probability, and analyzing the behavior of sensor nodes when cooperating with each other.
- Adding a weight coefficient to the recommendations provided by the recommender nodes. The weight of each recommendation is determined according to the trust level of the recommender nodes and the difference between the recommended trust and the calculated direct trust.

- Constructing a trusted routing tree based on the grasshopper optimization algorithm to form stable and trusted communication paths between the cluster heads (CHs) and the base station (BS).
- Designing a multi-objective fitness function according to the distance between each CH and its parent node, the trust level of each CH, and its energy.

In the following, the organization of this paper is as follows: Section 2 exhibits the most important trusted routing methods in WSNs. In Section 3, the grasshopper optimization algorithm is presented in summary. Section 4 discusses network settings, energy model, and threat model in CTTRG. In Section 5, our method is introduced in detail. Section 6 describes the simulation and evaluation results. Finally, the conclusion of the paper is stated in Section 7.

2 Related works

In [21], a trust-aware and energy-efficient routing protocol called TBSEER has been proposed. It obtains the comprehensive trust of each sensor node with regard to three criteria, including adaptive direct trust, indirect trust, and energy. TBSEER counteracts BH, GH, SH, WH, and FA attacks. Furthermore, this approach presents an adaptive punishment structure to detect malicious nodes quickly. After obtaining direct trust based on the mentioned criteria, the sink is responsible for extracting indirect trust. In this case, sensor nodes conserve their energy because they do not perform repeated operations to calculate this parameter. Finally, CHs find secure paths between themselves and the sink node and consider their trust values in this process. This secure routing process protects the network against various attacks. Simulation results show that this method decreases energy consumption in the network. Also, this approach detects malicious nodes quickly and resists routing attacks.

In [22], a trust-aware and energy-efficient secure routing method called TESRP is introduced for WSNs. This scheme utilizes a decentralized trust structure to separate hostile nodes from honest nodes. Furthermore, TESRP employs a multi-facet routing mechanism to decide on routing paths according to trust value, remaining energy, and the number of hops. This strategy has two main advantages, namely secure data transfer and balanced energy consumption. The evaluation results prove that TESRP is better than other routing approaches in terms of consumed energy, throughput, and network longevity.

In [23], a lightweight and attack-resistant trust-based routing scheme called TSSRM is proposed for WSNs. This approach applies a secure path selection mechanism, which considers the trust value and QoS requirements. The goal of TSSRM is to counteract routing attacks and balance energy consumption in the data transmission procedure. TSSRM designs a secure routing strategy and measures the trust of nodes. In the trust evaluation process, this scheme analyzes the behavior of nodes based on their energy and movement. Then, this scheme discovers different paths between sensor nodes. In the secure route selection mechanism, TSSRM calculates the trust of the discovered paths to choose the most prominent routes. Finally, TSSRM merges QoS requirements and the trust value using the Semiring theory. Evaluations performed in this paper confirm the performance of this scheme.

In [24], a secure energy-efficient routing scheme called ECATS is suggested in WSNs with the mobile sink. It utilizes a fuzzy C-Means and an adaptive TDMA scheduling in the clustering process. Also, it introduces a path construction operation to comfort communications and render data packets to the sink node. ECAT presents a new encryption algorithm called Neural Elliptic Galois (NEG) to provide data security and privacy in the network. Additionally, ECATS finds cluster heads based on their consumed energy in the data aggregation operation. ECATS utilizes an ant lion optimization-based TDMA scheduling to enhance energy efficiency and network reliability at the same time. The evaluation results prove the successful performance of ECATS compared to other routing methods.

In [25], an energy balancing secure routing algorithm using the ant colony optimization called QEBSR is introduced for WSNs. QEBSR employs an event-oriented scenario in the data transfer procedure between nodes and BS. In addition, it utilizes an enhanced technique to calculate latency in the data transfer operation and extract the trust coefficient of nodes in the routing procedure. The ACO algorithm is responsible for searching paths using a max-min system. Eventually, the comparison of QEBSR with DEBR and EENC shows that this scheme has a successful performance.

In [26], the authors suggested a blockchain-based routing scheme in WSNs. In this approach, the blockchain technology designs a common storage capacity between sensor nodes to balance network traffic, lower interference, and enhance network security in the routing process. The authors have assumed that nodes sense events and produce a high volume of data. Hence, this data must be transferred in several packets. In this scheme, sensor nodes play the role of coins, and the transaction means the ownership exchanged between nodes and the sink node. Blockchain stores these transactions and shares the network state using a real-time manner. In the route selection procedure, this scheme introduces a cost function, which includes the load density and interference level of nodes. In addition, blockchain is responsible for protecting the discovered paths. Experiments prove that this scheme can be implemented in real-time systems.

In [27], the authors offered a cluster-based routing approach based on neuro-fuzzy rules called FBCFP to perform the routing operation in WSN-based IoT networks. FBCFP executes the network learning procedure based on the energy value, the distance from CHs to BS, the change in the cluster area, and the degree of CH. FBCFP learns the network environment using a convolutional neural network (CNN) and adjusts its initial weights using a fuzzy system. Furthermore, FBCFP employs a fuzzy system to create a strong clustering structure in the network. It considers similar sizes for clusters and utilizes the suitable rules for training the machine learning algorithm to optimize energy usage and QoS requirements in the WSN-based IoT network. According to the experiments performed in this paper, it can be found that FBCFP is well in terms of used energy, PDR, latency, and network longevity.

In [28], the authors suggested a secure routing protocol along with multiple-variant tuples. This scheme employs a symmetric cryptography strategy called Two-Fish (TF) method to detect and separate attackers on WSNs. Furthermore, this method includes an encryption mechanism and an authentication technique to provide security in the network. It utilizes Eligibility Weight Function (EWF) to find guard nodes. This function is protected using a symmetric cryptography technique. The evaluation results confirm that this scheme utilizes more monitoring nodes than other routing schemes. In addition, it deals with mobile attackers and improves packet delivery.

In [29], a cluster-based routing protocol is offered in a WSN-based IoT network. This scheme performs the routing process and the cluster head selection using two metaheuristic algorithms. The rider optimization algorithm (ROA) is employed to find cluster heads and improve QoS and reliability in the network. ROA uses a multi-objective fitness function, which depends on residual energy, distance, and delay. CHs are refreshed after certain iterations to guarantee load balancing in the network. Furthermore, the sailfish optimization algorithm (SFO) is used to select efficient and optimal routes between sensor nodes. This routing process considers several parameters namely throughput, remaining energy, and link quality. The evaluation results show that this scheme can improve execution time, energy consumption, network delay, throughput, packet delivery ratio, and network lifetime.

In [30], a trust-aware cluster-based routing algorithm is suggested in WSNs. This scheme compresses the sensed data in the data aggregation process to reduce overhead. On the other hand, this scheme implements various meta-heuristic algorithms such as artificial bee colony algorithm, ant colony optimization, differential evolution, firefly algorithm, and particle swarm optimization to validate the trust-aware routing process in WSN and make a trade-off between transmission distance, hop-count, number of transmitted messages, and trusted path. The base station has the responsibility to reconstruct the compressed data and check the trust of CHs. Moreover, CHs perform compressed sensing and trust-based data aggregation operations. These operations enhance security and limit overhead in each CH. In this scheme presents an objective function, which minimizes the distance traveled, number of hops, and number of messages and maximizes the trust related to the path.

In [31], a trust-aware routing method (TARM) is presented for WSN-based IoT networks. This scheme utilizes a mobile edge node to receive data from valid nodes. The edge node separates abnormal nodes from normal nodes based on a trust evaluation method. TARM performs the clustering process using a gray wolf optimizer and obtains trust values for each cluster. Then, the edge node receives data packets only from normal nodes through the corresponding cluster heads. TARM uses the artificial bee colony optimization to find the most suitable routes between valid nodes and the edge node. Simulations show that the trust evaluation mechanism proposed in TARM provides high security and has a high detection rate and high accuracy in detecting abnormal nodes. Also, this scheme conserves energy efficiently.

In [32], a trusted clustering protocol is proposed for WSNs. This scheme offers a trust model to detect untrusted nodes. This trust model considers two trust factors namely energy trust and data trust. In addition, this scheme utilizes stochastic fractal search optimization to do the clustering process. For maximizing network lifetime and improving network security, the clustering method proposes a fitness function to choose CHs from the trusted nodes. This function depends on the remaining energy, the number of nodes, the distance to the base station, and the dissipated energy. This clustering method can make load balancing among CHs. Evaluations show the superiority of this scheme in comparison with existing protocols.

In [33], a cluster-based routing approach is presented for heterogeneous WSNs. In the clustering process, K-means algorithm and cat swarm optimization are combined to obtain a new evolutionary approach called calf search optimization algorithm (K-CSOA), which is used to create clusters in the network. In the clustering process, K-CSOA presents a fitness function, which considers six factors, namely node degree, distance from cluster members to CHs, distance from CHs to BS, average and remaining energy, and balancing factor for clusters. The routing process uses ant colony optimization (ACO) to find the most suitable paths in the network. Simulations performed in this paper show the effectiveness of K-CSOA in terms of energy consumption and delay.

Table 1 compares our proposed scheme with the related works.

3 Basic concepts

In recent decades, optimization algorithms inspired by nature have attracted the attention of researchers and academics. These algorithms have been used in engineering, computer science, and other fields to solve complex and real-world problems. In these algorithms, a set of solutions are generated and modified at each iteration to discover the optimal solution in the search space [34, 35]. Some nature-based algorithms include Particle Swarm Optimization (PSO) [36], Artificial Bee Colony (ABC) algorithm [37], Grey Wolf Optimizer (GWO) [38], Dragonfly Algorithm (DA) [39], and Grasshopper Optimization Algorithm (GOA) [40]. In 2017, Saremi et al. presented GOA, which simulates the food search behavior of grasshoppers

Table 1. Comparison of the related works.

Method	Publication year	Security mechanism	Routing technique	Energy efficiency	Strengths	Weakness
TBSEER [21]	2022	An adaptive trust mechanism based on a punishment factor	A trust-aware clustering routing protocol	✓	Considering the energy trust value in the trust evaluation mechanism, using an adaptive punishment factor to calculate the direct trust value, high accuracy, and high detection speed for identifying hostile nodes	Selecting CHs only based on their trust value
TESRP [22]	2016	A decentralized trust structure based on Beta probability density function	AODV protocol by considering a combination of nodes' trust, remaining energy, and hop counts	✓	Detecting and isolating hostile nodes, scalability, considering energy parameter in the routing protocol	High routing overhead, high delay in the route discovery process
TSSRM [23]	2017	A trust evaluation mechanism based on Analytic hierarchy process (AHP)	An enhanced GPSR algorithm based on the trust degree and other QoS requirements	✓	Taking into account energy metric in the trust evaluation mechanism, executing many experiment scenarios	Falling into the local minimum, not considering a clustering process
ECATS [24]	2018	A NEG encryption algorithm and a fault node detection model	A clustering method based on fuzzy C-means and ant lion optimization	✓	Considering energy metric in the CH selection process	Not defining a routing process between CHs, not evaluating the resistance of this method against various attacks
QEBSR [25]	2019	A trust evaluation mechanism based on the packet drop rate and the packet generation rate	An ant colony optimization-based routing protocol	✓	Balanced energy consumption in the network, considering QoS requirements such as delay in the routing process	In some cases, the determination of weight vectors are not possible or very difficult, not evaluating the resistance of this scheme against various attacks, high time complexity
Lazrag et al. [26]	2019	Blockchain	Deciding on routing paths based on a cost function	×	Balancing traffic load, reducing interferences, and increasing security in the network	Not considering energy efficiency in the routing process
FBCFP [27]	2019	×	A cluster-based routing approach based on neuro-fuzzy rules	✓	Considering energy metric in the routing process, improving energy consumption in the network	High time complexity
Deebak and Al-Turjman [28]	2019	Designing an authentication mechanism and applying symmetric key approaches	A hybrid routing scheme based on OLSR and AOMDV	×	Ability to act as proactive and reactive routing protocol, detecting, preventing, and isolating hostile nodes	Not considering energy parameter in the routing process
Joshi and Raghuvanshi [29]	2021	×	A CH selection process based on the ROA algorithm and a routing process based on the SFO algorithm	✓	Making load balancing in the network, considering energy in the routing and clustering processes	High time complexity
Gilbert et al. [30]	2019	A beta-based trust evaluation system	A K-means-based clustering method and a routing protocol based on meta-heuristic algorithms	×	Low routing overhead, employing compressed sensing and data aggregation techniques, detecting abnormal nodes	Not considering the energy parameter in the routing process
TARM [31]	2022	A trust evaluation system	A GWO-based clustering method and a ABC-based routing algorithm	✓	High detection rate and high accuracy in detecting abnormal nodes	High time complexity, not evaluating the resistance of this scheme against various attacks
Hriez et al. [32]	2021	A trust mechanism based on energy trust and data trust	A trusted clustering process based on stochastic fractal search optimization	✓	Maximizing network lifetime, improving network security, making load balancing	High time complexity

(*Continued*)

Table 1. (Continued)

Method	Publication year	Security mechanism	Routing technique	Energy efficiency	Strengths	Weakness
K-CSOA [33]	2022	×	A cluster-based routing approach based on K-means algorithm and cat swarm optimization and a ACO-based routing process	✓	Low delay, low energy consumption	Not considering a security mechanism

https://doi.org/10.1371/journal.pone.0289173.t001

in nature. Various studies have shown the use of this algorithm for solving many problems. For example, refer to [41–43]. In CTTRG, GOA is responsible for finding a secure and energy-efficient routing tree among cluster heads because the construction of a such routing tree among sensor nodes, especially in dense networks, is difficult and time-consuming. To solve this problem, GOA is chosen because it has been widely used in various fields, especially routing, and has proven its competence and effectiveness. In [44], extensive experiments have been conducted to evaluate GOA compared to other well-known algorithms such as PSO [36], Bat Algorithm (BA) [45], Flower Pollination Algorithm (FPA) [46], Cuckoo Search (CS) [47], Firefly Algorithm (FA) [48], Genetic Algorithms (GA) [49], Differential Evolution (DE) [50], and Gravitational Search Algorithm (GSA) [51]. These experiments have shown that GOA works very well and can be used to solve complex real-world problems because it can effectively balance exploration and exploitation and guide virtual grasshoppers towards the global optimum. In general, the most important advantages of GOA are high-quality exploration operations, avoidance of local optimum, and high convergence speed. The mathematical model presented in Eq 1 is used to model the behavior of grasshoppers in nature:

$$X_i = S_i + G_i + A_i \tag{1}$$

so that i is the index of grasshoppers, X_i indicates the position of grasshopper i, S_i shows the social interaction, G_i shows the gravity force, and A_i represents the direction of the wind. In order to create a random behavior, Eq 1 is written as $X_i = r_1 S_i + r_2 G_i + r_3 A_i$ where r_1, r_2, and r_3 are random coefficients in [0, 1].

$$S_i = \sum_{\substack{j=1 \\ j \neq i}}^{N} s\left(d_{ij}\right) \widehat{d_{ij}} \tag{2}$$

so that d_{ij} is the distance from grasshopper i to grasshopper j. This distance is equal to $d_{ij} = |x_j - x_i|$. Furthermore, $\widehat{d_{ij}} = \frac{x_j - x_i}{d_{ij}}$ represents a unit vector drawn from grasshopper i to grasshopper j. s is used to express social forces. It is obtained through Eq 3.

$$s(r) = fe^{\left(\frac{-r}{l}\right)} - e^{-r} \tag{3}$$

so f and l are the attraction intensity and the attractive length, respectively. Change in these parameters causes different behaviors in grasshoppers. This social interaction can be defined as attraction and repulsion. Assume that the distance between two grasshoppers is between 0 and 15. If the distance is in [0, 2.079], the social interaction is repulsion. If the distance is 2.079, the grasshoppers are in the comfort area. Also, if the distance is in [2.079, 4], the social interaction is attraction.

G is computed according to Eq 4.

$$G_i = -g\widehat{e_g} \tag{4}$$

where g displays the gravity constant, and $\widehat{e_g}$ shows a unit vector.

A is obtained from Eq 5.

$$A_i = u\widehat{e_w} \tag{5}$$

so that u is a fixed value and $\widehat{e_w}$ represents a unit vector in the wind direction. After putting S, G, and A in Eqs 1 and 6 is obtained.

$$X_i = \sum_{\substack{j=1 \\ j \neq i}}^{N} s\left(|x_j - x_i|\right)\frac{x_j - x_i}{d_{ij}} - g\widehat{e_g} + u\widehat{e_w} \tag{6}$$

where $s(r) = fe^{\frac{-r}{l}} - e^{-r}$ and N represents the number of grasshoppers.

However, this equation cannot be used to solve optimization problems because it cannot do exploration and exploitation in the search space around a response. In this mathematical model, grasshoppers reach the comfort area speedily and they cannot be concentrated at a particular point. Therefore, this model is modified as Eq 7 to obtain the new positions of grasshoppers in each iteration.

$$X_i^d = c\left(\sum_{\substack{j=1 \\ j \neq i}}^{N} c\frac{ub_d - lb_d}{2}s\left(x_j^d - x_i^d\right)\frac{x_j - x_i}{d_{ij}}\right) + \widehat{T_d} \tag{7}$$

where ub_d and lb_d indicate the upper and lower boundaries in the dimension d, respectively. $s(r) = fe^{\frac{-r}{l}} - e^{-r}$, $\widehat{T_d}$ is the best solution in the search area, and c indicates the decreasing coefficient, which lowers the comfort area, repulsion area, and attraction area. Note that S in Eq 7 is almost similar to the component S in Eq 1. However, this equation does not regard gravity force (G) and assumes that the wind (A) moves always toward $\widehat{T_d}$.

Eq 7 shows the next position of grasshoppers. It is dependent on their current position, the position of $\widehat{T_d}$, and the positions of other grasshoppers. c is an adaptive factor and has been used twice in Eq 7. The leftmost c plays the role of inertial weight in PSO. It is used to lower the motion of grasshoppers around $\widehat{T_d}$ and balance exploration and exploitation in this case. However, the second c reduces attraction, comfort, and repulsion areas. To balance exploration and exploitation, c must be reduced based on iterations. This mechanism strengthens exploitation by increasing iterations. c lowers the comfort area when increasing the number of iterations. c is obtained from Eq 8.

$$c = c\max - l\frac{c\max - c\min}{L} \tag{8}$$

where $c$max = 1, $c$min = 0.00001, l, and L are the maximum threshold, the minimum threshold, the current iteration, and the maximum number of iterations, respectively.

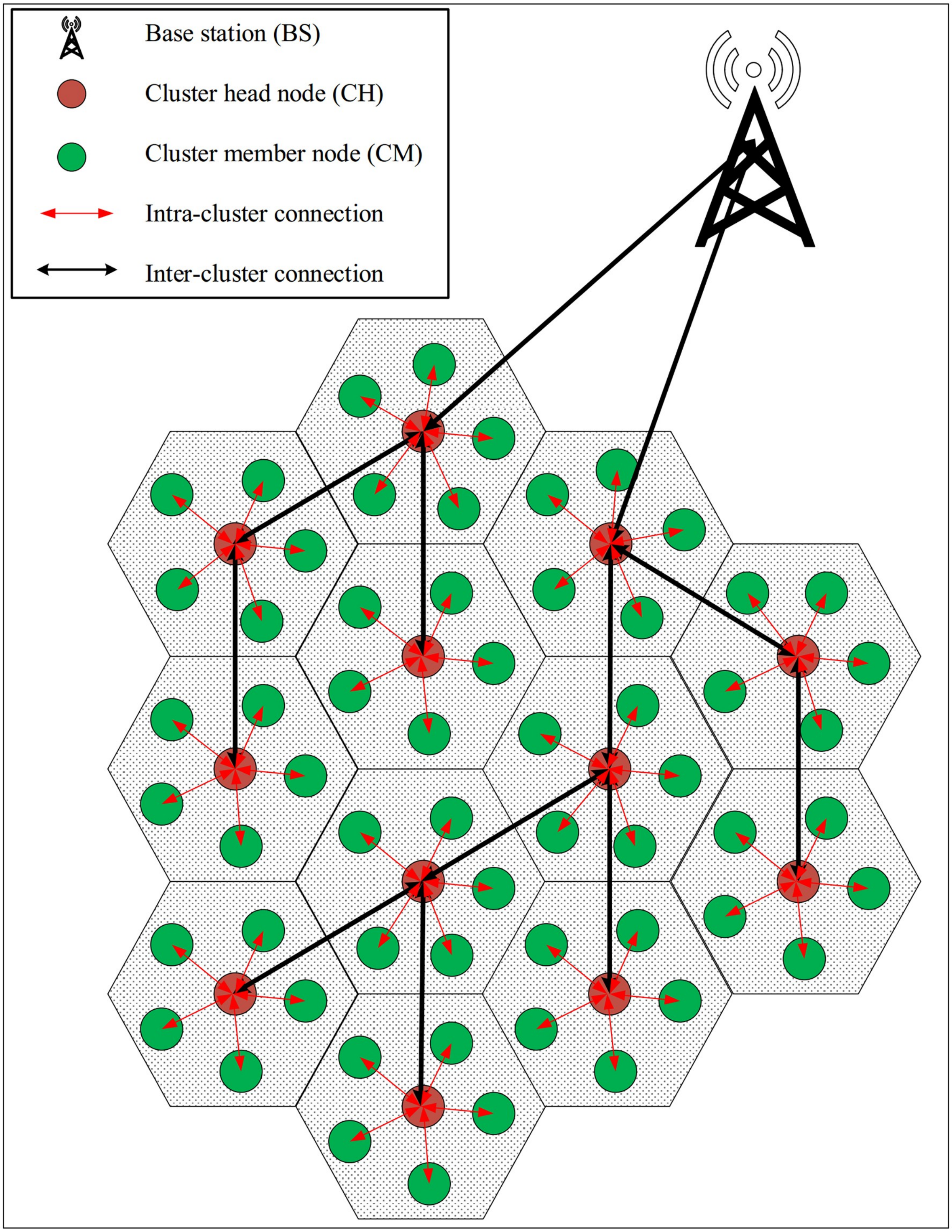

Fig 1. Network model in CTTRG.

https://doi.org/10.1371/journal.pone.0289173.g001

4 System model

The system model is formed of three items: network settings, energy consumption mechanism, and threat model.

4.1 Network settings

In CTTRG, sensor nodes (i.e. $SN_1, SN_2, \ldots, SN_i, \ldots, SN_N$ so that N is the number of nodes) have been randomly arranged in the network environment. Fig 1 displays the network model. Additionally, the nodes are partitioned into multiple clusters using the LEACH algorithm, and CHs are selected from sensor nodes rotationally. The following assumptions are summarized for the network model used in CTTRG:

- Network nodes and the BS are static.
- BS utilizes an unlimited energy source.
- Network nodes are homogeneous, meaning that they use a similar energy source.
- Some equipment installed on sensor nodes are radio communication modules and positioning devices.
- The identifier of each SN_i is unique.

4.2 Energy consumption mechanism

In CTTRG, the energy model is defined in two modes, namely free space and multi-path. To transfer k bits to SN_j, the energy used by SN_i is obtained from Eq 9.

$$E_{TX}(k,d) = \begin{cases} E_{elec} \times k + E_{fs} \times k + d^2, & d < d_0 \\ E_{elec} \times k + E_{mp} \times k + d^4, & d \geq d_0 \end{cases} \tag{9}$$

Moreover, the energy used by SN_j to receive this packet is calculated according to Eq 10:

$$E_{RX}(k,d) = E_{elec} \times k \tag{10}$$

so that $d = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ indicates the distance between SN_i and SN_j with spatial coordinates (x_i, y_i) and (x_j, y_j), respectively. E_{elec} represents the energy used for the transmitter/receiver electrical equipment. Also, E_{fs} and E_{mp} are the energy needed by an amplifier in the free space model and the multi-path model, respectively. d_0 expresses the transfer distance threshold so that $d_0 = \sqrt{\frac{E_{fs}}{E_{mp}}}$.

4.3 Attack model

In WSNs, there is a need to prevent or reduce security risks caused by dynamic topology, deploying in dangerous environments, lack of a central controller, and wireless links. Trust is an important component in cybersecurity. It determines the trust level of each node when interacting with other sensor nodes [52, 53]. In fact, a security system seeks to actively identify reliable nodes and reduce security risks because these risks may violate privacy, manipulate or delete data, and provide a bed for other cybersecurity attacks. This shows the importance of a trusted routing protocol [54, 55]. In this paper, CTTRG deals with routing attacks, especially black hole (BH), sinkhole (SH), wormhole (WH), gray hole (GH), and flooding attack (FA).

- The BH node communicates with other nodes and creates fake paths in the network. The goal of this communication is to prevent data packets from being delivered to the destination and delete all the packets. To build fake routes, the BH node is waiting to receive route requests (RREQs) from other network nodes. As soon as the request is received, the BH node quickly responds to the requesting node. Note that these routes are fake, and in fact, there is not any path to the desired node [56, 57]. Additionally, to increase the attractiveness of these fake routes and absorb network traffic, the BH node adjusts the parameters associated with these paths such as delay and hops in the best possible case.
- The SH node is similar to the BH node, except that the SH node is aware of the position of the sink node and tries to attract all traffic toward the sink. Then, it prevents the packets from being sent to the sink. The attack is more dangerous than BH.
- A GH node is similar to BH, except that GH is smarter. GH does not eliminate all data packets, but focuses on a particular type of packets or on a specific node and removes all packets sent to that node, in other cases, it shows a normal behavior [58, 59]. As a result, it is difficult to identify GH.
- WH attack will be carried out by two attacker nodes. These two nodes create a tunnel between themselves and encourage other nodes to use this tunnel for sending their data packets. They make this tunnel very attractive in terms of routing parameters to attract the network traffic. The attack provides a suitable bed for tracking the communications of transmitter nodes, copying data packets, manipulating the packets, or removing them.
- The FA node targets a specific node and sends a large number of fake route requests to it. Since the target node processes these requests and stores some information, its energy level is greatly reduced, and its memory overflows. Hence, the target node cannot respond to the real requests of legal nodes. Because of the constrained energy of sensor nodes, this attack causes serious damage to the network [60, 61].

5 The proposed method

In this section, the cluster-tree-based trusted routing method using the GOA algorithm (CTTRG) will be introduced for wireless sensor networks. This method includes two main mechanisms: the time-variant trust (TVT) model and the GOA-based trusted inter-cluster routing tree (GTRT). A diagram of proposed method is presented in Fig 2.

5.1 Time-variant trust (TVT) model

In a conventional trust model, the trust of nodes is periodically refreshed, but the trust value is constant in each period. Whereas, this is not true, and trust is a time-variant variable and has no fixed value in each period. Therefore, if a time-variant weight coefficient is considered for trust parameters, it can provide a more accurate estimation of the trust value. In CTTRG, a decentralized time-variant trust model is proposed to get the trust value of nodes. TVT contains three components: time-variant direct trust (TVDT), recommended trust (RT), and time-variant final trust (TVFT).

5.1.1 TVDT component. In CTTRG, the TVDT component includes an initial value and a dynamic coefficient. The initial value is dependent on the three trust criteria, namely the BH, SH, and GH probability (p_{SBG}), the WH probability (p_{WH}), and the FA probability (p_{FA}). These three criteria are defined based on the analysis of the behavior of sensor nodes when interacting with each other. Now, suppose that SN_i attempts to obtain an accurate estimation of the

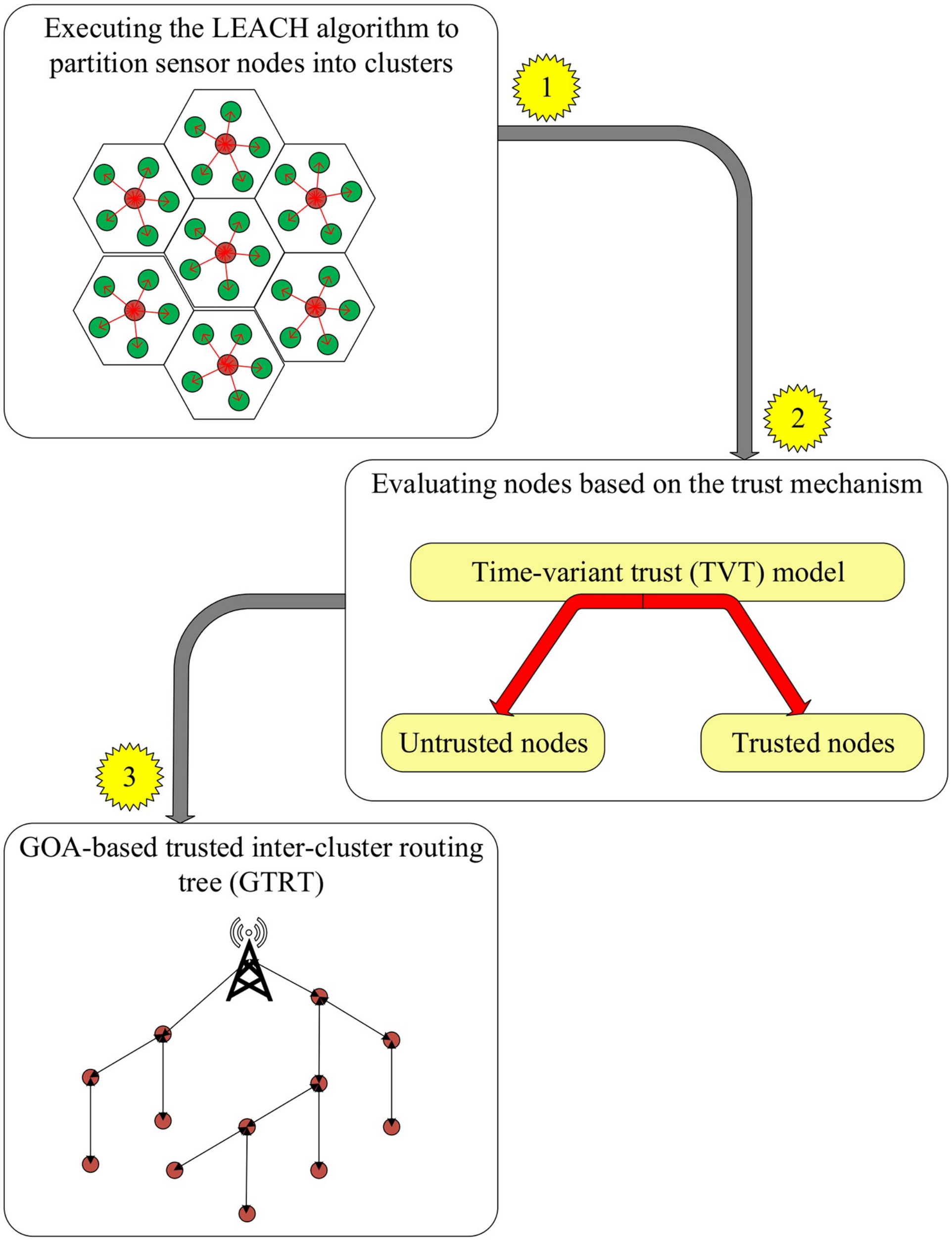
Executing the LEACH algorithm to partition sensor nodes into clusters
1
2
Evaluating nodes based on the trust mechanism
Time-variant trust (TVT) model
Untrusted nodes
Trusted nodes
3
GOA-based trusted inter-cluster routing tree (GTRT)

Fig 2. Diagram of CTTRG.

https://doi.org/10.1371/journal.pone.0289173.g002

trust value corresponding to SN_j. To achieve this goal, SN_i interacts directly with SN_j to acquire three criteria p^j_{SBG}, p^j_{WH}, and p^j_{FA}.

- $\mathbf{p^j_{SBG}}$: This criterion examines the possibility that SN_j is a SH node, a BH node, or a GH node. These three attacks are very similar to each other and have little difference, which was discussed in Section 4.3. The most important feature of the SH, BH, and GH nodes is that they have very low packet reception and sending rates and delete all (or more) data packets.

Therefore, p^j_{SBG} is obtained according to Eq 11.

$$p^j_{SBG} = \lambda\left(1 - \frac{PK^{received}_j}{PK^{total-receiving}_j}\right) + (1-\lambda)\left(1 - \frac{PK^{sent}_j}{PK^{total-sending}_j}\right) \tag{11}$$

where $PK^{received}_j$ indicates the number of packets received by SN_j and $PK^{total-receiving}_j$ is the total number of packets that should be received by SN_j. Moreover, PK^{sent}_j expresses the number of packets sent by SN_j and $PK^{total-sending}_j$ indicates the total number of packets, which should be sent by SN_j. λ is also a fixed number adjusted in [0, 1]. λ expresses the weight associated with the packet reception rate and determines the relative importance of the packet reception rate and the packet sending rate. This weight can be adjusted based on the requirements of the application.

- $\mathbf{p^j_{WH}}$: This criterion examines the possibility that SN_j is a WH node. The most important feature of WH nodes is that they tend to form various paths and absorb the traffic of the surrounding nodes. This tendency to absorb network traffic causes congestion in the WH nodes and consequently they experience a very long queuing delay. The second feature of these nodes is low package reception rate because they eliminate many received packets. Another feature of WH nodes is that they copy some data packets and relay the duplicated packets on the network. Therefore, they experience high redundancy rate. Finally, p^j_{WH} is defined in Eq 12:

$$p^j_{WH} = \Psi_1\left(\frac{T^Q_j}{\max\limits_{SN_k\in N_i}\{T^Q_k\}}\right) + \Psi_2\left(1 - \frac{PK^{received}_j}{PK^{total}_j}\right) + \Psi_3\left(\frac{DPK_j}{NPK_j + DPK_j}\right) \tag{12}$$

where T^Q_j indicates a queuing delay of SN_j. This parameter is inserted into hello packets. N_i expresses the set of neighbors of SN_i. Finally, NPK_j and DPK_j describe the number of new and duplicate packets received from SN_j, respectively. In addition, Ψ_1 is the weight coefficient related to the delay parameter, Ψ_2 is the weight coefficient related to the packet reception rate, and Ψ_3 is the weight coefficient related to the redundancy rate such that Ψ_1, Ψ_2, and Ψ_3 are fixed numbers in [0, 1] and $\sum\limits_{i=1}^{3}\Psi_i = 1$. These weights show the importance of these factors and can be set in accordance with the requirements of the application.

- $\mathbf{p^j_{FA}}$: This criterion examines the probability that SN_j is a FA node. The most important feature of FA nodes is high energy consumption, and the second feature is high route request sending rate. Another feature of these nodes is a large number of duplicate packets. According to the mentioned points above, Eq 13 calculates p^j_{FA}.

$$p^j_{FA} = \ell_1\frac{\left(\dfrac{E^{res,t-1}_j - E^{res,t}_j}{E_{ini}}\right)}{\Delta t} + \ell_2\left(\frac{\dfrac{PK^{sent}_j}{\max\limits_{k\in N_j\ and\ SN_j}\{PK^{sent}_k\}}}{\Delta t}\right) + \ell_3\left(\frac{DPK_j}{NPK_j + DPK_j}\right) \tag{13}$$

where ℓ_1 is the weight coefficient related to the energy factor, ℓ_2 is the weight coefficient related to the packet sending rate, and ℓ_3 is the weight coefficient related to the redundancy rate such that ℓ_1, ℓ_2, and ℓ_3 are fixed numbers in [0, 1] and $\sum\limits_{i=1}^{3}\ell_i = 1$. These weights show the importance of these factors and can be set in accordance with the requirements of the application. $E_j^{res,t}$ and $E_j^{res,t-1}$ represent the remaining energy of SN_j in two moments t and $t - 1$, respectively. E_{ini} is the initial energy of sensor nodes. $E_j^{res,t}$ is defined in Eq 14.

$$E_j^{res,t} = E_{ini} - EC_j^t \tag{14}$$

where EC_j^t expresses the energy consumption of SN_j at the moment t. It is obtained from Eq 15 according to the energy model stated in Section 4.2.

$$EC_j = \sum_{x=1}^{n_{EC}} \left(E_{tx}^j + E_{rx}^j \right) \tag{15}$$

where E_{tx}^j, E_{rx}^j, and n_{EC} express the energy needed to send the packets, the energy needed to receive the packets, and the number of data transfer operations performed by SN_j, respectively.

Finally, the initial value of the TVDT component in $[t - 1, t]$ (i.e. $TVDT_{ij}(t - 1)$) is defined in Eq 16:

$$TVDT_{ij}(t-1) = 1 - \max\left\{ p_{SBG}^j, p_{WH}^j, p_{FA}^j \right\} \tag{16}$$

Now, $TVDT_{ij}(t)$ will be calculated by Eq 17.

$$TVDT_{ij}(t) = TVDT_{ij}(t-1)e^{-\rho t}, \quad [t-1, t] \tag{17}$$

so that $e^{-\rho t}$ is the time-variant dynamic coefficient. In this coefficient, ρ is equal to the standard value of $TVDT_{ij}(t - 1)$, which is obtained from Eq 18.

$$\rho = \frac{TVDT_{ij}(t-1) - \mu_{TVDT}}{\sigma_{TVDT}} \tag{18}$$

where μ_{TVDT} and σ_{TVDT} are the mean and standard deviation of $TVDT_{ij}(t)$ calculated by Eqs 19 and 20.

$$\mu_{TVDT} = E(TVDT_{ij}) = \int_{t=0}^{t-1} tTVDT_{ij}(t)dt \tag{19}$$

$$\sigma_{TVDT} = \sqrt{E(TVDT_{ij}^2) - (E(TVDT_{ij}))^2} \tag{20}$$

5.1.2 The RT component.

In this section, the recommended trust component in the TVT model will be introduced. RT represents that SN_i not only relies on its interactions to calculate the trust of SN_j, but also uses the trust values recommended by the recommended nodes (RN_k). In TVT, RN_k is a common node between SN_i and SN_j, and $R = \{RN_1, RN_2, \ldots, RN_k, \ldots, RN_{|R|}\}$ is a set that contains all RN_k nodes. In TVT, SN_i considers a weight coefficient $CT_{ik}(t - 1)$ for accepting the trust recommended by each RN_k. This coefficient expresses the importance of the recommendation provided by RN_k. It includes two criteria and is obtained according to Eq 21:

- Initial trust of $\mathbf{SN_i}$ relative to $\mathbf{RN_k}$ ($\mathbf{TVDT_{ik}(t - 1)}$): According to this criterion, SN_i does not consider the recommendation provided by an unreliable RN_k.

- The difference between the trust recommended by $\mathbf{RN_k}$ and the trust calculated by $\mathbf{SN_i}$: According to this criterion, SN_i prefers the RN_k nodes that the TVDT calculated by them is closer to TVDT calculated by SN_i.

$$CT_{ik}(t-1) = TVDT_{ik}(t-1)\left(1 - \frac{|TVDT_{ij}(t-1) - TVDT_{kj}(t-1)|}{\max\limits_{RN_k \in R}\{TVDT_{kj}(t-1)\}}\right) \tag{21}$$

According to the above criteria, RT_{ij} is calculated according to Eq 22.

$$RT_{ij} = \frac{1}{|R|}\sum_{k \in R}^{|R|}(CT_{ik}(t-1) \cdot TVDT_{kj}(t-1)) \tag{22}$$

so that $TVDT_{kj}(t-1)$ is the initial trust value of RN_k relative to SN_j, and $|R|$ is the number of members of $R = \{RN_1, RN_2, \ldots, RN_k, \ldots, RN_{|R|}\}$.

5.1.3 TVFT component. Now, given that TVDT is a time-variant function. Therefore, TVFT is also defined as a time-variant trust function provided in Eq 23.

$$TVFT_{ij}^{t} = \alpha TVDT_{ij}(t) + (1-\alpha)RT_{ij} \tag{23}$$

so that $\alpha \in [0, 1]$ is a regulatory coefficient.

Algorithm 1 describes how to calculate the trust values of sensor nodes. The time complexity of this algorithm is calculated based on the following steps:

- Lines 1 and 2 of Algorithm 1 includes two nested *For* loops so that each loop is repeated N times.
- There is an *IF* command inside these nesting loops. It includes the following commands:
  - Lines 4 to 9 consist of 6 commands with fixed run times r_1, r_2, r_3, r_4, r_5, and r_6, respectively.
  - Line 10 contains a *For* loop, which is repeated $|R|$ times and has four commands (lines 11 to 14) with fixed run times r_7, r_8, r_9, and r_{10}, respectively.

$$T_{For}(N) = |R|(r_7 + r_8 + r_9 + r_{10}) \tag{24}$$

Suppose that there is a fixed number such as r so that $r > r_7+r_8+r_9+r_{10}$. In this case, Eq 24 is rewritten to obtain Eq 25:

$$T_{For}(N) = |R|(r_7 + r_8 + r_9 + r_{10}) < |R|(r) \tag{25}$$

  - Lines 16 and 17 are two commands with fixed execution times, r_{11} and r_{12}, respectively. Hence, the overall execution time of this *IF* is obtained from Eq 26:

$$T_{IF}(N) = r_1 + r_2 + r_3 + r_4 + r_5 + r_6 + |R|(r) + r_{11} + r_{12} \tag{26}$$

If a fixed number like p is considered:

$$T_{IF}(N) = r_1 + r_2 + r_3 + r_4 + r_5 + r_6 + |R|(r) + r_{11} + r_{12} < p|R| \tag{27}$$

According to the above, the time complexity of Algorithm 1 is calculated based on Eq 28:

$$T(N) = N^2(T_{IF}(N)) = N^2|R| \tag{28}$$

so that N indicates the number of sensor nodes and $|R|$ represents the number of recommender nodes.

Algorithm 1 Time variant trust model (TVT model)

```
Input: $SN_1$, $SN_2$, ..., $SN_i$, ..., $SN_N$: Sensor nodes in the network
Output: $TVFT_{ij}^{t}$: Time variant final trust of $SN_j$ estimated by $SN_i$.
Begin
for $i$ = 1 to $N$
  for $j$ = 1 to $N$ do
    if $i \neq j$ AND $SN_i$ and $SN_j$ are neighbors then
      $\mathbf{SN_i}$: Calculate $p_{SBG}^{j}$ using Eq 11;
      $\mathbf{SN_i}$: Evaluate $p_{WH}^{j}$ using Eq 12;
      $\mathbf{SN_i}$: Obtain $p_{FA}^{j}$ from Eq 13;
      $\mathbf{SN_i}$: Calculate $TVDT_{ij}(t-1)$ using Eq 16;
      $\mathbf{SN_i}$: Get $\rho$ in accordance with Eq 18;
      $\mathbf{SN_i}$: Achieve $TVDT_{ij}(t)$ based on Eq 17;
      for $k$ = 1 to $|R|$ do
        $\mathbf{SN_i}$: Assess $TVDT_{ik}(t-1)$ according to Eq 16;
        $\mathbf{RN_k}$: Compute $TVDT_{kj}(t-1)$ based on Eq 16;
        $\mathbf{SN_i}$: Obtain the difference between $TVDT_{ij}(t-1)$ and
        $TVDT_{kj}(t-1)$;
        $\mathbf{SN_i}$: Calculate the weight coefficient $CT_{ik}(t-1)$ according
        to Eq 21;
      end for
      $\mathbf{SN_i}$: Compute $RT_{ij}$ by Eq 22;
      $\mathbf{SN_i}$: Obtain $TVFT_{ij}^{t}$ from Eq 23;
    end if
  end for
end for
End
```

5.2 GOA-based trusted inter-cluster routing tree (GTRT)

In CTTRG, a GTRT tree is formed on the network to establish reliable connections between CHs and BS. BS uses the GOA algorithm to build a GTRT tree. It acquires information related to each CH node, for example, the distance to the BS, the trust level, and energy through the periodic exchange of hello messages. Furthermore, it puts all CHs in a set such as $TR = \{CH_1, CH_2, \ldots, CH_q, \ldots, CH_Q\}$ (so that Q is the number of CHs in the network). In the routing tree construction issue, each grasshopper acts as a GTRT tree and specifies the routing path between each CH and BS. The following steps are executed to find the best GTRT tree:

- **Population formation:** Each grasshopper plays the role of a GTRT tree and specifies the arrangement of CHs in the tree. This grasshopper is shown as an array with Q elements so that each element of this array represents the spatial coordinates of CH. In the population formation process, a CH is randomly chosen from the TR set, and its spatial coordinates are inserted into the relevant element of the array.
- **GTRT tree corresponding to each grasshopper:** This step contains four stages to extract a GTRT tree from a grasshopper:
  - **Stage 1:** In all grasshoppers, BS corresponds to the root of the GTRT tree.
  - **Stage 2:** In each grasshopper, the first and second elements of the array are the left and right children of BS namely LP and RP, respectively.

  - **Stage 3:** Note that GTRT is a binary tree, and at each level of this tree, the leftmost parent first identifies its left and right children based on the relevant array. For example, the third and fourth elements of the array are known as left and right children of *LP*, and the fifth and sixth elements of the array are known as left and right children of *RP*.
  - **Stage 4:** Stage 3 is repeated to join all CHs to the relevant tree.

- **Evaluation:** In the GTRT tree construction algorithm, a multi-objective fitness function is considered to evaluate GTRT trees. Then, the positions of grasshoppers will be updated based on this fitness function in each iteration. The purpose of this update process is to change the positions of cluster heads in the routing tree and build the most suitable GTRT tree. To achieve this goal, the GTRT tree construction algorithm considers a multi-objective strategy so that the GTRT tree is built based on three factors, i.e. the distance between CHs and their parent node, the remaining energy of the cluster heads, and their trust level. In this regard, GTRT trees are evaluated in accordance with the fitness function in Eq 29.

$$F_{fitness} = \beta f_1 + (1-\beta) f_2 \tag{29}$$

so that β is a fixed number in [0, 1] that determines the effect of f_1 and f_2 on $F_{fitness}$. After this evaluation, BS identifies the best response $(\widehat{T_d})$ in the population.

In the GTRT problem, the BS is looking for a tree in which the distance between each CH to its parent is short. The reason behind the selection of this factor is that in the data transmission process between a cluster head node and the base station, if the distance between each cluster head and its parent node is the shortest, this cluster head will transmit data packets to its parent node in the GTRT tree at a high speed (less delay). As a result, it will consume less energy in the data transfer process. Hence, f_1 focuses on the distance of each CH to its parent and is calculated through Eq 30.

$$f_1 = \frac{1}{\sum_{i=1}^{Q} d(CH_i, Parent_i)} \tag{30}$$

where $d(CH_i, Parent_i) = \sqrt{(x_i - x_p)^2 + (y_i - y_p)^2}$. Also, (x_i, y_i) and (x_p, y_p) express the coordinates of CH_i and its parent ($Parent_i$), respectively.

On the other hand, energy is a very effective factor on network performance because the energy of cluster heads is dropped after performing several data transmission processes. Therefore, if the cluster head nodes with less energy are placed in the higher levels of the GTRT tree, their energy will be depleted faster because the nodes placed in the higher levels of the GTRT tree must transmit more data packets, as a result, their energy consumption will be higher. Note that in addition to sending the data related to their cluster members, these nodes must also transmit the data received from the cluster heads in their subtree to the base station. Also, BS considers the trust level of CHs in the fitness function to build the most secure GTRT tree among the cluster head nodes. The meaning of the most secure routing tree is that nodes with higher trust level are placed in the higher levels of the GTRT tree because as mentioned above, these nodes have more responsibilities and their security is more important than the nodes in the lower levels of the GTRT tree. Hence, the BS is looking for a tree that puts the more secure and high-energy CHs at the higher level of GTRT. As a result, f_2 focuses on the order of CHs in GTRT based on their energy and trust through Eq 31.

$$f_2 = \sum_{D=1}^{\lfloor \log\ Q \rfloor} \frac{1}{D} \sum_{x=1}^{2^D} \left(\partial \left(\frac{E^x_{res,t} - E\min}{E_{ini} - E\min} \right) + (1-\partial) \left(\frac{TVFT_x(t) - \min\limits_{CH_k \in TR} \{TVFT_k(t)\}}{\max\limits_{CH_k \in TR} \{TVFT_k(t)\} - \min\limits_{CH_k \in TR} \{TVFT_k(t)\}} \right) \right) \quad (31)$$

where $E^x_{res,t}$ describes the remaining energy of CH_x, $E_{\min} = 15\% E_{ini}$ is the minimum energy threshold, and E_{ini} indicates the primary energy of the network nodes. Furthermore, $TVFT_x(t)$ is the trust of CH_x, and D indicates the tree depth, and ∂ is a fixed number in $[0, 1]$.

- **End condition:** This stage specifies the stopping condition of the GTRT algorithm, so that the GTRT algorithm is run on 300 iterations, and the optimized GTRT is determined at the final iteration. Finally, BS informs the status of CHs in GTRT by sending a GTRT message that includes the arrangement of CHs in the routing tree.
- **Grasshopper updating operation:** The position of CHs in the relevant grasshopper will be refreshed using Eq 7.

Algorithm 2 explains how to build a GTRT tree. Time complexity of Algorithm 2 is obtained based on the following steps:

- Line 1 contains a command with a constant execution time c_1.
- Line 2 is a *While* loop and emphasizes that Algorithm 2 is repeated throughout the simulation period (i.e. t_{sim}).
- In Line 3, there is an *IF* condition inside this *While* loop.
  - Inside this *IF* command, there is a *For* loop (lines 4–7). This loop is repeated Q times and includes two commands (Lines 5 and 6) with fixed run times c_2 and c_3.

    $$T_{For}(N) = Q(c_2 + c_3) \quad (32)$$

    If we consider a fixed number such as c so that $c > c_2 + c_3$. In this case, Eq 32 is rewritten to obtain Eq 33:

    $$T_{For}(N) = Q(c_2 + c_3) < Q(c) \quad (33)$$

    Therefore, the overall execution time of this *IF* is obtained from Eq 34:

    $$T_{IF}(N) = Q(c) \quad (34)$$

- In line 9, there is an *IF* command that includes the following commands:
  - Lines 10, 11 and 12 have three commands with fixed run times c_4, c_5, and c_6.
  - Time complexity of Line 13 is equal to $O(Q)$.
  - Time complexity of Line 14 is equal to $O(Q)$.
  - Time complexity of Line 15 depends on the fitness function presented in Eq 29. Its time complexity is equal to $O(Q)$.

- Line 16 is executed at a fixed time c_6.
  Therefore, the run time of this *IF* is obtained from Eq 35:

$$T_{IF}(N) = c_4 + c_5 + c_6 + 3O(Q) + c_6 \tag{35}$$

  There is a fixed number such as *h*, which meets the following condition:

$$T_{IF}(N) = c_4 + c_5 + c_6 + 3Q + c_6 < hQ \tag{36}$$

- In Line 17, a *While* loop is repeated 300 times (It is the end condition of the GOA algorithm. Generally, it is displayed as *K*).
  - Lines 18 and 19 contain two commands with fixed run times a_1 and a_2, respectively.
  - Line 20 is dependent on the number of grasshoppers (for example, *PG*).
  - Time complexity of Line 21 is equal to $O(Q)$.
  - Time complexity of Line 22 depends on the fitness function and is equal to $O(Q)$.
  - Time complexity of Line 23 is determined based on the number of grasshoppers.
    Therefore, the run time of this *While* is obtained from Eq 37:

$$T_{While}(N) = K(a_1 + a_2 + 2PG + 2Q) \tag{37}$$

    IF $PG < Q$ and there is a fixed number such as *a*, the run time of this *While* is calculated based on Eq 38:

$$T_{While}(N) = K(a_1 + a_2 + 2PG + 2Q) < a(KQ) \tag{38}$$

- Line 25 has a fixed runtime.

According to the points mentioned above, the time complexity of Algorithm 2 is $O(KQ)$, so that *K* is equal to the end condition and *Q* is the number of cluster heads.

Algorithm 2 GOA-based trusted routing tree (GTRT)

```
Input: $TR = \{CH_1, CH_2, \ldots, CH_q, \ldots, CH_Q\}$: Cluster head nodes
  BS: Base station
  $t_{sim}$: Simulation time
  $t_{hello}$: Hello message time period
  $t_{couter}$: Timer
Output: The best GTRT
    Begin
1: $t_{counter} = 0$;
2: while $t_{counter} \leq t_{sim}$ do
3:   if $t_{counter}$ mod $t_{hello} = 0$ then
4:   for $q = 1$ to $Q$ do
5:       $\mathbf{CH_q}$: Forward a Hello packet to the base station;
6:       BS: Save the position, the trust amount, and the remaining
         energy of $CH_q$ in its storage space;
7:     end for
8:   end if
9:     if CHs change in the network then
10:     BS: Determine the number of grasshoppers in the GTRT algorithm;
11:     BS: Specify cmax, cmin, and the stop condition in the GTRT
        algorithm;
        BS: Consider an array with $Q$ elements corresponding to each grasshopper;
        BS: Fulfill each element of grasshoppers with selecting CHs from the $TR$ set randomly;
        BS: Extract GTRT trees from grasshoppers;
        BS: Evaluate each GTRT tree based on fitness function presented in Eq 29;
        BS: Determine the best grasshopper and set it as $\widehat{T_d}$;
        while Stop condition is not met do
          BS: Update the coefficient $c$ using Eq 8;
          BS: Normalize the distance between grasshopper in [1, 4];
          BS: Update grasshoppers based on Eq 7;
          BS: Extract GTRT trees from grasshoppers;
          BS: Evaluate each GTRT tree based on fitness function presented in Eq 29;
          BS: Determine the best grasshopper and set it as $\widehat{T_d}$;
        end while
          BS: Extract the best GTRT from $\widehat{T_d}$;
      end if
      $t_{counter} = t_{counter} + 1$;
    end while
    End
```

6 Simulation and results

In order to analyze the performance of CTTRG, this method is run on NS2, and the experimental results of CTTRG are compared to those of TBSEER [21], TESRP [22] and TSSRM [23]. The reasons behind the selection of these methods are summarized below:

- CTTRG, TBSEER, TSSRM, and TESRP are energy-efficient methods and pay attention to the energy parameter in the routing process. In addition, CTTRG, TBSEER, and TSSRM have considered an energy parameter in the trust evaluation process.
- CTTRG, TBSEER, TSSRM, and TESRP have presented powerful and distributed trust mechanisms in their methods.
- CTTRG, TBSEER, TSSRM, and TESRP have the ability to detect and isolate hostile nodes in the network. As a result, a secure environment is provided for data transfer between trusted nodes.

Table 2. Simulation settings.

Scale	Value
Simulation software	NS2
Compared schemes	TBSEER, TSSRM, and TESRP
Routing attacks	BH, SH, WH, GH, and FA
Network dimensions	100×100m^2
The number of nodes	100
Maximum energy of nodes	1 J
Primary trust of nodes	0.5 J
Trust threshold	0.35
Control packet size	400 bits
Data packet size	4000 bits

https://doi.org/10.1371/journal.pone.0289173.t002

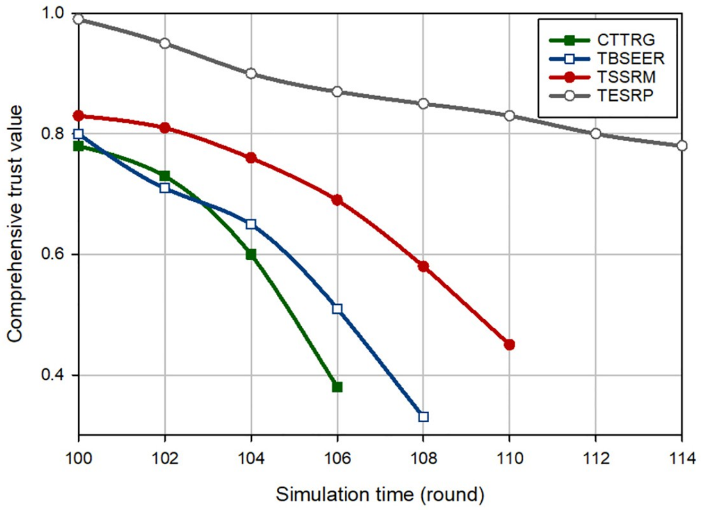

Fig 3. Comparison of the trust changes of BH nodes in different schemes.

https://doi.org/10.1371/journal.pone.0289173.g003

- TBSEER, CTTRG, and TSSRM can resist various attacks such as Wormhole, Gray hole, and Flooding.
- TBSEER and CTTRG are hierarchical methods and use the clustering technique, which will improve the energy efficiency of these methods.

In the simulation operation, various methods deal with five attacks, namely BH, SH, WH, GH, and FA, and their results are evaluated and analyzed. In this operation, there are 100 sensor nodes in a network with size 100×100m^2. Each node has energy equal to one joule, and its initial trust level is 0.5. Moreover, the trust threshold is 0.35 so, if the trust of the nodes is lower than this threshold, those nodes are marked as hostile nodes. Moreover, the sizes of control packets and data packets are 400 bits and 4, 000 bits, respectively. Table 2 states the most important simulation settings.

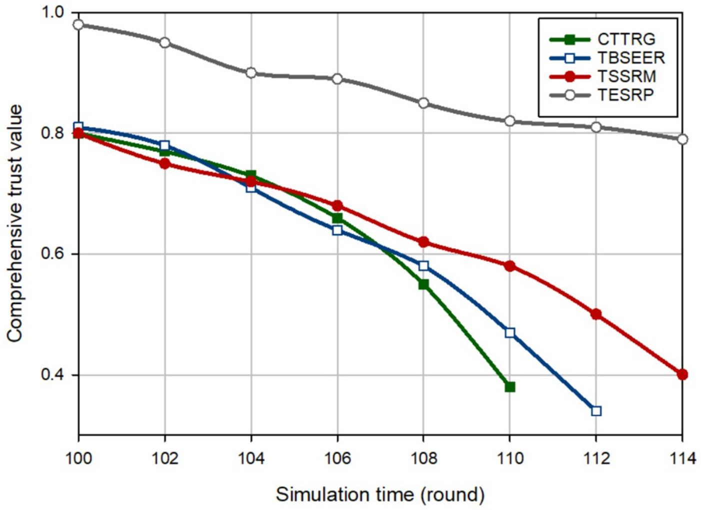

Fig 4. Comparison of the trust changes of GH nodes in different schemes.

https://doi.org/10.1371/journal.pone.0289173.g004

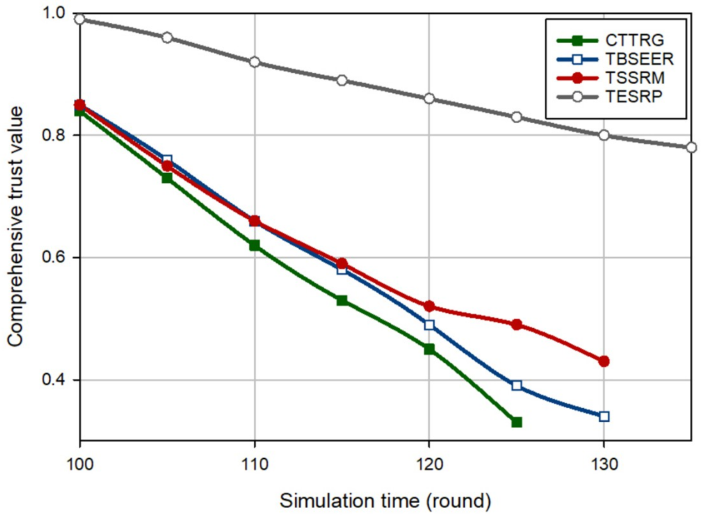

Fig 5. Comparison of the trust changes of SH nodes in different schemes.

https://doi.org/10.1371/journal.pone.0289173.g005

6.1 Trust evaluation

Fig 3 displays an evaluation of the trust of the hostile nodes (i.e. BH nodes) for different methods. In this experiment, it is assumed that in round 100, five BH nodes are injected into the network. According to Fig 3, CTTRG identifies these nodes quickly and only after seven rounds. This proves the powerfulness of the TVT model presented in CTTRG for detecting BH nodes. Among other routing schemes, TBSEER also works well so that the BH nodes have been identified and removed after 8 rounds. However, TESRP shows the weakest performance in identifying BH nodes. In Fig 4, it is assumed that five GH nodes are entered into the network in round 100. Note that it is more difficult to diagnose this attack compared with the BH attack because GH nodes are smarter and focus only on a particular type of packets and behave normally in other cases. This is well visible in Fig 4 because CTTRG detects these nodes at a slower speed and requires 11 rounds to isolate these nodes in the network. While TBSEER can

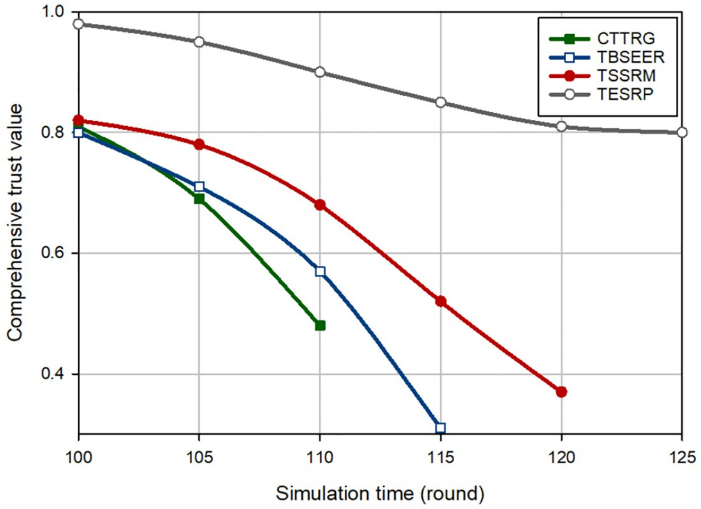

Fig 6. Trust changes of FA nodes in different schemes.

https://doi.org/10.1371/journal.pone.0289173.g006

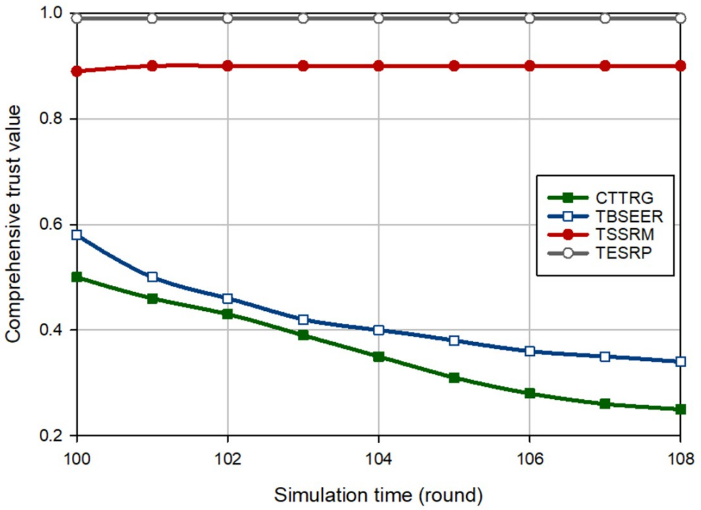

Fig 7. Trust changes of WH nodes in different schemes.

https://doi.org/10.1371/journal.pone.0289173.g007

recognize and separate GH nodes in 12 rounds. It has a good performance. In Fig 5, a SH attack occurs on the network. The detection speed of CTTRG and TBSEER is slow and close to TSSRM, so that CTTRG identifies the SH nodes after 25 rounds. However, TBSEER requires 28 rounds to detect and separate these SH nodes. The reason for the successful performance of the suggested scheme in this experiment is that the TVT system used in CTTRG contains a dynamic and time-variant coefficient. It will be reduced or increased using the historical trust values of each node. Hence, CTTRG quickly reduces the trust level of hostile nodes in each round and identifies these nodes in shorter rounds. In Fig 6, a FA attack occurs on the network, and 5% of the network nodes are hostile. All routing approaches slowly reduce the trust level of the FA nodes and detect such an attack. However, our scheme has shown the best performance in identifying this attack because CTTRG uses a parameter called the FA probability to calculate the trust value. Furthermore, it quickly detects these nodes based on the energy level change and the number of duplicate packets. Finally, in the last experiment, Fig 7

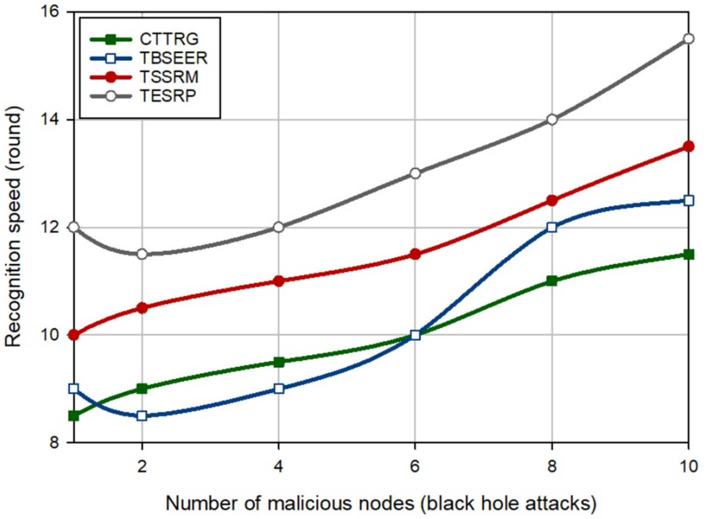

Fig 8. Comparison of the detection speed of different methods in a BH attack.

https://doi.org/10.1371/journal.pone.0289173.g008

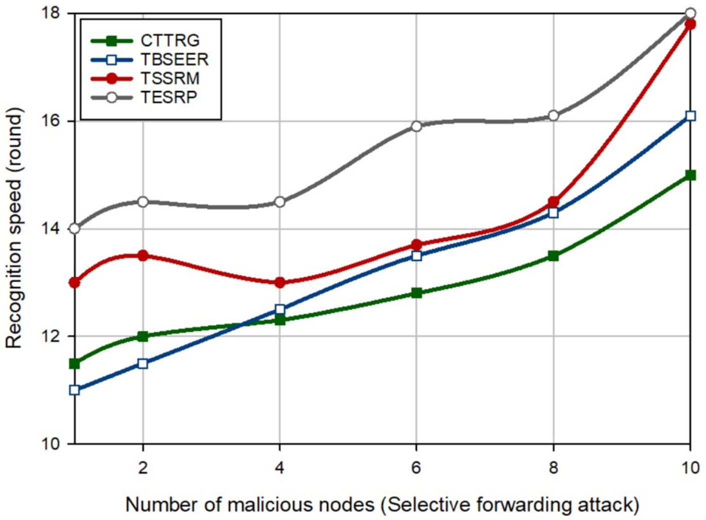

Fig 9. Comparison of the detection speed of different methods in a GH attack.

https://doi.org/10.1371/journal.pone.0289173.g009

considers the WH nodes in the network. CTTRG has identified the attack in 5 rounds and TBSEER has identified the WH nodes in 8 rounds. However, TSSRM and TESRP cannot detect this attack because they do not diminish the trust level of the WH nodes.

6.2 Detection speed

In the next experiment, the detection speed of different schemes is evaluated in the presence of several hostile nodes (between 1–10). Fig 8 shows the detection speed of different schemes for a BH attack. This figure proves that CTTRG has the best detection speed and diagnoses the BH nodes approximately 2.46%, 13.74%, and 23.69% faster than TBSEER, TSSRM, and TESRP, respectively. Moreover, Fig 9 compares the detection speed of different approaches for a GH attack. According to this figure, CTTRG has improved the detection speed of GH nodes by

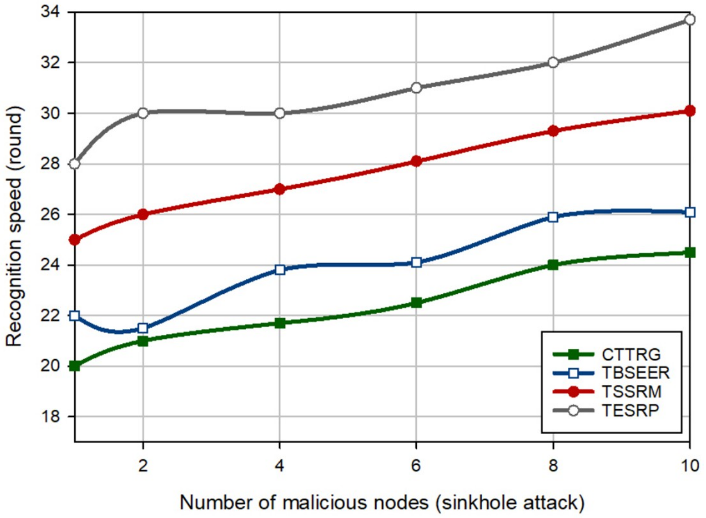

Fig 10. Comparison of the detection speed of different methods in SH attack.

https://doi.org/10.1371/journal.pone.0289173.g010

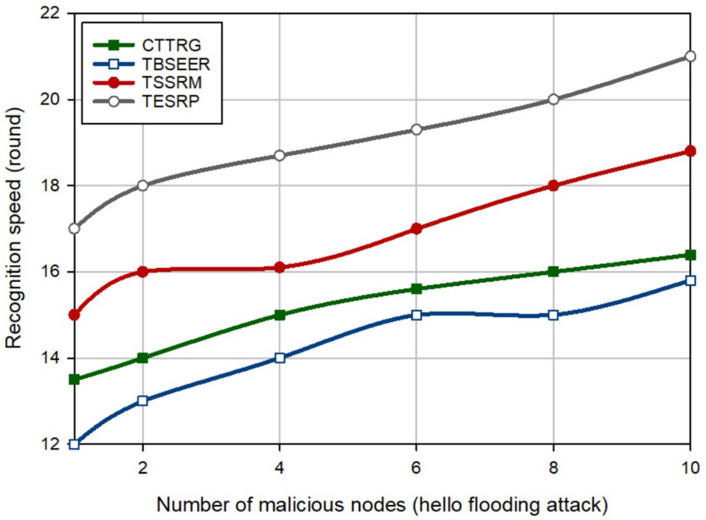

Fig 11. Comparison of the detection speed of different methods in FA attack.

https://doi.org/10.1371/journal.pone.0289173.g011

2.28%, 9.82%, and 17.097% compared to TBSEER, TSSRM, and TESRP, respectively. In addition, Fig 10 evaluates different schemes in terms of the detection speed of SH attacks. In this figure, CTTRG increases the detection speed of the SH nodes by 6.78%, 19.22%, and 27.62% in comparison with TBSEER, TSSRM, and TESRP, respectively. Finally, Fig 11 shows the performance of various schemes for identifying FA nodes. According to this figure, CTTRG has a lower speed (approximately 5.45%) than BSEER to diagnose FA nodes. However, our scheme is 10.34% and 20.63% faster than TSSRM and TESRP, respectively. Obviously, an opposite relationship is between the number of hostile nodes and the detection speed so that if a lot of hostile nodes attack the network, the detection rate will be lowered in different schemes because the nodes can participate with each other, and this decreases the accuracy of the recommendations provided by the recommended nodes. However, in CTTRG, these recommendations are

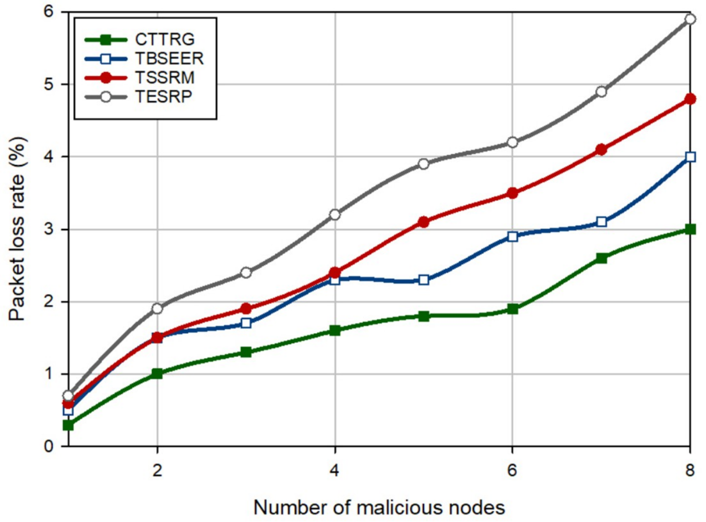

Fig 12. Comparison of PLR in different schemes.

https://doi.org/10.1371/journal.pone.0289173.g012

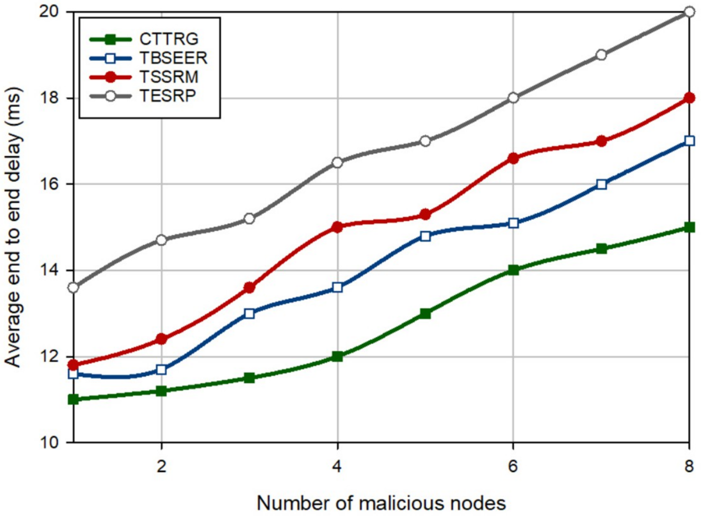

Fig 13. Comparison of delay in different methods.

https://doi.org/10.1371/journal.pone.0289173.g013

prioritized, meaning that if a recommender node is not reliable or secure, its recommendation is very different from the direct trust evaluated by the trusted nodes and hence, this recommendation has less priority than other recommendations, and has little effect on the final trust.

6.3 Packet loss rate (PLR)

PLR means the ratio of the number of packets, which do not arrive at the BS to all packets sent to BS. The PLR results in different schemes are stated in Fig 12 when there are 1–8 hostile nodes in the network. According to this figure, CTTRG has the lowest PLR and reduces it by 26.23%, 38.36%, and 50.18% in comparison with TBSEER, TSSRM, and TESRP, respectively. This is due to the powerful security mechanism designed in CTTRG, which identifies hostile nodes quickly. It will also prevent the effect of malicious nodes and reduces the number of

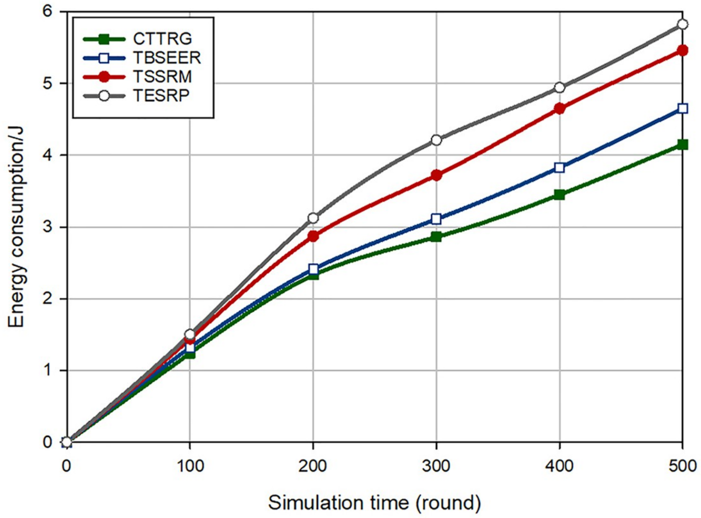

Fig 14. Comparison of energy consumption in different methods.

https://doi.org/10.1371/journal.pone.0289173.g014

missing data packets. On the other hand, three factors, namely the distance between each CH to its parent, the trust level of the network nodes, and their energy are considered when forming a GTRT tree. This causes the creation of a stable and secure tree between CHs.

6.4 Delay

Delay is the average time required to send a packet from the source node to the BS. In Fig 13, CTTRG decreases delay by 9.40%, 14.62%, and 23.73% compared to TBSEER, TSSRM, and TESRP, respectively. In CTTRG, delay is reduced in the routing process because it transfers data packets through the optimized GTRT tree. As shown in Fig 13, delay is directly proportional to the number of hostile nodes in the network, so if the number of these nodes is high, the data transfer operation is delayed in all routing methods. This is because the security systems can difficultly detect a high number of malicious nodes on the network and hence, some hostile nodes are not identified and will have a negative effect on network performance.

6.5 Energy consumption

In Fig 14, the energy consumed in different schemes is compared with each other. Based on this figure, it can be seen that CTTRG has improved energy consumption by 8.42%, 22.66%, and 28.38% compared to TBSEER, TSSRM, and TESRP, respectively. The main reason for this improvement is that CTTRG uses tree-cluster topology, which greatly increases the efficiency of our method in terms of energy consumption. On the other hand, in the GTRT tree construction process, the remaining energy of cluster heads is considered as an important factor in the fitness function. As a result, the designed GTRT tree balances the energy consumption of CHs in the network and increases network lifetime.

7 Conclusion

In this paper, a cluster-tree-based trusted routing method using the grasshopper optimization algorithm is proposed for WSNs. CTTRG contains two components: the time-variant trust mechanism and the GOA-based trusted routing tree. The TVT mechanism analyzes the behavior of sensor nodes and measures their trust level based on the three criteria, including the BH, GH, and SH probability, the WH probability, and the FA probability. Additionally, the GTRT tree is looking for safe and trust communication paths between CHs and BS. CTTRG is run on NS2 and its performance is compared with TBSEER, TSSRM, and TESRP. The experimental results show that CTTRG lowers the detection speed of BH nodes by 2.46%, 13.74%, and 23.69%, the detection speed of GH nodes by 2.28%, 9.82%, and 17.097%, and the detection speed of SH nodes by 6.78%, 19.22%, and 27.62% in comparison with TBSEER, TSSRM, and TESRP, respectively. However, CTTRG has a lower speed (approximately 5.45%) than TBSEER to diagnose FA nodes. In addition, our scheme lowers PLR by 26.23%, 38.36%, and 50.18% and delay by 9.40%, 14.62%, and 23.73% compared to TBSEER, TSSRM, and TESRP, respectively. In future research directions, we will use new techniques, for example, machine learning or meta-heuristic algorithms to enhance the strength of the trust system in CTTRG. Furthermore, GTRT tree can be constructed using different nature-based algorithms to obtain the best tree.

Acknowledgments

The result was created through solving the student project “Security analysis and developing lightweight ciphers and protocols” using objective-oriented support for specific university research from the University of Finance and Administration, Prague, Czech Republic. Authors thank Michal Merta, and Zdeněk Truhlář for their help with the research connected with the topic of the article.

Author Contributions

Formal analysis: Jan Lansky, Joon Yoo.

Investigation: Amir Masoud Rahmani.

Methodology: Mehdi Hosseinzadeh, Omed Hassan Ahmed, Mohammad Sadegh Yousefpoor.

Project administration: Mohammad Sadegh Yousefpoor.

Resources: Jan Lansky, Stanislava Mildeova, Efat Yousefpoor, Joon Yoo.

Supervision: Amir Masoud Rahmani.

Validation: Lilia Tightiz.

Writing – original draft: Mehdi Hosseinzadeh, Omed Hassan Ahmed, Mohammad Sadegh Yousefpoor, Amir Masoud Rahmani.

Writing – review & editing: Stanislava Mildeova, Efat Yousefpoor, Joon Yoo, Lilia Tightiz.

References

1. Yousefpoor M.S., Yousefpoor E., Barati H., Barati A., Movaghar A. and Hosseinzadeh M., 2021. Secure data aggregation methods and countermeasures against various attacks in wireless sensor networks: A comprehensive review. Journal of Network and Computer Applications, 190, p.103118. https://doi.org/10.1016/j.jnca.2021.103118
2. Yousefpoor M.S. and Barati H., 2019. Dynamic key management algorithms in wireless sensor networks: A survey. Computer Communications, 134, pp.52–69. https://doi.org/10.1016/j.comcom.2018.11.005
3. Jeong H., Lee S.W., Hussain Malik M., Yousefpoor E., Yousefpoor M.S., Ahmed O.H., et al. 2022. SecAODV: A secure healthcare routing scheme based on hybrid cryptography in wireless body sensor networks. Frontiers in Medicine, p.1224. https://doi.org/10.3389/fmed.2022.829055 PMID: 35935783
4. Hosseinzadeh M., Tanveer J., Masoud Rahmani A., Yousefpoor E., Sadegh Yousefpoor M., Khan F. et al. 2022. A Cluster-Tree-Based Secure Routing Protocol Using Dragonfly Algorithm (DA) in the Internet of Things (IoT) for Smart Agriculture. Mathematics, 11(1), p.80. https://doi.org/10.3390/math11010080
5. Yousefpoor M.S. and Barati H., 2020. DSKMS: a dynamic smart key management system based on fuzzy logic in wireless sensor networks. Wireless Networks, 26(4), pp.2515–2535. https://doi.org/10.1007/s11276-019-01980-1
6. Yousefpoor E., Barati H. and Barati A., 2021. A hierarchical secure data aggregation method using the dragonfly algorithm in wireless sensor networks. Peer-to-Peer Networking and Applications, 14 (4), pp.1917–1942. https://doi.org/10.1007/s12083-021-01116-3
7. Rahmani A.M., Ali S., Malik M.H., Yousefpoor E., Yousefpoor M.S., Mousavi A., et al. 2022. An energy-aware and Q-learning-based area coverage for oil pipeline monitoring systems using sensors and Internet of Things. Scientific Reports, 12(1), p.9638. https://doi.org/10.1038/s41598-022-12181-w PMID: 35688867
8. Gulati K., Boddu R.S.K., Kapila D., Bangare S.L., Chandnani N. and Saravanan G., 2022. A review paper on wireless sensor network techniques in Internet of Things (IoT). Materials Today: Proceedings, 51, pp.161–165.
9. Abdulzahra, A.M.K., Al-Qurabat, A.K.M. and Abdulzahra, S.A., 2023. Optimizing energy consumption in WSN-based IoT using unequal clustering and sleep scheduling methods. Internet of Things, p.100765.
10. Alawad, F. and Kraemer, F.A., 2022. Value of information in wireless sensor network applications and the IoT: A review. IEEE Sensors Journal.
11. Dampage U., Bandaranayake L., Wanasinghe R., Kottahachchi K. and Jayasanka B., 2022. Forest fire detection system using wireless sensor networks and machine learning. Scientific reports, 12(1), p.46. https://doi.org/10.1038/s41598-021-03882-9 PMID: 34996960

12. Tirandazi P., Rahiminasab A. and Ebadi M.J., 2022. An efficient coverage and connectivity algorithm based on mobile robots for wireless sensor networks. Journal of Ambient Intelligence and Humanized Computing, pp.1–23.

13. Spandonidis C., Theodoropoulos P., Giannopoulos F., Galiatsatos N. and Petsa A., 2022. Evaluation of deep learning approaches for oil & gas pipeline leak detection using wireless sensor networks. Engineering Applications of Artificial Intelligence, 113, p.104890. https://doi.org/10.1016/j.engappai.2022.104890

14. Hu B., Tang W. and Xie Q., 2022. A two-factor security authentication scheme for wireless sensor networks in IoT environments. Neurocomputing, 500, pp.741–749. https://doi.org/10.1016/j.neucom.2022.05.099

15. Osamy, W., Khedr, A.M., Salim, A., Al Ali, A.I. and El-Sawy, A.A., 2022. Coverage, deployment and localization challenges in wireless sensor networks based on artificial intelligence techniques: a review. IEEE Access.

16. Faris M., Mahmud M.N., Salleh M.F.M. and Alnoor A., 2023. Wireless sensor network security: A recent review based on state-of-the-art works. International Journal of Engineering Business Management, 15, p.18479790231157220. https://doi.org/10.1177/18479790231157220

17. Dai C. and Xu Z., 2022. A secure three-factor authentication scheme for multi-gateway wireless sensor networks based on elliptic curve cryptography. Ad Hoc Networks, 127, p.102768. https://doi.org/10.1016/j.adhoc.2021.102768

18. Han Y., Hu H. and Guo Y., 2022. Energy-aware and trust-based secure routing protocol for wireless sensor networks using adaptive genetic algorithm. IEEE Access, 10, pp.11538–11550. https://doi.org/10.1109/ACCESS.2022.3144015

19. Li Y. and Tian Y., 2022. A lightweight and secure three-factor authentication protocol with adaptive privacy-preserving property for wireless sensor networks. IEEE Systems Journal, 16(4), pp.6197–6208. https://doi.org/10.1109/JSYST.2022.3152561

20. Roy P.K. and Bhattacharya A., 2022. SDIWSN: A Software-Defined Networking-Based Authentication Protocol for Real-time Data Transfer in Industrial Wireless Sensor Networks. IEEE Transactions on Network and Service Management, 19(3), pp.3465–3477. https://doi.org/10.1109/TNSM.2022.3173975

21. Hu H., Han Y., Yao M. and Song X., 2021. Trust based secure and energy efficient routing protocol for wireless sensor networks. IEEE Access, 10, pp.10585–10596. https://doi.org/10.1109/ACCESS.2021.3075959

22. Ahmed A., Bakar K.A., Channa M.I. and Khan A.W., 2016. A secure routing protocol with trust and energy awareness for wireless sensor network. Mobile Networks and Applications, 21, pp.272–285. https://doi.org/10.1007/s11036-016-0683-y

23. Qin D., Yang S., Jia S., Zhang Y., Ma J. and Ding Q., 2017. Research on trust sensing based secure routing mechanism for wireless sensor network. IEEE Access, 5, pp.9599–9609. https://doi.org/10.1109/ACCESS.2017.2706973

24. Kavidha V. and Ananthakumaran S., 2019. Novel energy-efficient secure routing protocol for wireless sensor networks with Mobile sink. Peer-to-Peer Networking and Applications, 12, pp.881–892. https://doi.org/10.1007/s12083-018-0688-3

25. Rathee M., Kumar S., Gandomi A.H., Dilip K., Balusamy B. and Patan R., 2019. Ant colony optimization based quality of service aware energy balancing secure routing algorithm for wireless sensor networks. IEEE Transactions on Engineering Management, 68(1), pp.170–182. https://doi.org/10.1109/TEM.2019.2953889

26. Lazrag, H., Chehri, A., Saadane, R. and Rahmani, M.D., 2019, November. A blockchain-based approach for optimal and secure routing in wireless sensor networks and IoT. In 2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS) (pp. 411–415). IEEE.

27. Thangaramya K., Kulothungan K., Logambigai R., Selvi M., Ganapathy S. and Kannan A., 2019. Energy aware cluster and neuro-fuzzy based routing algorithm for wireless sensor networks in IoT. Computer Networks, 151, pp.211–223. https://doi.org/10.1016/j.comnet.2019.01.024

28. Deebak B.D. and Al-Turjman F., 2020. A hybrid secure routing and monitoring mechanism in IoT-based wireless sensor networks. Ad Hoc Networks, 97, p.102022. https://doi.org/10.1016/j.adhoc.2019.102022

29. Joshi P. and Raghuvanshi A.S., 2021. A Multi-Objective Metaheuristic Approach Based Adaptive Clustering and Path Selection in IoT Enabled Wireless Sensor Networks. International Journal of Computer Networks and Applications, 8(5), pp.566–584. https://doi.org/10.22247/ijcna/2021/209988

30. Gilbert E.P.K., Baskaran K., Rajsingh E.B., Lydia M. and Selvakumar A.I. (2019) 'Trust aware nature inspired optimized routing in clustered wireless sensor networks', Int. J. Bio-Inspired Computation, Vol. 14, No. 2, pp.103–113. https://doi.org/10.1504/IJBIC.2019.101637

31. Saleh A., Joshi P., Rathore R.S. and Sengar S.S., 2022. Trust-Aware Routing Mechanism through an Edge Node for IoT-Enabled Sensor Networks. Sensors, 22(20), p.7820. https://doi.org/10.3390/s22207820 PMID: 36298173

32. Hriez S., Almajali S., Elgala H., Ayyash M. and Salameh H.B., 2021. A novel trust-aware and energy-aware clustering method that uses stochastic fractal search in IoT-enabled wireless sensor networks. IEEE Systems Journal, 16(2), pp.2693–2704. https://doi.org/10.1109/JSYST.2021.3065323

33. Joshi, P., Kumar, S. and Raghuvanshi, A.S., 2022. A performance efficient joint clustering and routing approach for heterogeneous wireless sensor networks. Expert Systems.

34. Schranz M., Di Caro G.A., Schmickl T., Elmenreich W., Arvin F., Şekercioğlu A., et al., 2021. Swarm intelligence and cyber-physical systems: concepts, challenges and future trends. Swarm and Evolutionary Computation, 60, p.100762. https://doi.org/10.1016/j.swevo.2020.100762

35. Nasir M.H., Khan S.A., Khan M.M. and Fatima M., 2022. Swarm intelligence inspired intrusion detection systems—a systematic literature review. Computer Networks, p.108708. https://doi.org/10.1016/j.comnet.2021.108708

36. Wang D., Tan D. and Liu L., 2018. Particle swarm optimization algorithm: an overview. Soft computing, 22, pp.387–408. https://doi.org/10.1007/s00500-016-2474-6

37. Karaboga D., Gorkemli B., Ozturk C. and Karaboga N., 2014. A comprehensive survey: artificial bee colony (ABC) algorithm and applications. Artificial Intelligence Review, 42, pp.21–57. https://doi.org/10.1007/s10462-012-9328-0

38. Faris H., Aljarah I., Al-Betar M.A. and Mirjalili S., 2018. Grey wolf optimizer: a review of recent variants and applications. Neural computing and applications, 30, pp.413–435. https://doi.org/10.1007/s00521-017-3272-5

39. Mirjalili S., 2016. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural computing and applications, 27, pp.1053–1073. https://doi.org/10.1007/s00521-015-1920-1

40. Mirjalili S.Z., Mirjalili S., Saremi S., Faris H. and Aljarah I., 2018. Grasshopper optimization algorithm for multi-objective optimization problems. Applied Intelligence, 48, pp.805–820. https://doi.org/10.1007/s10489-017-1019-8

41. Moghanian S., Saravi F.B., Javidi G. and Sheybani E.O., 2020. GOAMLP: Network intrusion detection with multilayer perceptron and grasshopper optimization algorithm. IEEE Access, 8, pp.215202–215213. https://doi.org/10.1109/ACCESS.2020.3040740

42. Janabi S.M.A. and Kurnaz S., 2023. A new localization mechanism in IoT using grasshopper optimization algorithm and DVHOP algorithm. Wireless Networks, pp.1–21.

43. Jebi R.C. and Baulkani S., 2022. Mitigation of coverage and connectivity issues in wireless sensor network by multi-objective randomized grasshopper optimization based selective activation scheme. Sustainable Computing: Informatics and Systems, 35, p.100728.

44. Saremi S., Mirjalili S. and Lewis A., 2017. Grasshopper optimisation algorithm: theory and application. Advances in engineering software, 105, pp.30–47. https://doi.org/10.1016/j.advengsoft.2017.01.004

45. Yang, X.S., 2010. A new metaheuristic bat-inspired algorithm. Nature inspired cooperative strategies for optimization (NICSO 2010), pp.65–74.

46. Yang, X.S., 2012. Flower pollination algorithm for global optimization. In Unconventional Computation and Natural Computation: 11th International Conference, UCNC 2012, Orléan, France, September 3–7, 2012. Proceedings 11 (pp. 240–249). Springer Berlin Heidelberg.

47. Yang, X.S. and Deb, S., 2009, December. Cuckoo search via Lévy flights. In 2009 World congress on nature & biologically inspired computing (NaBIC) (pp. 210–214). Ieee.

48. Yang X.S., 2010. Firefly algorithm, stochastic test functions and design optimization. International journal of bio-inspired computation, 2(2), pp.78–84. https://doi.org/10.1504/IJBIC.2010.032124

49. Holland J.H., 1992. Genetic algorithms. Scientific american, 267(1), pp.66–73. https://doi.org/10.1038/scientificamerican0792-66

50. Storn R. and Price K., 1997. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization, 11(4), p.341. https://doi.org/10.1023/A:1008202821328

51. Rashedi E., Nezamabadi-Pour H. and Saryazdi S., 2009. GSA: a gravitational search algorithm. Information sciences, 179(13), pp.2232–2248. https://doi.org/10.1016/j.ins.2009.03.004

52. Goyal, S.B., Bedi, P., Rajawat, A.S. and Shrivastava, D.P., 2022. Secure Authentication in Wireless Sensor Networks Using Blockchain Technology. In AI-Enabled Agile Internet of Things for Sustainable FinTech Ecosystems (pp. 93–105). IGI Global.

53. Dewal, P., Narula, G.S., Jain, V. and Baliyan, A., 2018. Security attacks in wireless sensor networks: A survey. In Cyber Security: Proceedings of CSI 2015 (pp. 47–58). Springer Singapore.

54. Xie M., Han S., Tian B. and Parvin S., 2011. Anomaly detection in wireless sensor networks: A survey. Journal of Network and computer Applications, 34(4), pp.1302–1325. https://doi.org/10.1016/j.jnca.2011.03.004

55. Ezhilarasi M., Gnanaprasanambikai L., Kousalya A. and Shanmugapriya M., 2022. A novel implementation of routing attack detection scheme by using fuzzy and feed-forward neural networks. Soft Computing, pp.1–12.

56. Otair M., Ibrahim O.T., Abualigah L., Altalhi M. and Sumari P., 2022. An enhanced grey wolf optimizer based particle swarm optimizer for intrusion detection system in wireless sensor networks. Wireless Networks, 28(2), pp.721–744. https://doi.org/10.1007/s11276-021-02866-x

57. Pathak A., Al-Anbagi I. and Hamilton H.J., 2022. An Adaptive QoS and Trust-Based Lightweight Secure Routing Algorithm for WSNs. IEEE Internet of Things Journal, 9(23), pp.23826–23840. https://doi.org/10.1109/JIOT.2022.3189832

58. Ashraf, I., Park, Y., Hur, S., Kim, S.W., Alroobaea, R., Zikria, Y.B., et al. 2022. A survey on cyber security threats in IoT-enabled maritime industry. IEEE Transactions on Intelligent Transportation Systems.

59. Raimundo R.J. and Rosário A.T., 2022. Cybersecurity in the internet of things in industrial management. Applied Sciences, 12(3), p.1598. https://doi.org/10.3390/app12031598

60. Mezrag F., Bitam S. and Mellouk A., 2022. An efficient and lightweight identity-based scheme for secure communication in clustered wireless sensor networks. Journal of Network and Computer Applications, 200, p.103282. https://doi.org/10.1016/j.jnca.2021.103282

61. Tropea M., Spina M.G., De Rango F. and Gentile A.F., 2022. Security in wireless sensor networks: A cryptography performance analysis at mac layer. Future Internet, 14(5), p.145. https://doi.org/10.3390/fi14050145